Multi-Agent Diverse Generative Adversarial Networks

Arnab Ghosh*
University of Oxford, UK
arnabg@robots.ox.ac.uk

Viveka Kulharia*
University of Oxford, UK
viveka@robots.ox.ac.uk

Vinay Namboodiri
IIT Kanpur, India
vinaypn@iitk.ac.in

Philip H.S. Torr
University of Oxford, UK
philip.torr@eng.ox.ac.uk

Puneet K. Dokania
University of Oxford, UK
puneet@robots.ox.ac.uk

Abstract

We propose MAD-GAN, an intuitive generalization to the Generative Adversarial Networks (GANs) and its conditional variants *to address the well known problem of* mode collapse. *First, MAD-GAN is a multi-agent GAN architecture incorporating multiple generators and one discriminator. Second, to enforce that different generators capture diverse high probability modes, the discriminator of MAD-GAN is designed such that along with finding the real and fake samples, it is also required to identify the generator that generated the given fake sample. Intuitively, to succeed in this task, the discriminator must learn to push different generators towards different identifiable modes. We perform extensive experiments on synthetic and real datasets and compare MAD-GAN with different variants of GAN. We show high quality diverse sample generations for challenging tasks such as image-to-image translation and face generation. In addition, we also show that MAD-GAN is able to disentangle different modalities when trained using highly challenging diverse-class dataset (*e.g. *dataset with images of forests, icebergs, and bedrooms). In the end, we show its efficacy on the unsupervised feature representation task.*

1. Introduction

Generative models have attracted considerable attention recently. The underlying idea behind such models is to attempt to capture the distribution of high-dimensional data such as images and texts. Though these models are highly useful in various applications, it is computationally expensive to train them as they require intractable integration in a very high-dimensional space. This drastically limits their applicability. However, recently there has been considerable progress in *deep generative models* – conglomerate of deep neural networks and generative models – as they do not explicitly require the intractable integration, and can be efficiently trained using back-propagation algorithm. Two such famous examples are Generative Adversarial Networks (GANs) [11] and Variational Autoencoders [14].

Figure 1: Diverse-class data generation using MAD-GAN. Diverse-class dataset contains images from different classes/modalities (in this case, forests, icebergs, and bedrooms). Each row represents generations by a particular generator and each column represents generations for a given random noise input z. As shown, once trained using this dataset, generators of MAD-GAN are able to disentangle different modalities, hence, each generator is able to generate images from a particular modality.

In this paper we focus on GANs as they are known to produce sharp and plausible images. Briefly, GANs employ a generator and a discriminator where both are involved in a minimax game. The task of the discriminator is to learn the difference between *real* samples (from true data distribution p_d) and *fake* samples (from generator distribution p_g). Whereas, the task of the generator is to maximize the mistakes of the discriminator. At convergence, the generator learns to produce real looking images. A few successful applications of GANs are video generation [26], image inpainting [22], image manipulation [29], 3D object generation [27], interactive image generation using few brush strokes [29], image super-resolution [17], diagrammatic abstract reasoning [15] and conditional GANs [20, 24].

Despite the remarkable success of GAN, it suffers from the major problem of *mode collapse* [2, 7, 8, 19, 25].

*Joint first author

Though, theoretically, convergence guarantees the generator learning the true data distribution. However, practically, reaching the true equilibrium is difficult and not guaranteed, which potentially leads to the aforementioned problem of mode collapse. Broadly speaking, there are two schools of thought to address the issue: (1) improving the learning of GANs to reach better optima [2, 19, 25]; and (2) explicitly enforcing GANs to capture diverse modes [7, 8, 18]. Here we focus on the latter.

Borrowing from the multi-agent algorithm [1] and coupled GAN [18], we propose to use multiple generators with one discriminator. We call this framework the Multi-Agent GAN architecture, as shown in Fig. 2. In detail, similar to the standard GAN, the objective of each generator here is to maximize the mistakes of the *common* discriminator. Depending on the task, it might be useful for different generators to share information. This is done using the initial layer parameters of generators. Another reason behind sharing these parameters is the fact that initial layers capture low-frequency structures which are almost the same for a particular type of dataset (for example, faces), therefore, sharing them reduces redundant computations. However, when the dataset contains images from completely different modalities, one can avoid sharing these parameters. Naively using multiple generators may lead to the *trivial solution* where all the generators learn to generate *similar* samples. To resolve this issue and generate different visually plausible samples capturing diverse high probability modes, we propose to modify the objective function of the discriminator. In the modified objective, along with finding the real and the fake samples, the discriminator also has to correctly identify the generator that generated the given fake sample. Intuitively, in order to succeed in this task, the discriminator must learn to push generations corresponding to different generators towards different identifiable modes. Combining the Multi-Agent GAN architecture with the diversity enforcing term allows us to generate diverse plausible samples, thus the name Multi-Agent Diverse GAN (MAD-GAN).

As an example, an intuitive setting where mode collapse occurs is when a GAN is trained on a dataset containing images from different modalities/classes. For example, a diverse-class dataset containing images such as forests, iceberg, and bedrooms. This is of particular interest as it not only requires the model to disentangle intra-class variations, it also requires inter-class disentanglement. Fig. 1 demonstrates the surprising effectiveness of MAD-GAN in this challenging setting. Generators among themselves are able to disentangle inter-class variations, and each generator is also able to capture intra-class variations.

In addition, we analyze MAD-GAN through extensive experiments and compare it with several variants of GAN. First, for the proof of concept, we perform experiments in controlled settings using synthetic dataset (mixture of Gaussians), and complicated Stacked/Compositional MNIST datasets with hand engineered modes. In these settings, we empirically show that our approach outperforms all other GAN variants we compare with, and is able to generate high quality samples while capturing large number of modes. In a more realistic setting, we show high quality diverse sample generations for the challenging tasks of *image-to-image translation* [12] (conditional GAN) and *face generation* [8, 23]. Using the SVHN dataset [21], we also show the efficacy of our framework for learning the feature representation in an unsupervised setting.

We also provide theoretical analysis of this approach and show that the proposed modification in the objective of discriminator allows generators to learn together as a mixture model where each generator represents a mixture component. We show that at convergence, the global optimum value of $-(k+1)\log(k+1) + k\log k$ is achieved, where k is the number of generators.

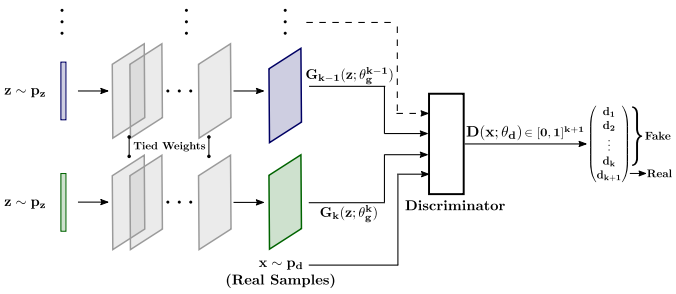


Figure 2: Multi-Agent Diverse GAN (MAD-GAN). The discriminator outputs $k+1$ softmax scores signifying the probability of its input sample being from either one of the k generators or the real distribution.

2. Related Work

The recent work called InfoGAN [8] proposed an information-theoretic extension to GANs in order to address the problem of mode collapse. Briefly, InfoGAN disentangles the latent representation by assuming a factored representation of the latent variables. In order to enforce that the generator learns factor specific generations, InfoGAN maximizes the mutual information between the factored latents and the generator distribution. Che *et al*. [7] proposed a mode regularized GAN (ModeGAN) which uses an encoder-decoder paradigm. The basic idea behind ModeGAN is that if a sample from the true data distribution p_d belongs to a particular mode, then the sample generated by the generator (fake sample) when the true sample is passed through the encoder-decoder is likely to belong to the same mode. ModeGAN assumes that there exists enough true samples from a mode for the generator to be able to capture it. Another work by Metz *et al*. [19] proposed a surrogate objective for the update of the generator with respect to the

unrolled optimization of the discriminator (UnrolledGAN) to address the issue of convergence of the training process of GANs. This improves the training process of the generator which in turn allow the generators to explore better coverage to true data distribution.

Liu *et al*. [18] presented Coupled GAN, a method for training two generators with shared parameters to learn the joint distribution of the data. The shared parameters guide both the generators towards similar subspaces but since they are trained independently on two domains, they promote diverse generations. Durugkar *et al*. [9] proposed a model with multiple discriminators whereby an ensemble of multiple discriminators have been shown to stabilize the training of the generator by guiding it to produce better samples.

W-GAN [3] is a recent technique which employs integral probability metrics based on the earth mover distance rather than the JS-divergences that the original GAN uses. BE-GAN [5] builds upon W-GAN using an autoencoder based equilibrium enforcing technique alongside the Wasserstein distance. DCGAN [23] was a seminal technique which used a fully convolutional generator and discriminator for the first time along with the introduction of batch normalization thus stabilizing the training procedure, and was able to generate compelling generations. GoGAN [13] introduced a training procedure for the training of the discriminator using a maximum margin formulation alongside the earth mover distance based on the Wasserstein-1 metric. [4] introduced a technique and theoretical formulation stating the importance of multiple generators and discriminators in order to completely model the data distribution. In terms of employing multiple generators, our work is closest to [4, 18, 10]. However, while using multiple generators, our method explicitly enforces them to capture diverse modes.

3. Preliminaries

Here we present a brief review of GANs [11]. Given a set of samples $\mathcal{D} = (x_i)_{i=1}^n$ from the true data distribution p_d, the GAN learning problem is to obtain the optimal parameters θ_g of a generator $G(z;\theta_g)$ that can sample from an approximate data distribution p_g, where $z \sim p_z$ is the prior input noise (*e.g*. samples from a normal distribution). In order to learn the optimal θ_g, the GAN objective (Eq. (1)) employs a discriminator $D(x;\theta_d)$ that learns to differentiate between a *real* (from p_d) and a *fake* (from p_g) sample x. The overall GAN objective is:

$$\min_{\theta_g}\max_{\theta_d} V(\theta_d,\theta_g) := \mathbb{E}_{x\sim p_d}\log D(x;\theta_d) + \mathbb{E}_{z\sim p_z}\log\big(1-D(G(z;\theta_g);\theta_d)\big) \quad (1)$$

The above objective is optimized in a block-wise manner where θ_d and θ_g are optimized one at a time while fixing the other. For a given sample x (either from p_d or p_g) and the parameter θ_d, the function $D(x;\theta_d) \in [0,1]$ produces a score that represents the probability of x belonging to the true data distribution p_d (or probability of it being real). The objective of the discriminator is to learn parameters θ_d that maximizes this score for the true samples (from p_d) while minimizing it for the fake ones $\tilde{x} = D(z;\theta_g)$ (from p_g). In the case of generator, the objective is to minimize $\mathbb{E}_{z\sim p_z}\log\big(1-D(G(z;\theta_g);\theta_d)\big)$, equivalently maximize $\mathbb{E}_{z\sim p_z}\log D(G(z;\theta_g);\theta_d)$. Thus, the generator learns to maximize the scores for the fake samples (from p_g), which is exactly the opposite to what discriminator is trying to achieve. In this manner, the generator and the discriminator are involved in a minimax game where the task of the generator is to maximize the mistakes of the discriminator. Theoretically, at equilibrium, the generator learns to generate real samples, which means $p_g = p_d$.

4. Multi-Agent Diverse GAN

In the GAN objective, one can argue that the task of a generator is much harder than that of the discriminator as it has to produce real looking images to maximize the mistakes of the discriminator. This, along with the minimax nature of the objective raise several challenges for GANs [2, 7, 8, 19, 25]: (1) mode collapse; (2) difficult optimization; and (3) trivial solution. In this work we propose a new framework to address the first challenge of *mode collapse* by increasing the capacity of the generator while using well known tricks to partially avoid other challenges [2].

Briefly, we propose a *Multi-Agent GAN architecture* that employs multiple generators and one discriminator in order to generate different samples from high probability regions of the true data distribution. In addition, theoretically, we show that our formulation allows generators to act as a mixture model with each generator capturing one component.

4.1. Multi-Agent GAN Architecture

Here we describe our proposed architecture (Fig. 2). It involves k generators and one discriminator. In the case of homogeneous data (all the images belong to same class, *e.g*. faces or birds), we allow all the generators to share information by tying most of the initial layer parameters. This is essential to avoid redundant computations as initial layers of a generator capture low-frequency structures which are almost the same for a particular type of dataset. This also allows different generators to converge faster. However, in the case of *diverse-class* data (*e.g*. dataset with a mixture of different classes such as forests, icebergs *etc*.), it is necessary to avoid sharing these parameters to allow each generator to capture content specific structures. Thus, the extent to which one should share these parameters depends on the task at hand.

More specifically, given $z \sim p_z$ for the i-th generator, similar to the standard GAN, the first step involves gen-

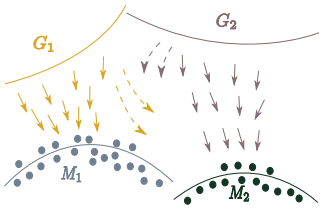


Figure 3: Visualization of different generators getting pushed towards different modes. Here, M_1 and M_2 could be a cluster of modes where each cluster itself contains many modes. The arrows abstractly represent generator specific gradients for the purpose of building intuition.

erating a sample (for example, an image) $\tilde{x}_i$. Since each generator receives the same latent input sampled from the same distribution, naively using this simple approach may lead to the *trivial solution* where all the generators learn to generate similar samples. In what follows, we propose an intuitive solution to avoid this issue and allow the generators to capture diverse modes.

4.2. Enforcing Diverse Modes

Inspired by the discriminator formulation for the semi-supervised learning [25], we use a generator identification based objective function that, along with minimizing the score $D(\tilde{x};\theta_d)$, requires the discriminator to identify the generator that generated the given fake sample $\tilde{x}$. In order to do so, as opposed to the standard GAN objective function where the discriminator outputs a scalar value, we modify it to output $k+1$ soft-max scores. In more detail, given the set of k generators, the discriminator produces a soft-max probability distribution over $k+1$ classes. The score at $(k+1)$-th index $(D_{k+1}(.))$ represents the probability that the sample belongs to the true data distribution and the score at $j \in \{1,\ldots,k\}$-th index represents the probability of it being generated by the j-th generator. Under this setting, while learning θ_d, we optimize the cross-entropy between the soft-max output of the discriminator and the Dirac delta distribution $\delta \in \{0,1\}^{k+1}$, where for $j \in \{1,\ldots,k\}$, $\delta(j)=1$ if the sample belongs to the j-th generator, otherwise $\delta(k+1)=1$. Thus, the objective of the discriminator, which is optimizing θ_d while keeping θ_g constant (refer Eq. (1)), is modified to:

$$\max_{\theta_d} \mathbb{E}_{x\sim p}\, H(\delta, D(x;\theta_d))$$

where, $Supp(p) = \cup_{i=1}^{k} Supp(p_{g_i}) \cup Supp(p_d)$ and $H(.,.)$ is the negative of the cross entropy function. Intuitively, in order to correctly identify the generator that produced a given fake sample, the discriminator must learn to push different generators towards different identifiable modes. However, the objective of each generator remains the same as in the standard GAN. Thus, for the i-th generator, the objective is to minimize the following:

$$\mathbb{E}_{x\sim p_d} \log D_{k+1}(x;\theta_d) + \mathbb{E}_{z\sim p_z} \log(1 - D_{k+1}(G_i(z;\theta_g^i);\theta_d))$$

To update the parameters, the gradient for each generator is simply computed as $\nabla_{\theta_g^i} \log(1 - D_{k+1}(G_i(z;\theta_g^i);\theta_d)))$. Notice that all the generators in this case can be updated in parallel. For the discriminator, given $x \sim p$ (can be real or fake) and corresponding δ, the gradient is $\nabla_{\theta_d} \log D_j(x;\theta_d)$, where $D_j(x;\theta_d)$ is the j-th index of $D(x;\theta_d)$ for which $\delta(j)=1$. Therefore, using this approach requires *very minor modifications to the standard GAN optimization algorithm* and can be easily used with different variants of GAN. An intuitive visualization is shown in Fig. 3.

Theorem 1 shows that the above objective function actually allows generators to form a mixture model where each generator represents a mixture component and the global optimum of $-(k+1)\log(k+1) + k\log k$ is achieved when $p_d = \frac{1}{k}\sum_{i=1}^{k} p_{g_i}$. Notice that, at $k=1$, which is the case with one generator, we obtain exactly the same Jensen-Shannon divergence based objective function as shown in [11] with the optimal value of $-\log 4$.

Theorem 1. *Given the optimal discriminator, the objective for training the generators boils down to minimizing*

$$KL\Big(p_d(x)||p_{avg}(x)\Big) + kKL\Big(\frac{1}{k}\sum_{i=1}^{k} p_{g_i}(x)||p_{avg}(x)\Big) - (k+1)\log(k+1) + k\log k \tag{2}$$

where, $p_{avg}(x) = \frac{p_d(x)+\sum_{i=1}^{k} p_{g_i}(x)}{k+1}$. *The above objective function obtains its global minimum if* $p_d = \frac{1}{k}\sum_{i=1}^{k} p_{g_i}$ *with the objective value of* $-(k+1)\log(k+1) + k\log k$.

Proof. The joint objective of all the generators is to minimize the following:

$$\mathbb{E}_{x\sim p_d} \log D_{k+1}(x) + \sum_{i=1}^{k} \mathbb{E}_{x\sim p_{g_i}} \log(1 - D_{k+1}(x))$$

Using Corollary 1, we substitute the optimal discriminator in the above equation and obtain:

$$\begin{aligned}
&\mathbb{E}_{x\sim p_d} \log\left[\frac{p_d(x)}{p_d(x)+\sum_{i=1}^{k} p_{g_i}(x)}\right] + \\
&\sum_{i=1}^{k} \mathbb{E}_{x\sim p_{g_i}} \log\left[\frac{\sum_{i=1}^{k} p_{g_i}(x)}{p_d(x)+\sum_{i=1}^{k} p_{g_i}(x)}\right] \\
&= \mathbb{E}_{x\sim p_d} \log\left[\frac{p_d(x)}{p_{avg}(x)}\right] + k\mathbb{E}_{x\sim p_g} \log\left[\frac{p_g(x)}{p_{avg}(x)}\right] \\
&- (k+1)\log(k+1) + k\log k
\end{aligned} \tag{3}$$

where, $p_g = \frac{\sum_{i=1}^{k} p_{g_i}}{k}$ and $p_{avg}(x) = \frac{p_d(x)+\sum_{i=1}^{k} p_{g_i}(x)}{k+1}$. Note that, Eq. (3) is exactly the same as Eq. (2). When $p_d = \frac{\sum_{i=1}^{k} p_{g_i}}{k}$, both the KL terms become zero and the global minimum is achieved. □

Corollary 1. *For fixed generators, the optimal distribution learned by the discriminator D has the following form:*

$$D_{k+1}(x) = \frac{p_d(x)}{p_d(x) + \sum_{i=1}^{k} p_{g_i}(x)},$$
$$D_i(x) = \frac{p_{g_i}(x)}{p_d(x) + \sum_{i=1}^{k} p_{g_i}(x)}, \forall i \in \{1, \cdots, k\}.$$

where, $D_i(x)$ represents the i-th index of $D(x; \theta_d)$, p_d the true data distribution, and p_{g_i} the distribution learned by the i-th generator.

Proof. For fixed generators, the objective function of the discriminator is to maximize

$$\mathbb{E}_{x \sim p_d} \log D_{k+1}(x) + \sum_{i=1}^{k} \mathbb{E}_{x_i \sim p_{g_i}} \log D_i(x_i)$$

where, $\sum_{i=1}^{k+1} D_i(x) = 1$ and $D_i(x) \in [0, 1], \forall i$. The above equation can be written as:

$$\int_x p_d(x) \log D_{k+1}(x) dx + \sum_{i=1}^{k} \int_x p_{g_i}(x) \log D_i(x) dx$$
$$= \int_{x \in p} \sum_{i=1}^{k+1} p_i(x) \log D_i(x) dx \quad (4)$$

where, $p_{k+1}(x) := p_d(x)$, $p_i(x) := p_{g_i}(x), \forall i \in \{1, \cdots, k\}$, and $Supp(p) = \bigcup_{i=1}^{k} Supp(p_{g_i}) \bigcup Supp(p_d)$, Therefore, for a given x, the optimum of objective function defined in Eq. (4) with constraints defined above can be obtained using Proposition 1. □

Proposition 1. *Given $\mathbf{y} = (y_1, \cdots, y_n)$, $y_i \geq 0$, and $a_i \in \mathbb{R}$, the optimal solution for the objective function defined below is achieved at $y_i^* = \frac{a_i}{\sum_{i=1}^{n} a_i}, \forall i$*

$$\max_{\mathbf{y}} \sum_{i=1}^{n} a_i \log y_i, \quad s.t. \sum_{i}^{n} y_i = 1$$

Proof. The Lagrangian of the above problem is:

$$L(\mathbf{y}, \lambda) = \sum_{i=1}^{n} a_i \log y_i + \lambda(\sum_{i=1}^{n} y_i - 1)$$

Differentiating w.r.t y_i and λ, and equating to zero,

$$\frac{a_i}{y_i} + \lambda = 0 \ , \ \sum_{i=1}^{n} y_i - 1 = 0$$

Solving the above two equations, we obtain $y_i^* = \frac{a_i}{\sum_{i=1}^{n} a_i}$. □

GAN Variants	**Chi-square**($\times 10^5$)	**KL-Div**
DCGAN [23]	0.90	0.322
WGAN [3]	1.32	0.614
BEGAN [5]	1.06	0.944
GoGAN [13]	2.52	0.652
Unrolled GAN [19]	3.98	1.321
Mode-Reg DCGAN [7]	1.02	0.927
InfoGAN [8]	0.83	0.21
MA-GAN	1.39	0.526
MAD-GAN (Our)	**0.24**	**0.145**

Table 1: Synthetic experiment on 1D GMM (Fig. 4).

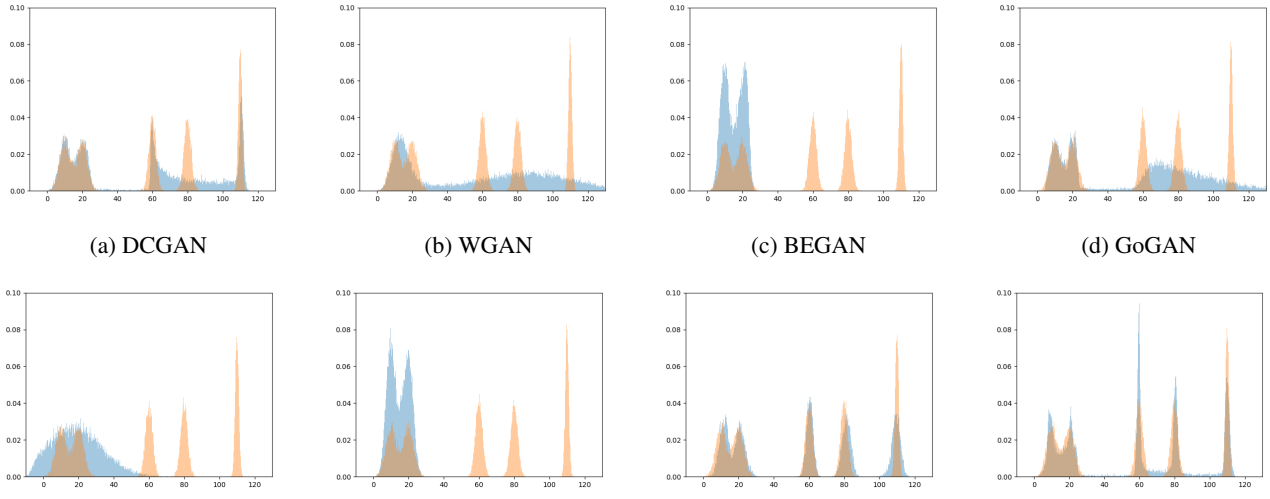
5. Experiments

We present an extensive quantitative and qualitative analysis of MAD-GAN on various synthetic and real-world datasets. First, we use a simple 1D mixture of Gaussians and also Stacked/Compositional MNIST dataset (1000 modes) to compare MAD-GAN with several known variants of GANs, such as DCGAN [23], WGAN [3], BEGAN [5], GoGAN [13], Unrolled GAN [19], Mode-Reg GAN [7] and InfoGAN [8]. Furthermore, we created another baseline, called *MA-GAN (Multi-Agent GAN)*, which is a trivial extension of GAN with multiple generators and one discriminator. As opposed to MAD-GAN, MA-GAN has a simple Multi-Agent architecture without modifications to the objective of the discriminator. This comparison allows us to understand the effect of explicitly enforcing diversity in the objective of the MAD-GAN. We use KL-divergence [16] and number of modes recovered [7] as the criterion for comparisons and show superior results compared to all the other methods. Additionally, we show diverse generations for the challenging tasks of *image-to-image translation* [12], *diverse-class data generation*, and face generation. It is non-trivial to devise a metric to evaluate diversity on these high quality generation tasks, so we perform qualitative assessment. Note that, the image-to-image translation objective is known to learn the delta distribution, thus, it is agnostic to the input noise vector. However, we show that MAD-GAN is able to produce highly plausible diverse generations for this task. In the end, we show the efficacy of MAD-GAN in unsupervised feature representation learning task. We provide detailed overview of the architectures, datasets, and the parameters used in our experiments in the supplementary.

In the case of InfoGAN [8], we varied the dimension of the categorical variable, depicting the number of modes, to obtain the best cross-validated results.

5.1. Non-Parametric Density Estimation

In order to understand the behavior of MAD-GAN and different state-of-the-art GAN models, we first perform a



(a) DCGAN (b) WGAN (c) BEGAN (d) GoGAN

(e) Unrolled GAN (f) Mode-Reg DCGAN (g) InfoGAN (h) MAD-GAN (Our)

Figure 4: A toy example to understand the behaviour of different GAN variants in order to compare with MAD-GAN (each method was trained for 198000 iterations). The orange bars show the density estimate of the training data and the blue ones for the generated data points. After careful cross-validation, we chose the bin size of 0.1.

very simple synthetic experiment, much easier than generating high-dimensional complex images. We consider a distribution of 1D GMM [6] having five mixture components with modes at 10, 20, 60, 80 and 110, and standard deviations of 3, 3, 2, 2 and 1, respectively. While the first two modes overlap significantly, the fifth mode stands isolated as shown in Fig. 4. We train different GAN models using $200,000$ samples from this distribution and generate $65,536$ data points from each model. In order to compare the learned distribution with the ground truth distributions, we first estimate them using bins over the data points and create the histograms. These histograms are carefully created using different bin sizes and the best bin (found to be 0.1) is chosen. Then, we use Chi-square distance and the KL-divergence to compute distance between the two histograms. From Fig. 4 and Tab. 1 it is evident that MAD-GAN is able to capture all the clustered modes which includes significantly overlapped modes as well. MAD-GAN obtains the minimum value in terms of both Chi-square distance and the KL-divergence. In this experiment, both MAD-GAN and MA-GAN used four generators. In the case of InfoGAN, we used 5 dimensional categorical variable, which provides the best result.

5.2. Stacked and Compositional MNIST

We now perform experiments on a more challenging setup, similar to [7, 19], in order to examine and compare MAD-GAN with other GAN variants. [19] created a Stacked-MNIST dataset with $25,600$ samples where each sample has three channels stacked together with a random MNIST digit in each of them. Thus, it creates 1000 distinct

GAN Variants	KL Div	# Modes Covered
DCGAN [23]	2.15	712
WGAN [3]	1.02	868
BEGAN [5]	1.89	819
GoGAN [13]	2.89	672
Unrolled GAN [19]	1.29	842
Mode-Reg DCGAN [7]	1.79	827
InfoGAN [8]	2.75	840
MA-GAN	3.4	700
MAD-GAN (Our)	**0.91**	**890**

Table 2: Stacked-MNIST experiments and comparisons. Note that three generators are used for MAD-GAN.

GAN Variants	KL Div	# Modes Covered
DCGAN [23]	0.18	980
WGAN [3]	0.25	1000
BEGAN [5]	0.19	999
GoGAN [13]	0.87	972
Unrolled GAN [19]	0.091	1000
Mode-Reg DCGAN [7]	0.12	992
InfoGAN [8]	0.47	990
MA-GAN	1.62	997
MAD-GAN (Our)	**0.074**	**1000**

Table 3: Compositional-MNIST experiments and comparisons. Note that three generators are used for MAD-GAN.

Figure 6: Diverse generations for *edges-to-handbags* generation task. In each sub-figure, the first column represents the input, columns 2-4 represents generations by MAD-GAN (using three generators), and columns 5-7 are generations by InfoGAN (using three categorical codes). It is evident that different generators of MAD-GAN are able to produce diverse results capturing different colors, textures, design patterns, among others. However, InfoGAN generations are visually almost the same, indicating mode collapse.

modes in the data distribution. [19] used a stripped down version of the generator and discriminator pair to reduce the modeling capacity. We do the same for fair comparisons and used the same architecture as mentioned in their paper. Similarly, [7] created Compositional-MNIST whereby they took 3 random MNIST digits and place them at the 3 quadrants of a 64×64 dimensional image. This also resulted in a data distribution with 1000 hand-designed modes. The distribution of the resulting generated samples was estimated using a pretrained MNIST classifier to classify each of the digits either in the channels or the quadrants to decide the mode it belongs to.

Tables 2 and 3 provide comparison of our method with variants of GAN in terms of KL divergence and the number of modes recovered for the Stacked and Compositional MNIST datasets, respectively. In Stacked-MNIST, as evident from the Tab. 2, MAD-GAN outperforms all other variants of GAN in both the criteria. Interestingly, in the case of Compositional-MNIST, as shown in Tab. 3, MAD-GAN, WGAN and Unrolled GAN were able to recover all the 1000 modes. However, in terms of KL divergence, the distribution generated by MAD-GAN is the closest to the true data distribution.

5.3. Diverse Samples for Image-to-Image Translation and Comparison to InfoGAN

Here we present experimental results on the challenging task of image-to-image translation [12] which uses conditional variant of GANs [20]. Conditional GAN for this task is known to learn the delta distribution, thus, generates the same image irrespective of the variations in the input noise vector. Generating diverse samples in this setting in itself is an open problem. We show that MAD-GAN is able to generate diverse samples in these experiments as well. We use three generators for MAD-GAN experiments and show three diverse generations. Note that, we do not claim to capture all the possible modes present in the data distribution because firstly we cannot estimate the number of modes a priori, and secondly, even if we could, we do not know how diverse the generations would be after using certain number of generators. We follow the same approach as [12] and employ patch based conditional GAN.

Figure 8: Diverse generations for *night-to-day* image generation task. First column in each sub-figure represents the input. The remaining three columns show the diverse generations of three different generators of MAD-GAN (Our).

We compare MAD-GAN with InfoGAN [8] in these experiments as it is closest to our approach and can be used in image-to-image translation task. Theoretically, latent codes in InfoGAN should enable diverse generations. However, InfoGAN can only be used when the bias introduced by the categorical variables have significant impact on the generator network. For image-to-image translation and high resolution generations, the categorical variable does not have

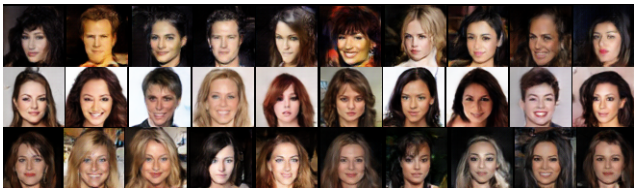


Figure 9: Face generations using MAD-GAN. Each generator employed is DCGAN. Each row represents a generator. Each column represents generations for a given random noise input z. Note that, the first generator is generating faces pointing to the left. The second generator is generating female faces with long hair, while the third generator generates images with light background.

sufficient impact on the generations. As will be seen shortly, we validate this hypothesis by comparing our method with InfoGAN for this task. For the InfoGAN generator, to capture three kinds of distinct modes, the categorical code is chosen to take three values. Since we are dealing with images, in this case, the categorical code is a 2D matrix in which we set one third of the entries to 1 and remaining to 0 for each category. The generator is fed input image along with categorical code appended channel wise to the image. Architecture of the Q network is same as that of the pix2pix discriminator [12], except that the output is a vector of size 3 for the prediction of the categorical codes. The discriminator and the Q network parameters are unshared. We observed that sharing them and even adding noise besides categorical code did not make any difference. Note that, we tried different variations of the categorical codes but did not observe any significant variation in the generations.

Fig. 6 shows generations by MAD-GAN and InfoGAN for the *edges-to-handbags* task, where given the edges of handbags, the objective is to generate real looking handbags. Clearly, each MAD-GAN generator is able to produce meaningful images but different from remaining generators in terms of color, texture, and patterns. However, InfoGAN generations are almost the same for all the three categorical codes. In addition, in Fig. 8, we show diverse generations for the *night-to-day* task, where given night images of places, the objective is to generate their corresponding day images. As can be seen, the generated day images in Fig. 8 differ in terms of lighting conditions, sky patterns, weather conditions, and many other minute yet useful cues.

5.4. Diverse-Class Data Generation

To further explore the mode capturing capacity of MAD-GAN, we experimented with a much more challenging task of diverse-class data generation. In detail, we trained MAD-GAN (three generators) on a combined dataset consisting of various highly diverse images such as *islets*, *icebergs*, *broadleaf-forest*, *bamboo-forest*, and *bedroom*, obtained from the Places dataset [28]. Images were randomly selected from each of them, creating a training dataset of $24,000$ images. The generators have the same architecture as that of DCGAN. In this case, as the images in the dataset belong to different classes, we did not share the generator parameters. As shown in Fig. 1, to our surprise, we found that even in this highly challenging setting, the generations from different generators belong to different classes. This clearly indicates that the generators in MAD-GAN are able to disentangle inter-class variations. In addition, each generator for different noise input is able to generate diverse samples, indicating intra-class diversity.

5.5. Diverse Face Generation

Here we show diverse face generations (CelebA dataset) using MAD-GAN where we use DCGAN [23] as each of our three generators. Again, we use the same setting as provided in DCGAN. The high quality face generations are shown in the Fig. 9.

5.6. Unsupervised Representation Learning

Similar to DCGAN [23], we train our framework using SVHN dataset [21]. The trained discriminator is used to extract features. Using these features, we train an SVM for the classification task. For the MAD-GAN, with three generators, we obtained misclassification error of 17.5% which is almost 5% better than the results reported by DCGAN (22.48%). This clearly indicates that our framework is able to learn a better feature space in an unsupervised setting.

6. Conclusion

We presented a very simple and effective framework, Multi-Agent Diverse GAN (MAD-GAN), for generating diverse and meaningful samples. We showed the efficacy of our approach and compared it with various variants of GAN that it captures diverse modes while producing high quality samples. We presented a theoretical analysis of MAD-GAN with conditions for global optimality. Looking forward, an interesting future direction would be to estimate a priori the number of generators needed for a particular dataset. It is not clear how to do that given that we do not have access to the true data distribution. In addition, we would also like to theoretically understand the limiting cases that depend on the relationship between the number of generators and the complexity of the data distribution. Another interesting direction would be to exploit different generators such that their combinations can be used to capture diverse modes.

7. Acknowledgements

This work was supported by the EPSRC, ERC grant ERC-2012-AdG 321162-HELIOS, EPSRC grant Seebibyte EP/M013774/1 and EPSRC/MURI grant EP/N019474/1.

References

[1] M. Abadi and D. Andersen. Learning to protect communications with adversarial neural cryptography. *arXiv preprint arXiv:1610.06918*, 2016.

[2] M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[3] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. In *International Conference on Machine Learning*, 2017.

[4] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, 2017.

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

[6] C. Bishop. Pattern recognition and machine learning (information science and statistics), 1st edn. 2006. corr. 2nd printing edn. *Springer, New York*, 2007.

[7] T. Che, Y. Li, A. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[8] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2016.

[9] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. In *International Conference on Learning Representations*, 2017.

[10] A. Ghosh, V. Kulharia, and V. Namboodiri. Message passing multi-agent gans. *arXiv preprint arXiv:1612.01294*, 2016.

[11] 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*, 2014.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition*, 2017.

[13] F. Juefei-Xu, V. N. Boddeti, and M. Savvides. Gang of gans: Generative adversarial networks with maximum margin ranking. *arXiv preprint arXiv:1704.04865*, 2017.

[14] D. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

[15] V. Kulharia, A. Ghosh, A. Mukerjee, V. Namboodiri, and M. Bansal. Contextual rnn-gans for abstract reasoning diagram generation. In *AAAI Conference on Artificial Intelligence*, 2017.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 1951.

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Computer Vision and Pattern Recognition*, 2017.

[18] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, 2016.

[19] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *International Conference on Learning Representations*, 2017.

[20] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[21] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition*, 2016.

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

[24] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *International Conference on Machine Learning*, 2016.

[25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2016.

[26] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, 2016.

[27] J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, 2016.

[28] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[29] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. A. Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, 2016.