One-Step Detection Paradigm for Hyperspectral Anomaly Detection via Spectral Deviation Relationship Learning

Jingtao Li, Xinyu Wang, *Member, IEEE*, Shaoyu Wang, Hengwei Zhao, and Yanfei Zhong, *Senior Member, IEEE*

***Abstract*— Hyperspectral anomaly detection (HAD) aims to find small targets deviating from surroundings in an unsupervised manner. Recently, various deep models have been applied to HAD, such as autoencoder series and generative adversarial networks (GANs) series, which mainly use a proxy task, i.e., iteratively reconstructing low-frequency components (backgrounds) to separate anomalies (two-step paradigm). However, in such an unsupervised manner, most deep HAD model is trained and tested on the same image. Since the learned low-frequency background varies from image to image and the trained model cannot be directly transferred to unseen images. In this article, the one-step detection paradigm is first proposed, where the model is optimized directly for the HAD task and can be transferred to unseen datasets. The one-step paradigm is optimized to identify the spectral deviation relationship according to the anomaly definition. Compared with learning the specific background distribution in the two-step paradigm, the spectral deviation relationship is universal for different images and guarantees transferability. Furthermore, we instantiated the one-step paradigm as an unsupervised transferred direct detection (TDD) model. To train the TDD model in an unsupervised manner, an anomaly sample simulation strategy is proposed to generate numerous pairs of anomaly samples. A global self-attention module (GAM) and a local self-attention module (LAM) are designed to help the model focus on the "spectrally deviating" relationship. The TDD model was validated on six public datasets. The results show that TDD is superior to the recent two-step methods in detection and transferability aspects.**

***Index Terms*— Anomaly detection, deep learning, hyperspectral imagery (HSI), spectral deviation, unified model.**

Manuscript received 13 January 2024; revised 11 March 2024; accepted 29 March 2024. Date of publication 22 April 2024; date of current version 13 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3903404; in part by the National Natural Science Foundation of China under Grant 42325105; and in part by the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS) Special Research Funding. *(Corresponding author: Xinyu Wang.)*

Jingtao Li, Hengwei Zhao, and Yanfei Zhong are with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Hubei Provincial Engineering Research Center of Natural Resources Remote Sensing Monitoring, Wuhan University, Wuhan 430072, China (e-mail: jingtaoli@whu.edu.cn; whu_zhaohw@whu.edu.cn; zhongyanfei@whu.edu.cn).

Xinyu Wang is with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China (e-mail: wangxinyu@whu.edu.cn).

Shaoyu Wang is with the College of Agriculture and Life Sciences, Seoul National University, Seoul 151742, South Korea (e-mail: wsy1995@snu.ac.kr).

Digital Object Identifier 10.1109/TGRS.2024.3392189

I. Introduction

Due to the rich spectral information, hyperspectral imagery (HSI) can be used to detect anomalies with little spatial information [1], [2]. Hyperspectral anomaly detection (HAD) is aimed at detecting pixels that deviate spectrally from the surroundings [3] and has been proven valuable in many areas, including infected tree detection [4], rare mineral detection [5], and defense application. Anomalies always occupy a low proportion of the image and refer to man-made targets, natural objects, and other interferers [4]. To some extent, HAD is equivalent to unsupervised small target detection, where the single target ratio is in range [0.0019%, 0.48%] statistically.

In recent years, deep-learning-based methods have become the mainstream methods [6], [7], [8], [9], where most methods follow the two-step detection paradigm in the first step, and the model learns the background distribution via a proxy task, such as background reconstruction [10], [11], [12], [13]. In the second step, the anomalies are identified by measuring the difference between the input image and the reconstructed background. This two-step detection paradigm leads to two serious problems: 1) a proxy task may obtain suboptimal results because the basic assumption of the proxy task for HAD may not hold [14]. The autoencoder (AE) assumes that the background is more easily reconstructed than the anomalies. Unfortunately, the reconstruction ability of the AE may be migrated to the anomalies, especially when the background distribution is complex [15]. Similarly, generative adversarial networks (GANs) consider that the background can be generated more easily than anomalies [16], but GAN models may also generate samples that are out of the normal background manifold [14], [17]; and 2) the trained model lacks transferability to different images because it aims to learn a certain background distribution in the training stage [18]. The background distribution varies in different images, which hinders the transferability. Thus, the deep-learning-based HAD models are trained and tested on the same image [1], [6]. Although Li et al. [19] attempted to give an HAD model transferability, the HAD model was trained in a supervised manner and was not suitable for a real application.

To this end, the one-step detection paradigm is proposed, where the detection model is optimized *directly* for the HAD

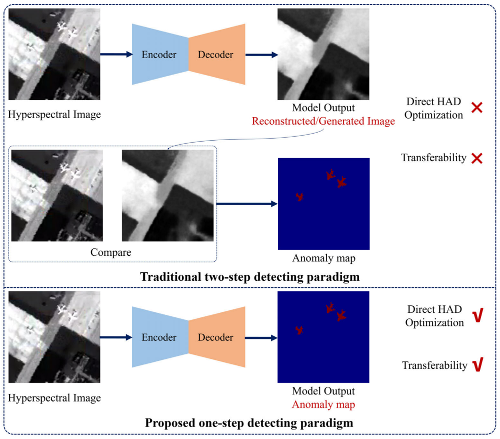

Hyperspectral Image
Encoder
Decoder
Model Output
Reconstructed/Generated Image
Direct HAD Optimization ✗
Transferability ✗
Compare
Anomaly map
Traditional two-step detecting paradigm
Hyperspectral Image
Encoder
Decoder
Model Output
Anomaly map
Direct HAD Optimization ✓
Transferability ✓
Proposed one-step detecting paradigm

Fig. 1. Comparison between the two-step paradigm of the traditional model and the proposed one-step paradigm.

task, and the trained model can be *transferred* to unseen datasets. Fig. 1 shows the comparison of two paradigms. The two-step paradigm focuses on learning specific background distribution while one-step paradigm aims to learn the spectral deviation relationship according to the anomaly definition. The anomaly map is optimized at the training stage and output directly at the test stage without the need of any postprocessing steps. The deviation relationship is modeled as the conditional probability of the original spectra given the background. Since the spectral deviation relationship is universal in varying background, it guarantees the model transferability for unseen scenes. The proposed one-step detection paradigm provides an abstract framework for designing the transferring detection model.

Furthermore, we instantiated the one-step paradigm as the transferred direct detection (TDD) model. To train the TDD model in an unsupervised manner, an anomaly sample simulation strategy is designed. Following the anomaly definition, three points including "surrounding context," "spectral difference," and "infinite" are focused to inspire the simulation process. Given an HSI image, patches are cropped to act as the surrounding context rather than the whole image. Some data argumentation techniques (e.g., channel shuffle) are used to simulate anomaly pixels with spectral difference. The infinite property is guaranteed by making the sample simulating process full of randomness. Numerous simulated samples train the TDD to learn the spectral deviation relationship. The TDD architecture simply follows the U-Net architecture. To make the TDD forget the specific background and learn the deviating relationship between anomalies and background, we design the global self-attention module (GAM) and a local self-attention module (LAM) in the decoder part. The GAM calculates the correlation for all the pixels, while the LAM calculates the correlation for pixels in the local range. GAM and LAM are plug-and-play without changing the feature dimension. The TDD model was validated on six public HAD datasets, where the hyperspectral digital imagery collection experiment (HYDICE) dataset was used for unsupervised training and the remaining datasets were inferred directly, without further fine-tuning. The results show that the proposed TDD model can successfully overcome the limitation of the traditional model training and testing on a single image and has a powerful detection ability and excellent transferability.

The main contributions of this article can be summarized as follows.

1) The one-step detection paradigm for the HAD task is proposed, where the detection model is optimized directly for the HAD task, and the trained model can be transferred to unseen datasets. The two-step paradigm focuses on learning specific background distribution while the one-step paradigm aims to learn the spectral deviation relationship according to the anomaly definition.

2) The one-step paradigm is instantiated as the TDD model. An anomaly sample simulation strategy is designed to train the TDD model in an unsupervised manner. The generated anomaly samples then optimize the TDD model directly for the HAD task.

3) A GAM and an LAM are designed to help the model focus on a "spectrally deviating" relationship, rather than a specific background. The modules are both plug-and-play without changing the feature dimension.

The rest of this article is organized as follows. Section II introduces the current algorithms for the HAD task. Section III describes the one-step detection paradigm and the TDD model. Section IV presents the comparative results and the model analysis. Finally, Section V concludes the article. Code is available at https://github.com/Jingtao-Li-CVer/TDD.

II. Related Work

The current HAD models can be divided into three main categories: 1) statistics-based models; 2) representation-based models; and 3) deep-learning-based models. In addition to reviewing the above algorithms, the definition of the hyperspectral anomalies is also discussed to be clearer.

A. Statistics-Based Models

The statistics-based detection models assume that the background conforms to a certain statistical distribution [20], and the pixels far from the background distribution are regarded as the anomalies [4]. The Reed–Xiaoli anomaly detector (RX-AD) [21] is a milestone statistical method which assumes that the background obeys a Gaussian distribution, and the Mahalanobis distance between the test pixel and the obtained distribution is used to measure the anomaly degree. Inspired by the classical RX detector, a series of extensions based on the generalized likelihood ratio test (GLRT) have [22] been proposed such as the kernel RX-AD [23], weighted-RX-AD and linear-filter-based RX-AD [24], the support vector data description (SVDD) [25], multiple-window AD (MWAD) in [26] using multiple windows to perform anomaly detection adaptively, guided filtering-based AD [27], and spectral–spatial feature extraction-based AD [3]. Most recently, Chang [28] and [29] deviated from the GLRT-based detectors and designed a signal-to-noise ratio (SNR)-based AD, including the GLRT-based detectors as its special cases. Subsequently, Chang [22] and [30] proposed a target-to-anomaly conversion mechanism, which converts many well-known target detectors (e.g., constrained energy minimization (CEM) subpixel detector) to the corresponding anomaly versions (e.g., the CEM anomaly detector). Chang et al. [31] showed that these detectors can be further improved via an iterative process. However, the statistics-based methods make various distribution assumptions, such as the assumption of a Gaussian distribution, which may not hold in complex backgrounds and lead to unsatisfactory detection results [32].

B. Representation-Based Models

The representation-based models detect anomalies using some of the HSI properties, such as the low-rank characteristic of the background or the sparse characteristic of the anomalies [33], [34]. The obtained sparse component then represents the anomalies. Low-rank and sparse-matrix decomposition (LSDM) [35] has been successfully applied for the HAD task, by decomposing the HSI into a low-rank background and sparse anomalies. Candès et al. [36] implemented the LSDM technique with the robust principal component analysis (RPCA) algorithm. Zhang et al. [37] focused more on the low-rank prior and proposed a new Mahalanobis-distance-based detector. To increase the background suppression ability further, Chang has proposed many improved LSDM-based detectors such as OSP-GoDec [38], OSP-AD [39], SDP [40], and effective anomaly space (EAS) [41]. Besides, some researchers have used tensor representation to reflect the 3-D structure of the HSI [42] and collaborative representation to reconstruct the pixel such as the tensor-decomposition-based method (TDAD) [43], tensor principal component analysis (TPCA) [44], and a prior-based tensor approximation (PTA) [45]. However, the above representation-based methods rely on image-specific priors or a constructed dictionary for the certain background, which limits the transferring ability for different scenes.

C. Deep-Learning-Based Models

The deep-learning-based methods always assume that the background can be reconstructed better than the anomalies. They mostly follow the two-step detection paradigm, where the first step involves training a deep reconstruction model and the second step involves outputting the detection map using the reconstructed background [6], [46], [47]. Wang et al. [6] proposed a classical autonomous HAD network (Auto-AD), in which the background is reconstructed by the AE and the anomalies appear as reconstruction errors. An adaptive-weighted loss function was also designed to further suppress the anomaly reconstruction. Xie et al. [11] proposed a spectral constrained adversarial AE (SC_AAE) to perform background suppression and discriminative representation extraction. Wang et al. [1] designed a deep low-rank prior-based method (DeepLR), which combines a model-driven low-rank prior and a data-driven AE. DeepLR can be seen as the extension of Auto-AD. Li et al. [12] developed a sparse coding (SC)-inspired GAN for weakly supervised HAD, which learns a discriminative latent reconstruction with small errors for background pixels and large errors for anomalous ones. Arisoy et al. [13] trained a GAN model to generate a synthetic background image which is close to the original background image. Despite the excellent performance, the two-step detection paradigm can cause suboptimal results and poor transferability (as discussed in Section I). Although Li et al. [19] have introduced a convolutional neural network (CNN) that can also output the anomaly map directly, the model is trained in a supervised manner and limited in certain anomaly categories. The proposed one-step paradigm is aimed to tackle these problems.

D. Definition of Hyperspectral Anomalies

Most related works define the hyperspectral anomalies in a qualitative manner, which stresses that anomaly pixels deviate from the surroundings. There are no specific metrics for the anomaly definition. This is understandable since the anomalies tend to defy any kind of precise specification [48]. In other words, since the anomalies are always varied and unknown, it is easy to find counterexamples for any specific quantitative definition. Despite this, the difference between the hyperspectral anomalies and spatial anomalies can be clearly defined. Due to the high spectral resolution, both spectral and spatial deviations exist for the hyperspectral anomalies while only spatial deviation exists for the spatial anomalies [49].

Readers can understand the hyperspectral anomaly definition by tracing the origin of the task. As claimed by Chang et al. [31], this taxonomy was initially used in military field, where they want to find the small targets without any prior knowledge. From this perspective, the HAD task is equivalent to the unsupervised small target detection.

For the anomaly score, it can be quantitatively defined depending on the proxy tasks. Statistics-based models learn the statistical background distribution [4], [20], and the anomaly score is computed by some distance metric between the pixel and the background distribution. Representation-based models mostly decompose the background into the low-rank and sparse components [33], [34]. The sparsity degree is always treated as the anomaly degree. Deep-learning-based models are mostly trained to reconstruct the background [6], [46], [47], and the reconstruction error is considered as the anomaly score. The above three categories are detailed before. The definition of anomaly score is varying according to the detecting principle. For the proposed one-step detector, the anomaly score is defined as the conditional probability given the image distribution (detailed in Section III-A).

III. Proposed Model

In this section, we first formulate the one-step detection paradigm, and then introduce the instantiated model TDD (as shown in Fig. 2). Converting the focus from certain background into the spectral deviation relationship, the one-step detection paradigm optimizes the model directly for the HAD task and the trained model can be transferred. The TDD model is introduced in four aspects including the designed anomaly sample simulation strategy, model architecture, training loss and model transferability.

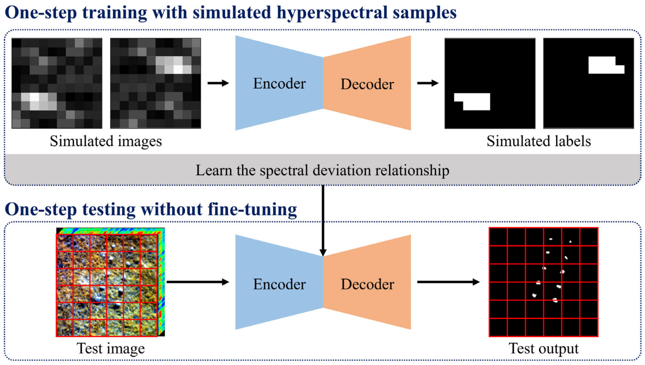

One-step training with simulated hyperspectral samples
Encoder
Decoder
Simulated images
Simulated labels
Learn the spectral deviation relationship
One-step testing without fine-tuning
Encoder
Decoder
Test image
Test output

Fig. 2. Overview of the TDD model, which instantiates the proposed one-step detection paradigm. The TDD model is trained using simulated sample pairs and has the ability to infer the test image without any fine-tuning.

A. Two-Step Versus One-Step Detection Paradigm

Given a hyperspectral image $\mathbf{X} \in R^{H\times W\times R}$, where H, W, and R are the height, width, and channel number, respectively. $\mathbf{X} = \mathbf{B} + \mathbf{A}$ [6], where $\mathbf{B}$ is the background component and $\mathbf{A}$ is the anomaly component. The HAD task is a function f mapping $\mathbf{X}$ to the estimated detection map $\widehat{\mathbf{M}} \in R^{H\times W}$. The value of $\widehat{\mathbf{M}}$ represents the anomaly degree of the corresponding pixel. In this section, we first review the traditional two-step detection paradigm and then introduce the one-step one to show their difference.

1) Two-Step Detection Paradigm: The traditional deep HAD models separate f into two steps: f_1 and f_2. The mappings of f_1 and f_2 are shown in (1) and (2). f_1 adopts a proxy task and always outputs a reconstruction version $\mathbf{X}'$. $\mathbf{X}'$ is expected to only reconstruct the background. f_2 then computes the difference between $\mathbf{X}$ and $\mathbf{X}'$ using a certain metric to obtain $\widehat{\mathbf{M}}$. The model optimization objective is to minimize the difference between $\mathbf{X}$ and $\mathbf{X}'$ of f_1 to learn $P(\mathbf{X})$, as in (3), where L_d is used to compute the matrix difference with some metric

$$f_1 : \mathbf{X} \rightarrow \mathbf{X}'(\text{Learn} P(\mathbf{X})) \tag{1}$$

$$f_2 : \left(\mathbf{X},\mathbf{X}'\right) \rightarrow \mathbf{M}(\text{Compute}\mathbf{A}) \tag{2}$$

$$\text{Loss} = L_d\left(\mathbf{X},\mathbf{X}'\right). \tag{3}$$

The anomaly score is defined as the reconstruction error at test stage. Although the two-step strategy can achieve satisfactory results, the adopted proxy task can lead to suboptimal results and poor transferability, as discussed in Section I.

2) One-Step Detection Paradigm: To overcome these limitations, we propose a one-step detection paradigm, where the model outputs $\widehat{\mathbf{M}}$ from $\mathbf{X}$ directly, as in (4). The optimization objective directly minimizes the difference between the estimated $\widehat{\mathbf{M}}$ and the corresponding ground truth $\mathbf{M}$, as in (5)

$$f : \mathbf{X} \rightarrow \widehat{\mathbf{M}}(\text{Learn} P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})) \tag{4}$$

$$\text{Loss} = L_d\left(\mathbf{M},\widehat{\mathbf{M}}\right). \tag{5}$$

The anomaly score for each pixel $\mathbf{x_i}$ in $\widehat{\mathbf{M}}$ is described as the conditional probability $P(\mathbf{x_i} \mid \mathbf{X})$. A lower occurrence probability $P(\mathbf{x_i} \mid \mathbf{X})$ implies a higher anomaly degree. Different from the two-step methods learning certain $P(\mathbf{X})$, the one-step paradigm learns the deviation relationship $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$. The model is trained to output $P(\mathbf{x_i} \mid \mathbf{X})$ constrained by the inequality. Since anomalies are considered appeared with a low probability and the deviation inequality between $\mathbf{A}$ and $\mathbf{B}$ holds true in all the hyperspectral scenes, the trained detector can be used directly for the unseen scenes.

```
Algorithm 1 One-Step Detecting Paradigm
Input: Generated anomaly samples with some strategy
Training stage (1 epoch):
    Select one paired sample $(\mathbf{X}, \mathbf{M})$, where (10) holds.
    Predict estimated anomaly map $\widehat{\mathbf{M}}$ for $\mathbf{X}$
    Compute the loss $L_d(\mathbf{M},\widehat{\mathbf{M}})$
    Network backward
Testing stage for any unseen image:
    Cropped the test image into patches
    Predict the anomaly map for each patch
    Combine into the final anomaly map
Output:A trained model with the transferring ability
```

To train the one-step detector to output $P(\mathbf{x_i} \mid \mathbf{X})$ constrained by $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$, lots of paired samples $(\mathbf{X}, \mathbf{M})$ are needed due to the data-driven property of the deep learning (as in **Algorithm 1**). For each image pixel $\mathbf{X}_i$, its ideal corresponding ground truth $\mathbf{M}_i$ is the anomaly score as in (6). The ranking relationship of anomaly score is given in (7). The required training samples can be acquired with many methods in practice and we only give a high-level description in this section. The optimization loss for the deviation relationship $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$ can be ranking loss or proxy classification loss [50], [51]

$$\mathbf{M}_i = P(\mathbf{x}_i \mid \mathbf{X}) \tag{6}$$

$$\mathbf{M}_i(\mathbf{x}_i \in \mathbf{B}) > \mathbf{M}_i(\mathbf{x}_i \in \mathbf{A}). \tag{7}$$

At the test stage, the patches are cropped and inferred in turn as in **Algorithm 1**. The inferring patch size is an adjusted hyperparameter. The inferring patch size decides what extent of the surrounding environment is used to judge the anomaly. In the Section IV-C, we showed that TDD was robust to the inferring patch size. This is understandable since the hyperspectral anomalies always occupy a low ratio of the image and even 1/10 of the image surrounding is enough for the TDD to judge the deviation relationship.

Compared with the two-step detection paradigm, the one-step detection paradigm has two main advantages, as shown in Fig. 1: 1) the model is optimized directly for the HAD task, without any postprocessing process (i.e., f_2); and 2) transferability can be achieved since the model learns unified relationship $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$ rather than a specific distribution of $P(\mathbf{B})$ or $P(\mathbf{A})$.

B. Anomaly Sample Simulation

Anomaly samples are necessary to optimize the model as in (5). Considering that real anomaly samples are difficult to obtain in real applications, anomaly simulation is needed. The

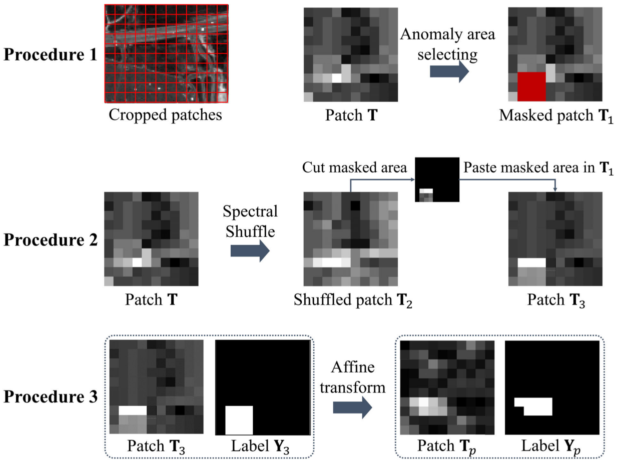


Fig. 3. Proposed anomaly sample simulation strategy.

proposed simulation strategy is designed to create a lot of paired samples meeting the deviation inequality (7).

To simulate the required samples $(\mathbf{X}, \mathbf{M})$ for the model to learn $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$, three key points need to be concerned in particular. First, most works treat $\mathbf{B}$ as the local context [14], [21], and thus, we can use a patch as the input unit rather than the whole image. This design not only avoids interference from distant pixels but also increases the sample number. Second, a large spectral difference between $\mathbf{A}$ and $\mathbf{B}$ must exist to make the inequality (10) hold. Increasing the spectral difference between $\mathbf{x}_i$ and $\mathbf{X}$ means the decrease in $P(\mathbf{x}_i \mid \mathbf{X})$. Third, the simulated anomalies must be large in number. Due to the powerful feature representation ability of deep models, a small number of $\mathbf{A}$ and $\mathbf{B}$ may train the model to learn specific anomaly categories $P(\mathbf{A})$ or $P(\mathbf{B})$ rather than the desired relationship learning.

Based on the above three points, we designed an anomaly sample simulation strategy consisting of three procedures, an example of which is provided in Fig. 3. Note that both $\mathbf{T}$ and $\mathbf{X}$ are hyperspectral images and we use $\mathbf{T}$ to denote the cropped patch of $\mathbf{X}$. Inequality (10) holds true for both $\mathbf{T}$ and $\mathbf{X}$.

Procedure 1: Anomalous region selection. After cropping the entire training image into patches, this operation is performed for each patch in turn. Although hyperspectral anomalies occupy the low ratio in the whole image, they would have relatively larger ratio in the cut patches. This procedure requires the control of the anomaly area size and training patch size. Anomaly area size determines how large an anomaly will be detected at test stage. The training patch size defines how large the concept of "surrounding" is. For the size hyperparameter of training patches, the proposed model is robust to it in a wide range except for the very small patch size (e.g., patch size 5 in the HYDICE dataset) (detailed in the analysis in Section IV-C). For input patch $\mathbf{T}$, the processed patch is denoted as $\mathbf{T}_1$. The selected anomaly area is rectangular and masked in $\mathbf{T}_1$.

Procedure 2: Anomaly spectrum generation. To create deviated spectra meeting the inequality (10), each pixel of patch $\mathbf{T}$ is first randomly shuffled in the spectral dimension to create $\mathbf{T}_2$. The shuffle operation remains consistent for all the pixels within each patch. Since the shuffled $\mathbf{T}_2$ has different spectral distribution with $\mathbf{T}_1$, the deviation inequality $P(\mathbf{x}_i \in \mathbf{T}_1 \mid \mathbf{T}_1) > P(\mathbf{x}_i \in \mathbf{T}_2 \mid \mathbf{T}_1)$ can be met. The pixels in $\mathbf{T}_2$ can be considered as the anomalies given $\mathbf{T}_1$ as the background. Next, cut anomaly pixels at the masked area in $\mathbf{T}_2$ and paste them into $\mathbf{T}_1$ to obtain $\mathbf{T}_3$. This procedure aims to generate deviated spectra from $\mathbf{T}$, some other strategies in addition to the spectral shuffle are also worked such as adding the white noise (detailed in the analysis in Section IV-C).

Procedure 3: Anomalous region affine transformation. Although $\mathbf{T}_3$ is already a trainable anomaly sample, the anomalous regions in $\mathbf{T}_3$ are uniformly rectangular, which does not correspond to the real-world situation where anomalies have different shapes. For this reason, random affine transformation is performed on $\mathbf{T}_3$ with its corresponding label $\mathbf{Y}_3$ to give the anomaly shape information. The applied affine transformation is a combined sequence of rotation, scaling, and translation. For patch $\mathbf{T}_3$ and $\mathbf{Y}_3$, the affine transformation process can be formulated as follows:

$$\begin{aligned} \mathbf{T}_p &= \mathbf{R}\mathbf{T}_3 + b \\ \mathbf{Y}_p &= \mathbf{R}\mathbf{Y}_3 + b \end{aligned} \tag{8}$$

where $\mathbf{R}$ is the transformation matrix and b is the translation bias. The resulting $(\mathbf{T}_p, \mathbf{Y}_p)$ is the finally generated anomaly sample.

Ideally, the generated label needs to have continuous score representing the probability density as in (6). The scores can be computed by modeling $P(\mathbf{T}_p)$ explicitly such as the multivariate Gaussian distribution. Considering the certain distribution may not fit the real scene well [32], we deal with it in a simpler manner, where the labels $\mathbf{Y}_3$ and $\mathbf{Y}_p$ are generated as the binary map and convert the detection problem as a binary classification problem. For the deep classification model [52], the output value of the last Sigmoid layer is positively correlated with the probability value [53]. The anomaly label is assigned to one for highlighting the anomaly pixels at test stage.

The designed anomaly simulation strategy incorporates strong anomaly location randomness, spectral randomness, and shape randomness, based on the deviation ranking $P(\mathbf{B} \mid \mathbf{X}) > P(\mathbf{A} \mid \mathbf{X})$ in the one-step paradigm. These properties force the model to learn a "spectrally deviating" relationship rather than a specific background or anomaly target.

C. Model Architecture

To output the anomaly score map directly, thus not relying on any postprocessing steps and enabling direct optimization of the HAD task, the proposed architecture is based on the U-Net architecture with encoding and decoding parts (as shown in Fig. 4) [54]. The encoder part contains six cascaded feature extraction blocks for extracting multiscale, multilevel features. The decoder part contains the decoding blocks corresponding to the encoding blocks in turn. The skip connection between the encoder and decoder parts helps maintain the important spatial information.

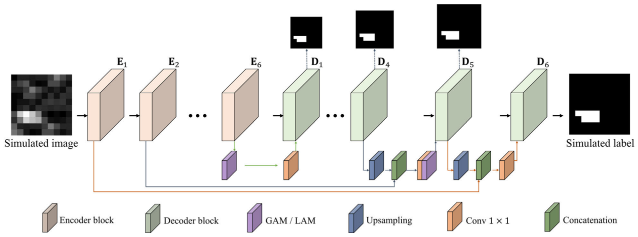


Fig. 4. TDD model architecture.

The main architectural innovation is the design of the GAM and the LAM in the decoding part. Because we expect the model to learn the deviating relationship between the anomalies and background, rather than being dependent on a specific background, the relationship modeling between pixels in the input image is particularly important. Self-attention mechanisms help with this as they model the correlation between each pixel and the rest of the image pixels [55], avoiding the problem of convolution failing to capture the long-distance dependencies. The GAM calculates the correlation for all the pixels, while the LAM calculates the correlation for pixels in the local range. To reduce the computational burden and suppress local noise, the decoder uses alternating global and local perceptual attention modules. The designed self-attention module can be plug-and-play, without changing the spatial size and feature dimension of the original feature map.

1) Encoder: Six cascaded feature extraction blocks form the encoder. Each extraction block consists mainly of several convolutional layers and rectified linear unit (ReLU) activation layers. In view of the small size of the hyperspectral anomaly objects, the convolution kernel is set to 3 × 3 to prevent the loss of anomaly targets during the network forward propagation. Dilated convolution is used in the last block to keep the large size of the feature maps, to avoid losing too many spatial details [56]. As the network layer deepens, the spatial dimension of the extracted feature cube continues to shrink and the feature dimension gradually increases. Changes in the spatial dimensions are implemented using dilation pooling, and the feature dimension changes are implemented using convolution. Finally, the encoder outputs six feature cubes, named $\mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_6$, in sequential order.

2) Decoder: The role of the decoder is to decode the features obtained from the encoder into the final anomaly score map. To make full use of the multilevel feature output from the encoder and maintain important spatial details, the decoder is designed as a symmetrical structure having six decoding blocks. The output feature cubes are named $\mathbf{D}_1, \mathbf{D}_2, \ldots, \mathbf{D}_6$ in sequential order. Each decoding block can be thought of as a function f in the following equation:

$$\mathbf{D}_i = g(\mathbf{D}_{i-1}, \mathbf{E}_{7-i}). \tag{9}$$

Each decoding block $\mathbf{D}_i$ has two inputs: $\mathbf{D}_{i-1}$ from the previous block and the feature cube $\mathbf{E}_{7-i}$ corresponding to the output of the encoder. $\mathbf{D}_1$ can be seen as a special case where $\mathbf{D}_0$ is none. For the subsequent fusion between $\mathbf{D}_{i-1}$ and $\mathbf{E}_{7-i}$, each $\mathbf{D}_{i-1}$ is first interpolated to reach the same spatial dimension as $\mathbf{E}_{7-i}$. The fusion step of g can be divided into three substeps, as shown in the following equation:

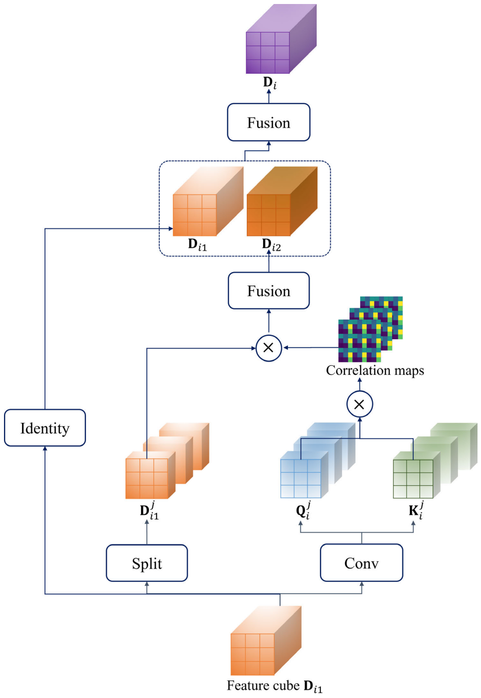


Fig. 5. Global self-attention module.

$$\mathbf{D}_{i1} = \text{Conv}_{1\times1}(\text{Concat}(\mathbf{D}_{i-1}, \mathbf{E}_{7-i})) \tag{10}$$

$$\mathbf{D}_i = \text{GAM/LAM}(\mathbf{D}_{i1}). \tag{11}$$

Equation (10) fuses $\mathbf{D}_{i-1}$ and $\mathbf{E}_{7-i}$ in the channel dimension using 1 × 1 convolution [57], which is a special type of the convolution operation with the kernel size 1 × 1. It can communicate the channel information and reduce the channel dimension of the feature cube. We let the channel size of $\mathbf{D}_{i-1}$ and $\mathbf{E}_{7-i}$ be C. The convolution in (10) reduces the concatenated feature block channel dimension from $2C$ to C. Equation (11) then processes $\mathbf{D}_{i1}$ using the designed GAM or LAM, which is elaborated below. The GAM and LAM do not change the feature block size and can achieve a plug-and-play effect. The last decoding block $\mathbf{D}_6$ outputs the anomaly score map directly, without the GAM or LAM (i.e., $\mathbf{D}_i = \mathbf{D}_{i1}$).

Global Self-Attention Module. To make the model focus on the "spectrally deviating" relationship between the anomalies and background, we designed the GAM to explicitly model the pixel correlation in the whole patch. The internal architecture of the GAM is shown in Fig. 5. Inspired by Dosovitskiy et al. [58] and Pu et al. [59], the query $\mathbf{Q}_i$ and key $\mathbf{K}_i$ are first generated for $\mathbf{D}_{i1}$ using 1 × 1 convolution. Due to the large spectral dimension of the feature cube in the decoder, the generated query and key values are split and thus used in the computation of the multihead self-attention mechanism. Correspondingly, the same split operation is performed for $\mathbf{D}_{i1}$. Assuming that $\mathbf{Q}_i$, $\mathbf{K}_i$, and $\mathbf{D}_{i1}$ are all divided into n segments in the spectral dimension, each segment is named in turn as $\mathbf{Q}_i^j$, $\mathbf{K}_i^j$, and $\mathbf{D}_{i1}^j$ $(1 \leq j \leq n)$. For each combination $(\mathbf{Q}_i^j, \mathbf{K}_i^j, \mathbf{D}_{i1}^j)$, the self-attention mechanism relies on scaled dot-product

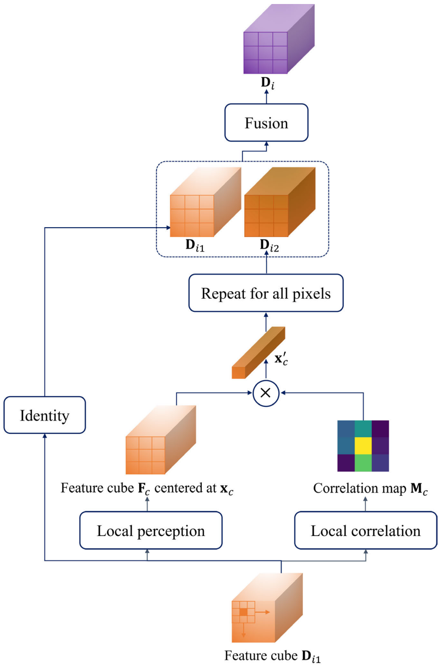


Fig. 6. Local self-attention module.

attention, as given in the following equation:

$$\text{Attention}\left(\mathbf{Q}_i^j, \mathbf{K}_i^j, \mathbf{D}_{i1}^j\right) = \text{Softmax}\left(\frac{\mathbf{Q}_i^j\left(\mathbf{K}_i^j\right)^T}{\sqrt{d}^k}\right)\mathbf{D}_{i1}^j \quad (12)$$

where d^k is the key dimension and $\text{Softmax}((\mathbf{Q}_i^j(\mathbf{K}_i^j)^T/d)^k$ is the obtained correlation map. To make the processed feature cube have the same dimensionality as $\mathbf{D}_{i1}$, the n obtained heads ($\text{Head}_j = \text{Attention}(\mathbf{Q}_i^j, \mathbf{K}_i^j, \mathbf{D}_{i1}^j)$, $1 \leq j \leq n$) are fused as shown in (13), where $\mathbf{W}$ is the linear mapping parameter

$$\mathbf{D}_{i2} = \text{Concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_n)\mathbf{W}. \quad (13)$$

$\mathbf{D}_{i2}$ contains the fused global pixel correlation information. Considering that $\mathbf{D}_{i2}$ may lose the information of the pixel itself due to the introduction of other pixel features, $\mathbf{D}_{i2}$ and $\mathbf{D}_{i1}$ are further fused using 1×1 convolution to obtain $\mathbf{D}_i$ in (14). Equation (14) uses the convolution to reduce the concatenated feature dimension to be the same as that of $\mathbf{D}_{i1}$

$$\mathbf{D}_i = \text{Conv}_{1\times 1}(\text{Concat}(\mathbf{D}_{i1}, \mathbf{D}_{i2})). \quad (14)$$

*Local Self-Attention Module.*Unlike the GAM, the LAM only computes self-attention in local windows, which is more computationally efficient. The LAM can be used as the refinement of the GAM to better maintain local consistency and eliminate the influence of noise on HAD. The internal architecture of the LAM is shown in Fig. 6. We let the size of the local window be $\widehat{H} \times \widehat{W}$ centered at feature $\mathbf{x}_c$. For the centered $\mathbf{x}_c$, the local perception operation is used to extract the corresponding feature cube $\mathbf{F}_c$ with the size $\widehat{H} \times \widehat{W} \times C$. The local correlation operation computes the correlation value between $\mathbf{x}_c$ and each remaining feature vector in $\mathbf{F}_c$. The obtained correlation map $\mathbf{M}_c$ has a size of $\widehat{H} \times \widehat{W}$. For computational efficiency, $\mathbf{M}_c$ is calculated using convolution on $\mathbf{F}_c$ and the Softmax activation function. The processed contextual feature $\mathbf{x}'_c$ is then the weighted average of $\mathbf{F}_c$ according to $\mathbf{M}_c$, as shown in the following equation:

$$\mathbf{x}'_c = \sum_{h=1}^{\widehat{H}} \sum_{w=1}^{\widehat{W}} \mathbf{M}_c^{hw} \mathbf{F}_c^{hw} \quad (15)$$

where $\mathbf{F}_c^{hw}$ is the feature vector in $\mathbf{F}_c$ at spatial location (h, w). The above process can be repeated efficiently for all the features in $\mathbf{D}_{i1}$ by matrix operations, and then the contextual cube $\mathbf{D}_{i2}$ is obtained. Similar to the GAM, the LAM has the same fusion process as shown in (14), after which $\mathbf{D}_i$ is finally obtained.

D. Training Loss

The anomaly sample simulation process generates many pairs of hyperspectral data and anomaly labels, which provide a strong supervised signal for model training. To make full use of the simulated labels, we add a 1×1 convolutional layer and sigmoid activation layer on top of each $\mathbf{D}_i$ to generate anomaly maps $\widehat{\mathbf{M}_i}$, as shown in the following equation:

$$\widehat{\mathbf{M}_i} = \text{Sigmoid}(\text{Conv}_{1\times 1}(\mathbf{D}_i)) \quad (16)$$

The six generated $\widehat{\mathbf{M}_i}$ have different spatial sizes, keeping the same size as their corresponding $\mathbf{D}_i$. Finally, the ground truth $\mathbf{M}$ is resized to the same size for each $\mathbf{D}_i$, denoted as $\mathbf{M}_i$ with spatial size $H_i \times W_i$, to optimize the model based on the weighted average cross-entropy (CE) loss L_{wce} [60] between the probability distribution $P(\widehat{\mathbf{M}_i})$ and $P(\mathbf{M}_i)$

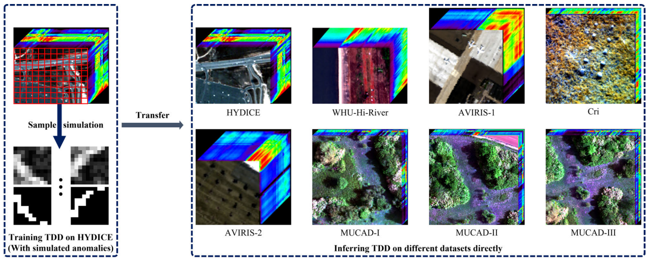
$$L_{wce} = -\sum_{i=1}^{6} \omega_i \sum_{h=1}^{H_i} \sum_{w=1}^{W_i} \text{CE}\left(P\left(\widehat{M}_i\right), P(\mathbf{M}_i)\right) \quad (17)$$

where ω_i is the weight for the CE loss. The CE optimizes the model to output the anomaly map distribution $\widehat{\mathbf{M}_i}$ as close to the ground-truth distribution $\mathbf{M}_i$ generating from the simulating process (e.g., $\mathbf{Y}_p$ in Fig. 3) as possible.

E. Model Transferability

Unlike the previous deep models that can only train and reason on a single image, the proposed TDD model has the ability to migrate between different images. In other words, we only need to train once on a single image to infer on many other unseen images. This is because the TDD model is required to learn the "spectrally deviating" properties of anomalies, rather than the specific background, as described in detail in Sections III-A–III-C.

Despite this, there is still a remaining problem to be solved when the TDD model infers between different data. The input layer of the built network architecture requires a fixed number of bands, but different numbers of bands for different data. To solve this problem, we need to process the data in terms of the channel dimensions, before inference. We let the number of channels of the training data be B_1 and the number of channels of the test data be B_2. If $B_2 < B_1$, bilinear interpolation of the test data is performed in the spectral dimension to achieve



Note: Traditional models are trained and inferred on each dataset

Fig. 7. Experimental design for validating the transferability of TDD. In contrast, the comparative models are trained on each dataset separately.

a spectral dimension of B_1. If $B_2 > B_1$, the test data are cut into many segments along the spectral dimension, where the spectral dimension of each segment is B_1. When the spectral dimension of the last segment is less than B_1, the B_1 bands of the last segment are taken from the end of the spectrum. Finally, the mean of all the segment detection results is used as the final output.

It is worth noting that the whole process does not abandon any of the original bands. The above trick that is used to deal with the varying image channels is simple but effective in practice.

IV. Experiments and Analysis

A. Experimental Settings

1) Datasets: In the experiments, the proposed TDD model was validated on six public datasets as shown in Table I. Except for the five hyperspectral datasets, the multispectral dataset MUCAD [61] is also used to validate the model extensibility. The used MUCAD dataset consists of three scenes with different camouflages (i.e., MUCAD-I, MUCAD-II, and MUCAD-III).

To validate the transferability of TDD, we trained the TDD model using the simulated anomaly samples from the HYDICE dataset and then inferred the model on all the datasets directly (as shown in Fig. 7). We chose the HYDICE dataset as the training set because its background contains more homogeneous categories, and tiny anomalies bring less contamination to the simulated labels. Actually, any hyperspectral dataset can act as the training image to obtain a valid model (discussed later).

2) Comparison Models and Evaluation Metrics: The TDD model was compared with the following six models: the global RX detector (GRX) [21], the low-rank and sparse matrix decomposition-based Mahalanobis distance (LRASR) method [62], the abundance and dictionary-based low-rank decomposition (ADLR) [63] method, the collaborative-representation-based (CRD) method [64], the Auto-AD based on a fully convolutional AE (Auto-AD) [6], and the deep low-rank prior-based method (DeepLR) [1]. The comparison methods cover the three categories of RXD-based, representation-based, and deep0learning-based methods, and all of them are classic algorithms. Among the different methods, Auto-AD and DeepLR are the very recently proposed state-of-the-art models.

TABLE I
Introduction of Six Public Datasets

Dataset	Sensor	Spectral range (nm)	Size	Spatial resolution (m/pixel)	Anomalies
HYDICE	HYDICE	400–2500	80×100×162	1	vehicles
WHU-Hi-River	Headwall Nano	400-1000	105×168×135	0.06	plastic plates, gray panels
AVIRIS-1	AVIRIS	400–2500	100×100×186	3.5	planes
Cri	Nuance Cri	650-1100	400×400×46	\	rocks
AVIRIS-2	AVIRIS	370-2510	60×60×189	3.5	planes
MUCAD	XT2, Altum	438-10500	512×512×6	0.1	camouflages

Eight metrics about the 3-D ROC curve are used to evaluate the quantitative performance following Chang [65] and [66]. Compared with the single metric $\mathrm{AUC}_{(D,F)}$, the other metrics also consider the threshold dimension and can provide a comprehensive evaluation. The larger value means the better performance for all the metrics.

3) Implementation Details: We set the cluster number and the selected pixels of LRASR to 15 and 20, respectively, according to the original paper. The regularization parameter of CRD was set to 10^{-0}. The CRD model adopted a dual-window strategy with the inner window size w_{in} and the outer window size w_{out}. $(w_{\mathrm{in}}, w_{\mathrm{out}})$ were set to (7, 15) for the HYDICE dataset and the WHU-Hi-River dataset, and (11, 17) for the remaining datasets. The threshold of DeepLR was set to 0.00001 for the HYDICE dataset and 0.0001 for the other datasets. The GAM and LAM were used alternately in the decoder part of the TDD model. The training patches are cropped under the overlap setting, where the overlap size is set as the half of the patch size. For the first five decoding blocks, the order of use was LAM-GAM-LAM-GAM-LAM. ω_1, ω_2, and ω_3 were set to 0.5. ω_4, ω_5, and ω_6 were set to 1.0. The CPU was an Intel[1] Xeon[1] CPU E5-2690 v4 at 2.60 GHz with 62-GB memory, and the GPU was a Tesla P100-PCIE with 16 GB of memory.

B. Model Comparison

The model comparison results are given in Tables II–V, Figs. 8 and 9. The three scenes in MUCAD are separately evaluated. TDD has achieved the highest $\mathrm{AUC}_{(D,F)}$ values under six datasets. For the AVIRIS-2 and MUCAD-III datasets, TDD surpassed the suboptimal models by 3 and 5 points, respectively. Considering all the datasets and eight metrics together, TDD has achieved 25 optimal metrics surpassing the suboptimal model DeepLR by 15 metrics. Thus, TDD has the best overall quantitative performance.

There are great differences among the comparative anomaly maps in Fig. 8. Ideally, a good anomaly map needs to be both anomaly-discriminative and background-suppressive. However, the two conditions are difficult to meet simultaneously in practice. The map with higher anomaly pixel responses always has higher false alarms (e.g., ADLR and LRASR). Differently, Auto-AD and DeepLR have both low

[1] Registered trademark.

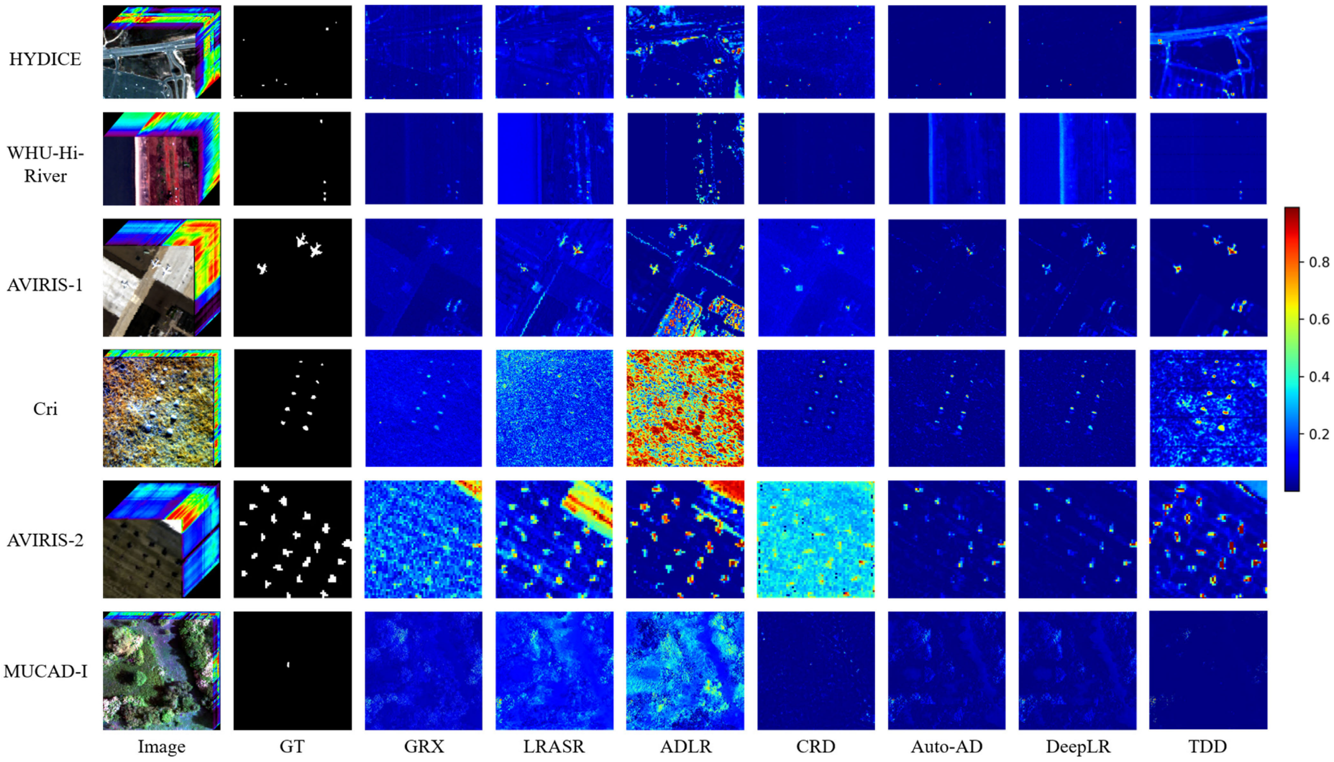
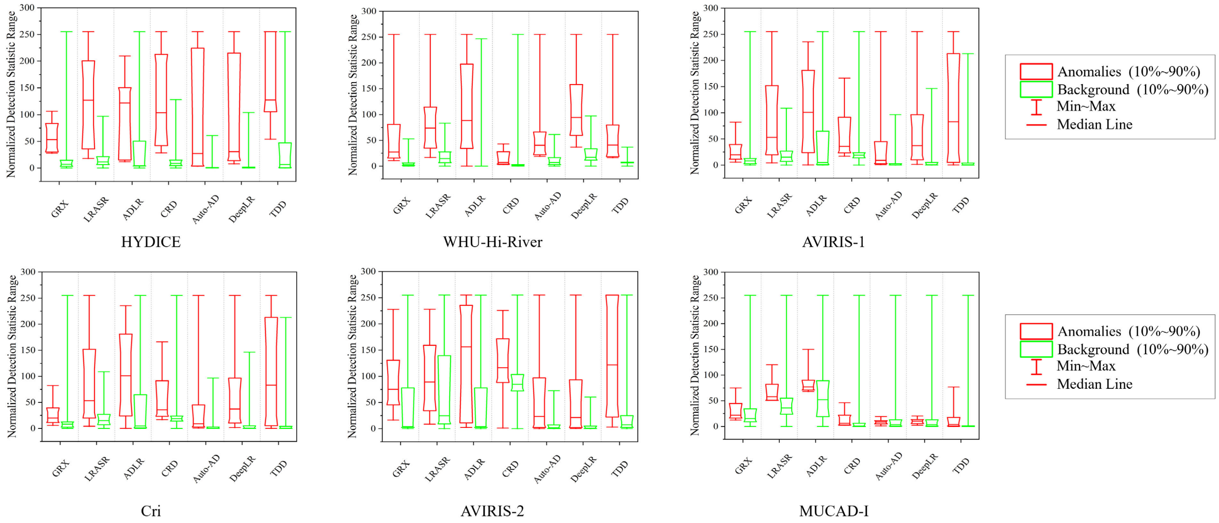


Fig. 8. Qualitative comparison results on six public datasets.



HYDICE WHU-Hi-River AVIRIS-1

Cri AVIRIS-2 MUCAD-I

Fig. 9. Separability maps on six public datasets.

responses for anomalies and background. The anomaly maps of TDD have made a successful balance, where the anomalies have discriminative values and the brining false alarms are few.

The above results are consistent with the separating maps in Fig. 9, where the larger margin between anomalies box and background box means higher anomaly-background separating degree. TDD is the only model where the anomalies box and background box have obvious margin in all the six datasets especially for the HYDICE and MUCAD-I datasets. Overall, the instantiated one-step model (i.e., TDD) has surpassed the current two-step deep detection methods in both quantitative and qualitative results.

C. Model Analysis

In this section, we describe the ablation experiments about the designed attention modules (LAM and GAM), different training datasets, the anomaly generating strategy, the training patch size, the inferring hyperparameters and execution time.

1) Ablation Analysis of LAM and GAM: Being plug-and-play, LAM and GAM can be used in U-like architectures.

TABLE II

Quantitative Comparison Results on the HYDICE and WHU-Hi-River Datasets

Model	HYDICE								WHU-Hi-River							
	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$
GRX	0.9938	0.2487	0.9429	1.2425	1.9367	1.1916	4.3555	2.1854	0.9988	0.1999	0.9789	1.1987	1.9777	1.1788	9.4739	2.1776
LRASR	0.9920	0.5189	0.9510	1.5109	1.9430	1.4699	10.5898	2.4619	0.9815	0.4090	0.9318	1.3905	1.9133	1.3408	5.9971	2.3223
ADLR	0.9624	0.4640	0.9287	1.4264	1.8911	1.3927	6.5077	2.3551	0.9560	0.3955	0.9860	1.3515	1.9420	1.3815	**28.2500**	2.3375
CRD	0.9991	0.5145	0.9424	1.5136	1.9415	1.4569	8.9323	2.4560	0.9885	0.0449	**0.9917**	1.0334	**1.9802**	1.0366	5.4096	2.0251
Auto-AD	0.9991	0.2756	**0.9930**	1.2747	**1.9921**	1.2686	**39.3714**	2.2677	0.9725	0.2467	0.9627	1.2192	1.9352	1.2094	6.6139	2.1819
DeepLR	**0.9996**	0.3054	0.9844	1.3050	1.9840	1.2898	19.5769	2.2894	0.9925	**0.4692**	0.9224	**1.4617**	1.9149	**1.3916**	6.0464	**2.3841**
TDD	0.9960	**0.8757**	0.9340	**1.8717**	1.9300	**1.8097**	13.2682	**2.8057**	**0.9999**	0.3985	0.9721	1.3984	1.9720	1.3706	14.2832	2.3705

TABLE III

Quantitative Comparison Results on the AVIRIS-1 and Cri Datasets

Model	AVIRIS-1								Cri							
	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$
GRX	0.9370	0.0968	0.9691	1.0338	1.9061	1.0659	3.1327	2.0029	0.9678	0.2254	0.9104	1.1932	1.8782	1.1358	2.5156	2.1036
LRASR	0.9146	0.2956	0.9335	1.2102	1.8481	1.2291	4.4451	2.1437	0.8652	0.2327	0.8092	1.0979	1.6744	1.0419	1.2196	1.9071
ADLR	0.9081	**0.3997**	0.9148	1.3078	1.8229	1.3145	4.6913	2.2226	0.9579	**0.9674**	0.3580	**1.9253**	1.3159	1.3254	1.5069	2.2833
CRD	0.9530	0.1857	0.9314	1.1387	1.8844	1.1171	2.7070	2.0701	0.9186	0.2164	0.9552	1.1350	1.8738	1.1716	4.8304	2.0902
Auto-AD	0.9628	0.0884	**0.9947**	1.0512	1.9575	1.0831	16.6792	2.0459	0.9643	0.2496	0.9850	1.2139	1.9493	1.2346	16.6400	2.1989
DeepLR	**0.9845**	0.2013	0.9902	1.1858	**1.9747**	1.1915	20.5408	2.1760	0.9815	0.2650	**0.9872**	1.2465	**1.9687**	1.2522	**20.7031**	2.2337
TDD	0.9728	0.3741	0.9907	**1.3469**	1.9635	**1.3648**	**40.2258**	**2.3376**	**0.9915**	0.6383	0.8878	1.6298	1.8793	**1.5261**	5.6889	**2.5176**

TABLE IV

Quantitative Comparison Results on the AVIRIS-2 and MUCAD-I Datasets

Model	AVIRIS-2								MUCAD-I							
	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$
GRX	0.7521	0.3230	0.7788	1.0751	1.5309	1.1018	1.4602	1.8539	0.7445	0.1075	0.9218	0.8520	1.6663	1.0293	1.3747	1.7738
LRASR	0.8097	0.3763	0.8189	1.1860	1.6286	1.1952	2.0779	2.0049	0.9129	0.2508	0.8480	**1.1637**	1.7609	1.0988	1.6500	**2.0117**
ADLR	0.8918	**0.5643**	0.9035	**1.4561**	1.7953	**1.4678**	5.8477	**2.3596**	0.8197	**0.3282**	0.7872	1.1479	1.6069	**1.1154**	1.5423	1.9351
CRD	0.8729	0.4831	0.6687	1.3560	1.5416	1.1518	1.4582	2.0247	0.8801	0.0487	0.9876	0.9288	1.8677	1.0363	3.9274	1.9164
Auto-AD	0.8939	0.0613	0.9875	0.9552	1.8814	1.0488	4.9040	1.9427	0.7898	0.0398	0.9729	0.8296	1.7627	1.0127	1.4686	1.8025
DeepLR	0.9163	0.1807	0.9914	1.0970	1.9077	1.1721	21.0116	2.0884	0.7942	0.0396	0.9738	0.8338	1.7680	1.0134	1.5115	1.8076
TDD	**0.9466**	0.1500	**0.9988**	1.0966	**1.9454**	1.1488	**125.0000**	2.0954	**0.9365**	0.0301	**0.9961**	0.9666	**1.9326**	1.0262	**7.7179**	1.9627

TABLE V

Quantitative Comparison Results on the MUCAD-II and MUCAD-III Datasets

Model	MUCAD-II								MUCAD-III							
	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$	AUC$_{(D,F)}$	AUC$_{ADP}$	AUC$_{BDP}$	AUC$_{JAD}$	AUC$_{JBS}$	AUC$_{ADBS}$	AUC$_{SNPR}$	AUC$_{OADP}$
GRX	0.4757	0.1192	0.8572	0.5949	1.3329	0.9764	0.8347	1.4521	0.5896	0.1169	0.8846	0.7065	1.4742	1.0015	1.0130	1.5911
LRASR	0.9191	**0.4713**	0.7082	**1.3904**	1.6273	**1.1795**	1.6151	**2.0986**	0.8065	**0.3496**	0.7729	**1.1561**	1.5794	**1.1225**	1.5394	**1.9290**
ADLR	0.6559	0.1993	0.7859	0.8552	1.4418	0.9852	0.9309	1.6411	0.6977	0.2180	0.8703	0.9157	1.5680	1.0883	1.6808	1.7860
CRD	0.9192	0.0437	**0.9878**	0.9629	**1.9070**	1.0315	**3.5820**	1.9507	0.7990	0.0283	0.9896	0.8273	1.7886	1.0179	2.7212	1.8169
Auto-AD	0.7135	0.0707	0.9424	0.7842	1.6559	1.0131	1.2274	1.7266	0.7744	0.0587	0.9532	0.8331	1.7276	1.0119	1.2543	1.7863
DeepLR	0.7167	0.0736	0.9414	0.7903	1.6581	1.0150	1.2560	1.7317	0.8013	0.1093	0.9380	0.9106	1.7393	1.0473	1.7629	1.8486
TDD	**0.9243**	0.0522	0.9752	0.9765	1.8995	1.0274	2.1048	1.9517	**0.8557**	0.0375	**0.9912**	0.8932	**1.8469**	1.0287	**4.2614**	1.8844

Table VI showed the related ablation results about whether to use and how to use the modules. The model setting represents the case of how the GAM (G for short) or LAM (L for short) is used in the decoder. Compared with the case of no attention module, both the GAM and LAM can significantly increase the transferability of the TDD model in most cases (e.g., the Cri dataset). It is worth noting that the "LLLLL" setting may cause some failed detections (AUC$_{(D,F)}$ < 0.5) while "GGGGG" does not. This may be caused by the patch input form, where the model input is already a local area, and thus the global attention is more necessary. Compared with using only one kind of attention mechanism (i.e., the case of only the GAM or LAM), the alternating use of the LAM and GAM can further improve the detection ability lightly.

2) Sensitivity Analysis of Different Training Datasets: The TDD model in model comparison section is trained on the HYDICE dataset. Actually, the transferability can be observed under different training datasets. To clarify this, we use different training datasets and observe the model performance. Table VII lists the related results. Since the MUCAD dataset

TABLE VI

ABLATION EXPERIMENTS ABOUT THE ATTENTION MODULES IN THE DECODER ($\mathrm{AUC}_{(D,F)}/\mathrm{AUC}_{\mathrm{OADP}}$)

Model Setting	HYDICE	WHU-Hi-River	AVIRIS-1	Cri	AVIRIS-2	MUCAD-I	MUCAD-II	MUCAD-III
No (L/G)	0.9911/**2.9202**	0.9846/2.0924	0.9024/2.0458	0.8706/2.2796	0.8672/1.8551	0.9164/1.9326	0.9271/1.9332	**0.9311/1.9472**
GGGGG	0.9925/2.7107	0.9664/2.3661	**0.9740/2.5300**	0.9494/2.3510	0.9074/1.8972	**0.9449/2.0429**	0.9152/**2.0622**	0.8288/1.9254
LLLLL	0.9936/2.7303	0.9831/2.3135	0.8916/2.1991	**0.9981/2.5967**	0.2158/1.1970	0.5103/1.5031	0.3378/1.3198	0.2426/1.2326
LGLGL	**0.9960**/2.8057	**0.9999/2.3705**	0.9728/2.3376	0.9915/2.5176	**0.9466/2.0953**	0.9365/1.9626	**0.9243**/1.9518	0.8557/1.8844

TABLE VII

ABLATION EXPERIMENTS ABOUT THE DIFFERENT TRAINING DATASETS ($\mathrm{AUC}_{(D,F)}$/AUCOADP)

Training dataset	HYDICE	WHU-Hi-River	AVIRIS-1	Cri	AVIRIS-2	MUCAD-I	MUCAD-II	MUCAD-III
HYDICE	**0.9960**/2.8057	**0.9999/2.3705**	0.9728/2.3376	0.9915/**2.5176**	0.9466/**2.0953**	0.9365/1.9626	**0.9243**/1.9518	0.8557/1.8844
WHU-Hi-River	0.8588/2.0592	0.9996/2.3629	0.5740/1.5768	0.9678/2.3572	0.7139/1.9004	0.7542/1.8056	0.8580/1.8455	0.7653/1.6634
AVIRIS-1	0.9744/2.3877	0.9771/2.1406	**0.9914/2.7042**	0.9855/2.2203	0.7985/1.7891	**0.9619/2.4291**	0.9106/**2.3745**	0.8081/2.1089
Cri	0.9952/**2.9905**	0.9777/2.3280	0.7354/1.9618	**0.9997**/2.3280	**0.9566**/2.0654	0.8880/1.8919	0.8741/1.8606	0.6918/1.6922
AVIRIS-2	0.8462/2.0771	0.9757/2.2561	0.6429/1.8398	0.9854/2.4079	0.8810/2.0709	0.7688/1.7487	0.8793/2.0847	**0.8927/2.3033**

TABLE VIII

COMPARATIVE ANALYSIS OF THE STRATEGIES FOR GENERATING THE ANOMALY PIXELS ($\mathrm{AUC}_{(D,F)}$/AUCOADP)

Strategy	HYDICE	WHU-Hi-River	AVIRIS-1	Cri	AVIRIS-2	MUCAD-I	MUCAD-II	MUCAD-III
Gaussian white noise	0.9858/2.7930	0.9724/2.3402	0.9522/2.0966	**0.9948**/2.4965	0.9181/2.0796	0.9458/1.9196	0.8896/1.9311	**0.8780**/1.8642
Random white noise	**0.9970/2.8274**	0.9814/2.3028	**0.9849/2.5136**	0.9940/**2.6298**	0.9408/2.0757	**0.9582/2.0121**	0.8742/1.9247	0.7760/1.7947
Spectral shuffle	0.9960/2.8057	**0.9999/2.3705**	0.9728/2.3376	0.9915/2.5176	**0.9466/2.0954**	0.9365/1.9627	**0.9243/1.9517**	0.8557/**1.8844**

TABLE IX

SENSITIVITY ANALYSIS OF PATCH SIZE OF THE SIMULATED SAMPLES TO THE DETECTION RESULTS ($\mathrm{AUC}_{(D,F)}$/AUCOADP)

Patch size	HYDICE	WHU-Hi-River	AVIRIS-1	Cri	AVIRIS-2	MUCAD-I	MUCAD-II	MUCAD-III
5	0.9488/2.5057	0.9827/2.3618	0.9387/2.4765	0.9008/2.0905	**0.9566/2.6631**	0.9213/**2.4732**	0.9312/**2.5678**	**0.9060/2.3132**
10	**0.9960**/2.8057	**0.9999**/2.3705	0.9728/2.3376	0.9915/2.5176	0.9466/2.0954	**0.9365**/1.9627	0.9243/1.9517	0.8557/1.8844
20	0.9888/2.7343	0.9957/2.3485	**0.9860**/2.5010	**0.9979/2.8556**	0.9386/2.1095	0.9363/1.9565	**0.9364**/2.0725	0.8613/1.8463
30	0.9933/**2.8411**	0.9979/**2.6074**	0.9794/**2.7244**	0.9969/2.7615	0.9231/2.0789	0.9005/2.3205	0.8831/2.3602	0.9032/1.9254
80	0.9893/2.8319	0.9983/2.4532	0.9834/2.4553	0.9965/2.6068	0.9429/2.0716	0.9193/1.9334	0.9219/1.9252	0.8364/1.8525

only has six bands, the corresponding trained TDD model is difficult to exploit the spectral features, and we only showed the HSI datasets. The trained TDDs on five HSI datasets have all demonstrated the detection transferability (many $\mathrm{AUC}_{(D,F)} > 0.9$). The optical results can be observed across different training images. If training and inferring the TDD model on a single dataset, higher $\mathrm{AUC}_{(D,F)}$ scores can be achieved for many datasets (0.9914 on the AVIRIS-1 dataset, 0.9997 on the Cri dataset, 0.9566 on the AVIRIS-2 dataset, etc.).

On the other hand, the transfer suffers from instability issue and the instability degree varies according to the training datasets. The HYDICE dataset shows the robust detection transferability. We speculate that the instability may be related to the heterogeneity of background spectral and the noise degree brought by the real anomalies. If the background spectral only has few categories, the trained discriminative ability for deviation relationship is relatively weak (e.g., WHU-Hi-River dataset).

3) Sensitivity Analysis of Anomaly Generating Strategies: The proposed anomaly simulation strategy uses the channel shuffling operation to create the deviated spectra. In practice, we found adding random white noise was also useful. Their small difference is that the channel shuffle does not create new radiance/reflectance values, whereas adding random noise does. In Table VIII, we have compared the spectral shuffle with the Gaussian white noise and the random white noise at the same amplitude range. It is difficult to judge which one is better or worse in terms of an overall assessment. Adding white noise strategy has also refreshed the optimal accuracy on many datasets. From the view of the experimental performance, their difference can be ignored and both the strategies can generate the effective samples.

4) Sensitivity Analysis of Training Patch Sizes: The proposed anomaly simulation strategy has cropped the training image into patches for generating lots of the paired samples, where the training patch size is an adjusted hyperparameter. The related analysis is shown in Table IX. The simulated samples are all generated from the HYDICE dataset. When the patch size is set to 5, the performance of the trained model will decrease significantly, especially for the HYDICE, AVIRIS-1, and Cri datasets. For the patch sizes 10, 20, and 30, they have similar performance and all reached at a high level. We deduce that the too small patch size would lead to smooth and single background spectra, inconsistent with the spectral heterogeneity in real scenes. When the patch

TABLE X

SENSITIVITY ANALYSIS OF SIMULATED ANOMALY SIZE ON THE DETECTION RESULTS ($\text{AUC}_{(D,F)}$/AUCOADP)

Anomaly size	HYDICE	WHU-Hi-River	AVIRIS-1	Cri	AVIRIS-2	MUCAD-I	MUCAD-II	MUCAD-III
(2, 8)	0.7972/1.1852	0.9014/1.2412	0.9370/1.5249	0.9817/1.2961	**0.9442**/1.1604	0.9072/1.1017	**0.9736**/1.5620	**0.9117**/1.0345
(7, 13)	**0.9933**/**2.8411**	0.9979/**2.6074**	**0.9794**/**2.7244**	**0.9969**/**2.7615**	0.9231/**2.0789**	0.9005/**2.3205**	0.8831/**2.3602**	0.9032/**1.9254**
(12, 18)	0.9865/1.8037	**0.9986**/1.4413	0.9738/1.6416	0.9962/1.4925	0.8568/1.1105	**0.9536**/1.0612	0.9323/1.4719	0.8149/1.3706
(17, 23)	0.9383/1.6676	0.9872/1.3195	0.9200/1.3716	0.9455/1.3091	0.9055/1.0043	0.9372/1.0371	0.9122/1.4054	0.8220/1.0978

size grows to the limit of 80, satisfactory results can be still obtained. It benefits from the incorporating randomness (strong anomaly location randomness, spectral randomness, and shape randomness), which can help generate varying samples even only several original images are available. Overall, the model is robust to the training patch size except for the very small patch size (e.g., patch size 5 in the HYDICE dataset).

5) Sensitivity Analysis of Simulated Anomaly Size: The size of the simulated anomaly is an important hyperparameter for the deviation relationship learning. When training TDD, the simulated anomaly size is randomly selected from a preset range. We set the size range from the following two considerations.

1) Smaller than the half the training patch size. Since we treat the training patch as the surroundings and generate the anomalies deviating from it. The dominant area should be background and the anomaly size cannot be more than half of the patch size.

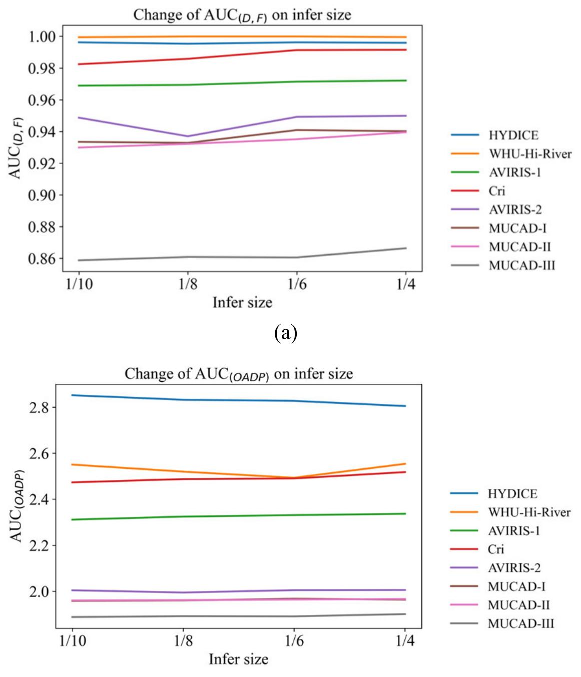
2) Simulated size cannot be too small. The training process of TDD is to learn to discriminate the deviation relationship. The too small size would bring relatively few supervisions compared with the background, increasing the risk of the model overfitting the background.

To make the effect of anomaly size clear, we have conducted related sensitivity analysis on the HYDICE dataset. Four different size ranges are set as $(\delta - 3, \delta + 3)$, where δ is set as 5, 10, 15, and 20, respectively. For each size range, we fixed the patch size 30 × 30 according to Table IX and randomly chose the anomaly height and width from this range.

The results are shown in Table X. Each setting of the anomaly size can train a valid transferring model but has an obviously different effect for the detection results. The size range (2, 8) has many worst detections such as the HYDICE ($\text{AUC}_{(D,F)}$ 0.7972) and WHU-Hi-River datasets ($\text{AUC}_{(D,F)}$ 0.9014). When the anomaly size increases from 7 to 17, the overall performance decreases obviously. The size range (17, 23) has totally exceeded half of the patch size (i.e., 15) and a large drop appears in many datasets (e.g., HYDICE, AVIRIS-I, and Cri). The size range (7, 13) has achieved the overall best performance and validated our two considerations.

Note that although the simulated size in the training process is not too small, once the trained model has learned to recognize the deviation relationship between anomalies and background, it can be generalized to tiny anomalies successfully (e.g., many anomalies in the HYDICE dataset only has 1–2 pixels and the trained model can detect them correctly).

6) Sensitivity Analysis of Infer Hyperparameters: Since TDD can infer the test image without retraining as the traditional two-step methods, there are no training hyperparameters needed to be adjusted. This is very friendly considering the great impact of training hyperparameters on the detection results of the two-step methods. Despite this, TDD has two hyperparameters at the infer stage including the infer size and the overlap setting. Given a test image, the TDD model infers the whole image patch by patch. The infer size decides the context area to define the anomaly and the overlap decides whether to keep the marginal results. To figure out the sensitivity of TDD on these infer hyperparameters, we varied the hyperparameters and analyzed the resulting accuracy change.



(a)

(b)

Fig. 10. Sensitivity analysis of $\text{AUC}_{(D,F)}$ (a) and AUC_{OADP} (b) on the infer size.

Fig. 10 and Table XI show the sensitivity results about the infer size and the overlap setting, respectively. There are huge differences in the image size of different datasets, and the horizontal axis in Fig. 10 represents the ratio of patch to the entire image. It can be clearly seen that most of the lines are nearly flat and there is no significant performance drop. In Table XI, we found using the overlap setting can bring a slight and robust promotion for both the metrics. Using the overlap setting can be considered as the optimal and default infer hyperparameter without adjusting. The results in center area are always with high confidence and accuracy. Although the overlap may increase the amount of processed data, the resulting time increasing is slightly under GPU acceleration.

TABLE XI
SENSITIVITY ANALYSIS OF USING THE OVERLAP SETTING (AUC$_{(D,F)}$/AUCOADP)

Dataset	w/o Overlap	w/ Overlap
HYDICE	0.9956/**2.9117**	**0.9960**/2.8057
WHU-Hi-River	0.9990/**2.5600**	**0.9999**/2.3705
AVIRIS-1	0.9694/2.3242	**0.9728/2.3376**
Cri	0.9863/2.5171	**0.9915/2.5176**
AVIRIS-2	0.9171/1.9042	**0.9466/2.0953**
MUCAD-I	0.9348/1.9608	**0.9365/1.9626**
MUCAD-II	0.9181/1.9430	**0.9243/1.9518**
MUCAD-III	**0.8561**/1.8817	0.8557/**1.8844**

TABLE XII
EXECUTION TIME COMPARISON

Dataset	GRX	LRASR	ADLR	CRD	Auto-AD	DeepLR	TDD
HYDICE	0.14	23.48	49.32	60.31	67.81	296.78	55.35
WHU-Hi-River	0.21	45.76	74.65	47.45	59.48	38.53	5.49
AVIRIS-1	0.33	28.81	30.46	79.47	154.16	159.59	1.91
Cri	3.73	519.39	1258.50	1024.84	8.91	31.49	4.21
AVIRIS-2	0.09	6.24	17.57	19.13	48.89	23.56	1.91
MUCAD-I	0.20	757.04	23.74	145.76	29.38	11.69	7.88
MUCAD-II	0.21	774.31	15.31	140.13	37.34	14.74	33.00
MUCAD-III	0.20	814.94	28.96	144.37	23.86	13.71	32.85

Overall, TDD only has one infer hyperparameter (i.e., infer size) actually, and TDD is very insensitive to it.

7) Execution Time: One of the advantages of the one-step paradigm is the execution efficiency because the training time is removed. The one-step paradigm is more practical in many real-time anomaly applications, where the captured image can be processed directly. The execution times are recorded in Table XII. Compared with the SOTA two-step paradigm methods Auto-AD and DeepLR, TDD can reduce the processing time by an order of magnitude on most datasets. For example, the processing time on the AVIRIS-1 dataset was reduced from 159.59 to 1.91. A larger reduction can be observed compared with the representation-based methods (i.e., LRASR, ADLR, and CRD). GRX is the only model faster than TDD. It is a matter of balance between the time extension and accuracy improvement of TDD compared with GRX. Compared with the remaining comparison methods, TDD is better in both accuracy and time.

V. CONCLUSION

In this article, we propose and instantiate the one-step detecting paradigm for HAD task. Compared with the mainstream two-step paradigm, "spectral deviating relationship" is focused and there are two main advantages: 1) no proxy task is adopted (e.g., reconstructed or generated) and the model is optimized directly for the HAD task and 2) the model of one-step paradigm has the transfer ability. Given an unseen image, it can be inferred directly without retraining. This is practical especially for the real-time applications.

Furthermore, TDD is designed as a paradigm instantiation, where the proposed anomaly sample simulation strategy drives the unsupervised model training, and two plug-and-play attention modules are designed to enhance the relationship learning. Under six public datasets, TDD has demonstrated the superior detection and transfer ability of the one-step paradigm. In the model analysis section, we also found the transfer ability of TDD is bounded, which may be limited by the used training samples. With the development of the large foundation model and datasets, we hope this research can facilitate more robust instantiation of one-step paradigm.

ACKNOWLEDGMENT

The authors would like to thank the authors of LRASR, ADLR, CRD, Auto-AD, and DeepLR for sharing their codes.

REFERENCES

[1] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Deep low-rank prior for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5527017.

[2] C. Li, B. Zhang, D. Hong, J. Yao, and J. Chanussot, "LRR-Net: An interpretable deep unfolding network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5513412.

[3] J. Lei, W. Xie, J. Yang, Y. Li, and C.-I. Chang, "Spectral–spatial feature extraction for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 8131–8143, Oct. 2019.

[4] S. Matteoli, M. Diani, and G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 5–28, Jul. 2010.

[5] K. Tan, F. Wu, Q. Du, P. Du, and Y. Chen, "A parallel Gaussian–Bernoulli restricted Boltzmann machine for mining area classification with hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 2, pp. 627–636, Feb. 2019.

[6] S. Wang, X. Wang, L. Zhang, and Y. Zhong, "Auto-AD: Autonomous hyperspectral anomaly detection network based on fully convolutional autoencoder," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5503314.

[7] N. Huyan, X. Zhang, D. Quan, J. Chanussot, and L. Jiao, "AUD-Net: A unified deep detector for multiple hyperspectral image anomaly detection via relation and few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 27, 2022, doi: 10.1109/TNNLS.2022.3213023.

[8] L. Gao, D. Wang, L. Zhuang, X. Sun, M. Huang, and A. Plaza, "BS3LNet: A new blind-spot self-supervised learning network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504218.

[9] J. Li, X. Wang, H. Zhao, S. Wang, and Y. Zhong, "Anomaly segmentation for high-resolution remote sensing images based on pixel descriptors," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 4, pp. 4426–4434.

[10] X. Lu, W. Zhang, and J. Huang, "Exploiting embedding manifold of autoencoders for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1527–1537, Mar. 2020.

[11] W. Xie, J. Lei, B. Liu, Y. Li, and X. Jia, "Spectral constraint adversarial autoencoders approach to feature representation in hyperspectral anomaly detection," *Neural Netw.*, vol. 119, pp. 222–234, Nov. 2019.

[12] Y. Li, T. Jiang, W. Xie, J. Lei, and Q. Du, "Sparse coding-inspired GAN for hyperspectral anomaly detection in weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5512811.

[13] S. Arisoy, N. M. Nasrabadi, and K. Kayabol, "GAN-based hyperspectral anomaly detection," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 1891–1895.

[14] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[15] F. Ye, C. Huang, J. Cao, M. Li, Y. Zhang, and C. Lu, "Attribute restoration framework for anomaly detection," *IEEE Trans. Multimedia*, vol. 24, pp. 116–127, 2022.

[16] D. Wang, L. Gao, Y. Qu, X. Sun, and W. Liao, "Frequency-to-spectrum mapping GAN for semisupervised hyperspectral anomaly detection," *CAAI Trans. Intell. Technol.*, vol. 8, no. 4, pp. 1258–1273, Dec. 2023.

[17] Z. Wang, X. Wang, K. Tan, B. Han, J. Ding, and Z. Liu, "Hyperspectral anomaly detection based on variational background inference and generative adversarial network," *Pattern Recognit.*, vol. 143, Nov. 2023, Art. no. 109795.

[18] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*.

[19] W. Li, G. Wu, and Q. Du, "Transferred deep learning for anomaly detection in hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 597–601, May 2017.

[20] J. Liu, Z. Hou, W. Li, R. Tao, D. Orlando, and H. Li, "Multipixel anomaly detection with unknown patterns for hyperspectral imagery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5557–5567, Oct. 2022.

[21] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Aug. 1990.

[22] C.-I. Chang, "Target-to-anomaly conversion for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5540428.

[23] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 2, pp. 388–397, Feb. 2005.

[24] Q. Guo, B. Zhang, Q. Ran, L. Gao, J. Li, and A. Plaza, "Weighted-RXD and linear filter-based RXD: Improving background statistics estimation for anomaly detection in hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2351–2366, Jun. 2014.

[25] P. Gurram and H. Kwon, "Support-vector-based hyperspectral anomaly detection using optimized kernel parameters," *IEEE Trans. Geosci. Remote Sens.*, vol. 8, no. 6, pp. 1060–1064, Nov. 2011.

[26] W.-M. Liu and C.-I. Chang, "Multiple-window anomaly detection for hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 644–658, Apr. 2013.

[27] W. Xie, T. Jiang, Y. Li, X. Jia, and J. Lei, "Structure tensor and guided filtering-based algorithm for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4218–4230, Jul. 2019.

[28] C.-I. Chang, "Hyperspectral target detection: Hypothesis testing, signal-to-noise ratio, and spectral angle theories," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–23, 2021.

[29] C.-I. Chang, "Hyperspectral anomaly detection: A dual theory of hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5511720.

[30] C.-I. Chang, "Constrained energy minimization anomaly detection for hyperspectral imagery via dummy variable trick," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5517119.

[31] C.-I. Chang, C.-Y. Lin, P.-C. Chung, and P. F. Hu, "Iterative spectral–spatial hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5504330.

[32] T. Jiang, W. Xie, Y. Li, J. Lei, and Q. Du, "Weakly supervised discriminative learning with spectral constrained generative adversarial network for hyperspectral anomaly detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6504–6517, Nov. 2022.

[33] T. Guo, L. He, F. Luo, X. Gong, Y. Li, and L. Zhang, "Anomaly detection of hyperspectral image with hierarchical antinoise mutual-incoherence-induced low-rank representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5510213.

[34] W. Xie, X. Zhang, Y. Li, J. Lei, J. Li, and Q. Du, "Weakly supervised low-rank representation for hyperspectral anomaly detection," *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3889–3900, Aug. 2021.

[35] L. Du, Z. Wu, Y. Xu, W. Liu, and Z. Wei, "Kernel low-rank representation for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 477–480.

[36] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[37] Y. Zhang, D. Bo, L. Zhang, and S. Wang, "A low-rank and sparse matrix decomposition-based Mahalanobis distance method for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1376–1389, Mar. 2015.

[38] C.-I. Chang, H. Cao, S. Chen, X. Shang, C. Yu, and M. Song, "Orthogonal subspace projection-based go-decomposition approach to finding low-rank and sparsity matrices for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 3, pp. 2403–2429, Mar. 2021.

[39] C.-I. Chang, H. Cao, and M. Song, "Orthogonal subspace projection target detector for hyperspectral anomaly detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 4915–4932, 2021.

[40] C.-I. Chang and J. Chen, "Hyperspectral anomaly detection by data sphering and sparsity density peaks," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526321.

[41] C.-I. Chang, "Effective anomaly space for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5526624.

[42] S. Sun, J. Liu, X. Chen, W. Li, and H. Li, "Hyperspectral anomaly detection with tensor average rank and piecewise smoothness constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8679–8692, Nov. 2023.

[43] X. Zhang, G. Wen, and W. Dai, "A tensor decomposition-based anomaly detection algorithm for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5801–5820, Oct. 2016.

[44] Z. Chen, B. Yang, and B. Wang, "A preprocessing method for hyperspectral target detection based on tensor principal component analysis," *Remote Sens.*, vol. 10, no. 7, p. 1033, Jun. 2018.

[45] L. Li, W. Li, Y. Qu, C. Zhao, R. Tao, and Q. Du, "Prior-based tensor approximation for anomaly detection in hyperspectral imagery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 3, pp. 1037–1050, Mar. 2022.

[46] D. Wang, L. Zhuang, L. Gao, X. Sun, M. Huang, and A. Plaza, "PDBSNet: Pixel-shuffle down-sampling blind-spot reconstruction network for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5511914.

[47] L. Wang, X. Wang, A. Vizziello, and P. Gamba, "RSAAE: Residual self-attention-based autoencoder for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5510614.

[48] J. Theiler, "By definition undefined: Adventures in anomaly (and anomalous change) detection," in *Proc. 6th IEEE Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Jun. 2014, pp. 1–4.

[49] C.-I. Chang, S. Chen, S. Zhong, and Y. Shi, "Exploration of data scene characterization and 3D ROC evaluation for hyperspectral anomaly detection," *Remote Sens.*, vol. 16, no. 1, p. 135, Dec. 2023.

[50] H. Zhao, Y. Zhong, X. Wang, and H. Shu, "One-class risk estimation for one-class hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5518017.

[51] J. Li, X. Wang, H. Zhao, X. Hu, and Y. Zhong, "Detecting pine wilt disease at the pixel level from high spatial and spectral resolution UAV-borne imagery in complex forest landscapes using deep one-class classification," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102947.

[52] H. Zhao, X. Wang, J. Li, and Y. Zhong, "Class prior-free positive-unlabeled learning with Taylor variational loss for hyperspectral remote sensing imagery," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 16827–16836.

[53] Y. Kong and T. Yu, "A deep neural network model using random forest to extract feature representation for gene expression data classification," *Sci. Rep.*, vol. 8, no. 1, p. 16477, Nov. 2018.

[54] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, vol. 9351. Cham, Switzerland: Springer, 2015, pp. 234–241.

[55] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.

[56] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning pixel-wise contextual attention for saliency detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3089–3098.

[57] C. Szegedy et al., "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[58] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[59] C. Pu, H. Huang, X. Shi, and T. Wang, "Semisupervised spatial–spectral feature extraction with attention mechanism for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[60] C. M. Bishop, *Pattern Recognition and Machine Learning*, vol. 2. Cham, Switzerland: Springer, 2006, pp. 5–43.

[61] T. Hupel and P. Stütz, "Adopting hyperspectral anomaly detection for near real-time camouflage detection in multispectral imagery," *Remote Sens.*, vol. 14, no. 15, p. 3755, Aug. 2022.

[62] Y. Xu, Z. Wu, J. Li, A. Plaza, and Z. Wei, "Anomaly detection in hyperspectral images based on low-rank and sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 1990–2000, Apr. 2016.

[63] Y. Qu et al., "Hyperspectral anomaly detection through spectral unmixing and dictionary-based low-rank decomposition," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4391–4405, Aug. 2018.

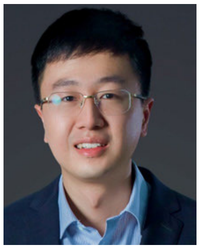
[64] Y. Yang, H. Su, Z. Wu, and Q. Du, "Saliency-guided collaborative-competitive representation for hyperspectral anomaly detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 6843–6859, 2023.

[65] 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.

[66] C.-I. Chang, "Comprehensive analysis of receiver operating characteristic (ROC) curves for hyperspectral anomaly detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5541124, doi: 10.1109/TGRS.2022.3211786.

Jingtao Li received the B.S. degree from the School of Geography and Information Engineering, China University of Geosciences, Wuhan, China, in 2021. He is currently pursuing the Ph.D. degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan.

His major research interests anomaly detection in remote sensing.



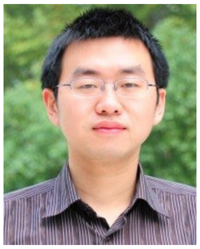
Xinyu Wang (Member, IEEE) received the B.S. degree in photogrammetry and remote sensing and the Ph.D. degree in communication and information systems from Wuhan University, Wuhan, China, in 2014 and 2019, respectively.

Since 2019, he has been an Associate Research Fellow at the School of Remote Sensing and Information Engineering, Wuhan University. His major research interests include hyperspectral data processing and applications.

Shaoyu Wang received the Ph.D. degree from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University, Wuhan, China, in 2022.

He is now conducting postdoc researches with the College of Agriculture and Life Sciences, Seoul National University. His major research interests include hyperspectral information processing and remote sensing of GHG fluxes.

Hengwei Zhao received the B.S. degree in surveying and mapping engineering from the School of Resources and Civil Engineering, Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree in photogrammetry and remote sensing with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China.



Yanfei Zhong (Senior Member, IEEE) received the B.S. degree in information engineering and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2002 and 2007, respectively.

Since 2010, he has been a Full Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing (LIESMARS), Wuhan University. He organized the Intelligent Data Extraction, Analysis and Applications of Remote Sensing (RSIDEA) Research Group. He has published more than 100 research articles in international journals, such as *Remote Sensing of Environment*, *ISPRS Journal of Photogrammetry and Remote Sensing*, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING. His research interests include hyperspectral remote sensing information processing, high-resolution remote sensing image understanding, and geoscience interpretation for multisource remote sensing data and applications.

Dr. Zhong is a fellow of the Institution of Engineering and Technology (IET). He was a recipient of the 2016 Best Paper Theoretical Innovation Award from the International Society for Optics and Photonics (SPIE). He won the Second-Place Prize in 2013 IEEE GRSS Data Fusion Contest and the Single-view Semantic 3-D Challenge of the 2019 IEEE GRSS Data Fusion Contest, respectively. He is currently serving as an Associate Editor for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing and the *International Journal of Remote Sensing*.