This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

Deep Adversarial Subspace Clustering

Pan Zhou* Yunqing Hou* Jiashi Feng*
* National University of Singapore, Singapore
pzhou@u.nus.edu elehouy@nus.edu.sg elefjia@nus.edu.sg

Abstract

Most existing subspace clustering methods hinge on self-expression of handcrafted representations and are unaware of potential clustering errors. Thus they perform unsatisfactorily on real data with complex underlying subspaces. To solve this issue, we propose a novel deep adversarial subspace clustering (DASC) model, which learns more favorable sample representations by deep learning for subspace clustering, and more importantly introduces adversarial learning to supervise sample representation learning and subspace clustering. Specifically, DASC consists of a subspace clustering generator and a quality-verifying discriminator, which learn against each other. The generator produces subspace estimation and sample clustering. The discriminator evaluates current clustering performance by inspecting whether the re-sampled data from estimated subspaces have consistent subspace properties, and supervises the generator to progressively improve subspace clustering. Experimental results on the handwritten recognition, face and object clustering tasks demonstrate the advantages of DASC over shallow and few deep subspace clustering models. Moreover, to our best knowledge, this is the first successful application of GAN-alike model for unsupervised subspace clustering, which also paves the way for deep learning to solve other unsupervised learning problems.

1. Introduction

In this paper, we aim to develop new deep learning solutions to the unsupervised subspace clustering problem. Compared with conventional "shallow" subspace clustering methods [5, 14, 34, 35] which are confined to linear subspaces, deep subspace clustering is obviously advantageous. It can provide more powerful sample representation through deep learning and effectively cluster samples from non-linear subspaces [9], which may greatly extend subspace clustering to more complicated real data.

Recently, a deep auto-encoder based subspace clustering model was proposed [9] aiming to learn better sample representations. However, like those conventional "shallow" methods [5, 14, 34, 35], it still hinges on the self-expression as supervision, which may not perform well on samples with unfriendly distributions (*e.g.* the intrinsic subspaces are not independent or have significant intersection). Moreover, existing subspace clustering methods [5, 9, 14, 34, 35, 38, 39] do not consider potential error in obtained clusters, leading to noisy learned representations and consequently degraded clustering performance.

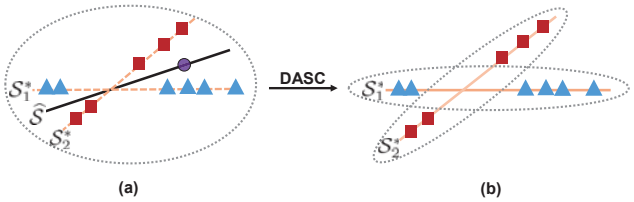


Figure 1: Illustration of our idea. Given the current clustering with error in (a), the discriminator in DASC can recognize its poor quality by differentiating the fake sample "◯" generated from current clustering (*i.e.* the estimated subspace $\widehat{\mathcal{S}}$ by generator) and the original "△" and "□" samples, as they have inconsistent subspace properties. With such evaluation information as supervision, the generator in DASC progressively improves sample representation learning and outputs correct clustering results in (b).

In this work, targeting at the above drawbacks of existing methods, we develop a novel unsupervised deep subspace clustering model following a GAN framework [7] due to its recent success in unsupervised data generation, which is termed Deep Adversarial Subspace Clustering (DASC). It consists of a subspace (and cluster) generator and a discriminator that learns to supervise the generator by evaluating clustering quality in an unsupervised manner. If the clustering produced by the generator is correct, the samples within the same cluster would all lie in the same intrinsic subspace, meaning their arbitrary linear combinations would also stay in the same subspace and have identical subspace properties, as stated in the well known linear subspace property. In this case, the discriminator cannot detect the difference between the re-sampled data from the intrinsic subspace and original samples in this cluster. Otherwise, given inaccurate clusters, as shown in Fig. 1 (a), the spanned subspace

$\widehat{\mathcal{S}}$ by the samples within the same cluster deviates from the intrinsic subspaces $\mathcal{S}_1^*$ and $\mathcal{S}_2^*$ that samples lie in. Then the discriminator can easily distinguish the re-sampled outlying data ("◯") drawn from the subspace $\widehat{\mathcal{S}}$ from those original data ("△" and "□"), and feeds back such information to the generator as supervision to produce better subspace estimation and sample clustering. Thus, the generator progressively improves subspace clustering, where the "△" and "□" are clustered correctly, as shown in Fig. 1 (b).

In particular, the role of the generator is three-fold. First, it uses a deep auto-encoder to transform raw input samples into better representations that are enforced to locate in a union of linear subspaces via a self-expression layer. In this way, DASC effectively relieves the linear subspace assumption on samples. Secondly, the generator produces subspace clustering results based on the sample affinity matrix produced by the internal self-expression layer. Thirdly, it generates new "fake" samples by sampling from the estimated clusters (or equivalently spanned subspaces) and feeds them to the discriminator for differentiation to evaluate the subspace clustering quality accordingly. On the other hand, the discriminator is trained to distinguish the generated "fake" samples from the provided real ones, and evaluates the quality of clustering results based on the linear subspace property. It feeds back the evaluation information to supervise the generator for subspace clustering. Different from conventional discriminators in GAN-alike models [7] providing over-complex classification boundaries to accurately leverage the subspace property to distinguish the samples, we propose an energy-based discriminator to detect real and outlying samples by inspecting how well they fit the subspace of interest. With supervision from such a discriminator, the generator will learn more discriminative representations and improve the clustering performance.

Experimental results on the handwritten recognition, face and object clustering tasks well testify the advantages of our method. To sum up, this paper makes the following contributions.

1) We propose a novel deep adversarial subspace clustering method, termed as DASC. By introducing adversarial learning, the discriminator of DASC can faithfully evaluate the current clustering quality and supervise the generator's learning to produce more favorable representations for better subspace clustering.
2) We design a simple but effective energy-based discriminator to extensively exploit the subspace property, which is novel and complementary to the auto-encoder induced self-expression loss.

2. Related Work

To date, various subspace clustering methods [2, 4, 5, 14, 32] have been developed. Most of them leverage self-expression to solve underlying subspaces and sample clustering, which can be written in the following uniform formulation:

$$\min_{\Theta_c} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{X}\Theta_c\|_F^2 + \lambda\|\Theta_c\|_p, \tag{1}$$

where the i-th column $\Theta_{c,i}$ in the self-expression coefficient matrix $\Theta_c \in \mathbb{R}^{n\times n}$ denotes the representation coefficients of the i-th data point $\boldsymbol{X}_i$ in the data matrix $\boldsymbol{X} \in \mathbb{R}^{d\times n}$. Here $\|\Theta_c\|_p$ denotes the prior term, *e.g.* the sparsity penalty term $\|\Theta_c\|_1$ [4, 5], the nuclear norm penalty term $\|\Theta_c\|_*$ [14], and the F-norm $\|\Theta_c\|_F$ [8, 15]. Then these methods perform normalized cuts (NCut) [24] or spectral clustering [19] on the affinity matrix $\Lambda = \frac{1}{2}(|\Theta_c| + |\Theta_c|^T)$ to cluster the data points.

However, they can only cluster linear subspaces, which limits their application. To solve this problem, kernel based subspace clustering methods [20, 33] have been developed. But it is difficult to choose a proper kernel capturing the underlying subspaces [9]. Moreover, all these existing methods only seek linear representation coefficients, which may not be discriminative for clustering tasks. In contrast, our proposed DASC uses a discriminator to effectively evaluate the clustering performance and supervise the generator to learn more discriminative representations.

Recently, several deep learning based clustering methods [3, 9, 25, 31] have been presented. Song *et al.* [25] integrated an auto-encoder [22] with k-means to learn and cluster the latent features. Similarly, Xie *et al.* [31] proposed a deep embedded clustering method. But neither of them is applicable to subspace clustering. More recently, Ji *et al.* [9] proposed a deep subspace clustering network (DSC-Net), which uses an auto-encoder to learn representations for input samples and obtain the linear representation coefficients (like Eqn. (1)) through a self-expressive layer. Although DSC-Net avoids the linear subspace assumption, it is not capable of self-tuning to learn better representations according to the current clustering results. Comparatively, the proposed DASC introduces adversarial learning between discriminator and generator to enforce the latter to improve the current clustering results by learning better representations.

3. Deep Adversarial Subspace Clustering

In this section we detailedly introduce the proposed method. We first explain the network formulation, then introduce each component, and finally elaborate its training and clustering process.

3.1. Formulation

As aforementioned, subspace clustering methods [5, 14, 34] cluster observed samples by recovering multiple low-dimensional subspaces to fit and separate them. However, for realistic data of complex natures, it is difficult to find

subspaces fitting their raw representations well. In this case, samples from different subspaces may be wrongly clustered into the same cluster, harming the clustering performance. To solve such an unsupervised learning problem, existing state-of-the-art methods (*e.g.*, SSC [5] and LRR [14]) typically hinge on the sample self-expression property within the same subspace, which usually do not perform well for samples with unfriendly distributions (*e.g.*, the intrinsic subspaces are not independent or have significant intersection). Besides, these methods do not consider potential error in obtained clusters and thus cannot improve themselves by evaluating the current clustering quality.

To address the above critical issues within a unified model, we propose the Deep Adversarial Subspace Clustering (DASC) model, which learns more favorable sample representations for subspace clustering via deep learning and introduces subspace adversarial learning to complement self-expression for better sample clustering. As illustrated in Fig. 2, it consists of a subspace (and cluster) generator and a discriminator, respectively denoted by G and D. For brevity, let $\{\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n\}$ denote the input samples and $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n\}$ denote their corresponding latent representations learned by the encoder in G. Namely, $\boldsymbol{z}_i \in \mathbb{R}^d$ is the d-dimensional representation of the i-th 2D sample $\boldsymbol{X}_i \in \mathbb{R}^{h \times w}$. The cluster number, *i.e.*, number of subspaces, is denoted by K.

Specifically, DASC learns the latent representations $\boldsymbol{z}_i$ for the input samples $\boldsymbol{X}_i$ through a multi-layer non-linear encoder in its generator G. To enforce the new representations $\boldsymbol{z}_i$ to be more suitable for subspace clustering than the raw representation $\boldsymbol{X}_i$, G introduces a self-expressive layer and minimizes the following self-expression loss:

$$\mathcal{L}_s(\Theta_c; \boldsymbol{Z}) = \|\boldsymbol{Z} - \boldsymbol{Z}\Theta_c\|_F^2 + \lambda\|\Theta_c\|_F^2, \tag{2}$$

where $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_n] \in \mathbb{R}^{d \times n}$ and $\Theta_c \in \mathbb{R}^{n \times n}$ is the self-expression coefficient matrix. Benefiting from the more powerful representations $\boldsymbol{z}_i$, applying a spectral clustering algorithm (*e.g.*, NCut [24]) on the induced affinity matrix $\Lambda = \frac{1}{2}(|\Theta_c| + |\Theta_c^T|)$ gives reasonably good sample clusters C_i $(i = 1, \ldots, K)$.

To address complex sample distributions (*e.g.*, intersected subspaces) and improve subspace clustering results, DASC introduces adversarial learning as a novel and complementary unsupervised solution. Concretely, let $\mathcal{S}_i^*$ be the i-th ground-truth subspace to recover and $\{\boldsymbol{z}_{i_1}, \ldots, \boldsymbol{z}_{i_{m_i^*}}\}$ be the representations of m_i^* authentic samples lying in $\mathcal{S}_i^*$. We have $\text{span}(\boldsymbol{z}_{i_1}, \ldots, \boldsymbol{z}_{i_{m_i^*}}) = \mathcal{S}_i^*$ and any linear combinations of $\boldsymbol{z}_i$ still lie in $\mathcal{S}_i^*$. When the obtained cluster C_i is not accurate due to the complex distribution, samples within C_i, denoted as $\{\boldsymbol{z}_{i_1}, \ldots, \boldsymbol{z}_{i_{m_i^*}}, \boldsymbol{z}_{i_{m_i^*+1}}, \ldots, \boldsymbol{z}_{i_{m_i}}\}$, would span a subspace $\widehat{\mathcal{S}}_i = \text{span}(C_i)$ deviating from $\mathcal{S}_i^*$. In other words, linearly combining random samples from

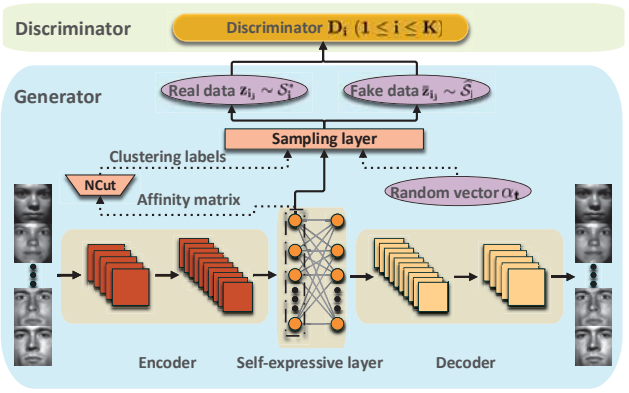


Figure 2: Illustration of overall architecture. DASC consists of a generator and a discriminator that learn against each other. In particular, the generator contains an auto encoder-decoder for learning sample representations, a self-expressive layer for producing sample affinity matrix and clustering, and a sampling layer for generating real and fake data for subspace quality evaluation. The subspace-wise discriminator takes in the generated samples and learns to distinguish real data from fake ones.

C_i will generate samples lying out of $\mathcal{S}_i^*$ which are different from $\boldsymbol{z}_{i_1}, \ldots, \boldsymbol{z}_{i_{m_i^*}}$.

DASC aims to inspect such differences between regenerated samples from noisy C_i and authentic samples from $\mathcal{S}_i^*$ to define an effective and computable metric for the cluster quality and obtain extra supervision. Formally, DASC introduces a novel discriminative loss to measure the quality of the cluster C_i:

$$\xi(C_i) := \mathbb{E}_{\boldsymbol{z}_{i_j} \sim C_i} \log(D_i(\boldsymbol{z}_{i_j})), \tag{3}$$

where D_i is a discriminative model for the subspace $\mathcal{S}_i^*$ and trained by minimizing the following discriminative loss:

$$\mathcal{L}_D = \mathbb{E}_{\bar{\boldsymbol{z}}_{i_j} \sim \widehat{\mathcal{S}}_i} \log(D_i(\bar{\boldsymbol{z}}_{i_j})) - \mathbb{E}_{\boldsymbol{z}_{i_j} \sim \mathcal{S}_i^*} \log(D_i(\boldsymbol{z}_{i_j})). \tag{4}$$

Here $\sim$ denotes a composite operation including sampling and linear combination, and will be explained in more details in Sec. 3.2. The sampling and linear combination operations are important, which are inspired by the key property of a subspace—linearly combining samples within a valid subspace would not generate outlying samples. The discriminator D_i in DASC is trained to distinguish the "real" samples from the ground-truth intrinsic subspace $\mathcal{S}_i^*$ and "fake" samples from its estimate $\widehat{\mathcal{S}}_i$. Different from conventional discriminators in GAN-alike models, the discriminator in DASC needs to estimate the probability of each sample belonging to a certain subspace. Namely, it needs to be subspace-wisely discriminative. We introduce a new energy based discriminator model in Sec. 3.3 to fulfill such a critical requirement.

If all the "fake" samples (up to linear combination) from the cluster C_i are classified into $\mathcal{S}_i^*$ by the discriminator D, *i.e.*, a large value for the quantity in Eqn. (3), then the cluster C_i has a high quality. Otherwise, the clusters are erroneous. By feeding back such supervision to the generator G, G is then enforced to maximize the quality or equivalently the discriminator loss in Eqn. (4), leading to better cluster sample representation and clustering results, *i.e.* the samples within C_i being distinguishable to the discriminator D. In the following subsections, we explain details on the generator G and the discriminator D, to implement methods introduced above.

3.2. Generator

The generator G of DASC learns discriminative representations that are favorable for subspace clustering, by exploiting the low-dimensional structure of samples, and generates subspace clustering results. In particular, it learns to transform raw input samples into a latent representation space where samples can be fitted well by a union of linear subspaces, clusters the samples using their subspace memberships, and produces "real" and "fake" samples from each cluster as explained above.

As shown in Fig. 2, the generator uses a deep convolutional auto-encoder to non-linearly transform the samples $\boldsymbol{X}_i$ into representations $\boldsymbol{z}_i$. Then it uses a self-expressive layer following the encoder to produce self-expression coefficients Θ_c (see Eqn. (2)). The new representations $\boldsymbol{Z}\Theta_{c,i}$ are subsequently fed into the decoder. Here the decoder has a symmetrical structure to the encoder, which aims to reconstruct the original input $\boldsymbol{X}_i$ from $\boldsymbol{Z}\Theta_c$ to ensure that the representation $\boldsymbol{Z}$ preserve sufficient sample information. The affinity matrix $\Lambda = \frac{1}{2}(|\Theta_c| + |\Theta_c^T|)$ is used for clustering.

Another important function of G is to generate "real" and "fake" samples conditioned on the cluster C_i $(i = 1, \ldots, K)$, which is implemented by the sampling layer in Fig. 2. Based on the sample affinity matrix Λ learned by the self-expressive layer, we apply the NCut algorithm [24] to cluster $\boldsymbol{z}_i$ $(i = 1, \cdots, n)$ into K clusters C_i's. Meanwhile, as shown in Fig. 3, our discriminator is designed to learn a linear subspace $\mathcal{S}_i$ to fit the intrinsic ground-truth subspace $\mathcal{S}_i^*$ of cluster C_i. Then according to the projection residuals (see Eqn. (5) in Sec. 3.3) of data points on their corresponding subspaces learned by the discriminator, the discriminator can identify whether the input data are real or fake. See details of the discriminator in Sec. 3.3. Accordingly, the sampling layer in G computes the projection residual onto $\mathcal{S}_i$ for each sample in the cluster C_i and selects $\bar{m}_i^*$ "real" data with smaller residuals (flavescent points in Fig. 3). See the setting of $\bar{m}_i^*$ in Sec. 3.4. In this way, the selected samples approximately lie in the correct intrinsic subspace with high probability and serve as "real data" for spanning the subspace $\mathcal{S}_i$. Moreover, the discriminator is powerful enough to handle the possible noise in these selected real data. Such a novel projection residual based strategy for sampling real data is adopted to closely match the energy-based discriminator for clustering quality evaluation and its validity is verified by our experiments.

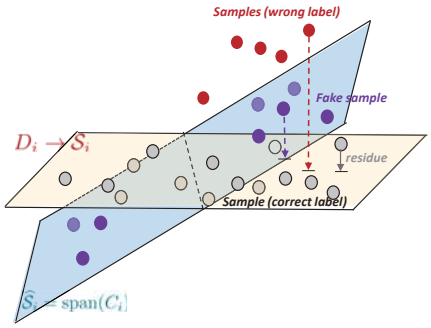


Figure 3: Illustration on real/fake sampling. Real data (with correct clustering label) and fake data are sampled from an erroneous cluster C_i which spans an inaccurate subspace estimation $\widehat{\mathcal{S}}_i$. Real data have much smaller projection residuals onto the subspace $\mathcal{S}_i$ learned by the discriminator D_i than sampled fake data, thus they can be distinguished by D_i. Best viewed in color.

To produce fake data, for each cluster C_i, the sampling layer in the generator performs random sampling from the estimated subspace $\widehat{\mathcal{S}}_i$ (see purple point in Fig. 3). Since directly sampling is non-differentiable, we employ the reparameterization trick in [11] to enable differentiable sampling. Concretely, for each cluster C_i of m_i samples, the sampling layer first samples $\bar{m}_i^*$ random vectors $\alpha_t \in \mathbb{R}^{m_i}$ from the uniform distribution within $(0, 1]$ and then generates $\bar{m}_i^*$ fake data as $\bar{\boldsymbol{z}}_t = \sum_{j=1}^{m_i} \alpha_{tj} \boldsymbol{z}_{i_j}$ $(t = 1, \ldots, \bar{m}_i^*)$, where α_{tj} denotes the j-th entry in α_t. As for the current training state α_ts are fixed, the gradient can be propagated to the encoder through the representations $\boldsymbol{z}_{i_j}$s. Compared with explicitly computing $\widehat{\mathcal{S}}_i$ and then sampling fake data, producing fake data as above is more efficient.

3.3. Discriminator

We build the discriminator to implement an energy function which assigns low energy to the regions near the data subspace and higher energy to other regions. Since for each cluster C_i, the discriminator D aims to verify whether its real data $\boldsymbol{z}_i$ and fake data $\bar{\boldsymbol{z}}_i$ belong to the same intrinsic subspace, it only needs to learn a subspace discriminative model that fits the desired intrinsic subspace for each cluster. For cluster C_i, let the basis of the subspace $\mathcal{S}_i$ learned by the discriminator be parameterized by $\boldsymbol{U}_i \in \mathbb{R}^{d \times \boldsymbol{r}_i}$ where $\boldsymbol{r}_i$ denotes the subspace dimension. Then as shown in Fig. 3, the discriminator can distinguish real data from fake ones by their projection residuals onto $\boldsymbol{U}_i$:

$$\mathcal{L}_r(\boldsymbol{z}_{i_j}) = \|\boldsymbol{z}_{i_j} - \boldsymbol{U}_i \boldsymbol{U}_i^T \boldsymbol{z}_{i_j}\|_2^2, \tag{5}$$

as real samples will be closer to the subspace than those fake ones if the clusters are not accurate. We use the above projection residual to define the probability of a sample belonging to subspace $\mathcal{S}_i$ which is the output of the discriminator D_i, namely, $D_i(\boldsymbol{z}_{i_j}) = \mathbb{P}(\boldsymbol{z}_{i_j} \in \mathcal{S}_i) \propto \exp(-\mathcal{L}_r(\boldsymbol{z}_{i_j}))$. Substituting it to Eqn. (4) gives the following objective to train the discriminator D_i of cluster C_i:

$$\min_{\boldsymbol{U}_i} \mathcal{L}_{D_i} := \frac{1}{\bar{m}_i^*} \sum_{j=1}^{\bar{m}_i^*} \mathcal{L}_r(\boldsymbol{z}_{i_j}) + [\varepsilon - \mathcal{L}_r(\bar{\boldsymbol{z}}_{i_j})]_+, \tag{6}$$

where $[\cdot]_+ = \max(0, \cdot)$ is an additional margin loss with a small positive margin parameter ε, inspired by [37]. Considering all K clusters, the objective for training the discriminator is

$$\min_{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_K} \mathcal{L}_D := \frac{1}{K} \sum_{i=1}^{K} \mathcal{L}_{D_i}, \tag{7}$$

where each column $\boldsymbol{U}_{i,j}$ in all $\boldsymbol{U}_i$ $(i = 1, \cdots, K)$ obeys $\|\boldsymbol{U}_{i,j}\|_2^2 = 1$. To establish one-to-one correspondence between the cluster C_i and $\boldsymbol{U}_i$ to compute the loss $\mathcal{L}_D$, for all candidates $\boldsymbol{U}_i$, each C_i computes its average projection residual onto them and chooses the one of smallest average projection residual $\frac{1}{|C_i|} \sum_{\boldsymbol{z}_i \in C_i} \|\boldsymbol{z}_i - \boldsymbol{U}_j \boldsymbol{U}_j^T \boldsymbol{z}_i\|_2^2$. If multiple clusters compete for the same $\boldsymbol{U}_i$, $\boldsymbol{U}_i$ will choose the one with smallest average projection. For any cluster without a matched $\boldsymbol{U}_i$, we use the QR decomposition on its feature matrix constituted by $\boldsymbol{z}_{i_j}$s to compute its corresponding $\boldsymbol{U}_i$. In this way, learning of the generator and discriminator is consistent, since at each time the cluster C_i will always find the $\boldsymbol{U}_j$ that is closest to its intrinsic subspace.

To promote separability of subspaces corresponding to different clusters, we introduce the regularization term $R_1 = \beta_1 \sum_{i \neq j} \|\boldsymbol{U}_i^T \boldsymbol{U}_j\|_F^2$, where $\beta_1 > 0$ is a constant. It also benefits clustering, via the discriminator D, by encouraging the generator G to produce representations discriminative for different subspaces. Although we do not require the basis in $\boldsymbol{U}_i$ to be strictly orthonormal, we still use $R_2 = \beta_2 \sum_{i=1}^{K} \|\boldsymbol{U}_i^T \boldsymbol{U}_i - \boldsymbol{I}\|_F^2$ as regularization to reduce redundancy in each $\boldsymbol{U}_i$, where $\beta_2 > 0$ is a constant and $\boldsymbol{I}$ is the identity matrix with compatible dimensions.

Thus for the discriminator, the final training objective is

$$\min_{\boldsymbol{U}_1,\cdots,\boldsymbol{U}_K} \mathcal{L}_D + R_1 + R_2. \tag{8}$$

The discriminator in DASC can be implemented by K linear networks of two fully connected layers. Besides, in each network the two layers share their parameters $\boldsymbol{U}_i$. For input data $\boldsymbol{z}_j$, the outputs of the first and second layers are respectively $\boldsymbol{U}_i \boldsymbol{z}_j$ and $\boldsymbol{U}_i \boldsymbol{U}_i^T \boldsymbol{z}_j$. By minimizing the discriminator loss, the parameter $\boldsymbol{U}_i$ $(1 \leq i \leq K)$ can be learned.

Obviously, the discriminator in DASC is a variant of the auto-encoder yet with more simplicity than the ones used for image generation in [1, 37], since ours has only two linear mapping layers. Such appealing simplicity is benefited from the self-expressive layer in G, which offers convenience to utilize linear subspace structure to design the discriminator. In contrast, the auto-encoders in [1, 37] are much deeper and involve non-linear mappings, since the data structure remains unknown and thus complex auto-encoders are necessary to learn the data structure.

3.4. Training and Clustering

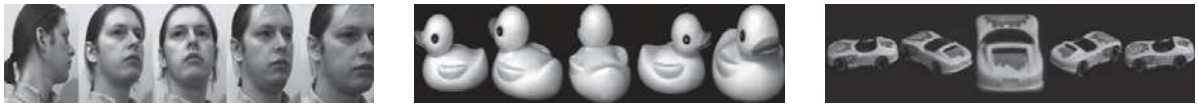
We are ready to define training objective of the generator G. Following the adversarial learning scheme, we encourage the generator G to minimize $\mathcal{L}_a = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{\bar{m}_i^*} \sum_{j=1}^{\bar{m}_i^*} \mathcal{L}_r(\bar{\boldsymbol{z}}_{i_j})$, *i.e.*, to encourage the generated fake data to be closer to the subspace learned by the discriminator indicating higher clustering quality through tuning the representation learning and subspace clustering results from G. Combining this adversarial loss $\mathcal{L}_a$ with the sample reconstruction loss and the self-expression loss gives the final training objective of G:

$$\begin{aligned} \min_{\Theta} \mathcal{L}_G(\Theta) := \mathcal{L}_a + \lambda_1 \|\boldsymbol{X} - \widehat{\boldsymbol{X}}\|_F^2 \\ + \lambda_2 \|\boldsymbol{Z} - \boldsymbol{Z}\Theta_c\|_F^2 + \lambda_3 \|\Theta_c\|_F^2, \end{aligned} \tag{9}$$

where $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots, \boldsymbol{X}_n]$ denotes all the input samples and $\widehat{\boldsymbol{X}}$ denotes the reconstruction of $\boldsymbol{X}$ by the auto-decoder. Here Θ denotes the parameters of the generator, including the parameters in encoder and decoder, and the representation coefficients Θ_c in the self-expressive layer as well. The second term denotes the reconstruction loss of the auto-encoder, while the last two terms correspond to self-expressive loss $\mathcal{L}_s(\Theta_c; \boldsymbol{Z})$ in Eqn. (2). F-norm penalty on Θ_c is adopted, since compared with the non-smooth penalty term, *e.g.* the ℓ_1 norm, it can be learned more easily and can also achieve comparable or even better performance [9].

We pre-train the generator G without considering the discriminator D at first, *i.e.*, minimizing the generator loss in Eqn. (9) while discarding the first term $\mathcal{L}_a$. In this way, the generator can produce reasonably good initial representations. Then we train the whole DASC network as follows. First, we initialize the discriminator parameters $\boldsymbol{U}_i \in \mathbb{R}^{d \times \boldsymbol{r}_i}$ by the following method. See Sec. 4 for the setting of $\boldsymbol{r}_i$. We apply NCut on the affinity matrix Λ to cluster the latent representations. Then we use the representations falling in cluster C_i to compute $\boldsymbol{U}_i$ via QR decomposition. As the pre-trained generator gives relatively good clustering results, such a strategy initializes the model better than random initialization. During the joint training of discriminator and generator, as the architecture of the discriminator is much simpler than the generator, we asynchronously update D and G for 5 times and 1 time within each epoch respectively. For cluster C_i of m_i data points, we set the number $\bar{m}_i^*$ of real and fake data to αm_i $(\alpha \in [0.8, 0.95])$ which

(a) Mnist (b) ORL (c) Extended YaleB



(d) Umist (e) COIL-20 (f) COIL-100

Dataset	# Class	# Images/class	Total #	Size	Difficulty
Mnist	10	100	1000	28×28	deformation
ORL	40	10	400	32×32	deformation, pose and expression
YaleB	38	64	2432	48×42	illumination
Umist	20	24	480	32×32	deformation and very different pose
COIL-20	20	72	1400	32×32	deformation and rotation
COIL-100	100	72	7200	32×32	deformation and rotation

(g) Statistics of the evaluation datasets.

Figure 4: Examples and descriptions of the six testing datasets.

Table 1: Clustering results on Mnist.

Metric	SSC	ENSC	KSSC	SSC-OMP	EDSC	LRR	LRSC	AE+SSC	DSC-Net-L_1	DSC-Net-L_2	DASC
ACC	0.4530	0.4983	0.5220	0.3400	0.5650	0.5386	0.5140	0.4840	0.7280	0.7500	**0.8040**
NMI	0.4709	0.5495	0.5623	0.3272	0.5752	0.5632	0.5576	0.5337	0.7217	0.7319	**0.7800**
PUR	0.4940	0.5483	0.5810	0.3560	0.6120	0.5684	0.5550	0.5290	0.7890	0.7991	**0.8370**

Table 2: The channel number of generator in DASC used for ORL, Extended YaleB, and Umist.

Dataset	encoder-1 /decoder-3	encoder-2 /decoder-2	encoder-3 /decoder-1
ORL	5	3	3
YaleB	10	20	30
Umist	15	10	5

works well in all experiments. Both pre-training and fine-tuning adopt the Adam algorithm [10]. But their learning rates are 10^{-3}, the same as [9], and 2×10^{-4} respectively.

During testing, we perform spectral clustering on the learned affinity matrix Λ output by the generator G. For fairness, we use the NCut algorithm as in [9].

4. Experiments

We evaluate the clustering performance of our proposed DASC with three subspace clustering tasks: handwritten digit recognition, face clustering and object clustering. Among them, the first two tasks are relatively easier, since handwriting and face images approximately lie on a union of linear subspaces [4, 5, 28, 29, 36, 40, 41]. We compare DASC with state-of-the-art subspace clustering methods, including sparse subspace clustering (SSC) [4,5], elastic net subspace clustering (ENSC) [34], kernel SSC (KSSC) [21], SSC by orthogonal matching pursuit (SSC-OMP) [35], efficient dense subspace clustering (EDSC) [8], low-rank representation (LRR) [14], low-rank subspace clustering (LRSC) [26], the latest deep subspace clustering network (DSC-Net) [9], and SSC with pre-trained convolutional auto-encoder features. For all baselines, we use their released source codes and tune their performance to be best.

In our experiments, we fix parameters of DASC as $\beta_1 = \beta_2 = 0.01$, $\lambda_1 = 0.5$ and $\varepsilon = 0.1$. If the encoder of the generator in DASC has three layers, their kernel sizes are always set to 5×5, 3×3 and 3×3, respectively. For other encoders of different layers, their kernel size setting will be introduced in the corresponding section. In all experiments, the stride of these kernels is fixed as 2. The decoder always has a symmetrical structure to the encoder. We use ReLU [12] as the non-linear activations. For fair comparison, the DSC-Net (DSC-Net-L_1 and DSC-Net-L_2) always adopts the same architecture as our generator in DASC, including its encoder, self-expressive layer and decoder. We adopt following popular clustering metrics to measure the clustering performance: accuracy (ACC), normalized mutual information (NMI) [27] and purity (PUR) [16].

4.1. Handwritten Digit Recognition

We evaluate DASC on the Mnist dataset [13] for handwritten digit recognition. We randomly select 100 images for each digit, resulting in a subset of 1,000 images. Examples are shown in Fig. 4. Both the encoder and decoder in the generator G of DASC have three layers which in the encoder respectively have 20, 10 and 5 channels. The output

Table 3: Clustering results on ORL, Extended YaleB, and Umist.

Dataset	Metric	SSC	ENSC	KSSC	SSC-OMP	EDSC	LRR	LRSC	AE+SSC	DSC-Net-L$_1$	DSC-Net-L$_2$	DASC
ORL	ACC	0.7425	0.7525	0.7143	0.7100	0.7038	0.8100	0.7200	0.7563	0.8550	0.8600	**0.8825**
	NMI	0.8459	0.8540	0.8070	0.7952	0.7799	0.8603	0.8156	0.8555	0.9023	0.9034	**0.9315**
	PUR	0.7875	0.7950	0.7513	0.7463	0.7138	0.8225	0.7542	0.7950	0.8585	0.8625	**0.8925**
YaleB	ACC	0.7354	0.7537	0.6921	0.7372	0.8814	0.8499	0.7931	0.7480	0.9681	0.9733	**0.9856**
	NMI	0.7796	0.7915	0.7359	0.7803	0.8835	0.8636	0.8264	0.7833	0.9687	0.9703	**0.9801**
	PUR	0.7467	0.7654	0.7183	0.7542	0.8800	0.8623	0.8013	0.7597	0.9711	0.9731	**0.9857**
Umist	ACC	0.6904	0.6931	0.6531	0.6438	0.6937	0.6979	0.6729	0.7042	0.7242	0.7312	**0.7688**
	NMI	0.7489	0.7569	0.7377	0.7068	0.7522	0.7630	0.7498	0.7515	0.7556	0.7662	**0.8042**
	PUR	0.6554	0.6628	0.6256	0.6171	0.6683	0.6670	0.6562	0.6785	0.7204	0.7276	**0.7688**

dimension of the encoder in G is 80. We set basis number $\boldsymbol{r}_i$ in $\boldsymbol{U}_i$ $(i = 1, \cdots, K)$ to 10, $\lambda_2 = 0.1$ and $\lambda_3 = 1.0$.

Table 1 summarizes the clustering results on Mnist. DASC outperforms the baselines in all three metrics. Specifically, it improves over the second best DSC-Net-L$_2$ by 5.40%, 4.81% and 3.79% in terms of ACC, NMI and PUR, respectively. Moreover, DASC achieves much better clustering results than the shallow subspace clustering methods, *e.g.*, SSC and LRR. This is because compared with shallow methods, DASC uses a multi-layer convolutional auto-encoder as the feature extractor. So DASC can well handle translation, rotation and shifting in the handwritten images (see Fig. 4 (a)) and map input data into a union of linear subspaces. Besides, the adversarial learning in DASC is effective at benefiting the representation learning and clustering. Concretely, it improves about 5.40% over its baseline DSC-Net on the commonly used ACC metric. This shows DASC can improve itself by quantifying the current clustering results.

4.2. Face Clustering

We then evaluate DASC on three widely used face databases: ORL [23], Extended YaleB (YaleB for short) [6], and Umist [30]. As shown in Fig. 4 (b)-(d), these datasets are challenging due to their different properties. For the 40 subjects in ORL, each category has only 10 face images taken with varying poses and expressions. In comparison, YaleB is relatively simpler,containing 38 subjects and 64 near frontal images per subject under different illumination. Although Umist only contains 20 persons, each with only 24 images is taken under very different poses. For these datasets, both the encoder and decoder in the generator G have three layers. Their architecture details are given in Table 2. For ORL, YaleB and Umist, the outputs of the encoder in G have respectively 80, 1,080 and 80 dimensions. We set the number $\boldsymbol{r}_i$ of basis in $\boldsymbol{U}_i$ as 10, $\lambda_3 = 1$, and respectively set λ_2 as 0.1, 3.0 and 0.1 for the three datasets.

Table 3 reports clustering results on these datasets. One can observe that DASC consistently outperforms the baselines for all three metrics. On the ORL and Umist datasets, DASC respectively improves by 2.25% and 3.76% over the second best DSC-Net-L$_2$ on ACC. For both NMI and PUR metrics, DASC also brings about 3% improvement on ORL and improves by about 4% on Umist over the state-of-the-arts. As aforementioned, compared with ORL and Umist, YaleB is relatively simpler (see Fig. 4) and all methods perform very well. However, DASC still brings about 1.23% improvement even though state-of-the-art accuracy on YaleB is as high as 97.33%. All these results clearly prove the superior effectiveness and robustness of DASC.

These results also clearly demonstrate that deep clustering methods except AE+SSC perform much better than the shallow ones, benefiting from integrating representation learning with self-expression learning. The deep auto-encoder extracts more powerful representations and the following self-expression layer enforces the representations to favorably locate in a union of linear subspaces, effectively getting rid of strict linear subspace assumptions. Comparatively, DASC outperforms the DSC-Net, including both DSC-Net-L$_1$ and DSC-Net-L$_2$, on the three testing datasets w.r.t. all metrics. This outstanding performance is attributed to the adversarial learning between generator and discriminator in DASC. Unlike DASC and DSC-Net, the AE+SSC method does not benefit much from using a deep auto-encoder. This is because the deep auto-encoder only considers the reconstruction of original images, and it does not guarantee the latent feature to lie in the linear subspaces without the self-expressive layer. Meanwhile, although SSC works well on data lying in linear subspaces, it cannot well handle data from non-linear subspaces. So it does not gain much benefit from the deep auto-encoder either.

4.3. Object Clustering

Here we evaluate DASC on the most challenging object clustering task, using the COIL-20 [18] and COIL-100 [17] datasets which provide various objects as shown in Fig. 4 (e)-(f). COIL-20 has 1,440 toy images from 20 classes, and COIL-100 contains 7,200 images for 100 objects. In both datasets, each object is taken with poses varying at an interval of 5 degrees, producing totally 72 images per object.

Table 4: Clustering results on COIL-20 and COIL-40.

Dataset	Metric	SSC	ENSC	KSCC	SSC-OMP	EDSC	LRR	LRSC	AE+SSC	DSC-Net-L_1	DSC-Net-L_2	DASC
COIL-20	ACC	0.8631	0.8760	0.7087	0.6410	0.8371	0.8118	0.7416	0.8711	0.9314	0.9368	**0.9639**
	NMI	0.8892	0.8952	0.8243	0.7412	0.8828	0.8747	0.8452	0.8990	0.9353	0.9408	**0.9686**
	PUR	0.8747	0.8892	0.7497	0.6667	0.8585	0.8361	0.7937	0.8901	0.9306	0.9397	**0.9632**
COIL-40	ACC	0.7191	0.7426	0.6549	0.4431	0.6870	0.6493	0.6327	0.7391	0.8003	0.8075	**0.8354**
	NMI	0.8212	0.8380	0.7888	0.6545	0.8139	0.7828	0.7737	0.8318	0.8852	0.8941	**0.9196**
	PUR	0.7716	0.7924	0.7284	0.5250	0.7469	0.7109	0.6981	0.7840	0.8646	0.8740	**0.8972**

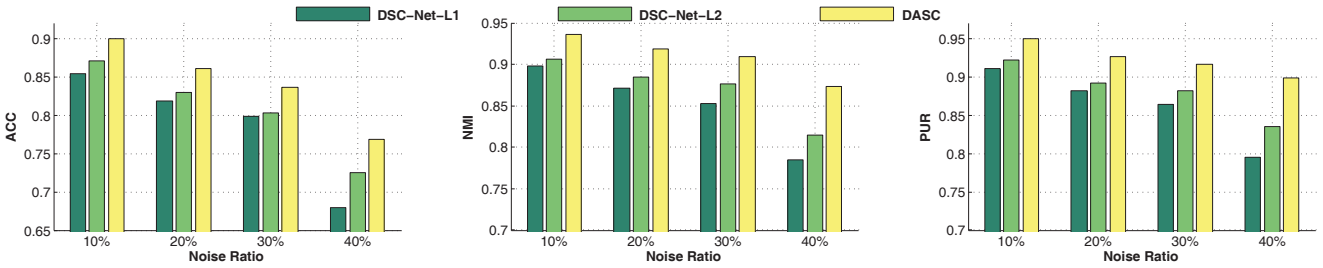


Figure 5: Clustering results on the noisy COIL-20.

This implies that the images are not distributed in a union of linear subspaces and thus are more challenging. Due to computational memory limit, we select the first 40 classes in COIL-100 with totally 2,880 images for evaluation, which we call COIL-40. For both COIL-20 and COIL-40, the generator encoders in DASC have only one layer of 15 and 20 channels respectively. The kernel size is 3×3. Accordingly, the output feature dimension of the generator encoder are respectively 3,840 and 5,120 on COIL-20 and COIL-40. We uniformly set the basis number $\boldsymbol{r}_i$ in $\boldsymbol{U}_i$ as 30, and set $\lambda_2 = 15.0$ and $\lambda_3 = 1.0$.

Table 4 reports the results. One can observe that DASC achieves the best clustering performance. Specifically, compared with shallow subspace clustering methods on COIL-20, it brings about 8.79%, 7.34% and 7.40% improvement over the best shallow method ENSC in terms of ACC, NMI and PUR metrics. On the more challenging COIL-40, it improves by 9.28%, 8.16% and 10.48% respectively. These results clearly verify that deep solution offered by DASC to subspace clustering is more favorable and effective. Compared with another deep subspace clustering method DSC-Net, on COIL-20 our DASC outperforms it by about 2.71%, 2.78% and 2.35% on ACC, NMI and PUR respectively, and on COIL-40 it improves 2.79%, 2.55% and 2.32%. As they share the same network for latent representation learning, the better performance of DASC is benefited from adversarial learning which is effective at providing complementary supervision for clustering improvement.

Finally, to more comprehensively compare our proposed DASC with DSC-Net, we evaluate them in a noisy scenario. For each image in COIL-20, we respectively randomly convert $10\% \sim 40\%$ of pixels to random values in $[0, 255]$ for evaluation. As reported in Fig. 5, w.r.t. different noise ratios, our DASC always achieves the best performance in terms of all three metrics. When the noise ratio is $10\% \sim 30\%$, DASC respectively brings about at least 2.97%, 3.10% and 3.32% over the second-best DSC-Net-L_1 in terms of ACC. When the noise ratio increases to 40%, DASC improves by about 4.32%. On NMI and PUR, DASC makes at least 2.99% and 3.82% improvements, respectively. Compared with noiseless cases, such improvements are more notable. This is because DASC can handle more complex distributions incurred by noise benefiting from adversarial learning, while the deep auto-encoder in absence of the adversarial learning generally fails which results in the quick drop of the performance of DSC-Net.

5. Conclusion

We proposed a novel deep adversarial subspace clustering (DASC) model. It adopts adversarial learning to effectively supervise sample representation learning and subspace clustering. The discriminator of DASC evaluates the current clustering performance and feeds back the evaluation information to the generator to produce better sample representations and subspace clustering. Extensive experimental results demonstrated the superior advantages of DASC on subspace clustering problems over state-of-the-arts, including the latest deep learning based method.

Acknowledgements

Jiashi Feng was partially supported by NUS startup R-263-000-C08-133, MOE Tier-I R-263-000-C21-112, NUS IDS R-263-000-C67-646 and ECRA R-263-000-C87-133.

References

[1] D. Berthelot, T. Schumm, and L. Metz. BEGAN: Boundary equilibrium generative adversarial networks. *ICLR*, 2017.

[2] B. Cheng, G. Liu, J. Wang, Z. Huang, and S. Yan. Multi-task low-rank affinity pursuit for image segmentation. In *IEEE ICCV*, pages 2439–2446. IEEE, 2011.

[3] N. Dilokthanakul, P. Mediano, M. Garnelo, M. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. In *ICLR*, 2017.

[4] E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE CVPR*, 2009.

[5] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE TPAMI*, 35(11):2765–2781, 2013.

[6] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE TPAMI*, 23(6):643–660, 2001.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[8] P. Ji, M. Salzmann, and H. Li. Efficient dense subspace clustering. In *IEEE Winter Conference on Applications of Computer Vision*, pages 461–468. IEEE, 2014.

[9] P. Ji, T. Zhang, H. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In *NIPS*, 2017.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

[11] D. Kingma and M. Welling. Auto-encoding variational bayes. *ICLR*, 2013.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[14] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, 2010.

[15] C. Lu, H. Min, Z. Zhao, L. Zhu, D. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. *ECCV*, pages 347–360, 2012.

[16] C. Manning, P. Raghavan, and H. Schutze. Introduction to information retrieval. *Cambridge University Press*, 2010.

[17] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-100). 1996.

[18] S. Nene, S. Nayar, and H. Murase. Columbia object image library (COIL-20). 1996.

[19] A. Y. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002.

[20] V. Patel, H. V. Nguyen, and R. Vidal. Latent space sparse subspace clustering. In *IEEE ICCV*, pages 225–232, 2013.

[21] V. Patel and R. Vidal. Kernel sparse subspace clustering. In *IEEE ICIP*, pages 2849–2853. IEEE, 2014.

[22] C. Poultney, S. Chopra, and Y. Cun. Efficient learning of sparse representations with an energy-based model. In *NIPS*, pages 1137–1144, 2007.

[23] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, pages 138–142. IEEE, 1994.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[25] C. Song, F. Liu, Y. Huang, L. Wang, and T. Tan. Auto-encoder based data clustering. In *Iberoamerican Congress on Pattern Recognition*, pages 117–124. Springer, 2013.

[26] R. Vidal and P. Favaro. Low rank subspace clustering (LRSC). *Pattern Recognition Letters*, 43:47–61, 2014.

[27] N. Vinh, J. Epps, and J. Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *JMLR*, 11:2837–2854, 2010.

[28] Y. Wang, C. Xu, C. Xu, and D. Tao. Beyond RPCA: Flattening complex noise in the frequency domain. In *AAAI*, 2017.

[29] Y. Wang, C. Xu, S. You, C. Xu, and D. Tao. DCT regularized extreme visual recovery. *IEEE TIP*, 26(7):3360–3371, 2017.

[30] J. Woodall, M. Agúndez, A. Markwick-Kemper, and T. Millar. The UMIST database for astrochemistry 2006. *Astronomy & Astrophysics*, 466(3):1197–1204, 2007.

[31] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016.

[32] A. Yang, J. Wright, Y. Ma, and S. Sastry. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.

[33] M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *IEEE CVPR*, pages 5157–5164, 2016.

[34] C. You, C. Li, D. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *IEEE CVPR*, pages 3928–3937, 2016.

[35] C. You, D. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *IEEE CVPR*, pages 3918–3927, 2016.

[36] H. Zhang, Z. Lin, C. Zhang, and J. Gao. Robust latent low rank representation for subspace clustering. *Neurocomputing*, 145:369–373, 2014.

[37] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *ICLR*, 2017.

[38] P. Zhou, C. Fang, Z. Lin, C. Zhang, and E. Chang. Dictionary learning with structured noise. *Neurocomputing*, 2017.

[39] P. Zhou and J. Feng. Outlier-robust tensor PCA. In *IEEE CVPR*, 2017.

[40] P. Zhou, Z. Lin, and C. Zhang. Integrated low-rank-based discriminative feature learning for recognition. *IEEE TNNLS*, 27(5):1080–1093, 2016.

[41] P. Zhou, C. Zhang, and Z. Lin. Bilevel model based discriminative dictionary learning for recognition. *IEEE TIP*, 26(3):1173–1187, 2017.