Generative Adversarial Networks (GANs): What it can generate and What it cannot?

P Manisha
manisha.padala@research.iiit.ac.in

Sujit Gujar
sujit.gujar@iiit.ac.in

International Institute of Information Technology,Hyderabad, India

April 3, 2018

Abstract

Why are Generative Adversarial Networks (GANs) so popular? What is the purpose of designing GANs? Can we justify functioning of GANs theoretically? How are the theoretical guarantees? Are there any shortcomings?

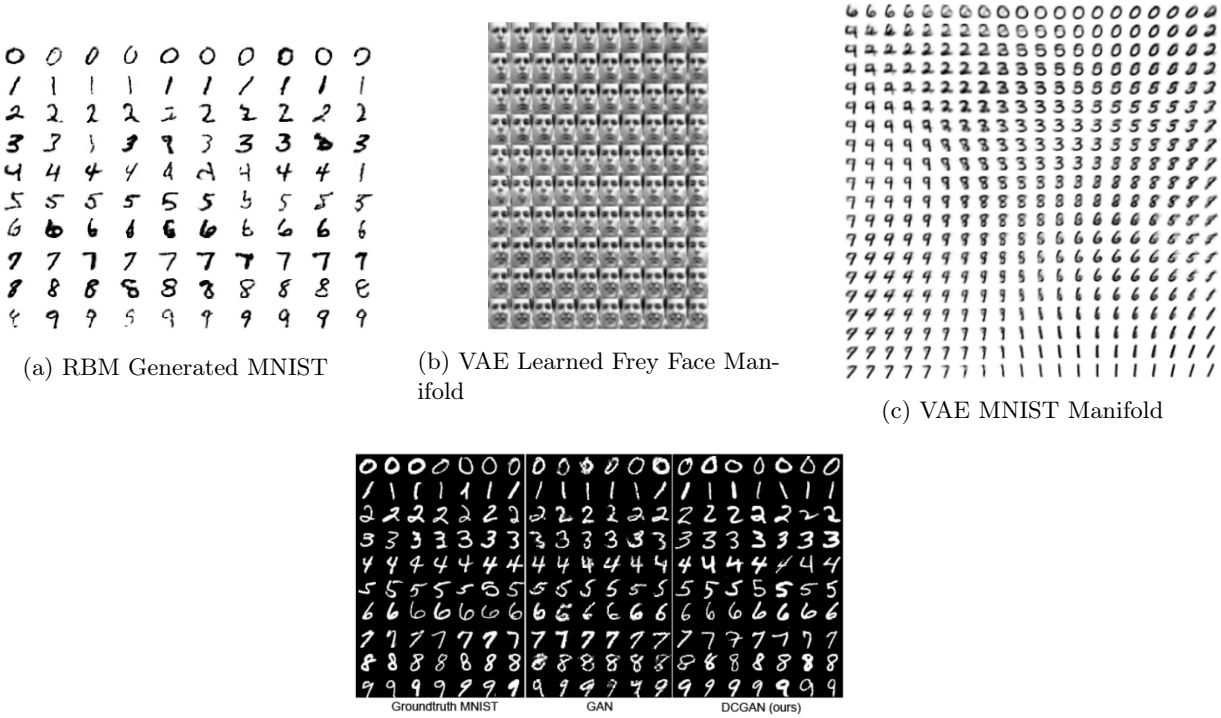
With the popularity of GANs, the researchers across the globe have been perplexed by these questions. In the last year (2017), a plethora of research papers attempted to answer the above questions. In this article, we put in our best efforts to compare and contrast different results and put forth a summary of theoretical contributions about GANs with focus on image/visual applications. Our main aim is to highlight the primary issues related to GANs that each of these papers examine. Besides we provide insight into how each of the discussed articles solve the concerned problems. We expect this summary paper to give a bird's eye view to a person wishing to understand the theory behind GANs.

1 Introduction

A *generative model* is trained to learn the underlying distribution of the data. The idea behind having such models is not to memorize the entire data, but to learn those specific semantic and structural properties which help the model create new samples. These samples need not belong to the training set, yet can convincingly become a part of it. The other popular models such as *Restricted Boltzmann Machines* (RBMs) [9], *Variational Auto-encoders* (VAEs) [11] make use of latent variables as a hidden representation of the data samples. These models specify an explicit parameterized log-likelihood functions representing the data. The parameters are learned from the data. Estimating the maximum likelihood of the parameters requires integrating over the entire space of latent variables, which is intractable. Hence approximation techniques are used which may not always yield the best results. On the other hand, *Generative Adversarial Networks*, GANs, are one of the few implicit probabilistic models which define a stochastic procedure that directly generates data from a latent variable that belongs to a lower dimensional space.

GANs were first proposed by Goodfellow *et. al.* [24]. It was further improved to DCGAN by Alec *et. al.* [22]. Figure 1 illustrates the images generated by RBMs, VAEs, and DCGANs when all the models are trained on MNIST dataset. The images by GANs are significantly sharper as compared to RBMs and VAEs. With simple implementation and striking results, GANs have caught the attention of the entire research community. RMBs or VAEs fail to produce complex images when trained on other datasets such as CIFAR, SVHN, etc. On the other hand, GANs generate far better images on these datasets too. How does a GAN achieve this?

The typical architecture of a GAN consists of two different neural networks: (i) a *generator* and (ii) a *Discriminator* (Figure 2). An input to the Generator is a low dimensional noise vector. It transforms the noise into a data vector which forms a potential data sample. The Discriminator takes this data vector as input and assigns it a score based on how likely the data vector is from the original data distribution. The data, sampled from both the real distribution and from the generator, is used to train the discriminator. Based on the score the generator learns how to produce vectors such that the Discriminator is confused. In practice, each training iteration consists of one step of discriminator training followed by one step of generator training. We observe that the real data distribution is not explicitly learned but we can sample data from it via the trained generator. As mentioned in *Mohamed et. al.* [16], it follows the likelihood-free



(a) RBM Generated MNIST

(b) VAE Learned Frey Face Manifold

(c) VAE MNIST Manifold

Figure 1: Generative reults on MNIST from the RBM, VAE and DCGAN paper to compare the quality of the results generated.

inference approach. The paper [16] shows that in a GAN the probability density function is estimated using density comparison techniques. The working of GANs is not as effortless as it may appear. One of the primary challenges with GANs is ensuring high resolution and diversity in the generated images (mode collapse).

Apart from the above two difficulties, GANs also face problems such as convergence, stability, etc while training the models. To address such issues, researchers are striving towards building a fundamental model of GANs. Besides, researchers are also trying to obtain rigorous theoretical justification for these models; especially for the issues related to stability and convergence. The primary challenge encountered when trying to solve any problem, is of designing the appropriate loss function which guarantees the optimal result. The next, equally significant issue is of optimizing over the defined loss. Ideally, one aims to design a loss function with right properties such that the optimization algorithms can avoid bad local minima and guarantee a polynomial-time convergence.

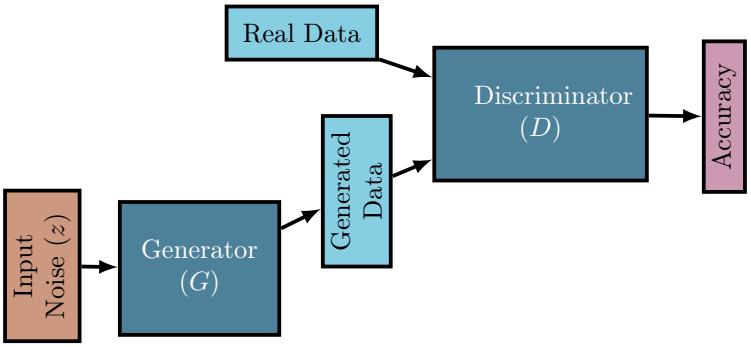


Figure 2: GANs Block Diagram

A related work by Hitawala [10] also presents a comparative study of models with various modifications over vanilla GANs. Yet the author does not focus on the issues related to the model and its training. Hence the models he choses to discuss about are entirely different from ours. In this article, we start with a technical description of GANs. Then we explain different challenges in implementing them. We

classify these challenges into four categories:

1. Mode Collapse
2. Vanishing Gradients
3. Non-Convergence
4. Existence of Equilibrium and Generalization.

For each problem, we explain the approach followed in some of the papers to resolve it. We provide a summary of collected research articles [1] which examine different aspects of GANs as well as take a step further towards understanding the theory behind GANs to improve its performance. We summarize all the discussion in Table 2 for a quick overview. We believe such challenge based classification of GAN related theory papers is novel and such study should be useful for a researcher working on issues in GANs.

2 Generative Adversarial Networks

In this section, we provide technical details of GANs. We elaborate the objective function used in the training of a GAN. We also discuss it's interpretation as a divergence minimization problem as well as a two-player zero-sum game. Finally, we write about the challenges known to be present in these models. The following table lists the notations that we will be using henceforth.

Symbol	Parameter
D	Discriminator
G	Generator
D_θ	D parameterized by θ
G_ϕ	G parameterized by ϕ
p_{data}	Data distribution
p_g	Model(G) distribution
$p_z(z)$	Random noise distribution
V	Utility function for D
U	Utility function for G

Table 1: **Notation**

2.1 Success Stories

The original paper by Goodfellow et al. [7] motivates the learning of a data distribution without requiring any explicit modeling or approximation of it. The problem is posed as a two player zero-sum game,

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))] \tag{1}$$

It is shown that for $p_g = p_{data}$, the above game has a global optima. Simultaneous gradient descent method is used for optimizing the objective. The convergence of the algorithm to global optimum is guaranteed if both the networks are given enough capacity and discriminator is trained to optimality for a fixed G. Additional assumption is that, D and G are convex w.r.t the parameters.

The optimal discriminator for a fixed generator is given by $D^*_G(x) = \frac{p_{data}(x)}{p_{data}(x)+p_g(x)}$. For the optimal discriminator as given above, the generator is shown to minimize the following Jenson Shannon Divergence(JSD),

$$C(G) = -\log(4) + KL\left(p_{data} \parallel \frac{p_{data}+p_g}{2}\right) + KL\left(p_g \parallel \frac{p_{data}+p_g}{2}\right) \tag{2}$$

$$C(G) = -\log(4) + 2.JSD(p_{data} \parallel p_g) \tag{3}$$

Other generative models learn by maximizing the log-likelihood which is same as minimizing KL divergence, i.e., $KL(P \parallel Q)$. KL divergence is not symmetric, and few others claim reverse KL divergence, i.e., $KL(Q \parallel P)$ is the metric that should be ideally minimized. JSD, however, is symmetric and includes

[1] *Disclaimer: The list is not exhaustive. The papers which we are describing are chosen based on our understanding of the importance of the contribution in the article. We might have missed relevant papers.*

both the divergences, hence likely to perform better than the above approaches. Another major deviation introduced in the loss is that rather than training G to minimize $\log(1-D(G(z)))$, it is trained as follows.

$$\max_{G_\theta} \log(D(G(z))) \tag{4}$$

This is to provide better gradients early in training.

The other significant analogy for GAN is derived from game theory. Two-player zero-sum games as defined in game theory are strategic games such that the utilities of both players sum to zero. It is a strictly competitive game where maximizing one's utility will minimize the opponent's utility. In a typical GAN setting G and D form the two players, while θ and ϕ form their respective strategy spaces. From Equation 1 we get the utility of the D which is V. The corresponding utility for G is $-V$. Such games are said to converge to a **Nash Equilibrium** [18] at optimality. Under such an equilibrium no player is better off by unilaterally deviating from her strategy. If each players equilibrium strategy is a best response to the equilibrium strategy of the opponent, then such strategies form the pure strategy Nash equilibrium. It is believed that training of GANs also converges to a pure strategy Nash equilibrium where both G and D have fixed, trained parameters.

The follow-up research has vastly improvised upon the quality of results produced from vanilla GANs. Alec et al. DCGAN [22] have proposed a stable architecture and suitable values for the hyper-parameters for better training. Besides extensions of the adversarial training has led to its adaptation to other fields like text prediction, conversion of an image to text and vice versa, style transfer from one domain to another, image inpainting, etc. In the process, different variants of GANs have been designed, like conditional GANS, super-resolution GANs, etc. so much that this trend has earned a fair share of ridicule. Is Vanilla GAN or DCGAN solving the data generation problem? The answer is no. In practice, there are multiple challenges which we describe in the next subsection.

2.2 Challenges

Despite its popularity, there are major issues with quality and variety of the data generated. The fundamental cause leading to these issues is unknown. To broadly classify there are two primary challenges, first being the inappropriate consequences of the loss function as originally defined. Second is the difficulty of optimizing a non-convex error function once defined. Many papers in the past year have attempted to articulate the problems precisely. In this section, we summarize the main idea in few of those papers that we felt were significant.

2.2.1 Mode Collapse

The most significant and widely discussed problem is *Mode Collapse*. Mode collapse seems to arise as a direct consequence of the way the adversarial loss is defined. When many input noise values are mapped to the same data point, the generator lacks diversity in the samples it can generate. Primarily it could be seen as an issue of under-fitting. The problem of mode collapse has been studied in varied contexts.

2.2.2 Vanishing Gradients

The other acute problem of *vanishing gradients* is related to the assumption of infinite capacity of the models. This happens because the data distribution and model distribution are in a non-overlapping lower dimensional manifold. In such a case the discriminator can be trained to perfection leading to vanishing gradients. Under the assumption of infinite capacity of the models, the JSD saturates.

2.2.3 Non Convergence

While performing simultaneous gradient descent for training the generator and discriminator, it is tricky to decide upon the number of iterations for which the generator and discriminator have to be trained. Improper balance of the number of iterations results in *Undamped Oscillations*. Ideally training the discriminator to optimality between every generator updates not only is computationally expensive but also results in a pessimistic discriminator leading to the problem of vanishing gradients. The convergence of the GAN game as proved in the original paper Goodfellow et al. [7] holds true if the function is convex, the model has infinite capacity, and enough training samples are available. These assumptions are not valid in practice. Non-convergence of the simultaneous gradient descent has been discussed in many papers. The convergence is not guaranteed sometimes even in convex cases like $xy = 0$.

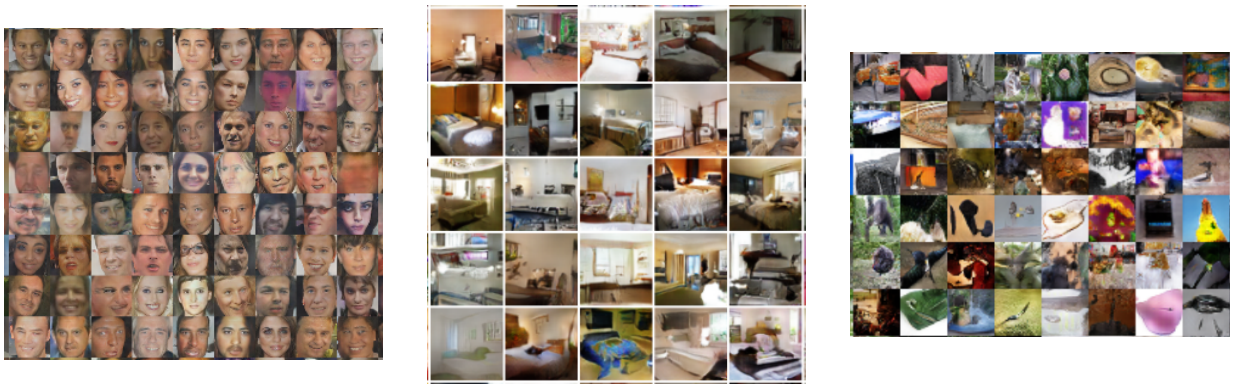


Figure 3: GANs used to generate complex images Images from DCGAN and WGAN paper. Gans do not work as well on High resolution Imagenet dataset.

2.2.4 Existence of Equilibrium

The GAN game is modeled as a two-player zero-sum game on continuous strategy space. Guarantees on the existence of Unique Pure strategy Nash Equilibrium in a continuous strategy space is also an important question to be asked.

2.2.5 Evaluation of Generative Models

The foremost challenge is that there is no precise and unique quantitative or qualitative measure for evaluating the generated images. The generative images in the original paper [7] use Parzen Window estimates of the model's log-likelihood. The evaluation is also compared based on the model's performance on some surrogate tasks like classification, de-noising or missing value imputation. The most widely accepted measure, for now, is the Inception score, proposed by Salimans et al. [24]. [4] is a good survey discussing the pros and cons of the evaluation metrics. Ultimately, it is desirable to have a metric that evaluates both diversity and visual fidelity simultaneously.

3 Addressing the challenges

In this section, we summarize few approaches followed in different papers to tackle some of the challenges discussed above.

3.1 Mode Collapse

Mode collapse originally mentioned in the GAN paper as "the Helvetica scenario," occurs when the generator collapses too many values of (z) noise values to one value of (x) real data, hence losing on the diversity of data generated. This shortcoming of using a vanilla GAN architecture is more of a form of under-fitting to the training data. The following papers discuss this problem at various levels.

On Distinguishability Criteria for Estimating Generative Models

Ideally in the GAN objective given by Equation 1, G should minimize the loss once D is trained to optimality on the present parameters of G. Training D likewise at every iteration is computationally expensive, hence by using simultaneous gradient descent, we approximate to the optimal. It was raised as a concern in the subsequent paper by Goodfellow [6]; he states that in non-convex setting, the alternative gradient descent may result in minimization of a lower bound. It is also referred to as the cause for under fitting observed in GANs.

Improves Techniques for Training GANs

This paper [24] Salimans *et al.* presents certain heuristics for overcoming the problem of mode collapse. They introduce the technique of *Mini batch discrimination*. A discriminator looking at multiple examples in combination, could potentially help avoid collapse of the generator. The following models the closeness between examples in a mini-batch. $f(x_i) \in \mathbb{R}^A$ denote a vector of features for input x_i produces by some

intermediate layer in the discriminator. $f(x_i)$ is multiplied by a tensor $T \in \mathbb{R}^{\mathbb{A}\times\mathbb{B}\times\mathbb{C}}$, which results in a matrix $M_i \in \mathbb{R}^{B\times C}$.

$$c_b(x_i, x_j) = exp(- \parallel M_{i,b} - M_{j,b} \parallel_{L_1}) \in \mathbb{R}$$

$$o(x_i)_b = \sum_{j=1}^{n} c_b(x_i, x_j) \in \mathbb{R},\ o(x_i) = [o(x_i)_1, o(x_i)_2, \ldots, o(x_i)_B] \in \mathbb{R}^B,\ o(X) \in \mathbb{R}^{n\times B}$$

The $o(x_i)$ is concatenated with $f(x_i)$ and fed to the next layer of the discriminator.

Unrolled GANs

The paper [15] Metz *et. al.* directly addresses the issue of mode collapse. The authors suggest a new architecture altogether for overcoming this problem. If one agent becomes more powerful than the other, the learning signal becomes useless. For a minimax loss as given in original paper, the optimal discriminator $D^*(x)$ is a known smooth function of the generator probability $p_G(x)$. These smoothness guarantees are lost when $D(x;\theta_D)$ and $G(x;\theta_G)$ are drawn from parametric families. Explicitly solving for the optimal discriminator parameters $\theta_D^*(\theta_G)$ for every update step of the generator G is computationally infeasible. As a result GAN training suffers from mode collapse. A surrogate loss function $f_K(\theta_G, \theta_D)$ is introduced for training the generator which more closely resembles the true generative objective $f(\theta_G, \theta^* D(\theta_G))$. $K = 0$ (Normal Gan loss), $K \to \infty$ (True generative objective function). The gradient updates:

$$\theta_G \leftarrow \theta_G - \eta \frac{df_K(\theta_G, \theta_D)}{d\theta_G},\ \theta_D \leftarrow \theta_D + \eta \frac{df(\theta_G, \theta_D)}{d\theta_D}$$

$$\frac{df_K(\theta_G, \theta_G)}{d\theta_G} = \frac{\partial f(\theta_G, \theta_D^K(\theta_G, \theta_D))}{\partial \theta_G} + \frac{\partial f(\theta_G, \theta_D^K(\theta_G, \theta_D))}{\partial \theta_D^K(\theta_G, \theta_D)} \frac{d\theta_D^K(\theta_G, \theta_D)}{d\theta_G}$$

In a standard GAN the G tries to move as much mass to a single point that maximizes the ratio of the probability density. The D tracks the point and assigns lower probability to it and uniform elsewhere. This cycle will repeat forever. In this paper, however, using the surrogate loss, G's update takes into account the response of D before hand. This helps G to spread it's mass making the next D step less effective instead of collapsing to a point.

Towards Principled Methods for training GANs

Arjovsky *et al.*[1], study the form of gradients for the modified GAN objective which is mentioned in Equation 4. The gradients for an optimal discriminator $D^* = \frac{P_r}{P_{g_{\theta_0}} + P_r}$, fixed for a value θ_0,

$$\mathbb{E}_{z\sim p(z)}[-\nabla_\theta \log D^*(g_\theta(z))|_{\theta=\theta_0}] = \nabla_\theta[KL(\mathbb{P}_{g_\theta} \parallel \mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta} \parallel \mathbb{P}_r)]|_{\theta=\theta_0}$$

When re-written as above, the JSD is in opposite direction, pushing for the distributions to be different. The inverted KL is not maximum likelihood, instead it assigns extremely high cost to generating fake looking samples, and extremely low cost to mode dropping.

On Convergence and Stability of GANs

The authors in [12] view GANs objective as Regret minimization as opposed to divergence minimization. They make a connection between no regret algorithms and alternating SGD and prove it's convergence in convex-concave case. They introduce an additional term in the objective which they refer to as local smoothing. According to their findings mode collapse is often accompanied by the discriminator function having sharp gradients around some real data points. Hence they introduce the following penalty term in the overall GAN loss,

$$\lambda.\mathbb{E}_{x\sim P_{real}, \delta\sim N_d(0,cI)}[\parallel \nabla_x D_\theta(x+\delta) \parallel - k]^2$$

Mode Regularized GAN

Che *et al.* [5] give an intuition behind the problem of missing modes and also propose regularizers to circumvent this problem. It is usually the case that the data and the model distribution manifolds are disjoint. In such a case, the discriminator assigns zero probability to all the model points and one probability to all the data points. Thus, large modes usually have a much higher chance of attracting the gradient of the discriminator. For a typical GAN model, since all modes have similar D values, there

is no reason why the generator cannot collapse to just a few major modes. For most z the gradient of the generator pushes the generator distribution towards the major mode. It is highly unlikely to have z which is close to the other minor modes, hence the problem of missing modes.

- Geometric Metric Regularizer - Having another similarity metric such as L_2-norm with nice geometric properties, in addition to the gradient information from the discriminator. Together with the G, they also have an Encoder $E(x) : X \to Z$. Assuming d to be some similarity metric in the data space, the authors add the following term to the loss as a regularizer,

  $$\mathbb{E}_{x \sim p_d}[d(x, G \circ E(x))]$$

  The encoder is trained by minimizing the reconstruction error.

- Mode Regularizer - This is proposed to penalize the missing modes.
  - The areas near the missing modes are rarely visited by the G
  - Both missing modes and non-missing modes correspond to high values of D.

  Consider a minor mode M_0. For $x \in M_0$, $G(E(x))$ will be located close to the mode M_0. They add the following to the loss,

  $$\mathbb{E}_{x \sim p_d}[\log D(G \circ E(x))]$$

The overall loss for G is given by,

$$T_G = -\mathbb{E}_z[\log D(G(z))] + \mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))]$$

The overall loss for E is given by,

$$T_E = \mathbb{E}_{x \sim p_d}[\lambda_1 d(x, G \circ E(x)) + \lambda_2 \log D(G \circ E(x))]$$

Energy-Based GAN

Another encoder-decoder based approach for training GAN was put forth in the Energy-based GAN paper by [25] Zhao *et. al.*. The paper views the discriminator as an energy function, which assigns low energy values to real data and high to fake data. The generator is a trainable parameterized function that produces samples in regions to which the discriminator assigns low energy. The objective function is given by,

$$\mathcal{L}_D(x, z) = D(x) + [m - D(G(z))]^+$$

$$\mathcal{L}_G(z) = D(G(z))$$

where, $[.]^+ = max(0, .)$; m - positive margin; $\mathcal{L}_D$ - discriminator loss; $\mathcal{L}_G$ - generator loss The discriminator is modeled as an auto-encoder

$$D(x) = \parallel Dec(Enc(x)) - x \parallel$$

With the binary logistic loss, only two targets are possible, so within a minibatch, the gradients corresponding to different samples are most likely far from orthogonal. This leads to inefficient training, and reducing the minibatch sizes is often not an option on current hardware. According to the paper, the reconstruction loss introduced will likely produce very different gradient directions within the minibatch, allowing for larger minibatch size without loss of efficiency. When an EBGAN auto-encoding model is trained to reconstruct a real sample, the discriminator contributes to discovering the data manifold by itself without the need for explicit negative samples. To prevent the auto-encoder from learning identity function, the framework is regularized with the generator producing contrastive samples. A Repelling regularizer is introduced to prevent mode collapse used only with the generator loss, $S \in \mathbb{R}^{s \times N}$ where S is a batch of sample representations taken from encoder output layer.

$$f_{PT}(S) = \frac{1}{N(N-1)} \sum_i \sum_{j \neq i} \left(\frac{S_i^T S_j}{\parallel S_i \parallel \parallel S_j \parallel} \right)^2$$

The PT term attempts to orthogonalize the pairwise sample representations.

3.2 Non convergence or Non stability

Simultaneous Gradient Descent is used for training both G and D. This way of optimization is not guaranteed to converge to the saddle point of the GAN game. Considering the actual surface is non-convex and at each step, we get an approximation to the perfect discriminator. Hence the training curve is not smooth and, its convergence properties are not clear.

On Distinguishability Criteria for Estimating Generative Models

[6] addresses the non-convergence of GANs and their departure from MLE. As explained in the paper, the asymptotic convergence of GANs is guaranteed in a convex function space. In non-convex setting, the alternative gradient descent may result in minimization of a lower bound. It could make the bound looser rather than minimizing the underlying objective, which results in oscillations. He draws primary differences between GANs and another generative model based on Noise Contrastive Estimation(NCE). They illustrate how the expected gradient of the NCE can be realized to match with the expected gradient of the MLE if one is allowed to use a non-stationary noise distribution for NCE (Self Contrastive Estimation). No choice of discriminator network can make the expected gradient for the GAN Generator match that of MLE.

Improved Techniques for Training GANs

As mentioned in the previous Subsection, 3.1 [24] presents certain heuristics for the better convergence of GAN game. The primary issue that it points out in the paper is about the difficulty in finding a Nash equilibrium, where the cost function is non-convex, the parameter space is continuous and high dimensional. The next issue is about the non-convergence of simultaneous gradient descent even in simple cases. Like the algorithm enters a stable orbit instead of global optima when applied on a cost function like $xy = 0$. The primary technique mentioned is historical averaging. In which each player's cost is modified by including a term $\| \theta - \frac{1}{t}\sum_{i=1}^{t}\theta[i] \|^2$, where $\theta[i]$ is the values of parameters at a past time i. This approach is inspired by **fictitious play** algorithm that can find equilibrium in all kinds of games.

fGAN: Training Generative Neural Samplers using Variational Divergence Minimization

Couple of years after the original GAN, [20] Nowozin et al. generalized the idea of generative models which use probabilistic feed forward neural networks. They call it generative neural samplers. They even generalized the notion of statistical divergences which measure the distances between two distributions. Given two distributions P and Q, they define $f-$divergence,

$$D_f(P \parallel Q) = \int_{\mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

where the generator function $f : \mathbb{R}_+ \to \mathbb{R}$ is a lower semi continuous function satisfying $f(1) = 0$. They also mention the variational lower bound of the $f-$divergences. f^* defined as follows,
Fenchel Conjugate

$$f^*(t) = \sup_{u \in dom_f} \{ut - f(u)\}$$

Variational Divergence Minimization is the new method they suggest for estimating the paramters of the model Q_θ. Given that $\psi : \mathcal{X} \to \mathbb{R}$ is the variational function parameterized by w, the f-GAN objective is given by,

$$F(\theta, w) = \mathbb{E}_{x\sim P}[\psi_w(x)] - \mathbb{E}_{x\sim Q_\theta}[f^*(\psi_w(x))] \tag{5}$$

It is shown that GAN objective is a particular instance of the above loss function. They propose a single-step gradient descent algorithm and prove its convergence to the saddle point if there is a neighborhood around it in which F is strongly convex in θ and strongly concave in w .

Stabilizing Training of GANs through Regularization

Roth *et al.* [23] is a follow up on the above f-gan paper. They claim that the fragility of gan training is due to non-overlapping model distribution and data distribution manifolds in the high dimensional space, which is termed as dimensionality misspecification. $f-$GAN models fail under such conditions. Usually, such issue is taken care by adding high-dimensional noise, which introduces significant variance in

the parameter estimation hence making the solution impractical. Instead, the authors propose analytic convolution of the densities with the Gaussian noise which yields a weighted penalty function on the norm of the gradients w.r.t the input. The following noise induced regularization results in a stable GAN training procedure.

- Noise Induced Regularization:
  f-Gan Objective as in Equation 5: $F(\mathbb{P}, \mathbb{Q}; \psi) = \mathbf{E}_{\mathbb{P}}[\psi] - \mathbf{E}_{\mathbb{Q}}[f^c \circ \psi]$
  Noise convolution, adding white noise $\xi \sim \Lambda = \mathcal{N}(0, \gamma I)$ to samples $x \sim \mathbb{P}, \mathbb{Q}$:

$$\mathbf{E}_{\mathbb{P}}\mathbf{E}_{\Lambda}[\psi(x+\xi)] = \int \psi(x) \int p(x-\xi)\lambda(\xi)d\xi dx = \int \psi(x)(p * \lambda)(x)dx = \mathbf{E}_{\mathbb{P}*\Lambda}[\psi]$$

  p and λ are probability densities of $\mathbb{P}$ and Λ, $\lambda(x) > 0$ and $(p * \lambda)(x) > 0$ $(\forall x)$.
  Regularized f-GAN

$$F_\gamma(\mathbb{P}, \mathbb{Q}; \psi) = \mathbf{E}_{\mathbb{P}}[\psi] - \mathbf{E}_{\mathbb{Q}}[f^c \circ \psi] - \frac{\gamma}{2}\Omega_f(\mathbb{Q}; \psi)$$

$$\Omega_f(\mathbb{Q}; \psi) := \mathbf{E}_{\mathbb{Q}}[(f^{c''} \circ \psi) \parallel \nabla\psi \parallel^2]$$

The Numerics of GANS

In the paper Mescheder et al. [14], the authors identify the main reason for non-convergence of GANs to local Nash equilibria. Let $\bar{x} = (\bar{\phi}, \bar{\theta})$ be a point of Nash equilibrium given by,

$$\bar{\phi} \in \underset{\phi}{argmax}\ f(\phi, \bar{\theta}) \quad \text{and} \quad \bar{\theta} \in \underset{\theta}{argmax}\ f(\bar{\phi}, \theta)$$

. Every differentiable two-player game defines a vector field $v(\phi, \theta) = \begin{bmatrix} \nabla_\phi f(\phi, \theta) \\ \nabla_\theta g(\phi, \theta) \end{bmatrix}$. $\bar{x}$ is a stationary point of $v(x)$ and $v'(\bar{x})$ is negative semidefinite iff $\bar{x}$ is a local Nash equilibrium. $v'(\bar{x})$ has eigen values with small real part and big imaginary part which results in slow convergence. This is in particular a problem of simultaneous gradient ascent for two-player games (in contrast to gradient ascent for local optimization), where the Jacobian $v'(x)$ is not symmetric and can therefore have non-real eigenvalues. Finding a stationary field is equivalent to solving the equation $v(x) = 0$. They define $L(x) = \frac{1}{2} \parallel v(x) \parallel^2$. Minimizing $L(x)$ directly leads to unstable stationary points, hence they consider a modified vector field $w(x) = v(x) - \gamma\nabla L(x)$ for some $\gamma > 0$. The modified utility functions for the two player game is now,

$$\hat{f}(\phi, \theta) = f(\phi, \theta) - \gamma L(\phi, \theta) \quad and \quad \hat{g}(\phi, \theta) = g(\phi, \theta) - \gamma L(\phi, \theta)$$

The $L(\phi, \theta)$ term encourages agreement between the two players, hence is called *Consensus Optimization.*

Gradient Descent GAN Optimization is Locally Stable

Another paper [17], which is a follow-up work on the previous paper. They also suggest the addition of a regularization term on the norm of the discriminant gradient. Besides they establish that under suitable conditions GAN optimization is locally exponentially stable. WGAN although can perennially cycle around an equilibrium point without converging. The regularization that they propose enhances the local stability of the optimization procedure, for any general gan framework. They suggest the following update of Generator,

$$\theta_G := \theta_G - \alpha\nabla_{\theta_G}(V(D_{\theta_D}, G_{\theta_G})) + \eta \parallel \nabla_{\theta_D} V(D_{\theta_D}, G_{\theta_G}) \parallel$$

3.3 Vanishing Gradient

As mentioned in Subsection 3.1 mode regularized GAN, the main reason for mode collapse is the problem of vanishing gradients. It refers to cases where the discriminator perfectly classifies real and fake examples, such that around fake examples, D is nearly zero. In such cases, the generator will receive no gradient to improve itself.

Towards Principled Methods for Training GANs

This issue is raised in the paper [1]. The show that on training the Discriminator till convergence, the error $2\log 2 - 2JSD(\mathbb{P}_r \parallel \mathbb{P}_g)$ goes to 0. This points to the fact that the JSD is maxed out which happens when the distributions have disjoint supports. As $\mathbb{P}_r$ and $\mathbb{P}_g$ exist in lower dimensional manifolds, we can have a smooth optimal discriminator that has accuracy one. It results in zero gradients everywhere. They also prove, why the generator gradient vanishes as the discriminator gets better when training the generator using $\mathbf{E}_{z\in p(z)}[\log(1 - D(g_\theta(z)))]$. Hence we use $\mathbf{E}_{z\in p(z)}[-\log D(g_\theta(z))]$ but which again leads to unstable updates as mentioned in the Section 3.1.

One of the ways for fixing the instability is by adding continuous noise to the inputs of the discriminator, therefore smoothening the distribution of the probability mass. If the noise ϵ used is $\mathcal{N}(0, \sigma^2 I)$, the gradients of the generator have the following form,

$$\mathbb{E}_{z\sim p(z)}[\nabla_\theta log(1 - D^*(g_\theta(z)))] = \mathbb{E}_{z\sim p(z)}\Bigg[a(z)\int_{\mathcal{M}} P_\epsilon(g_\theta(z) - y)\nabla_\theta \parallel g_\theta(z) - y \parallel^2 d\mathbb{P}_r(y)$$

$$-b(z)\int_{\mathcal{P}} P_\epsilon(g_\theta(z) - y)\nabla_\theta \parallel g_\theta(z) - y \parallel^2 d\mathbb{P}_g(y)\Bigg]$$

where $a(z)$ and $b(z)$ are positive functions. Furthermore, $b > a$ iff $P_{r+\epsilon} > P_{g+\epsilon}$, and $b < a$ iff $P_{r+\epsilon} < P_{g+\epsilon}$. This theorem proves that we will drive our samples $g_\theta(z)$ towards points along the data manifold, weighted by their probability and the distance from our samples. Furthermore, the second term drives our points away from high probability samples, again, weighted by the sample manifold and distance to these samples. This is similar in spirit to contrastive divergence, where we lower the free energy of our samples and increase the free energy of data points.

Mode Regularized GAN

As mentioned in the previous Subsection 3.1 [5] claim that, if the generated data and the real data come from the same low dimensional manifold, the discriminator can help the generator distribute its probability mass. This is because the gradients of the discriminator around the fake samples is near zero. Hence in addition to the Mode regulaizer they also have a metric regularizer. It is supposed to have nice geometric properties, to enable proper gradients in addition to the gradient information from the discriminator. Together with the G, they also have an Encoder $E(x) : X \to Z$. Assuming d to be some similarity metric in the data space, the add the following term to the loss as a regularizer,

$$\mathbb{E}_{x\sim p_d}[d(x, G \circ E(x))]$$

The encoder is trained by minimizing the reconstruction error.

Wasserstein GAN

In [2], the authors elaborate on another suggestion for overcoming the instability is by using a different distance metric altogether. The Earth Mover distance can be useful for learning distributions in lower dimensional manifold. The EM distance or Wasserstein distance is given by, $W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\parallel x - y \parallel]$. $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. The Wasserstein distance is much weaker distance and is continuous loss function in θ if g is continuous in θ. The infimum is highly intractable we use Kantorovich-Rubinstein duality to transform the objective into, $W(\mathbb{P}_r, \mathbb{P}_\theta) = \sup_{\|f\|_L\leq 1} \mathbb{E}_{x\sim\mathbb{P}_r}[f(x)] - \mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)]$, where the supremum is over all the 1-Lipschitz functions. Thus if we have a parameterized family of 1-Lipschitz functions $\{f_w\}_{w\in\mathcal{W}}$, we solve the following problem,

$$\max_{w\in\mathcal{W}} \mathbb{E}_{x\sim\mathbb{P}_r}[f_w(x)] - \mathbb{E}_{z\sim\mathbb{P}_z}[f_w(g_\theta(z))]$$

The f_w can be modeled as a neural network, where the fact that f is 1-Lipschitz depends on $\mathcal{W}$ being compact. One way of enforcing the compactness is to clamp the weights to a fixed box. Using the above objective waives the need for balancing the generator and discriminator. In this case discriminator referred to as the critic could be trained till optimality without losing gradients.

Improved Training of WGANs

The problem of exploding or vanishing gradients may resurface even in a WGAN setting, because of the use of weight clipping to enforce Lipschitz constraints. The subsequent paper after WGAN, Guljarani *et. al.* [8] addresses the issues related to weight clipping. Apparently, weight clipping leads to capacity under use, i.e., the critic is biased towards much simpler functions. They introduce an alternative way of maintaining the Lipschitz constraints by introducing a gradient penalty term. They prove that the optimal critic has unit norm gradients everywhere, hence their penalty term constrains the gradients of the critic to be 1,

$$L = \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] + \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\parallel \nabla_{\hat{x}} D(\hat{x}) \parallel_2 -1)^2]$$

$\mathbb{P}_{\hat{x}}$ sampling uniformly along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$. The authors claim an increase in sample quality and training speed.

3.4 Generlizability and Existence of Equilibrium

In the paper generalization and equilibrium of GANs by Arora et al. [3], the authors question the generalization of GAN objective as well as the existence of pure equilibrium in the two-player game. Generalization in GANs as defined by the authors means that the population distance between the true and the generated distribution is close to the empirical distance between the empirical distribution.

$$|d(\mathcal{D}_{real}, \mathcal{D}_G) - d(\hat{\mathcal{D}}_{real}, \hat{\mathcal{D}}_G)| \leq \epsilon$$

where $\hat{\mathcal{D}}_{real}$ is the empirical version of $\mathcal{D}_{real}$ with polynomial number of samples. They prove that Jenson Shanon Divergence and Wasserstein distance do not generalize with a polynomial number of examples. Further analysis show that GANs actually minimize a surrogate distance called the Neural Network distance,

Definition 1 *Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^d$ to $[0,1]$ such that if $f \in \mathcal{F}, 1-f \in \mathcal{F}$. Let ϕ be a concave measuring function. Then the $\mathcal{F}$-divergence with respect to ϕ between two distributions μ and ν supported on $\mathcal{R}^d$ is defined as*

$$d_{\mathcal{F},\phi}(\mu,\nu) = \sup_{D\in\mathcal{F}} \mathop{\mathbb{E}}_{x\sim\mu}[\phi(D(x))] + \mathop{\mathbb{E}}_{x\sim\nu}[\phi(1-D(x))] - 2\phi(1/2)$$

The major theorem stated in the paper claims that, since there are not infinitely many discriminators, given enough samples the expectation over the empirical distribution converges to the expectation over the true distribution for all discriminators. Although this analysis guarantees generalization, the assumption of finite discriminators results in lack of diversity in the generated distribution. For JS and Wasserstein distance, when the distance between two distributions μ, ν is small would imply that the distributions are close. However $d_{NN}(\mu,\nu)$ can be small even if the distributions are not close. A neural network with p parameters cannot distinguish between a distribution μ and distribution with support $\tilde{O}(p/\epsilon^2)$. Such a limited capacity network cannot learn the distribution although it has access to a lot of samples from the distribution μ.

The other important aspect that the paper addresses is the existence of equilibrium. Although it is unknown as to what equilibrium GAN converges, the author proves the existence of a particular equilibrium. The main motivation is obtained from the min-max theorem [19] which shows that if both players are allowed to play mixed strategies, then the game has an equilibrium which is the min-max solution. The paper models mixed strategies by considering a mixture of generators. As an infinite mixture is not possible; it admits an approximate solution with a finite mixture of generators. Thus they show the existence of $\epsilon-$approximate equilibrium.

4 Discussion and Conclusion

The main challenges faced by GANs are as a consequence of our lack of precise knowledge of how neural network work. Moreover, the presence of simultaneous interaction of two models has made the training even more unstable. The problems of Mode collapse, Vanishing Gradients, Non-convergence have been studied extensively in the last couple of years, yet there is still no concrete solution or cause for them. The theoretical results have been able to give few insights although none of them have a significant effect on the results. The following table summarizes the contributions in few of the papers. It focuses on the problems that each of the articles discussed here and identify the novel approach they propose.

Table 2: **Summary of GAN Papers**

Date	Paper Name	Problems Raised	Novel approach
May 2015	On Distinguish-ability Criteria For Estimating Generative Models [6]	The expected gradient of generator does not match that of MLE. Non convergence of independent SGD results in under-fitting in gans	In gans G being the primary model causes it to differ from MLE. No close relation between NCE and GAN.
June 2016	Improved Techniques for GANs [24]	Overtraining of the discriminator Mode collapse of Generator Gradient descent may not converge Vulnerable to adversarial examples. GAN outputs depend on the inputs within a same batch due to BN	Feature Matching Mini-batch Discrimination Historical Averaging (Fictitious play) Label-smoothing Virtual Batch normalization
Jan 2017	Towards Principled Methods for Training GANs [1]	Perfect Discriminator resulting in zero grads when distributions are in dimensional manifolds The $-\log D$ alternative causes-unstable updates	Softer Metrics- Adding Gaussian Noise (Contrastive Divergence) Earth-Mover Distance
Mar 2017	WGAN [2]	Learning distributions supported in lower dimension manifolds. JSD maxes and $KL \to \infty$ Mode Collapse	Define EM Distance (KR Duality) with well defined gradients and avoids balancing the critic and generator. Consistent decrease in loss with training. Weakest metric
May 2017	Improved Techniques on WGAN [8]	Lipschitz enforced by weight clipping. Capacity underuse (critic has high bias), Exploding and vanishing gradients.	Introduced a Gradient Penalty term. wrt sample as following was observed. Optimal critic has unit-grad norm between $\mathbb{P}_r$ and $\mathbb{P}_g$
Mar 2017	Loss-Sensitive GAN on Lipschitz Densities[21]	Over-pessimistic D as fake samples closer to data are still negative Assuming infinite capacity which leads to mode collapse and vanishing grads WGAN is unbounded from above as critic is minimized over gen samples	Loss having a data-dependent margin. Limited to the space of Lipschitz-continuous functions Pairwise comparisons as against WGAN which decomposes the loss into two first order moments
ICLR 2017	EBGAN [25]	With binary logistic loss only two targets are possible, so gradients in a mini batch are far from orthogonal. Mode Collapse.	Auto-encoder Discriminator to evade the need of negative samples Repelling regularizer to prevent mode collapse
Feb 2017	Least Squares GAN [13]	Vanishing gradients with sigmoid cross-entropy loss when updating the gen using fake samples that are on the correct side of decision boundary.	Pearson χ^2 Divergence. The following constants , a- encoding for real data b - encoding for fake data c- encoding for value that G wants D to believe is fake.
June 2016	f-GAN[20]	No convergence to saddle point when using single step GD	General f-divergence variational estimation of $f-$ divergence Variational Divergence Minimization
Nov 2017	Stabilizing Training of[23] GANs through Regularization	Empirical estimation Density misspecification Dimensional misspecification Addition of high dim noise introduces huge variance in parameter estimation	Convolving with noise Noise induced regularization
NIPS 2017	The Numerics of GANs [14]	Failure of simultaneous gradient descent as the eigenvalues gradients have zero real part and large imaginary part	Consensus Optimization: Alternative method for finding the Nash Equilibrium Introduced norm of the gradients w.r.t parameters in the loss.

Nov 2016	Unrolled GANs [15]	Mode collapse, Oscillations of G and D. G tries to move much mass to single point, D tracks it and assigns lower probability The cycle continues forever.	Training the D to optimality is expensive.A surrogative loss which in limit equals optimal D The G's updates consider future steps of the discriminator.
Aug 2017	Generalization and Equilibrium in GANs [3]	Generalization is not guaranteed i.e. $d(\mathcal{D}_{real}, \mathcal{D}) = 0$ but $d(\hat{\mathcal{D}}_{real}, \hat{\mathcal{D}}) \neq 0$ where $\hat{D}$ is the empirical distribution Non Existence of equilibrium in GAN	NN Distance which generalizes as it assumes finite capacity D At the cost of diversity of samples. Mixed Strategy Nash using mixture of Generators which on folding results in pure NE.
May 2017	On Convergence and Stability in GANs [12]	Gradient descent is unstable and results in mode collapse Divergence minimization hypothesis doesn't explain gan learning swiss roll distribution WGAN and LS-GAN regularize the discriminator's gradients in domain space.	Regret Minimization converges to ϵ-equilibrium in non-convex case. Sharp gradients results in mode collapse hence a penalty term is introduced.
Nov 2017	Gradient Descent GAN Optimization is locally stable	It is assumed that updates occur in functional space, with models having infinite capacity. Discriminator is assumed to be fully optimized between gen updates.	local asymptotic stability of GAN (not WGAN) optimization under proper conditions, gradient-based regularizer(GAN + WGAN)

References

[1] M. Arjovsky and L. Bottou. Towards Principled Methods for Training Generative Adversarial Networks. *ArXiv e-prints*, Jan. 2017.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *ArXiv e-prints*, Jan. 2017.

[3] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and Equilibrium in Generative Adversarial Nets (GANs). *ArXiv e-prints*, Mar. 2017.

[4] A. Borji. Pros and Cons of GAN Evaluation Measures. *ArXiv e-prints*, Feb. 2018.

[5] T. Che, Y. Li, A. P. Jacob, Y. Bengio, and W. Li. Mode regularized generative adversarial networks. *CoRR*, abs/1612.02136, 2016.

[6] I. J. Goodfellow. On distinguishability criteria for estimating generative models. *ArXiv e-prints*, Dec. 2014.

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. *ArXiv e-prints*, June 2014.

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[9] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[10] S. Hitawala. Comparative Study on Generative Adversarial Networks. *ArXiv e-prints*, Jan. 2018.

[11] D. P. Kingma and M. Welling. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.

[12] N. Kodali, J. D. Abernethy, J. Hays, and Z. Kira. How to train your DRAGAN. *CoRR*, abs/1705.07215, 2017.

[13] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Multi-class generative adversarial networks with the L2 loss function. *CoRR*, abs/1611.04076, 2016.

[14] L. M. Mescheder, S. Nowozin, and A. Geiger. The numerics of gans. *CoRR*, abs/1705.10461, 2017.

[15] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016.

[16] S. Mohamed and B. Lakshminarayanan. Learning in Implicit Generative Models. *ArXiv e-prints*, Oct. 2016.

[17] V. Nagarajan and J. Z. Kolter. Gradient descent GAN optimization is locally stable. *CoRR*, abs/1706.04156, 2017.

[18] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

[19] J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

[20] S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. *ArXiv e-prints*, June 2016.

[21] G. Qi. Loss-sensitive generative adversarial networks on lipschitz densities. *CoRR*, abs/1701.06264, 2017.

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

[23] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann. Stabilizing training of generative adversarial networks through regularization. *CoRR*, abs/1705.09367, 2017.

[24] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016.

[25] J. J. Zhao, M. Mathieu, and Y. LeCun. Energy-based generative adversarial network. *CoRR*, abs/1609.03126, 2016.