CVA^{2}E: A Conditional Variational Autoencoder With an Adversarial Training Process for Hyperspectral Imagery Classification

Xue Wang, Kun Tan, *Senior Member, IEEE*, Qian Du, *Fellow, IEEE*, Yu Chen, and Peijun Du, *Senior Member, IEEE*

***Abstract*— Deep generative models such as the generative adversarial network (GAN) and the variational autoencoder (VAE) have obtained increasing attention in a wide variety of applications. Nevertheless, the existing methods cannot fully consider the inherent features of the spectral information, which leads to the applications being of low practical performance. In this article, in order to better handle this problem, a novel generative model named the conditional variational autoencoder with an adversarial training process (CVA^{2}E) is proposed for hyperspectral imagery classification by combining variational inference and an adversarial training process in the spectral sample generation. Moreover, two penalty terms are added to promote the diversity and optimize the spectral shape features of the generated samples. The performance on three different real hyperspectral data sets confirms the superiority of the proposed method.**

***Index Terms*— Generative adversarial network (GAN), hyperspectral image (HSI) classification, variational autoencoder (VAE).**

I. Introduction

HYPERSPECTRAL image classification has recently attracted considerable attention in the field of Earth observation as the contiguous spectral information can be utilized to discriminate different categories [1]. Discriminative models such as support vector machine (SVM) [2], multiple logistic regression (MLR) [3], convolutional neural networks (CNNs) [4], long short-term memory (LSTM) networks [5], and CapsuleNet [6] are advantageously used in classification because of their sampling models and higher precision when compared with generative models. However, the conditional distribution cannot describe the distribution characteristic and *a priori* knowledge of hyperspectral data. Moreover, obtaining enough labeled samples to train a classifier might not be realistic because of the wide spatial coverage and the costly field surveying and labeling. To address these issues, many generative models have been put forward in the last few years. Generative models explore a joint distribution and focus on high-order correlation rather than classification boundaries only. However, the shallow generative models show a poor performance in optimization because of their limited representation ability for high-dimensional remote sensing data sets. Motivated by deep learning, which has made great progress in many fields, deep generative models [7] have been very successful in computer vision tasks. The most important representative methods are the variational autoencoder (VAE) [8] and the generative adversarial network (GAN) [9]. The principle of the VAE and the GAN is to learn a mapping from a latent distribution to a data space.

The difference between these two generative models is that variational inference is carried out to reduce the distance between different distributions in the VAE, while the GAN trades the complexity of sampling for the complexity of searching for a Nash equilibrium in minimax games. Both models have a remarkable ability to generate samples that are similar to real samples, which have been successfully applied in data augmentation. Before the deep generative models, data augmentation for remote sensing classification depended on the following strategies: 1) sample transformation, such as rotation and translation and 2) label propagation, driven by data or sample simulation based on a physical model, such as spectral curve simulation under different illumination of the same ground field. However, these enhanced approaches are heavily reliant on the assumption of the data and the physical environment. The VAE and the GAN have provided a new pathway for feature learning and sample augmentation, to address the issue of insufficient samples. Although the VAE has worked well in generating reliable samples, the L2 norm gives rise to blur in the generated samples. On the contrary, the GAN can create clear samples, but it suffers from model

Manuscript received October 4, 2019; revised November 17, 2019; accepted December 24, 2019. Date of publication February 10, 2020; date of current version July 22, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 41871337 and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions. *(Xue Wang and Kun Tan contributed equally to this work.) (Corresponding author: Kun Tan.)*

Xue Wang and Kun Tan are with the Key Laboratory of Geographic Information Science (Ministry of Education), East China Normal University, Shanghai 200241, China, and also with the Key Laboratory for Land Environment and Disaster Monitoring of National Administration of Surveying and Geoinformation (NASG), China University of Mining and Technology, Xuzhou 221116, China (e-mail: tankuncu@gmail.com).

Qian Du is with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762 USA.

Yu Chen is with the Key Laboratory for Land Environment and Disaster Monitoring of National Administration of Surveying and Geoinformation (NASG), China University of Mining and Technology, Xuzhou 221116, China.

Peijun Du is with the Key Laboratory for Satellite Mapping Technology and Applications of National Administration of Surveying and Geoinformation (NASG), Nanjing University, Nanjing 210023, China (e-mail: dupjrs@gmail.com).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.2968304

collapse and gradient vanishing, which can result in the samples generated from the GAN being far from natural. To tackle these defects, the structure of the VAE and the GAN has been unceasingly perfected. Some works have used an adversarial training process for the VAE [10], [11] or an additional network [12] as the generator or discriminator, which have been proved to be effective ways to fix the blurring problem for the VAE. For the model collapse and gradient vanishing of the GAN, the loss function has been modified, as in the Wasserstein GAN [13] and the least-squares GAN [14]. Other tricks include batch normalization, distribution sampling, and the choice of the optimizer, which are not considered in this article. Moreover, considering the label information, the VAE and the GAN can be modified to perform supervised learning and semisupervised learning tasks.

Several articles have addressed classification in remote sensing images using the GAN [15]–[17]. Zhu *et al.* [18] put forward a hyperspectral image (HSI) classification algorithm based on an auxiliary classifier GAN [19], where the samples generated from the GAN are added to improve the performance of the classifier. Experiments on real hyperspectral data sets confirmed the validity of the generated samples. However, the effect of the improvement was proven to be limited by Xu *et al.* [20], and the reason for the limitation is that the generated samples cannot cover all the feature space. To solve the problem of the limited improvement, some works have focused on the loss function and training samples. Wang *et al.* [21] proposed a removal strategy to weaken the side effects of data outliers and generate high-quality samples. Audebert *et al.* [22] introduced the Wasserstein distance to ensure diversity of the generated samples, and the experiments undertaken in this study showed that the cluster centers of the generated samples are consistent with those of real samples. ShiftingGAN [23] uses an "online-output" model to obtain multiple generators, so that the generated samples are more diverse. Moreover, to keep the high quality of the generated samples, two additional shifting processes are added.

Although these works have proved that the generated samples from the GAN can improve the performance of classification, the improvement is unstable because of the limitation in the enhancement of quantity and diversity.

Recently, there have been a few articles utilizing the VAE in remote sensing, instead of the GAN. For example, Gemp *et al.* [24] proposed a deep semisupervised generative model, in which the VAE is employed to extract the spectra of the endmembers and retrieve the mineral spectra. Gong *et al.* [25] utilized the VAE for change detection in multispectral imagery, but the experiments indicated that the difference images were blurred because of the reconstruction loss of the training data in the VAE. Su *et al.* [26] proposed a VAE-based hyperspectral unmixing method named the deep autoencoder network (DAEN), in which the VAE performs blind source separation after the spectral signatures are extracted, and the VAE can ensure the nonnegativity and sum-to-one constraints when estimating the abundances.

Inspired by the adversarial autoencoder (AAE) [10] and the conditional GAN (CGAN) [27], we propose a variational GAN with label information named conditional variational autoencoder with an adversarial training process (CVA^{2}E). The new framework consists of a variational encoder to ensure the diversity when using the latent variable distribution, a generator to reconstruct the samples from the latent variable distribution, and a discriminator to determine if the data are from real data or have a model distribution. Moreover, two fully connected layers are added in the discriminator to improve the classification ability of the framework. Considering the inherent difference (hundreds of spectral bands in HSIs) in the spectral dimensionality between HSIs and the common images used in computer vision tasks, the spectral angle distance is used as one of the observation items.

The rest of this article is organized as follows. Section II gives the background to this article. Section III details the CVA^{2}E framework. Section IV describes the three real HSIs used in the experiments, the experimental results, and the comparisons with other methods. Finally, the conclusions of this article are drawn in Section V.

II. Previous Work

In this article, the main work is based on two kinds of generative models: the GAN and the VAE. Therefore, in this section, we briefly review these two models.

A. Generative Adversarial Network

The GAN uses deep neural networks to approximate an unknown data distribution and is typically composed of two networks named the generator and discriminator. The former is aimed at learning the distribution characteristic of the data and creating new data, and the discriminator learns to infer whether the sample is from a model distribution or a real distribution. When the training is over, the generator and discriminator converge to a Nash equilibrium, in which the discriminator cannot distinguish whether the sample is real data or generated data from the generator, i.e., the generated samples are indistinguishable from real samples. These two roles are both deep neural networks, and the generator's training target is a certain distribution $p_z(z)$ which is consistent with the sample $G(z)$'s data space. The generator is denoted as $G(z;\theta_g)$, and θ_g represents the parameters of the deep neural networks. For the discriminator, $D(x;\theta_d)$ represents a deep neural network with parameters θ_d. The training process is to solve a minimax problem by a two-player game

$$\min_G \max_G V(G,D) = \mathbb{E}_{x\sim p(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

where $x \sim p(x)$ is the real data distribution. The optimal model $D(x)$ is $p(x)/(p(x)+p_g(x))$, and the global equilibrium of this two-player game is obtained when $p(x)=p_g(x)$. However, the prototype of the GAN model is difficult to converge in the training stage, and the samples generated from the GAN are often far from natural. To this end, many works have tried to improve the stability by modifying the loss function. For example, the Wasserstein GAN substitutes the original Kullback–Leibler divergence and Jensen–Shannon divergence for the Earth mover's distance. The Wasserstein GAN is aimed at solving the problem

$$\max_{w \in w} E_{x \sim p(x)}[f_w(x)] - E_{z \sim p_z(z)}[f_w(g_\theta(z))] \tag{2}$$

where w represents the parameters of function f, and $w \in W$ is a strong assumption akin to assuming that w meets the K-Lipschitz constraint $\|f\|_w \leq K$ when proving the consistency of f. The Wasserstein GAN can improve the stability of the learning and get rid of problems such as mode collapse, whereas the range of the parameters of the discriminator is limited, to meet the K-Lipschitz constraint, which decreases the discriminative power.

B. Variational Autoencoder

The VAE uses a Kullback–Leibler divergence penalty to make its hidden code vector like a prior distribution, i.e., the VAE performs efficient approximate inference and learning with directed models using a continuous latent intractable posterior distribution. The reparameterization is carried out to yield a simple differentiable unbiased estimator of the variational lower bound, whose Kullback–Leibler divergence is straightforward to optimize using the standard stochastic gradient descent technique. The encoder Q of the autoencoder is used as the probabilistic function approximator $q_\tau(z \mid x)$, and the decoder P is used as the approximation of the posterior of the generative model $p_\theta(x, z)$. The parameters τ and θ are optimized jointly with the objective function

$$V(P, Q) = -\mathrm{KL}(q_\tau(z \mid x) \| p_\theta(z \mid x)) + \mathrm{Recost}(x) \tag{3}$$

where Recost() calculates the reconstruction loss of a given sample x through the VAE.

There have been many developments based on the VAE and the GAN. For example, combining the VAE and the GAN to construct a more powerful deep generative model [10], [11]. Introducing an adversarial training process to the autoencoder or using variational inference and elementwise measurement in the GAN's observation function can also improve the drawbacks of the original model. Furthermore, the VAE and the GAN can be modified to consider label information and trained to conduct conditional generation, e.g., the conditional variational autoencoder (CVAE) and the CGAN.

III. Proposed Method

In this section, we describe the proposed CVA^{2}E framework for HSI classification and generation, which is illustrated in Fig. 1. First, we introduce the notations that are adopted throughout this article. If we suppose that a hyperspectral data set with b spectral bands contains N labeled samples for L classes, and each is represented by $\{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^{1\times b}$, then the corresponding label vector is $Y = \{y_1, y_2, \ldots, y_L\} \in \mathbb{R}^{1\times L}$.

As mentioned earlier, the GAN and the VAE have shown good performances in data generation, whereas their latent variables cannot learn the label knowledge during the adversarial training and variational inference process. Moreover, the intrinsic loss measurement of the VAE gives rise to blurry generated samples, which result in defective pattern learning for hyperspectral data. There will be various samples in the same category in hyperspectral imagery, so the diversity of the generative model can be guaranteed. The GAN can create clear samples, which is an improvement over the VAE, but it always overfits on local properties, which leads to the generated spectra samples appearing disordered. To handle the above problems, we propose CVA^{2}E.

As shown in Fig. 1, the spectral feature vectors are utilized as the input of CVA^{2}E as we try to explore the individual hyperspectral pixels with no spatial context. Inspired by Larsen *et al.* [12], the CVA^{2}E framework consists of three networks: an encoder network E, a generator network G, and a discriminator network D. There are two connected parts of discriminator network D, which execute the distinguishment and the classification task, respectively. All the networks are deep fully connected networks. For HSI classification, it is critical to assign a high-dimensional spectral pixel with the correct label, which can be represented by the conditional distribution $p(y \mid x)$. The input of E and G concatenates the original spectral vector x with the 1-D categorical vector y. With the connection of the spectral and categorical features, the encoder network E maps x to a latent variable z through the probabilistic function approximator $P_E(z \mid x, y)$, where y is the category of x. Following the encoder process, the sampled z is utilized to generate the fake data through the other learned probabilistic function approximator $P_G(x \mid z, y)$ in the generative network G. After this, D establishes whether the data are real data or a model distribution, and G tries to learn the real data distribution during the adversarial training process.

A. Variational Inference Process of CVA^{2}E

The variational inference process in CVA^{2}E is the same as in the VAE, where the input data consider the category information. The principle of the VAE is to learn a distribution which has a certain sampling space to sample the real data distribution x. The learned distribution is described as $z = N(\mu, \sigma^2)$. The latent variable z is encoded by $E(z, \theta_g)$ and is sampled from this distribution, which is based on reparameterization, where θ_g represents the parameters of the deep neural network. The objective is to minimize the distance between z and a given distribution (in this work, the given distribution is a standard normal distribution). The distance can be denoted as follows:

$$L = -D(\mathcal{N}(\mu, \sigma^2)||\mathcal{N}(0, 1)) + \mathbb{E}_E[\log P(x|\mu, \sigma^2)] \tag{4}$$

where μ and σ are obtained by the encoder network E. It is often possible to express a random variable z which obeys the Gaussian distribution as a deterministic variable $z = N(\mu, \sigma^2)$, where μ and σ are obtained by feedforward operation of the deep neural network in the VAE. The values of μ and σ can be calculated and taken as derivatives, whereas this is not possible for the sample process. In this case, reparameterization is useful since it can be used to rewrite the sample process as $z = \mu + \sigma\xi$, where ξ is an auxiliary noise variable and $\xi = \mathcal{N}(0, 1)$.

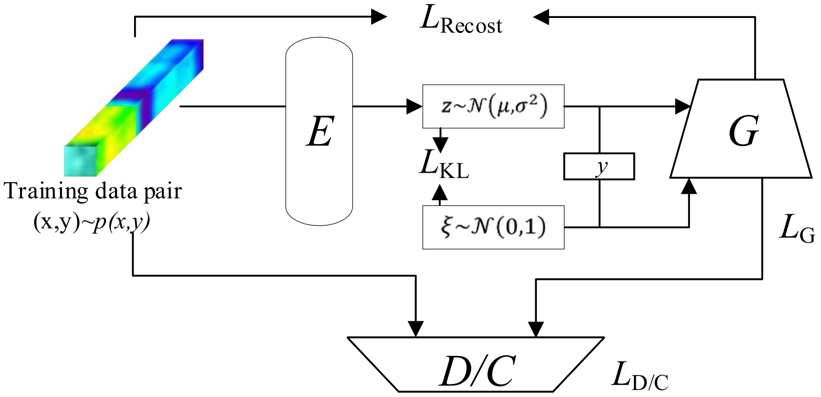


Fig. 1. Construction of CVA^{2}E.

Therefore, $\mathbb{E}_{z\sim\mathcal{N}(\mu,\sigma^2)}[f(z)] = \mathbb{E}_{\xi\sim\mathcal{N}(0,1)}[f(\mu+\sigma^2\xi)]$. After this, the sample process is no longer involved in the gradient descent but the result of the sampling is used, which makes the model straightforward to optimize using the standard stochastic gradient descent technique. Therefore, the variational inference process $D(\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1))$ can be transformed by reparametrizing as follows:

$$\begin{aligned} L_{\text{KL}} &= D(\mathcal{N}(\mu,\sigma^2)||\mathcal{N}(0,1)) \\ &= 0.5(1+\log\sigma^2 - \mu^2 - \exp(\log\sigma^2)) \end{aligned} \tag{5}$$

where σ and μ are the latent variables encoded by E. The sample process of z is based on reparameterization of variable ξ. The Kullback–Leibler divergence penalty is utilized to make its latent variable z like a prior distribution. The second part of (4) is replaced by binary cross entropy, which is the same as the VAE, to represent the reconstruction loss

$$\begin{aligned} L_{\text{Recost}} &= \mathbb{E}_E[\log P(x|\mu,\sigma^2)] \\ &= x\log(G(z,y)) + (1-x)\log(1-G(z,y)) \end{aligned} \tag{6}$$

where $G(z,y)$ represents the consideration of the category in the generative process.

B. Least-Squares Loss of CVA^{2}E

The GAN is based on a minimax two-player game, which can provide a powerful sample approach to estimate the target distribution and generate new samples. The adversarial training process of the regular GAN is shown in (1), which can be transformed into a two-player game as follows:

$$L_{\text{D}} = -\mathbb{E}_{x\sim p(x)}[\log D(x)] - \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{7}$$

$$L_{\text{G}} = -\mathbb{E}_{z\sim p_z(z)}[\log(D(G(z)))]. \tag{8}$$

Equations (7) and (8) denote the two trained targets corresponding to D and G. The optimizing direction is to close the distance between the real distribution and the model distribution. The measurement of the distance impacts on the convergence of these two probability distributions. In general, the convergence of the distribution is easier when the distance induces a weaker topology. Unfortunately, Jensen–Shannon divergence is utilized in the regular GAN, which is not a stable cost function when learning distributions supported by low-dimensional manifolds. This is because the supports of $x \sim p(x)$ and $x \sim p_g(x)$ have an empty intersection. Intuitively, for the trained discriminator, we have $D(x) \to 1$ and $D(G(z)) \to 0$, so the gradient $\partial L_{\text{G}}/\partial(D(G(z))) \to -\infty$, which may lead to the gradient-vanishing problem during the learning process. Therefore, $x \sim p_g(x)$ cannot be represented explicitly, and D must be synchronized well with G during the training process, which causes the regular GAN to be unstable. This kind of GAN works well for either categorical discrimination or generation, but cannot be optimal at the same time. The Earth mover's distance-based Wasserstein GAN improves the instability, but the convergence is much slower than the regular GAN. In this work, we introduce least-square loss for the discriminator to address the gradient-vanishing problem during the learning process, which is an approach that is inspired by the least-squares GAN [13]. The idea is that the least-squares loss function is able to move the fake samples toward the decision boundary because the least-squares loss function penalizes samples that lie a long way from the correct side of the decision boundary and pulls them toward the decision boundary, even though they are correctly classified. In this way, the least-squares GAN is more stable during the learning process. Moreover, when the generator is updated, the parameters of the discriminator are fixed, which results in the training of the least-squares GAN being faster to converge than the Wasserstein GAN, as the Wasserstein GAN requires multiple updates for the discriminator. The objective functions for the traditional least-squares GAN are defined as follows:

$$\begin{aligned} L_{\text{D}} &= -0.5\mathbb{E}_{x\sim p(x)}[(D(x)-1)^2] \\ &\quad - 0.5\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-0)^2] \end{aligned} \tag{9}$$

$$L_{\text{G}} = -0.5\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-1)^2] \tag{10}$$

where 1 and 0, respectively, denote the labels for real data and fake data in (9), and 1 denotes the value that the generator wants the discriminator to believe for fake data in (10). Because the CVA^{2}E framework incorporates the variational inference process, the generated inputs are not only from the normal distribution samples but also from the latent distribution encoded from E. Furthermore, to create the deterministic relationship between the multiple categories, the label information is added. Therefore, the objective functions of least-squares loss are transformed as follows:

$$\begin{aligned} L_{\mathrm{D}} = &-0.5\mathbb{E}_{x\sim p(x)}[(D(x|y)-1)^2] \\ &-0.5\mathbb{E}_{z\sim p_z(z)}[(D(G(z|y))-0)^2] \\ &-0.5\mathbb{E}_{\xi\sim\mathcal{N}(0,1)}[(D(G(\xi|y))-0)^2] \end{aligned} \tag{11}$$

$$\begin{aligned} L_{\mathrm{G}} = &-0.5\mathbb{E}_{z\sim p_z(z)}[(D(G(z|y))-1)^2] \\ &-0.5\mathbb{E}_{\xi\sim\mathcal{N}(0,1)}[(D(G(\xi|y))-1)^2] \end{aligned} \tag{12}$$

where $G(z|y)$ denotes that the initial samples are sampled from $z\sim\mathcal{N}(\mu,\sigma^2)$, and μ,σ are obtained by encoder network E. $G(\xi|y)$ denotes that the initial samples are sampled from $\xi\sim\mathcal{N}(0,1)$. The initial samples are then input into the generator network G to obtain the final fake data. Therefore, the discriminator network D should recognize if a sample is from the real data distribution or from the two fake distributions, which are described as the three parts of (11), and the generator network should use both the conditional normal distribution and the latent distribution to fool the discriminator.

C. Enhancement of Diversity and Spectral Characteristics With CVA^{2}E

The motivation of the CVA^{2}E framework is to learn the spectral distribution characteristics of individual hyperspectral pixels. While the deep neural network is a remarkable function approximator, the inherent difference (hundreds of spectral bands) in spectral dimensionality between HSIs and the common images used in computer vision tasks should be taken into account. Regular loss functions are generally based on cross-entropy or least-squares loss, which focuses on the local feature matching. Here we introduce the spectral angle distance to match the generated samples based on the similarity of the curves of the spectra. The spectral angle match method was proposed by Kruse [28]. This method regards the spectrum of an individual hyperspectral pixel in the image as a high-dimensional vector and measures the similarity between the spectra by calculating the vectorial angle between the two high-dimensional vectors, where the smaller the angle, the more similar the two spectra are, and the more reliable the generator will be. Here we utilize cosine similarity to replace the spectral angle

$$\begin{aligned} L_{\mathrm{SAD}} = \frac{1}{\mathrm{bs}}\sum\Bigg[&\left(\frac{G(z|y)^{\mathrm{T}}x}{\sqrt{G(z|y)^{\mathrm{T}}G(z|y)}\sqrt{x^Tx}}+1\right) \\ &+\left(\frac{G(\xi|y)^{\mathrm{T}}x}{\sqrt{G(\xi|y)^{\mathrm{T}}G(\xi|y)}\sqrt{x^{\mathrm{T}}x}}+1\right)\Bigg] \end{aligned} \tag{13}$$

where bs represents the batch size in the training process. The spectral angles are calculated between the real spectra and the two kinds of generative spectra. Inspired by the spectral angle match method, we utilize the vectorial angle measurement on an intermediate layer of the generator network. The difference in purpose is that we want the feature space of the generative samples to be big enough to contain the entire training data distribution, and thus reduce the likelihood of mode collapse, so the vectorial angle between pairwise features of an intermediate layer should be the maximum. The vectorial angle of the features is calculated as follows:

$$\begin{aligned} &L_{\mathrm{FVA}} \\ &=\frac{1}{L}\sum_{l}^{L}\frac{1}{S^l(S^l-1)} \\ &\times\sum_{m}^{S^l}\sum_{n\neq m}^{S^l}\frac{F(z^m\mid y_l)^{\mathrm{T}}F(z^n\mid y_l)}{\sqrt{F(z^m\mid y_l)^{\mathrm{T}}F(z^m\mid y_l)}\sqrt{F(z^n\mid y_l)^{\mathrm{T}}F(z^n\mid y_l)}}+1 \end{aligned} \tag{14}$$

where bs represents the batch size in the training process, and $F(\cdot)$ denotes the features of an intermediate layer. In this work, the output of the penultimate layer of the generator network is chosen as the compared features. The vectorial angle is calculated among the samples which belong to the same category. Each $F(z)$ should be calculated with the feature which is the same category in the ergodic case, except itself. After L iterations (L is the number of categories), the vectorial angle of the features is the cumulative sum in $S^l(S^l-1)L$ times, and the average is obtained using division by the number of times. A larger L_{FVA} means that the features are similar and that the generative model should have low diversity.

The final structure of CVA^{2}E is shown in Fig. 2. The input of the discriminator network is the concatenation of the spectral and categorical vectors, which is used to estimate the two joint probability distributions $p(x_g,y)$ and $p(x,y)$. Moreover, the discriminator network is reused to execute the classification task. When the network plays the role of classifier, the categorical vector is replaced by a zero vector, so that the weights in the discriminator network corresponding to categorical features are disabled, and the last layer is replaced by a softmax layer to obtain the posterior probability $p(y|x)$ or $p(y|x_g)$.

Up until now, the final goal of CVA^{2}E can be shown as follows:

$$L = L_{\mathrm{D}}+L_{\mathrm{G}}+L_{\mathrm{C}}+L_{\mathrm{KL}}+\lambda_1L_{\mathrm{FVA}}+\lambda_2L_{\mathrm{SAD}}+\lambda_3L_{\mathrm{Recost}} \tag{15}$$

where each part is given the explicit expression above. λ_1, λ_2, and λ_3 are empirically set to 0.3, 0.6, and 1. As shown in Fig. 2, "red" means the data pair from the real distribution, and "blue" and "green" denote the data pair from the generated distribution. The difference is that the "blue" data pairs are generated based on the latent distribution encoded by E, and the "green" data pairs are based on a certain distribution. L_{D} is related to the ability to distinguish between real and fake data pairs, L_{G} represents the ability of G to fool D, L_{C} denotes the capability of the network to classify spectra from different categories, L_{KL} influences the latent distribution encoded by E to obey a certain distribution, L_{Recost} is related to the variational inference, L_{FVA} improves the diversity of G, and L_{SAD} considers the spectral similarity between the real spectrum and the generated spectrum. Finally, in order to summarize the whole training process, Table I gives a detailed description of the proposed learning algorithm.

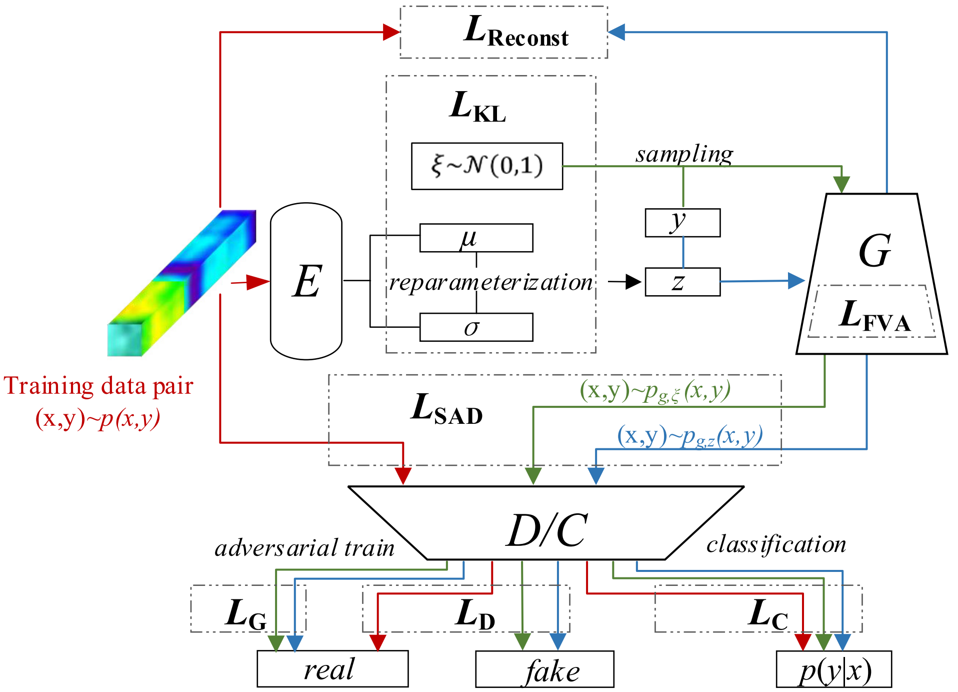
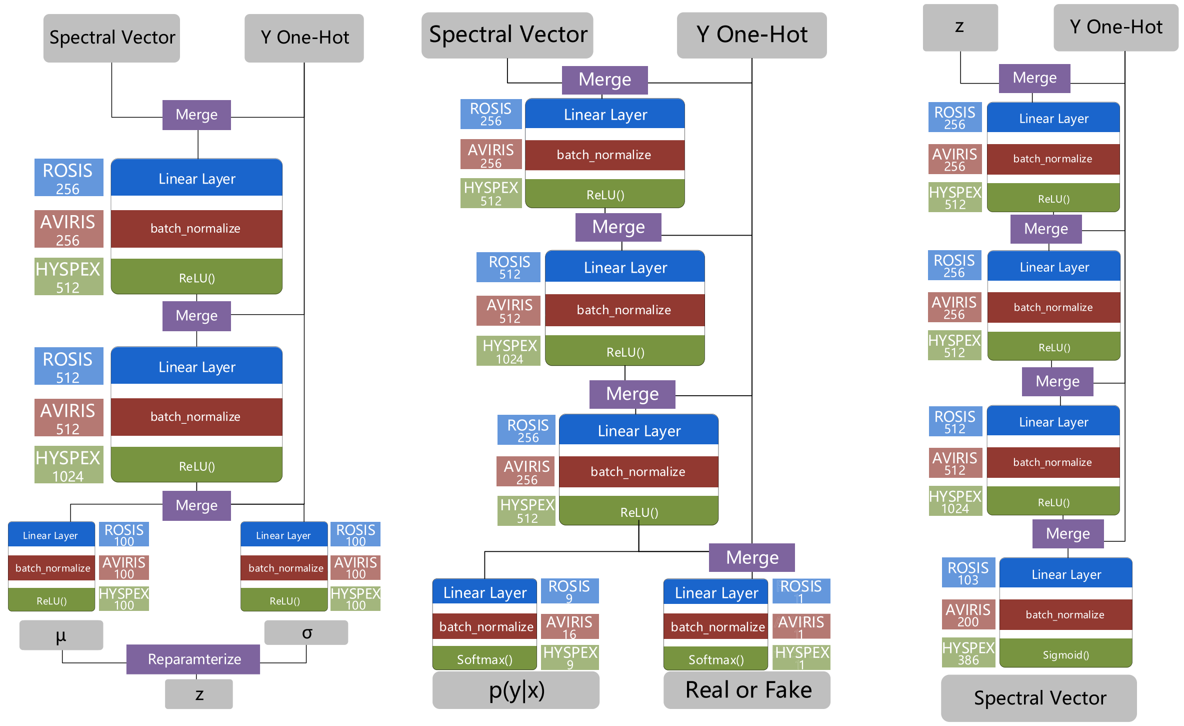


Fig. 2. Detailed illustration of the CVA^{2}E framework. The data pairs from different distributions are colored in red, blue, and green, respectively, and each loss function is labeled on the corresponding samples.

TABLE I
PSEUDOCODE OF THE PROPOSED ALGORITHM

```
**Algorithm** CVA$^2$E framework applied to a hyperspectral dataset
**Input:** HSI $X \in R^{b*rows*cols}$, training data pairs $\mathrm{T} \in \{(x_1,y_1),\ (x_2,y_2) \ldots (x_n,y_n)\}, x_n \in R^{1\times b}$ , test data pairs $\mathrm{S} \in \{(x_1,y_1),\ (x_2,y_2) \ldots (x_s,y_s)\}, x_n \in R^{1\times b}$
**Initialize: Network:** Discriminator D, generator G, encoder E; Batch size bs, $\lambda_1$ 0.3, $\lambda_2$ 0.6, $\lambda_3$ 1.
 **for** number of training iterations **do**:
    Sample $(x,y)\sim p(x,y)$ on batch size m from T
    Encode $(x,y)\sim p(x,y)$ to obtain $\mu,\sigma$
    Obtain $z$ by the reparameterization operation
      $L_{KL} \leftarrow \sum_{(x,y)\sim p(x,y)}[-0.5(1+log\sigma^2-\mu^2-\exp(log\sigma^2))]$
    Sample $(x,y)\sim p_{g,z}(x,y)$ by G on batch size $bs$ using the given label values and latent variable $z$
      $L_{Recost} \leftarrow \sum_{z\sim\mathcal{N}(\mu,\sigma^2)} -x\ log\big(G(z,y)\big)-(1-x)\ log\big(1-G(z,y)\big)$
    Sample $(x,y)\sim p_{g,\xi}(x,y)$ by G on batch size $bs$ using the given label values and $\xi$
      $L_{SAD} \leftarrow \left[\left(\sum_{(x,y)\sim p_{g,z}(x,y)} \frac{G(z|y)^T x}{\sqrt{G(z|y)^T G(z|y)}\sqrt{x^T x}}\right)+\left(\sum_{(x,y)\sim p_{g,\xi}(x,y)} \frac{G(\xi|y)^T x}{\sqrt{G(\xi|y)^T G(\xi|y)}\sqrt{x^T x}}\right)\right]$
      $L_{FVA} \leftarrow \sum_l^L \frac{1}{S^l(S^l-1)} \sum_m^{S^l} \sum_{n\neq m}^{S^l} \frac{F(z^m|y_l)^T F(z^n|y_l)}{\sqrt{F(z^m|y_l)^T F(z^m|y_l)}\sqrt{F(z^n|y_l)^T F(z^n|y_l)}}$
     Construct $(x_d,y_d)$ with $(x,y)\sim p_{g,\xi}(x,y),(x,y)\sim p_{g,z}(x,y)$ and $(x,y)\sim p(x,y)$ to train D. Assign $(x_{g,\xi},y)$ and $(x_{g,z},y)$ with a negative label, $(x,y)$ with a positive label.
       $L_D \leftarrow \left\{-0.5\sum_{(x,y)\sim p(x,y)}[(D(x|y)-1)^2]-0.5\sum_{(x,y)\sim p_{g,z}(x,y)}\left[\big(D(x|y)\big)^2\right]-0.5\sum_{(x,y)\sim p_{g,\xi}(x,y)}\left[\big(D(x|y)\big)^2\right]\right\}$
       $L_G \leftarrow \left\{-0.5\sum_{(x,y)\sim p_{g,z}(x,y)}[(D(x,y)-1)^2]-0.5\sum_{(x,y)\sim p_{g,\xi}(x,y)}[(D(x,y)-1)^2]\right\}$
     Update D with the gradient:
                    $\nabla_{\theta_d}\left[\frac{1}{\mathrm{bs}}L_D\right]$
     Calculate the cost of classification on $(x,y)\sim p(x,y)$, $(x,y)\sim p_{g,z}(x,y)$ and $(x,y)\sim p_{g,\xi}(x,y)$
     Update the softmax layer of D with the gradient:
          $\nabla_{Para_{D\_softmax}}\left[\frac{1}{\mathrm{bs}}\left(\sum_{(x,y)\sim p_{g,z}(x,y)}[-logp(y|x)]+\sum_{(x,y)\sim p(x,y)}[-logp(y|x)]\right)\right]$
     Update G with the gradient:
               $\nabla_{\theta_g}\left[\frac{1}{\mathrm{bs}}\left(L_G+\frac{\lambda_1}{\mathrm{bs}-1}L_{FVA}+\lambda_2 L_{SAD}+\lambda_3 L_{Reconst}\right)\right]$
     Update E with the gradient:
               $\nabla_{\theta_e}\left[\frac{1}{\mathrm{bs}}\left(L_G+\frac{\lambda_1}{\mathrm{bs}-1}L_{FVA}+\lambda_2 L_{SAD}+\lambda_3 L_{Reconst}+L_{KL}\right)\right]$

**Endfor**
**Output:** the final label is dependent on the output of softmax in D
```



(a) (b) (c)

Fig. 3. Implementation of each network in CVA^{2}E. (a) Encoder network. (b) Discriminator network. (c) Generator network.

TABLE II

NUMBER OF TRAINABLE PARAMETERS AND MEMORY OVERHEAD IN CVA^{2}E

Network	Indicator	ROSIS	AVIRIS	HySpex
Encoder	E_trainable_parameters	271,056	302,664	947,152
	E_ total size	1.05MB	1.18 MB	3.65 MB
Discriminator	D_trainable_parameters	303,460	337,475	1,276,024
	D_ total size	1.18MB	1.32 MB	4.91 MB
Generator	G_trainable_parameters	288,262	347,464	1,261,332
	G_ total size	1.12MB	1.35 MB	4.85 MB

IV. EXPERIMENTS

In order to demonstrate the performance of the proposed technique, three real HSIs were utilized. To explore the distribution of the real spectra, the fully connected layers were utilized to constitute the different compositions of the model. In our experiments, the category information, which was transformed as a "one-hot vector," was merged with the spectral vector. A "one-hot vector" is a vector with one single "1" and all the others as "0," where the position of "1" indicates which category the pixel belongs to. The construction of the three networks is shown in Fig. 3, where the encoder network E is a three-layer fully connected network, and the input of each layer is merged by the spectral vector and "one-hot vector" in advance. μ and σ are from two different layers, and the rectified linear unit is utilized as the activation function to obtain μ and σ. The discriminator D and generator G consist of four-layer fully connected networks. For D, when it determines real or fake samples, the "one-hot" vector is applied; otherwise, the categorical vector is set to zero, and the softmax and sigmoid functions are utilized for the two outputs. The former output is *a posteriori* probability of the categories, and the latter is used for the loss calculation of D in the adversarial training process. For G, the "one-hot" vector determines the category of the generated sample, which is represented by the results of the sigmoid function of the last layer. The number of neurons in each part of the network for three different data sets is shown in Fig. 3, the mini-batch size for training is set as 80, and the learning rate is set to 0.0002. To exhibit the complexity referring to the previous research [29], the number of trainable parameters and memory overhead in CVA^{2}E are given in Table II.

The experimental analysis starts with the generated samples by the given generative models. The compared methods were chosen from the mainstream deep generative models. In order to fairly compare each method, we used the same network structure to implement the different models. After verification of the validity of the generated samples, the overall accuracy (OA) and kappa coefficient (kappa) are used to report the performance of all the models. To demonstrate the performance of the proposed technique, three real HSIs were utilized. The training samples in all the experiments were made up of 10% of the labeled data.

A. Data Set Description

1) Pavia University ROSIS Data Set: The Pavia University data set was acquired by the Reflective Optics System Imaging

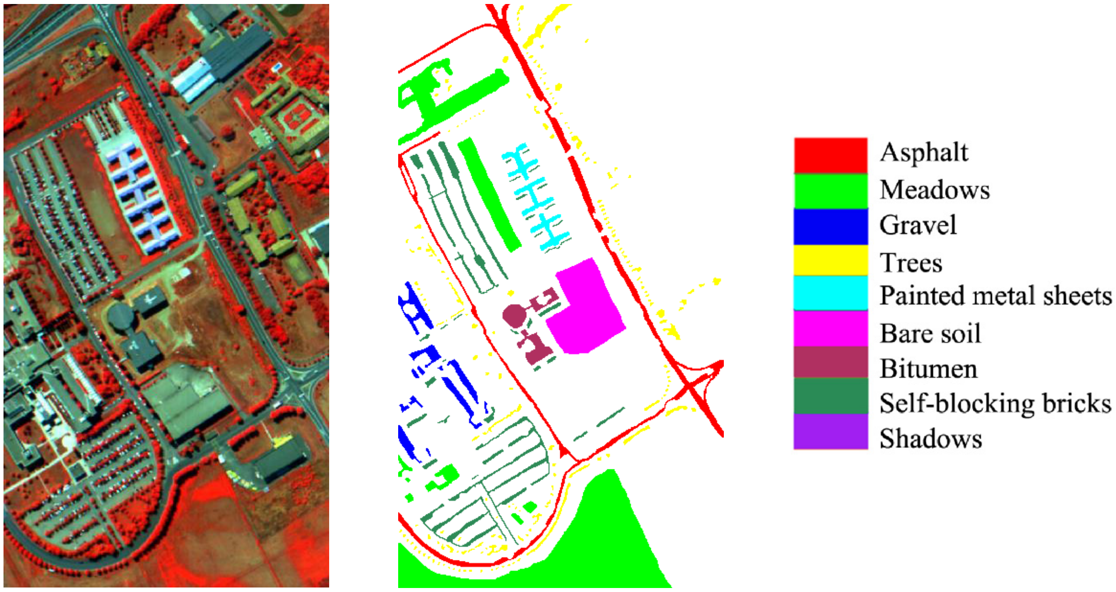


Fig. 4. Pseudocolor composite image and the corresponding ground truth for the Pavia University ROSIS data set.

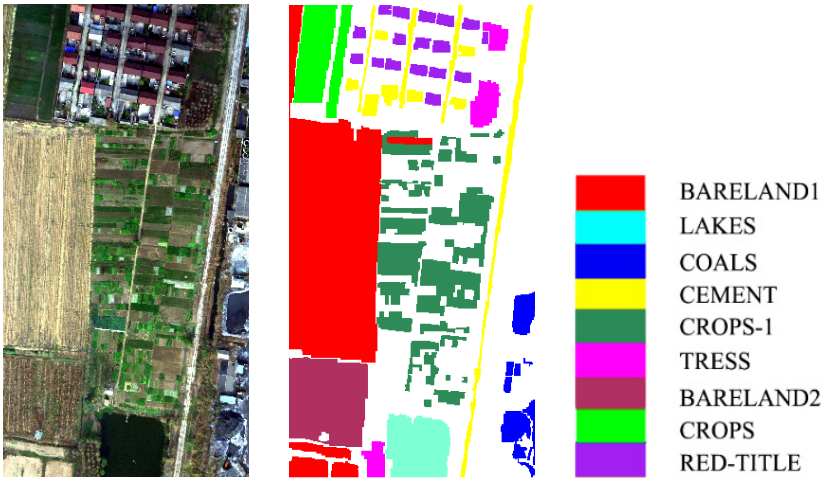


Fig. 5. Pseudocolor images and the corresponding ground truth of the Xuzhou HYSPEX data set.

Spectrometer (ROSIS) sensor over the Engineering School of the University of Pavia, Italy. This data set consists of 610 × 340 pixels, with a spatial resolution of 1.3 m/pixel. A total of 103 spectral bands ranging from 430 to 860 nm were used in the experiments after removing the noisy bands. The data set contains nine categories of interest. There are large differences in the spectral features between the different categories, which can verify the learning ability of the generative model. The pseudo-color composite image and the labeled categories are shown in Fig. 4.

2) Xuzhou HYSPEX Data Set: The Xuzhou data set was collected by an airborne HySpex hyperspectral camera over the Xuzhou peri-urban site. This data set consists of 500 × 260 pixels, with a very high spatial resolution of 0.73 m/pixel. A total of 368 bands ranging from 415 to 2508 nm were used in the experiments after removing the noisy bands. The spectral range is close to that of the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) data set, but the higher spectral resolution and more noise bands cause disturbance during the learning process of the generative model. The pseudo-color composite image and the labeled categories are shown in Fig. 5.

3) Indian Pines AVIRIS Data Set: The Indian Pines data set was collected by the AVIRIS sensor over the northwestern Indiana agricultural test site. This data set consists of 145 × 145 pixels, with a spatial resolution of 17 m/pixel. A total of 200 bands ranging from 400 to 2500 nm were used in the experiments after removing the noisy bands. The available training samples cover 16 categories of interest, which are mostly different types of vegetation. The similar spectral characteristics among the different categories brings great difficulty for the learning of the spectral feature distribution. The pseudo-color composite image and the labeled categories are shown in Fig. 6.

B. Visualization Analysis of the Generated Samples

In order to illustrate the performance of the proposed approach, we utilized a 10% labeled training set randomly

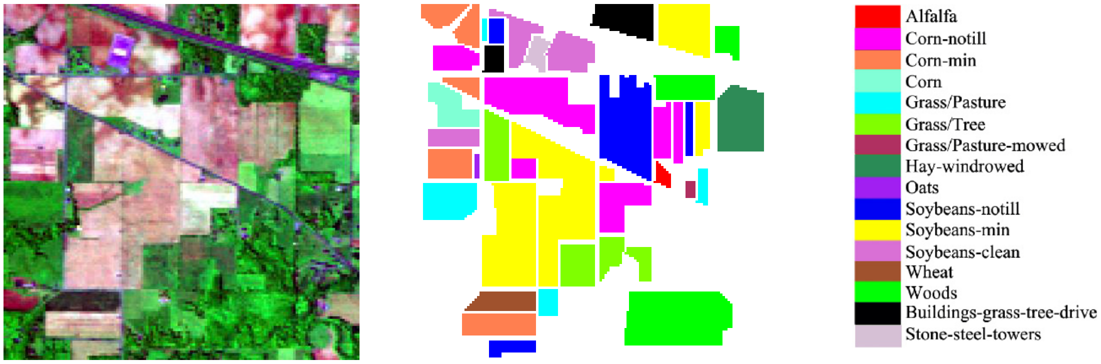
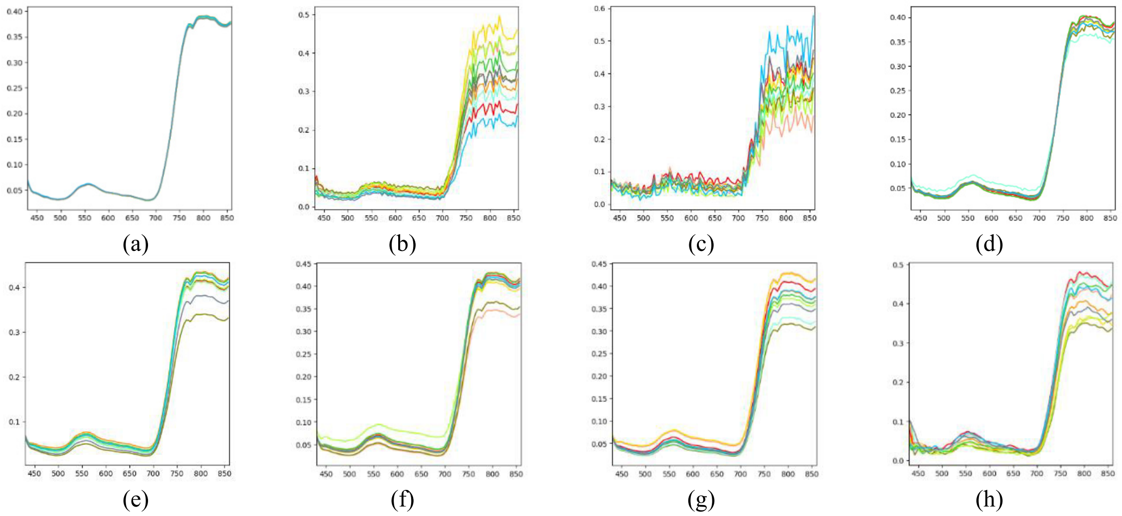


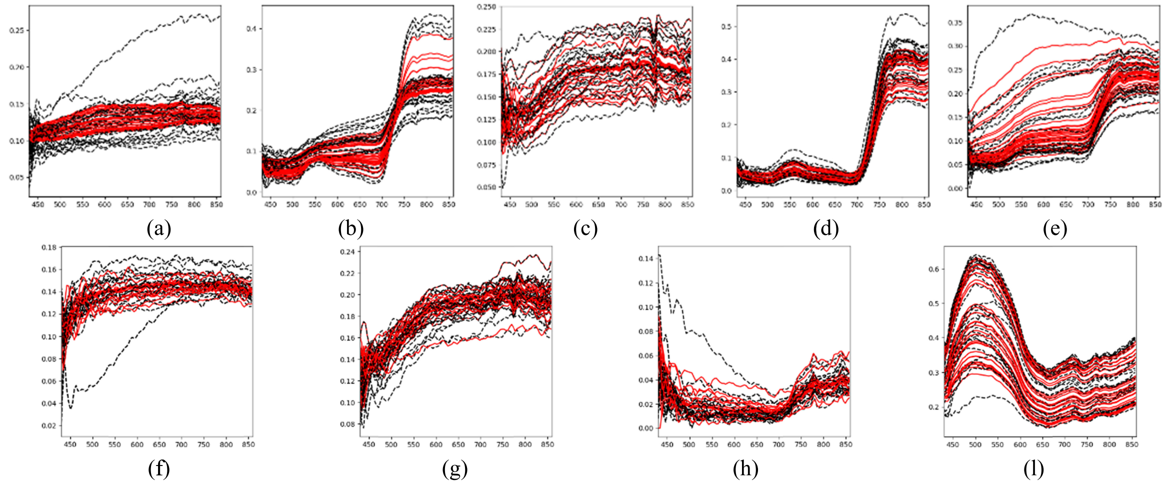
Fig. 6. Pseudocolor composite image and the corresponding ground truth for the Indian Pines AVIRIS data set.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Comparison of the generated “Tree” samples on the ROSIS data set. (a) CVAE. (b) CGAN. (c) CAAE. (d) CVA^{2}E. (e) CVA^{2}E_SAD. (f) CVA^{2}E_FVA. (g) CVA^{2}E_SAD_FVA. (h) Real spectra.



(a) (b) (c) (d) (e)

(f) (g) (h) (l)

Fig. 8. Samples generated by CVA^{2}E_SAD_FVA in all the categories of the ROSIS data (the red solid lines denote the fake spectra and the black dashed lines denote the real spectra). (a) Asphalt. (b) Meadows. (c) Gravel. (d) Trees. (e) Bare soil. (f) Bitumen. (g) Self-blocking bricks. (h) Shadows. (i) Painted metal sheets.

selected from the test data set, so that the generative ability of CVA^{2}E could be verified. We chose the samples from a vegetation-related category in the three data sets to explore the performance of the different methods. For a comparison, CVAE and CGAN were chosen as two representative kinds of deep generative models, and the conditional AAE (CAAE) represents a joint model that consists of a GAN and a VAE. The CAAE is a probabilistic autoencoder that uses the GAN to perform variational inference by matching the aggregated posterior of the hidden code vector with an arbitrary prior distribution. CVA^{2}E was implemented in four different forms: 1) the prototype CVA^{2}E; 2) CVA^{2}E_SAD with the spectral angle penalty term; 3) CVA^{2}E_FVA with the feature angle penalty term; and 4) CVA^{2}E_SAD_FVA with both these penalty terms. In this experiment, the input of all the models was a concatenation of normally distributed sampling and the categorical indicator. For the ROSIS, AVIRIS, and HySpex data sets, the indicator vector was "Tree," "Tree," and "Grass-trees," respectively, which are in the same category. The quality of the generated samples was assessed in terms of the smoothness of the curves, the diversity, and the spectral absorption features.

For the ROSIS data set, Fig. 7 shows that each generative model can generate a spectrum similar to that of vegetation, where the "red edge" was learned well by all seven models. However, the models perform differently in the learning of the detailed feature distribution. The spectrum derived from CVAE is the smoothest. CAAE obtains the worst shape, which appears noisier than the others. CGAN is the second noisiest, and CVA^{2}E with the spectral angle penalty term obtains the most realistic curve. The variation within a category is small in CGAN and CVAE. The generated spectra have a bigger diversity in CAAE, but some spectral structures are lost. The samples from CVA^{2}E_FVA with the feature angle penalty term perform better in diversity than CVA^{2}E. The spectrum of CVA^{2}E_SAD_FVA is clear with well-preserved features. Samples of all the categories can be obtained by CVA^{2}E_SAD_FVA, as shown in Fig. 8.

The HySpex data set has a wider band range than the ROSIS data set, and more spectral absorption characteristics of the vegetation can be explored in the spectrum. It can be observed from Fig. 9 that the spectrum is clear, with well-preserved absorption features, which are consistent with the vegetation biochemical response. These signatures are characterized by absorption with wave troughs around 450, 550, 1450, and 1950 nm, which are captured successfully by the generative model. Some samples are not well controlled by CGAN and CAAE. CVA^{2}E with the spectral angle penalty term shows a poor performance compared with CVA^{2}E without this penalty term. The spectral angle constrains the spectrum fit for the real samples, but with the abundant noise in the HySpex data set, the local noisy bands disturb all the generated samples. In this case, the constraint should be reduced to avoid the impact of the noise. CVA^{2}E_FVA obtains the best performance.

All categories of the HySpex data are obtained by CVA^{2}E_FVA, as shown in Fig. 10. The synthetic spectra can cover almost all the feature space, and have a higher signal-to-noise ratio than the real spectra.

The AVIRIS data set has the widest band range and a lower spectral resolution than HySpex. Fig. 11 shows that the absorption characteristics at 450, 1450, and 1950 nm are successfully captured by CVA^{2}E_FVA, while some features are missing in the results of CGAN and CAAE (e.g., the absorption characteristic at 550 nm).

The spectra derived from CVAE are blurry and show a poor performance in diversity. CAAE obtains the worst shape curves, which appear noisier than the others. CVA^{2}E with the spectral angle penalty term obtains the most realistic curves, as with the ROSIS data set. The samples from CVA^{2}E with the feature angle penalty term perform better in diversity than CVA^{2}E without L_{FVA}. The spectra of CVA^{2}E_SAD_FVA are the clearest, with the well-preserved features showing the strength of the proposed method. When the signal-to-noise ratio of the bands is high, the utilization of L_{SAD} positively improves the quality of the spectra; otherwise, it has a negative effect. All categories of the AVIRIS data set can be obtained by CVA^{2}E_SAD_FVA, as shown in Fig. 5. All categories of the AVIRIS data can also be obtained by CVA^{2}E_SAD_FVA, as shown in Fig. 12.

C. Training on Real and Fake Data Sets

In this section, the validity of the generated samples for the classification task is explored. In order to fairly compare each method, the generated samples and the model are separated, which means that the classification part of the conditional generative model is discarded and replaced by an independent classifier. This has the advantage of making the comparison between samples from all the generative models straightforward, and the classification accuracy directly indicates the validity of the samples. Moreover, the experiments included two scenarios for each data set.

1) Training on the real data set and testing on the generated data. This can describe the separability of the generated data using hyperplane learning from the real data, which indicates whether the generative model has captured the boundaries among all the categories of real data.
2) Training on the generated data set and testing on the real data set. The purpose here is to explore if the features from the generative model support the features of the real data set under a dimensional manifold space.

SVM with a linear kernel was utilized as the universal classifier. In the first experiment, the training data set of the generative model was utilized to train the SVM classifier, and 600 generated samples per class were used in the testing. In the second experiment, 600 generated samples per class were used to train the SVM classifier, and all the labeled data were used in the testing.

Table III shows that all the models obtain a high accuracy. CVAE obtains the highest accuracy with the ROSIS and AVIRIS data sets, and CGAN obtains the highest accuracy with the HySpex data set. The results show that the generated samples from these two models can be separated precisely by the real-data learned boundaries. For example, the learned features have a high correlation with the training data, and the generated samples are similar to the training data. CAAE obtains the lowest accuracy, which is likely caused by the distortion of the spectra. CVA^{2}E obtains a lower accuracy than CVAE on the three data sets and a higher accuracy than CGAN on AVIRIS data set. The samples generated by CVA^{2}E are more diverse compared with the samples generated by CVAE, which results in the higher accuracy of CVAE as given in Table II. Moreover, Figs. 7–12 show that CVA^{2}E outperforms CGAN in the sample generation, the generated sample is more diverse than CVA^{2}E while it

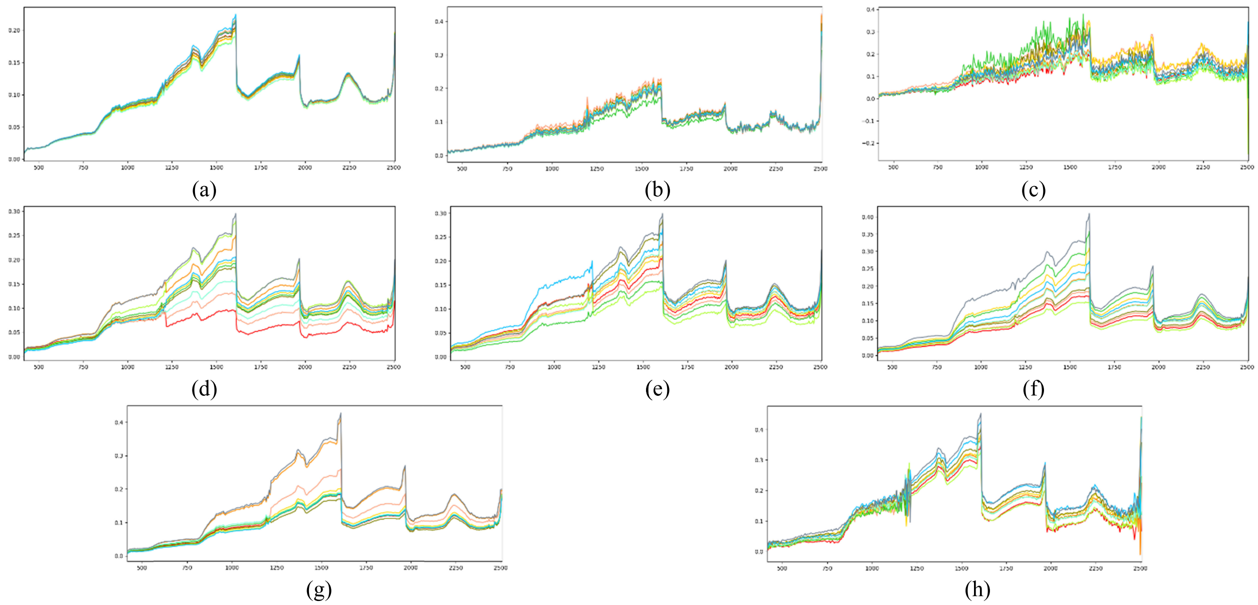


Fig. 9. Comparison of the generated “Tree” samples on the HySpex data set. (a) CVAE. (b) CGAN. (c) CAAE. (d) CVA^{2}E. (e) CVA^{2}E_SAD. (f) CVA^{2}E_FVA. (g) CVA^{2}E_SAD_FVA. (h) Real spectra.

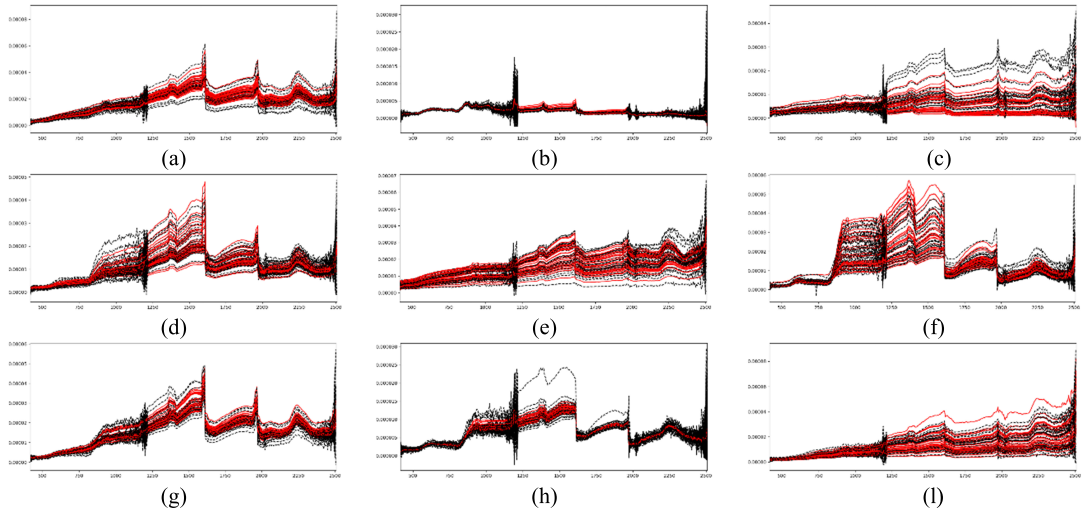


Fig. 10. Samples generated by CVA^{2}E_FVA in all categories of the HySpex data (the red solid lines denote the fake spectra and the black dashed lines denote the real spectra). (a) Bare Land-1. (b) Lakes. (c) Coals. (d) Tree. (e) Cement. (f) Crops-1. (g) Bare Land-2. (h) Crops-2. (i) Red title.

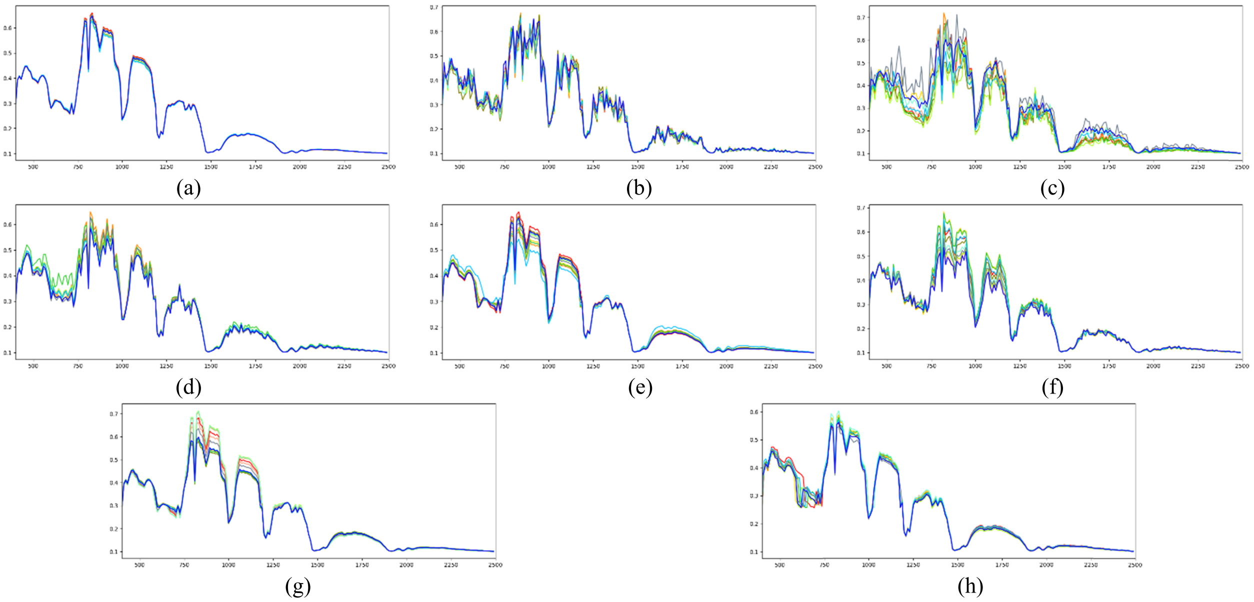
TABLE III

TRAINING ON THE REAL DATA AND TESTING ON THE GENERATED DATA

Algorithm	ROSIS		HySpex		AVIRIS	
	OA	Kappa	OA	Kappa	OA	Kappa
CVAE	**99.03**	**0.984**	99.18	0.9855	**98.91**	**0.9794**
CGAN	98.94	0.9836	**99.73**	**0.9888**	97.36	0.9674
CAAE	96.8	0.9598	98.20	0.9753	95.5	0.9488
CVA^{2}E	98.52	0.9788	99.55	0.9859	98.81	0.9814
CVA^{2}E_SAD	98.96	0.9845	98.23	0.9729	98.33	0.9784
CVA^{2}E_FVA	97.33	0.9652	98.89	0.9826	97.62	0.969
CVA^{2}E_SAD_FVA	98.15	0.9754	98.41	0.9747	97.90	0.9412

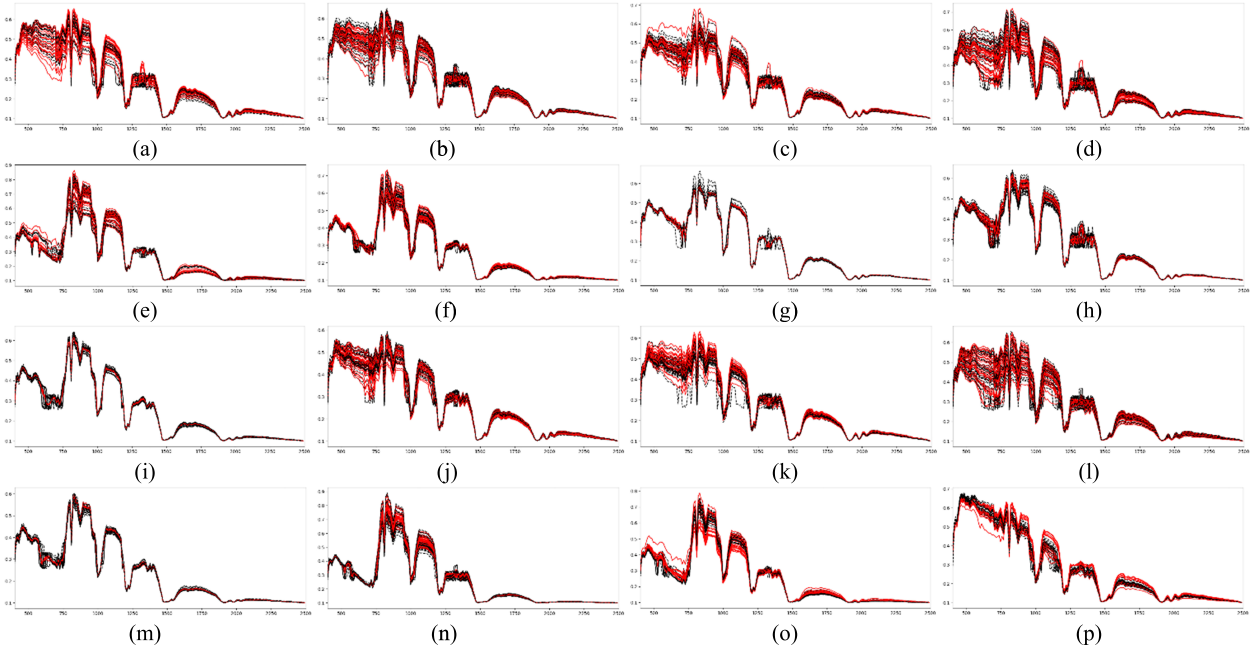
*The best results are shown in bold

is more anamorphic which caused that the difference between CVA^{2}E and CGAN is not obvious in Table II. CVA^{2}E with the feature angle penalty term obtains a lower accuracy than CVA^{2}E without the penalty term on the ROSIS and AVIRIS data sets, where the feature angle penalty term forces the intermediate output of the generator to be different in



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 11. Comparison of the generated "Grass-Tree" samples on the AVIRIS data set. (a) CVAE. (b) CGAN. (c) CAAE. (d) CVA^{2}E. (e) CVA^{2}E_SAD. (f) CVA^{2}E_FVA. (g) CVA^{2}E_SAD_FVA. (h) Real spectra.



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l) (m) (n) (o) (p)

Fig. 12. Samples generated by CVA^{2}E_SAD_FVA in all categories of the AVIRIS data (the red solid lines denote the fake spectra and the black dashed lines denote the real spectra). (a) Alfalfa. (b) Corn-Notill. (c) Corn-Mintill. (d) Corn. (e) Grass-Pasture. (f) Grass–Trees. (g) Grass-Pasture-Mowed. (h) Hay-Windrowed. (i) Oats. (j) Soybean-Notill. (k) Soybean-Mintill. (l) Soybean-Clean. (m) Wheat. (n) Woods. (o) Buildings–Grass–Trees–Drives. (p) Stone–Steel–Towers.

TABLE IV

TRAINING ON THE GENERATED DATA AND TESTING ON THE REAL DATA

Algorithm	ROSIS		HySpex		AVIRIS	
	OA	Kappa	OA	Kappa	OA	Kappa
CVAE	84.37	0.8157	92.9	0.9124	76.2	0.7357
CGAN	85.9	0.8410	91.41	0.8881	75.81	0.7297
CAAE	79.26	0.7761	90.2	0.8854	74.44	0.7190
CVA^{2}E	87.8	0.8597	93.65	0.9152	77.39	0.7578
CVA^{2}E_SAD	88.02	0.853	92.94	0.9054	77.83	0.7661
CVA^{2}E_FVA	88.36	0.8626	**95.14**	**0.9465**	78.6	0.7668
CVA^{2}E_SAD_FVA	**90.79**	**0.8802**	94.3	0.9240	**80.27**	**0.7742**

*The best results are shown in bold

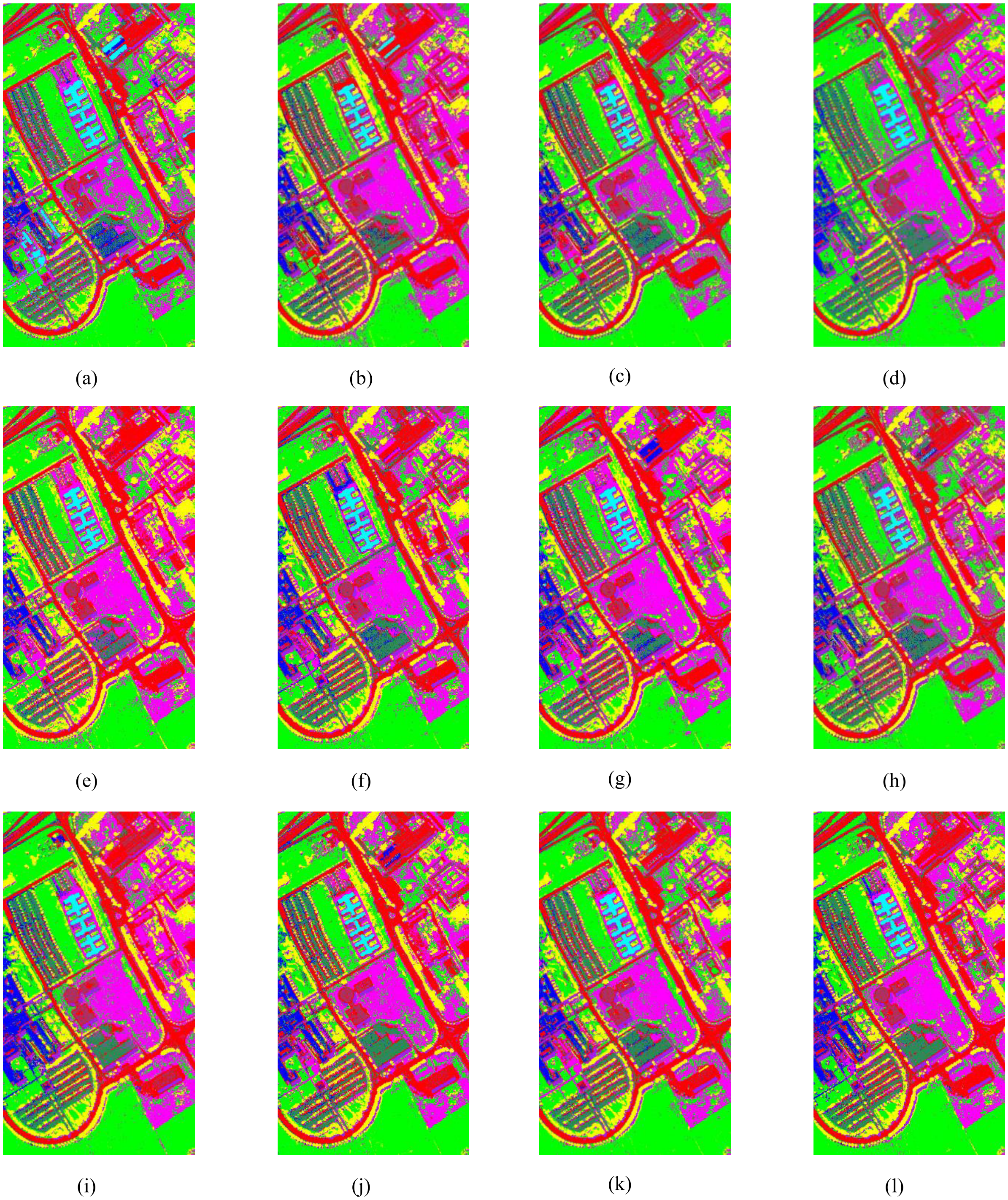
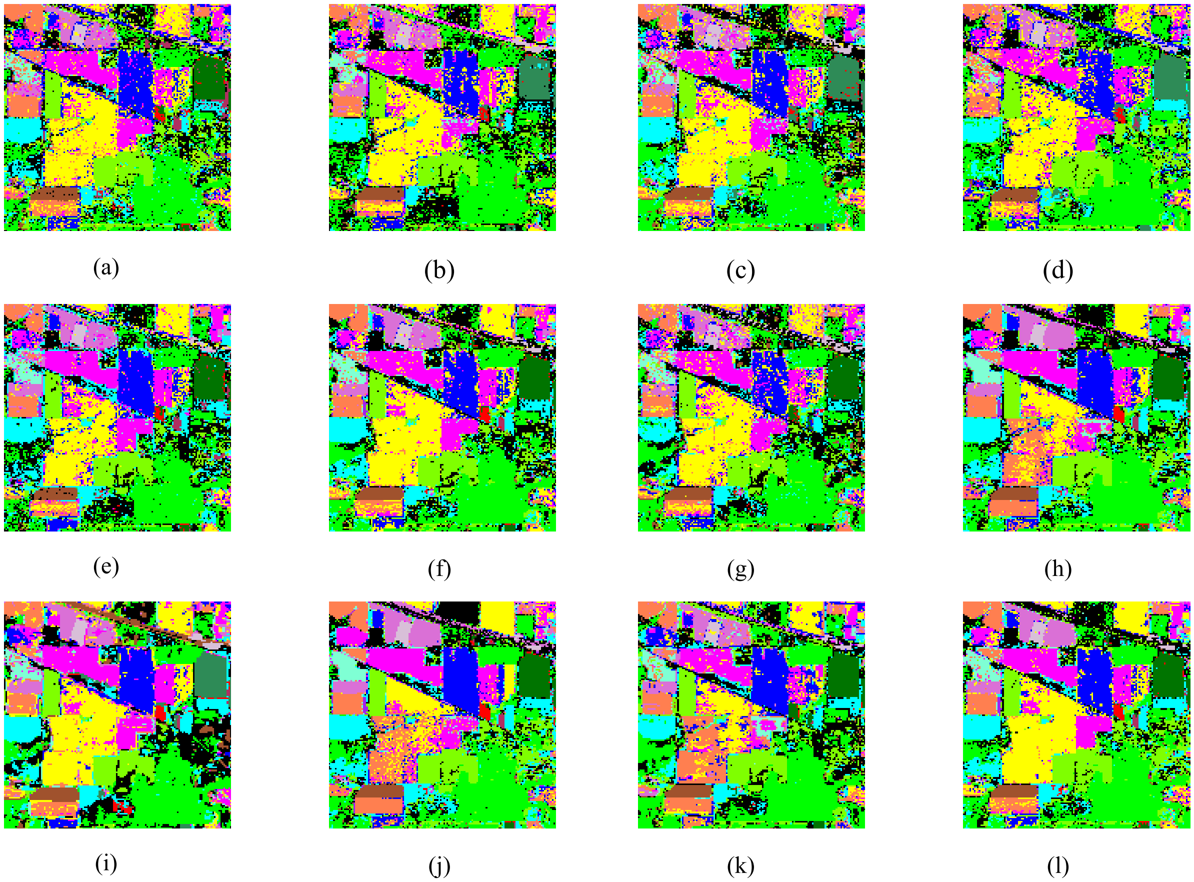


Fig. 13. Results of different approaches for the ROSIS data sets. (a) Classification part of ROSIS. (b) Stacked denoising autoencoder (SDA) of ROSIS. (c) CNN of ROSIS. (d) LSTM of ROSIS. (e) CVAE of ROSIS. (f) CGAN of ROSIS. (g) CAAE of ROSIS. (h) CVA^{2}E of ROSIS. (i) CVA^{2}E_TT of ROSIS. (j) CVA^{2}E_SAD of ROSIS. (k) CVA^{2}E_FVA of ROSIS. (l) CVA^{2}E_SAD_FVA of ROSIS.

a training batch, which is on account of the improvement of the diversity. These diverse samples can be regarded as "distortion samples" for the trained SVM. The difference is that the diverse samples do not coincide with the features of the training data, which is helpful for the training of the classifier, while the samples from CAAE mismatch the spectral features and are thus poor data that degrade the classification performance.

The accuracies of the classifiers trained by the generated samples are given in Table IV. CAAE obtains the worse performance on all three data sets, which verifies that the samples are distorted. CVA^{2}E_SAD_FVA obtains the best results of 90.79% and 80.27% with the ROSIS data set and AVIRIS data set, respectively, which indicates that the support set of generated samples from CVA^{2}E_SAD_FVA is bigger than that of the other models, and is more consistent with the real data



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Fig. 14. Results of different approaches for the AVIRIS data sets. (a) Classification part of AVIRIS. (b) SDA of AVIRIS. (c) CNN of AVIRIS. (d) LSTM of AVIRIS. (e) CVAE of AVIRIS. (f) CGAN of AVIRIS. (g) CAAE of AVIRIS. (h) CVA^{2}E of AVIRIS. (i) CVA^{2}E_TT of AVIRIS. (j) CVA^{2}E_SAD of AVIRIS. (k) CVA^{2}E_FVA of AVIRIS. (l) CVA^{2}E_SAD_FVA of AVIRIS.

TABLE V

TRAINING ON THE JOINT GENERATED AND REAL DATA

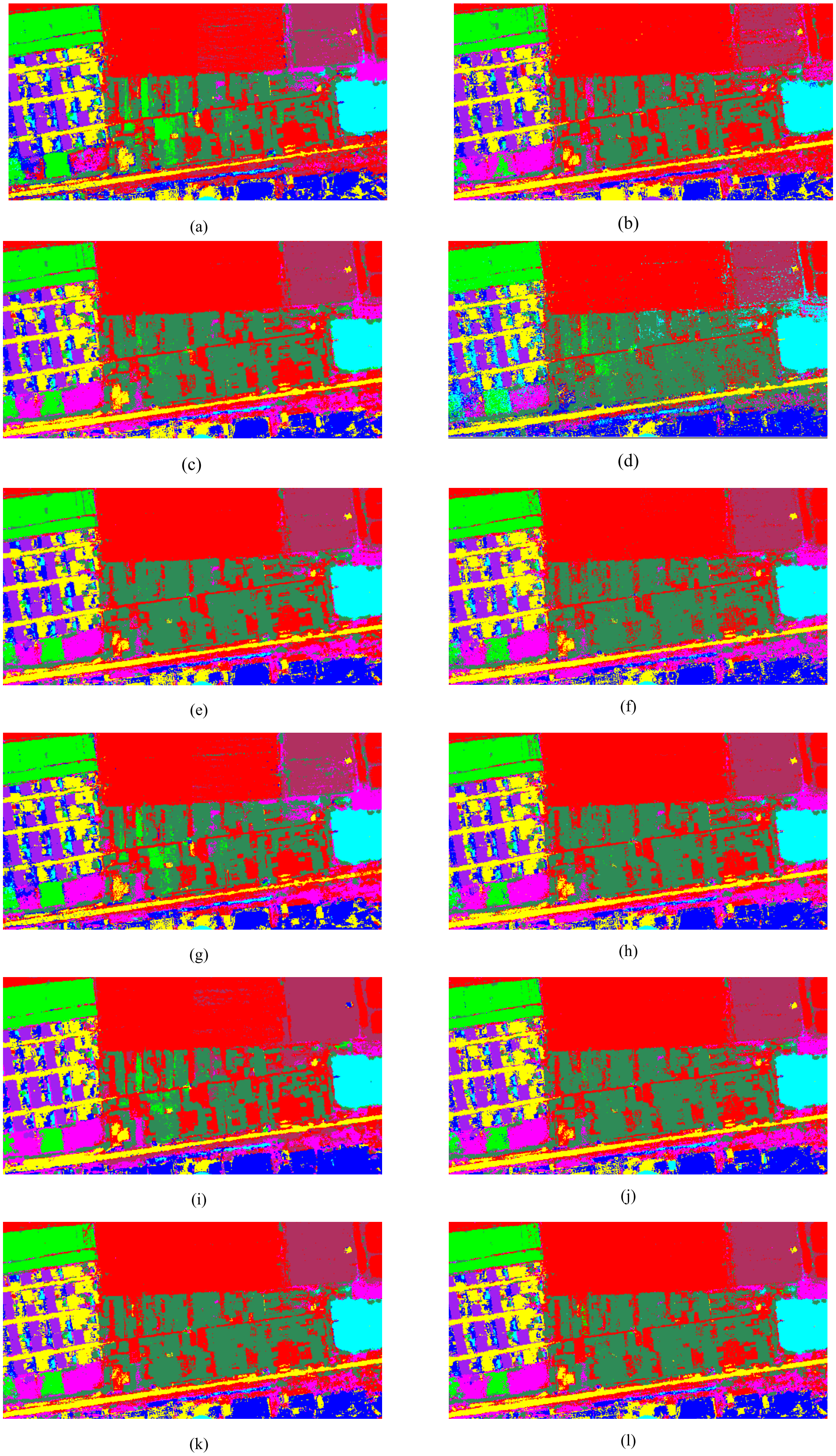
Algorithm	ROSIS		HySpex		AVIRIS	
	OA	Kappa	OA	Kappa	OA	Kappa
Classification part	89.34	0.8564	90.11	0.8755	73.55	0.7045
SDA	92.27	0.9014	96.51	0.9468	82.28	0.7833
CNN	93.29	0.9062	96.68	0.9579	82.39	0.7985
LSTM	88.18	0.8541	93.57	0.9197	79.10	0.7612
CVAE	94.36	0.9219	97.09	0.9515	86.33	0.8358
CGAN	93.79	0.9191	96.82	0.953	85.5	0.8546
CAAE	92.88	0.901	96.02	0.9452	84.56	0.812
CVA^{2}E	94.54	0.9133	97.5	0.9531	86.98	0.8304
CVA^{2}E_TT	93.9	0.9112	96.33	0.9487	86.232	0.8491
CVA^{2}E_SAD	95.1	0.9301	97.81	0.9561	87.63	0.8565
CVA^{2}E_FVA	96.38	0.936	**98.33**	**0.9663**	88.8	0.8721
CVA^{2}E_SAD_FVA	**96.74**	**0.9439**	97.14	0.9452	**89.7**	**0.8803**

*The best results are shown in bold

distribution. CVA^{2}E_FVA obtains a better performance than CVA^{2}E_SAD_FVA by 0.84% on the HySpex data set, which may have been caused by the noise disturbance of the spectra in this data set.

D. Performance Comparison for Classification

Section IV-C discarded the classification part of the model, which is more suitable for the classification task than the traditional SVM because the training process is integral, and the classification part reuses the discriminator network. During the categorical distribution learning process, the network learns the features for classification from the real and generated data. In this section, the classification ability of the generative models is explored. As mentioned above, each model is combined with the classification part. In order to fairly compare each method, the structure was implemented as the same as the discriminator in CVA^{2}E.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

(j)

(k)

(l)

Fig. 15. Results of different approaches for the HySpex data sets. (a) Classification part of HySpex. (b) SDA of HySpex. (c) CNN of HySpex. (d) LSTM of HySpex. (e) CVAE of HySpex. (f) CGAN of HySpex. (g) CAAE of HySpex. (h) CVA^{2}E of HySpex. (i) CVA^{2}E_TT of HySpex. (j) CVA^{2}E_SAD of HySpex. (k) CVA^{2}E_FVA of HySpex. (l) CVA^{2}E_SAD_FVA of HySpex.

In this experiment, the training process of the classification part was divided into two stages. The first was with the variational inference and the adversarial training process, where the classification part was trained with the real data. When the game converged or the training epochs approached a certain number, the output from the generator was regarded as a reliable data pair$(x; y)$. The classification part was then trained by the hybrid of the generated and real data. The mainstream deep learning algorithms such as SDA, LSTM, and CNN were chosen as comparative methods. The "Classification part" was extracted as an independent network, which was trained using the real labeled data to carry out the classification task. Moreover, to verify the valid of augmented samples from CVA^{2}E, we include additional experiment "CVA^{2}E_TT," which trained on the real labeled data without any data augmentation.

Table V shows that CVA^{2}E_SAD_FVA obtains the best results of 96.74% and 89.7% with the ROSIS data set and AVIRIS data set, respectively. CVA^{2}E_FVA obtains the best result of 98.33% with the HySpex data set. HySpex and AVIRIS data sets. LSTM obtains the worst performance on ROSIS data set. CVA^{2}E with two training stages outperforms that without any data augmentation, which demonstrates that CVA^{2}E can provide sufficient generative samples to improve the classification performance. The results of each generative model are better than those based on the traditional SVM classifier as shown in Table III. This proves that CVA^{2}E_SAD_FVA is more effective at excavating efficient features for classification. Figs. 13–15 show the results of classification by each model.

V. Conclusion

In this article, an innovative generative network named CVA^{2}E has been proposed, which combines variational inference and an adversarial training process to obtain a more powerful performance. From the visualization analysis on three standard hyperspectral data sets, we showed that CVA^{2}E can outperform the other methods in its capacity for spectral synthesis. Moreover, to understand the fine-grained spectral distribution characteristics of individual hyperspectral pixels, the spectral angle distance and vectorial angle measurement are introduced in the loss calculation of CVA^{2}E. The improved CVA^{2}E showed a superior performance in the spectral synthesis of different categories. To demonstrate the ability of the generated samples for the classification task, three kinds of scenarios were carried out, and all the results showed that the proposed model gave the best performance.

In our future work, spatial–spectral features will be taken into account in CVA^{2}E. In addition, the denoising ability will also be considered during the generative process for the applications of low signal-to-noise ratio data.

Acknowledgment

The authors would like to thank Prof. D. Landgrebe and Prof. P. Gamba for providing the data used in the experiments.

References

[1] K. Tan, J. Zhu, Q. Du, L. Wu, and P. Du, "A novel tri-training technique for semi-supervised classification of hyperspectral images based on diversity measurement," *Remote Sens.*, vol. 8, no. 9, p. 749, Sep. 2016.

[2] K. Tan, "Hyperspectral remote sensing image classification based on support vector machine," *J. Infr. Millim. Waves*, vol. 27, no. 2, pp. 123–128, Dec. 2008.

[3] D. Ou, K. Tan, Q. Du, J. Zhu, X. Wang, and Y. Chen, "A novel tri-training technique for semi-supervised classification of hyperspectral images based on diversity measurement," *Remote Sens.*, vol. 11, no. 6, p. 654, 2019.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[5] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Netw. (ICANN)*, Edinburgh, U.K., 1999, pp. 850–855.

[6] 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.

[7] 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.

[8] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. https://arxiv.org/abs/1312.6114

[9] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[10] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: https://arxiv.org/abs/1511.05644

[11] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, "CVAE-GAN: Fine-grained image generation through asymmetric training," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2764–2773.

[12] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: https://arxiv.org/abs/1512.09300

[13] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: https://arxiv.org/abs/1701.07875

[14] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Venice, Italy, Oct. 2017, pp. 2794–2802.

[15] 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, Mar. 2019.

[16] M. Zhang, M. Gong, Y. Mao, J. Li, and Y. Wu, "Unsupervised feature extraction in hyperspectral images based on Wasserstein generative adversarial network," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 5, pp. 2669–2688, May 2019.

[17] Y. Duan, X. Tao, M. Xu, C. Han, and J. Lu, "GAN-NL: Unsupervised representation learning for remote sensing image classification," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Anaheim, CA, USA, Nov. 2018, pp. 375-379.

[18] 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.

[19] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, vol. 70, 2017, pp. 2642–2651.

[20] Y. Xu, B. Du, and L. Zhang, "Can we generate good samples for hyperspectral classification?—A generative adversarial network based method," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Valencia, Spain, Jul. 2018, pp. 5752–5755.

[21] D. Wang *et al.*, "Early tomato spotted wilt virus detection using hyperspectral imaging technique and outlier removal auxiliary classifier generative adversarial nets (OR-AC-GAN)," in *Proc. ASABE Annu. Int. Meeting*, Detroit, MI, USA, 2018, p. 1.

[22] N. Audebert, B. L. Saux, and S. Lefèvre, "Generative adversarial networks for realistic synthesis of hyperspectral samples," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Valencia, Spain, Jul. 2018, pp. 4359–4362.

[23] D. Ma, P. Tang, and L. Zhao, "SiftingGAN: Generating and sifting labeled samples to improve the remote sensing image scene classification baseline *in vitro*," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 7, pp. 1046–1050, Jul. 2019.

[24] I. Gemp, I. Durugkar, M. Parente, M. D. Dyar, and S. Mahadevan, “Inverting variational autoencoders for improved generative accuracy,” 2016, *arXiv:1608.05983*. [Online]. Available: https://arxiv.org/abs/1608.05983

[25] M. Gong, X. Niu, P. Zhang, and Z. Li, “Generative adversarial networks for change detection in multispectral imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2310–2314, Dec. 2017.

[26] Y. Su, J. Li, A. Plaza, A. Marinoni, P. Gamba, and S. Chakravortty, “DAEN: Deep autoencoder networks for hyperspectral unmixing,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 7, pp. 4309–4321, Jul. 2019.

[27] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784

[28] F. A. Kruse, “Mapping spectral variability of geologic targets using airborne visible/infrared imaging spectrometer (AVIRIS) data and a combined spectral feature/unmixing approach,” *Proc. SPIE*, vol. 2480, pp. 213–224, Jun. 1995.

[29] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

Xue Wang received the B.S. degree in geographic information system and the Ph.D. degree in photogrammetric and remote sensing from the China University of Mining and Technology, Xuzhou, China, in 2014 and 2019, respectively.

He is currently a Post-Doctoral Researcher with East China Normal University, Shanghai, China. His research interests include the hyperspectral imagery processing, deep learning, and ecological monitoring.

Kun Tan (Senior Member, IEEE) received the B.S. degree in information and computer science from the Hunan Normal University, Changsha, China, in 2004, and the Ph.D. degree in photogrammetric and remote sensing from the China University of Mining and Technology, Xuzhou, China, in 2010.

From September 2008 to September 2009, he was a Joint Ph.D. Candidate of remote sensing with the Columbia University, New York, NY, USA. From 2010 to 2018, he was with the Department of Surveying, Mapping and Geoinformation, China University of Mining and Technology. He is currently a Professor with the East China Normal University, Shanghai, China. His research interests include hyperspectral image (HSI) classification and detection, spectral unmixing, quantitative inversion of land surface parameters, and urban remote sensing.

Qian Du (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Maryland Baltimore County, Baltimore, MD, USA, in 2000.

She is currently the Bobby Shackouls Professor with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS, USA. Her research interests include hyperspectral remote sensing image analysis, pattern recognition, and machine learning.

Dr. Du served as a Co-Chair for the Data Fusion Technical Committee of the IEEE Geoscience and Remote Sensing Society (GRSS) from 2009 to 2013 and the Chair for the Remote Sensing and Mapping Technical Committee of the International Association for Pattern Recognition (IAPR) from 2010 to 2014. She currently serves as the Chief Editor of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.

Yu Chen received the master’s degree in photogrammetry and remote sensing from the China University of Mining and Technology (CUMT), Xuzhou, China, in 2012, and the Ph.D. degree in earth and planetary science from the University of Toulouse, Toulouse, France, in 2017.

She worked as an Assistant Researcher with the Géoscience Environnement Toulouse (GET) Laboratory, French National Center for Scientific Research, Paris, France, in 2013. She is currently a Lecturer with CUMT. Her research interest is synthetic aperture radar (SAR) interferometry, with particular emphasis on its application for geophysical studies.

Peijun Du (Senior Member, IEEE) is currently a Professor of photogrammetry and remote sensing with the Department of Geographic Information Sciences, Nanjing University, Nanjing, China, and also the Deputy Director of the Key Laboratory for Satellite Surveying Technology and Applications, National Administration of Surveying and Geoinformation. His research interests are remote sensing image processing and pattern recognition, remote sensing applications, hyperspectral remote sensing information processing, multisource geospatial information fusion and spatial data handling, integration and applications of geospatial information technologies, and environmental information science.

Prof. Du is an Associate Editor of IEEE Geoscience and Remote Sensing Letters.