



Diffusion Subspace Clustering for Hyperspectral Images

Jiaxin Chen , Shujun Liu , *Student Member, IEEE*, Zhongbiao Zhang, and Huajun Wang 

Abstract—Hyperspectral image (HSI) subspace clustering remains a challenging task due to the poor spatial and rich spectral resolutions of HSIs. Most of the existing HSI subspace clustering approaches just extract the spatial and spectral features, ignoring the intrinsic distribution information of data and leading to low accuracy of clustering generally. To solve this problem, this article presents a diffusion subspace clustering model (DiffSC) that learns distribution information of HSI data simultaneously through a diffusion module (DM). Specifically, due to the diffusion probabilistic model (DPM) learning raw object data distribution to generate data of the same distribution, which has received wide attention in generation tasks and outperforms other generative models significantly, we attempt to apply the DPM in the field of feature extraction. DiffSC performs distribution information extraction of HSIs by the DM and fuses them with spatial-spectral features extracted by deep subspace clustering for training jointly. Experiment outcomes demonstrate that intermediate activation of specific timestep in the inverse diffusion process captures latent distribution information of images effectively and improves the HSI clustering accuracy significantly. Since the DPM is simplified, it can be easily trained from scratch. We evaluate the presented DiffSC on five real HSI datasets, and the experiments indicate that DiffSC can obtain the most advanced clustering outcomes that notably outperform most existing HSI subspace clustering approaches.

Index Terms—Diffusion probabilistic model (DPM), feature extraction, hyperspectral image (HSI), representation learning, subspace clustering (SC).

I. INTRODUCTION

HYPERSPECTRAL images (HSIs) captured by remote sensing satellites and drones have high resolution, rich spectral resources, and large spatial scale information, allowing us to pinpoint areas of interest. HSIs have been used in a variety of applications in recent years, including aerospace, geological surveys, food safety, military, agriculture, and biomedicine [1], [2].

Manuscript received 27 April 2023; revised 30 May 2023 and 21 June 2023; accepted 9 July 2023. Date of publication 12 July 2023; date of current version 21 July 2023. This work was supported by the Key Laboratory of Earth Exploration and Information Techniques of Ministry of Education, Chengdu University of Technology, Chengdu, China. (*Corresponding author: Shujun Liu.*)

Jiaxin Chen is with the School of Computer Science and Cyber Security, Chengdu University of Technology, Chengdu 610059, China (e-mail: jiaxin.chen1@hotmail.com).

Shujun Liu and Zhongbiao Zhang are with the Key Laboratory of Earth Exploration and Information Techniques of Ministry of Education, Chengdu University of Technology, Chengdu 610059, China (e-mail: suldier@outlook.com; fanzhangqaz@sina.com).

Huajun Wang is with the School of Mathematics and Physics, Chengdu University of Technology, Chengdu 610059, China (e-mail: wanghuajun@cdut.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3294623

HSI classification plays a crucial role in the aforementioned HSI application domain, aiming to classify each pixel point with a fixed label, which is the foundation of HSI applications, and the classification method commonly used today is supervised classification based on category labels [3], [4]. In recent years, greater progress has been made in supervised learning-based HSI classification, with excellent classification accuracy obtained on several publicly available HSI datasets, such as Salinas, Pavia University, and Indian Pines images [5], [6], [7], particularly with deep-learning-based models [8], [9], [10], [11], [12], [13], [14], such as convolutional neural networks [15], [16]. However, supervised learning methods, especially deep-learning-based models, usually require large amounts of labeled training samples, but labeling HSI data cost a lot of money and time. In addition, supervised learning-based methods have difficulties in dealing with unknown data as they are modeled based on preexisting labeled data, thus limiting the application of supervised learning-based HSI classification in practice.

To avoid being short of manually labeled data in practical applications, a large number of works have focused on the study of unsupervised HSI classification, called HSI clustering. Instead of using category labels, HSI clustering focuses on the intrinsic connections between individual pixel points and classifies them automatically in an unsupervised approach [17]. The key to HSI clustering is determining the degree of similarity between individual pixel points [18]. Conventional unsupervised approaches, such as K -means clustering [19], use pairwise distances normally, such as Euclidean distance, to measure similarity. Due to the problem of pixel mixing in HSIs [20], the predictions of these methods are often unreliable. High dimensionality and band redundancy of HSIs also pose a great challenge to traditional HSI clustering methods. For unsupervised classification, there is less research related to it and the accuracy of the classification results is not competitive [21], [22].

In recent years, subspace clustering (SC) received more attention in HSI clustering [23], [24], [25], [26], [27], [28] because of its ability to process high-dimensional data and dependable results. SC aims to divide high-dimensional data located in a nonlinear space union into their respective subspaces, each subspace forms a manifold of lower dimensionality. SC has achieved greater success in computer vision, such as face clustering and motion segmentation. SC methods fall into three broad categories [29], including statistical methods, algebraic methods, and spectral-clustering-based methods, where spectral-clustering-based methods have achieved outstanding results [30], [31]. Spectral clustering-based methods include

two stages. First, the data points are assumed to be linearly representable by other points of data in the same subspace, and the affinity matrix is obtained using the self-expressive (SE) coefficient matrix. Then, the affinity matrix is passed into the spectral clustering algorithm to group the data points into various clusters.

For traditional SC methods, sparse subspace clustering (SSC) [32] proposes a sparse SE coefficient matrix, and low-rank representation (LRR) [33] proposes a low-rank SE coefficient matrix, which is constrained by l_1 -norm and l_* -norm (nuclear norm), respectively, to obtain the affinity matrix. Guided by this idea, a large amount of SC approaches have been used to obtain more accurate affinity matrices by employing different constraints. Cai et al. [34] presented an efficient graph convolution-based subspace clustering framework (EGCSC), a new convolutional self-representation of the graph is generated by taking the feature matrix and adjacency matrix of the graph as input, and the Frobenius norm is used to constrain to improve the performance of SC. Vidal et al. [35] constrained the SE coefficient matrix jointly by the nuclear norm and Frobenius norm so that the learned SE coefficient matrix can better increase the intraclass similarity and interclass differential, thus obtaining a more robust SE coefficient matrix.

Although traditional SC has made remarkable achievements in HSI clustering, it is a linear model. Good clustering results cannot be guaranteed for some complex application scenarios, such as dealing with nonlinear problems. With the advancement of deep networks and representation learning, deep-learning-based SC has gained widespread attention. Such algorithms are often named deep clustering models [36], [37]. Deep subspace clustering (DSC) can be used not only for HSI pixel classification but also has a wide range of applications in other areas, such as HSI band selection [38]. To learn the SE coefficient matrix from input data, the deep clustering approaches typically employ an encoder–decoder architecture, which can better extract the deep feature of the data. Because of their nonlinear feature representation capacity, they usually function better than traditional SC approaches, however, they are not extensively used in the field of HSI. Ji et al. [39] first proposed DSC, a deep autoencoder extracting deep features, which improves the SC performance significantly. Lei et al. [40] presented deep spatial-spectral subspace clustering (DS³C) by feature fusion that spatial features and spectral features are extracted at the same time to learn a more robust deep SE coefficient matrix. Cai et al. [41] came up with a graph regularized residual subspace clustering network (GR-RSCNet) by recasting a graph regularized SC model as a special SE layer that is integrated into a deep residual convolutional autoencoder, improving the performance of SC. Wang et al. [42] proposed multiscale DSC with discriminative learning (MDSCDL), which obtains multiscale SE matrixes by learning global and local features, then performs discriminative fusion to obtain the final fusion SE coefficient matrix. In 2022, Cai et al. [43] presented a hypergraph-structured autoencoder (HyperAE) that adopts a deep autoencoder regularized by the hypergraph structure as the backbone network with the capacity of preserving the high-order structured information, which outperforms many existing methods of DSC. Li et al. [44] presented a deep mutual information subspace clustering (DMISC)

network that the convolutional autoencoder is used to learn the self-expression coefficient matrix through mutual information maximization, residual connection, and graph regularization, which effectively improves the SC accuracy, but lacks robustness on some datasets. Although these deep learning methods have good clustering performance, however, they just extract the spatial and spectral features and ignore the intrinsic features, such as the data distribution information.

To extract the data distribution information to further improve SC accuracy and obtain robust deep affinity matrices, a diffusion subspace clustering (DiffSC) method is proposed in this article. While the diffusion probabilistic model (DPM) [45], [46] has yielded astounding results on a variety of generative tasks through learning the raw data distribution [47], [48], [49], [50], it is unclear whether the DPM can also be a learner of feature representations. In this article, we make use of a noise prediction network in the DPM to perform distribution information feature learning on the raw HSI data, and then, learn the SE coefficient matrix together with the spatial-spectral features of the encoder output of DSC jointly. Results show that it is feasible for the DPM to be used for representation learning in HSI SC.

To sum up, the main contributions of this article are as follows.

- 1) We investigate the most advanced DPM learning representation and the experiment indicates that it captures HSI data distribution information, which is effective for HSI SC.
- 2) We design the DiffSC for the HSI DSC, which is jointly trained using data distribution and spatial-spectral features. As far as we know, this is the first work using DPM representation learning to HSI SC.
- 3) We compare the DiffSC with other existing traditional or DSC methods on five real HSI datasets, and the outcome of the experiments shows that DiffSC is superior to many existing HSI SC methods.

II. RELATED WORKS

In this section, we describe existing research directions relevant to our research.

A. Clustering Methods

1) *Subspace Clustering (SC)*: SC supposes that all high-dimensional space data are sampled from the union of multiple low-dimensional subspaces, in other words, that is low-dimensional manifolds embedding in a high-dimensional space so it can cluster by dividing the subspace. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M]^T \in \mathbb{R}^{M \times d}$ is the set of M data points, which $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^M$ come from the sum $\mathcal{S} = \{\cup \mathcal{S}_j\}_{j=1}^n$ of low-dimensional subspaces, where M , d , and n indicate numbers of raw data points, original space dimension, and quantity of subspaces, respectively. Each subspace dimension is $\{d_k\}_{k=1}^n$. If the set $\mathbf{X}_t \subseteq \mathbf{X}$ of t points belong to the same subspace \mathcal{S}_k ($1 \leq k \leq n$) of dimension d_k , this d_k -dimensional manifold can be considered to be embedded in a d -dimensional Euclidean space ($d_k < d$), and thus, can be clustered. SSC [32] and LRR [33] are pioneering work in this area, finding sparse SE coefficient matrices and low-rank SE coefficient matrices by means of l_1 -norm and l_* -norm constraints, respectively, which

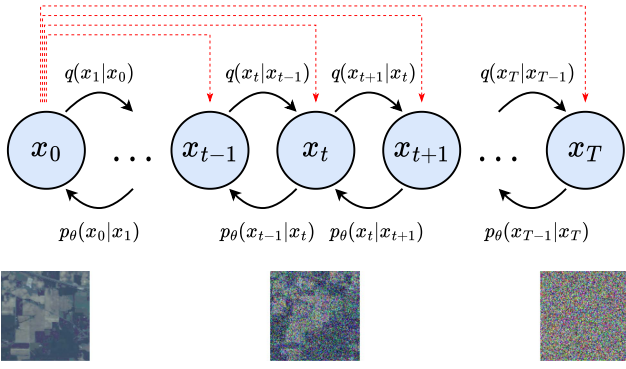


Fig. 1. Directed graph structure of the DPM.

define the following objective to optimize the problem

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 \quad \text{s.t. } \mathbf{C}\mathbf{X} = \mathbf{X}, \quad \text{diag}(\mathbf{C}) = 0 \quad (1)$$

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* \quad \text{s.t. } \mathbf{C}\mathbf{X} = \mathbf{X}, \quad \text{diag}(\mathbf{C}) = 0 \quad (2)$$

respectively, where the SE coefficient matrix $\mathbf{C} \in \mathbb{R}^{M \times M}$ and $c_{i,j}$ in \mathbf{C} denotes the correlation between the i th point x_i and the j th point x_j , which tends to be higher if the two data points belong to identical subspace, and also set $\text{diag}(\mathbf{C}) = 0$ to avoid obtaining a trivial solution. $\|\mathbf{C}\|_1$ indicates the l_1 -norm and $\|\mathbf{C}\|_*$ denotes nuclear norm. Usually, \mathbf{C} is used for the construction of the affinity matrix by $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}^T|$ and passes \mathbf{A} into the spectral clustering [25] approach to obtain the ultimate clustering results. Because the method that passes the two constraints of (1) and (2) is still a shallow SC method, different from DSC, the clustering performance of the shallow method is usually ordinary.

2) *Spectral Clustering*: Spectral clustering is an algorithm that evolved from the graph theory and has been widely used in clustering. Its main idea is to consider all data as points in space, and these points can be connected with edges between them. The edge weight between two points farther away is low, while closer together is high. The purpose of clustering is achieved by cutting the graph composed of all data points so that the edge weight sum between different subgraphs after the cut is as low as possible, while the edge weight sum within the subgraph is as high as possible.

In contrast to conventional clustering algorithms, spectral clustering has the benefit of being able to cluster on a sample space of any form and converge to a globally optimum solution.

B. Diffusion Probabilistic Model (DPM)

DPMs are a class of hidden variable models for generative tasks that approximate complex original data distributions through simple distributions, typically standard Gaussian distributions. The DPM excels at many generative tasks, to generate samples that are in the same distribution as the original data. It can be divided into two processes, as shown in Fig. 1, including diffusion processes and inverse diffusion processes, both of these processes can be viewed as parametric Markov chains. The diffusion process involves adding different scale Gaussian noise gradually to a sample of original data x_0 until it becomes random

noise

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I) \quad (3)$$

$$q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1}) \quad (4)$$

where variance schedule $\beta_1 < \beta_2 \dots < \beta_t < \dots < \beta_T < 1$.

To avoid iteration, the noise sample x_t at any moment t can be computed directly using x_0 and β_t by means of the reparameterization trick

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1 - \bar{\alpha}_t)I) \quad (5)$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \quad (6)$$

where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$. T is the total number of iterations. When $t \rightarrow T$, $x_t \sim \mathcal{N}(0, I)$.

While the inverse diffusion process is the recovery from random Gaussian noise to the original distribution data gradually, it can construct a parameterized model p_θ by deep neural networks to approximate this distribution, where joint distribution and conditional probability distribution of the inverse diffusion process are

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t) \quad (7)$$

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \quad (8)$$

where $\Sigma_\theta(x_t, t) = \sigma_t^2 I = \beta_t I$.

We model p_θ according to the posterior conditional probability q of the diffusion process. It is possible to obtain the variational lower bound (VLB) as the minimization optimization objective by variational inference as follows:

$$L_{\text{VLB}} := \mathbb{E}_{q(x_{0:T})} \left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})} \right]. \quad (9)$$

During the inverse diffusion process, we need to iterate to obtain x_{t-1} from x_t , thus the joint probability of the inverse diffusion process $p_\theta(x_{0:T})$ in (9) is not easy to solve. Considering this factor, we transform it into the conditional probability, which is easy to solve as follows

$$\begin{aligned} L_{\text{VLB}} = \mathbb{E}_q \left[\underbrace{D_{KL}(q(x_t|x_0)||p(x_T))}_{L_T} \right. \\ \left. + \sum_{t=2}^T \underbrace{D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \right. \\ \left. - \underbrace{\log p_\theta(x_0|x_1)}_{L_0} \right]. \quad (10) \end{aligned}$$

The optimization function after simplification with a total of $T + 1$ terms. L_T is a constant that can be omitted, and L_0 is solved by a separate discrete decoder that is not involved in network training. The remaining $T - 1$ terms, which is the middle L_{t-1} in the aforementioned equation, need to be optimized. Since q

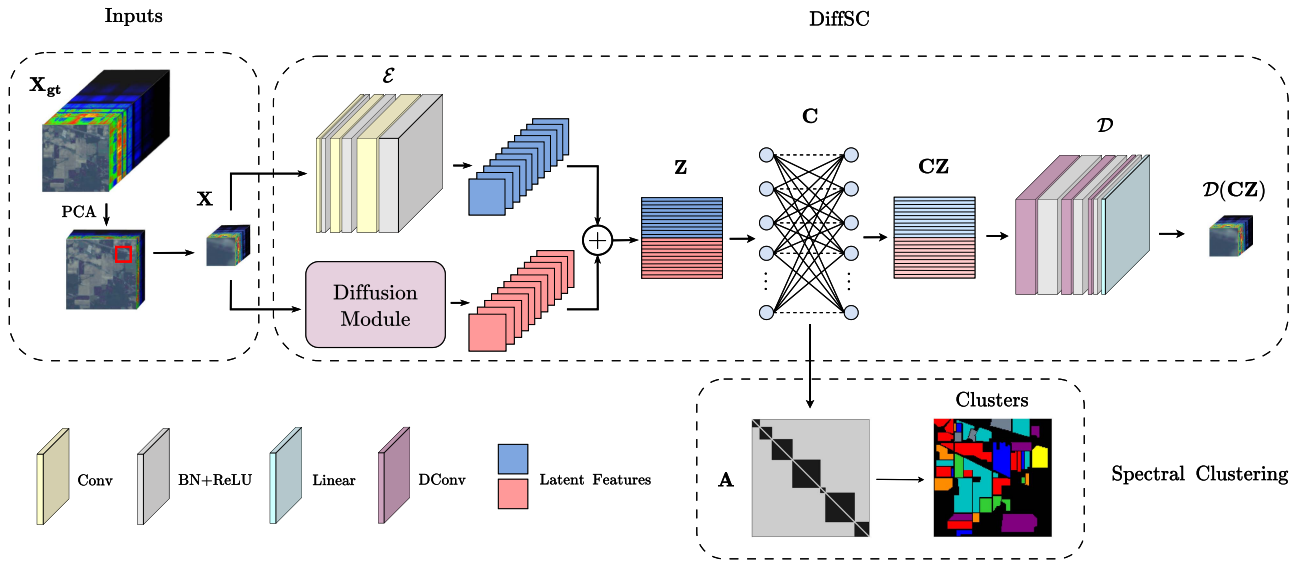


Fig. 2. Schematic diagram of the structure of our proposed DiffSC, including the Encoder, Decoder, and DM. Specifically, DiffSC includes two stages, the first stage learning the SE coefficient matrix, and the second stage, using spectral clustering methods to produce clustering results.

and p_θ can be viewed as two Gaussian distributions with no parameters in q and only the mean in p_θ contain parameters, the L_{T-1} in (10) is transformed by the KL divergence formula for two univariate Gaussian distributions as

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] + C \quad (11)$$

where $\tilde{\mu}_t$ is the mean of q , and μ_θ is the mean of p_θ .

Due to modeling the predicted mean μ_θ directly is not the best choice, we transfer the parameters in mean to added noise ϵ_θ by reparameterization trick, and then, directly predict the noise by modeling D_θ . Substitute this into (11) and omit the fixed weights yields the most simplified optimization objective

$$L_{\text{simple}}(\theta) := \mathbb{E}_{x_0, t, \epsilon} [\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}\epsilon, t)\|^2]. \quad (12)$$

The inverse diffusion process trains the noise prediction network D_θ by predicting the noise added at each step of the diffusion process.

Recent work [49] has developed different generative targets and better model architectures, allowing DPMs to outperform GANs [51] in quality and diversity of generative tasks. DPMs have already been widely used in several fields, including speech synthesis [52], molecular conformation generation [53], and video generation [54]. In our work, the experiment outcomes indicate that DPMs can also be used for representation learning.

III. PROPOSED METHOD

In this section, we investigate representation learning by DPMs, first giving a brief introduction to the DiffSC framework, and then, using it in HSI SC to improve the clustering performance.

A. Diffusion Subspace Clustering Model (DiffSC)

DPMs excel in generation tasks by learning the distribution information of the data, so we try to use it for distribution information extraction. In the generation task, the inverse diffusion process trains the noise prediction network by randomly selecting time step t and iterates step by step to generate the target distribution data finally. Since we only use it for extracting data distribution features and do not need the whole iterative process, we pick a few fixed-time steps t for training.

To extract the distribution information of the HSI data, we use a simplified DPM for joint training, which does not use the inverse diffusion generation process. The original data distribution information is obtained by the intermediate output of the noise prediction network D_θ at a specific time step t . Specifically, the complete framework of DiffSC is shown in Fig. 2, including a diffusion module (DM), encoder with three convolutional layers, and decoder with three deconvolutional layers. We use (12) in the DPM as the loss function of the noise prediction network D_θ in DM, as shown in Fig. 3, and train D_θ by minimizing the loss function. To train DiffSC, we define joint loss function as

$$\mathcal{L} = \|\epsilon_t - \hat{\epsilon}\|_F^2 + \lambda \|\mathbf{Z} - \mathbf{CZ}\|_F^2 + \gamma \|\mathbf{C}\|_2^2 + \|\mathbf{X} - \mathcal{D}(\mathbf{CZ})\|_F^2 \quad (13)$$

$$\mathbf{Z} = \text{concat}(\mathcal{E}(\mathbf{X}), \text{DM}(\mathbf{X}))$$

where λ and γ are hyperparameters; $\|\cdot\|_F$ represents the Frobenius norm; and $\|\cdot\|_2$ represents l_2 -norm.

$\mathcal{E}(\cdot)$, $\text{DM}(\cdot)$, and $\mathcal{D}(\cdot)$ denote the output of Encoder, DM, and Decoder, respectively. The joint loss function \mathcal{L} consists of four parts. $\|\epsilon_t - \hat{\epsilon}\|_F^2$ represents the loss of D_θ network that ϵ_t is the Gaussian noise added to data at time step t and $\hat{\epsilon}$ is noise predicted by D_θ , allowing the model to learn more efficient data distribution features. $\|\mathbf{Z} - \mathbf{CZ}\|_F^2$ represents the loss of the subspace, where \mathbf{Z} is a fusion of spatial-spectral features and distribution information features that can be used together to learn the SE coefficient matrix \mathbf{C} . $\|\mathbf{C}\|_2^2$ denotes the constraint

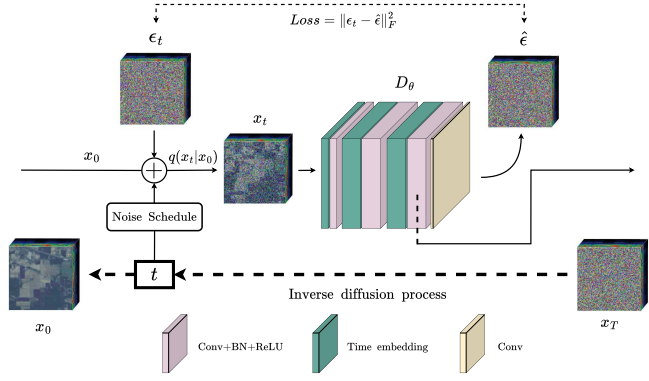


Fig. 3. DM structure. The noise added at timestep t of the diffusion process is predicted by the timestep t and x_t .

to obtain the sparse SE coefficient matrix. $\|\mathbf{X} - \mathcal{D}(\mathbf{CZ})\|_F^2$ represents the loss of the original data and reconstructed data.

B. DiffSC for HSI Clustering

HSI clustering includes two stages, of which the first stage learns the SE coefficient matrix, and the second stage uses spectral clustering methods to obtain clustering results.

First, we process the raw HSI data to match the input requirements of the model. HSI data are a data cube that combines spatial and spectral dimensions, which we denote as $\mathbf{X} \in \mathbb{R}^{H \times W \times B}$, where H denotes height, W denotes width, and B denotes spectral channel. Since HSI generally has hundreds of spectral channels and many redundancies, in order to eliminate redundancies, PCA is first used to reduce the dimensionality. If the number of channels is D after the reduction, the data are represented as $\mathbf{X} \in \mathbb{R}^{H \times W \times D}$. If only a single pixel point is utilized for training, the training data become $\mathbf{X} \in \mathbb{R}^{M \times 1 \times 1 \times D}$, and M denotes the total number of pixel points. Then, when using the network for training, we can only extract the spectral feature information and lose the rich spatial feature information contained in the HSI. If we want to extract both spectral features and spatial features in training, we can consider each pixel point as the center and the surrounding $S \times S$ neighborhood as the spatial information of that pixel point together as the training data, then the training data will be expressed as $\mathbf{X} \in \mathbb{R}^{M \times S \times S \times D}$. The label of the center pixel point is used as the label of each pixel data cube.

For data points at boundary locations in the original HSI, which cannot take full advantage of the neighborhood information, we can make them also acquire sufficient spatial features by mirror filling. To ensure that each pixel point is the absolute centroid of the $S \times S$ neighborhood, S is usually odd, so the mirror fill size is $(S-1)/2$ around the original data before dividing the neighborhood patches.

The divided HSI patches are used as the input of DiffSC to extract the spatial-spectral features and distribution information features, assuming that the obtained feature map $\mathbf{Z} \in \mathbb{R}^{M \times S \times S \times T}$. To get the SE coefficient matrix $\mathbf{C} \in \mathbb{R}^{M \times M}$, we reshape the dimension of feature map to $\mathbf{Z} \in \mathbb{R}^{M \times (S \times S \times T)}$ so

Algorithm 1: Pseudocode of DiffSC for HSI Clustering.

Input: HSI \mathbf{X}_{gt} ; cluster number n ; hyperparameters λ, γ
Output: clustering results

- 1: Generate low-dimensional and small-scale HSI data \mathbf{X}
- 2: Initialize Diffusion module (DM), encoder (\mathcal{E}), decoder (\mathcal{D}), and SE coefficient matrix \mathbf{C}
- 3: **While** stopping conditions are not satisfied **do**
- 4: $\mathbf{Z} = \text{concat}(\mathcal{E}(\mathbf{X}), \text{DM}(\mathbf{X}))$
- 5: Calculate \mathcal{L} according to (13)
- 6: Optimization by gradient descent algorithm
- 7: **endwhile**
- 8: Calculate the deep affinity matrix $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}^T|$
- 9: Clustering results obtained with spectral clustering

TABLE I
SUMMARY OF SALINAS, PAVIA UNIVERSITY, INDIAN PINES, HONGHU, AND LONGKOU DATASETS

| Datasets | Pixels | Channels | Samples | Sensor |
|------------------|---------|----------|---------|----------------|
| SalinasA | 83×86 | 204 | 5348 | AVIRIS |
| Pavia University | 200×100 | 103 | 6445 | ROSIS |
| Indian Pines | 85×70 | 200 | 4391 | AVIRIS |
| HongHu | 100×80 | 270 | 6527 | Nano-Hyperspec |
| LongKou | 110×67 | 270 | 5954 | Nano-Hyperspec |

that we can learn the SE coefficient matrix \mathbf{C} by $\mathbf{Z} = \mathbf{CZ}$, then get the affinity matrix by $\mathbf{A} = |\mathbf{C}| + |\mathbf{C}^T|$, and finally, get the clustering results by spectral clustering.

IV. EXPERIMENTS AND RESULTS

In this section, we assess the clustering capabilities of DiffSC on five commonly used HSI datasets and perform extensive ablation studies to analyze the proposed DiffSC in many aspects.

A. Configuration

1) *Datasets and Preprocessing:* We conducted extensive experiments on five commonly used HSI datasets, namely Salinas, Pavia University, and Indian Pines, and two recent Chinese HSI datasets called WHU-Hi-HongHu and WHU-Hi-LongKou [55]. The WHU-Hi-HongHu dataset was acquired in HongHu, Hubei province, in 2017, and the WHU-Hi-LongKou dataset was acquired in LongKou, Hubei province, in 2018. The Salinas and Indian Pines datasets were acquired through AVIRIS sensors, the Pavia University dataset was obtained through ROSIS sensors, and WHU-Hi-HongHu and WHU-Hi-LongKou were got with Nano-Hyperspec sensors. To improve clustering performance, we only intercepted some scenes from these datasets. Specifically, these scenes are located at [158:240,591:676], [150:350,100:200], [30:115,24:94], [800:900,90:170], and [110:220,208:275], respectively, and the scenes intercepted from Salinas, WHU-Hi-HongHu, and WHU-Hi-LongKou are also called the SalinasA, HongHu, and LongKou dataset. Table I summarizes the specific information of the five datasets.

TABLE II
CONFIGURATION OF THE DIFFSC FOR THE PAVIA UNIVERSITY DATASET

| Component | Layer | Kernel | Activation |
|-----------------------|--------------|-------------------------------|------------|
| Diffusion module (DM) | T Embedding1 | neighbor \times neighbor | BN+ReLU |
| | Conv1-2d | $3\times 3\times b\times 16$ | |
| | T Embedding2 | neighbor \times neighbor | BN+ReLU |
| | Conv2-2d | $3\times 3\times 16\times 32$ | |
| | T Embedding3 | neighbor \times neighbor | BN+ReLU |
| | Conv3-2d | $3\times 3\times 32\times 32$ | |
| Encoder | Conv4-2d | $1\times 1\times 32\times b$ | Linear |
| | Conv5-2d | $3\times 3\times b\times 24$ | BN+ReLU |
| | Conv6-2d | $3\times 3\times 24\times 24$ | BN+ReLU |
| | Conv7-2d | $3\times 3\times 24\times 32$ | BN+ReLU |
| Decoder | DConv3-2d | $3\times 3\times 64\times 32$ | BN+ReLU |
| | DConv2-2d | $3\times 3\times 32\times 24$ | BN+ReLU |
| | DConv1-2d | $3\times 3\times 24\times 24$ | BN+ReLU |
| Output | Conv8-2d | $1\times 1\times 24\times b$ | Linear |

In the preprocessing stage, we downscale the channels of the HSI to a suitable size by PCA and select an appropriate neighborhood size to generate training data. We set the neighborhood size of SalinasA, Pavia University, Indian Pines, HongHu, and LongKou to 9, 13, 13, 11, and 9, respectively; the channel sizes were set to 4, 10, 6, 5, and 5, respectively; and the time step t was set to 5, 9, 9, 7, and 11, respectively. All sample eigenvalues were normalized to $[0,1]$ before clustering.

2) *Evaluation Indices*: Three indices [22], [34] are most common for evaluating clustering capability: overall accuracy (OA), normalized mutual information (NMI), and kappa coefficient (κ). The OA index directly calculates the accuracy according to the best alignment of the predicted clusters and ground-truth labels. The three indices take values range $[0, 1]$, with larger values indicating better clustering.

3) *Baselines and Comparative Approaches*: We compared the DiffSC with many existing clustering approaches, including four traditional clustering approaches and six popular deep clustering approaches. For the comparison experiments, the traditional methods include sparse SSC, LRR, low-rank subspace clustering (LRSC), and EGCSC, and the deep learning methods include DS³C, GR-RSCNet, MDSCDL, HyperAE, and DMISC network. Among them, HyperAE is the more effective HSI clustering method at present. DSC is compared as a baseline.

4) *Realization Details*: For the aforementioned HSI clustering approaches, i.e., LRR, SSC, LRSC, EGCSC, DSC, DS³C, GR-RSCNet, DMISC, and HyperAE, we use the settings in the corresponding articles. For MDSCDL, we choose one suitable parameter configuration. We set T in the DM to 1000 and the β range to $[0.0001, 0.02]$. Taking the Pavia University dataset for an example, the detailed configuration of DiffSC is shown in Table II. We adopt Adam optimizer for training DiffSC, setting the learning rate to 0.0002 and time step t to 9 and passing it into the network through the T embedding layer. The epochs of SalinasA, Pavia University, Indian Pines, HongHu, and LongKou were all set to 100. The hyperparameters in joint loss function \mathcal{L} were set $\lambda = 100$ and $\gamma = 0.1$. All baseline approaches are implemented on an Intel Xeon Gold 5218 2.30-GHz CPU with 128-GB RAM with Python 3.7. For speeding up DiffSC training, we run it on an NVIDIA GeForce RTX 2080 Ti GPU with 11 GB of graphic memory.

B. Results

1) *Quantitative Results*: Tables III–VII show clustering results of various approaches on SalinasA, Pavia University, Indian Pines, HongHu, and LongKou datasets. The results indicate that DiffSC presented in this article gets the best clustering performance and is obviously better than other clustering approaches in OA, NMI, and κ metrics. From the results, it is evident that the traditional clustering methods are notably improved when combined with deep learning. DiffSC significantly outperforms the SSC and LRR traditional methods, which indicates that conventional clustering approaches combined with deep representation learning have greatly improved clustering capabilities. From Tables III to VII, it can be seen that few methods are able to achieve 93% OA on five different datasets at the same time, while DiffSC obtains an OA above 93% for all of them. In particular, DiffSC achieves perfect (100% OA) clustering capabilities on the SalinasA and Pavia University datasets. DiffSC outperforms DSC on all datasets, indicating that the capabilities of DSC can be notably enhanced by adding diffusion feature representation, which means that the SE coefficient matrix C learned by DiffSC is better than that learned by DSC. In the later ablation experiments, we will further investigate the effectiveness of the diffusion feature representation and demonstrate that multifeature fusion is the key to HSI clustering. It is seen that almost all clustering approaches with higher performance try to fuse multiple features, for example, EGCSC incorporates graph embedding information during training. Therefore, the methods after fusion by features usually work better than traditional methods.

The results of DiffSC are comparable to many supervised methods, obtaining 100%, 100%, 97.43%, 99.64%, and 93.68% OA on SalinasA, Pavia University, Indian Pines, HongHu, and LongKou datasets, respectively. In general, supervised methods are trained with labeling information, so results are better than unsupervised approaches. But the approach presented in this article builds an effective HSI clustering model competition with the supervised HSI classification approaches.

2) *Comparison of Visualization Results of Various Methods*: We visualize clustering results of various approaches as illustrated in Figs. 4–8. Usually, the same color represents the same object category, but some of the predicted category labels may have slight differences due to the presence of misclassification cases. As shown in Figs. 4 and 5, the clustering outcomes acquired through our proposed DiffSC on the SalinasA and Pavia University datasets are completely consistent with the ground truth without any clustering errors, while other clustering models cannot achieve 100% accuracy on both datasets at the same time. For Indian Pines (see Fig. 6), HongHu (see Fig. 7), and LongKou (see Fig. 8) datasets, DiffSC obtained the closest clustering effect to the ground truth compared with other clustering models, while the results clustered with other methods either had a higher number of misclassifications or a higher number of noise points, and the clustering effect was significantly weaker than DiffSC. The clustering outcomes indicate that our approach robustness and has high accuracy, moreover, the clustering effect is significantly better than most of the existing clustering models.

TABLE III
CLUSTERING RESULTS OF LRR, SSC, LRSC, EGCSC, DSC, DS³C, GR-RSCNet, MDSCDL, HYPERAE, DMISC, AND DIFFSC ON THE SALINASA DATASET

| Class | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|------------------------------|-------|-------|-------|------------|-------|-------------------|------------|--------|------------|------------|------------|
| Broccoli_green_weeds_1(%) | 100 | 0.00 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Lettuce_romaine_4wk(%) | 79.38 | 62.62 | 97.69 | 100 | 100 | 96.43 | 100 | 98.70 | 100 | 100 | 100 |
| Lettuce_romaine_5wk(%) | 91.02 | 99.84 | 2.60 | 100 | 100 | 97.31 | 100 | 100 | 100 | 100 | 100 |
| Lettuce_romaine_6wk(%) | 69.73 | 75.74 | 99.08 | 100 | 91.74 | 100 | 100 | 100 | 100 | 100 | 100 |
| Lettuce_romaine_7wk(%) | 86.98 | 98.96 | 89.61 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Corn_senesced_green_weeds(%) | 71.56 | 100 | 97.87 | 100 | 100 | 90.99 | 100 | 99.92 | 100 | 100 | 100 |
| OA(%) | 82.16 | 76.23 | 86.31 | 100 | 97.64 | 96.56 | 100 | 99.83 | 100 | 100 | 100 |
| NMI(%) | 71.02 | 82.72 | 82.02 | 100 | 95.75 | 92.84 | 100 | 99.40 | 100 | 100 | 100 |
| κ (%) | 77.75 | 71.14 | 82.48 | 100 | 97.07 | 95.71 | 100 | 99.79 | 100 | 100 | 100 |

The bold values represent the highest values of the three evaluation indicators among all experimental methods.

TABLE IV
CLUSTERING RESULTS OF LRR, SSC, LRSC, EGCSC, DSC, DS³C, GR-RSCNet, MDSCDL, HYPERAE, DMISC, AND DIFFSC ON THE PAVIA UNIVERSITY DATASET

| Class | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|-------------------------|-------|-------|-------|-------|-------|-------------------|-----------|--------|---------|-------|------------|
| Painted metal sheets(%) | 100 | 100 | 98.86 | 100 | 100 | 99.92 | 100 | 100 | 100 | 99.77 | 100 |
| Asphalt(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.24 | 0.00 | 100 | 98.35 | 0.05 | 96.00 | 100 |
| Trees(%) | 77.78 | 0.00 | 0.00 | 4.76 | 100 | 87.30 | 100 | 100 | 100 | 0.00 | 100 |
| Meadows(%) | 62.76 | 61.07 | 89.84 | 100 | 100 | 100 | 100 | 99.74 | 100 | 100 | 100 |
| Self-Blocking Bricks(%) | 1.06 | 67.02 | 0.00 | 0.00 | 88.30 | 11.70 | 100 | 93.62 | 88.30 | 88.30 | 100 |
| Bare Soil(%) | 54.94 | 50.14 | 94.45 | 89.29 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Bitumen(%) | 96.98 | 95.23 | 97.21 | 99.53 | 100 | 100 | 94.88 | 100 | 100 | 95.23 | 100 |
| Shadows(%) | 59.56 | 57.62 | 0.00 | 59.28 | 100 | 59.56 | 41.44 | 100 | 100 | 59.56 | 100 |
| OA(%) | 66.75 | 64.50 | 81.35 | 84.42 | 93.25 | 89.71 | 96.15 | 99.77 | 93.27 | 95.64 | 100 |
| NMI(%) | 77.87 | 70.71 | 75.35 | 79.68 | 95.84 | 90.27 | 95.59 | 99.20 | 95.81 | 94.66 | 100 |
| κ (%) | 58.93 | 55.39 | 75.24 | 84.01 | 91.08 | 86.34 | 95.68 | 99.69 | 91.10 | 94.24 | 100 |

The bold values represent the highest values of the three evaluation indicators among all experimental methods.

TABLE V
CLUSTERING RESULTS OF LRR, SSC, LRSC, GCOT, EGCSC, EKGCS, GR-RSCNet, DSC, AND DIFFSC ON THE INDIAN PINES DATASET

| Class | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|--------------------|-------|-------|-------|-------|-------|-------------------|-----------|--------|---------|-------|--------------|
| Corn-notill(%) | 16.52 | 0.00 | 89.65 | 70.85 | 74.83 | 45.07 | 72.74 | 79.20 | 84.68 | 78.31 | 92.04 |
| Grass-trees(%) | 68.08 | 80.14 | 64.05 | 100 | 99.73 | 98.90 | 99.86 | 100 | 100 | 99.45 | 100 |
| Soybean-notill(%) | 91.80 | 85.38 | 96.04 | 87.02 | 100 | 82.24 | 97.81 | 84.29 | 100 | 92.62 | 100 |
| Soybean-mintill(%) | 84.20 | 95.43 | 50.31 | 93.40 | 93.24 | 96.67 | 96.47 | 92.57 | 95.06 | 96.47 | 98.28 |
| OA(%) | 67.30 | 69.37 | 74.47 | 88.27 | 91.23 | 82.83 | 91.82 | 89.36 | 94.33 | 92.17 | 97.43 |
| NMI(%) | 48.54 | 54.00 | 64.05 | 83.08 | 77.99 | 61.54 | 79.95 | 72.09 | 83.24 | 79.34 | 91.49 |
| κ (%) | 51.43 | 53.25 | 66.23 | 69.76 | 87.41 | 74.89 | 88.15 | 84.62 | 91.90 | 88.65 | 96.31 |

The bold values represent the highest values of the three evaluation indicators among all experimental methods.

TABLE VI
CLUSTERING RESULTS OF LRR, SSC, LRSC, EGCSC, DSC, DS³C, GR-RSCNet, MDSCDL, HYPERAE, DMISC, AND DIFFSC ON THE HONGHU DATASET

| Class | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|-----------------------------|-------|-------|-------|-------|-------|-------------------|-----------|--------|---------|--------------|--------------|
| Bare soil(%) | 77.17 | 82.28 | 83.75 | 100 | 95.28 | 97.64 | 98.88 | 97.36 | 98.48 | 97.98 | 99.33 |
| Rape(%) | 100 | 86.45 | 100 | 84.43 | 100 | 65.57 | 99.82 | 84.25 | 91.21 | 96.15 | 100 |
| Tuber mustard(%) | 0.00 | 65.69 | 0.00 | 92.20 | 0.00 | 74.66 | 91.81 | 99.22 | 100 | 100 | 96.30 |
| White radish(%) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 96.53 | 0.00 | 100 | 100 | 100 |
| Cotton(%) | 100 | 97.03 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| Small brassica chinensis(%) | 0.00 | 70.34 | 100 | 88.17 | 80.39 | 80.23 | 91.57 | 96.43 | 100 | 100 | 100 |
| Pakchoi(%) | 100 | 80.49 | 100 | 56.50 | 98.95 | 96.64 | 100 | 100 | 100 | 100 | 100 |
| Red roof(%) | 100 | 73.48 | 82.02 | 100 | 94.83 | 99.78 | 98.65 | 100 | 99.10 | 99.55 | 100 |
| OA(%) | 72.05 | 77.63 | 82.07 | 83.02 | 84.00 | 87.45 | 97.99 | 93.15 | 98.79 | 99.10 | 99.64 |
| NMI(%) | 81.56 | 69.40 | 83.14 | 88.86 | 85.59 | 84.60 | 95.66 | 93.57 | 97.45 | 98.86 | 98.75 |
| κ (%) | 66.26 | 73.05 | 78.47 | 79.89 | 80.65 | 84.75 | 97.58 | 91.74 | 98.54 | 98.91 | 99.43 |

The bold values represent the highest values of the three evaluation indicators among all experimental methods.

3) *Time Cost Comparison*: Table VIII records the running times of different methods on different datasets. For the fairness of the comparison, all methods are performed on the same GPU. As shown in Table VIII, the runtimes of the original shallow methods (LRR, SSC, LRSC, and EGCSC) are shorter compared to the deep clustering models (DSC, DS³C, GR-RSCNet, MDSCDL, HyperAE, DMISC, and

DiffSC), but are much inferior to the deep models in terms of clustering performance. Since deep models contain millions or even tens of millions of training parameters, it is very time-consuming to train deep models, but these deep models can extract deeper feature information compared to shallow models, so the increased time cost yields better clustering results.

TABLE VII
CLUSTERING RESULTS OF LRR, SSC, LRSC, EGCSC, DSC, DS³C, GR-RSCNet, MDSCDL, HYPERAE, DMISC, AND DIFFSC ON THE LONGKOU DATASET

| Class | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|------------------------|-------|-------|-------|-------|-------|-------------------|-----------|--------------|---------|-------|--------------|
| Water(%) | 100 | 100 | 99.30 | 99.95 | 99.86 | 100 | 99.91 | 99.91 | 99.91 | 99.81 | 99.91 |
| Roads and houses(%) | 82.87 | 84.10 | 78.05 | 98.21 | 96.98 | 76.15 | 97.87 | 95.86 | 93.62 | 95.74 | 97.87 |
| Broad-leaf soybean(%) | 49.65 | 57.61 | 31.42 | 65.93 | 66.73 | 58.67 | 61.68 | 63.89 | 64.87 | 60.09 | 69.73 |
| Mixed weed(%) | 0.42 | 58.54 | 84.65 | 98.46 | 0.00 | 47.41 | 96.16 | 99.42 | 99.62 | 99.42 | 98.27 |
| Corn(%) | 100 | 0.00 | 98.69 | 99.84 | 91.50 | 100 | 99.84 | 99.84 | 99.84 | 99.84 | 97.87 |
| Narrow-leaf soybean(%) | 86.72 | 96.03 | 100 | 0.00 | 100 | 100 | 99.24 | 100 | 100 | 98.78 | 99.54 |
| OA(%) | 78.03 | 75.23 | 81.96 | 82.10 | 83.56 | 83.98 | 91.94 | 92.43 | 92.29 | 91.52 | 93.68 |
| NMI(%) | 68.04 | 64.13 | 74.47 | 84.17 | 79.74 | 78.08 | 90.04 | 90.37 | 89.82 | 89.62 | 89.28 |
| κ (%) | 72.01 | 67.18 | 77.25 | 77.04 | 78.86 | 79.60 | 89.75 | 90.36 | 90.18 | 89.21 | 91.96 |

The bold values represent the highest values of the three evaluation indicators among all experimental methods.

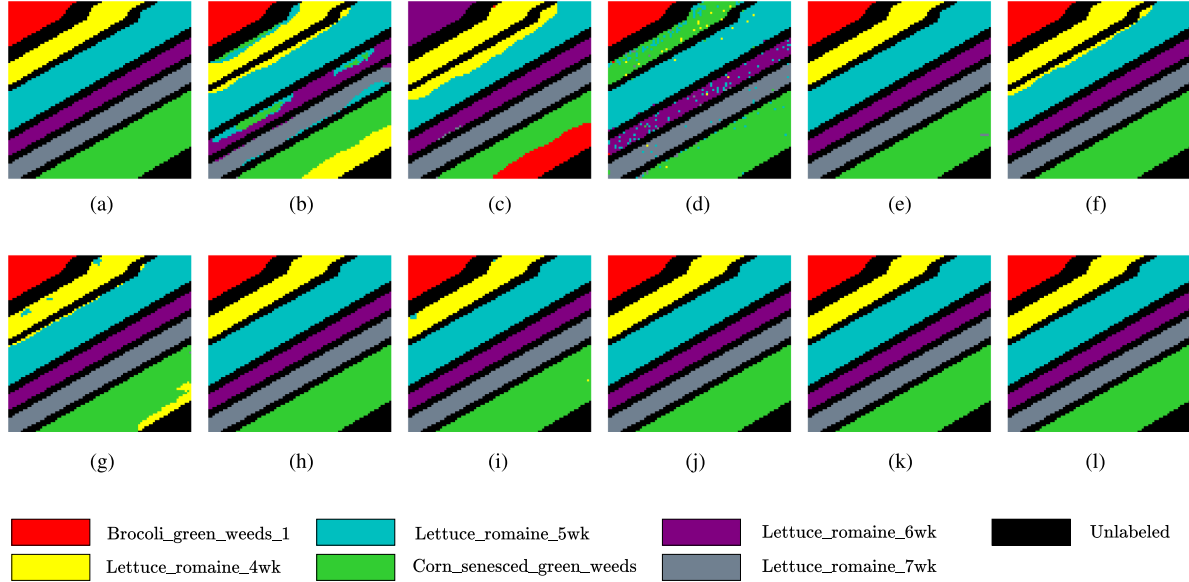


Fig. 4. Segmentation results obtained by different approaches on the SalinasA dataset. (a) Ground truth. (b) LRR 82.16%. (c) SSC 76.23%. (d) LRSC 86.31%. (e) EGCSC 100%. (f) DSC 97.64%. (g) DS³C 96.56%. (h) GR-RSCNet 100%. (i) MDSCDL 99.83%. (j) HyperAE 100%. (k) DMISC 100%. (l) DiffSC 100%.

TABLE VIII
RUNNING TIME OF DIFFERENT MODELS (SECOND)

| Dataset | LRR | SSC | LRSC | EGCSC | DSC | DS ³ C | GR-RSCNet | MDSCDL | HyperAE | DMISC | DiffSC |
|------------------|--------|--------|--------|--------|---------|-------------------|-----------|---------|---------|---------|---------|
| SalinasA | 52.635 | 55.578 | 43.467 | 48.135 | 219.924 | 245.578 | 196.785 | 106.553 | 199.305 | 105.218 | 214.039 |
| Pavia University | 26.054 | 25.335 | 22.244 | 74.220 | 453.363 | 556.823 | 339.005 | 138.229 | 186.627 | 173.255 | 309.387 |
| Indian Pines | 32.214 | 77.429 | 28.093 | 73.835 | 214.506 | 202.146 | 157.564 | 173.096 | 176.823 | 89.768 | 221.920 |
| HongHu | 42.833 | 38.710 | 39.989 | 74.144 | 306.483 | 554.529 | 246.292 | 665.403 | 123.826 | 488.307 | 319.381 |
| LongKou | 58.196 | 42.647 | 45.020 | 64.159 | 236.491 | 658.906 | 195.507 | 239.263 | 159.717 | 125.901 | 225.950 |

C. Ablation Study

1) *Impact of DM*: To demonstrate the effect of introducing the DM, we compared the capability of various DSC models. Since the DM is essentially a prediction of the noise added to the original data at a specific time step t during the diffusion process, and therefore, also belongs to the noise feature, we use the output of each convolutional layer of D_θ as an intermediate activation to learn deep SE coefficient matrix \mathbf{C} together after connecting with DSC separately, thus proving that the intermediate activation feature of our extracted third convolutional layer is valid. We use random noise added to DSC to prove the noise feature extracted by the DM is efficient. We refer to the baseline

as DSC and compare the network clustering performance of adding random noise (DSC + noise), the intermediate activation of the first layer of D_θ (DSC + D_{θ_1}), the intermediate activation of the second layer (DSC + D_{θ_2}), the intermediate activation of the third layer (DiffSC), and the intermediate activation of the last layer (DSC + $D_{\theta_{last}}$), respectively. For the fairness of the comparison, the aforementioned methods use an identical model configuration and random seed to initialize and train on the same GPU. The outcomes are presented in Table IX. As we can see from the table that DSC + D_{θ_1} , DSC + D_{θ_2} , DSC + $D_{\theta_{last}}$, and DiffSC all have a considerable improvement in clustering performance compared with DSC.

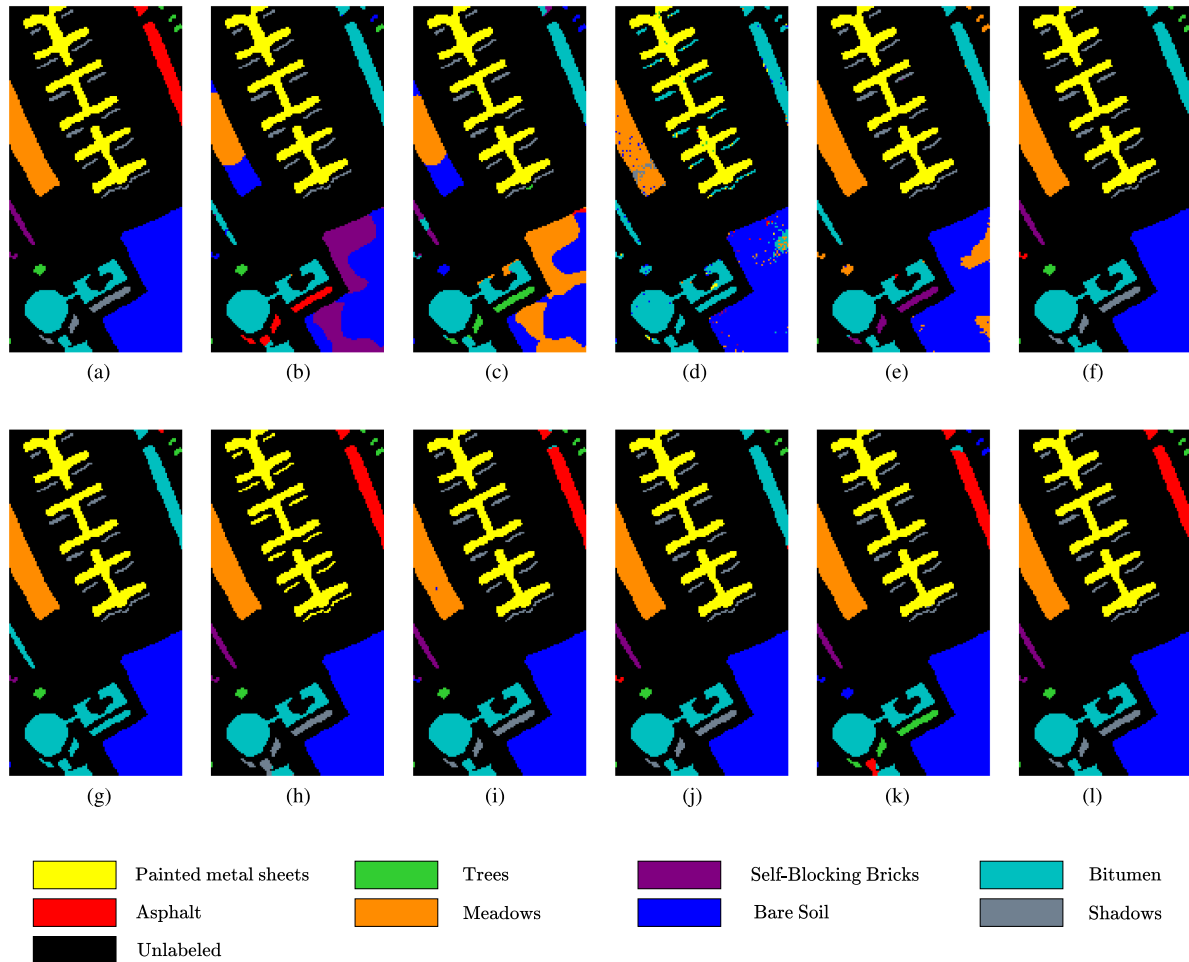


Fig. 5. Segmentation results obtained by different approaches on the Pavia University dataset. (a) Ground truth. (b) LRR 66.75%. (c) SSC 64.50%. (d) LRSC 81.35%. (e) EGCSC 84.42%. (f) DSC 93.25%. (g) DS^3C 89.71%. (h) GR-RSCNet 96.15%. (i) MDSCDL 99.77%. (j) HyperAE 93.27%. (k) DMISC 95.64%. (l) DiffSC 100%.

TABLE IX
PERFORMANCE OF DIFFERENT MODELS

| Approach | SalinasA | | | Pavia University | | | Indian Pines | | | HongHu | | | LongKou | | |
|--------------------------|----------|--------|--------------|------------------|--------|--------------|--------------|--------|--------------|--------|--------|--------------|---------|--------|--------------|
| | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) |
| DSC | 97.64 | 95.75 | 97.07 | 93.25 | 95.85 | 91.08 | 91.23 | 77.99 | 87.41 | 84.00 | 85.59 | 80.65 | 83.56 | 79.74 | 78.86 |
| DSC+noise | 95.37 | 94.46 | 94.57 | 74.84 | 83.23 | 69.20 | 81.12 | 57.45 | 72.42 | 80.83 | 88.27 | 77.33 | 75.51 | 74.40 | 68.32 |
| DSC+ D_{θ_1} | 99.96 | 99.83 | 99.95 | 97.52 | 96.14 | 96.73 | 92.51 | 78.09 | 89.45 | 87.70 | 92.62 | 85.35 | 89.97 | 85.81 | 87.26 |
| DSC+ D_{θ_2} | 100 | 100 | 100 | 99.42 | 98.55 | 99.25 | 95.76 | 85.80 | 93.93 | 98.70 | 97.24 | 98.43 | 93.03 | 90.87 | 91.13 |
| DSC+ $D_{\theta_{last}}$ | 100 | 100 | 100 | 98.53 | 97.12 | 95.52 | 93.58 | 82.25 | 90.15 | 98.73 | 97.45 | 98.47 | 91.70 | 89.92 | 89.47 |
| DiffSC | 100 | 100 | 100 | 100 | 100 | 100 | 97.43 | 91.49 | 96.31 | 99.64 | 98.75 | 99.43 | 93.68 | 89.28 | 91.96 |

Specifically, we first add random Gaussian noise to the DSC, and the experimental effect proves that the random noise does not improve the model performance, but after adding different layers of intermediate activations of the D_{θ} network, all of them have significant improvements over the original DSC, which proves that our proposed diffusion representation learning is effective, and the best effect is DiffSC, which adds the third layer of intermediate activations of the D_{θ} network to the DSC.

2) *Robustness Evaluation*: Considering that the image data used for HSI SC are more or less affected by the surrounding environment in real situations, we evaluate the robustness of

DiffSC by adding random noise to the HSIs to simulate realistic disturbances. Since the pixel values of the images to be processed have been normalized to $[0, 1]$, we add Gaussian noise with 0 of mean and 0.1 of variance to the HSIs to simulate the HSIs subject to environmental interference in real situations and compare the SC performance of the HSIs without noise and those with noise. The results are shown in Table X, where the clustering accuracies are 100%, 100%, 97.43%, 99.64%, and 93.68% for the noise-free data and 100%, 99.95%, 96.95%, 99.19%, and 92.96% for the noise-containing data, respectively. It can be seen that the clustering performance is slightly reduced after

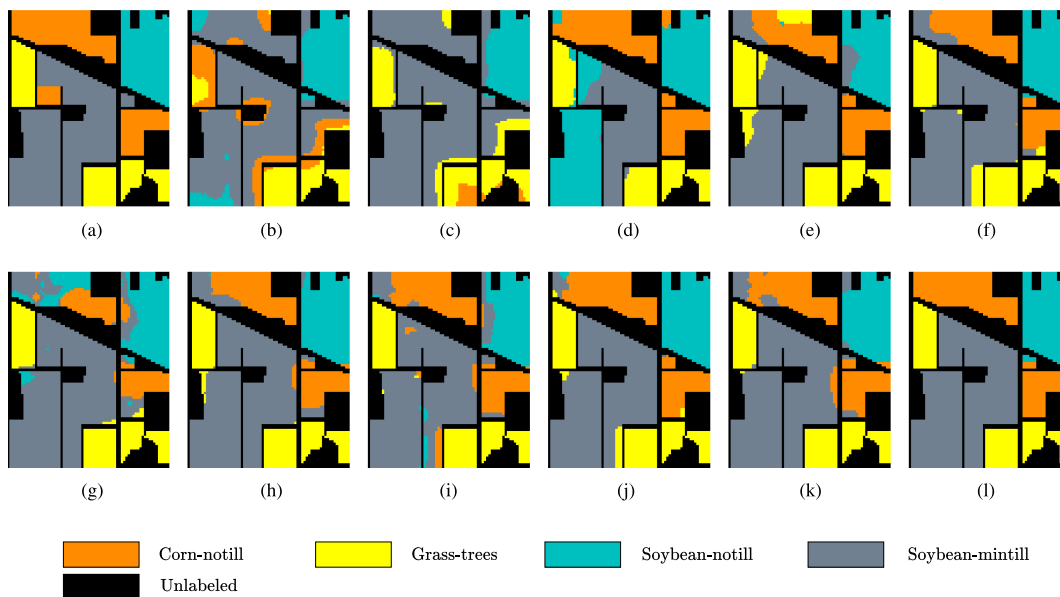


Fig. 6. Segmentation results obtained by different approaches on the Indian Pines dataset. (a) Ground truth. (b) LRR 67.30%. (c) SSC 69.37%. (d) LRSC 77.47%. (e) EGCSC 88.27%. (f) DSC 91.23%. (g) DS³C 82.83%. (h) GR-RSCNet 91.82%. (i) MDSCDL 89.36%. (j) HyperAE 94.33%. (k) DMISC 92.17%. (l) DiffSC 97.43%.

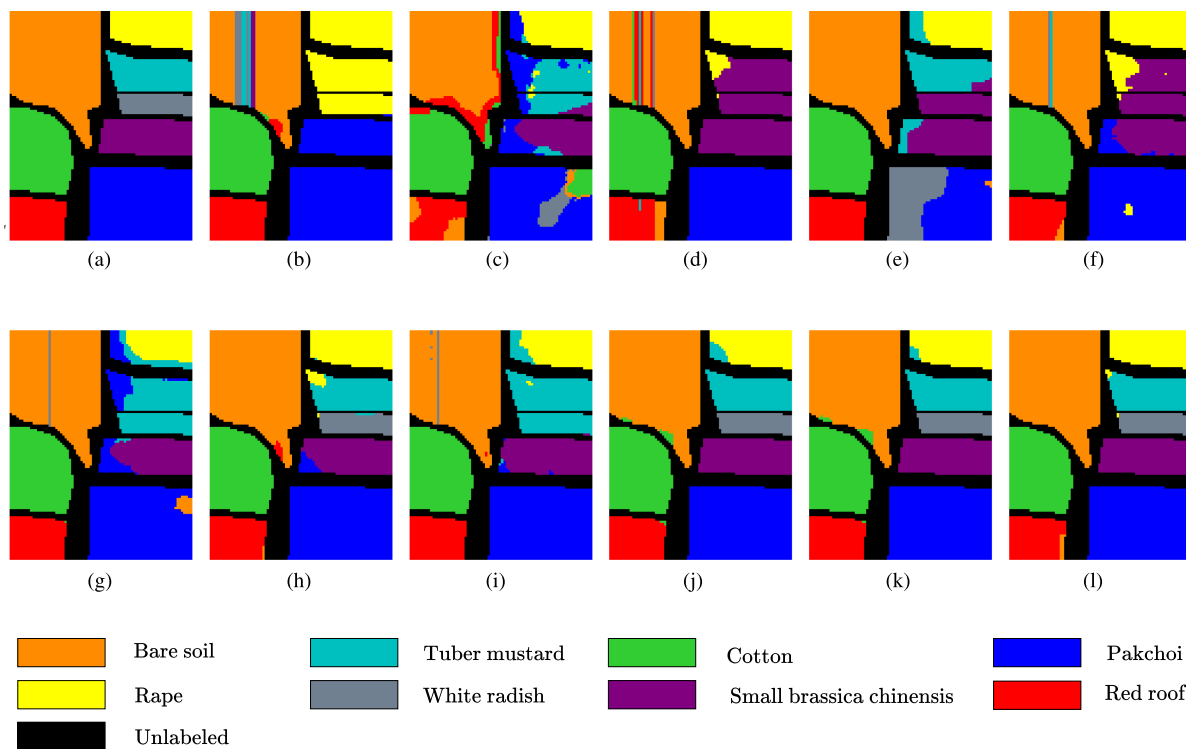


Fig. 7. Segmentation results obtained by different approaches on the HongHu dataset. (a) Ground truth. (b) LRR 72.05%. (c) SSC 77.63%. (d) LRSC 82.07%. (e) EGCSC 83.02%. (f) DSC 84.00%. (g) DS³C 87.45%. (h) GR-RSCNet 97.99%. (i) MDSCDL 93.15%. (j) HyperAE 98.79%. (k) DMISC 99.10%. (l) DiffSC 99.64%.

TABLE X
ROBUSTNESS EVALUATION OF DIFFSC

| Approach | SalinasA | | | Pavia University | | | Indian Pines | | | HongHu | | | LongKou | | |
|-------------------|----------|--------|--------------|------------------|--------|--------------|--------------|--------|--------------|--------|--------|--------------|---------|--------|--------------|
| | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) | OA(%) | NMI(%) | κ (%) |
| DiffSC(HSI) | 100 | 100 | 100 | 100 | 100 | 100 | 97.43 | 91.49 | 96.31 | 99.64 | 98.75 | 99.43 | 93.68 | 89.28 | 91.96 |
| DiffSC(HSI+noise) | 100 | 100 | 100 | 99.95 | 99.85 | 99.94 | 96.95 | 89.63 | 95.64 | 99.19 | 98.33 | 99.02 | 92.96 | 91.90 | 91.06 |

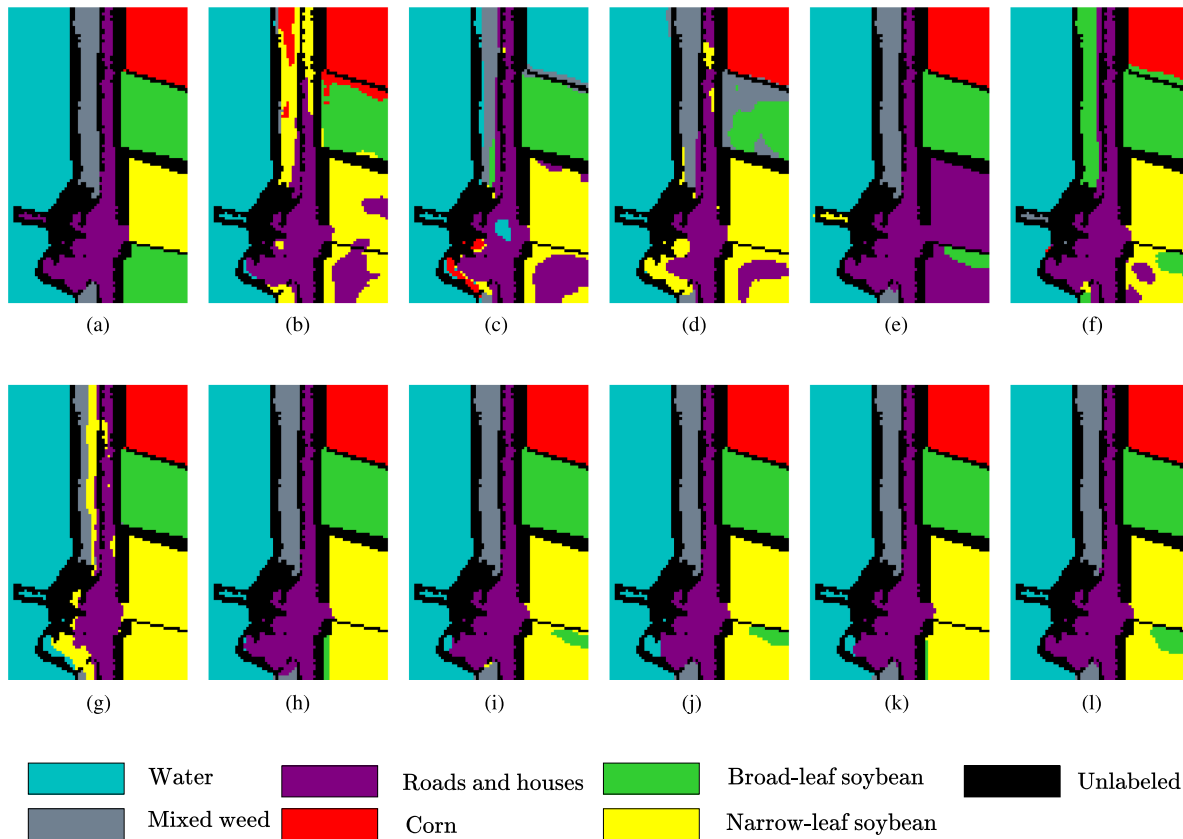


Fig. 8. Segmentation results obtained by different approaches on the LongKou dataset. (a) Ground truth. (b) LRR 78.03%. (c) SSC 75.23%. (d) LRSC 81.96%. (e) EGCSC 82.10%. (f) DSC 83.56%. (g) DS³C 83.98%. (h) GR-RSCNet 91.94%. (i) MDSCDL 92.43%. (j) HyperAE 92.29%. (k) DMISC 91.52%. (l) DiffSC 93.68%.

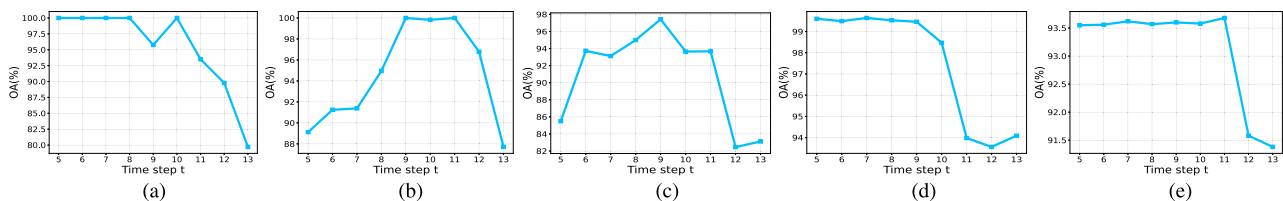


Fig. 9. Influence of the time step t on (a) SalinasA, (b) Pavia University, (c) Indian Pines, (d) HongHu, and (e) LongKou datasets.

adding noise, but it is also better than most of the compared methods. The clustering performance fluctuates within 0.72%, which is sufficient to show the high robustness of the proposed method.

3) *Impact of the Time Step T* : The scale of the noise removed in the inverse diffusion process decreases with the time step t , moreover, when the smaller t is, the closer it is to the real data distribution. It is reasonable to believe that the finer the noise is, the more information the data distribution contains, so the t values we selected are on the small side.

We conducted experiments to research the time step t effect on the clustering capability, where we spaced the time steps from 5 to 13 with an interval of 1. As shown in Fig. 9, the optimal time steps t for the SalinasA, Pavia University, Indian Pines, HongHu, and LongKou datasets are 5, 9, 9, 7, and 11, respectively. Since the closer the inverse diffusion process is to the original data,

the more data information contained in the features extracted by the D_θ network, the greater the boost to the clustering results, therefore, the selected time step t is usually on the small side.

4) *Influence of Neighborhood Size*: We investigate the impact of HSI neighborhood size on clustering performance by varying the neighborhood size from 3×3 to 17×17 with an interval of 2×2 . It is shown in Fig. 10 that the optimal neighborhood sizes of SalinasA, Pavia University, Indian Pines, HongHu, and LongKou datasets are 9, 13, 13, 11, and 9, respectively. Since identical objects usually present as a homogeneous region in HSI, appropriately increasing the neighborhood size enhances the clustering performance of DiffSC. However, it is not the case that the larger the neighborhood size, the more powerful the clustering ability. When it is larger than 13, the clustering performance of DiffSC on the five datasets gradually decreases, which is due to the fact that the larger neighborhood

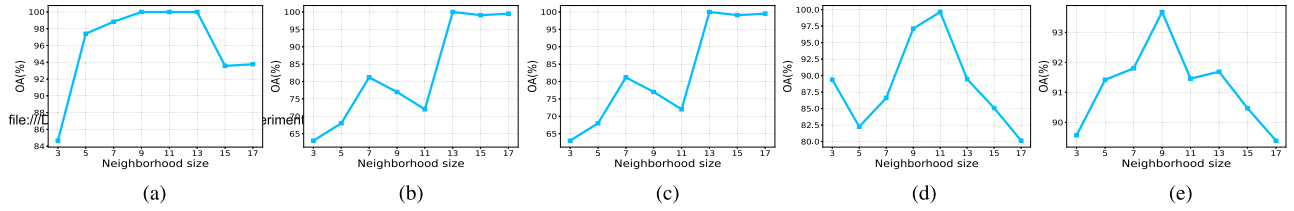


Fig. 10. Influence of neighborhood size on (a) SalinasA, (b) Pavia University, (c) Indian Pines, (d) HongHu, and (e) LongKou datasets.

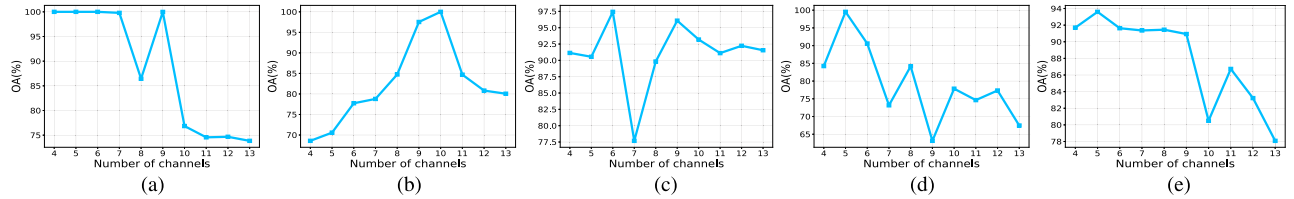


Fig. 11. Influence of number of channels on (a) SalinasA, (b) Pavia University, (c) Indian Pines, (d) HongHu, and (e) LongKou datasets.

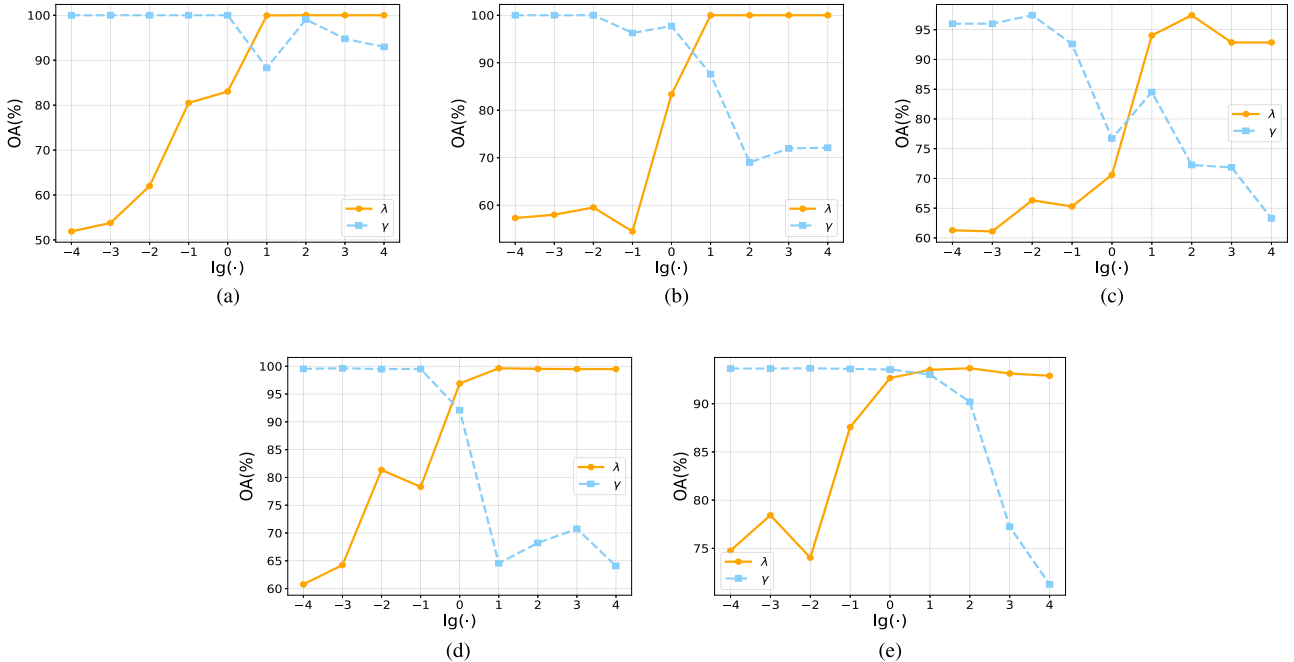


Fig. 12. Influence of the loss function parameters on (a) SalinasA, (b) Pavia University, (c) Indian Pines, (d) HongHu, and (e) LongKou datasets.

contains more noise points and more interference, which has a greater influence on the clustering outcomes, especially the boundaries of different classes of objects.

5) *Influence of Numbers of Channels*: Due to the number of channels of HSI data being large, usually hundreds, and the disadvantage of information redundancy, it is usually needed to perform channel dimensionality reduction using PCA before clustering. In order to ensure that HSI data retain the most information of original data in low-dimensional space, we research the impact of the number of channels for clustering results after dimensionality reduction using PCA. We set the number of channels from 4 to 13 with an interval of 1, and the outcomes are illustrated in Fig. 11. We know from the figure that the optimal channel sizes for SalinasA, Pavia University,

Indian Pines, HongHu, and LongKou datasets are 4, 10, 6, 5, and 5, respectively, instead of our belief that the higher the number of channels after dimensionality reduction, the more information is retained and the better the clustering performance, which requires dimensionality reduction to different dimensions depending on the data. And the channel size should not be too small. If the number of channels is too small, it is not enough to represent the information of the original data and the clustering performance will be degraded in some cases.

6) *Influence of λ and γ* : In this subsection, we research the effect of two significant regularization coefficients in joint loss function \mathcal{L} on the clustering performance, namely λ and γ . We vary these two hyperparameters from 0.0001 to 10000, computing one while fixing the other. As shown in Fig. 12, we use

the logarithm of the parameters as the horizontal coordinate for clear representation. From the figure, it can be seen that the two parameters have different effects on the clustering performance. It also shows a trend that in order to get better clustering results, λ needs to set a larger value, γ need to set a lesser value, i.e., $\lambda > 10$ and $\gamma < 1$. It can be seen from the joint loss that a larger λ makes the network more focused on the loss of \mathbf{Z} and \mathbf{CZ} in the subspace, which is important to get a better deep SE coefficient matrix \mathbf{C} , which results in a robust deep affinity matrix \mathbf{A} and more accurate clustering results.

V. DISCUSSION

The success of the DiffSC illustrates that considering the intrinsic distribution features of the data is effective for HSI SC. This provides a direction for generative models like the DPM to be used for extracting data distribution features. Since the DiffSC model belongs to feature fusion by nature, all the improvement strategies to the traditional DSC model are applicable to DiffSC.

However, it can be seen from the experiment results that the difference in OA on different datasets is also very obvious, for example, the OA on the LongKou dataset is only 93.68%. This indicates that although DiffSC obtains better clustering results than the existing DSC methods, however, it does not achieve high OA for all datasets.

Furthermore, it should be noted that D_θ in our proposed model is simplified, only using the intermediate activation of one of the layers, which has a low feature dimension and the extracted features may not be comprehensive. We will investigate the solution to this problem in future research to make the extracted features more effective.

VI. CONCLUSION

We present a new HSI SC framework on the basis of DSC, namely DiffSC. The key to this framework is the use of a diffusion model trained jointly with DSC methods for extracting the deep affinity matrix. To obtain robust deep affinity matrices, a distribution feature constraint is added in the joint loss function to learn information about the distribution of raw data, moreover, we simplify the noise prediction network of the DPM, making it easy to train the model from scratch. Outcomes of the experiment on five HSI datasets indicate that DiffSC obtain the most advanced clustering results, where DiffSC achieves 100%, 100%, 97.43%, 99.64%, and 93.68% clustering OA on SalinasA, Pavia University, Indian Pines, HongHu, and LongKou datasets, respectively. This result is already comparable to the supervised HSI classification accuracy, which also implies that traditional clustering models and deep learning models combined have great promise.

REFERENCES

- [1] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.
- [2] G. Lu and B. Fei, "Medical hyperspectral imaging: A review," *J. Biomed. Opt.*, vol. 19, no. 1, pp. 010901–010901, 2014.
- [3] H. Sun, X. Zheng, and X. Lu, "A supervised segmentation network for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 30, pp. 2810–2825, Feb. 4, 2021, doi: [10.1109/TIP.2021.3055613](https://doi.org/10.1109/TIP.2021.3055613).
- [4] P. Zhang, H. He, and L. Gao, "A nonlinear and explicit framework of supervised manifold-feature extraction for hyperspectral image classification," *Neurocomputing*, vol. 337, pp. 315–324, 2019.
- [5] I. Dópido, J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 7, pp. 4032–4044, Jul. 2013.
- [6] L. Yang, S. Yang, P. Jin, and R. Zhang, "Semi-supervised hyperspectral image classification using spatio-spectral Laplacian support vector machine," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 3, pp. 651–655, Mar. 2014.
- [7] L. Sun, Z. Wu, J. Liu, L. Xiao, and Z. Wei, "Supervised spectral-spatial hyperspectral image classification with weighted Markov random fields," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1490–1503, Mar. 2015.
- [8] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, Mar. 2018.
- [9] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral-spatial residual network for hyperspectral image classification: A 3-D deep learning framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 847–858, Feb. 2018.
- [10] B. Liu, X. Yu, A. Yu, P. Zhang, G. Wan, and R. Wang, "Deep few-shot learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2290–2304, Apr. 2019.
- [11] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.
- [12] S. Zhou, Z. Xue, and P. Du, "Semisupervised stacked autoencoder with cotraining for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3813–3826, Jun. 2019.
- [13] M. E. Paoletti, J. M. Haut, R. Fernandez-Beltran, J. Plaza, A. J. Plaza, and F. Pla, "Deep pyramidal residual networks for spectral-spatial hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 2, pp. 740–754, Feb. 2019.
- [14] W. Zhao and S. Du, "Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4544–4554, Aug. 2016.
- [15] Y. Chen, Y. Wang, Y. Gu, X. He, P. Ghamisi, and X. Jia, "Deep learning ensemble for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1882–1897, Jun. 2019.
- [16] Z. Zhong, J. Li, L. Ma, H. Jiang, and H. Zhao, "Deep residual networks for hyperspectral image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2017, pp. 1824–1827.
- [17] W. Sun, L. Zhang, B. Du, W. Li, and Y. Mark Lai, "Band selection using improved sparse subspace clustering for hyperspectral imagery classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2784–2797, Jun. 2015.
- [18] H. Zhai, H. Zhang, P. Li, and L. Zhang, "Hyperspectral image clustering: Current achievements and future lines," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 4, pp. 35–67, Dec. 2021.
- [19] J. Zhu, Z. Jiang, G. D. Evangelidis, C. Zhang, S. Pang, and Z. Li, "Efficient registration of multi-view point sets by K-means clustering," *Inf. Sci.*, vol. 488, pp. 205–218, 2019.
- [20] P. Hu, X. Liu, Y. Cai, and Z. Cai, "Band selection of hyperspectral images using multiobjective optimization-based sparse self-representation," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 3, pp. 452–456, Mar. 2019.
- [21] L. Zhang, L. Zhang, B. Du, J. You, and D. Tao, "Hyperspectral image unsupervised classification by robust manifold matrix factorization," *Inf. Sci.*, vol. 485, pp. 154–169, 2019.
- [22] S. Liu and H. Wang, "Graph convolutional optimal transport for hyperspectral image spectral clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 1, 2022, Art. no. 4414013, doi: [10.1109/TGRS.2022.3203481](https://doi.org/10.1109/TGRS.2022.3203481).
- [23] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, "A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2017.
- [24] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral-spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.
- [25] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 11, pp. 2003–2007, Nov. 2017.
- [26] C. Hinojosa, E. Vera, and H. Arguello, "A fast and accurate similarity-constrained subspace clustering algorithm for hyperspectral image," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 10773–10783, Oct. 14, 2021, doi: [10.1109/JSTARS.2021.3120071](https://doi.org/10.1109/JSTARS.2021.3120071).

- [27] Y. Wan, Y. Zhong, A. Ma, and L. Zhang, "Multi-objective sparse subspace clustering for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2290–2307, Apr. 2020.
- [28] S. Huang, H. Zhang, and A. Pižurica, "Hybrid-hypergraph regularized multiview subspace clustering for hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, May 3 2021, Art. no. 5505816, doi: [10.1109/TGRS.2021.3074184](https://doi.org/10.1109/TGRS.2021.3074184).
- [29] R. Vidal, "Subspace clustering," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 52–68, Mar. 2011.
- [30] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3918–3927.
- [31] S. Liu and H. Wang, "Geometric weighting subspace clustering on nonlinear manifolds," *Multimedia Tools Appl.*, vol. 81, pp. 42971–42990, 2022.
- [32] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.
- [33] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013.
- [34] Y. Cai, Z. Zhang, Z. Cai, X. Liu, X. Jiang, and Q. Yan, "Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4191–4202, May 2021.
- [35] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognit. Lett.*, vol. 43, pp. 47–61, 2014.
- [36] D. Bo, X. Wang, C. Shi, M. Zhu, E. Lu, and P. Cui, "Structural deep clustering network," in *Proc. Web Conf.*, 2020, pp. 1400–1410.
- [37] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 139–156.
- [38] S. Das, S. Pratiher, C. Kyal, and P. Ghamisi, "Sparsity regularized deep subspace clustering for multicriterion-based hyperspectral band selection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4264–4278, May 3 2022, doi: [10.1109/JSTARS.2022.3172112](https://doi.org/10.1109/JSTARS.2022.3172112).
- [39] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid, "Deep subspace clustering networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 23–32.
- [40] J. Lei, X. Li, B. Peng, L. Fang, N. Ling, and Q. Huang, "Deep spatial-spectral subspace clustering for hyperspectral image," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 7, pp. 2686–2697, Jul. 2021.
- [41] Y. Cai, M. Zeng, Z. Cai, X. Liu, and Z. Zhang, "Graph regularized residual subspace clustering network for hyperspectral image clustering," *Inf. Sci.*, vol. 578, pp. 85–101, 2021.
- [42] J. Wang, B. Wu, Z. Ren, and Y. Zhou, "Multi-scale deep subspace clustering with discriminative learning," *IEEE Access*, vol. 10, pp. 91283–91293, 2022.
- [43] Y. Cai, Z. Zhang, Z. Cai, X. Liu, and X. Jiang, "Hypergraph-structured autoencoder for unsupervised and semisupervised classification of hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, Feb. 5 2021, Art. no. 5503505, doi: [10.1109/LGRS.2021.3054868](https://doi.org/10.1109/LGRS.2021.3054868).
- [44] T. Li, Y. Cai, Y. Zhang, Z. Cai, and X. Liu, "Deep mutual information subspace clustering network for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, May 26 2022, Art. no. 6009905, doi: [10.1109/LGRS.2022.3178168](https://doi.org/10.1109/LGRS.2022.3178168).
- [45] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
- [46] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 2256–2265.
- [47] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10684–10695.
- [48] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022, *arXiv:2207.12598*.
- [49] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 8780–8794.
- [50] R. Huang, Z. Zhao, H. Liu, J. Liu, C. Cui, and Y. Ren, "ProDiff: Progressive fast diffusion model for high-quality text-to-speech," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 2595–2605.
- [51] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018.
- [52] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," 2021, *arXiv:2009.09761*.
- [53] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang, "GeoDiff: A geometric diffusion model for molecular conformation generation," 2022, *arXiv:2203.02923*.
- [54] J. Ho et al., "Imagen video: High definition video generation with diffusion models," 2022, *arXiv:2210.02303*.
- [55] Y. Zhong, X. Hu, C. Luo, X. Wang, J. Zhao, and L. Zhang, "WHU-Hi: UAV-borne hyperspectral with high spatial resolution (H2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with CRF," *Remote Sens. Environ.*, vol. 250, 2020, Art. no. 112012.



Jiaxin Chen received the B.S. degree in applied physics from the Tianjin University of Technology, Tianjin, China, in 2020. He is currently working toward the M.E. degree in computer technology with the Chengdu University of Technology, Chengdu, China. His research interests include hyperspectral image processing, computer vision, and speech synthesis.



Shujun Liu (Student Member, IEEE) received the M.S. degree in computer software and theory, in 2017, from the Chengdu University of Technology, Chengdu, China, where he has been working toward the Ph.D. degree in earth exploration and information technology since 2021. His research interests include machine learning theory and hyperspectral image processing.



Zhongbiao Zhang is currently working toward the master's degree in resources and environment with the Chengdu University of Technology, Chengdu, China.



Huajun Wang received the B.S. degree in physics from the Beijing Normal University, Beijing, China, in 1986, and the M.S. and Ph.D. degrees in electronic science and technology from the University of Electronic Science and Technology of China, Chengdu, China, in 1997 and 2001, respectively. He was a Postdoctoral Researcher from 2000 to 2004 with the University of Electronic Science and Technology of China, and the Lomonosov Moscow State University, Moscow, Russia. Since 2005, he has been a Full Professor with the Chengdu University of

Technology, Chengdu.