IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 11, 2018, accepted September 15, 2018, date of publication September 28, 2018,
date of current version October 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2872695

SCGAN: Disentangled Representation Learning by Adding Similarity Constraint on Generative Adversarial Nets

XIAOQIANG LI[1,2,3], (Member, IEEE), LIANGBO CHEN[1], LU WANG[1], PIN WU[1], AND WEIQIN TONG[1,2]

[1]School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
[2]Shanghai Institute for Advanced Communication and Data Science, Shanghai University, Shanghai 200444, China
[3]Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 200433, China

Corresponding author: Xiaoqiang Li (xqli@shu.edu.cn)

This work was supported by the Shanghai Innovation Action Plan Project under Grant 16511101200.

ABSTRACT We proposed a novel generative adversarial net called similarity constraint generative adversarial network (SCGAN), which is capable of learning the disentangled representation in a completely unsupervised manner. Inspired by the smoothness assumption and our assumption on the content and the representation of images, we design an effective similarity constraint. SCGAN can disentangle interpretable representations by adding this similarity constraint between conditions and synthetic images. In fact, similarity constraint works as a tutor to instruct generator network to comprehend the difference of representations based on conditions. SCGAN successfully distinguishes different representations on a number of datasets. Specifically, SCGAN captures digit type on MNIST, clothing type on Fashion-MNIST, lighting on SVHN, and object size on CIFAR10. On the CelebA dataset, SCGAN captures more semantic representations, e.g., poses, emotions, and hair styles. Experiments show that SCGAN is comparable with InfoGAN (another generative adversarial net disentangles interpretable representations on these datasets unsupervisedly) on disentangled representation learning. Code is available at https://github.com/gauss-clb/SCGAN.

INDEX TERMS Generative adversarial nets, representation learning, unsupervised learning.

I. INTRODUCTION

Representation learning [1] is an important field of machine learning, whose goal is to find an appropriate representation of data in order to perform a machine learning task. Many downstream tasks can benefit from representation learning, e.g., speech recognition, object recognition, natural language processing, transfer learning. Representation basically can be divided into disentangled representation and distributed representation. Loosely the disentangled representation tends to disentangle the factors of variation which the distributed representation doesn't.

There is a large amount of work involving representation learning, which is introduced in Section II. A significant fraction of representation learning research is driven by generative models, since the ability to synthesize the more realistic data entails some form of understanding of the representation. And the most prominent models are the variational autoencoder (VAE) [2] and the generative adversarial network (GAN) [3]. For example, conditional GAN [4] learns disentangled representations of data by conditioning the model on the additional information (supervisedly), InfoGAN [5] learns disentangled representations by maximizing the mutual information of conditions and synthetic images (unsupervisedly). In this paper, we propose a novel approach which drives GAN to learn disentangled representation unsupervisedly.

The smoothness assumption [1] assumes that the function to be learned f is s.t. $x \approx y$ generally implies $f(x) \approx f(y)$, which is the most basic prior in most machine learning. Motivated by this assumption, we desire to learn the relation mapping that the similar representations is corresponding to the similar conditions, by adding the similarity constraint on synthetic images given the conditions. In order to design an effective similarity constraint, we make an assumption that an image contains content and representation which are controlled by noise $\boldsymbol{z}$ and condition $\boldsymbol{c}$ respectively, and this assumption is validated by our experiments, see Sub-Section IV-A. We refer to our model as SCGAN (Similarity

Constraint Generative Adversarial Network), as it utilizes the similarity constraint in the training process. In addition, we apply our method on a number of image datasets and find that the experimental results match or overpass InfoGAN, see Section V.

In summary, the contribution of the paper is as follows:

- We propose a novel generative model (called SCGAN) to disentangle representations unsupervisedly.
- We make an assumption that an image contains content and representation which are controlled by noise $\boldsymbol{z}$ and condition $\boldsymbol{c}$ respectively.
- We switch similarity constraint on representation difference to pixel-level difference, the latter is easier to get by using metric on pixel space.
- We conduct experiments on five well-known datasets, several semantic representations are separated from other variations and results show good performance.
- We conduct an stability experiments on the processed MNIST (digits are shifted randomly), the result shows SCGAN is more stable than InfoGAN.

This paper is organized as follows. In Section II, we review several relevant work on distributed representation learning and disentangled representation learning. In Section III, we review GAN, conditional GAN (basis of SCGAN) and InfoGAN (similar models as SCGAN). In Section IV, we describe how to add the similarity constraint on GAN and give an intuitive interpretation why it works. In Section V, we first compare SCGAN with prior approaches (e.g., conditional GAN, InfoGAN) on relatively clean datasets (e.g., MNIST, Fashion-MNIST) and then show that SCGAN can learn interpretable representations on complicated datasets (e.g., SVHN, CelebA) as InfoGAN, and we also attempt to learn interpretable representations on CIFAR10 (a dataset of natural images) which InfoGAN [5] struggles to learn. Then, we add experiments to verify our assumptions and an stability testing to show the robustness of SCGAN. Finally, we will make a conclusion and discuss the future work in Section VI.

II. RELATED WORK

Distributed representation and disentangled representation are two forms of representations. There is a lot of work on distributed representation learning and disentangled representation learning in the past years.

Some research are conducted on distributed representation. For example, autoencoders encode images as embedded features and then reconstruct inputs from these embeddings. There exists a large body of variants of autoencoders (e.g., denoising autoencoder [6], sparse autoencoder [7]) to increase robustness of embeddings. Restricted Boltzmann machines [8] use energy function to define probability distributions over hidden and visible vectors, which are applied to dimensionality reduction and feature learning. More recently, VAE [2] attempts to map data to a distribution and reconstruct inputs by decoding latent variables draw from the distribution. DCGAN [9] learn an image representation that supports basic linear algebra on code space.

Another line of work is to disentangle representations. Some prior research attempt to disentangle representations using supervised data. For example, bilinear models [10] separate style and content within face images, multi-view perceptron [11] separate face identity and view point. Dosovitskiy *et al.* [12] propose training a convolutional neural network to generate 2D projections of the chairs given the chair type, viewpoint, and other parameters. Similarly, conditional VAE [13], [14] and Adversarial Autoencoders [15] are shown to separate digit types from other variations on MNIST and SVHN, conditional GAN [4] also separates digit types on MNIST but have a little bit difficulty on SVHN, as the background of images on SVHN is more complicated than that on MNIST.

There are several weakly supervised methods to remove the need of explicitly labeling variations. DisBM [16] is a higher-order Boltzmann machine which learns the disentangled representation by clamping hidden units for data points matched in some factor of variation. This concept of clamping is very similar to the idea that fix noise variables and vary condition variables while capturing representations in almost GANs for representation learning. DC-IGN [17] extends this clamping idea to VAE and learns interpretable graphics codes for complex transformation (e.g., object out-of-plane rotations and lighting variations).

Some unsupervised methods to learn disentangled representation are developed recently. Desjardins *et al.* [18] propose the hossRBM based on the spike-and-slab restricted Boltzmann machine that can disentangle emotion from identity on the Toronto Face Dataset (TFD) [19]. However, hossRBM can only disentangle discrete latent factors. InfoGAN [5] is proposed to disentangle both discrete and continuous latent factors by maximizing the mutual information between a fixed small subset of the GAN's noise variables and the observations. However, InfoGAN introduces an extra network to simulate the variational lower bound of the mutual information, that makes network architectures more complicated and hard to extend. And when many representations are to be disentangled, InfoGAN may need more networks, as each network disentangles a representation. Although we can use one network and share parameters, it may make representations mixed together and prevent from disentangling representations. Our SCGAN only adds a regularization (similarity constraint) to a vanilla GAN, that make it easy to integrate Wasserstein GAN (WGAN) [20] and train on complicated datasets.

III. BACKGROUND

Generative Adversarial Networks (GANs) [3] are very powerful generative models in recent years. The idea behind them is to shorten the distance (e.g., Jenson-Shannon divergence, Earth-Mover (also called Wasserstein) distance [20]) between generative distribution and data distribution through a two-player minimax game. Vanilla GAN has many issues, e.g., vanishing gradient, mode collapse, blurred image generation quality. Therefore, there are many variants of

GANs [9], [20]–[23] capable of addressing these issues.

A. GENERATIVE ADVERSARIAL NETWORK

Generative Adversarial Network is proposed by Goodfellow *et al.* [3] which is designed to learn a generator distribution $\mathbb{P}_g$ that matches the real data distribution $\mathbb{P}_{data}$. A generator network G maps noise z drawn from $\mathbb{P}_{noise}$ to input data. The generator is trained to make generative data $\tilde{\boldsymbol{x}} = G(z)$ close to real data $\boldsymbol{x}$ so that it can cheat the discriminator. A discriminator network D receives the generative data or the true data and distinguishes them as far as possible.

Formally, the objective of minimax game between generator G and discriminator D is given as follow:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{data}}[logD(\boldsymbol{x})] + \mathbb{E}_{z\sim\mathbb{P}_{noise}}\left[1 - logD(\tilde{\boldsymbol{x}})\right] \quad (1)$$

B. CONDITIONAL GAN

Conditional GAN (CGAN) introduces some extra information $\boldsymbol{c}$ to both the generator and the discriminator. $\boldsymbol{c}$ is a discrete variable which represents some discrete attributes, e.g., class labels. Since both noise z and condition $\boldsymbol{c}$ are fed into the generator and the discriminator, they are now $G(z, \boldsymbol{c})$ and $D(\boldsymbol{x}, \boldsymbol{c})$ respectively. From probabilistic point of view, the data generated by the generator subject to the conditional probability distribution $\mathbb{P}_{G(z|\boldsymbol{c})}$. Note the discriminator requires ground truth $\boldsymbol{c}$ for real data $\boldsymbol{x}$. Hence, it behaves like the supervised learning.

The objective of conditional GAN is a little different from vanilla GAN:

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim\mathbb{P}_{data}}[logD(\boldsymbol{x},\boldsymbol{c})] + \mathbb{E}_{z\sim\mathbb{P}_{noise}}\left[1 - logD(\tilde{\boldsymbol{x}},\boldsymbol{c})\right] \quad (2)$$

where $\tilde{\boldsymbol{x}} = G(z, \boldsymbol{c})$.

C. INFORMATION GAN

InfoGAN is a variant of GAN that learns interpretable representations unsupervisedly by maximizing the mutual information between the conditional variables and the generative data, i.e. $I(\boldsymbol{c}, G(z, \boldsymbol{c}))$. The mutual information is assigned as the regularization of CGAN's loss:

$$\min_G \max_D V(D,G) - \lambda I(\boldsymbol{c}, G(z,\boldsymbol{c})) \quad (3)$$

where λ is the hyperparameter and $V(D, G)$ is the objective of CGAN except that the discriminator of InfoGAN doesn't take $\boldsymbol{c}$ as input.

But it is difficult to maximize $I(\boldsymbol{c}, G(z, \boldsymbol{c}))$ directly as it requires access to the posterior $P(\boldsymbol{c}|\boldsymbol{x})$. Fortunately, they obtain a variational lower bound by the technique known as Variational Information Maximization [24]. They define an auxiliary distribution $Q(\boldsymbol{c}|\boldsymbol{x})$ to approximate $P(\boldsymbol{c}|\boldsymbol{x})$ and $Q(\boldsymbol{c}|\boldsymbol{x})$ can be parametrized as a neural network. Then the lower bound is easy to approximate with Monte Carlo simulation. More details can be referred to their paper [5].

IV. MODEL

A. ASSUMPTION

An image consists of content and representation. Vanilla GAN only manipulates noise z to generate images $G(z)$, which means z controls the mixture of content and representation. However, conditional GANs manipulate both noise z and condition $\boldsymbol{c}$ to generate images $G(z, \boldsymbol{c})$. We give out an assumption that z controls content and $\boldsymbol{c}$ controls representation respectively. We define the difference of content(DC) and the difference of representation(DR) on two synthetic images $G(z_i, \boldsymbol{c}_i)$, $G(z_j, \boldsymbol{c}_j)$. Under this assumption, DC merely depends on z_i, z_j and DR on $\boldsymbol{c}_i, \boldsymbol{c}_j$, i.e., $DC(z_i, z_j)$, $DR(\boldsymbol{c}_i, \boldsymbol{c}_j)$. Apparently, the difference(Sim, defined in Section IV-B) between $G(z_i, \boldsymbol{c}_i)$ and $G(z_j, \boldsymbol{c}_j)$ equals to the difference of content plus the difference of representation.

$$Sim(G(z_i,\boldsymbol{c}_i), G(z_j,\boldsymbol{c}_j)) = DC(z_i,z_j) + DR(\boldsymbol{c}_i,\boldsymbol{c}_j) \quad (4)$$

It seems that there is no proper theory can explain why noise z and condition $\boldsymbol{c}$ control different information. This assumption may be dependent on datasets, as these well-known datasets are highly processed, e.g., objects are center aligned and background is relatively single. To our knowledge, we argue that the network is inclined to assign slightly varying attributes (representations) to condition $\boldsymbol{c}$, as $\boldsymbol{c}$ is constrained. On the contrary, the network will assign content (hugely changing) to noise z, as z is unconstrained. Similarly, InfoGAN only adds the mutual information between the condition and synthetic images, it is also hard to explain why the condition control the representation rather than the content or anything else. Although lacking theoretical support, we still conduct an experiment to verify this assumption, see Section V-D.

B. SIMILARITY

Similarity($Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$) is defined on a pair of images $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^n$, in order to measure the difference between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. And $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$ can be any form as you like, as long as the following condition is satisfied.

1) CONDITION 1.

- If $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are similar on the measure space, $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is small, otherwise $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is large.
- Similarity is symmetric, i.e., $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j) = Sim(\boldsymbol{x}_j, \boldsymbol{x}_i)$.

Similarity constraint is added on synthetic images $G(z, \boldsymbol{c})$, and we argue that $G(z, \boldsymbol{c})$ is continuous with respect to z and $\boldsymbol{c}$. Inspired by the smoothness assumption, we argue that when z (content of images) or $\boldsymbol{c}$ (representation of images) changes smoothly, similarity also changes smoothly. It means any measurement with smooth characteristics on pixel space is a good choice, e.g., Euclidean distance($\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2$) and Gaussian radial basis function($\exp\left(-\frac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2}{2\sigma^2}\right), \sigma > 0$).

C. CONDITIONAL VARIABLE

Conditional variable can be discrete or continuous, which is sampled from a prior distribution. Discrete conditional

variable can capture the most obvious difference among real data, especially discrete attributes (e.g. class labels). In contrast, continuous conditional variable tends to capture the slowly changing attributes (e.g. azimuth of face, size of object). Therefore, discrete conditional variable is sampled from a multinomial distribution and encoded as an one-hot vector, but continuous conditional variable is sampled from a continuous distribution (e.g. uniform distribution).

For supervised learning, one knows labels of real data, so they can prompt network to learn conditional variable and control representations (e.g. class labels) of synthetic data by varying the conditional variable. Unfortunately, most data don't have sufficient labels due to a tremendous amount of labeling work. In this situation, one needs to use some unsupervised learning techniques to make network learn the conditional variable by itself without any labels. This is not the same as the former because one doesn't know any information about labels, so that anyone won't know what attributes the network will learn. In other words, one can't control representations of synthetic data, i.e., representations can be any attributes (e.g., lighting, shape, style).

In this work, we attempt to train SCGAN on both discrete and continuous variable. There is a little difference on the similarity constraint, see Section IV-D.

D. CONSTRAINT

It is the main idea of this paper. Let us define SC(similarity constraint) as follows:

If conditional variable is discrete, then

$$SC(\boldsymbol{x}, \boldsymbol{c}) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\langle \boldsymbol{c}_i, \boldsymbol{c}_j \rangle Sim(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{1 - \langle \boldsymbol{c}_i, \boldsymbol{c}_j \rangle}{Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right). \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes inner product, N is the batch-size of $\boldsymbol{x}$ and $\boldsymbol{c}$.[1]

If conditional variable is continuous, then

$$SC(\boldsymbol{x}, c) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\left(1 - \left|c_i - c_j\right|\right) Sim(\boldsymbol{x}_i, \boldsymbol{x}_j) + \frac{\left|c_i - c_j\right|}{Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right). \tag{6}$$

where $|\cdot|$ denotes absolute value, N is the batch-size of $\boldsymbol{x}$ and c.[2]

The similarity constraint will fit in GAN loss as a regularization and be minimized. Our motivation is to add the constraint between the difference of condition($\boldsymbol{c}$) and the difference of representation(DR), but it's not easy to get DR since the difference of content(DC) is mixed. Fortunately, constraint is added on $(\boldsymbol{c_i}, \boldsymbol{c_j})$, which means $(\boldsymbol{z_i}, \boldsymbol{z_j})$ isn't affected while optimizing $\boldsymbol{c}$, i.e., DC is fixed. That is to say, DR is equivalent to Sim according to (4) when DC is fixed.

Therefore, the definition of similarity constraint has an intuitive interpretation: In the discrete case, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have the different representations(DR is large, i.e., Sim is large), then the inner product $\langle \boldsymbol{c}_i, \boldsymbol{c}_j \rangle = 0$ due to orthogonality of one-hot vector for different representations, otherwise $\langle \boldsymbol{c}_i, \boldsymbol{c}_j \rangle = 1$. In other words, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have the different representations, only item $\frac{1}{Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)}$ contributes to $SC(\boldsymbol{x}, \boldsymbol{c})$, and minimizing $SC(\boldsymbol{x}, \boldsymbol{c})$ is equivalent to maximize $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$. In contrast, if $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have the same representations, minimizing $SC(\boldsymbol{x}, \boldsymbol{c})$ is equivalent to minimize $Sim(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This exactly matches the Condition IV-B.1 satisfied by similarity. And in the continuous case, $1 - |\cdot|$ plays the same role as $\langle \cdot, \cdot \rangle$, that is to say, pull the value of $1 - \left|c_i - c_j\right|$ closer to 1 when $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ have the same representations, otherwise pull it closer to 0.

We propose to solve the following loss function of GAN regularized by our similarity constraint:

$$\min_G \max_D V(D, G) - \lambda SC(\boldsymbol{x}, \boldsymbol{c}) \tag{7}$$

where λ is the hyperparameter to tune.

E. MODEL ARCHITECTURE

InfoGAN attempts to separate $\boldsymbol{c}$ from synthetic images $G(\boldsymbol{z}, \boldsymbol{c})$ by introducing an extra network Q, while SCGAN only adds the similarity constraint on the generator. Therefore, the architecture of InfoGAN is more complicated than that of SCGAN. And for InfoGAN, more parameters of the networks are required to train. We show their architectures in Fig. 1.

V. EXPERIMENTS

We conduct experiments on five famous datasets. Details on the usage of each of these datasets are given below. Basically, we sampled discrete conditional variables from the multinomial distribution and continuous conditional variables from the uniform distribution. For all experiments, we use Adam [25] as optimizer and euclidean distance as similarity, sample mini-batches of size 32 and set λ to 1.[3] At last, we add experiments to verify assumptions in Section IV-A and show the stability of training models on the processed MNIST. In terms of time consumption, SCGAN is 3-5 times slower than InfoGAN on training, as it adds pair-wise loss.

A. MNIST AND FASHION-MNIST

MNIST [26] and Fashion-MNIST [27] are very clean datasets, the former contains a lot of handwritten digits and the latter contains various clothing. Both datasets include 10 classes which are likely to be captured by categorical conditional variables. We train CGAN, InfoGAN and SCGAN on MNIST and Fashion-MNIST dataset respectively and use

[1] Here $\boldsymbol{x}$ denotes synthetic data, not real data and $\boldsymbol{c}$ is encoded as one-hot vector.

[2] $\boldsymbol{x}$ also denotes synthetic data and c denotes a scalar variable drawn from a uniform distribution $Uinf(0, 1)$.

[3] It's very crucial to tune λ, we run an exhaustive grid search to choose it. If λ is too small, conditional variables can't capture disentangled representations. And if λ is too large, generator is not well converged(may generates the average image).

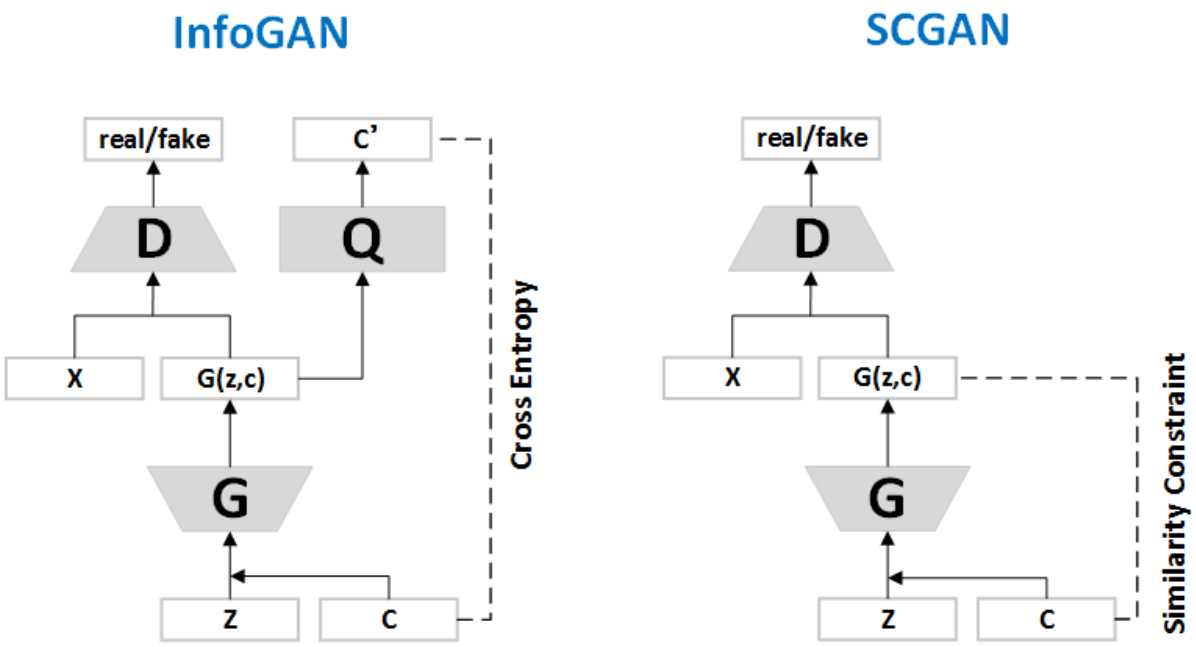


FIGURE 1. The architectures of InfoGAN and SCGAN. InfoGAN attempts to separate the condition which is encoded in the synthetic images (Left). SCGAN attempts to learn the mapping relationship of the condition and the synthetic images through the correlation between them (Right).

Gaussian Parzen window to estimate log-likelihood of each GAN.

1) IMPLEMENTION DETAILS

In the generator network, a 90-dimensional noise is drawn from a uniform distribution $Unif(-1, 1)$ and a 10-dimensional categorical conditional variable is drawn from a multinomial distribution, resulting in a concatenated dimension of 100. We use batch normalization [28] for all layers and Rectified Linear Unit(ReLU) [29] activation for all but the last layer. In the discriminator network, we replace ReLU with LeakyReLU [30] as activation, and batch normalization is not applied on the first and last layer. The network architectures of three kinds of GANs are similar except CGAN needs supervision to feed discriminator. For more details, see Appendix A-A.

We train all models 25 epoches and find categorical conditional variables capture class labels (e.g., digit type, clothing type). Synthetic images are shown in Fig. 2.

2) EVALUATATION

We fit Gaussian Parzen window to the samples generated by G of each GAN and estimate log-likelihood under distribution $\mathbb{P}_g$. More specifically, 10000 samples are drawn from $\mathbb{P}_g$ to fit Gaussian Parzen window and 1000 samples are drawn from validation set to estimate standard deviation σ of Gaussians by cross validation. Table. 1 shows Gaussian Parzen window log-likelihood estimate for three kinds of GANs on the MNIST and Fashion-MNIST test data. Experiments show that SCGAN has the largest log-likelihood on test data than other three GANs, which means the images generated by SCGAN is more realistic.

TABLE 1. Gaussian Parzen window-based log-likelihood estimates for MNIST and Fashion-MNIST. The reported numbers are mean log-likelihood of samples on test set, with the standard error of the mean computed across examples. The latter three use the same network structures, trained and tested in the same environment. The result of GAN on MNIST refers to original paper [3].

Model	MNIST	Fashion-MNIST
GAN [3]	225 ± 2	$-$
CGAN	228.1 ± 2.2	312.7 ± 2
InfoGAN	231 ± 2.2	320.2 ± 2
SCGAN	$\mathbf{233.6 \pm 2.2}$	$\mathbf{321.7 \pm 2}$

B. SVHN AND CIFAR10

The Street View House Number(SVHN) [31] dataset is more complicated than MNIST though they all contain digits(0-9). It seems that networks can't learn to generate digits conditioned on digit type(0-9), because the background of image and the color of digits are different, and many digits are much more blurred. Therefore, the most obvious difference between two images may be color rather than type of digits. In other words, it is hard for networks to distinguish digit type even if we use 10-dimensional categorical variables.

The CIFAR10 [32] dataset consists of 32×32 colour images in 10 classes. Nevertheless, it is too difficult to

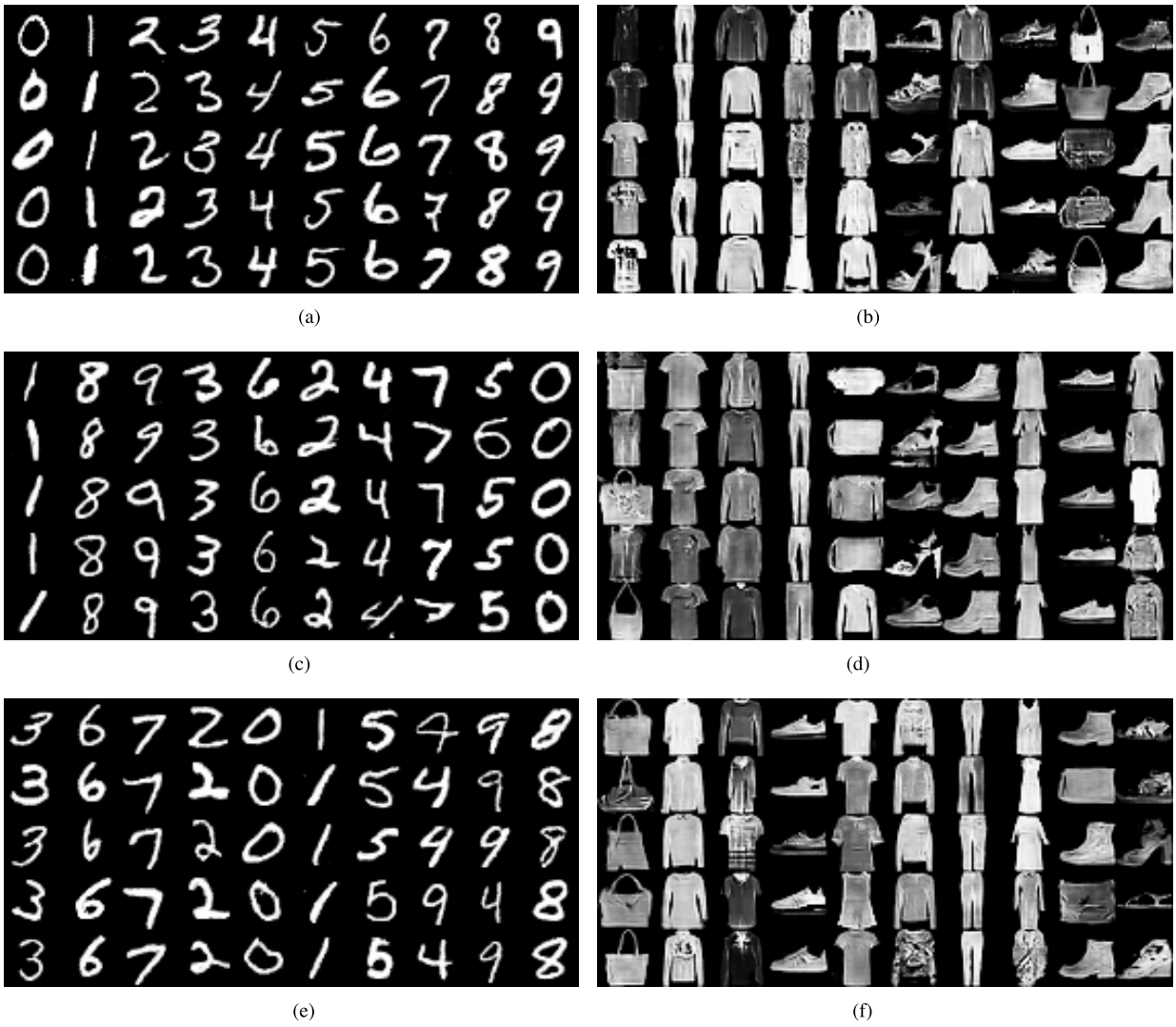


FIGURE 2. Generated samples on MNIST and Fashion-MNIST. Each column is conditioned on one mode of data. (a)(b) are generated by CGAN, which know ground truth of each mode. (c)(d) are generated by InfoGAN and (e)(f) are generated by SCGAN, they are unsupervised. On these dataset, InfoGAN and SCGAN both misclassify some modes, e.g., InfoGAN confuses digit 5 with digit 6 (9th column of (c)) and confuses T-shirt with bag (1st column of (d)), SCGAN confuses digit 4 with digit 9 (8th, 9th columns of (e)) and confuses bag with ankle boot (9th column of (f)). Note, there is no guarantee on the ordering of condition c for InfoGAN(c)(d) and SCGAN(e)(f), since they do not utilize class labels as supervision. (a) CGAN on MNIST. (b) CGAN on Fashion-MNIST. (c) InfoGAN on MNIST. (d) InfoGAN on Fashion-MNIST. (e) SCGAN on MNIST. (f) SCGAN on Fashion-MNIST.

generate realistic images subject to the distribution of CIFAR10 images, even if one uses the-state-of-art models, e.g. SNGAN [33]. Therefore, it is more challenging to train conditional GANs on CIFAR10 than many other datasets. The paper of InfoGAN doesn't show experimental results on CIFAR10. But in this paper, we give it a try.

1) IMPLEMENTION DETAILS

In order to generate more realistic images, we use Wasserstein GAN with gradient penalty(WGAN-GP) [21] instead of vanilla GAN. The principle of using batch normalization on generator and activation is the same as V-A, except the output activation of discriminator is *tahn* rather than *sigmoid*, because we normalize images to $[-1, 1]$. The discriminator can't use batch normalization, since WGAN-GP penalizes the norm of the discriminator's gradient with respect to each input independently, and not the entire batch. We train two scalar continuous variables(c_1, c_2, drawn from $Unif(0, 1)$ respectively) joining with a 126-dimensional noise on both datasets. More details about network architectures and hyperparameter setting are in Appendix A-B.

To observe what variations these continuous variables capture, we fix the noise z and one condition, then vary the other condition from 0 to 1. On SVHN, we find that c_1, c_2 capture the variation of lighting and digit types. They are shown in Fig. 3. On CIFAR10, we find continuous variables also capture some latent representations even though the quality of synthetic images is not so good due to bottleneck on generative capacity of WGAN-GP. We show variation of object size and colour in Fig. 4.

(a) (b)

FIGURE 3. Generated samples on SVHN. In (a), lighting becomes more intense from left to right on each row. In (b), a continuous variable controls the variation of digit types, for example in the 1st row, digit 6 is transformed into white background and blue background is transformed into digit 3, and in the 5th row, digit 4 is transformed into digit 1 by changing the shape gradually. (a) Lighting. (b) Digit types.

(a) (b)

FIGURE 4. Generated samples on CIFAR10. In (a), the size of objects becomes larger from left to right on each row. In (b), there is a subtle change on the colour of objects and background. (a) Object size. (b) Colour.

C. CELEBA

CelebA [34] is a large-scale face dataset with more than 200K celebrity images, which covers large pose variations and background clutter. There are many semantic variations (e.g., presence or absence of glasses, hair styles and emotion) that can be captured in the dataset.

1) IMPLEMENTION DETAILS

We train 4 categorical conditional variables(each of dimension 10) joining with a 216-dimensional noise on CelebA datasets. And we use cropped face images(reduce the interference of background) and resize them to 32×32. The network architectures are the same as V-B except the dimension of input doubles. More details about network architectures and hyperparameter setting, see Appendix A-C.

We train the model 300K steps and find categorical variables indeed capture many visual concepts like variations in pose, glasses, hair and sex, demonstrating a level of visual understanding is acquired without any supervision. They are shown in Fig. 5.

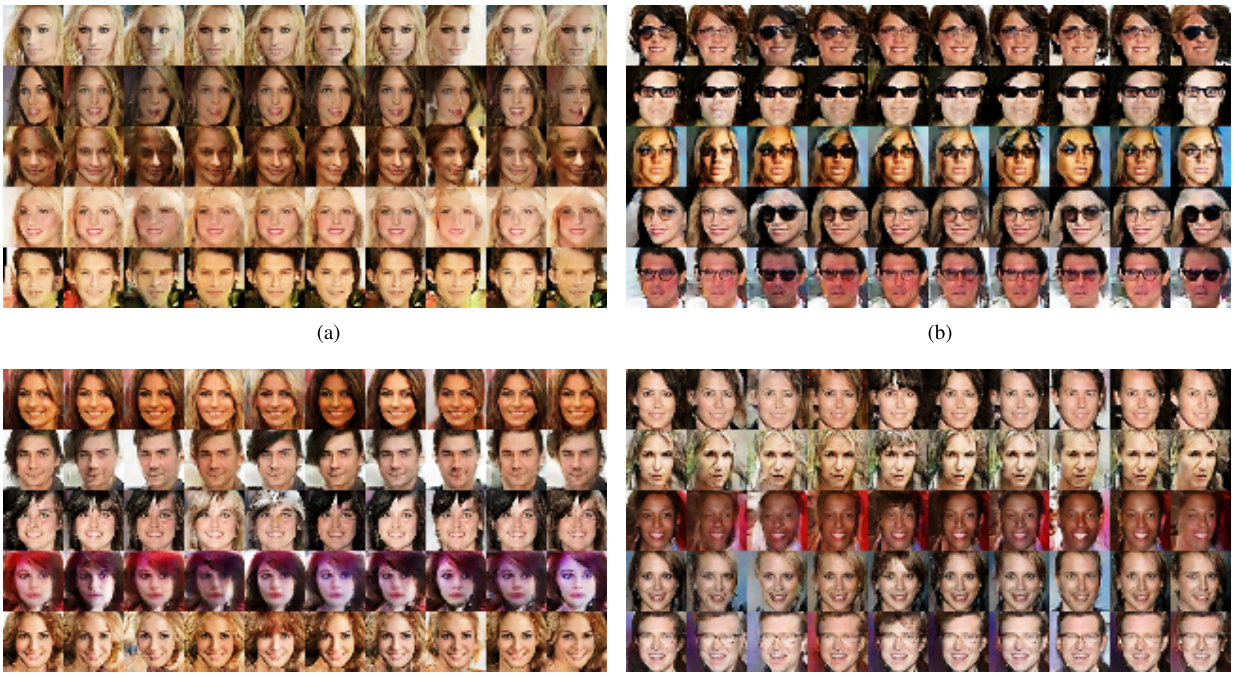
D. VERIFICATION EXPERIMENT

The assumption that z controls content and $\boldsymbol{c}$ controls representation is easy to verify. We fix z and randomly sample $\boldsymbol{c}$, then observe the difference of representation, or fix $\boldsymbol{c}$ and randomly sample z, then observe the difference of content. We conduct the experiment on CelebA and in this scenario, content represents whether the face is from the identical celeb or not. The results are shown in Fig. 6.

In order to verify assumption that *DR* is equivalent to *Sim* while fixing *DC* (4), we randomly sample a large number of z on both cases: the same or different $\boldsymbol{c}$, and then compare *mSim*(the mean of *Sim*) of synthetic image-pairs between the two different cases. The result shows that *mSim*(13.6) with the same $\boldsymbol{c}$ is much less than *mSim*(17.8) with the different $\boldsymbol{c}$ on CIFAR10. As z is randomly sampled on both cases, there is no great difference on *DC* of two cases statistically(averagely), so the reduction(17.8-13.6) in *Sim* is due to *DR*.

TABLE 2. The discriminator and generator CNNs used for MNIST and Fashion-MNIST dataset.

discriminator D	**generator** G
Input 28×28 gray images	Input $\in \mathbb{R}^{100}$
conv(64)-4c2s-LeakyReLU	fc(1024)-BN-ReLU
conv(128)-4c2s-BN-LeakyReLU	fc($128 \times 7 \times 7$)-BN-ReLU
fc(1024)-BN-LeakyReLU	deconv(64)-4c2s-BN-ReLU
fc(1)	deconv(1)-4c2s-BN-Sigmoid



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

FIGURE 5. Generated samples on CelebA. (a) shows variation in pose, e.g., frontal face and profile face. (b) shows variation in glasses, e.g., sunglasses fade into glasses. (c) shows variation in hair, e.g., the color and style of hair change. (d) shows variation in sex, e.g., each row includes faces of women and men. Since categorical conditions are discrete, there is no guarantee on the ordering of images on each row. (a) Pose. (b) Glasses. (c) Hair. (d) Sex.

(a) (b)

FIGURE 6. (a) is sampled from different z while fixing c, all the faces are from different celebs. As a contrast, (b) is sampled from different c while fixing z, all the faces are from an identical celeb except different representations, e.g., pose, hair.

E. STABILITY EXPERIMENT

We think representations can be disentangled well depending on datasets as images of these datasets are highly center aligned. If not aligned, infogan, cgan and scgan may all fail. In order to verify our point of view. we randomly shift digits and make them not center aligned on MNIST. And we find that cgan and infogan are not stable and diverge at some steps while training, but scgan is stable. See Fig. 7. We suspect the reason for failure is that *DC* is so large that it influences to learn *DR*.

VI. CONCLUSIONS

In this work, we introduce a new model called SCGAN to learn interpretable representations unsupervisedly. Comparing with InfoGAN, SCGAN is based on intuition more and easier to implement. And it's more likely to combine with all kinds of variations of GAN (e.g., WGAN, WGAN-GP, SNGAN), as it doesn't introduce another network to approximate the lower bound of mutual information. In addition, SCGAN can learn effective representations as InfoGAN. Particularly, SCGAN is likely to be improved since our similarity can be more forms, not just euclidean distance in our experiments. Perhaps more efficient similarity can drive SCGAN to learn more abundant representations and we leave it for the future work. And other possible work in the future is to apply SCGAN on semi-supervised learning.

APPENDIX A
EXPERIMENT SETUP

For MNIST and Fashion-MNIST, we use vanilla GAN architectures, and for SVHN, CIFAR10 and CelebA, we use Wasserstein GAN architectures. For all experiments, we use Adam as optimizer and apply batch normalization

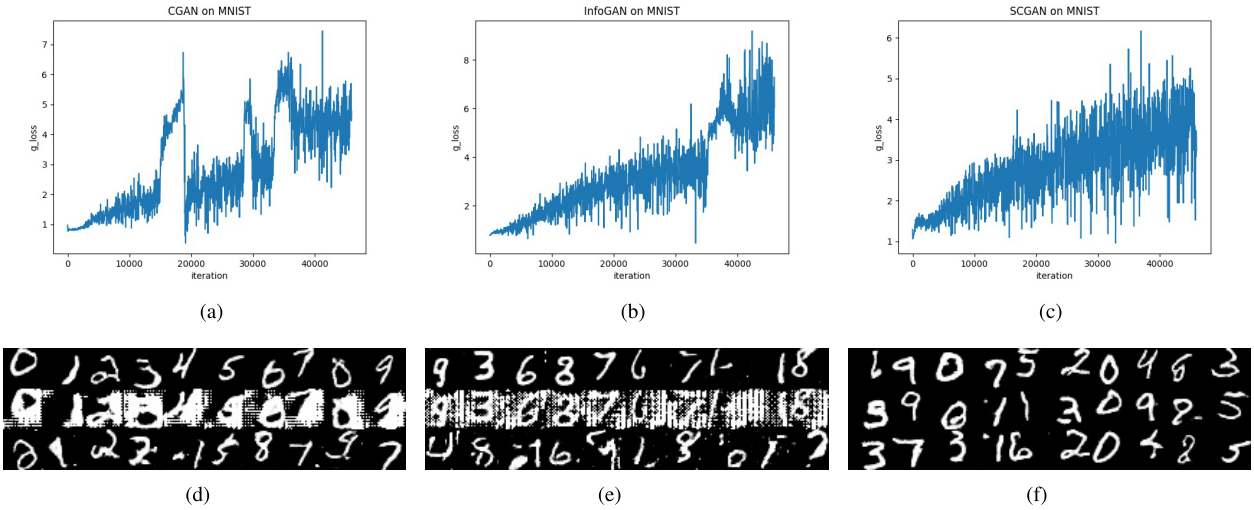


FIGURE 7. Loss in (a)(b) changes suddenly at some steps, and synthetic images collapse, 2nd row in (d)(e). Each row of (d)(e)(f) is acquired from different training steps. 3rd row of (e) shows InfoGAN fails at the end of training. 1st row of (d) shows CGAN is able to learn digit types at the start of training due to supervised learning. (f) shows SCGAN has a little difficulty learning digit types due to unsupervised learning. (a) g_loss of CGAN. (b) g_loss of InfoGAN. (c) g_loss of SCGAN. (d) CGAN on MNIST. (e) InfoGAN on MNIST. (f) SCGAN on MNIST.

TABLE 3. The hyperparameters for MNIST and Fashion-MNIST dataset.

discriminator D	**generator** G
Learning rate for D	2e-4
Learning rate for G	1e-3
Learning rate for Q	1e-3
Iterations for D every step	1
β_1 for Adam	0.5
β_2 for Adam	0.999
λ for SC	1

TABLE 4. The discriminator and generator CNNs used for SVHN and CIFAR10 dataset.

discriminator D	**generator** G
Input 32×32 colour images	Input $\in \mathbb{R}^{128}$
conv(64)-5c2s-LeakyReLU	fc(4096)-BN-ReLU
conv(128)-5c2s-LeakyReLU	deconv(128)-5c2s-BN-ReLU
conv(256)-5c2s-LeakyReLU	deconv(64)-5c2s-BN-ReLU
fc(1)	deconv(3)-5c2s-BN-Tanh

after most layers. We use Leaky ReLU with leaky rate 0.2 for discrminator networks and ReLU for generator networks.

A. MNIST AND FASHION-MNIST

The network architectures are shown in Table. 2. InfoGAN has an extra Q network to simulate the variational lower bound, the architecture of Q is the same as D except the output of the last fully connected layer. In these experiments, the output of Q network is 10 dimension, for we use 10-dimensional categorical variables. And the hyperparameter setting is shown in Table. 3.

TABLE 5. The hyperparameters for SVHN and CIFAR10 dataset.

discriminator D	**generator** G
Learning rate for D	1e-4
Learning rate for G	1e-4
Iterations for D every step	5
β_1 for Adam	0.5
β_2 for Adam	0.9
λ for SC	1
λ for GP	10

TABLE 6. The discriminator and generator CNNs used for CelebA dataset.

discriminator D	**generator** G
Input 32×32 colour images	Input $\in \mathbb{R}^{256}$
conv(64)-5c2s-LeakyReLU	fc(4096)-BN-ReLU
conv(128)-5c2s-LeakyReLU	deconv(128)-5c2s-BN-ReLU
conv(256)-5c2s-LeakyReLU	deconv(64)-5c2s-BN-ReLU
fc(1)	deconv(3)-5c2s-BN-Tanh

B. SVHN AND CIFAR10

The network architectures are shown in Table. 4. We use more complex architectures, in order to get better quality images in these experiments. And the hyperparameter setting is shown in Table. 5.

C. CELEBA

The network architectures are shown in Table. 6. We just use the same network architectures as shown in Table. 4. The only

TABLE 7. The hyperparameters for CelebA dataset.

discriminator D	**generator** G
Learning rate for D	1e-4
Learning rate for G	1e-4
Iterations for D every step	5
β_1 for Adam	0.5
β_2 for Adam	0.9
λ for SC	1
λ for GP	10

difference is that the input dimension of G doubles, since CelebA dataset has more images. And the hyperparameter setting is shown in Table. 7.

REFERENCES

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[2] D. P. Kingma and M. Welling. (2013). "Auto-encoding variational bayes." [Online]. Available: https://arxiv.org/abs/1312.6114

[3] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[4] M. Mirza and S. Osindero. (2014). "Conditional generative adversarial nets." [Online]. Available: https://arxiv.org/abs/1411.1784

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[6] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.

[7] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, p. 607, 1996.

[8] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[9] A. Radford, L. Metz, and S. Chintala. (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1511.06434

[10] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural Comput.*, vol. 12, no. 6, pp. 1247–1283, Jun. 2000.

[11] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: A deep model for learning face identity and view representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 217–225.

[12] A. Dosovitskiy, J. T. Springenberg, and T. Brox, "Learning to generate chairs with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1538–1546.

[13] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 3483–3491.

[14] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[15] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. (2015). "Adversarial autoencoders." [Online]. Available: https://arxiv.org/abs/1511.05644

[16] S. Reed, K. Sohn, Y. Zhang, and H. Lee, "Learning to disentangle factors of variation with manifold interaction," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. II-1431–II-1439.

[17] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2539–2547.

[18] G. Desjardins, A. Courville, and Y. Bengio. (2012). "Disentangling factors of variation via generative entangling." [Online]. Available: https://arxiv.org/abs/1210.5474

[19] J. Susskind, A. Anderson, and G. E. Hinton, "The toronto face dataset," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. UTML TR 1, 2010.

[20] M. Arjovsky, S. Chintala, and L. Bottou. (2017). "Wasserstein GAN." [Online]. Available: https://arxiv.org/abs/1701.07875

[21] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[22] J. Zhao, M. Mathieu, and Y. LeCun. (2016). "Energy-based generative adversarial network." [Online]. Available: https://arxiv.org/abs/1609.03126

[23] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.

[24] D. Barber and F. Agakov, "The IM algorithm: A variational approach to Information Maximization," in *Proc. 16th Int. Conf. Neural Inf. Process. Syst.* Cambridge, MA, USA: MIT Press, 2003, pp. 201–208.

[25] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[27] H. Xiao, K. Rasul, and R. Vollgraf. (2017). "Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms." [Online]. Available: https://arxiv.org/abs/1708.07747

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.

[30] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, Jun. 2013, vol. 30, no. 1, p. 3.

[31] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *Proc. NIPS Workshop Deep Learn. Unsupervised Feature Learn.*, 2011, pp. 1–9.

[32] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[33] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. (2018). "Spectral normalization for generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1802.05957

[34] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 3730–3738.

XIAOQIANG LI (M'14) received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2004.

He is currently an Associate Professor of computer science with Shanghai University, Shanghai. He has published over 60 technical articles in refereed journals and proceedings, including the IEEE TRANSACTIONS ON CYBERNETICS and ACCV 2016. His current research interests include image processing, pattern recognition, computer vision, and machine learning. He is currently an ACM Member, a Senior Member of the Chinese Computer Society, and the Deputy Director of the Multimedia Special Committee of the Shanghai Computer Society.

LIANGBO CHEN received the B.S. degree in computer science from Shanghai University in 2016, where he is currently pursuing the M.S. degree.

He is also a member of the Machine Vision Laboratory, School of Computer Engineering and Science. His research interests include computer vision, image processing, and machine learning.

LU WANG received the B.S. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 2002, and the Ph.D. degree in automatic control from Tsinghua University, Beijing, China, in 2008.

He is currently an Assistant Professor of computer science with Shanghai University, Shanghai, China. His research interests include computer vision, image processing, and machine learning.

PIN WU received the B.S. and Ph.D. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 1998 and 2003, respectively. She held a post-doctoral position with Zhejiang University from 2003 to 2005 and was a Senior Visiting Scholar with Michigan State University from 2012 to 2013.

She is currently an Associate Professor of computer science with Shanghai University, Shanghai, China. Her current research interests include high-performance computing, computational fluid dynamics, and image processing. Over the past 10 years, she has published over 30 technical papers in the related fields. She will keep on the research work addressing cross disciplinary of computer and mechanics.

WEIQIN TONG received the B.S. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 1984, and the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 1995.

He is currently a Professor of computer science with Shanghai University, Shanghai. His research interests include parallel computing, cloud computing and big data, and machine learning.