Supplementary Materials

1 Network Architectures

The architectures of the encoders with respect to different size of images are described in Table 1, Table 2 and Table 3. The ResBlocks used in the encoders are presented in Fig. 1. Table 4 and Table 5 show the architecture of the critic and the discriminator, respectively. The critic and the discriminator both consist of three fully-connected layers. The discriminator shares some layers with the encoder to reduce computations, *i.e.*, the input X in Table 5 is a flatten vector created by the encoder. For the encoder in Table 1, X is the output of the last convolutional layer, while for the encoders in Table 2 and Table 3, X is the output of the second last ResBlock. The slopes of lReLU functions for all architectures are set to 0.2.

2 Data Augmentation

The data augmentation adopted in DCCS includes four commonly used approaches:

(1) Random cropping: randomly crop a rectangular region whose aspect ratio and area are randomly sampled in the range of $[3/4, 4/3]$ and $[40\%, 100\%]$, respectively, and then resize the cropped region to the original image size.
(2) Random horizontal flipping: flip the image horizontally with 50% probability.
(3) Color jittering: scale brightness, contrast and saturation with coefficients uniformly drawn from $[0.6, 1.4]$, while scale hue with coefficients uniformly drawn from $[0.875, 1.125]$.
(4) Channel shuffling: randomly shuffle the RGB channels of the image.

Random cropping and color jittering are employed for all datasets. Following [5], random horizontal flipping is used for all datasets except MNIST due to the direction sensitive nature of the digits. Channel shuffling is applied to color images before graying. Note that channel shuffling can also change the brightness of the grayscale images because the RGB channels are summed with different weights for graying.

3 $\boldsymbol{\beta_{\mathrm{Aug}}}$ Configuration

As previously stated, a small β_{Aug} cannot disentangle the style information well, while a large β_{Aug} may lead the clusters to overlap by generating high confidence of the overlapping part of two clusters. Therefore, we propose an applicable way for β_{Aug} configuration by visualizing the t-SNE figure of the latent representation. As shown in Fig. 2a and Fig. 2c, with β_{Aug} being set to 2 for Fashion-MNIST and 4 for CIFAR-10, the clusters are well separated. However, the clusters start to overlap after increasing β_{Aug} to 3 for Fashion-MNIST (Fig. 2b) or 5

Table 1: The encoder architecture for MNIST and Fashion-MNIST, similarly as the architecture used in [9]

Input $X \in \mathbb{R}^{28\times 28}$
4×4, stride=2 conv, BN 64 lReLU
4×4, stride=2 conv, BN 128 lReLU
Dense, BN 1024 lReLU
Dense softmax for Z_c Dense linear for Z_s

Table 2: The encoder architecture for CIFAR-10, similarly as the architecture used in [2] with images converted to grayscale

Input $X \in \mathbb{R}^{32\times 32}$
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock 512
BN, ReLU, global average pooling
Dense softmax for Z_c Dense linear for Z_s

Table 3: The encoder architecture for STL-10 and ImageNet-10, similarly as the architecture used in [2] with images converted to grayscale

Input $X \in \mathbb{R}^{96\times 96}$
ResBlock down 64
ResBlock down 128
ResBlock down 256
ResBlock down 512
ResBlock 512
BN, ReLU, global average pooling
Dense softmax for Z_c Dense linear for Z_s

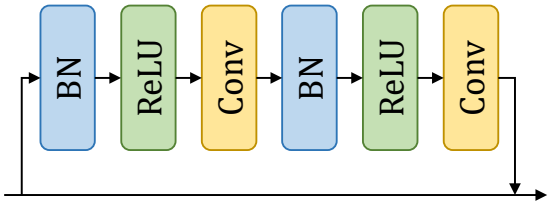


Fig. 1: ResBlock architecture. The kernel size of the convolutional layer is 3×3. 2×2 average pooling is employed for downsampling after the second convolution, while the nearest-neighbor upsampling is applied for upsampling before the first convolution

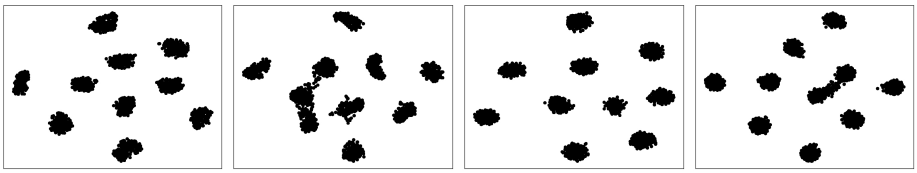
for CIFAR-10 (Fig. 2d). Experiments show that using the biggest β_{Aug} without overlapping clusters in the t-SNE visualization can always yield decent clustering performance.

Table 4: The critic architecture

Input $Z = (Z_c, Z_s)$
Dense, 1024 lReLU
Dense, 512 lReLU
Dense, 1 linear

Table 5: The discriminator architecture

Input (X, Z)
Dense, 1024 lReLU
Dense, 512 lReLU
Dense, 1 sigmoid



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

Fig. 2: The t-SNE visualizations of the latent representations, including Fashion-MNIST with $\beta_{\text{Aug}} = 2$ (a), Fashion-MNIST with $\beta_{\text{Aug}} = 3$ (b), CIFAR-10 with $\beta_{\text{Aug}} = 4$ (c) and CIFAR-10 with $\beta_{\text{Aug}} = 5$ (d). Note that the visualization is completely based on the latent representation without any usage of the ground truth label

4 Discriminator vs. Decoder

The proposed DCCS adopts a discriminator to maximize the mutual information $I(X, Z)$ between the input image X and its latent representation Z to avoid learning arbitrary representations. Autoencoder is another popular approach to embed the image information into the latent representation. The discriminator in the proposed framework could be replaced by a decoder with the mutual information loss being replaced by the reconstruction loss. The performance of these two approaches is compared in Table 6. The decoder architectures for Fashion-MNIST and CIFAR-10 are described in Table 7 and Table 8, respectively. The weight of the reconstruction loss is set to 5 for its best performance. The results show that the reconstruction strategy delivers inferior performance, suggesting that the representations learned by the decoder based DCCS may contain generative information which is irrelevant for clustering.

Table 6: Comparison of different ways to avoid learning arbitrary representations

Method	Fashion-MNIST			CIFAR-10		
	ACC	NMI	ARI	ACC	NMI	ARI
Discriminator	**0.756**	**0.704**	**0.623**	**0.656**	**0.569**	**0.469**
Decoder	0.732	0.703	0.611	0.651	0.565	0.464

Table 7: The decoder architecture for Fashion-MNIST, similarly as the architecture used in [9]

Input $Z = (Z_c, Z_s)$
Dense, BN 1024 lReLU
Dense, BN $7 \times 7 \times 128$ lReLU
4×4, stride=2 deconv, BN 64 lReLU
4×4, stride=2 deconv, BN 1 tanh

Table 8: The decoder architecture for CIFAR-10, similarly as the architecture used in [2] with images converted to grayscale

Input $Z = (Z_c, Z_s)$
Dense, $4 \times 4 \times 512$
ResBlock up 512
ResBlock up 256
ResBlock up 128
BN, ReLU, 3×3 conv, 1 tanh

Table 9: The impact of different preprocessing for Fashion-MNIST

Preprocessing	ACC	NMI	ARI
None	0.756	0.704	0.623
Sobel filtering	**0.758**	**0.706**	**0.625**

Table 10: The impact of different preprocessing for CIFAR-10

Preprocessing	ACC	NMI	ARI
None	0.635	0.544	0.448
Grayscale	**0.656**	**0.569**	**0.469**
Sobel filtering	0.652	0.564	0.464

5 Impact of the Image Preprocessing

For preprocessing, we only convert the color images to grayscale, while IIC [5] further applies Sobel filtering to extract gradient information. Table 9 and Table 10 compare the clustering performance with different preprocessing strategies on Fashion-MNIST and CIFAR-10, respectively. When performing Sobel filtering, a convolutional layer with the Sobel kernel is added before the encoder. For Fashion-MNIST, using Sobel filtering achieves slightly better performance. For CIFAR-10, grayscale without Sobel filtering has the best performance, while clustering on the color images yields the worst performance, indicating that the color information may be trivial for clustering on CIFAR-10.

6 Results on STL-10 with Pretrained Model

Several methods use ResNet-50 [3] pretrained with ImageNet [1] to extract features for clustering. For a fair comparison with these methods, we replace the encoder of DCCS with the same network, *i.e.*, a pretrained ResNet-50 followed by three fully-connected layers with 500, 500, 2000 units, respectively. Batch normalization and ReLU activation function are applied on each fully-connected layer. The parameters of the ResNet-50 are fixed during optimization the same as in previous studies. We use RGB images as inputs and resize them to 224×224 pixels. The input X of the discriminator in Table 5 is the average pooled vector of the last residual block of ResNet-50. As shown in Table 11, DCCS outperforms other state-of-the-art methods, *e.g.* 1.43% accuracy higher than IMSAT [4]. The NMI and ARI metrics of DCCS are 0.9030 and 0.9051, respectively.

Table 11: Comparison of the clustering accuracy with other state-of-the-art methods on STL-10 (without the unlablled subset, using ResNet-50 [3] pre-trained with ImageNet [1]). The best two results are highlighted in **bold**

Method	ACC (%)
AE+GMM [11]	79.83
DEC [10]	80.64
VaDE [6]	84.45
RIM [7]	92.50
IMSAT [4]	**94.10**
LTVAE [8]	90.00
DGG [11]	90.59
DCCS (Proposed)	**95.53**

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009)
2. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems. pp. 5767–5777 (2017)
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
4. Hu, W., Miyato, T., Tokui, S., Matsumoto, E., Sugiyama, M.: Learning discrete representations via information maximizing self-augmented training. In: Proceedings of the International Conference on Machine Learning. pp. 1558–1567 (2017)
5. Ji, X., Henriques, J.F., Vedaldi, A.: Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9865–9874 (2019)
6. Jiang, Z., Zheng, Y., Tan, H., Tang, B., Zhou, H.: Variational deep embedding: An unsupervised and generative approach to clustering. arXiv preprint arXiv:1611.05148 (2016)
7. Krause, A., Perona, P., Gomes, R.G.: Discriminative clustering by regularized information maximization. In: Advances in Neural Information Processing Systems. pp. 775–783 (2010)
8. Li, X., Chen, Z., Poon, L.K., Zhang, N.L.: Learning latent superstructures in variational autoencoders for deep multidimensional clustering. arXiv preprint arXiv:1803.05206 (2018)
9. Mukherjee, S., Asnani, H., Lin, E., Kannan, S.: ClusterGAN: Latent space clustering in generative adversarial networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4610–4617 (2019)
10. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: Proceedings of the International Conference on Machine Learning. pp. 478–487 (2016)
11. Yang, L., Cheung, N.M., Li, J., Fang, J.: Deep clustering by gaussian mixture variational autoencoders with graph embedding. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6440–6449 (2019)