Sub-GAN: An Unsupervised Generative Model via Subspaces

Jie Liang[1], Jufeng Yang[1*], Hsin-Ying Lee[2], Kai Wang[1], Ming-Hsuan Yang[2,3]

[1]Nankai University [2]University of California, Merced [3]Google Cloud

Abstract. The recent years have witnessed significant growth in constructing robust generative models to capture informative distributions of natural data. However, it is difficult to fully exploit the distribution of complex data, like images and videos, due to the high dimensionality of ambient space. Sequentially, how to effectively guide the training of generative models is a crucial issue. In this paper, we present a subspace-based generative adversarial network (Sub-GAN) which simultaneously disentangles multiple latent subspaces and generates diverse samples correspondingly. Since the high-dimensional natural data usually lies on a union of low-dimensional subspaces which contain semantically extensive structure, Sub-GAN incorporates a novel clusterer that can interact with the generator and discriminator via subspace information. Unlike the traditional generative models, the proposed Sub-GAN can control the diversity of the generated samples via the multiplicity of the learned subspaces. Moreover, the Sub-GAN follows an unsupervised fashion to explore not only the visual classes but the latent continuous attributes. We demonstrate that our model can discover meaningful visual attributes which is hard to be annotated via strong supervision, *e.g.*, the writing style of digits, thus avoid the mode collapse problem. Extensive experimental results show the competitive performance of the proposed method for both generating diverse images with satisfied quality and discovering discriminative latent subspaces.

1 Introduction

Significant progress has been made in deep generative modeling, of which the ability to synthesize data requires a deep understanding of the data structure. Recently, generative adversarial network (GAN) [1] has emerged as a promising framework for generating complex data distribution in a data-driven manner. GAN is composed of a generator and a discriminator, where the generator maps samples from an arbitrary latent distribution to ambient data space and the adversarial discriminator attempts to distinguish between real and generated samples. Both modules are optimized via adversarial training.

While GAN has shown promising results in simulating complex data distribution like images and videos, the realistic distribution is not fully exploited.

*Corresponding Author

The complexity of real data makes it difficult for generative models to learn useful and detailed attributes without any guidance. Sequentially, the conditional GAN [2] proposes to provide direct clustering guidance in a supervised manner where the labels of data are given. However, the requirements of annotations constrain the generative models to limited applications with strong prior of the distinctive classes, *e.g.*, the 10 digits in the MNIST dataset. Furthermore, there are far more intrinsic patterns which are hard to be labeled, such as the various styles of the hand-written digits. Full exploitations on these latent structures can obviously alleviate the mode collapse problem in the generation process.

Research has shown that high-dimensional data can always be modeled as a union of low-dimensional subspaces [3]. Numerous subspace clustering methods have been developed to explore the high-dimensional data distribution [3,4]. The disentangling of the underlying low-dimensional subspaces serves as a guidance on approximate the data distribution and can facilitate the generation on complex data space.

In this work, we propose a joint framework, *i.e.*, subspace-based generative adversarial network (Sub-GAN), to simultaneously discover intrinsic subspaces in an unsupervised manner and generate realistic samples from each of them. Sub-GAN consists of three modules, a clusterer, a generator and a discriminator. The clusterer aims to discover distinctive subspaces of high-dimensional data in an unsupervised fashion. It is updated on each epoch based on the feedback from the discriminator. The generator produces samples conditioned on a one-hot vector indicating the belonged cluster and a base vector of subspace derived from the clusterer. The discriminator not only needs to distinguish between real and fake samples, but also requires to classify them to belonged subspaces. It also provides distinctive representations of data samples for updating the clusterer. We conduct extensive experiments to validate the effectiveness of the proposed framework. Specifically, based on both visualized and quantitative results, we demonstrate that the generated samples are not only visually appealing but diverse with multiple latent attributes. We also show that our model achieves favorable performance on image clustering tasks.

Our contributions are of two folds. First, we present a joint unsupervised framework to simultaneously learn the subspaces of the ambient space and generating instances accordingly, where both tasks are mutually optimized. Second, we address the mode collapse problem by specifying the number of distinct subspaces, from which we generate meaningful and diverse images with informative visual attributes. Extensive experiments demonstrate the effectiveness of the proposed Sub-GAN model.

2 Related Work

Deep Generative Models Deep generative frameworks have recently drawn significant attention due to the ability of modeling large-scale unlabeled data [5–10]. The generative models can be applied to various low-level vision problems, *e.g.*, image super-resolution [11,12] and semantic segmentation [13,14].

Generative models aim to fit the space of real data samples, *e.g.*, a set of natural images [15–17]. To capture the real distribution, most generative models optimize an aggregated probabilistic problem conditioned on latent noises over multiple variables. They assume that all data samples are drawn from a single low-dimensional latent space. Early studies focused on learning embedded representations in an unsupervised manner, *e.g.*, the restricted Boltzmann machines (RBM [18, 19]) and the stacked auto-encoders (AE [20]). For instance, Hinton *et al.* [21] propose to efficiently train the deep belief nets (DBN) by using the contrastive divergence algorithm. Both DBN and AE learn a low-dimensional representation for each data sample on a single latent space, followed by generating new instances via a decoding network [22]. However, these methods suffer from the difficulty to disentangle an intractable probabilistic optimization problem while maximizing the training data likelihood, especially for the data of high-dimensionality [23]. More recently, Goodfellow *et al.* [1] propose GAN as an alternative adversarial strategy for training the generative models. The minimax game between the generator and the discriminator induces a data-driven approximation process from a low-dimensional latent distribution, *e.g.*, standard Gaussian, to a high-dimensional real distribution. During training, the adversarial module is used to optimize a loss function and sidesteps the requirement to explicitly calculate or approximate the complicated ambient space. Nevertheless, due to the high-dimensional and contradictory nature of the two counterparts, traditional GANs suffer from the mode collapse problem as well as the instable training [24, 25], which are crucial for further improvement.

Built upon these generative models, various conditional image generation methods (*e.g.*, CGAN [2]) are proposed to generate a specific deterministic output from a given conditioning latent vector, which somehow controls the diversity of the generation. In particular, the latent variable is designed to encode the object class by concatenating the ground-truth labels so that the generator can produce samples from specific visual category [26, 27]. The CGAN has the advantage of providing better representations for multi-modal data generation, but such inference process relies on the extensively annotated training data, some of which is hard to explicitly labeled, *e.g.*, the writing styles of the digits [28]. Recently, InfoGAN [29] optimizes the mutual information of latent codes, which is constructed by a mixture of Gaussian instead of uniform noise. However, it lacks explicit categorical assignments as well as distinctive embedding vectors of samples. In this paper, we present a joint model to simultaneously learn the informative latent category of the real samples and conduct the generation from each subspace, the inference and generation process are totally unsupervised and mutually optimized.

Subspace Learning Modeling high-dimensional data has been one of the most critical issues in computer vision [30]. As the high-dimensional data is usually distributed in a union of low-dimensional subspaces [4, 31], numerous deep subspace clustering methods [32–35] have been developed in the literature.

The goal of subspace learning methods is to find a given number of disentangled low-dimensional subspaces [34]. Traditional algorithms focus on calculating

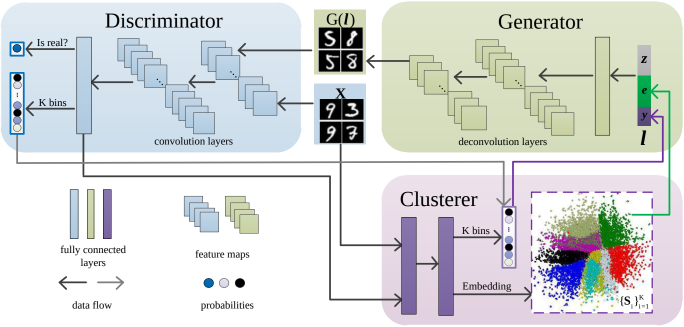


Fig. 1. Main steps of the proposed Sub-GAN method. The three boxes represent the clusterer C (*purple*), generator G (*green*) and discriminator D (*blue*), respectively. We design C to disentangle K subspaces $\{\boldsymbol{S}_i\}_{i=1}^K$ for the given dataset $\boldsymbol{X}$ (when initializing) or the deep features derived from D (during training). For G, the input $\boldsymbol{l}$ consists of three components, *i.e.*, $\boldsymbol{e}$ and $\boldsymbol{y}$ derived from C, and the noise vector $\boldsymbol{z} \in \mathcal{N}(0,1)$. The D can not only discriminate the real or fake images by outputting a binary prediction, but also calculate probabilities of each subspace to refine the C. The K bins from both C and D are unified for comprehensive prediction.

the similarity/dissimilarity relationship among instances [36], followed by constructing graphs and conducting spectral clustering [37]. Recently, researchers propose to extract more distinctive representations of each sample via a deep embedding network [38]. Xie *et al.* [39] propose the deep embedding clustering (DEC) algorithm for learning a non-linear mapping from data space to a latent feature space with a denoising stacked autoencoder (DAE), followed by refining the clustering assignments. The DEC framework first pre-trains the DAE and then fine-tunes it stacked by iteratively optimizing a clustering objective function based on the Kullback-Leibler (KL) divergence with a self-training target distribution. However, it requires the layer-wise pre-training and a non-joint embedding and clustering [34]. In this paper, we propose a joint model for training all modules simultaneously with an adversarial strategy, which is demonstrated to be effective to extract distinctive subspaces.

3 Subspace-based GAN

Given a set of unlabeled high-dimensional data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^N$, the goal of generative models is to approximate the real distribution $p_{\boldsymbol{x}}(\boldsymbol{x})$ via a mapping $G(\cdot)$ from a low-dimensional latent variable $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$, *i.e.*, $\boldsymbol{x} = G(\boldsymbol{z})$. However, directly modeling the raw space may suffer from the problem of mode collapse, *i.e.*, the generated samples are of similar pattern which caters the objective function [40]. It also leads to an instable training [41].

The data samples $\boldsymbol{X}$ could be drawn from multiple subspaces $\{\boldsymbol{S}_i\}_{i=1}^K$, which depict informative attributes and are easier to be approximated than the high-dimensional ambient space. Hence, we present a joint unsupervised framework

Algorithm 1 : Training of Sub-GAN

```
Input: $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^{d_x}$, $K$, $N_i$, $N_b$.
Calculate the correlation matrix $\boldsymbol{C}$ by solving the self-representation problem (1);
Compute the Laplacian matrix $\boldsymbol{M}$ by (2);
Disentangle the subspaces by calculating $\{\boldsymbol{e}_i\}_{i=1}^{K}$ using (3);
Calculate an initialized cluster assignment $\hat{\boldsymbol{y}}_{ini}$ using $K$-means;
while $I < N_i$ do
    Calculate comprehensive latent codes $\boldsymbol{L} = \{\boldsymbol{l}_i\}_{i=1}^{K}$ for each subspace using (8);
    Update the generator $G(\boldsymbol{L})$ by optimizing (6);
    Update the discriminator $D$ by optimizing (9);
    Calculate new distinctive representations for each sample $\boldsymbol{x}$;
    if $I\%(\frac{N}{N_b}) = 0$ then
        Calculate and update $\boldsymbol{C}$ and $\boldsymbol{e}$;
        Update $C$ according to (12);
    end if
end while
Output: Cluster assignment $\hat{\boldsymbol{y}}$; Generator $G$.
```

termed as Sub-GAN, to seek auxiliary distributions which effectively cover multiple modes of the multi-modal data $\boldsymbol{X}$. In the rest of this section, we first describe the deep clustering module C, which disentangles the $\{\boldsymbol{S}_i\}_{i=1}^{K}$ of the ambient space $\boldsymbol{X}$. Afterwards, we explain the formulation of deep generative modules, including a generator G and discriminator D which alternate between updating model parameters for both generation and clustering. Fig. 1 shows the pipeline of Sub-GAN and Algorithm 1 illustrates the training process, where N_i and N_b denote the number of iterations and batch size, respectively.

3.1 Clusterer for Subspace Disentangling

We consider the task of clustering a set of N samples $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^{N} \in \mathbb{R}^{d_x}$ into K clusters $\{\boldsymbol{S}_i\}_{i=1}^{K}$, where d_x denotes the dimension of $\boldsymbol{X}$ and K is fixed based on the diversity and intrinsic structure of the $\boldsymbol{X}$. Note we allow the user control on K to generate either diverse or compact samples. To satisfy the requirement on subspace disentangling, we design a clusterer C which is jointly learned in the adversarial framework. We first initialize the soft assignments $\hat{\boldsymbol{P}}$ via subspace clustering [30]. Then, we minimize the KL divergence between predicted assignment $\boldsymbol{P}$ and an auxiliary target distribution $\boldsymbol{T}$. During training of Sub-GAN, we iteratively map the raw data samples into a distinctive embedding space $\boldsymbol{U} \in \mathbb{R}^{d_u}$ where we have $d_u \ll d_x$. Meanwhile, the adversarial process can provide gradients for refining the soft assignment.

For initializing the assignment with raw data samples, we follow a two-step subspace clustering approach. Specifically, we disentangle the multiple affine subspaces $\{\boldsymbol{S}_i\}_{i=1}^{K}$ using self-representation and graph clustering techniques. We first tackle the following ℓ_1-norm optimization problem [4] to calculate the self-representation of the data samples:

$$\min_{\boldsymbol{C}} \|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}\|_2^2 + \lambda \|\boldsymbol{C}\|_1, \quad s.t.\ \mathrm{diag}(\boldsymbol{C}) = \boldsymbol{0}, \tag{1}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm and the constraint $\mathrm{diag}(\boldsymbol{C}) = \boldsymbol{0}$ eliminates the trivial solution of representing sample $\boldsymbol{x}$ as a linear combination of itself. Here, $\boldsymbol{C}$ denotes the coefficient matrix where each entry c_{ij} reflects the similarity between samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Afterwards, the $\boldsymbol{C}$ is used to define a directed graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ where the each vertex in $\boldsymbol{V}$ represents a data sample and the edge $(v_i, v_j) \in \boldsymbol{E}$ is weighted by c_{ij}. We construct a balanced graph $\hat{\boldsymbol{G}}$ with the adjacency matrix $\boldsymbol{W}$ where we have $\boldsymbol{W} = |\boldsymbol{C}| + |\boldsymbol{C}^\top|$. Then, the Laplacian matrix $\boldsymbol{M}$ of the graph is calculated by

$$\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{W}, \tag{2}$$

where $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ is computed as $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. Given the Laplacian matrix $\boldsymbol{M}$, we calculate the first K eigenvectors as

$$[\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_K] = \mathrm{eig}(\boldsymbol{M}), \tag{3}$$

where $\mathrm{eig}(\cdot)$ is the decomposition function to extract the eigenvectors of a matrix. Note the multiplicity of the zero eigenvalues of $\boldsymbol{M}$ reflects the number of connected components in $\boldsymbol{G}$ [4], thus the eigenvectors are distinguishable in the latent spectral space. We finally use the K-means to calculate the initialized clustering assignment $\hat{\boldsymbol{P}}$ [42].

Given $\hat{\boldsymbol{P}}$, we refine the model predictions iteratively in the following training process. In each iteration I, we feed the G and D with both data sample $\boldsymbol{x}_i$ and the current predicted subspace assignment $\hat{p}_i^I$. On top of the deep network of the discriminator which is illustrated in Section 3.3, we generate the deep embedding feature $\boldsymbol{f}_i^I$ of each sample $\boldsymbol{x}_i$ which is distinguishable on discriminating the K subspaces. Given $\boldsymbol{f}$, we calculate the soft assignment $\boldsymbol{P}_b$ for each local training set $\boldsymbol{X}_b$, where $\boldsymbol{X}_b$ is composed of images in each batch. We then define a clustering objective function $\mathcal{L}_C$ for each iteration I by using the Kullback-Leibler (KL) divergence to minimize the distance between the prediction $\boldsymbol{P}^I$ and a target variable $\boldsymbol{Q}^I$:

$$\mathcal{L}_C^I = \mathrm{KL}(\boldsymbol{Q}^I || \boldsymbol{P}^I) = \frac{1}{N_b} \sum_{i=1}^{N} \sum_{k=1}^{K} q_{ik} \log \frac{q_{ik}}{p_{ik}}, \tag{4}$$

where N_b denotes the batch size, N is the number of training samples and K is the number of subspaces. Here, we induce a sparse prediction matrix $\boldsymbol{P}$ where each $\boldsymbol{p}_i$ is a one-hot vector, *i.e.*, $p_{ik} = 1$ for $\boldsymbol{x}_i \in \boldsymbol{S}_k$ and $\{p_{ij}\}_{j \neq k} = 0$. In the clusterer C, we update the target distribution $\boldsymbol{Q}$ by normalizing based on the frequency for each cluster:

$$q_{ik} = \frac{p_{ik}^2 / f_k}{\sum_m (p_{im}^2 / f_m)}, \tag{5}$$

where $f_k = \sum_i p_{ik}$ denotes the predicted frequency of each cluster. Fig. 1 shows that in each iteration, the predicted K bins in C is refined by the dense K bins derived from D.

3.2 Generator for Subspace Approximation

Deep generative models aim to approximate the real data space $\boldsymbol{X}$ from a latent space $\boldsymbol{L}$. Consequently, they optimize a non-linear mapping function $f_\theta : \boldsymbol{l} \rightarrow \boldsymbol{x}$, where $\boldsymbol{l}$ denotes the latent vector which encodes the intrinsic attributes of the ambient space, θ denotes the set of parameters.

Traditional generative frameworks approximate a single ambient space by optimizing an aggregated posterior $p_\theta(\boldsymbol{x}|\boldsymbol{z})$, where $\boldsymbol{z}$ denotes the source of incompressible noise in the latent space. In this section, we design a generator $G(\boldsymbol{l})$ to realize the non-linear mapping of $f_\theta : \boldsymbol{l} \rightarrow \boldsymbol{x}$. We demonstrate that the proposed $G(\boldsymbol{l})$ captures informative intrinsic structures, *i.e.*, generates diverse samples from multiple subspaces. More concretely, we express $G(\boldsymbol{l})$ as a deterministic feed forward network $G : \Omega_{\boldsymbol{L}} \rightarrow \Omega_{\boldsymbol{S}}$, where $\Omega_{\cdot}$ denotes the corresponding distribution, $\boldsymbol{L} = \{\boldsymbol{l}_i\}_{i=1}^K$ denotes the latent space and $\boldsymbol{S} = \{\boldsymbol{S}_i\}_{i=1}^K$ denotes the K subspaces of the data $\boldsymbol{X}$. We formulate the optimization process as:

$$p_G(\boldsymbol{x} \in \boldsymbol{S}_i) = \mathbb{E}_{\boldsymbol{l}_i \sim p_{\boldsymbol{L}}}[p_G(\boldsymbol{x}|\boldsymbol{l}_i)], \tag{6}$$

where $p_G(\boldsymbol{x}|\boldsymbol{l}_i) = \mathcal{L}(\boldsymbol{x} - G(\boldsymbol{l}_i))$ and $\boldsymbol{l}_i$ denotes the latent code induced from $\boldsymbol{S}_i$. We finally train the $G(\boldsymbol{l})$ via an adversarial manner such that

$$p_G(\boldsymbol{x}) \approx p_{\boldsymbol{S}_i}(\boldsymbol{x}), \quad \forall \boldsymbol{S}_i \in \boldsymbol{S}. \tag{7}$$

Given the disentangled subspaces $\{\boldsymbol{S}_i\}_{i=1}^K$, we design $\boldsymbol{l}$ to depict the independent attributes of each $\boldsymbol{S}_i$ with a comprehensive combination, *i.e.*,

$$\boldsymbol{l} = \boldsymbol{z} \oplus \boldsymbol{e} \oplus \hat{\boldsymbol{y}}. \tag{8}$$

Here, $\oplus$ denotes the concatenation operation and $\hat{\boldsymbol{y}} \in \mathbb{R}^K$ denotes the one-hot vector in current assignment. The eigenvector $\boldsymbol{e}$ in (3) reflects intrinsic base of a subspace derived from C, where C is updated in each iteration. We set the prior on the noise variable $p_{\boldsymbol{z}}(\boldsymbol{z}) = \mathcal{N}[0, 1]$ where $\mathcal{N}$ denotes normal distribution. The green box in Fig. 1 provides a visualization of this concatenation operation.

3.3 Discriminator for Adversarial Training

GAN [1] is an adversarial framework which trains a deep generative model via a minimax game. Traditional GAN is composed of a generator G and a discriminator D of which the ability of generation or discrimination are mutually improved during training. The G always non-linearly maps a latent noise variable $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})$ to the data space $\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})$. Meanwhile, the discriminator D calculates a probability of belief $p = D(\boldsymbol{x}) \in [0, 1]$ for real samples and assigns a probability of $1-p$ when for generated samples $G(\boldsymbol{z})$. During training, a minimax objective is used to alternatively train both networks:

$$\min_G \max_D \ \mathcal{L}(G, D) \ = \ \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{x}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{9}$$

Here D is optimized to be a binary classifier which provides the optimal probability estimation between real and fake samples, *i.e.*, $\boldsymbol{x}$ and $G(\boldsymbol{z})$. Simultaneously,

G is encouraged to resemble the data distribution, *i.e.*, $G(\boldsymbol{z}) \sim p_{\boldsymbol{x}}(\boldsymbol{x})$, followed by challenging the discriminator D with $G(\boldsymbol{z})$. Both G and D are updated alternatively via back-propagation.

For refining the subspace assignment, we incorporate the adversarial loss with the clustering loss, *i.e.*, $\mathcal{L}_C$ in (4) which discriminates whether the samples are generated from a single $\boldsymbol{S}$. The hybrid Sub-GAN training objective is defined as a minimax optimization:

$$\min_{G,C} \max_{D} \quad \mathcal{L}(D, G, C), \tag{10}$$

where we have

$$\mathcal{L}(D, G, C) = \mathbb{E}_{\boldsymbol{x} \sim p_{\boldsymbol{S}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{l} \sim p_{\boldsymbol{L}}(\boldsymbol{l})}[\log(1 - D(G(\boldsymbol{l})))] + \mathrm{KL}(\boldsymbol{Q}_{\boldsymbol{S}} || \boldsymbol{P}_{\boldsymbol{S}}). \tag{11}$$

Here, $\boldsymbol{S} = \{\boldsymbol{S}_k\}_{k=1}^{K}$ denotes the predicted subspaces in each iteration.

In the training process, we followed the alternating gradient based optimization technique as is used in [1]. Specifically, each module in Sub-GAN is a parametric function with parameters θ_D, θ_G and θ_C, respectively. We jointly optimize the Sub-GAN framework using an alternating stochastic gradient step. For each iteration I, we update the θ_D for the discriminator by calculating the single or multiple steps of the positive gradient direction, *i.e.*, $\nabla_{\theta_D} \mathcal{L}^{I-1}(D, G, C)$. Then, we simultaneously update the parameter θ_G and θ_C for the G and C, respectively. We take a single step in the negative gradient direction $-\nabla_{\theta_G, \theta_C} \mathcal{L}^{I-1}(D, G, C)$. In particular, for the clusterer C, we have

$$\theta_C^I = \{\boldsymbol{C}^I, \boldsymbol{P}^I, \boldsymbol{Q}^I\}. \tag{12}$$

To update the coefficient matrix $\boldsymbol{C}$, we calculate the ℓ_1-norm optimization in (1) in each epoch on top of favorable data features derived from D^{I-1}. Consequently, for the generator G, we update the eigenvector $\boldsymbol{e}_i^I$ for representing each subspace $\boldsymbol{S}_i$ through $\boldsymbol{C}^{I-1}$. For all modules, the first two terms of $\mathcal{L}(D, G, C)$ are calculated based on the mini-batches of n samples $\{\boldsymbol{x}_i \sim p_{\boldsymbol{S}}\}_{i=1}^{n}$ and the latent codes $\{\boldsymbol{l}_i \sim p_{\boldsymbol{L}}\}_{i=1}^{n}$ drawn from the underlying subspaces.

4 Experimental Results

4.1 Datasets and Methods

We conduct experiments on the MNIST and CIFAR-10 datasets. The MNIST is a standard handwritten digits dataset which is composed of $70,000$ images of 28×28 grayscale. We use this dataset to demonstrate the comprehensive characteristics of the proposed Sub-GAN. The CIFAR-10 dataset consists of $60,000$ 32×32 color images in 10 classes, which cover common objects such as airplanes or automobiles. Both datasets are of informative intrinsic attributes apart from the existing label, *e.g.*, the writing style of each digits in MNIST, the various scene of the automobiles in CIFAR-10.

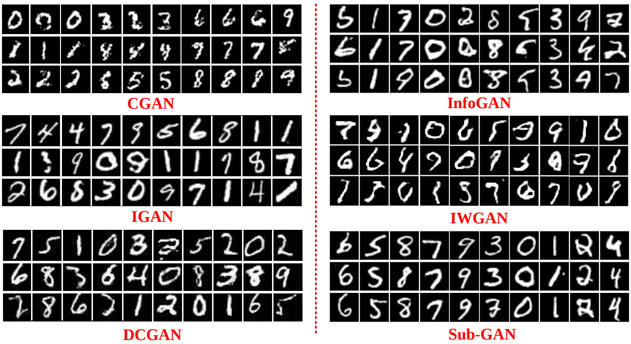


Fig. 2. Generated images on the MNIST dataset by the CGAN [2], InfoGAN [29], IGAN [43], IWGAN [44], DCGAN [45] and the proposed Sub-GAN. The first two methods explore the class information during the generation, however, the generated digits do not look visually appealing. While the the samples from the others look better, the diversity is hard to control. In contrast, the proposed Sub-GAN can simultaneously discover the subspaces $\{\boldsymbol{S}_i\}_{i=1}^{K}$ and generate diverse samples from each $\boldsymbol{S}_i$.

For evaluating the generation quality, we compare the proposed Sub-GAN with various state-of-the-art generative models, *i.e.*, CGAN [2], Improved GAN (IGAN, [43]), Improved WGAN (IWGAN, [44]), DCGAN [45] and InfoGAN [29]. Furthermore, we perform experiments to evaluate the unsupervised clustering performance of the Sub-GAN. We make the comparison with K-means, SSC [4], LSR [46], SMR [47], NSN [48], SSC-OMP [35], ORGEN [31], iPursuit [49], DEC [39], CatGAN [50] and InfoGAN [29]. Here, the SSC, LSR, SMR, NSN, SSC-OMP, ORGEN and iPursuit are subspace clustering algorithms. The DEC concentrate on deep embedding clustering while the CatGAN and InfoGAN are based on the generative models.

4.2 Evaluation Metrics

We employ various metrics to quantitatively evaluate the proposed Sub-GAN in terms of both generation and clustering capacity. Specifically, we assess the image quality by using the Inception Score [43] and Diversity Score [51]. We then quantify the clustering assignments by calculating the Adjusted Accuracy [52].
Inception Score: The inception score [43] is widely adopted in evaluating generative tasks which uses a pre-trained neural network classifier to capture both highly classifiable and diverse properties with respect to class labels. For evaluated samples, it calculates average KL divergences between conditional label distributions (expected to have low entropy for easily classifiable samples) and marginal distribution (expected to have high entropy if all classes are equally presented). We follow the same routine in [53] for evaluation, *i.e.*, using the Inception network [54] trained on the ImageNet dataset [55].
Diversity Score The diversity score [51] is based on the cosine distances among features (the maximum score is 5). In this paper, we use it to quantitatively

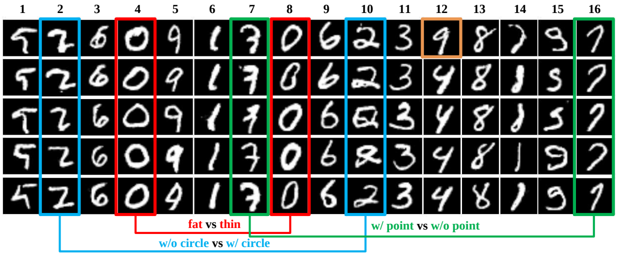


Fig. 3. Samples generated from joint unsupervised training on the MNIST dataset using the proposed Sub-GAN. Here, we set $K = 16$ (*top row*) to disentangle diverse subspaces $\{\boldsymbol{S}_i\}_{i=1}^{16}$. Note we construct this figure according to the sequence derived from the clusterer C. The *bottom* illustrates three pairs of different writing styles of digits $2, 0$ and 7. The Sub-GAN can discover not only the 10 subspaces with the digits $0 - 9$, but the different writing style, *e.g.*, the fat and thin '0' in the 4-th and 8-th, respectively. The brown box reflects the failure case which confuses between 4 and 9.

evaluate the diversity of generated samples thus validate the alleviation on mode collapse problem in GAN training.

Adjusted Accuracy for Clustering The Adjusted Accuracy [52] is a common metric for evaluating the clustering performance when $K \neq K_g$ where K_g denotes the ground-truth number of clusters. For each cluster $\boldsymbol{S}_k$, we found the validation example $\boldsymbol{x}_i$ that maximizes $q(y_k|\boldsymbol{x}_i)$, and assigned the label of $\boldsymbol{x}_i$ to all the points in the cluster $\boldsymbol{S}_k$. We then compute the test accuracy based on the assigned class labels. Note it is identical to standard clustering accuracy when $K = K_g$.

4.3 Network Architecture

The generator G is mainly composed of two deconvolution layers (deconv) and two fully connected layers (FC). Specifically, the input latent vector is $\boldsymbol{l} \in \mathbb{R}^{110}$. Afterwards, the network architecture of G is: (1) FC.1024 w/ ReLU and batch-norm; (2) FC.6272 w/ ReLU and batch-norm; (3) reshape to $7 \times 7 \times 128$; (4) deconv.4×4, stride$= 2$, feature maps$= 64$, w/ ReLU and batch-norm; (5) deconv.4×4, stride$= 2$, feature maps$= 1$.

The discriminator D is mainly composed of two convolution layers (conv) and two fully connected layers. In particular, the input image size is 28×28 with 1 gray channel. The network architecture of D is: (1) conv.4×4, stride$= 2$, feature maps$= 64$, w/ lReLU; (2) conv.4×4, stride$= 2$, feature maps$= 128$, w/ lReLU and batch-norm; (3) FC.1024 w/ lReLU and batch-norm; (4) FC.1 for classifying whether the image is real.

The clusterer C shares a similar structure with D. However, the last layer of C, *i.e.*, $FC.K$, is designed to calculate a K-bins probabilities of the K subspaces.

4.4 Implementation Details

To setup the experiments of the proposed joint framework, we first initialize the soft assignments of the subspaces $\{\hat{\boldsymbol{S}}_i\}_{i=1}^{K}$ by employing an unsupervised

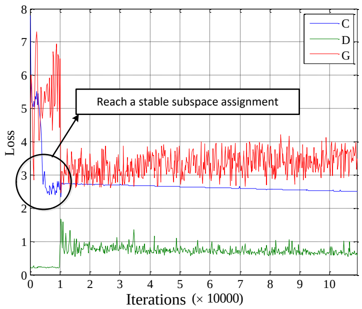


Fig. 4. Optimization losses of three modules, *i.e.*, C (*blue*), D (*green*) and G (*red*), over training iterations on the MNIST dataset. The $\mathcal{L}_C$ demonstrates a downward trend before around $10,000$-th iteration. Sequentially, the training of G and D is unstable, *e.g.*, D can easily discriminate the fake images from the real one so that the $\mathcal{L}_D$ is low. After C reaches a stable subspace assignment, the framework begins a normal adversarial training of G, D and C.

subspace clustering termed SMR [47] on the raw data space. Then, for stabilizing the training of Sub-GAN, we design G and D based on state-of-the-art techniques in DCGAN [45] and InfoGAN [29]. Specifically, we construct both networks with multiple convolution and deconvolution layers, followed by ReLU and leaky RelU (lReLU) activations in G and D, respectively. We also incorporate batch normalizations in both networks. We train the proposed joint model for $100,000$ iterations, with the batch size 100. We provide more training details in the supplementary material.

4.5 Generated Images by Sub-GAN

In this section, we analyze the generation performance of the proposed Sub-GAN on both MNIST and CIFAR-10 datasets.

Different K's on MNIST We first conduct experiments on the MNIST dataset. Fig. 2 shows the visualized comparison among samples derived from five contrastive generative models and the proposed Sub-GAN. The CGAN and InfoGAN generate samples from a union of subspaces. However, the samples derived from CGAN have unsatisfied consistency to human judgments, *e.g.*, the components of the digits are broken in several cases. In addition, CGAN relies on the strong supervision of the annotations, which can only accessed on limited applications. The generated digits from IGAN, DCGAN or IWGAN have satisfied quality, yet the algorithm can not discover informative subspaces of the ambient space. As a result, the attribute of generated samples is hard to control. With $K = 10$, the proposed Sub-GAN framework generates diverse samples from each subspace, which alleviate the mode collapse problem in training GANs.

In Fig. 3, we also demonstrate that the proposed Sub-GAN can discover informative visual attributes which can hardly be annotated by strong supervi-

Table 1. Comparison of the diversity scores on both MNIST and CIFAR datasets with $K = 10$. The proposed Sub-GAN achieves best performance against contrastive methods, which alleviates the mode collapse problem in training GANs. The column of 'Real' indicates the diversity scores of real images from respective datasets.

Datasets	Real	CGAN [2]	IGAN [43]	IWGAN [44]	DCGAN [45]	InfoGAN [29]	Sub-GAN
MNIST	2.96	0.92	1.81	1.78	1.63	2.11	**2.36**
CIFAR	3.21	1.02	2.20	2.03	1.95	2.48	**2.72**

sion. As a result, the algorithm handles the mode collapse problem in an unsupervised way. In the experiment, we set $K = 16$ and generate images on the MNIST dataset. We can see that the Sub-GAN discovers multiple writing styles for the digits, and thus generates diverse new samples for different attributes. For example, the red boxes reflects that the digit '0' are divided into two types, *i.e.*, the fat '0' and the thin one. It also finds several writing styles for other digits such as '2' and '7'. We provide more results of different attributes in the supplementary material.

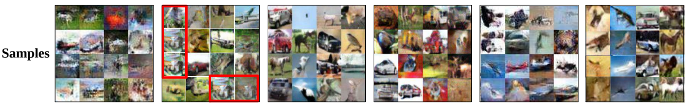
Furthermore, for quantitatively evaluating the diversity of the generated digits, we calculate the diversity scores among contrastive methods and report them in Table 1. While Sub-GAN achieves the best performance on this metric, it demonstrate that the proposed method alleviates the mode collapse problem due to the incorporation of subspace analysis.

To exposure the training process, we visualize the optimization loss in Fig. 4. The clusterer is iteratively optimized in about the first 10^4 iterations. During this process, the training of both generator and discriminator is unstable, *i.e.*, the loss of G is high and unstable while the loss of D is close to 0. It reflects that the generated samples are not visually appealing and can be easily discriminate by D. After reaching a stable subspace assignment, the joint unsupervised model starts a normal adversarial training of all modules.

CIFAR-10 In this section, we conduct experiments on the CIFAR-10 dataset. We set $K = 10$ in the training procedure and show the example results in Fig. 5. We also collect the generated samples from existing frameworks and calculate the inception score for each of them.

The Sub-GAN achieves favorable performance on the metric of inception score, which demonstrates the consistency to human judgments and thus the effectiveness of our method in terms of the generation ability. The proposed model can also generate samples from each subspace, which handles the mode collapse problem. The samples generated from the IGAN get a slight higher score than Sub-GAN, however, the samples in red boxes reflect the mode collapse problem of IGAN, *i.e.*, many generated samples are very similar. The other methods suffer from the same problem, while the quality of generated images is lower than ours. We provide more comparisons of the generated samples derived from the contrasted methods in the supplementary material.

We also quantitatively evaluate the diversity of generated samples on CIFAR dataset in Table 1, where the proposed Sub-GAN shows favorable performance



Models	CGAN [2]	IGAN [43]	IWGAN [44]	DCGAN [45]	InfoGAN [29]	Sub-GAN
Inception Score	4.28±0.08	8.09±0.07	7.86±0.07	6.16 ±0.07	7.26±0.05	7.95±0.04

Fig. 5. Inception scores for samples derived from various generative models on the CIFAR-10 dataset. Higher score indicates more consistence with human judgment. The experimental results demonstrate that the proposed Sub-GAN generates favorable samples against other state-of-the-art methods in terms of both visual expression and diversity. The IGAN achieves state-of-the-art inception score, however, the red boxes in the figure shows that it suffers from the mode collapse problem, which is tackled by Sub-GAN via subspace analysis.

against contrastive methods. The results of diversity scores show consistency to the visualized results in Fig. 5. For example, the diversity score of IGAN is lower than the proposed Sub-GAN (2.20 vs 2.72).

4.6 Image Clustering Performance

The clusterer is an auxiliary module which disentangles the subspaces to facilitate the generation. An effective interaction among G, D and C is important for both generating samples and clustering. In this section, we analyze the clustering performance of the proposed Sub-GAN on both MNIST and CIFAR datasets.

In this paper, the clusterer updates the clustering assignment of the whole dataset in each epoch, while the D refine the assignment of one mini-batch in each iteration. Some samples might be wrongly grouped based on the global similarity to all others, hence we refine the assignment in D based on the similarity of samples in local batches. We have conducted an ablation studies on the MNIST dataset with $K = 10$ and reported the clustering accuracy (%) in Table 2, which demonstrates the effectiveness of the refinement operation.

We report the adjusted accuracy of contrastive methods on both datasets in Table 3. The K-means method does not perform well on this task, since it lacks the ability on handling the high-dimensional large-scale data. In contrast, the subspace clustering (SSC, LSR, SMR, NSN, SSC-OMP, ORGEN and iPursuit), the deep embedding based clustering method (DEC) and the generation based methods (CatGAN and InfoGAN) show better performance due to the distinctive representation or iteratively optimization. In contrast, the proposed Sub-GAN achieves favorable performance against contrastive methods under all configurations, since the deep representation is iteratively updated according to the guidance of adversarial training process.

Note in the experiment with $K = 10$, the accuracy of initialized assignment using SMR is 73.39% for MNIST and 56.24% for CIFAR, while the joint training of three modules induces about 12% and 22% improvement, respectively. Consequently, the information interaction facilitates not only the generation of diverse

Table 2. Clustering accuracy (%) on the MNIST dataset under $K = 10$ with/without the refinement operation in the discriminator.

Refinement in D	1st Epoch	20th Epoch	40th Epoch	Last Epoch
W/o	75.23	82.96	83.11	83.87
W/	**77.12**	**83.45**	**84.24**	**85.32**

Table 3. Unsupervised clustering performance (adjusted clustering accuracy) of the contrasted methods on the MNIST and CIFAR datasets with different K's. The clusterer in Sub-GAN shows favorable performance against various clustering algorithms.

Methods	MNIST			CIFAR		
	$K = 10$	$K = 16$	$K = 20$	$K = 10$	$K = 16$	$K = 20$
K-means	53.49	60.36	62.55	42.62	46.81	51.02
SSC [4]	62.71	66.82	70.19	50.31	52.77	53.98
LSR [46]	66.85	70.21	73.83	53.97	55.80	59.24
SMR [47]	73.39	81.27	83.63	56.24	59.02	62.73
NSN [48]	68.75	71.04	73.67	52.29	56.55	59.03
SSC-OMP [35]	76.33	79.25	82.52	51.21	53.02	57.84
ORGEN [31]	71.04	74.07	78.65	52.29	55.61	58.08
iPursuit [49]	61.35	64.28	68.84	59.21	62.53	65.66
DEC [39]	84.30	83.28	83.02	61.03	65.29	67.31
CatGAN [50]	80.21	84.92	90.30	67.42	67.85	68.76
InfoGAN [29]	70.63	73.77	78.69	71.02	73.64	74.07
Sub-GAN	**85.32**	**90.36**	**90.81**	**78.95**	**81.35**	**82.44**

samples but also the clustering performance. In addition, for both datasets, the Sub-GAN shows better performance with increasing K's, since the model can disentangle informative subspace structure with a large number of clusters.

5 Conclusions

In this work, we present an unsupervised Sub-GAN model for jointly learning the latent subspaces of the ambient space and generating instances correspondingly. We incorporate a novel clusterer into the GAN framework, where the clusterer disentangles the subspaces and is updated based on the deep representations of the samples derived from the discriminator. Meanwhile, the generator is fed with both random vectors of a normal distribution and low-dimensional eigenvectors derived from the clusterer. Here, the eigenvectors reflect latent structures of the disentangled subspaces. The discriminator is sequentially designed to reward high scores for samples which fit a specific subspace distribution, and provide feedbacks to refine the cluster assignment. Both quantitative evaluation and the visualization demonstrate that Sub-GAN can not only discover meaningful latent subspaces of the datasets but also generate photo-realistic and diverse images.

Acknowledgments

This work was supported in part by NSFC under Grants 61620106008, 61572264, NSF CAREER under Grant 1149783, the Open Project Program of the National Laboratory of Pattern Recognition and the Natural Science Foundation of Tianjin, China, under Grant 18JCYBJC15400.

References

1. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. (2014) 1, 3, 7, 8
2. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) 2, 3, 9, 12
3. Vidal, R.: Subspace clustering. IEEE Signal Processing Magazine **28**(2) (2011) 52–68 2
4. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR. (2009) 2, 3, 5, 6, 9, 14
5. Yu, L., Zhang, W., Wang, J., Yu, Y.: SeqGAN: Sequence generative adversarial nets with policy gradient. In: AAAI. (2017) 2
6. Shen, W., Liu, R.: Learning residual images for face attribute manipulation. In: CVPR. (2017) 2
7. Dong, H., Yu, S., Wu, C., Guo, Y.: Semantic image synthesis via adversarial learning. In: ICCV. (2017) 2
8. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Smolley, S.P.: Least squares generative adversarial networks. In: ICCV. (2017) 2
9. Deng, Z., Zhang, H., Liang, X., Yang, L., Xu, S., Zhu, J., Xing, E.P.: Structured generative adversarial networks. In: NIPS. (2017) 2
10. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: ECCV. (2016) 2
11. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: ICCV. (2017) 2
12. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. (2017) 2
13. Luc, P., Couprie, C., Chintala, S., Verbeek, J.: Semantic segmentation using adversarial networks. In: NIPS. (2016) 2
14. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR. (2017) 2
15. Donahue, J., Krähenbühl, P., Darrell, T.: Adversarial feature learning. In: ICLR. (2017) 3
16. Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A., Clune, J.: Plug & play generative networks: Conditional iterative generation of images in latent space. In: CVPR. (2017) 3
17. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: ECCV. (2016) 3
18. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: ICML. (2010) 3
19. Tieleman, T.: Training restricted boltzmann machines using approximations to the likelihood gradient. In: ICML. (2008) 3
20. Rifai, S., Vincent, P., Muller, X., Glorot, X., Bengio, Y.: Contractive auto-encoders: Explicit invariance during feature extraction. In: ICML. (2011) 3
21. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation **18**(7) (2006) 1527–1554 3
22. Lee, H., Ekanadham, C., Ng, A.Y.: Sparse deep belief net model for visual area V2. In: NIPS. (2008) 3

23. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786) (2006) 504–507 3
24. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. In: ICLR. (2017) 3
25. Arjovsky, M., Bottou, L.: Towards principled methods for training generative adversarial networks. In: ICLR. (2017) 3
26. van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with pixelCNN decoders. In: NIPS. (2016) 3
27. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier GANs. In: ICML. (2017) 3
28. Yan, X., Yang, J., Sohn, K., Lee, H.: Attribute2image: Conditional image generation from visual attributes. In: ECCV. (2016) 3
29. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In: NIPS. (2016) 3, 9, 11, 12, 14
30. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations Newsletter **6**(1) (2004) 90–105 3, 5
31. You, C., Li, C.G., Robinson, D.P., Vidal, R.: Oracle based active set algorithm for scalable elastic net subspace clustering. In: CVPR. (2016) 3, 9, 14
32. Patel, V.M., Van Nguyen, H., Vidal, R.: Latent space sparse subspace clustering. In: CVPR. (2013) 3
33. Yang, J., Parikh, D., Batra, D.: Joint unsupervised learning of deep representations and image clusters. In: CVPR. (2016) 3
34. Dizaji, K.G., Herandi, A., Huang, H.: Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. ICCV (2017) 3, 4
35. You, C., Robinson, D., Vidal, R.: Scalable sparse subspace clustering by orthogonal matching pursuit. In: CVPR. (2016) 3, 9, 14
36. Elhamifar, E., Vidal, R.: Sparse subspace clustering: Algorithm, theory, and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(11) (2013) 2765–2781 4
37. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS. (2002) 4
38. Peng, X., Xiao, S., Feng, J., Yau, W.Y., Yi, Z.: Deep subspace clustering with sparsity prior. In: IJCAI. (2016) 4
39. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: ICML. (2016) 4, 9, 14
40. Nguyen, T.D., Le, T., Vu, H., Phung, D.: Dual discriminator generative adversarial nets. In: NIPS. (2017) 4
41. Roth, K., Lucchi, A., Nowozin, S., Hofmann, T.: Stabilizing training of generative adversarial networks through regularization. In: NIPS. (2017) 4
42. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing **17**(4) (2007) 395–416 6
43. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: NIPS. (2016) 9, 12
44. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein GANs. In: NIPS. (2017) 9, 12
45. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015) 9, 11, 12
46. Lu, C.Y., Min, H., Zhao, Z.Q., Zhu, L., Huang, D.S., Yan, S.: Robust and efficient subspace segmentation via least squares regression. In: ECCV. (2012) 9, 14

47. Hu, H., Lin, Z., Feng, J., Zhou, J.: Smooth representation clustering. In: CVPR. (2014) 9, 11, 14
48. Park, D., Caramanis, C., Sanghavi, S.: Greedy subspace clustering. In: NIPS. (2014) 9, 14
49. Rahmani, M., Atia, G.K.: Innovation pursuit: A new approach to subspace clustering. In: ICML. (2017) 9, 14
50. Springenberg, J.T.: Unsupervised and semi-supervised learning with categorical generative adversarial networks. In: ICLR. (2016) 9, 14
51. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation. In: NIPS. (2017) 9
52. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial autoencoders. In: ICLR. (2016) 9, 10
53. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks. arXiv preprint arXiv:1706.04987 (2017) 9
54. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. (2016) 9
55. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. (2009) 9