Sparse Generative Adversarial Network

Shahin Mahdizadehaghdam
Department of Electrical
and Computer Engineering
North Carolina State University
Raleigh, NC 27695
smahdiz@ncsu.edu

Ashkan Panahi
Department of Electrical
and Computer Engineering
North Carolina State University
Raleigh, NC 27695
apanahi@ncsu.edu

Hamid Krim
Department of Electrical
and Computer Engineering
North Carolina State University
Raleigh, NC 27695
ahk@ncsu.edu

Abstract

We propose a new approach to Generative Adversarial Networks (GANs) to achieve an improved performance with additional robustness to its so-called and well recognized mode collapse. We first proceed by mapping the desired data onto a frame-based space for a sparse representation to lift any limitation of small support features prior to learning the structure. To that end we start by dividing an image into multiple patches and modifying the role of the generative network from producing an entire image, at once, to creating a sparse representation vector for each image patch. We synthesize an entire image by multiplying generated sparse representations to a pre-trained dictionary and assembling the resulting patches. This approach restricts the output of the generator to a particular structure, obtained by imposing a Union of Subspaces (UoS) model to the original training data, leading to more realistic images, while maintaining a desired diversity. To further regularize GANs in generating high-quality images and to avoid the notorious mode-collapse problem, we introduce a third player in GANs, called reconstructor. This player utilizes an auto-encoding scheme to ensure that first, the input-output relation in the generator is injective and second each real image corresponds to some input noise. We present a number of experiments, where the proposed algorithm shows a remarkably higher inception score compared to the equivalent conventional GANs.

1. Introduction [1]

In recent years, Generative Adversarial Networks (GANs) [25, 15, 33] showed impressive results in various image generation problems, such as image super-resolution [16, 5, 27], dialogue generation [17], and image translation [36, 30, 18, 14]. As an implicit method of probability density estimation, GANs commonly consist of two main units: A generator function that synthesizes various images from different realizations of an input noise vector, and a discriminator function which examines the quality of the produced images by the generator. The generator and discriminator are usually differentiable-functions based on deep convolutional networks. In GANs, the input noise vector is transformed by the generator to produce a "fake" image. The discriminator function receives either a fake sample from the generator or an "original" sample from the data set and decides on its genuineness. The two functions are successively trained in a min-max optimization framework [9], where the weights of the generator network are gradually adapted to generate more realistic images, in order to fool the discriminator.

The research area of GANs still takes its first steps toward full growth. Due to the complex structure of real-world images, the images generated by GANs widely lack details and do not usually look completely realistic. Images are known to follow complicated probability distributions, and estimating these distributions at the full image scale is generally a difficult task. Mode collapse, a well-known phenomenon in

[1]This work was supported by the U.S. Army Research Office under Grant W911NF-18-10209.

GANs, where highly similar images are frequently generated from different inputs is widely attributed to the problem of complexity of the image distributions (small support events). To alleviate the above-mentioned issues, we propose a novel generator model which learns the structure of images in smaller scales and uses the learned structures to synthesize full-scale images. To this end, we adopt dictionary learning and sparse representation [21], as an effective method to learn and leverage the structure of data. Parsimonious data representation by learning overcomplete dictionaries has shown promising results in a variety of problems such as image denoising [8, 12], image restoration [29], audio processing [10], and image classification [32]. The learned dictionaries constitute a frame, whose atoms are employed to linearly represent data vectors with their corresponding sparse vectors of coefficients [28]. (This effectively lifts the singular support [24]).

We hence propose a novel generator network which generates a sparse vector of coefficients for each image patch instead of generating the entire image. Full-size output images are assembled by tiling the image patches, produced by multiplying the generated sparse coefficient vectors to a pre-trained dictionary. This approach avoids the complexities with the conventional method of generating the entire image in one step. Instead, it affords a multi-stage solution, where at the first stage, simpler structures at the scale of image patches are generated, and a full-size image is assembled at the final stage. Incorporating sparse representations also limits the search space of the produced images to a linear combination of a set of dictionary atoms, tailored for representing real-world images, enhancing the procedure of generating realistic images. To further regularize with respect to the mode collapse problem, we also introduce a third player to GANs that we call the reconstructor. The reconstructor has two goals. First, it makes sure that different inputs of the generator result in dissimilar outputs i.e., the generator is an injective map. Second, it guarantees that the real images in the dataset can also be synthesized from some input noise i.e., the range space of the generator includes the entire set of real images. These goals are obtained by treating the generator network as a ”decoder” from a latent input space and simultaneously training an ”encoder” network, which reversely computes the input of the generator from its output. The encoder function as a regularizer to the generator network is minimized successively along with the generator and the discriminator losses. Existence of such an auto-encoder scheme guarantees that the generator network is injective and model collapse is avoided. Moreover by training this auto-encoder scheme on real images, we achieve the second goal.

The balance of this paper is organized as follows: In Section 2, we provide an overview to the state of-the-art works relevant to this paper. In Section 3, we briefly recall the mathematical basis of GANs, as well as some background information of relevance to this paper. We formulate and propose our new approach in Section 4. Substantiating experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2. Related Studies

There have been plenty of different studies on GANs since their advent in [9]. We summarize these works into three main groups and briefly overview them.

Stable training of GANs: Due to the non-convex nature of the underlying optimization problems in GANs, they are notoriously difficult to train. In particular, they are generally prone to non-convergence, diminishing gradient and mode collapse, the latter happening when the generator frequently outputs a narrow set of highly similar samples. Several efforts were devoted to alleviating these problems. An autoencoder-based regularization was proposed in [3] by penalizing missing modes. To further mitigate the mode collapse problem, an unrolled optimization of the discriminator is proposed in [22]. This technique results in a more power-balanced generator and discriminator and is successful in preventing mode collapse, but the computational cost is high. In WGAN [1], the Earth-Mover (Wasserstein-1) distance is adopted as the objective for the generator. This objective is approximated by restricting the discriminator to 1-Lipschitz function through weight clipping. An enhancement over WGAN is proposed in [11]. In this work, a different method for bounding the gradients is proposed by adding a gradient penalty term to the objective. Furthermore, in a recent effort [23], the Lipschitz constant of the discriminator function is controlled by limiting the spectral norm of the weights in the discriminator.

Architectures of GANs: The Deep Convolutional Generative Adversarial Network, DCGAN, architecture is proposed in [25]. This architecture is usually a set of (four) fractionally-strided convolutional layers with no pooling and fully connected layers. Stacked Generative Adversarial Networks, StackGAN [34], aims to generate 256×256 realistic images from text descriptions in two steps. Initially, lower resolution images are generated from text descriptions, and then, higher-resolution images are generated from low-resolution images. In [15], the authors proposed to progressively develop the generator and discriminator as training continues. This method helps to speed up the training and improve image quality.

Applications of GANs: GANs have been utilized in different areas of machine learning, including Natural language processing (NLP) and computer vision. In [35] an LSTM network and a CNN have been trained in an adversarial way to generate realistic text. The LSTM network takes the random input vector and generates text while the CNN discriminates between real text and generated one. An unsupervised word translation method has been proposed in [4], which translates words without any cross-lingual supervision. In this work, a generator network has been used to map word embeddings from one domain to another, while the discriminator aims to detect the origin of the embedding. A variation of GANs with class label information, known as Conditional GANs, have been utilized in image-to-image translation [14] and demonstrated successful performance in reconstructing objects from edge maps and colorizing tasks. The generator network is used in [16, 26] to produce super-resolution images from low resolution images. The discriminator networks in these works are trained to discriminate between super-resolved images and real high-resolution images. GANs are also used in transforming person images to arbitrary poses and synthesizing clothing images and styles from an image [31, 20].

From architectural point of view, our proposed approach is similar to DCGAN [25] with an additional provision for a layer to generate sparse coefficients. We use the WGAN technique in [11] to stabilize the training of GANs. Furthermore, we propose a third player to GANs (reconstructor) which assists the generator network in regularizing and generating more realistic images. The reconstructor network in GANs, to the best of our knowledge, is a novel idea without precedent.

3. Mathematical Basis of GANs

The initial formulation of GANs in [9] is as follows,

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

where $\boldsymbol{x}$ and $\boldsymbol{z}$ are the real data and the noise input vectors respectively and D and G are the discriminator and the generator networks. $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{z}}$ are the distributions of the real data and noise vector respectively.

The above optimization problem is solved by an alternating optimization scheme. At each step, the generator or the discriminator variables are fixed and the problem is solved for the other player. Defining $p_{\boldsymbol{g}}$ as the distribution of the generated images ($G(\boldsymbol{z}) \sim p_{\boldsymbol{g}}$), Eqn. (1) is equivalent to minimizing the Jensen-Shannon divergence between $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{g}}$ [9].

The Jensen-Shannon divergence between $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{g}}$ is not always continuous with respect to the variables in the generator [1] and this leads to an instability in training GANs. Therefore, in [1], the Jensen-Shannon divergence is replaced by the earth-mover's distance $W(p_{\boldsymbol{x}}, p_{\boldsymbol{g}})$ which is continuous everywhere and leads to the following optimization problem,

$$\min_G \max_{D\in\mathcal{D}} \ \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))], \tag{2}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions. Considering an optimal discriminator function, the solution of the above optimization problem minimizes the earth-mover's distance between $p_{\boldsymbol{x}}$ and $p_{\boldsymbol{g}}$.

In order to impose the Lipschitz constraint on the discriminator ($D \in \mathcal{D}$), in [1], the weights of the discriminator are simply clipped within a compact interval $[c, c]$. Alternatively, the following gradient penalty term is minimized in [11] to impose the Lipschitz constraint.

$$\mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}[(||\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})||_2 - 1)^2], \tag{3}$$

where $\hat{\boldsymbol{x}} \sim p_{\hat{\boldsymbol{x}}}$ is a random sample, uniformly randomly chosen on the straight line segment connecting $\boldsymbol{x}$ and $G(\boldsymbol{z})$. The gradient penalty in Eqn. (3) is motivated from the fact that a function is 1-Lipschtiz if and only if its gradient norm is upper bounded by 1. Authors in [11] showed that the gradient penalty in Eqn. (3) is more effective in stabilizing training of GANs than the weight clipping technique.

4. The Proposed Method

In contrast to Eqn. (1), our proposed formulation of GAN is composed of three players: a discriminator function which decides whether an input is a real or a generated sample, a generator which synthesizes images by initially generating sparse representations of image patches, and a reconstructor which ensures that the generator is capable of generating all the real samples from some associated noise signals. We formulate our proposed method by defining the following optimization problem,

$$\begin{aligned} &\min_G \max_{D\in\mathcal{D}} \ \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))] \\ &\qquad\qquad + \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[||\boldsymbol{x} - G(E^*(\boldsymbol{x}))||_2^2] \\ \text{s.t}: \quad & E^* \in \arg\min_E \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[||\boldsymbol{z} - E(G(\boldsymbol{z}))||_2^2], \end{aligned}$$

where D, G and E are the discriminator, the sparse generator and the encoder networks, respectively. Moreover, E^* is an optimal encoder network for G, which serves as a left inverse of G, according to the constraint. The discriminator belongs to the set $\mathcal{D}$ of 1-Lipschitz functions and we explain the details of the Sparse Generator Network (SPGAN) in Section 4.1. $\boldsymbol{x}$ and $\boldsymbol{z}$ are the real data and the noise input vectors, respectively sampled from the real data distribution $p_{\boldsymbol{x}}$ and a fixed noise distribution $p_{\boldsymbol{z}}$. The last term in Eqn. (4), minimizes the error between the input real image $\boldsymbol{x}$ and the generated image by G from the corresponding noise $z_{\boldsymbol{x}} = E^*(\boldsymbol{x})$ to the data point $\boldsymbol{x}$. This term will be more thoroughly discussed in Section 4.2.

In order to solve the above optimization problem, we define the following three loss functions,

$$\begin{aligned} L_D &= \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[D(\boldsymbol{x})] + \\ &\quad \lambda \mathbb{E}_{\hat{\boldsymbol{x}}\sim p_{\hat{\boldsymbol{x}}}}[(||\nabla_{\hat{\boldsymbol{x}}} D(\hat{\boldsymbol{x}})||_2 - 1)^2], \\ L_G &= -\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D(G(\boldsymbol{z}))], \\ L_R &= \mathbb{E}_{\boldsymbol{x}\sim p_{\boldsymbol{x}}}[||\boldsymbol{x} - G(E^*(\boldsymbol{x}))||_2^2], \end{aligned} \tag{4}$$

where L_D, L_G and L_R are respectively the loss functions for the discriminator, the generator and the reconstructor. We explain the details of the reconstructor loss L_R in Section 4.2. The formulation of L_D is similar to the one in [11] with the last term being the gradient penalty to enforce the Lipschitz constraint. $\hat{\boldsymbol{x}} \sim p_{\hat{\boldsymbol{x}}}$ is a random sample, randomly uniform chosen on the straight line segment connecting $\boldsymbol{x}$ and $G(\boldsymbol{z})$. In the following, we explain the sparse generator network and the reconstructor loss in detail.

4.1. Sparse Generator Network

As we elaborated in Section 1, our generator network consists of a deep neural network that generates a vector of sparse coefficients $S(\boldsymbol{z})$ from a random input noise vector $\boldsymbol{z}$ as an input. These coefficients are used as a representation of the image patches in a linear model. Hence, the image patches can be simply computed by multiplying the coefficients to a pre-trained dictionary $\boldsymbol{\Omega}$. Later, the generated patches are connected to form a full-size image. One can uniformly extract image patches from the training images. These image patches are generally in small sizes (3×3) and may overlap with each other. Given vectorized image patches, $\boldsymbol{g}_i \in R^m$ $i \in \{1, .., s\}$, as columns of a matrix $\boldsymbol{G}$, the dictionary $\boldsymbol{\Omega}^* \in R^{m \times k}$ is trained by minimizing the following reconstruction loss,

$$\{\boldsymbol{\Omega}^*,\ \boldsymbol{R}\} = \underset{\boldsymbol{\Omega},\ \boldsymbol{R}}{\arg\min}\ \frac{1}{2}||\boldsymbol{G} - \boldsymbol{\Omega}\boldsymbol{R}||_F^2 + \lambda||\boldsymbol{R}||_1, \boldsymbol{\Omega} \in \mathcal{C}, \tag{5}$$

where the columns $\boldsymbol{r}_i \in R^k$ of the matrix $\boldsymbol{R}$ are the sparse representations of the image patches and $\mathcal{C}$ is a convex set of matrices with unit L_2-norm columns.

In Fig. 1, we show the sequence of computational steps of generating an image from a random input noise vector and their associated specifications. The random noise vector $\boldsymbol{z}$ is transformed to the tensor $A(\boldsymbol{z})$ of shape $w \times h \times k$. This transformation is by reshaping the input noise vector and applying transposed convolutions [25]. The soft thresholding function $S_\lambda(.)$ [6] is applied to each depth-vector $\boldsymbol{a}^T$ to create a sparse vector. The sparse vector $S_\lambda(\boldsymbol{a}^T)$ multiplicatively scales a pre-trained dictionary $\boldsymbol{\Omega}^*$ to generate an image patch $\boldsymbol{g}$. The image patches are aligned next to each other (in overlapping areas we average the pixel values) to produce the full image $G(\boldsymbol{z})$.

In our method, the image patches are produced by generating the sparse vectors and multiplying them to the dictionary $\boldsymbol{\Omega}^*$. The generated image patches are therefore, limited to a linear combination of a small subset of the dictionary atoms. This constraint helps to ensure that besides generating new samples, the generator network can generate images that resemble the training samples. Note that the image patches have smaller sizes and are less complicated in structure.

4.2. Reconstructor Loss

In this section, we explain the details of the reconstructor loss L_R in Eqn. (4). Considering the noise vectors as samples from a latent space, $G(\boldsymbol{z})$ can be interpreted as a map (decoder) from the latent space to the space of generated images (blue arrows in Fig. 2). The role of the generator network is to produce an image for any sample from the latent space. We also ensure that the map $G(\boldsymbol{z})$ can generate a wide variety of real-world images by verifying that it is injective and hence mode-collapse is hard. To this end, we require an "encoder" function, E, to exist such that $E(G(\boldsymbol{z})) = \boldsymbol{z}$ approximately holds true for every $\boldsymbol{z}$. Figure 2 explains why this requirement guarantees injectivity: If two realizations $\boldsymbol{z}_1, \boldsymbol{z}_2$ are mapped to the same image $\boldsymbol{x}$, then it is impossible for E to map $\boldsymbol{x}$ back to the latent space. We obtain the encoder E by employing a neural network E_ϕ and performing the optimization in the constraint of Eq. (4) by solving the following optimization problem with respect to the parameters of the encoder network, $\boldsymbol{\phi}$,

$$\boldsymbol{\phi}^* = \underset{\boldsymbol{\phi}}{\arg\min}\ \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}}||E_{\boldsymbol{\phi}}(G(\boldsymbol{z})) - \boldsymbol{z}||_2. \tag{6}$$

Algorithm 1

```
Initialization:
 1: Initialize the discriminator, the generator and the encoder parameters ($w$, $\theta$, and $\phi$) randomly.
 2: Train the dictionary $\boldsymbol{\Omega}$ by solving Eqn. (5)
Training:
 3: while $\theta$ has not converged do:
 4:   for $t$ in $\{1, n_{discr.}\}$ do:
 5:     for $i$ in $\{1, b\}$ do:
 6:       Sample $\boldsymbol{x}_i$, $\boldsymbol{z}_i$, and $\boldsymbol{\epsilon}_i$ from $p_{\boldsymbol{x}}$, $p_{\boldsymbol{z}}$, and the uniform distr. $U[0,1]$ respectively,
 7:       $\hat{\boldsymbol{x}}_i \leftarrow \boldsymbol{\epsilon}_i \boldsymbol{x}_i + (1 - \boldsymbol{\epsilon}_i) G_\theta(\boldsymbol{z}_i)$
 8:       $L_D^i \leftarrow D_w(G_\theta(\boldsymbol{z}_i)) - D_w(\boldsymbol{x}_i) + \lambda(||\nabla_{\hat{\boldsymbol{x}}} D_w(\hat{\boldsymbol{x}}_i)||_2 - 1)^2$,
 9:       $w \leftarrow w - \nabla_w(\frac{1}{b}\sum_{i=1}^b L_D^i)$,
10:   for $i$ in $\{1, b\}$ do:
11:     Sample latent variable $\boldsymbol{z}_i$, from $p_{\boldsymbol{z}}$,
12:     $L_G^i \leftarrow -D_w(G_\theta(\boldsymbol{z}_i))$,
13:     $\theta \leftarrow \theta - \nabla_\theta(\frac{1}{b}\sum_{i=1}^b L_G^i)$,
14:   for $t$ in $\{1, n_{reconst.}\}$ do:
15:     Train the encoder network Eqn. (6),
16:     for $i$ in $\{1, b\}$ do:
17:       Sample real data $\boldsymbol{x}_i$ from $p_{\boldsymbol{x}}$. Encode the real image to latent variable $\boldsymbol{z}_{\boldsymbol{x}_i} = E^*(\boldsymbol{x_i})$ ,
18:       $L_R^i \leftarrow ||\boldsymbol{x}_i - G(\boldsymbol{z}_{\boldsymbol{x}_i})||_2$,
19:       $\theta \leftarrow \theta - \nabla_\theta(\frac{1}{b}\sum_{i=1}^b L_R^i)$,
```

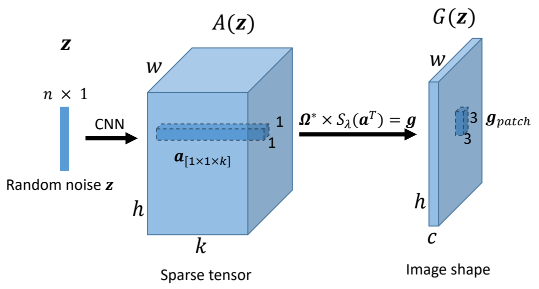


Figure 1: Sequential steps of a sparse generator network.

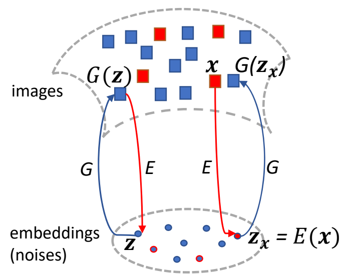


Figure 2: Generating and encoding images. Blue/red squares and circles represent generated/real images and their associated latent vectors respectively.

In our experiments, a ResNet [13] architecture with 50 layers is used to approximate E. In conclusion, the reconstructor, as a third player in GANs, guides the generator network to avoid mode-collapse and generate a wide variety of realistic images.

4.3. Algorithm

The optimization problem in Eqn. (4) is not jointly convex in variables of G and D. Therefore, we adopt an alternating stochastic gradient descent approach to minimize the loss functions in Eqn. (4), and a block coordinate descent strategy for satisfying its constraint (updating E^*). This procedure leads to Algorithm 1, where the details of the updating procedure are discussed below:

Lines 1-2: We randomly initialize the generator and the discriminator parameters. The dictionary $\mathbf{\Omega}$ is trained using the online dictionary learning approach [21].

Lines 4-9: We randomly select a subset of images and noise vectors. We uniformly sample a real number ϵ. The random noise vector z_i is fed to the generator network and we calculate the discriminator loss in (line 8) and update the discriminator parameters (line 9).

Lines 10-13: We randomly select noise vectors and calculate the generator loss in (line 11). We update the generator parameters in (line 13).

Lines 14-19: We train the encoder network (line 15) by generating samples and solving Eqn. (6) through a standard back propagation scheme. We skip some details due to lack of space. Then, we randomly select noise vectors associated with the real images, and we calculate the reconstructor loss in (line 18). We update the generator parameters in (line 19).

5. Experiments

We evaluate our proposed methodology on CelebA dataset [19], and CIFAR10 object images. The performance of our proposed method is compared to the state-of-the-art methods. In all the following experiments, the reconstructor network E is a 50-layer ResNet [13] network trained on 50,000 randomly generated samples.

5.1. CIFAR 10 dataset

There are 60,000 color images in CIFAR-10 dataset which are divided into 50,000 training images and 10,000 test images. The size of images in this dataset is 32×32 and there are in total 10 classes in this dataset. In order to generate a 32×32 image from the 128 dimensional noise vector, a fully connected layer first expands the input dimension to a 4×4×1024 tensor (reshaped as tensor) and subsequently a series of three fractionally-strided convolutions transform the spatial dimension to 32×32 (doubling the spatial dimension in each layer). The number of channels has changed from 1024 to 512 and 100. The size of the pre-trained dictionary, Ω, in this experiment is 27×100.

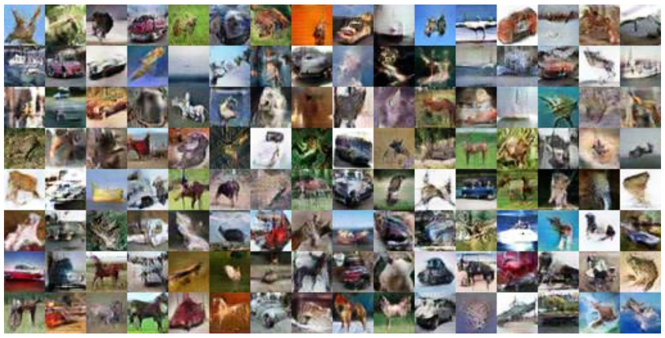
Fig. 3 presents the images generated in this experiment and compares them with the images generated by WGAN [1] and Improved WGAN [11]. As seen, the images generated by the proposed method have higher quality and variance compared to the images generated by the other methods. It is worth mentioning that the experiments in this section are conducted in an unsupervised setting. Fig. 4 presents the images generated in this experiment using residual blocks in each layer of the generator network and compares them with the images generated by Improved WGAN [11]. We observe that our method leads to a higher quality and diversity.

Table 1 and Table 2 compare the inception score of different methods in case of a simple DCGAN generator and a generator with residual blocks respectively. Our proposed method increases the inception of WGAN and Improved WGAN in both cases. In case of a simple DCGAN generator, the inception score increased of Improved WGAN increased dramatically by 0.78 point and in case of a generator with residual blocks, the score increased by 0.09 point.

5.2. CelebA dataset

There are 200,000 color images of celebrity faces in CelebA dataset [19]. The size of images in this dataset

(a) SPGAN (using Improved WGAN)


(b) WGAN

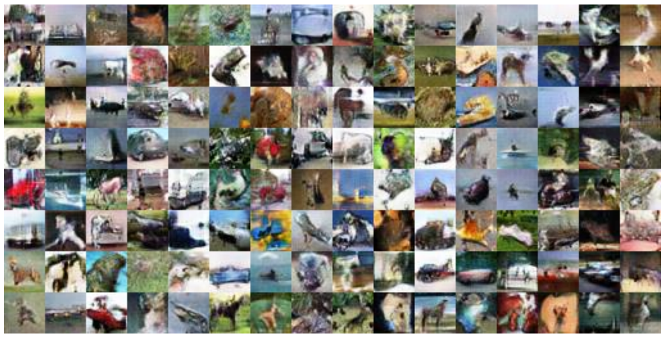

(c) Improved WGAN

Figure 3: Generated images using CIFAR10 dataset

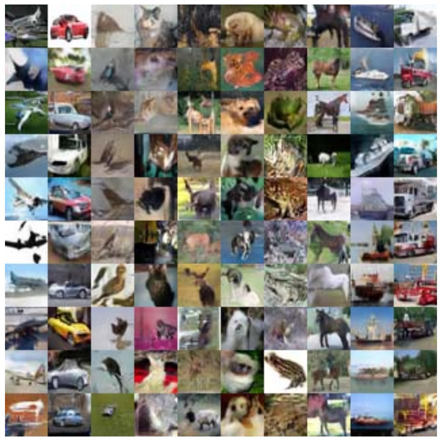
Table 1: Inception score on CIFAR10 images without residual blocks in generator

Method	score	SPGAN	SPGAN recon.
ALI [7]	5.36	-	-
BEGAN [2]	5.62	-	-
WGAN [1]	5.76	6.1	6.6
Im-WGAN [11]	5.92	6.2	6.7

Table 2: Inception score on CIFAR10 images with residual blocks in generator

Method	score	SPGAN	SPGAN recon.
ALI [7]	5.36	-	-
BEGAN [2]	5.62	-	-
WGAN [1]	7.73	7.85	7.88
Im-WGAN [11]	7.86	7.93	7.95

is 64×64. In order to generate a 64×64 image from the 256 dimensional noise vector, at the first layer, a fully connected layer expands the input dimension to a 4×4×2048 tensor (reshaped as tensor) and subsequently a series of four fractionally-strided convolutions transform the spatial dimension to 64×64 (doubling the spatial dimension in each layer). The number of channels has changed from 2048, to1024 at the third layer, 512 at the fourth layer and to 100 at the fifth layer. The size of the pre-trained dictionary Ω in this experiment is 27×100.


(a) SPGAN (using Improved WGAN)

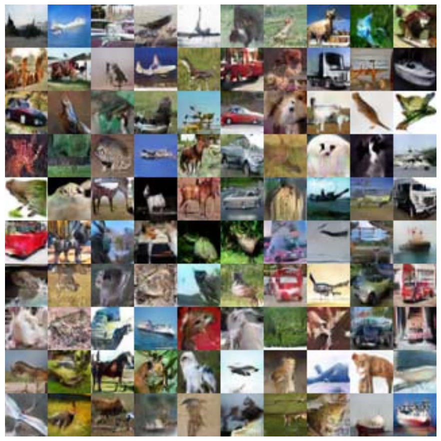
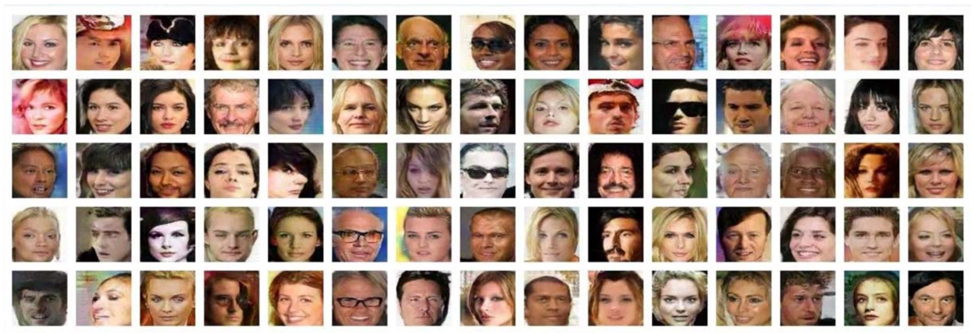

(b) Improved WGAN

Figure 4: Generated images using CIFAR10 dataset using Resnet blocks


(a) SPGAN (using Improved WGAN)

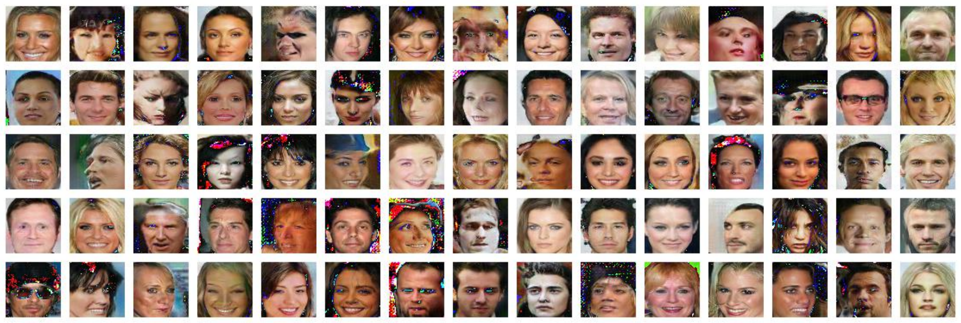

(b) Improved WGAN

Figure 5: Generated images using celebraty face dataset

Fig. 5 shows the generated images in this experiment and compares them with the generated images by WGAN [1] and Improved WGAN [11]. As one can see the images generated by the proposed method have higher quality and variance compared to the images generated by the other methods.

6. Conclusions

In this paper, we developed an enhanced GAN architecture, which generates images from patches, obtained through a UoS model. We also proposed a third player, called reconstructor to ensure high variability of the output images. We used two image datasets to evaluate the performance of the proposed generative adversarial method and demonstrated the remarkable advantage of regularizing the generative network by a reconstructor network and learning image characteristics at multiple scales. The evaluation results show the merit of the proposed method for generating images. The idea can be generalized to an arbitrary number of layers in different datasets.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[2] D. Berthelot, T. Schumm, and L. Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[3] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

[4] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

[5] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015.

[6] D. L. Donoho. De-noising by soft-thresholding. *IEEE transactions on information theory*, 41(3):613–627, 1995.

[7] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[8] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariance sparse coding for audio classification. *arXiv preprint arXiv:1206.5241*, 2012.

[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

[12] A. B. Hamza and H. Krim. Image denoising: A nonlinear robust statistical approach. *IEEE transactions on signal processing*, 49(12):3045–3054, 2001.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[15] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[16] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[17] J. Li, W. Monroe, T. Shi, S. Jean, A. Ritter, and D. Jurafsky. Adversarial learning for neural dialogue generation. *arXiv preprint arXiv:1701.06547*, 2017.

[18] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[20] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool. Pose guided person image generation. In *Advances in Neural Information Processing Systems*, pages 406–416, 2017.

[21] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proceedings of the 26th annual international conference on machine learning*, pages 689–696. ACM, 2009.

[22] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

[23] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[24] L. Na, G. Yang, A. Dongsheng, S. Kehua, L. Shixia, L. Zhongxuan, Y. Shing-Tung, and G. Xianfeng. Optimal transportation view of generative adversarial networks. 2019.

[25] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[26] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

[27] K. Tran, A. Panahi, A. Adiga, W. Sakla, and H. Krim. Nonlinear multi-scale super-resolution using deep learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3182–3186. IEEE, 2019.

[28] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[29] M. Xu, X. Jia, M. Pickering, and A. J. Plaza. Cloud removal based on sparse representation via multitemporal dictionary learning. *IEEE Transactions on Geoscience and Remote Sensing*, 54(5):2998–3006, 2016.

[30] Z. Yi, H. Zhang, P. Tan, and M. Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2849–2857, 2017.

[31] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon. Pixel-level domain transfer. In *European Conference on Computer Vision*, pages 517–532. Springer, 2016.

[32] D. Zhang, P. Liu, K. Zhang, H. Zhang, Q. Wang, and X. Jing. Class relatedness oriented-discriminative dictionary learning for multiclass image classification. *Pattern Recognition*, 59:168 – 175, 2016. Compositional Models and Structured Learning for Visual Recognition.

[33] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv preprint arXiv:1710.10916*, 2017.

[34] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017.

[35] Y. Zhang, Z. Gan, and L. Carin. Generating text via adversarial training. In *NIPS workshop on Adversarial Training*, volume 21, 2016.

[36] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, A. A. Efros, O. Wang, and E. Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, pages 465–476, 2017.