

Distance Constraint-Based Generative Adversarial Networks for Hyperspectral Image Classification

Anyong Qin^{ID}, Zhuolin Tan^{ID}, Ran Wang, Yongqing Sun^{ID}, Member, IEEE, Feng Yang^{ID}, Yue Zhao^{ID}, and Chenqiang Gao^{ID}

Abstract—Hyperspectral image (HSI) classification suffers from two serious problems: one is the limited labeled pixels and the other is the class imbalance problem. As a result, the number of labeled pixels in many categories is not sufficient to characterize the spectral–spatial information and train a satisfying deep model. By making full use of the information of unlabeled pixels, semisupervised methods can provide better classification performance in the case of limited labeled pixels. However, they do not consider the imbalance in the HSI data. As a method of data enhancement, generative adversarial networks (GANs) focus on the above two problems and have also been widely used for the task of HSI classification. In this work, we propose a distance constraint-based GAN (DGAN) method for HSI classification to address these two problems. The DGAN employs the convolution autoencoder (AE) to extract the latent features of the HSI samples and considers the reconstructed samples from the AE as the real samples for the later classifier and discriminator. In addition, the DGAN uses two distance constraints to solve the problems of the few labeled samples and class imbalance: the one latent-data distance constraint enforcing the generator to generate HSI samples for each class (especially the minority class) and another discriminator-score distance constraint guiding the generator to synthesize samples that resemble the real HSI samples. Finally, the generated samples are combined classwise with the reconstructed samples and the real HSI samples to learn the parameters of the classifier and discriminator. Experimental results show that our method achieves the state-of-the-art performance in terms of overall accuracy (OA) when trained with only 0.5%–4% of datasets from Indian Pines, Pavia University, and Botswana. Specifically, our method demonstrates improvements of 5.48%, 8.79%, and 0.91% on these three datasets. It reveals the great potential of the DGAN model in generating the HSI samples for each class,

Manuscript received 5 November 2022; revised 15 January 2023, 10 April 2023, and 24 April 2023; accepted 28 April 2023. Date of publication 10 May 2023; date of current version 23 May 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62176035 and Grant 62201111; in part by the Natural Science Foundation of Chongqing, China, under Grant cstc2020jcyj-msxmX0835 and Grant cstc2021jcyj-bshX0088; in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZDK202100606, Grant KJQN202000647, and Grant KJQN202100646; in part by the China Postdoctoral Science Foundation under Grant 2021MD703940 and Grant 2022MD713684; and in part by the Special Funding for Chongqing Postdoctoral Research Project. (*Corresponding author: Chenqiang Gao.*)

Anyong Qin, Zhuolin Tan, Feng Yang, Yue Zhao, and Chenqiang Gao are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China, and also with the Chongqing Key Laboratory of Signal and Information Processing, Chongqing 400065, China (e-mail: qinay@cqupt.edu.cn; gaocq@cqupt.edu.cn).

Ran Wang is with the School of International, Chongqing University of Posts and Telecommunications, Chongqing 400065, China.

Yongqing Sun is with the Computer and Data Science Laboratories, Nippon Telegraph and Telephone Corporation, Kanagawa 239-0847, Japan.

Digital Object Identifier 10.1109/TGRS.2023.3274778

which contributes to improving the classification performance of the HSI data.

Index Terms—Deep learning, distance constraints, generative adversarial networks (GANs), hyperspectral image (HSI) classification.

I. INTRODUCTION

HYPERSPECTRAL images (HSIs) contain hundreds of narrow spectral bands from a region, which can be the surface of the Earth, the atmosphere, ocean, and so on. It is a critical prerequisite in a wide range of applications, including geological science, ecological science, and hydrological science [1], [2], [3]. HSI classification that assigns each pixel a category is a fundamental and challenging problem in the remote sensing field, and how to fully explore the spectral bands to classify the pixels is one of the most vibrant fields in the HSI community [4].

The traditional pixel-based classification methods deal with a single image pixel separately. These methods mainly use spectral features and lack attention to the adjacent regions. The information of domain pixel greatly affects the accuracy of information extraction. They also tend to have salt-and-pepper effects in classification results [5]. On the contrary, object-based remote sensing feature extraction methods involve pixel merging and object segmentation and can overcome the salt-and-pepper effects. These methods comprehensively consider the spectral characteristics, size, texture, adjacent relationship, and other factors, which can provide more basis for the classification of ground objects [6]. It has become the mainstream method in the current remote sensing field [7]. Due to these advantages, our method uses the same spatial information processing for hyperspectral data to fully consider its intraclass similarity and interclass dissimilarity in adjacent pixels. At present, the deep learning technique has attracted broad attention in the remote sensing field and the deep learning-based methods have become the mainstream [4], [8], [9]. A series of deep learning-based methods has emerged for the HSI classification [8]. However, these methods heavily rely on a large number of training samples (usually 20%–30%). We take the Indian Pines HSI as an example. Fig. 1 shows the accuracies of the state-of-the-art 2-D convolutional neural network (2DCNN) model [8] on two different numbers of labeled samples. As the labeled samples decrease from 20% to 3%, the performance drops significantly by about 14.61%. Different from other computer vision tasks, e.g., object detection,

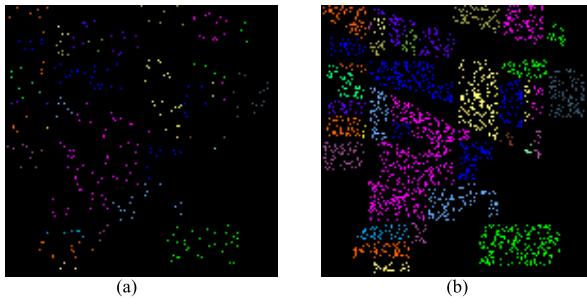


Fig. 1. Comparison between (a) few labeled pixels (labeling ratio: 3% and OA of the state-of-the-art 2DCNN [8]: 80.10%) and (b) lot of labeled pixels (labeling ratio: 20% and OA of the state-of-the-art 2DCNN: 94.71%) in the Indian Pines HSI.

labeling pixels in HSIs is particularly time-consuming and expensive because it requires experts to obtain the ground-truth data for these images by field sampling. Therefore, exploring the HSI classification for good performance with few labeled samples becomes particularly significant [10].

One of the popular methods to improve the classification performance with few labeled samples is generative adversarial networks (GANs). The GANs, which combine the benefits of the CNNs and the generative models, have received a lot of attention due to their abilities to generate high-quality samples [11]. Therefore, GAN models have begun to show their major superiority for the HSI classification task [12], [13], [14], [15]. Specifically, the generators typically use the CNN to generate hyperspectral samples from random noise, and then, the generated ones and the real samples together form the training set of the model. However, due to ignoring the class information of the HSI data, the advanced GANs in [12] and [13] produce nearly identical hyperspectral samples from different random noises, which has limited improvement for the classification performance. In addition, the above uncontrolled generated samples do not alleviate the problem of few labeled samples and further enlarge the problem of class imbalance, which already exists in HSI datasets. In fact, class information can play a crucial role for generating a large number of diverse samples. More specifically, if the GAN can preserve the local structures of different classes in the latent space when generating hyperspectral samples, then the generator can generate high-quality samples for each class.

To address the above local structures problem in the process of generating sufficient hyperspectral samples for each class, we propose a novel distance constraint-based GAN (DGAN) model for HSI classification, motivated by the improved GAN using distance constraints to improve the quality of the generated samples [16]. Different from some advanced GAN methods in [12], [13], [14], and [15] for HSI classification task, the proposed DGAN can preserve the local structures of the latent features into the high-dimensional generated hyperspectral samples, to focus on the class information in the HSI data. In other words, the proposed DGAN mainly depends on two distance constraints to improve the generator, and as a result, produces high-quality hyperspectral samples for each class. One of the distance constraints is used to prevent the generator from producing hyperspectral samples that are close to each other, to deal with the class imbalance problem of the

HSI data. Another distance constraint is used to improve the generator to synthesize sufficient hyperspectral samples that can alleviate overfitting to deal with the few labeled pixels problem of the HSI data. The main contributions of this work can be summarized as follows.

- 1) We propose a novel DGAN for the HSI classification task, to tackle the problems of few labeled pixels and class imbalance for the HSI data classification.
- 2) The proposed DGAN model preserves the local structures by restricting the latent-data distance and then can generate sufficient hyperspectral samples for each class. Another discriminator-score distance is used to improve the generator to synthesize enough high-quality hyperspectral samples.
- 3) Compared with other state-of-the-art methods, the proposed DGAN model can achieve a competitive result on three well-known hyperspectral datasets with few labeled training pixels.

The rest of this article is organized as follows. The related works and introduction of several models are briefly reviewed in Section II. Section III presents the proposed DGAN method for the HSIs classification. The experimental results and analysis part are shown in Section IV, and Section V summarizes the key points of this article and discussions.

II. RELATED WORKS

In this section, we will briefly review the traditional, autoencoder (AE) network-, CNN-, and GAN-based methods for the HSI classification.

Early HSI classification methods mainly adopted the traditional machine learning algorithms to process the spectral features [17], [18], [19], [20], [21]. However, many factors, such as changes in illumination, environmental, atmospheric, and temporal conditions, usually bring the spectral intraclass variability and interclass similarity. Therefore, it is difficult to obtain the accurate classification results only using the spectral information of the HSI data. To deal with these limitations, a large number of HSI classification methods that make full use of the spatial arrangement of HSI data are proposed [22], [23], [24], [25].

Compared with the traditional approaches, the deep learning-based methods can automatically learn the high-level spectral-spatial features from the original HSI data [26], [27], [28]. AE network is a symmetrical neural network to obtain the latent features of the original data. It is a process that the encoder maps the input to a hidden space with small size, and then, the decoder decodes the features in the hidden space to obtain the reconstructed samples of the input, resulting in a compact, representative, nonlinear representation of the input. Stacked AEs (SAEs) are made up of several AEs stacked layer by layer. For example, Chen et al. [29] first applied the deep learning-based method to the HSI classification task, using SAE to extract the spectral-spatial features of HSI. Ma et al. [30] proposed a spatial updated deep AE (SDAE), which incorporates a spatial constraint to obtain the spatial information. In [31], an unsupervised feature learning method for HSI classification based on recursive SAEs was proposed.

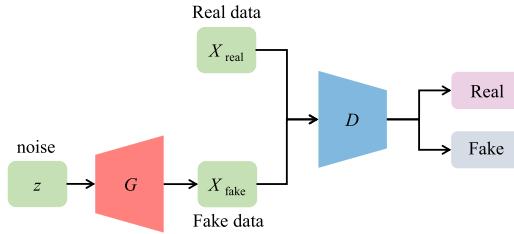


Fig. 2. Architecture of the GAN model.

CNN was also employed in AE to explore the spatial context of feature extraction. Mei et al. [32] proposed an unsupervised feature learning based on 3-D convolutional AE. The combination of AE and convolution enables full exploration of spatial structure information and can learn effective spatial–spectral features in an unsupervised manner.

In addition, traditional feature extraction methods have been gradually replaced by the CNN [33], [34], [35]. Makantasis et al. [26] applied the 2DCNN to HSI classification, conducting the classification task with a multilayer perceptron. He et al. [33] proposed a three-dimensional CNN (3DCNN) feature extraction model, which directly uses the 3-D cube HSI data to extract the joint spectral–spatial features via an end-to-end manner. Compared with the features extracted by the 2DCNN, the 3DCNN method has higher interclass distinguishability. Gong et al. [34] designed a CNN with multiscale convolution in which the multiscale filter banks further facilitate the representation power of the deep model. Moreover, in [35], the feedback attention mechanism is combined with the dense CNN to improve the discrimination of the feature representation [36]. However, the CNN-based HSI classification methods still face the problem of insufficient labeled samples, which is a common phenomenon in the hyperspectral field.

In recent years, GANs [11] as a method of data augmentation have attracted great attention. GANs are an unsupervised deep learning model proposed by Goodfellow et al. [11]. The architecture of GAN is shown in Fig. 2. It learns the real sample distribution through a game-theoretic approach. The GAN provides a good solution for image generation. GAN consists of a generator G and a discriminator D . The generator G tries to generate fake data similar to real data to deceive the discriminator D . The discriminator D tries to distinguish the data generated by the generator G from the real data. G and D constitute a dynamic game process. The generator receives a random noise vector z as input. GAN achieves an ideal Nash equilibrium by optimizing and enhancing the generator and discriminator through such an adversarial training process. Also, GAN can alleviate the problem of insufficient samples in the HSI classification task through the samples’ generation [37]. For example, Zhu et al. [12] applied an auxiliary classifier in the framework of GAN to discriminate the HSI samples. Feng et al. [38] constructed a multiclass spatial–spectral GAN model to synthesize the spectral information samples and spatial information samples by two generators. Zhong et al. [13] proposed a framework that combines GAN and conditional random field (GAN-CRF), which integrated the semisupervised deep learning and

probabilistic graphical models. Based on the CapsNET [39], Wang et al. [40] continued to propose the 1-D structure triple generative adversarial network (TripleGAN), which considered the location and orientation in the process of the feature extraction. To avoid these GAN-based models generating nearly identical hyperspectral samples, Wang et al. [14] proposed an adaptive DropBlock-enhanced GAN (ADGAN) for the HSI classification. In addition, in order to avoid these GAN-based models ignoring the minority class and aggravating the class imbalance problem, Roy et al. [15] proposed a 3-D generative adversarial minority oversampling (3D-HyperGAMO) method to deal with the class imbalance problem in the HSI classification.

Although the GAN can alleviate the problem that CNN methods need enough training samples by generating samples. Most of the above GAN methods focus on addressing the problem from the perspective of increasing the overall number of labeled samples and ignore the class imbalance that already exists in the HSI data. Moreover, the uncontrolled generated samples not only can not alleviate the problem of few labeled samples but also are not conducive to solving the problem of class imbalance. Thus, this work proposes a novel DGAN model, to ensure that the generator can generate high-quality samples for each class.

III. PROPOSED METHOD

In fact, manually labeling the HSI data is very time-consuming and difficult, and the most available HSI datasets are highly imbalanced. This results in insufficient samples and class imbalance in the training set. Here, we propose a DGAN model that samples the noises from the latent feature space of each class and then generates more new HSI samples using the noises.

More specifically, as shown in Fig. 3, the DGAN first uses the AE to obtain the latent feature of each pixel and considers the reconstructed hyperspectral samples from AE as the real samples for the later discriminator and classifier. Then, by explicitly restricting the distances between the latent features and the corresponding data samples, the proposed DGAN can prevent the generator from producing many HSI samples that are close to each other, which, as a result, guarantees the class diversity of the generated hyperspectral samples. Simultaneously, we use the discriminator-score distance constraint to guide the generator to synthesize samples that resemble the real hyperspectral samples, in order to increase the size of the labeled hyperspectral samples needed to train the classifier.

A. Architecture of the Proposed DGAN

The framework of the proposed DGAN model is shown in Fig. 3. In the data preparation part, due to the high redundancy of the HSI bands, it is difficult for the generator to understand the distribution of the real data. Therefore, it is usually necessary to reduce the dimensionality of the data first. Also, we reduce the spectral bands of the hyperspectral data to three components by the principal component analysis

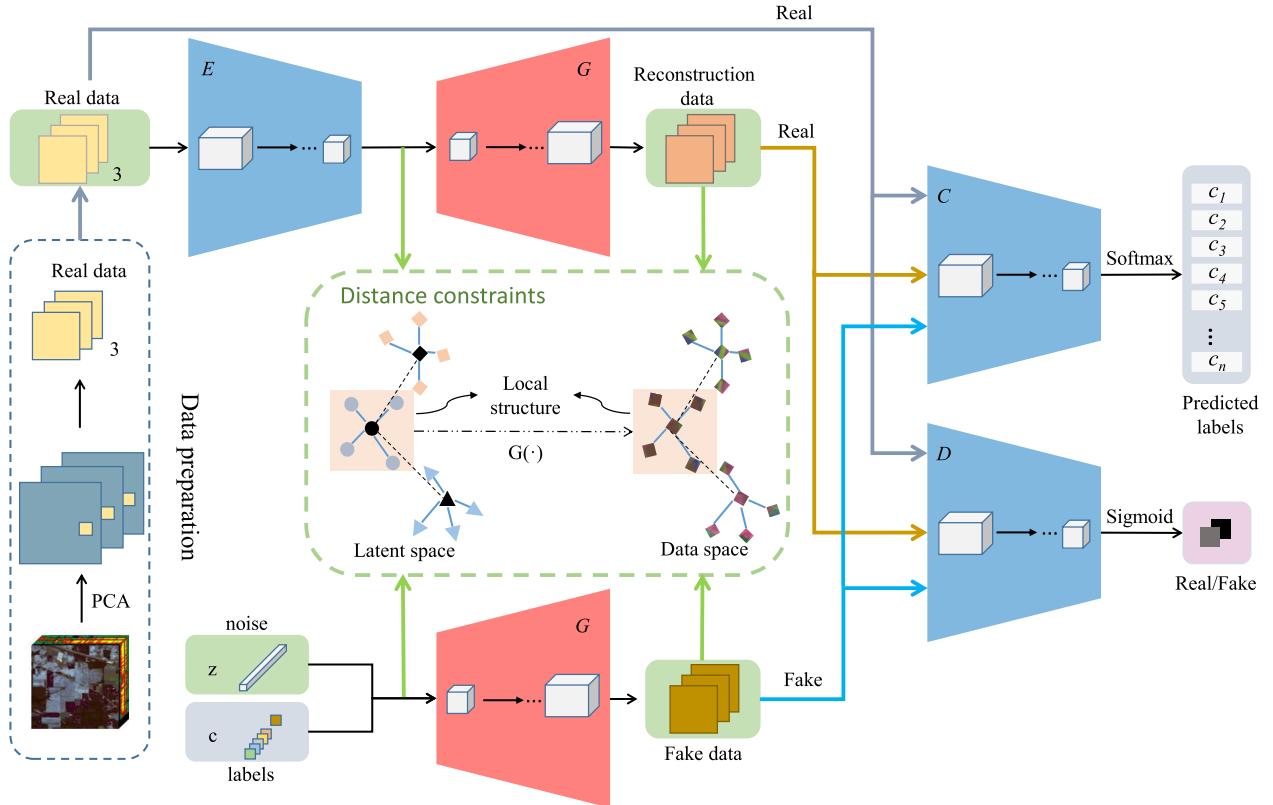


Fig. 3. Framework of the DGAN for HSI classification. E, G, C, and D represent the encoder, generator (decoder), classifier, and discriminator in the model, respectively. The green dashed box explains the variation of the data structure from the latent space to the data space under the distance constraints, which aims to preserve the local structure of the latent data.

(PCA) [41], [42], which helps to get a robust generator and reduces computational complexity.

To extract deep features from the HSI data, we use an AE [43] to improve the generator. The AE consists of an encoder and a decoder. The AE compresses the input data into a latent space representation and then uses this representation to reconstruct the input samples at the output, leading to a compact and significant representation of the input data. To keep the training stable and simple, the generator uses the same structure as the decoder. Then, the reconstructed data generated by the decoder are fed into the classifier and discriminator as “real” data. We use the difference between the reconstructed samples and the real input samples to fool the discriminator, thus slowing down the convergence rate of the discriminator [16]. Once the discriminator converges fast, the generator will no longer update the training, thus performing poorly.

As shown in Fig. 3, the second G receives the noise z and the labels c as input. The noise z obeys a uniform distribution in $[-1, 1]$. In the process of merging the noise and label information, we transform the noise to a dense vector in multidimensional space by the embedding layer and then multiply by the labels c . The way to fuse two vectors has been verified to be better than direct cascade and makes the fused vector space more abundant [44]. Finally, the generated fake data are input into the classifier and discriminator. After inputting the real data, reconstructed data, and fake data, the classifier C outputs the predicted class labels through a

softmax activation function, and the discriminator D outputs the real and fake judgments of the data through a sigmoid activation function. However, the ability of the GAN using the AE is still limited, which cannot fully tackle the insufficient labeled pixels and the imbalanced data. In Section III-B, we will introduce two distance constraints into the proposed framework.

B. Distance Constraints

Via the first distance constraint, we establish an important connection between the latent features and data samples to generate new samples for each class. As can be seen from Fig. 3, there are two kinds of variables in the latent space, the intermediate variables of the AE and the noise variables with label information. In the latent space, a cluster may belong to the same category. We aim to achieve: when these latent variables are mapped through the generator to data samples in the data space, the data samples can retain the same local structures as the latent variables to focus on the class invariance of the HSI data. This allows the generator to produce samples of the majority and minority classes and prevents the generator from producing many samples that are close to each other, which will further lead to a bias training dataset toward the majority classes.

Take the HSI data as an example. In the Indian Pines dataset, we obtain the feature representation of the pixels belonging to the Corn Notill, Alfalfa, and Grass Pasture Mowed class,

in which the last two classes belong to the minority class. After generating by G , the synthesized sample clusters will also belong to the three classes, and the samples within each cluster will also be the same class. The idea is to ensure that the generator can synthesize a sufficient number of high-quality samples for each category. To achieve this goal, we constrain the distance between the variables using two distance metrics

$$x_i = G(z_i), x_j = G(z_j) : g(z_i, z_j) > \delta_z \rightarrow f(x_i, x_j) > \delta_x \quad (1)$$

where $\{z_i\}$ are the latent variables and $\{x_i\}$ are the synthesized samples generated by G . g and f represent two distance metrics in latent and data space, respectively. Likewise, δ_z and δ_x are two different thresholds in the two spaces. Equation (1) denotes: if the distance g between the latent variables is small, the corresponding distance f between the data samples is also small and vice versa. Therefore, we apply the distance metrics to preserve the structures of the low-dimensional latent space into the high-dimensional data space. The local invariance promotes the generator G to synthesize high-quality samples for all classes of the HSI data.

In order to avoid the distance differences under the spaces of different dimensions, we calculate the matching scores of the two distributions, which is more accurate under the same criterion. Here, ω , θ , γ , and ζ are the parameters of the encoder E , generator G , discriminator D , and the classifier C , respectively. The matching score $g(E_\omega(x), v_z)$ of the latent variable $E_\omega(x)$ and the variable v_z is computed as follows:

$$g(E_\omega(x), v_z) = M_d(\mathbb{E}_x E_\omega(x) - \mathbb{E}_z v_z) \quad (2)$$

and the matching score $f(x, G_\theta(v_z))$ of the reconstructed data $G_\theta(E_\omega(x))$ and the generator data $G_\theta(v_z)$ is computed as

$$f(x, G_\theta(v_z)) = M_d(\mathbb{E}_x G_\theta(E_\omega(x)) - \mathbb{E}_z G_\theta(v_z)) \quad (3)$$

where x is the real HSI data, v_z is the variable obtained by the noise z and label c , \mathbb{E}_x and \mathbb{E}_z denote the expectation of the $G_\theta(E_\omega(x))$ and $G_\theta(v_z)$, respectively, and M_d represents the average of all dimensional values. After defining the distance of the different spaces, we establish the following distance restrictions on these two spaces (latent-data distance constraint):

$$L_W(\omega, \theta) = \|f(x, G_\theta(v_z)) - \lambda_w g(E_\omega(x), v_z)\|_2^2. \quad (4)$$

Since the output value of M_d in high-dimensional space is small, we use λ_w to balance the difference in dimensions. We compute it: $\lambda_w = (d_z/d_x)^{1/2}$, where d_z and d_x represent the dimensions of latent samples and data samples, respectively. Finally, the distance constraint [see (4)] between the latent and data space is added to the AE as a regularization term, which explicitly considers the local invariance of the data distribution. Only such diverse samples can provide sufficient class information for the classifier C and then improve the classification performance of HSI data.

The regular objective function of G is $L_s - L_c$ [44], i.e., $L_s = E[\log P(S = \text{real}|X_{\text{real}}) + E[\log P(S = \text{fake}|X_{\text{fake}})]$ and $L_c = E[\log P(C = c|X_{\text{real}}) + E[\log P(C = c|X_{\text{fake}})]$, in which L_s is the log likelihood of the right source of

the input data and L_c is the log likelihood of the right class labels. However, when using this objective function, the generator G is also prone to generate a large number of identical samples, and this leads to mode collapse easily [11]. Therefore, we propose to use the discriminator-score distance to measure the real HSI patches and the fake patches and then train a stronger generator. The distance constraint aligns the generated HSI samples and real HSI samples with ℓ_1 -norm. Also, the discriminator-score distance constraint is

$$L_G(\theta) = |\mathbb{E}_x \sigma(D_\gamma(x)) - \mathbb{E}_z \sigma(D_\gamma(G_\theta(v_z)))| \quad (5)$$

where $D_\gamma(x)$ represents the output of the discriminator for the real HSI samples, $D_\gamma(G_\theta(v_z))$ is the output of the discriminator for the fake HSI samples, and σ denotes the sigmoid activation function. We compute the ℓ_1 distance after the outputs.

The two distance constraints improve G from different views: 1) requires the generator to produce HSI samples for different classes and 2) guides the generator to synthesize samples that resemble the real HSI data. Their common goal is to prompt the DGAN to generate high-quality samples for the majority and minority classes of the HSI data.

C. Objective Functions

The objective function guides the network parameter learning and representation learning through the backpropagation of the errors between the predicted results and the ground truth. We apply the idea of (1) to improve the generator through the AE. The two main reasons for conducting AE training are: 1) to regularize the parameters of the generator during each training iteration and 2) to guide the generator to synthesize fake samples whose distribution is similar to that of the real training samples. Therefore, the objective function of AE is given as follows:

$$\min_{\omega, \theta} L_R(\omega, \theta) + \lambda_r L_W(\omega, \theta) \quad (6)$$

where ω and θ are the parameters of the encoder and generator, respectively. $L_R(\omega, \theta) = \|x - G_\theta(E_\omega(x))\|^2$ is the conventional objective function of AE and $L_W(\omega, \theta)$ is the additional distance regularization term of (4). λ_r is a constant. Due to the usual reconstruction error, we represent the reconstructed samples $G_\theta(E_\omega(x))$ obtained by the AE as $x + \epsilon$, where ϵ is the reconstruction error. Since the capacity of E and G is large enough, such errors are usually small. Thus, it is reasonable to regard the reconstructed samples as the “real” samples. Pixel-level reconstruction, such as AE, may not be accurate enough. To avoid this problem, we use feature matching distance [45] in discriminator to obtain $L_R(\omega, \theta)$

$$L_R(\omega, \theta) = \|\Phi(x) - \Phi(G_\theta(E_\omega(x)))\|^2 \quad (7)$$

where Φ is the output of the last convolution layer in the discriminator. The parameters of the first iteration are randomly initialized. In summary, we enforce the latent-data distance constraint on AE to prevent the generator from producing many HSI samples that are close to each other, which can avoid the imbalance problem in the HSI classification task.

In addition to the latent-data distance constraint, G is further improved by the discriminator-score distance constraint, i.e., (5). We propose to combine the benefits of the latent-data distance constraint and the discriminator-score distance constraint, thereby making the generator to generate sufficient high-quality HSI samples for each class. Then, the objective function of G is

$$\min_{\theta} \mathcal{L}_G(\theta) = L_G(\theta). \quad (8)$$

The traditional discriminator objective function will make the discriminator converge too quickly, which is not conducive to the adversarial training between the generator and the discriminator. This will also result in an inability to provide a reliable path for the gradient updates of the generator, causing the gradient vanishing. As a result, the GAN will be hard to generate high-quality HSI samples for the minority class. Therefore, we propose to train the D as follows:

$$\begin{aligned} \min_{\gamma} \mathcal{L}_D(\omega, \theta, \gamma) \\ = -(\mathbb{E}_x \log(D_\gamma(x)) + \mathbb{E}_z \log(1 - (D_\gamma(G_\theta(v_z)))) \\ + V_c - \lambda_p V_p \end{aligned} \quad (9)$$

where $V_c = \mathbb{E}_x \log(D_\gamma(G_\theta(E_\omega(x))))$, and it means that the reconstruction samples are also considered as the real HSI samples. In contrast, if we consider the reconstruction samples as fake HSI samples, it will accelerate the convergence of the discriminator. This easily leads to the gradient saturation of the discriminator D_γ . On the other hand, we apply the gradient penalty term $V_p = \mathbb{E}_{\hat{x}} (\|\nabla_{\hat{x}} D_\gamma(\hat{x})\|_2^2 - 1)^2$ for the discriminator, where λ_p is a constant, and $\hat{x} = \mu x + (1-\mu)G(v_z)$, where μ is a uniform random number $\mu \in U[0, 1]$. V_p enforces sufficient gradients to the discriminator, which originates from the Lipschitz constraint of Wasserstein-1 distance [46]. This also helps to stabilize the proposed model. We will continue to discuss the effect of different distance constraints on the HSI classification in Section V-G.

The classifier C categorizes the real, reconstructed, and fake HSI samples, which also plays an important role in the DGAN framework. We use the cross-entropy loss to evaluate the differences between the network output and the original labels. The objective function of C can be written as

$$\begin{aligned} \min_{\zeta} \mathcal{L}_C(\omega, \theta, \zeta) = -(\mathbb{E}_x \log(C_\zeta(x)) + \mathbb{E}_z \log(C_\zeta(G_\theta(v_z)))) \\ + \mathbb{E}_x \log(C_\zeta(G_\theta(E_\omega(x)))) \end{aligned} \quad (10)$$

In this work, via (6) and (8)–(10), we build a stable DGAN capable of generating high-quality samples for the few labeled and imbalanced HSI data.

D. Implementation Details of the Proposed DGAN

The proposed DGAN contains four modules: encoder E , generator G , discriminator D , and classifier C . All modules are constructed by the CNN of four convolution layers. The details are shown in Table I. We first divide the original HSI data into HSI cubes of size $w \times w \times n$, where w is the spatial window size and n represents the number of spectral bands after PCA. The size of input noise is $100 \times 1 \times 1$. It should be noted that our proposed DGAN can balance the training

TABLE I
STRUCTURE OF THE PROPOSED DGAN

Nets	No.	Type	Kernel size	BN	Stride	Activation function
E	1	Conv	5x5x256	no	2	LeakyRelu
	2	Conv	5x5x512	yes	2	LeakyRelu
	3	Conv	5x5x1024	yes	2	LeakyRelu
	4	Conv	5x5x2048	yes	2	LeakyRelu
	5	FC	-	-	-	-
G	1	FC	-	yes	-	Relu
	2	Reshape	4x4x1024	-	-	-
	3	TConv	5x5x512	yes	2	Relu
	4	TConv	5x5x256	yes	2	Relu
	5	TConv	5x5x128	yes	2	Relu
	6	TConv	5x5xn	no	2	Linear
D	1	Conv	5x5x256	no	2	LeakyRelu
	2	Conv	5x5x512	yes	2	LeakyRelu
	3	Conv	5x5x1024	yes	2	LeakyRelu
	4	Conv	5x5x2048	yes	2	LeakyRelu
	5	FC	-	-	-	Sigmoid
C	1	Conv	5x5x256	no	2	LeakyRelu
	2	Conv	5x5x512	yes	2	LeakyRelu
	3	Conv	5x5x1024	yes	2	LeakyRelu
	4	Conv	5x5x2048	yes	2	LeakyRelu
	5	FC	-	n_classes	-	Softmax

samples by controlling the number of input noises. In the discriminator D and classifier C , the only difference between them is the last activation function layer. The discriminator D uses a sigmoid activation function for binary classification, and the classifier C uses a softmax activation function for multiclassification. We also use the batch normalization to improve the training speed and classification accuracy. The algorithm of DGAN is presented in Algorithm 1. It has four parameters: γ , ω , θ , and ζ . The input sample x is first normalized. We first train the encoder E and generator G to minimize the reconstruction loss by (6), updating the parameters ω and θ . Second, we train the discriminator D according to (9), updating the parameter γ . Then, we retrain the generator according to (8), updating the parameter θ . Finally, we train the classifier according to (10), updating the parameter ζ .

IV. EXPERIMENTAL RESULTS

A. Datasets

In the following, three widely used hyperspectral datasets are adopted to evaluate the classification performance of our proposed DGAN.

- 1) The Indian Pines dataset was gathered by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [47] sensor over the IP test area in Northwestern Indiana. It is 145×145 in size and contains 224 spectral bands in the wavelength range of $0.4\text{--}2.5 \mu\text{m}$, in which 24 invalid and damaged bands have been removed. The false-color image and the ground truth are shown in Fig. 4, which contains 16 mutually exclusive vegetation categories.
- 2) The Pavia University dataset was captured at the Pavia University in Italy by the Reflective Optics System Imaging Spectrometer (ROSIS) sensor [48]. The size of

Algorithm 1 DGAN

Input: the input samples x , noise z , labels c
Output: updated parameters $D_\gamma, E_\omega, G_\theta, C_\zeta$ of DGAN

- 1: Normalize the input x :
$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (11)$$
- 2: Initialize discriminator, encoder, generator and classifier $D_\gamma, E_\omega, G_\theta, C_\zeta$
- 3: **for** every epoch **do**
- 4: $x'^m \leftarrow$ Random minibatch of m data points from x'
- 5: $v_z^m \leftarrow$ Random m samples from noise distribution P_z with labels c
- 6: // Training encoder and generator using x'^m and v_z^m by Eq. (6)
- 7: $\omega, \theta \leftarrow \min_{\omega, \theta} L_R(\omega, \theta) + \lambda_L L_W(\omega, \theta)$
- 8: // Training discriminator on x'^m and v_z^m by Eq. (9)
- 9: $\gamma \leftarrow \min_\gamma \mathcal{L}_D(\omega, \theta, \gamma)$
- 10: // Training generator on x'^m and v_z^m by Eq. (8)
- 11: $\theta \leftarrow \min_\theta \mathcal{L}_G(\theta)$
- 12: // Training classifier on x'^m and v_z^m by Eq. (10)
- 13: $\zeta \leftarrow \min_\zeta \mathcal{L}_C(\omega, \theta, \zeta)$
- 14: **end for**

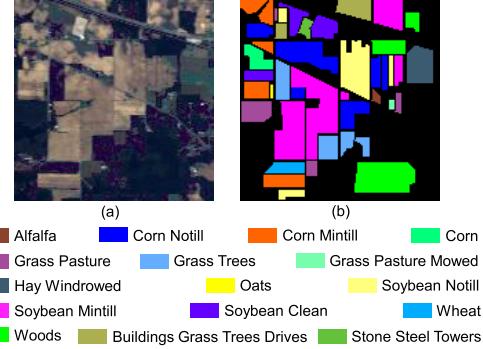


Fig. 4. Indian Pines dataset. (a) False-color composite image. (b) Ground-truth map.

the image is 610×340 and contains 103 spectral bands in the wavelength range of 430–860 nm. The false-color image and the ground truth are shown in Fig. 5, which contains nine representative urban categories.

- 3) The Botswana dataset was acquired by the Hyperion sensor on EO-1 over the Okavango, Delta, BW. The size of the data is 1476×256 and contains 242 spectral bands. After removing the uncalibrated and water-contaminated bands, there are 145 bands left. The false-color image and the ground truth are shown in Fig. 6. It contains 14 land-cover classes from seasonal swamps, occasional swamps, and drier woodlands located in the distal portion of the Delta.

B. Experimental Settings

To validate the performance of the proposed DGAN, we compare with the one-dimensional CNN (1DCNN) [49], 2DCNN [8], 3DCNN [50], three-dimensional GAN

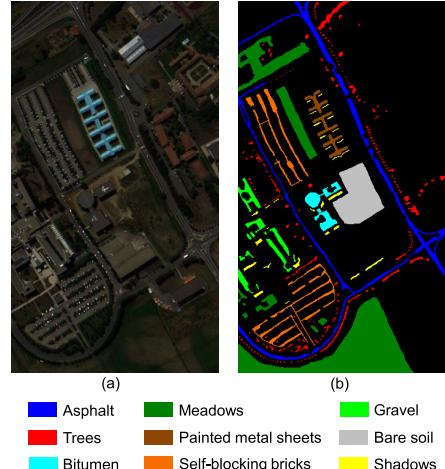


Fig. 5. Pavia University dataset. (a) False-color composite image. (b) Ground-truth map.

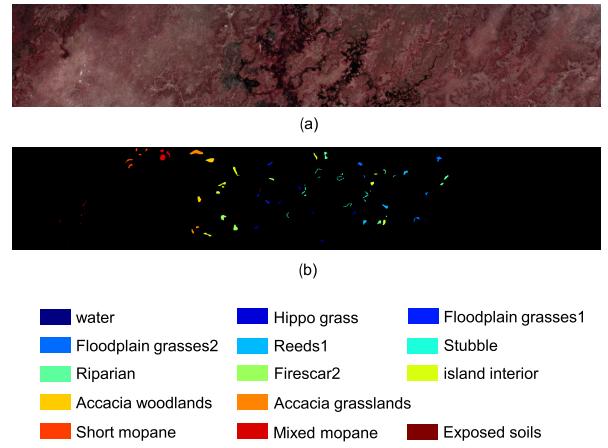


Fig. 6. Botswana dataset. (a) False-color composite image. (b) Ground reference map.

(3DGAN) [12], ADGAN [14], patchwise Spectralformer (Spectralformer) [51], and 3D-HyperGAMO [15]. To ensure the fairness of the experimental results, we use the same training sets for all methods. All experimental results are obtained by randomly splitting the training and testing sets. The details are shown in Tables II–IV. We randomly select 307 training samples on the Indian Pines dataset, 1000 training samples on the Pavia University, and 129 training samples on the Botswana dataset for training. A very small percentage of the training set is applied to simulate the scene with a few labeled samples. For the categories with only one sample, we carry out the resampling and other data enhancement operations, which is the same as in [38]. The results run more than 15 times independently. In the data preparation, all methods use the first three principal components by PCA as inputs. All experiments are implemented based on the TensorFlow or Pytorch library, and the NVIDIA TITAN Xp graphics card is used to implement GPU computing. The overall accuracy (OA) and average accuracy (AA) reflect the proportion of all test samples correctly classified and the balance of correct classification of each category, respectively. The kappa coefficient (K) is obtained based on the confusion

TABLE II

NUMBER OF THE TRAINING AND TESTING PIXELS OF EACH CLASS IN THE INDIAN PINE DATASET

No.	Class	Training	Testing
1	Alfalfa	1	45
2	Corn Notill	43	1385
3	Corn Mintill	25	805
4	Corn	7	230
5	Grass Pasture	14	469
6	Grass Trees	22	708
7	Grass Pasture Mowed	1	27
8	Hay Windrowed	14	464
9	Oats	1	19
10	Soybean Notill	29	943
11	Soybean Mintill	73	2382
12	Soybean Clean	18	575
13	Wheat	6	199
14	Woods	38	1227
15	Buildings Grass Trees Drives	12	374
16	Stone Steel Towers	3	90
Total		307	9942

TABLE III

NUMBER OF THE TRAINING AND TESTING PIXELS OF EACH CLASS IN THE PAVIA UNIVERSITY DATASET

No.	Class	Training	Testing
1	Asphalt	155	6476
2	Meadows	436	18213
3	Gravel	49	2050
4	Trees	72	2992
5	Painted metal sheets	31	1314
6	Bare soil	118	4911
7	Bitumen	31	1299
8	Self-blocking bricks	86	3596
9	Shadows	22	925
Total		1000	41776

matrix for the consistency test, an indicator that can reflect the “bias” of the model. The F1 score (F1) considers the precision and recall of the model. We use these metrics to characterize the classification accuracy and also provide the classification accuracy for each class.

For ADGAN, Spectralformer, and 3D-HyperGAMO, the provided source codes are used, with the default parameter mentioned in [14], [51], and [15]. For the 3DGAN, a neighborhood with a size of 64×64 is used, and the input images are normalized into the range $[-0.5, 0.5]$. The experimental setting of other comparison methods is according to the number of principal components based on [8]. For our proposed DGAN, the parameters of the model are optimized with the Adam optimizer, using a learning rate of 0.0002 for the three datasets. The number of the epoch is set as 700. The spatial window size is 64×64 , which is the same as the 3DGAN [12]. The minibatch size for training is set as 128 [16]. Moreover, the hyperparameters $\lambda_r = 1.0$ and $\lambda_p = 1.0$ [16]. We will discuss the two hyperparameters in Section V-F.

C. Classifications Results

Tables V–VII record the average classification accuracy and the corresponding standard deviation of the compared

TABLE IV

NUMBER OF THE TRAINING AND TESTING PIXELS OF EACH CLASS IN THE BOTSWANA DATASET

No.	Class	Training	Testing
1	water	11	259
2	Hippo grass	4	97
3	Floodplain grasses1	10	241
4	Floodplain grasses2	8	207
5	Reeds1	11	258
6	Stubble	11	258
7	Riparian	10	249
8	Firescar2	8	195
9	island interior	12	302
10	Acacia woodlands	10	238
11	Accacia grasslands	12	293
12	Short mopane	7	174
13	Mixed mopane	11	257
14	Exposed soils	4	91
Total		129	3119

algorithms and the proposed DGAN. The first several rows of the table represent the results of each category, and the last four rows are the results of the OA, AA, kappa coefficients, and the F1 score. The results in black bold font work best. The classification maps of the compared algorithms are also shown to verify the validity of our method.

1) *Results of the Indian Pines Dataset:* The number of training and testing samples is shown in Table II. The statistical classification results on the Indian Pines dataset are summarized in Table V. As shown in Table V, our proposed DGAN achieves the highest accuracies on 9 of the 16 classes compared to other methods. The 2DCNN, 3DCNN, 3DGAN, 3D-HyperGAMO, and other comparison methods outperform the 1DCNN in classification performance due to the introduction of spatial information. The proposed DGAN has the best performance for classes with few training samples such as Alfalfa, Corn, and Grass Pasture Mowed. The results show that DGAN can learn the essential characteristics of these samples and then generate high-quality HSI samples even for the classes with few labeled samples. The OA and kappa of 3DGAN are similar to 3D-HyperGAMO, but the AA and F1 of 3DGAN are lower than those of 3D-HyperGAMO. It shows that the ability of the basic GAN to classify the extremely imbalanced HSI samples is still limited. The GAN-based methods, ADGAN, 3D-HyperGAMO, and our proposed method, show their superiority with high OA and F1 on this dataset. Due to the focus on category information, the 3D-HyperGAMO has a higher AA than other methods. The Spectralformer does not have an advantage with this highly unbalanced and limited dataset. Importantly, the proposed DGAN has been greatly improved and far exceeds the 3D-HyperGAMO. The proposed method exhibits the best OA and kappa coefficients over the ADGAN by 1.95% and 0.92% and the best AA and F1 over 3D-HyperGAMO by 4.09% and 0.0538, respectively, which shows that generating the high-quality HSI samples is more efficient than oversampling duplicate samples.

As shown in Fig. 7, the map of the 1DCNN has lots of visually noisy scattered points. Many pixels are misclassified in the 2DCNN, 3DCNN, 3DGAN, and Spectralformer,

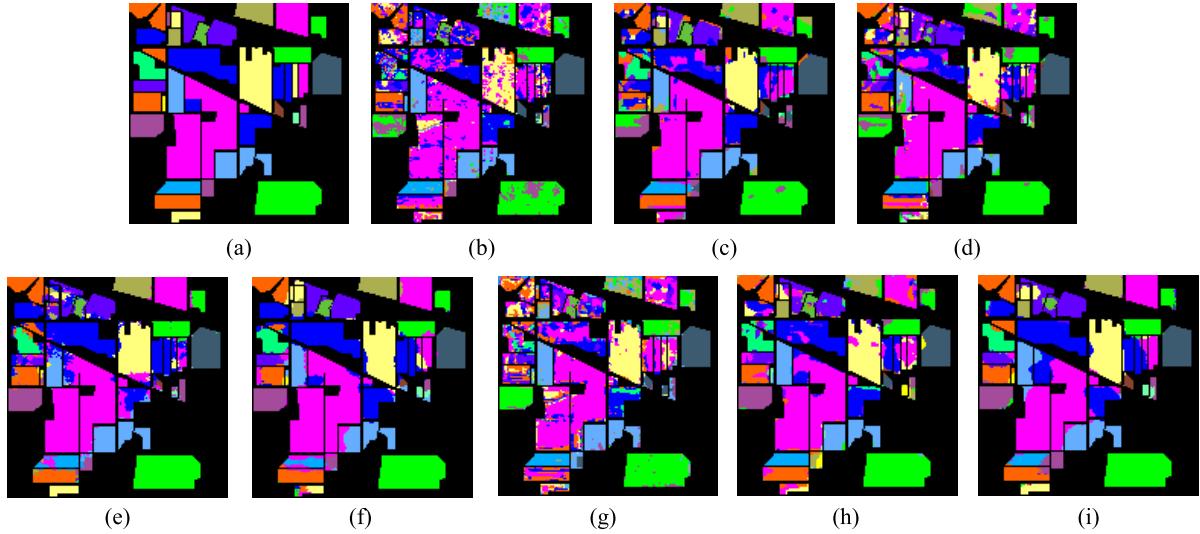


Fig. 7. Indian pines dataset: (a) ground truth. Classification maps obtained by (b) 1DCNN, (c) 2DCNN, (d) 3DCNN, (e) 3DGAN, (f) ADGAN, (g) Spectralformer, (h) 3D-HyperGAMO, and (i) proposed.

TABLE V
CLASSIFICATION ACCURACY (%) OF THE INDIAN PINES IMAGE WITH DIFFERENT METHODS

No.	1DCNN	2DCNN	3DCNN	3DGAN	ADGAN	Spectralformer	3D-HyperGAMO	Proposed
1	5.43±10.7	22.22±9.36	20.49±7.89	32.32±16.4	49.65±33.4	1.11±2.13	40.16±21.6	63.41±17.0
2	36.05±3.46	69.45±4.67	59.03±6.66	79.57±3.97	87.30±5.43	45.60±3.54	84.80±6.88	89.38±4.75
3	38.81±6.03	73.91±8.55	58.72±7.79	86.49±6.24	87.96±5.88	39.95±4.50	77.85±4.80	92.55±4.28
4	29.47±8.66	43.53±8.53	34.44±5.18	84.03±10.8	84.88±8.57	14.96±4.25	74.81±10.4	85.07±11.1
5	36.22±20.2	68.66±11.7	62.24±9.70	84.34±6.53	87.98±9.23	27.49±10.2	85.49±3.07	89.60±5.38
6	87.96±5.29	90.73±4.27	84.15±11.9	89.84±1.29	93.39±2.96	86.09±5.91	97.20±2.36	93.17±2.81
7	12.76±8.75	47.69±20.0	19.75±7.81	51.52±27.0	31.48±43.0	4.63±5.04	68.52±21.7	76.30±19.4
8	98.75±1.63	93.56±4.38	97.24±3.36	96.53±1.35	97.28±4.22	95.83±2.27	98.08±3.48	99.11±1.03
9	2.34±4.38	43.42±14.6	27.48±13.1	50.24±15.1	27.63±38.0	15.34±15.0	65.79±28.0	63.86±24.4
10	61.21±8.87	79.24±1.85	63.00±13.9	85.32±5.90	93.37±3.53	65.67±4.48	81.50±5.35	90.48±3.84
11	73.47±8.85	86.07±3.54	80.95±7.32	93.21±2.46	95.52±2.74	69.62±3.38	88.77±3.11	95.43±2.55
12	24.70±9.52	62.57±7.28	48.44±9.26	69.47±10.8	86.18±5.88	30.97±8.04	71.44±6.45	82.69±5.02
13	92.13±7.75	86.18±10.4	80.91±9.68	82.64±6.64	84.44±19.8	85.22±8.72	98.64±2.94	88.27±9.26
14	91.86±5.87	95.31±2.06	83.60±11.7	95.38±2.76	97.38±1.89	88.96±4.67	96.00±1.89	97.61±2.79
15	17.59±4.21	82.59±5.06	58.73±7.88	82.36±8.60	87.80±7.90	36.92±9.04	79.43±5.98	91.93±6.31
16	86.42±4.85	77.50±11.4	47.41±18.1	63.53±15.8	74.27±23.7	87.87±10.9	95.95±4.30	71.04±16.5
OA(%)	60.95±1.61	80.10±1.34	69.67±3.80	86.84±2.00	90.21±1.33	61.44±1.31	86.68±1.53	92.16±1.06
AA(%)	49.70±1.88	70.16±2.28	57.23±3.16	76.67±3.24	79.16±4.14	49.76±1.38	81.53±3.46	85.62±2.43
K×100	54.97±1.57	77.25±1.51	65.07±4.36	84.97±2.26	90.13±1.21	55.77±1.48	84.80±1.74	91.05±1.20
F1×100	64.15±1.20	81.27±1.29	72.96±2.36	85.04±1.66	91.26±0.79	60.10±1.29	86.93±1.45	92.31±0.74

especially the Alfalfa, Corn Mintill, Corn, Soybean Clean, and Buildings Grass Trees Drives classes. Compared with them, the ADGAN, 3D-HyperGAMO, and our DGAN distinguish better in these classes. Due to generating HSI samples for each class, the proposed DGAN performs better than the two comparison methods in the minority classes, such as the Alfalfa and Corn classes.

2) *Results of the Pavia University Dataset:* The numbers of training and testing samples for each class are listed in Table III. The statistical classification results of the Pavia University dataset are shown in Table VI. The first nine rows of Table VI represent the results of each class. The spectral-spatial methods are better than the spectral method 1DCNN. Also, the proposed DGAN performs the best. Specifically, for the minority class 7 (Bitumen), the accuracies of other methods except the 3DGAN are all lower than 89%, while the DGAN achieves more than 94% accuracy. The OA of the proposed DGAN is higher than that of the

3D-HyperGAMO and ADGAN by about 8.79% and 1.96%, respectively. The AA of the proposed DGAN is higher than that of the 3D-HyperGAMO by about 4.37%. The kappa and F1 are 96.47% and 0.9723, and 11.43% and 0.0797 higher than the 3D-HyperGAMO. Overall, the DGAN has achieved a better performance.

As shown in Fig. 8, the 2DCNN, 3DCNN, 3DGAN, and 3D-HyperGAMO generate similar classification maps. It is obvious that the bare soil category is easily incorrectly classified as the meadows category with similar spectra. In contrast, our proposed method can better distinguish these two confusing classes. Second, for class 1 (Asphalt), the DGAN is also less affected by the adjacent gravel and tree classes.

3) *Results of the Botswana Dataset:* The numbers of training and testing samples for each class are listed in Table IV. The classification results of the Botswana dataset are summarized in Table VII. The first 14 rows of Table VII represent the classification results of each category. Most of the comparison

TABLE VI
CLASSIFICATION ACCURACY (%) OF THE PAVIA UNIVERSITY IMAGE WITH DIFFERENT METHODS

No.	1DCNN	2DCNN	3DCNN	3DGAN	ADGAN	Spectralformer	3D-HyperGAMO	Proposed
1	85.65±1.98	95.53±2.06	96.10±0.84	93.98±0.51	92.36±1.33	87.40±2.84	92.87±2.89	96.42±1.98
2	93.70±1.46	98.33±1.16	98.75±0.48	99.44±0.08	96.18±0.45	96.71±1.65	87.04±3.31	99.62±0.20
3	58.77±5.14	82.15±4.63	83.87±3.15	78.16±3.18	91.49±1.45	66.54±6.07	81.86±7.97	92.61±4.35
4	79.99±5.53	97.09±1.13	97.58±0.72	84.91±1.22	97.29±0.25	92.61±2.54	96.76±1.22	92.86±2.96
5	98.55±1.03	97.65±5.79	99.95±0.06	99.27±0.33	98.54±0.16	100.00±0.00	99.81±0.19	97.17±4.41
6	43.15±2.88	84.69±11.7	86.72±4.34	96.81±0.67	96.37±0.27	29.63±8.53	80.46±4.16	99.55±0.37
7	67.90±5.41	85.41±4.68	82.47±3.17	92.28±2.50	88.40±1.54	42.46±20.0	88.99±3.32	94.68±3.83
8	79.18±4.08	95.53±2.93	94.24±0.91	92.47±1.59	95.75±1.25	86.13±3.0	90.06±7.48	94.71±1.91
9	99.53±0.33	90.77±11.2	97.10±2.01	80.03±4.04	96.02±0.26	95.96±3.18	96.65±1.09	86.18±5.01
OA(%)	82.04±0.35	94.58±1.76	95.22±0.44	94.94±0.32	95.38±0.54	83.10±0.86	88.55±1.53	97.34±0.46
AA(%)	78.49±0.67	91.91±1.98	92.98±0.83	90.82±0.85	94.71±0.35	77.49±2.92	90.50±1.29	94.87±0.57
K×100	75.66±0.45	92.77±2.41	93.62±0.61	93.26±0.43	95.21±0.41	76.85±1.31	85.04±1.93	96.47±0.61
F1×100	83.52±0.58	95.24±0.53	96.06±0.49	93.92±0.76	96.87±0.73	83.18±0.59	89.26±0.96	97.23±0.32

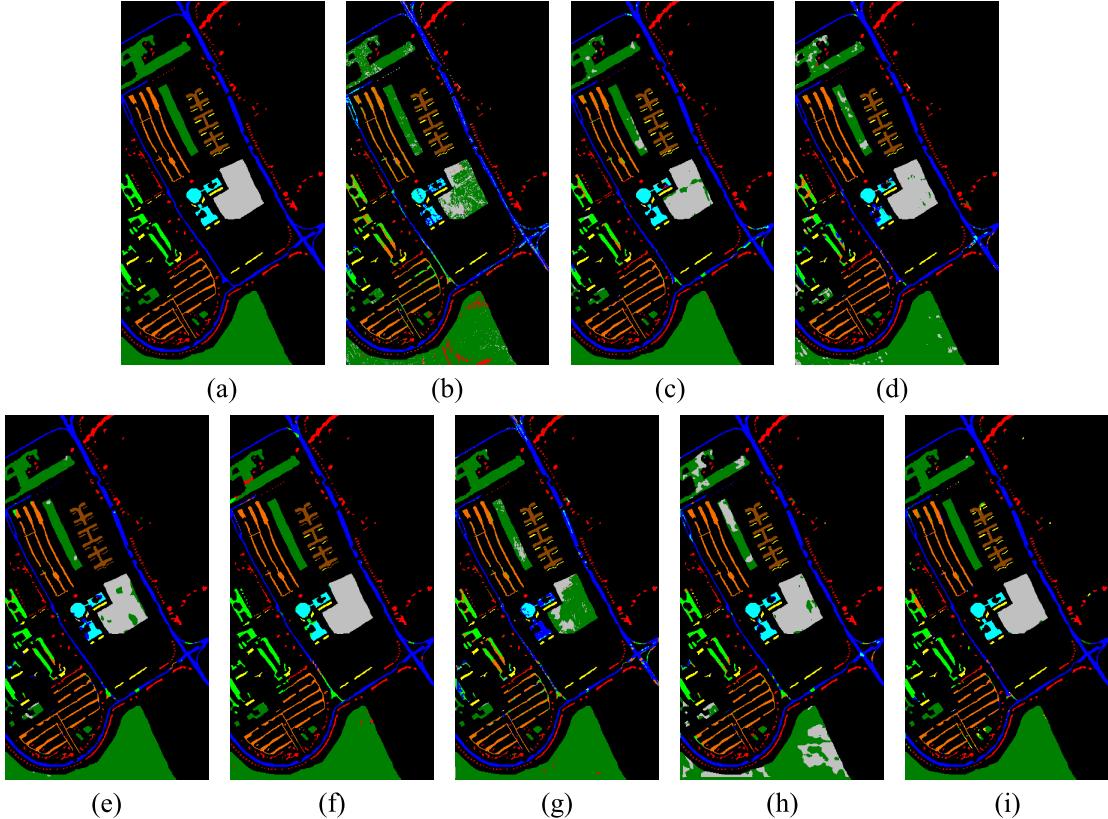


Fig. 8. Pavia University dataset: (a) ground truth. Classification maps obtained by (b) 1DCNN, (c) 2DCNN, (d) 3DCNN, (e) 3DGAN, (f) ADGAN, (g) Spectralformer, (h) 3D-HyperGAMO, and (i) proposed.

methods achieve more than 85% in terms of OA for the Botswana dataset due to the simpler scene compared to the other two datasets. In this case, the DGAN still outperforms the other methods. Specifically, the proposed DGAN achieves the highest OA (95.18%), AA (94.57%), kappa (94.78%), and F1 (0.9576), which exceeds the second best method 3D-HyperGAMO by 0.91%, 0.08%, 0.99%, and 0.013, respectively. The excellent performance of the classification results on three datasets demonstrates the advantage of our proposed method.

As shown in Fig. 9, the classification maps of the 1DCNN and 3DGAN contain a very large number of scatter points. Compared to them, the 2DCNN, 3DCNN, ADGAN,

Spectralformer, 3D-HyperGAMO, and DGAN generate better classification maps. Besides, the proposed method is closer to the ground-truth graph. Because our DGAN can generate more high-quality samples for the insufficient and imbalanced HSI samples by learning the essential characteristics of the real data. In the three datasets, the proposed DGAN shows excellent classification performance.

D. Impact of the Number of Labeled Examples

In this section, in order to evaluate the advantage and generalization ability of our proposed DGAN, we train the compared methods and our DGAN under different training

TABLE VII
CLASSIFICATION ACCURACY (%) OF THE BOTSWANA IMAGE WITH DIFFERENT METHODS

No.	1DCNN	2DCNN	3DCNN	3DGAN	ADGAN	Spectralformer	3D-HyperGAMO	Proposed
1	99.57±0.81	96.78±3.15	99.96±0.12	95.02±1.59	88.86±8.42	100.00±0.00	98.01±2.73	96.00±4.07
2	92.01±4.99	69.96±37.7	84.76±14.3	74.60±12.3	81.75±15.8	78.81±16.8	99.68±0.85	87.07±11.8
3	90.46±3.87	98.62±1.31	94.65±3.57	83.74±6.86	87.79±9.00	94.75±5.55	96.01±3.14	93.55±4.56
4	88.11±4.95	81.11±8.10	97.21±2.57	71.45±10.0	88.94±9.00	94.15±8.95	94.17±5.26	96.00±3.69
5	72.92±5.42	82.60±9.64	86.65±5.69	78.54±7.64	88.68±5.65	82.69±4.69	86.40±7.56	86.08±7.10
6	54.22±12.1	86.56±11.1	81.91±15.0	82.38±6.02	88.29±7.04	79.33±8.77	85.54±7.94	96.05±3.00
7	95.13±1.91	98.97±1.26	98.75±2.41	91.46±1.44	94.08±6.42	99.42±1.27	99.26±2.57	96.61±3.38
8	81.60±5.01	86.32±10.7	95.21±8.83	84.43±6.49	87.53±13.1	90.14±10.5	90.33±9.92	93.15±6.18
9	75.95±6.47	97.61±2.78	98.35±2.69	87.78±4.49	98.63±2.56	91.13±2.38	90.78±8.44	97.62±2.37
10	71.69±6.80	98.97±2.90	92.72±13.1	99.73±0.63	99.87±0.38	97.20±4.53	98.38±1.84	100.00±0.00
11	91.13±3.31	98.03±3.08	96.93±6.26	93.67±6.27	95.29±4.39	97.34±2.07	96.69±2.36	96.74±4.62
12	88.43±6.16	91.70±6.37	89.08±8.78	65.67±9.89	82.51±10.8	96.11±2.16	96.69±2.75	96.76±5.63
13	73.35±5.81	93.26±5.10	98.40±1.45	81.18±9.86	89.49±11.6	97.88±2.03	98.17±2.76	98.37±1.71
14	81.46±19.8	65.81±33.0	79.73±22.9	81.02±17.3	90.05±6.07	85.96±6.48	92.73±7.10	89.91±7.95
OA(%)	81.92±1.13	91.41±1.14	93.59±2.07	84.94±2.81	89.30±2.11	92.62±1.47	94.27±1.67	95.18±1.13
AA(%)	82.57±1.45	88.95±2.33	92.45±2.62	83.62±2.77	90.12±2.54	91.78±2.09	94.49±1.61	94.57±1.24
K×100	80.41±1.23	90.69±1.23	93.05±2.24	83.68±3.05	90.20±2.20	92.00±1.60	93.79±1.81	94.78±1.22
F1×100	81.62±0.81	91.90±2.86	94.01±0.76	85.34±2.77	90.75±2.35	92.45±1.37	94.46±1.25	95.76±0.83

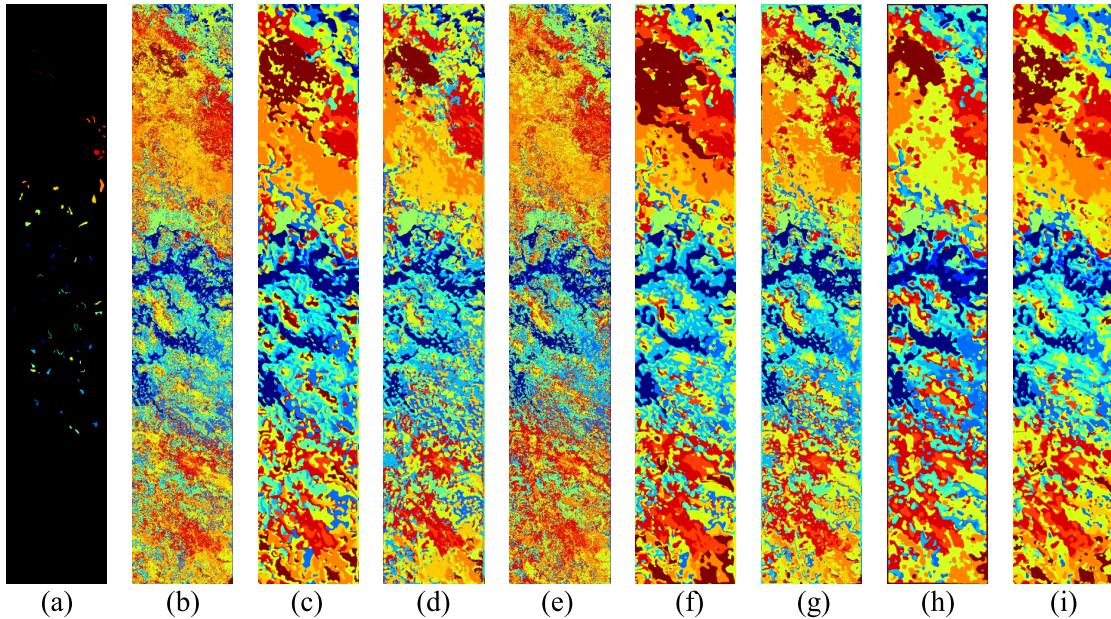


Fig. 9. Botswana dataset: (a) ground truth. Classification maps obtained by (b) 1DCNN, (c) 2DCNN, (d) 3DCNN, (e) 3DGAN, (f) ADGAN, (g) Spectralformer, (h) 3D-HyperGAMO, and (i) proposed.

set sizes. Fig. 10 shows the classification performance in terms of OA under the varying training sizes. As can be seen from Fig. 10, the performance of the 1DCNN, which only utilizes the spectral information, is usually lower than the 2DCNN, 3DCNN, 3DGAN, ADGAN, Spectralformer, 3D-HyperGAMO, and the proposed DGAN. This reveals the importance of spatial information in the HSI classification task.

Fig. 10(a) shows the OA performance under the Indian Pines dataset. Using the 3% training set, the 3DGAN and 3D-HyperGAMO show resembling performance. When the size of the training set increases to 10%, the 3D-HyperGAMO outperforms the 3DGAN. Fig. 10(b) shows the OA performance on the Pavia University dataset. Under different sizes of training percentages, the OA of our DGAN is the highest. With the increase of the training set ratio, the 3DCNN gradually

obtains equivalent performance to the ADGAN. Fig. 10(c) shows the OA for the Botswana dataset. Under 2% training size, the Spectralformer, 3D-HyperGAMO, and our DGAN behave similarly. When the training set increases to 4% and 6%, the performance improvement of the Spectralformer is not as good as that of the 3D-HyperGAMO and the DGAN. This experiment shows that the proposed DGAN consistently performs better than other methods under different training set sizes and reveals the advantage of the proposed method.

E. Parameter Analysis

In this experiment, we analyze the influence of the important parameters λ_r and λ_p in terms of OA. λ_r is related to the latent-data distance constraint in AE, and λ_p is associated with the gradient penalty term to the discriminator. To consider

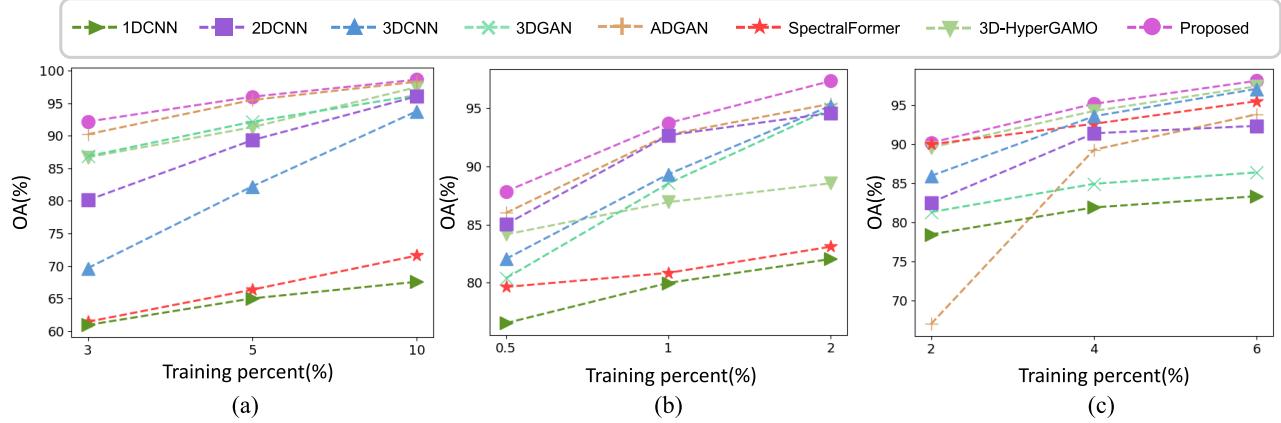


Fig. 10. OA results of 1DCNN, 2DCNN, 3DCNN, 3DGAN, ADGAN, Spectralformer, 3D-HyperGAMO, and the proposed DGAN with different training sizes on (a) Indian Pines dataset, (b) Pavia University, and (c) Botswana dataset.

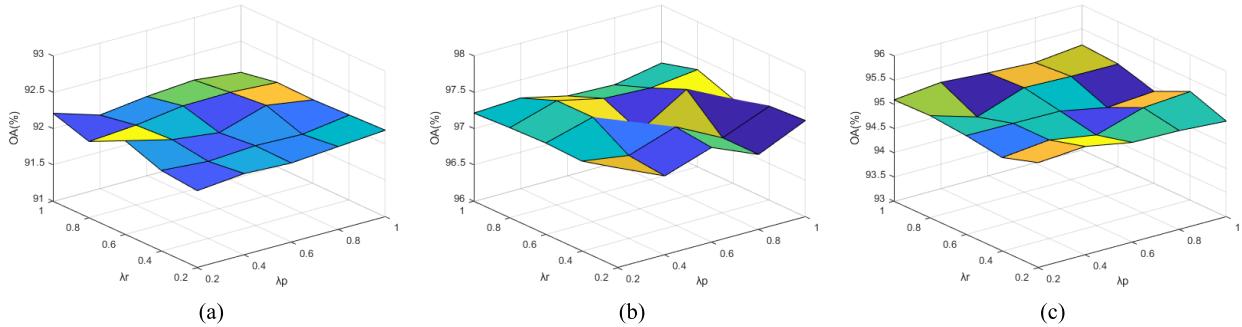


Fig. 11. Parameter analysis with different λ_r and λ_p values on (a) Indian Pines dataset, (b) Pavia University, and (c) Botswana dataset.

TABLE VIII

OA, AA (%), AND KAPPA COEFFICIENT OF THE INDIAN PINES IMAGE WITH DIFFERENT MODEL SETTINGS

Methods	DGAN-04	DGAN-03-fake	DGAN-03-real	DGAN-02	DGAN-01	DGAN
OA	90.78±0.71	87.25±0.91	90.91±0.44	90.80±1.25	91.00±0.82	92.16±1.06
AA	80.95±2.45	77.57±1.99	83.50±3.42	84.91±3.31	84.58±2.65	85.62±2.43
Kappa	89.45±0.82	85.46±0.99	89.63±0.49	90.39±1.01	89.71±0.94	91.05±1.20

TABLE IX

OA, AA (%), AND KAPPA COEFFICIENT OF THE PAVIA UNIVERSITY IMAGE WITH DIFFERENT MODEL SETTINGS

Methods	DGAN-04	DGAN-03-fake	DGAN-03-real	DGAN-02	DGAN-01	DGAN
OA	95.03±0.43	93.16±0.39	95.87±0.26	96.35±0.18	96.29±0.40	97.34±0.46
AA	91.48±0.71	86.70±1.19	92.27±1.52	93.71±0.48	93.89±1.02	94.87±0.57

common effects, we do not fix either parameter. λ_r and λ_p range between [0.2, 1.0]. Fig. 11(a)–(c) shows the classification results obtained by our DGAN under varying the λ_r and λ_p values. As can be seen from Fig. 11, we obtain stable OA performance under the three datasets. The OA is higher than 90% for the Indian Pines and Botswana datasets and higher than 97% for the Pavia University dataset. It indicates that the classification performance is not very sensitive to these two hyperparameters. In our method, we set $\lambda_r = 1.0$ and $\lambda_p = 1.0$ for the three hyperspectral datasets. In addition, the value of the parameter λ_w is determined by the latent sample dimension and the data sample dimension.

F. Ablation Study

Tables VIII–X show the classification performance of our ablation experiments under the three datasets. DGAN-04 trains the AE, the generator, and the discriminator only by conventional objective functions. DGAN-03 denotes the discriminator-score distance constraint for generator objective \mathcal{L}_G and the traditional objective function L_R for the

TABLE X

OA, AA (%), AND KAPPA COEFFICIENT OF THE BOTSWANA IMAGE WITH DIFFERENT MODEL SETTINGS

Methods	DGAN-04	DGAN-03-fake	DGAN-03-real	DGAN-02	DGAN-01	DGAN
OA	91.02±3.06	88.65±1.99	93.13±1.09	92.85±1.62	93.39±1.85	95.18±1.13
AA	90.68±3.46	88.13±2.04	92.17±1.11	93.01 ±1.23	93.56±1.45	94.57±1.24
Kappa	90.27±3.32	87.70±2.16	92.55±1.18	92.93±1.67	92.84±2.01	94.78±1.22

AE, without using the latent-data distance constraint L_W . DGAN-03-real uses the reconstructed HSI samples V_c as real samples in the discriminator objective function based on DGAN-03. Similarly, DGAN-03-fake uses the reconstructed HSI samples V_c as fake samples. DGAN-02 trains the AE by the latent-data distance constraint L_W based on DGAN-04. DGAN-01 adds the gradient penalty V_P in the discriminator objective function based on DGAN-03-real. DGAN improves from DGAN-01 by adding the latent-data distance constraint L_W . In Tables VIII–X, compared with DGAN-04, the performance improvement of DGAN-03-real is mainly manifested in AA, which reflects the effectiveness of the discriminator-score distance constraint. In addition, DGAN-03-real performs much

TABLE XI

PERFORMANCE OF DIFFERENT DIMENSIONALITY REDUCTION METHODS ON THE PROPOSED METHOD FOR THE INDIAN PINES DATASET

Methods	t-SNE	ICA	PCA
OA	89.99±0.56	90.79±1.80	92.16±1.06
AA	82.50±2.29	85.03±1.12	85.62±2.43
Kappa	88.58±0.65	89.47±2.05	91.05±1.20

TABLE XII

PERFORMANCE OF DIFFERENT DIMENSIONALITY REDUCTION METHODS ON THE PROPOSED METHOD FOR THE PAVIA UNIVERSITY DATASET

Methods	t-SNE	ICA	PCA
OA	92.27±0.69	97.03±0.52	97.34±0.46
AA	86.55±1.61	93.94±1.03	94.87±0.57
Kappa	88.82±0.91	96.06±0.70	96.47±0.61

TABLE XIII

PERFORMANCE OF DIFFERENT DIMENSIONALITY REDUCTION METHODS ON THE PROPOSED METHOD FOR THE BOTSWANA DATASET

Methods	t-SNE	ICA	PCA
OA	83.59±3.44	93.75±0.36	95.18±1.13
AA	82.00±3.31	92.10±1.50	94.57±1.24
Kappa	81.91±3.70	92.97±0.40	94.78±1.22

better than DGAN-03-fake, with a very great improvement in terms of the OA, AA, and Kappa compared to the other three ablation experiments. It shows that using reconstructed samples as real samples for the discriminator is better than fake HSI samples. DGAN-02 has varying degrees of improvement on all three datasets, especially AA and Kappa. The performance of DGAN-01 with gradient penalty added to the discriminator is further improved compared to DGAN-03. Compared with DGAN-01, the proposed DGAN improves the AA in the range of 1.05%–1.79% on the three datasets. Based on the above analysis, distance constraints, gradient penalty V_p , and the reconstructed samples V_c have different degrees of improvement in the performance of the model. Together, they promote our proposed DGAN to be stable and able to synthesize sufficient high-quality HSI samples under the few labeled training samples.

G. Performance Evaluation With Different Dimensionality Reduction Methods

In order to illustrate the influence of different dimensionality reduction methods on the proposed method in the data preparation part, we choose the commonly used linear dimension reduction methods independent component analysis (ICA) and nonlinear dimension reduction methods t-distributed stochastic neighbor embedding (t-SNE) for comparison with PCA in the proposed method. Among them, PCA achieves dimensionality reduction by calculating the covariance matrix of the data and finding the largest eigenvalues and the corresponding eigenvectors (principal components) [41]. ICA assumes that data are composed of a linear combination of several mutually independent signal sources and realizes signal separation by esti-

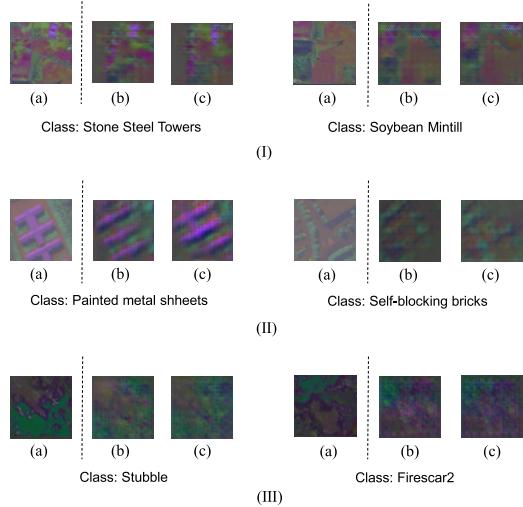


Fig. 12. Real samples and corresponding representative generated fake samples in different classes on three datasets. (I) Indian Pines dataset. (II) Pavia University dataset. (III) Botswana dataset. (a) Real samples. (b) First generated samples. (c) Second generated samples.

mating mixed matrix and source signals [52]. T-SNE achieves dimensionality reduction by mapping high-dimensional data to low-dimensional data by preserving the similarity relationship between data points in the high-dimensional space [53]. The experimental results are shown in Tables XI–XIII.

PCA achieves the best OA, AA, and Kappa. Linear dimension reduction methods ICA and PCA perform better than the nonlinear dimension reduction methods t-SNE. This phenomenon is more evident in the Botswana dataset shown in Table XIII. In fact, our proposed method only needs to retain the main information in the data preparation part without the complicated pre-processing step of feature extraction. Extracting features in the process of data preparation will lead to the information loss, which will affect the effect of subsequent feature learning. At the same time, PCA is computationally efficient and simple to implement, which does not require additional hyperparameter adjustment or involve complex matrix operations. Therefore, PCA is widely used in the current deep learning methods [12], [13], [14]. In order to compare different methods fairly, PCA is also chosen as the dimensionality reduction method in our proposed method.

H. Visualization of Adversarial Samples

In our DGAN model, the quality of the generated HSI samples can directly reflect the learning ability of the generator. Fig. 12 shows the visualization of the generated HSI samples under the Indian Pines dataset, Pavia University dataset, and the Botswana dataset. Each dataset shows the generated HSI samples of two classes that are representative categories under the dataset. As can be seen from Fig. 12, the proposed DGAN does a good job of learning many obvious features and edge details. It can also produce high-quality HSI samples for the majority and minority classes. In addition, the colors in these images are drawn by using the spectral segments as the three channels of the RGB images. Thus, in addition to the spatial information, the generator of DGAN can also learn the

TABLE XIV
RUNNING TIME OF DIFFERENT METHODS ON THE INDIAN PINES DATASET

Dataset	Method	Training Times(s)	Test Times(s)
Indian Pines	1DCNN	38.28	0.68
	2DCNN	74.27	0.51
	3DCNN	43.37	0.41
	3DGAN	283.51	2.25
	ADGAN	279.51	1.83
	Spectralformer	57.13	0.47
	3D-HyperGAMO	2049.32	0.46
	DGAN	376.65	2.37

TABLE XV
RUNNING TIME OF DIFFERENT METHODS ON THE PAVIA UNIVERSITY DATASET

Dataset	Method	Training Times(s)	Test Times(s)
Pavia University	1DCNN	218.56	1.48
	2DCNN	240.54	1.69
	3DCNN	157.12	2.85
	3DGAN	280.75	6.93
	ADGAN	383.13	6.95
	Spectralformer	419.14	2.02
	3D-HyperGAMO	1420.63	1.59
	DGAN	677.21	6.82

TABLE XVI
RUNNING TIME OF DIFFERENT METHODS ON THE BOTSWANA DATASET

Dataset	Method	Training Times(s)	Test Times(s)
Botswana	1DCNN	16.37	0.19
	2DCNN	26.37	0.18
	3DCNN	14.78	0.28
	3DGAN	161.86	1.35
	ADGAN	206.02	0.26
	Spectralformer	50.89	0.15
	3D-HyperGAMO	1736.23	0.20
	DGAN	1158.29	0.93

TABLE XVII
COMPARISON OF PARAMS AND FLOPS OF DIFFERENT METHODS

Method	Param(M)	FLOPs(G)
1DCNN	0.004	0
2DCNN	3.04	0.02
3DCNN	28.05	0.06
3DGAN	10.46	0.02
ADGAN	7.38	13.5
Spectralformer	1.10	0.18
3D-HyperGAMO	2.51	13.3
DGAN	72.5	0.25

spectral information of the real samples well. The excellent visualization results show that the generator of DGAN can learn the data distribution of the real HSI samples well. It also proves that the proposed DGAN still has the advantages in the face of limited and imbalanced training sets and then improves the performance of the HSI classification task.

I. Computation Time

Tables XIV–XVI record the computation time of the different classification methods on three datasets. In terms of training time, the GAN-related methods, such as 3DGAN, ADGAN, 3D-HyperGAMO, and DGAN, require more time to

converge due to the process of adversarial learning. In terms of test time, these methods also require more time due to the spatial information being processed with the larger patch size chosen. Thus, 3D-HyperGAMO has less time because of the smaller patch size. In general, although the computational time of the proposed DGAN ranks in the middle, it has the best classification performance.

Although there is no specific relationship between the number of parameters, floating-point operations (FLOPs), and the running speed, it can still measure the complexity of the model. Therefore, we use them as a reference to evaluate the model. We show the comparison results in Table XVII. The 1DCNN is simple with few trainable parameters, and its FLOPs are close to 0 in the unit GFLOPs. In addition, the FLOPs of the 2DCNN, 3DCNN, and 3DGAN are all low, while the number of parameters and FLOPs of the ADGAN and 3D-HyperGAMO are close. Although our model DGAN has the largest number of parameters, it has obvious advantages in computation and is far superior to other models in accuracy.

V. CONCLUSION

In this work, we propose a DGAN method for the HSI classification. The method aims to address the problem of limited and imbalanced labeled samples in the HSI classification task. In the method, the AE is first used to extract the latent features of the HSI. Then, the latent-data distance constraint and discriminator-score distance constraint are used to improve the generator to generate sufficient high-quality samples for all classes in HSI. Finally, the same number of generated HSI samples and the reconstructed samples as the batch size from the AE can be used to learn the parameters of a more powerful classifier. To verify the effectiveness of this method with limited labeled samples, we conduct extensive experiments on three hyperspectral datasets. The results are compared with the traditional, CNN-based, and GAN-based methods. The results show that the proposed DGAN can achieve the best performance under the three datasets. Also, in some extremely few number of training samples, the performance of the DGAN still has advantages.

Class imbalance is a common phenomenon in HSIs. The imbalance problem will cause the generated samples of the GANs to be biased toward classes with more samples, thus reducing the classification performance. In addition, such effects are further exacerbated by the low quality of the generated hyperspectral samples. To some extent, the proposed DGAN alleviates the above two problems by two distance constraints, and however, the proposed DGAN does not explicitly reduce the interclass similarity. Further enhancing the interclass dissimilarity and intraclass similarity among HSI data still needs to be studied in our future work.

REFERENCES

- [1] Y. Gu, J. Chanussot, X. Jia, and J. A. Benediktsson, “Multiple kernel learning for hyperspectral image classification: A review,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6547–6565, Nov. 2017.
- [2] Q. Li, Q. Wang, and X. Li, “Exploring the relationship between 2D/3D convolution for hyperspectral image super-resolution,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 10, pp. 8693–8703, Oct. 2021.

- [3] Q. Li, M. Gong, Y. Yuan, and Q. Wang, "Symmetrical feature propagation network for hyperspectral image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5536912.
- [4] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.
- [5] Z. Xie, C. Roberts, and B. Johnson, "Object-based target search using remotely sensed data: A case study in detecting invasive exotic Australian pine in south Florida," *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 6, pp. 647–660, Nov. 2008.
- [6] T. Blaschke, "Object based image analysis for remote sensing," *ISPRS J. Photogramm. Remote Sens.*, vol. 65, no. 1, pp. 2–16, Jan. 2010.
- [7] M. D. Hossain and D. Chen, "Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS J. Photogramm. Remote Sens.*, vol. 150, pp. 115–134, Apr. 2019.
- [8] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, Dec. 2019.
- [9] L. Zhang and L. Zhang, "Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities," *IEEE Geosci. Remote Sens. Mag.*, vol. 10, no. 2, pp. 270–294, Jun. 2022.
- [10] S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, and S. Yu, "A survey: Deep learning for hyperspectral image classification with few labeled samples," *Neurocomputing*, vol. 448, pp. 179–204, Aug. 2021.
- [11] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–10.
- [12] 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.
- [13] Z. Zhong, J. Li, D. A. Clausi, and A. Wong, "Generative adversarial networks and conditional random fields for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3318–3329, Jul. 2020.
- [14] J. Wang, F. Gao, J. Dong, and Q. Du, "Adaptive DropBlock-enhanced generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5040–5053, Jun. 2021.
- [15] S. K. Roy, J. M. Haut, M. E. Paoletti, S. R. Dubey, and A. Plaza, "Generative adversarial minority oversampling for spectral–spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5500615.
- [16] N.-T. Tran, T.-A. Bui, and N.-M. Cheung, "Dist-GAN: An improved GAN using distance constraints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 370–385.
- [17] P. H. Swain and H. Hauska, "The decision tree classifier: Design and potential," *IEEE Trans. Geosci. Electron.*, vol. GE-15, no. 3, pp. 142–147, Jul. 1977.
- [18] A. Villa, J. A. Benediktsson, J. Chanussot, and C. Jutten, "Hyperspectral image classification with independent component discriminant analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 4865–4876, Dec. 2011.
- [19] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," *Proc. SPIE*, vol. 3584, pp. 221–232, Jan. 1999.
- [20] B. Waske, S. van der Linden, J. A. Benediktsson, A. Rabe, and P. Hostert, "Sensitivity of support vector machines to random feature selection in classification of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2880–2889, Jul. 2010.
- [21] T. Xie and T. Zhang, "The fast clustering algorithm for the big data based on K-means," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 18, no. 6, Nov. 2020, Art. no. 2050053.
- [22] X. Chen, T. Fang, H. Huo, and D. Li, "Graph-based feature selection for object-oriented classification in VHR airborne imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 1, pp. 353–365, Jan. 2011.
- [23] X. Kang, S. Li, and J. A. Benediktsson, "Feature extraction of hyperspectral images with image fusion and recursive filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3742–3752, Jun. 2014.
- [24] B. Song et al., "Remotely sensed image classification using sparse representations of morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5122–5136, Aug. 2014.
- [25] Y. Yuan, J. Lin, and Q. Wang, "Hyperspectral image classification via multitask joint sparse representation and stepwise MRF optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2966–2977, Dec. 2016.
- [26] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 4959–4962.
- [27] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
- [28] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, "Scalable recurrent neural network for hyperspectral image classification," *J. Supercomput.*, vol. 76, no. 11, pp. 8866–8882, Nov. 2020.
- [29] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
- [30] X. Ma, H. Wang, and J. Geng, "Spectral–Spatial classification of hyperspectral image based on deep auto-encoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 9, pp. 4073–4085, Sep. 2016.
- [31] X. Zhang, Y. Liang, C. Li, N. Huyan, L. Jiao, and H. Zhou, "Recursive autoencoders-based unsupervised feature learning for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 1928–1932, Nov. 2017.
- [32] S. Mei, J. Ji, Y. Geng, Z. Zhang, X. Li, and Q. Du, "Unsupervised spatial–spectral feature learning by 3D convolutional autoencoder for hyperspectral classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6808–6820, Sep. 2019.
- [33] M. He, B. Li, and H. Chen, "Multi-scale 3D deep convolutional neural network for hyperspectral image classification," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3904–3908.
- [34] Z. Gong, P. Zhong, Y. Yu, W. Hu, and S. Li, "A CNN with multiscale convolution and diversified metric for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3599–3618, Jun. 2019.
- [35] 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, pp. 1–16, 2022.
- [36] X. Li, L. Wang, and H. Wang, "CSI-cluster indoor fingerprint localization algorithm based on 3DCNN," *J. Chongqing Univ. Posts Telecommun. Natural Sci. Ed.*, vol. 32, no. 3, pp. 345–355, 2020.
- [37] R. Hang, F. Zhou, Q. Liu, and P. Ghamisi, "Classification of hyperspectral images via multitask generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1424–1436, Feb. 2021.
- [38] J. Feng, H. Yu, L. Wang, X. Cao, X. Zhang, and L. Jiao, "Classification of hyperspectral images based on multiclass spatial–spectral generative adversarial networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5329–5343, Aug. 2019.
- [39] M. E. Paoletti et al., "Capsule networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2145–2160, Apr. 2019.
- [40] X. Wang, K. Tan, Q. Du, Y. Chen, and P. Du, "Caps-TripleGAN: GAN-assisted CapsNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7232–7245, Sep. 2019.
- [41] G. Licciardi, P. R. Marpu, J. Chanussot, and J. A. Benediktsson, "Linear versus nonlinear PCA for the classification of hyperspectral data based on the extended morphological profiles," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 447–451, May 2012.
- [42] H. Huang, G. Shi, H. He, Y. Duan, and F. Luo, "Dimensionality reduction of hyperspectral imagery based on spatial–spectral manifold learning," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2604–2616, Jun. 2020.
- [43] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1558–1566.
- [44] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 2642–2651.
- [45] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–11.
- [46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 5767–5777.
- [47] R. O. Green et al., "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, no. 3, pp. 227–248, Sep. 1998.

- [48] B. Kunkel, F. Blechinger, R. Lutz, R. Doerffer, H. van der Piepen, and M. Schroder, "ROSIS (Reflective optics system imaging spectrometer)—A candidate instrument for polar platform missions," in *Proc. SPIE*, Apr. 1988, pp. 134–141.
- [49] W. Hu, Y. Huang, L. Wei, F. Zhang, and H. Li, "Deep convolutional neural networks for hyperspectral image classification," *J. Sensors*, vol. 2015, pp. 1–12, Jun. 2015.
- [50] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-D deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, Aug. 2018.
- [51] D. Hong et al., "SpectralFormer: Rethinking hyperspectral image classification with transformers," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5518615.
- [52] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, nos. 4–5, pp. 411–430, Jun. 2000.
- [53] A. C. Belkina, C. O. Ciccolella, R. Anno, R. Halpert, J. Spidlen, and J. E. Snyder-Cappione, "Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets," *Nature Commun.*, vol. 10, no. 1, p. 5415, Nov. 2019.



Anyong Qin received the B.S. degree in information and computational science and the Ph.D. degree in computer science from Chongqing University, Chongqing, China, in 2012 and 2019, respectively.

He is currently a Lecturer with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing. His research interests include pattern recognition, image processing, machine learning, and hyperspectral images.



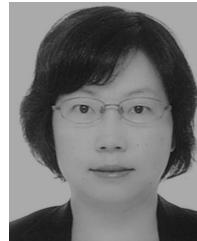
Zhuolin Tan received the bachelor's degree in electronic and information engineering from Leshan Normal University, Leshan, China, in 2020. She is currently pursuing the master's degree with the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China.

Her research interests include computer vision, image processing, deep learning, and hyperspectral images.



Ran Wang is currently pursuing the bachelor's degree with the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China.

His research interests include deep learning and hyperspectral images.



Yongqing Sun (Member, IEEE) received the B.E. and M.E. degrees in computer science from Xi'an Jiaotong University (XJTU), Xi'an, China, in 1998 and 2001, respectively, and the Ph.D. degree in information and computer science from Keio University, Tokyo, Japan, in 2005.

In 2005, she joined NTT Laboratories, Yokosuka, Japan, where she has been engaged in research and development of video/image processing technologies. Her research interests include image processing, pattern recognition, and machine learning.



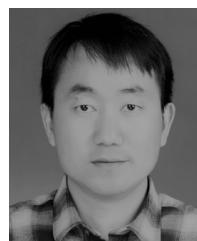
Feng Yang received the B.S. degree in science and technology of remote sensing and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2013 and 2018, respectively.

She is currently a Lecturer with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China. Her research interests include dynamic texture analysis, video processing, and remote sensing imaging.



Yue Zhao received the Ph.D. degree from Jilin University, Changchun, China, in 2017.

She is currently a Lecturer with the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China. Her research interests include image processing and analysis, pattern recognition, and medical image processing.



Chenqiang Gao received the B.S. degree in computer science from the China University of Geosciences, Wuhan, China, in 2004, and the Ph.D. degree in control science and engineering from the Huazhong University of Science and Technology, Wuhan, in 2009.

In August 2009, he joined the School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China. In September 2012, he joined the Informedia Group, School of Computer Science, Carnegie Mellon University (CMU), Pittsburgh, PA, USA, working on multimedia event detection (MED) and surveillance event detection (SED) as a Visiting Scholar. His research interests include infrared target detection, action recognition, and event detection.