

# Adversarial Domain Adaptive Subspace Clustering

Mahdi Abavisani  
Rutgers University  
Piscataway, New Jersey, USA  
mahdi.abavisani@rutgers.edu

Vishal M. Patel  
Rutgers University  
Piscataway, New Jersey, USA  
vishal.m.patel@rutgers.edu

## Abstract

We propose a novel method for clustering a collection of data that comes from several domains. Since members of the same class might look very different across different domains and because in a clustering problem we have no side information such as labels, the main challenge in a domain adaptive clustering problem is to group the data into different clusters regardless of their domain. We approach this problem by finding mappings that can transfer the data points between the domains. We use adversarial networks to approximate these mapping functions, and form a paired representation at each domain by mapping the data onto their counter domains. Finally, we employ a multimodal subspace clustering type algorithm to cluster the paired representations with respect to their subspaces. Various experiments on datasets with domain shifts show that our method performs significantly better than many competitive domain adaptive subspace clustering methods.

## 1. Introduction

In biometrics recognition, one is often faced with a challenge of matching biometric samples that are collected under different environmental conditions. For example, in face recognition one may have to match a well-lit face image with an image that is acquired in a poor illumination condition. Another issue that what we often face in biometrics recognition is the problem of cross-sensor matching, where the test samples are verified using data enrolled with a different sensor. As new sensors are being developed for acquiring the biometric samples and existing ones are being upgraded, this becomes an important issue. Regardless of the cause of the domain shift, any distributional change (i.e. environmental, cross-sensor change, resolution

2018 IEEE 4<sup>th</sup> International Conference on Identity, Security, and Behavior Analysis (ISBA) U.S. Government work not protected by U.S. copyright.

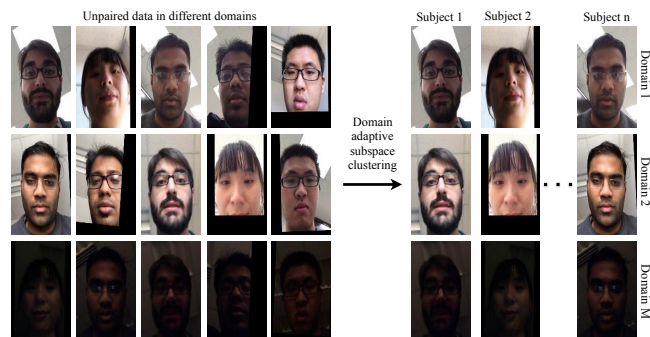


Figure 1. An overview of domain adaptive subspace clustering. Given  $N$  samples corresponding to  $n = 6$  subjects from  $M = 3$  different illumination conditions (i.e. domains), we want cluster the data into  $n$  subspaces as shown on the right side of this figure.

etc.) that occurs after learning a classifier can degrade its performance at test time. Various domain adaptation techniques have been developed in the literature to mitigate this degradation [16].

The domain adaptation problem can be defined in many different ways including semi-supervised domain adaptation [17, 23, 12, 18] and unsupervised domain adaptation [9, 3, 20, 21, 14]. In semi-supervised domain adaptation, both source and target domains are assumed to have partial labels. In contrast, in unsupervised domain adaptation, only the source domain is assumed to have partial labels and the target domain is assumed to be completely unlabeled. Another important domain adaptation problem that is often encountered in practice but is not widely studied in the literature is the problem of domain adaptive clustering, where no label information is assumed to be known [1]. Figure 1 gives an overview of the domain adaptive subspace clustering problem. Given  $N$  samples corresponding to  $n$  subjects from  $M$  different domains (i.e. different illumination conditions), we want to cluster the data into  $n$  subspaces (i.e. subjects) even though they might come from different domains. This is particularly a difficult problem because we have no side information such as labels to group the samples from the same class from different domains in a single

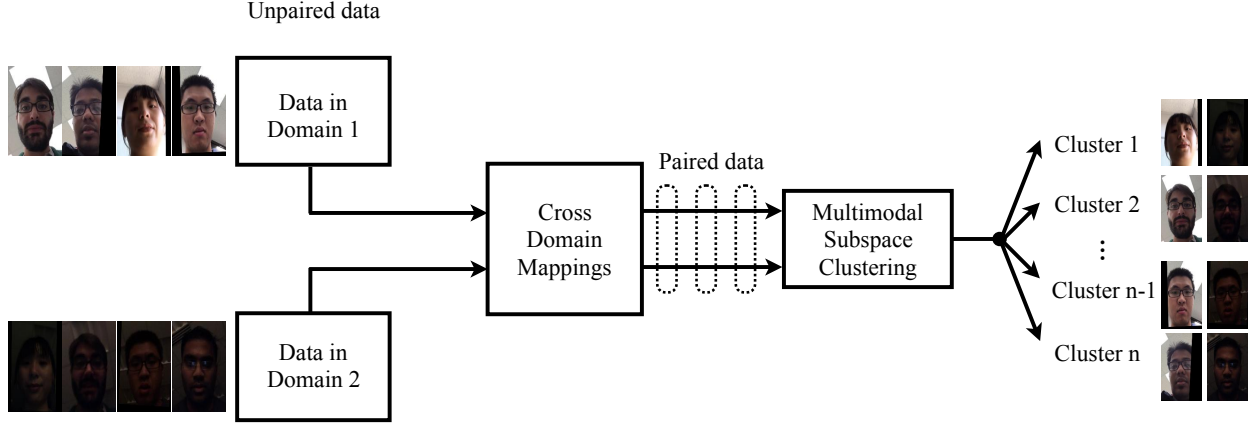


Figure 2. An overview of the proposed adversarial domain adaptive subspace clustering framework.

cluster.

In this paper, we propose a new method for domain adaptive subspace clustering in which we use adversarial networks to approximate the mapping functions that map the source data into the target domain and the target data into the source domain. Using these mapping functions, we map the available data to their counter domains and obtain a paired representation of the data corresponding to different domains. Once the paired representation of the data is obtained, we exploit their self expressiveness property and employ multimodal sparse and low-rank subspace clustering methods [2] to cluster the paired representations with respect to their subspaces. Figure 2 gives an overview of the proposed adversarial domain adaptive (ADA) subspace clustering framework.

## 2. Background and Related Work

In this section, we give a brief background on sparse and low-rank subspace clustering and related works.

### 2.1. Sparse and Low-Rank Subspace Clustering

Let  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_N] \in \mathbb{R}^{D \times N}$  be a collection of  $N$  signals  $\{\mathbf{y}_i \in \mathbb{R}^D\}_{i=1}^N$  drawn from a union of  $n$  linear subspaces  $\mathcal{S}_1 \cup \mathcal{S}_2 \cup \dots \cup \mathcal{S}_n$  of dimensions  $\{d_\ell\}_{\ell=1}^n$  in  $\mathbb{R}^D$ . Let  $\mathbf{Y}_\ell \in \mathbb{R}^{D \times N_\ell}$  be a sub-matrix of  $\mathbf{Y}$  of rank  $d_\ell$  with  $N_\ell > d_\ell$  points that lie in  $\mathcal{S}_\ell$  with  $N_1 + N_2 + \dots + N_n = N$ . Given  $\mathbf{Y}$ , the task of subspace clustering is to cluster the signals according to their subspaces.

#### 2.1.1 Sparse Subspace Clustering (SSC)

The SSC algorithm [5], which exploits the fact that noiseless data in a union of subspaces are *self-expressive*, i.e. each data point can be expressed as a *sparse* linear combination of other data points. Hence, SSC aims to find a sparse

matrix  $\mathbf{C}$  such that  $\mathbf{Y} = \mathbf{Y}\mathbf{C}$  and  $\text{diag}(\mathbf{C}) = \mathbf{0}$ , where the constraint prevents the trivial solution  $\mathbf{C} = \mathbf{I}$ . Since this problem is combinatorial and to deal with the presence of noise, SSC solves the following optimization problem instead

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 + \frac{\tau}{2} \|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2, \quad \text{s. t. } \text{diag}(\mathbf{C}) = \mathbf{0}, \quad (1)$$

where  $\|\mathbf{C}\|_1 = \sum_{i,j} |C_{i,j}|$  is the  $\ell_1$ -norm of  $\mathbf{C}$  and  $\tau > 0$  is a parameter.

#### 2.1.2 Low-Rank Representation-Based Subspace Clustering (LRR)

The LRR algorithm [13] for subspace clustering is very similar to the SSC algorithm except that a low-rank representation is found instead of a sparse representation. In particular, the following problem is solved

$$\min_{\mathbf{C}} \|\mathbf{C}\|_* + \frac{\tau}{2} \|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2, \quad (2)$$

where  $\|\mathbf{C}\|_*$  is the nuclear-norm of  $\mathbf{C}$  which is defined as the sum of its singular values. In SSC and LRR, once  $\mathbf{C}$  is found, spectral clustering methods [15] are applied on the affinity matrix  $|\mathbf{C}| + |\mathbf{C}^T|$  to obtain the segmentation of the data  $\mathbf{Y}$ .

### 2.2. Domain Adaptive Subspace Clustering

A domain adaptive subspace clustering algorithm was recently proposed in [1], in which projections are learned for different domains such that they map the data from the original domain onto a common latent subspace where the sparsity or low-rankness of data is maintained. Once the projection matrices and the sparse or low-rank coefficient matrix are found, they are used for subspace clustering. Let  $\mathbf{Y}_s$  and  $\mathbf{Y}_t$  denote the data from source and target domains,

respectively and let  $\mathbf{P}_s$  and  $\mathbf{P}_t$  be the corresponding projection matrices. Then, the following optimization problem is proposed in [1]

$$\begin{aligned} \min_{\mathbf{P}, \mathbf{C}} & \|\mathbf{C}\|_t + \frac{\tau}{2} \|\mathbf{P}\mathbf{Y} - \mathbf{P}\mathbf{Y}\mathbf{C}\|_F^2 \\ \text{s.t.} & \text{diag}(\mathbf{C}) = \mathbf{0}, \quad \mathbf{P}_s \mathbf{P}_s^T = \mathbf{P}_t \mathbf{P}_t^T = \mathbf{I}, \end{aligned} \quad (3)$$

where

$$\mathbf{P} = [\mathbf{P}_s, \mathbf{P}_t], \text{ and } \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_s & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_t \end{bmatrix}. \quad (4)$$

Note that  $t$  can be either 1 or  $*$  depending on whether SSC or LRR are used for subspace clustering. Through various face, digit and object recognition experiments, it was shown that this method can deal with the dataset shift problem in subspace clustering.

### 3. Adversarial Domain Adaptive Subspace Clustering

Due to the distributional change in the data samples, the direct comparison of data across two different domains is not reliable for the purpose of clustering. Hence, we need a way that makes the data or a representation of the data comparable with each other. Assume that we are given a collection of data that is partly distributed in the domain  $\mathcal{X}$ , and the remaining part is distributed in the domain  $\mathcal{Y}$ . If we map all the available data in the domain  $\mathcal{X}$  to  $\mathcal{Y}$ , their mapped representations are comparable to the real data in  $\mathcal{Y}$ . Furthermore, if finding such a mapping between the two domains is possible, we can also map the available data points from  $\mathcal{Y}$  to  $\mathcal{X}$ , and have more information for parsing the data. This way, for each datapoint we will have two representations - the actual domain representation and the mapped domain representation. We propose to find such mappings with the use of generative adversarial networks [8, 24]. These two representations can be paired to generate multimodal representation of the data. Finally, a multimodal subspace clustering algorithm [2] can be employed to group the multimodal (i.e. paired) data into different clusters (see Figure 2). In what follows, we present more details of finding the cross-domain mappings and the operation of multimodal subspace clustering on the mapped data.

#### 3.1. Finding Cross-Domain Mappings

Our goal is to find the mapping functions  $G : \mathcal{X} \rightarrow \mathcal{Y}$  and  $F : \mathcal{Y} \rightarrow \mathcal{X}$  that can transfer the data points between the two domains, and reveal representations of data points as if they were captured at their counter domain. Since, the exact mapping functions are not available, we propose to use trainable neural networks that can estimate these functions. To this end, we propose to use generative adversarial networks (GANs) [8] to approximate these functions and

use the corresponding generator networks to generate samples from one domain to the other.

Generative adversarial networks are unsupervised models which consist of two neural networks with distinguished adversarial tasks. The generator network receives a random variable  $z$ , drawn from a distribution  $p_z(z)$ , and learns how to generate a sample at a target data distribution  $p_{\mathcal{Y}}(y)$ . On the other hand, there is also a discriminator network that learns to catch and refuse the generated samples that are given by the generator network. The discriminator is a simple supervised classifier network that is fed by a generated sample coming from the generator or a real sample drawn from the data distribution  $p_{\mathcal{Y}}(y)$ . It is trained to recognize the generated samples. Over the time, the generator learns to generate better samples, and the discriminator learns to better recognize real samples. As a result, the generator learns to generate data points that are exactly at the data distribution so that they will not be distinguishable from the real samples [7]. Mathematically, the learning occurs through the following min-max optimization problem

$$\begin{aligned} \mathcal{L}_{GAN}(G, D) = & \mathbb{E}_{x \sim p_{\mathcal{Y}}(y)} [\log D(y)] \\ & + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))], \end{aligned} \quad (5)$$

where  $G$  is the generator network and  $D$  is the discriminator network.

It is clear that if we replace the random variable  $z$  in (5) with a sample point drawn from an actual data distribution  $p_{\mathcal{X}}(x)$ , the generator  $G$  can serve as a mapping function from the distribution  $p_{\mathcal{X}}(x)$  to the distribution  $p_{\mathcal{Y}}(y)$ . Note that this mapping is not necessarily a one to one mapping, and can map any point at the distribution  $p_{\mathcal{X}}(x)$  to any point over the distribution  $p_{\mathcal{Y}}(y)$ . Thus, methods such as pixel-to-pixel image transition [23] use supervision to transfer an image across domains to a specific area of interest at the target domain distribution. On the other hand, unpaired image transition [24] couples the generator  $G$  with another generator,  $F$ , from  $p_{\mathcal{Y}}(y)$  to  $p_{\mathcal{X}}(x)$ , and introduces additional constraints to have  $F(G(x)) \approx x$  and  $G(F(y)) \approx y$ . The additional constraints assure the existence of a cyclic path through transitions with the generators  $G$  and  $F$ . This condition pushes  $G$  and  $F$  to have an approximate one-to-one mapping property. The functions  $G$  and  $F$  can be learned by solving the following min-max optimization problem

$$\begin{aligned} \mathcal{L}_{cycGAN}(G, F, D_X, D_Y) = & \mathcal{L}_{GAN}(G, D_Y, \mathcal{X}, \mathcal{Y}) \\ & + \mathcal{L}_{GAN}(F, D_X, \mathcal{Y}, \mathcal{X}) + \lambda \mathcal{L}_{cyc}(G, F), \end{aligned} \quad (6)$$

where  $\lambda > 0$  is a regulation factor,  $\mathcal{L}_{cyc}$  enforces the cyclic path, and  $\mathcal{L}_{GAN}$  matches generated and actual distributions. These loss functions are defined as follows

$$\begin{aligned} \mathcal{L}_{cyc}(G, F) = & \mathbb{E}_{x \sim p_{\mathcal{X}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\mathcal{Y}}(y)} [\|G(F(y)) - y\|_1] \end{aligned} \quad (7)$$

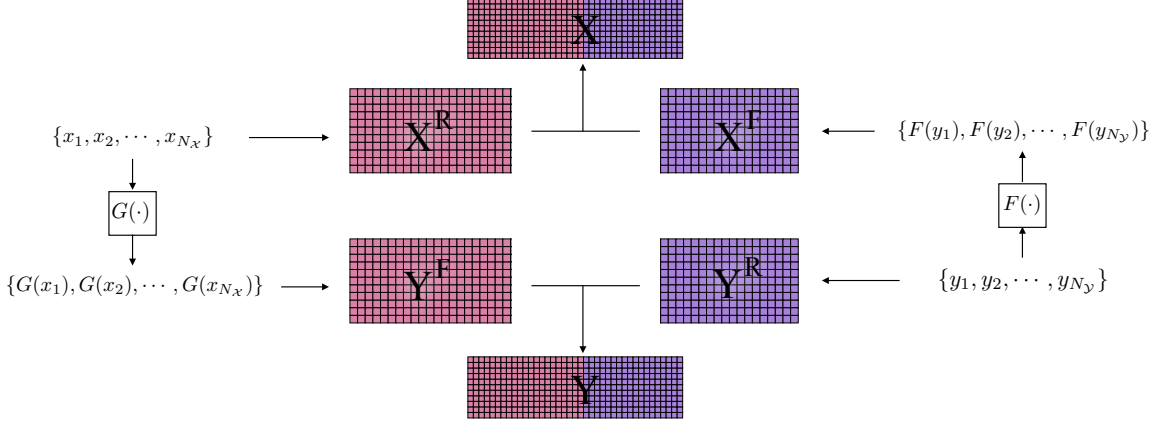


Figure 3. Building data matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , from  $\mathbf{X}^R$ ,  $\mathbf{X}^F$ ,  $\mathbf{Y}^R$  and  $\mathbf{Y}^F$ .

and

$$\mathcal{L}_{GAN}(G, D_Y, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{y \sim p_{\mathcal{Y}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\mathcal{X}}(x)}[\log(1 - D_Y(G(x)))] \quad (8)$$

Domain adaptive subspace clustering problem is a complete unsupervised learning problem. In other words, we do not have any class label information in any of the domains. So we cannot manipulate the supervised domain transfer methods for this problem. In contrast, optimization of (6) does not need class label information for training the generators. Thus, with the available information (i.e. the domain knowledge of the data points), we can train generator networks with the adversarial setup of (6), and use them as approximations for mapping functions that transfer the data points between the domains.

### 3.2. Multimodal Subspace Clustering

Once the mapping functions  $G : \mathcal{X} \rightarrow \mathcal{Y}$  and  $F : \mathcal{Y} \rightarrow \mathcal{X}$  are found, we use them to map all the data points to their counter domains. Assume that we are given  $N_{\mathcal{X}}$  data points in the domain  $\mathcal{X}$ , and  $N_{\mathcal{Y}}$  data points in the domain  $\mathcal{Y}$ . Let  $\mathbf{x}_i^R \in \mathcal{X}$ , and  $\mathbf{y}_j^R \in \mathcal{Y}$  be the  $i$ -th and  $j$ -th real sample points at the domains  $\mathcal{X}$  and  $\mathcal{Y}$ , respectively. Furthermore, assume  $\mathbf{y}_i^F \in \mathcal{Y}$  and  $\mathbf{x}_j^F \in \mathcal{X}$  are the generated ‘fake’ samples that are obtained using the learned mapping functions. In another words, we have

$$\mathbf{y}_i^F = G(\mathbf{x}_i^R), \text{ and } \mathbf{x}_j^F = F(\mathbf{y}_j^R).$$

We build the modified data matrices as follows. Fix the permutation of data points at the both domains to an arbitrary order, and concatenate all the  $\mathbf{x}_i^R$ s to construct the matrix  $\mathbf{X}^R$ . This matrix is associated with the real data points in the domain  $\mathcal{X}$ . Do the same with  $\mathbf{x}_j^F$ s and construct the matrix  $\mathbf{X}^F$ . This matrix is associated with the mapped data points from the domain  $\mathcal{Y}$  to  $\mathcal{X}$  (fake samples). Similarly,

build the matrices  $\mathbf{Y}^R$  and  $\mathbf{Y}^F$  while the arrangement of the data is kept in the same order. These two matrices represent the real and fake samples in the domain  $\mathcal{Y}$ . Finally, we merge  $\mathbf{X}^R$  and  $\mathbf{X}^F$  to create the matrix  $\mathbf{X}$ , and similarly merge  $\mathbf{Y}^R$  and  $\mathbf{Y}^F$  to create  $\mathbf{Y}$ . Figure 3 depicts the construction of the data matrices.

Note that  $\mathbf{X}$  and  $\mathbf{Y}$  both have  $N_{\mathcal{X}} + N_{\mathcal{Y}}$  columns containing two different representations for all the available data. In each representation (i.e. modality), the data points are in the same domain which makes the comparison of these points reliable for the task of clustering. In addition, since we have two representations for each datapoint, we can use this information to better cluster them. This can be done by exploiting a multimodal sparse and/or low-rank representation-based subspace clustering method [2] to segment the paired data in accordance with their subspaces.

In multimodal subspace clustering, the self expressiveness property of each sample is exploited in its respective modality with a common representation across the modalities [2]. In Multimodal Sparse Subspace Clustering (MSSC), the common representation is enforced to be sparse, while in Multimodal Low-Rank Representation-based subspace clustering (MLRR), the common representation is enforced to have a low-rank structure. The objective function corresponding to these methods is as follows

$$\min_{\mathbf{C}} \|\mathbf{X} - \mathbf{X}\mathbf{C}\|_F^2 + \|\mathbf{Y} - \mathbf{Y}\mathbf{C}\|_F^2 + \tau \|\mathbf{C}\|_p, \quad (9)$$

where  $\mathbf{C}$  is the common sparse or low-rank representation coefficient for the paired data,  $\tau > 0$  is a parameter and  $\|\cdot\|_p$  is  $\|\cdot\|_1$  in MSSC and  $\|\cdot\|_*$  in MLRR. These problems can be efficiently solved using the ADMM method [2]. Once  $\mathbf{C}$  is found, spectral clustering methods [15] are applied on the affinity matrix  $|\mathbf{C}| + |\mathbf{C}|^T$  to obtain the segmentation of the data [5, 13]. The proposed adversarial domain adaptive subspace clustering method is summarized in Algorithm 1.

---

**Algorithm 1** Adversarial domain adaptive subspace clustering algorithm

---

- 1: **procedure** ADA( $\{x_i\}_{i=1}^{N_x} \in \mathcal{X}, \{y_i\}_{i=1}^{N_y} \in \mathcal{Y}$ ).
  - 2: Find cross domain mappings  $G$  and  $F$  by optimizing the min-max problem (6).
  - 3:  $\{G(x_i)\}_{i=1}^{N_x} \in \mathcal{Y} \leftarrow$  Use  $G$  to map  $\{x_i\}_{i=1}^{N_x}$  into  $\mathcal{Y}$ .
  - 4:  $\{F(y_i)\}_{i=1}^{N_y} \in \mathcal{X} \leftarrow$  Use  $F$  to map  $\{y_i\}_{i=1}^{N_y}$  into  $\mathcal{X}$ .
  - 5:  $\mathbf{X}^R, \mathbf{Y}^R \leftarrow$  Store real data  $\{x_i\}_{i=1}^{N_x}, \{y_i\}_{i=1}^{N_y}$  in matrix forms.
  - 6:  $\mathbf{Y}^F, \mathbf{X}^F \leftarrow$  Store fake data  $\{G(x_i)\}_{i=1}^{N_x}, \{F(y_i)\}_{i=1}^{N_y}$  in matrix forms.
  - 7:  $\mathbf{X} = [\mathbf{X}^R, \mathbf{X}^F]$ .
  - 8:  $\mathbf{Y} = [\mathbf{Y}^F, \mathbf{Y}^R]$ .
  - 9: Find  $\mathbf{C}$  by solving the optimization problem (9).
  - 10: Normalize the columns of  $\mathbf{C}$  as  $\mathbf{c}_i \leftarrow \frac{\mathbf{c}_i}{\|\mathbf{c}_i\|_\infty}$ .
  - 11: Form a similarity graph with  $N_x + N_y$  nodes and set the weights on the edges between the nodes by  $\mathbf{W} = |\mathbf{C}| + |\mathbf{C}|^T$ .
  - 12: Apply spectral clustering to the similarity graph.
  - 13: **end procedure**
  - 14: **Output:** Segmented data.
- 

## 4. Experimental Results

We evaluate the applicability of the proposed domain adaptive subspace clustering method on two datasets: UMDAA-01 active authentication dataset [6] and ARL Polarimetric face dataset [10]. We compare the performance of our method with that of the recently published Domain Adaptive Sparse and Low rank Subspace Clustering methods (DASSC/DALRR) [1]. In addition, similar to [1], we use some of the recent domain adaptation methods to extract the domain adaptive features from each domain, and then feed them to the traditional SSC or LRR algorithms for clustering data. The compared domain adaptive methods include frustratingly easy domain adaptation (ED) method [4], Correlation Alignment (CORAL) [19], and a Grassmann manifold (GM) based method [9]. Based on the subspace clustering algorithm that the domain adaptive features are fed to, we call these methods as ED-SSC/ED-LRR, CO-SSC/CO-LRR and GM-SSC/GM-LRR. We also call our method ADA-SSC when we feed the data points and their mapped pairs to MSSC algorithm. Similarly, when we feed them to the MLRR algorithm, we call it ADA-LRR.

The performances of the traditional SSC and LRR algorithms are also reported as a baseline for comparison. In all the experiments, we repeat the process of subspace clustering 10 times and report the 10-fold average performance of the methods. To reduce the computations, we use PCA to reduce the dimension of data to 500 before feeding them to the subspace clustering methods. The regulation parameter  $\tau$  in equation (9) of our method is selected by 5-fold cross

validation.

We use the same network architectures as suggested in [24] for the generator and discriminator networks of the experiments with our method. The generator networks have nine blocks with two 2-stride convolutions along with several residual blocks, and two fractionally-strided convolution layers with stride of  $\frac{1}{2}$ . The discriminators are 70 patchGANs [11]. Parameter selection and experimental settings for the other methods are optimized according to their original papers. Subspace clustering error, defined as

$$\text{Clustering error} = \frac{\#\text{misclassified points}}{\text{total \# points}} \times 100.$$

is used to measure the performance of different algorithms in our experiments.

### 4.1. UMDAA-01 Dataset

In the first set of experiments, we use the facial images from the UMDAA-01 dataset [6] for evaluating our method on clustering data that are captured in different sessions. This dataset contains facial images of 50 users over 3 sessions corresponding to different illumination conditions. In each session more than 750 images are taken from each face. We randomly select 50 images of the first 10 subjects in each session and use them in our experiments. Different sessions in this experiments are considered as different domains. Figure 4 shows some sample images from this dataset. As various ambient conditions make this dataset very challenging, we do preprocessing on the data to normalize the illumination conditions before feeding them to different methods. For this purpose, we use the normalization method introduced in [22].

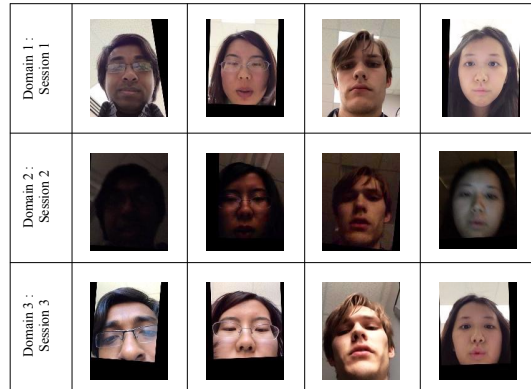


Figure 4. Sample images from the UMDAA-01 dataset [6]. Variations among different sessions are clear.

Table 1 compares the performance of our method with that of different subspace clustering methods on this dataset. As it can be seen from this table, while in comparison to the baseline methods SSC and LRR, other domain adaptive methods can bring some improvements to the performance

of the task, our methods improve the performance by a large margin. This is because our methods benefit simultaneously from two advantages. First that our adversarial networks generate paired samples for each datapoint at its counter domain where its distance from the other data points are reliable for clustering. Second, and more importantly, as we use a multimodal subspace clustering algorithm where the generated data are bundled with the real data in the original domain for clustering. This helps keeping them mutual clusters, and prevents over-segmentation. We discuss this effect in section 4.3 with more details.

Method	{1}, {2}	{2}, {3}	{1}, {3}	Avg. $\pm$ std.
SSC	62.12	65.61	63.44	63.72 $\pm$ 1.76
ED-SSC	63.53	67.33	65.32	65.39 $\pm$ 1.90
CO-SSC	61.10	62.67	61.54	61.77 $\pm$ 0.81
DA-SSC	59.43	60.21	61.08	60.24 $\pm$ 0.83
GM-SSC	60.63	61.33	59.65	60.54 $\pm$ 0.84
ADA-SSC	<b>51.32</b>	<b>52.64</b>	<b>49.76</b>	<b>51.24 <math>\pm</math> 1.44</b>
LRR	67.44	65.85	66.12	66.47 $\pm$ 0.85
ED-LRR	68.21	66.37	65.43	66.67 $\pm$ 1.41
CO-LRR	65.34	64.32	62.11	63.92 $\pm$ 1.65
DA-LRR	66.12	64.11	62.63	64.29 $\pm$ 1.75
GM-LRR	65.84	65.73	63.22	64.93 $\pm$ 1.48
ADA-LRR	<b>60.28</b>	<b>58.63</b>	<b>59.66</b>	<b>59.52 <math>\pm</math> 0.83</b>

Table 1. Subspace clustering performance of different methods on the UMDAA-01 dataset. The top performing method in each experiment is shown in boldface. Note that {1}, {2} and {3} correspond to first, second and third sessions in the dataset, respectively. Last column shows the overall average performance of the methods.

## 4.2. ARL Polarimetric Face Dataset

In this section, we use three modalities from the ARL Polarimetric face database [10] to evaluate our method on clustering data that are captured in different modalities. The ARL database [10] that is recently collected for cross-spectrum face recognition research, contains sample facial images of 60 subjects in visible and four different polarimetric thermal domains. Each person has several images per each modality that are collected from different distances. Sample images from this dataset are shown in Figure 5.

Table 2 presents the average performance of various subspace clustering methods on different combinations of the modalities in the ARL dataset. This table clearly shows that our method can perform well across various domains even if the domains are from different modalities. As the table suggests, in the experiments that the visible data is one of the domains, we generally see better performances for all the methods. This happens because the visible modality has more information to be used in adaptation compared to the other spectrums. Our methods, on average, outperform the



Figure 5. Sample images from three domains of the ARL Polarimetric face dataset [10].

other methods in all the experiments, including the experiments between polarimetric thermal modalities.

Method	{V}, {DP}	{S0}, {DP}	{V}, {S0}	Avg. $\pm$ std.
SSC	58.61	62.34	59.18	60.04 $\pm$ 2.01
ED-SSC	60.27	65.26	60.83	62.12 $\pm$ 2.73
CO-SSC	57.70	61.42	61.66	60.26 $\pm$ 2.22
DA-SSC	56.43	60.68	59.84	58.98 $\pm$ 2.25
GM-SSC	55.36	60.73	60.37	58.82 $\pm$ 3.00
ADA-SSC	<b>45.83</b>	<b>49.31</b>	<b>43.76</b>	<b>46.30 <math>\pm</math> 2.80</b>
LRR	59.52	58.43	58.35	58.77 $\pm$ 0.65
ED-LRR	62.06	59.88	53.33	58.42 $\pm$ 4.54
CO-LRR	63.91	60.66	56.11	60.23 $\pm$ 3.92
DA-LRR	59.45	57.21	55.42	57.36 $\pm$ 2.02
GM-LRR	57.65	57.27	57.32	57.41 $\pm$ 0.21
ADA-LRR	<b>51.11</b>	<b>53.21</b>	<b>52.14</b>	<b>52.15 <math>\pm</math> 1.05</b>

Table 2. Subspace clustering performance of different methods on the ARL dataset. The top performing method in each experiment is shown in boldface. Note that {V}, {S0}, and {S1} correspond to Visible, S0 and S1 modalities in the ARL database, respectively. Last column shows the overall average performance of the methods.

## 4.3. Effect of Bundling Representations

In this experiment, we evaluate the effect of pairing the mapped samples with their original data points on subspace clustering. Since finding the true mapping functions between the domains without supervision is a very difficult problem [24], sometimes the fake samples we generate with our method are not mapped onto the exact desired point in the target domain. That is, even though the generated sample is in the desired domain, it is not exactly on the desired point. For example, when we do the experiments between DP and visible domains using the ARL dataset, the mapped datapoint from DP to visible domain might not look like the true appearance of the person, even though it is in visible domain. Figure 6 shows some extreme cases corresponding to this issue. However, as in our method, these samples are attached to their true samples in the original domain, if

the true samples are fairly having good representations to be clustered with their class members within their original domain, they can compensate the bad representation of their mapping, and drag them to the true cluster in their domain.



Figure 6. Failure cases of generated facial images in visible domain by feeding the trained adversarial network with samples from the DP polarimetric thermal domain in the ARL dataset [10].

To evaluate this, we repeat the previous experiment on the visible and DP polarimetric data with same trained generator networks to generate counterpart samples in the other domain, but instead of using a multimodal subspace clustering at the end, we cluster the real and generated data in each domain separately. In other words, we use the traditional SSC and LRR algorithms on all the fake and real data per domain. Here, ADA-SSC- $\{a\}$  means that we only feed the data points in the domain  $\{a\}$  to the subspace clustering methods. Table 3 shows the effect of bundling the generated data with their original real data. For comparison, results corresponding to the ADA-SSC and ADA-LRR methods are also copied from table (2). As the table suggests, separate subspace clusterings on the single domains are also having less clustering error compared to the many of the subspace clustering methods reported in the second column of Table 2. However, when we use multimodal subspace clustering, we achieve much better performance (near 10% in the case of using MSSC). Figure 7 shows some of the generated samples and their original data points corresponding to this experiment which explains how bundling the imperfect generated samples in the counter domain with their original data points can help to reduce the clustering error.

## 5. Conclusion

We presented a new method called adversarial domain adaptive subspace clustering for clustering a collection of data that is distributed in multiple domains. We used the adversarial networks to learn the mapping functions between

Method	ADA-SSC- $\{V\}$	ADA-SSC- $\{DP\}$	ADA-SSC
Clustering error	58.04	55.69	<b>45.83</b>
Method	ADA-LRR- $\{V\}$	ADA-LRR- $\{DP\}$	ADA-LRR
Clustering error	57.71	55.46	<b>51.11</b>

Table 3. Effect of bundling data points across domains in domain adaptive subspace clustering.

the domains. We then demonstrated how to use these mapping functions to generate two representations per datapoint and use them in a multimodal subspace clustering setting to cluster the data. Extensive experiments on two face datasets with domain shifts showed that the proposed method can perform better than many state-of-the-art domain adaptive subspace clustering methods.

## Acknowledgment

This work was supported by the NSF grant 1618677.

## References

- [1] M. Abavisani and V. M. Patel. Domain adaptive subspace clustering. In *British Machine Vision Conference*, 2017.
- [2] M. Abavisani and V. M. Patel. Multimodal sparse and low-rank subspace clustering. *Information Fusion*, 39:168 – 177, 2018.
- [3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *arXiv preprint arXiv:1612.05424*, 2016.
- [4] H. Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009.
- [5] E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(11):2765–2781, 2013.
- [6] M. E. Fathy, V. M. Patel, and R. Chellappa. Face-based active authentication on mobile devices. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1687–1691, April 2015.
- [7] I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [9] R. Gopalan, R. Li, and R. Chellappa. Unsupervised adaptation across domain shifts by generating intermediate data representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 36(11):2288–2302, 2014.
- [10] S. Hu, N. J. Short, B. S. Riggan, C. Gordon, K. P. Gurton, M. Thielke, P. Gurram, and A. L. Chan. A polarimetric thermal database for face recognition research. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 119–126, 2016.

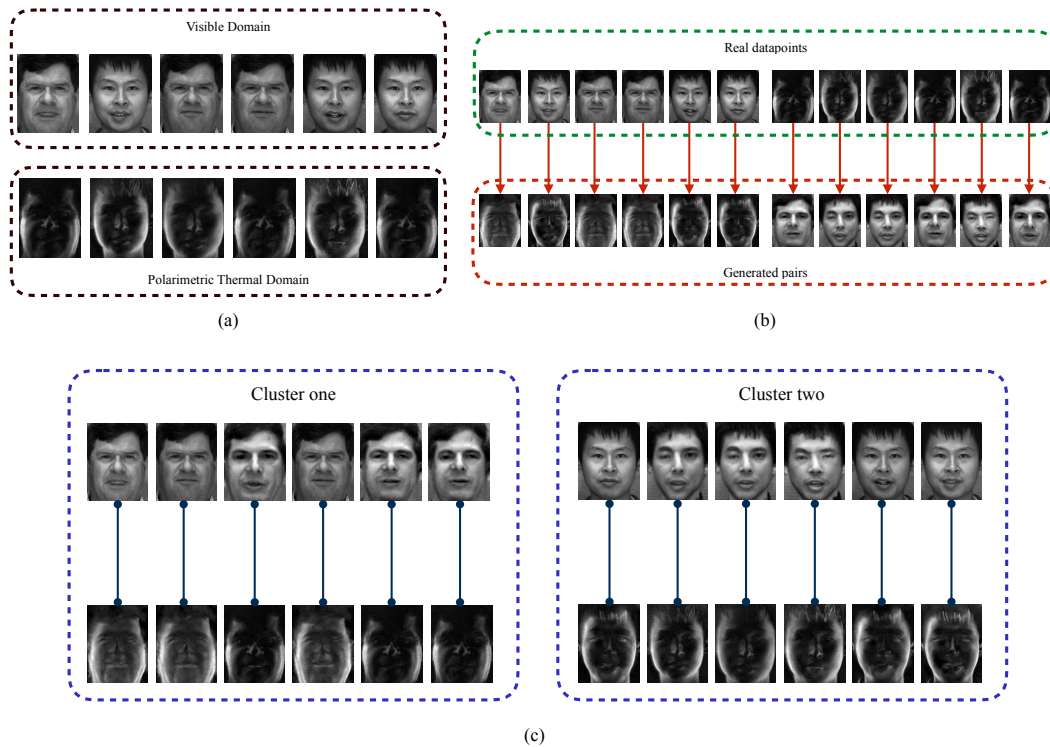


Figure 7. Samples corresponding to the adversarial domain adaptive clustering experiment for the visible and DP domains in the ARL database [10]. a) Available data points in the two domains. The only known information about the data points is their domains. b) Original data points and their generated representation at the counter domain. The generated representations are paired with their original data points to be fed into multimodal subspace clustering algorithm. c) Clustered pairs of data points.

- [11] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004*, 2016.
- [12] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang. Robust visual domain adaptation with low-rank reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2168–2175. IEEE, 2012.
- [13] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 35(1):171–184, 2013.
- [14] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.
- [15] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Neural Information Processing Systems (NIPS)*, volume 2, pages 849–856, 2002.
- [16] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, May 2015.
- [17] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226. Springer, 2010.
- [18] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 361–368, 2013.
- [19] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2058–2065, 2016.
- [20] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.
- [21] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. *arXiv preprint arXiv:1702.05464*, 2017.
- [22] H. Wang, S. Z. Li, and Y. Wang. Face recognition under varying lighting conditions using self quotient image. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 819–824, 2004.
- [23] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. In *European Conference on Computer Vision*, pages 517–532. Springer, 2016.
- [24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, 2017.