Understanding GANs in the LQG Setting: Formulation, Generalization and Stability

Soheil Feizi, Farzan Farnia, Tony Ginart, and David Tse

***Abstract*—Generative Adversarial Networks (GANs) have become a popular method to learn a probability model from data. In this paper, we provide an understanding of basic issues surrounding GANs including their formulation, generalization and stability on a simple LQG benchmark where the generator is *Linear*, the discriminator is *Quadratic* and the data has a high-dimensional Gaussian distribution. Even in this simple benchmark, the GAN problem has not been well-understood as we observe that existing state-of-the-art GAN architectures may fail to learn a proper generative distribution owing to (1) stability issues (i.e., convergence to bad local solutions or not converging at all), (2) approximation issues (i.e., having improper global GAN optimizers caused by inappropriate GAN's loss functions), and (3) generalizability issues (i.e., requiring large number of samples for training). In this setup, we propose a GAN architecture which recovers the maximum-likelihood solution and demonstrates fast generalization. Moreover, we analyze global stability of different computational approaches for the proposed GAN and highlight their pros and cons. Finally, through experiments on MNIST and CIFAR-10 datasets, we outline extensions of our model-based approach to design GANs in more complex setups than the considered Gaussian benchmark.**

***Index Terms*—Generative models, Wasserstein distance, PCA, stability, Lyapunov functions.**

I. Introduction

LEARNING a probability model from data is a fundamental problem in statistics and machine learning. Building off the success of deep learning, Generative Adversarial Networks (GANs) [1] have given this age-old problem a face-lift. In contrast to traditional methods of parameter fitting like maximum likelihood estimation, the GAN approach views the problem as a *game* between a *generator* whose goal is to generate fake samples that are close to the real data training samples and a *discriminator* whose goal is to distinguish between the real and fake samples. The generator and the discriminator are typically implemented by deep neural networks. GANs have achieved impressive performance in several domains (e.g., [2], [3]). However, training good GANs is still challenging and it is an active area to design GANs with better and more stable performance (e.g., [4], [5], [6] and Section I-A).

The game theoretic formulation in GANs can be viewed as the dual of an optimization that minimizes a distance measure between the empirical distributions of the fake and real (observed) samples. This optimization minimizes the distance between the generated distribution and the true distribution from which the data is drawn. In the original GAN framework [1] , this distance measure is the Jenson Shannon divergence. However, Arjovsky *et al.* [4] noted that this distance does not depend on the generated distribution whenever its dimension is smaller than that of the true distribution. To resolve this issue, [4] proposed the Wasserstein GAN (WGAN) which uses the (first-order) Wasserstein distance instead of Jensen-Shannon divergence. There are many other distance measures that satisfy this criterion leading to different GAN architectures. We review some of these GANs in Section I-A.

GANs' evaluations are primarily done on real data, typically images. Although clearly valuable, such evaluations are often subjective owing to not having clear baselines to compare against. To better understand GANs, we first report experiments of some state-of-the-art GANs on *synthetic* data where clear baselines are known. We chose one of the simplest high-dimensional distributions: the *Gaussian* distribution. Even in this simple benchmark, we observe that existing state-of-the-art GAN architectures may fail to learn a proper generative distribution owing to (1) stability issues (i.e., convergence to bad local solutions or not converging at all), (2) approximation issues (i.e., having improper global GAN optimizers caused by inappropriate GAN's loss functions), and (3) generalizability issues (i.e., requiring large number of samples for training) (see Section II for more details).

These empirical results motivate us to study intertwined aspects of GANs *jointly* including their formulations, generalization and stability. Our key intuition is that to have a good performance, it is critical to take into account all of the aforementioned aspects in designing GANs. We summarize our results below.

- *GAN's formulation:* In Section III, we study GAN's formulation and provide a principled approach to choose proper *loss functions* in GANs by establishing a connection between supervised and unsupervised learning.

Manuscript received October 11, 2019; revised April 17, 2020; accepted April 23, 2020. Date of publication April 29, 2020; date of current version June 8, 2020. This work was supported in part by the National Science Foundation CAREER AWARD under Grant 1942230, in part by the National Science Foundation under Grant CCF 0939370 (Center for Science of Information) and Grant CIF 1563098, and in part by the Simons Fellowship on "Foundations of Deep Learning" and a Sponsorship from Capital One. *(Corresponding author: Soheil Feizi.)*

Soheil Feizi is with the Department of Computer Science, University of Maryland, College Park, MD 20742 USA (e-mail: sfeizi@cs.umd.edu).

Farzan Farnia is with Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: farnia@mit.edu).

Tony Ginart and David Tse are with Stanford University, Stanford, CA 94305 USA (e-mail: tginart@stanford.edu; dntse@stanford.edu).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors. The material includes an Appendix file containing the proofs and additional information.

Digital Object Identifier 10.1109/JSAIT.2020.2991375

Leveraging this result, we design a GAN architecture for the Gaussian benchmark which provably recovers the maximum likelihood solution (Theorem 1 for the population case, and Theorem 3 for the empirical case.) We discuss extensions of these results to more complex distributions in Appendix Sections XI and XII.

- *GAN's generalization*: In Section IV, we study GAN's generalization and prove that GANs with unconstrained discriminators can have poor generalizations (Theorem 2). However, in Section V and for the Gaussian setup, we show that the poor generalization issue of GANs can be resolved by properly constraining the discriminator class to convex quadratic functions.
- *GAN's global stability*: A central computational question on GANs is to understand the global (or at least, local) stability of alternating gradient descent. Although there has been some recent efforts to understand local stability of GANs, understanding GAN's global stability seems to be a very challenging problem. In Section VI, we analyze the global stability of different computational approaches for a family of GANs and highlight their pros and cons. To the best of our knowledge, we provide the first study of global convergence of a GAN architecture using the Von Neumann divergence as a Lyapunov function (Theorem 5).

We believe these results make progress on our understanding of different intertwined aspects of GANs.

The paper is organized as follows: In Section II, we show that some modern GANs fail to learn a high dimensional Gaussian distribution. To study this problem, we designed the Quadratic GAN in three steps: in Section III, we formulated GAN's objective by specifying the appropriate loss to naturally match the Gaussian model for the data. This allows us to show that the global population solution of the minmax problem is the r-PCA of the (true) covariance matrix of the Gaussian model (Theorem 1). However, this initial architecture can have poor generalization performance (Section IV). Next, we further constrained the discriminator to keep the good optimal solution of the population-optimal architecture while enabling fast generalization (Section V). We refer to this architecture as the quadratic GAN (Figure 3). We show that the global optimizer of quadratic GAN applied on the empirical distribution is the *empirical* r-PCA (Theorem 3). Finally, we study the global stability of different computational approaches for solving the proposed GAN architecture. In particular, we prove that in the full-rank case alternating gradient descent converges globally to the minmax solution, under some conditions (Section VI). In what follows, we provide more details about these results. All proofs are presented in the Appendix.

A. Prior Work

Broadly speaking, previous work in GANs study three main properties: (1) Stability where the focus is on the convergence of the commonly used alternating gradient descent approach to global/local optimizers (equilibriums) for GAN's optimization (e.g., [6], [7], [8], [9], [10], etc.), (2) Formulation where the focus is on designing proper loss functions for GAN's optimization (e.g., WGAN+Weight Clipping [4], WGAN+Gradient Penalty [5], GAN+Spectral Normalization [11], WGAN+Truncated Gradient Penalty [12], relaxed WGAN [13], f-GAN [14], MMD-GAN [15], [16], Least-Squares GAN [17], Boundary equilibrium GAN [18], etc.), and (3) Generalization where the focus is on understanding the required number of samples to learn a probability model using GANs (e.g., [19]). We address all three issues in the design of the Quadratic GAN introduced in Section V.

Some references have also proposed model-based GANs for the Gaussian benchmark [7], [10]. For example, [10] uses a quadratic function as the discriminator in the WGAN optimization. This design, however, does not recover the maximum likelihood/PCA solutions in the Gaussian benchmark, unlike the Quadratic GAN. Moreover, no global stability results were proven.

II. Evaluating GANs on Gaussian Benchmarks

To better understand different aspects of GANs, first we evaluate the performance of some of the state-of-the-art GAN architectures on a high dimensional Gaussian benchmark. We choose this benchmark since optimal baselines are known in this case.

In our first set of experiments, we generate $n = 100,000$ samples from a $d = 32$ dimensional Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{K})$ when $\mathbf{K}$ is the normalized identity matrix; $\mathbf{K} = \mathbf{I}/\sqrt{d}$. We train two state-of-the-art GAN architectures in our experiments; WGAN+Weight Clipping (WGAN+WC) [4] and WGAN+Gradient Penalty (WGAN+GP) [5]. We use the neural net generator and discriminator with hyper-parameter settings as recommended in [5]. Each of the neural networks has three hidden layers, each with 64 neurons and ReLU activation functions. To evaluate GAN's performance, we compute the Frobenius norm between covariance matrices of observed and generative distributions.

Figure 1 shows the performance of GANs for various values of r, the dimension of the randomness (i.e., input to the generator, for which we use the standard Gaussian randomness) and for two random initializations for ReLU layers using the standard He *et al.* [20] and Glorot and Bengio [21] procedures. In these experiments, we observe two types of instability in GAN's performance; oscillating behaviour (e.g., WGAN-GP, $r = 4, 8$) and convergence to different and bad local solutions. Even after $20,000$ training epochs, the error does not approach zero in most cases. We observe similar trends when we use a random covariance instead of the normalized identity matrix (SM Figure 4).

In our next set of experiments, we attempt to improve performance by restricting the generator to be linear, since both the observed data and the randomness come from Gaussian distributions. (The discriminator is still the ReLU neural network). Since the generator is linear, zero error cannot be achieved in the case of $r < d$. In this case, a natural baseline is the r-PCA of the sample covariance matrix (SM Section IX). GAN's performance improves compared to the case of nonlinear generator (Figure 2). We do not observe oscillating behavior in WGAN+GP. However,

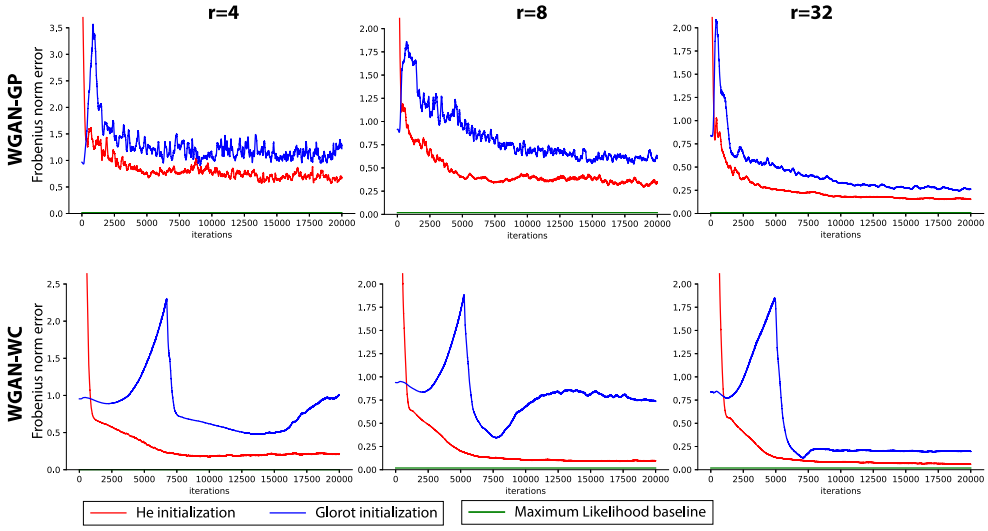

Fig. 1. An illustration of the performance of WGAN+GP and WGAN+WC in different values of r (the dimension of the input randomness to the generator) with different initialization procedures when the generator and the discriminator functions are both neural networks.

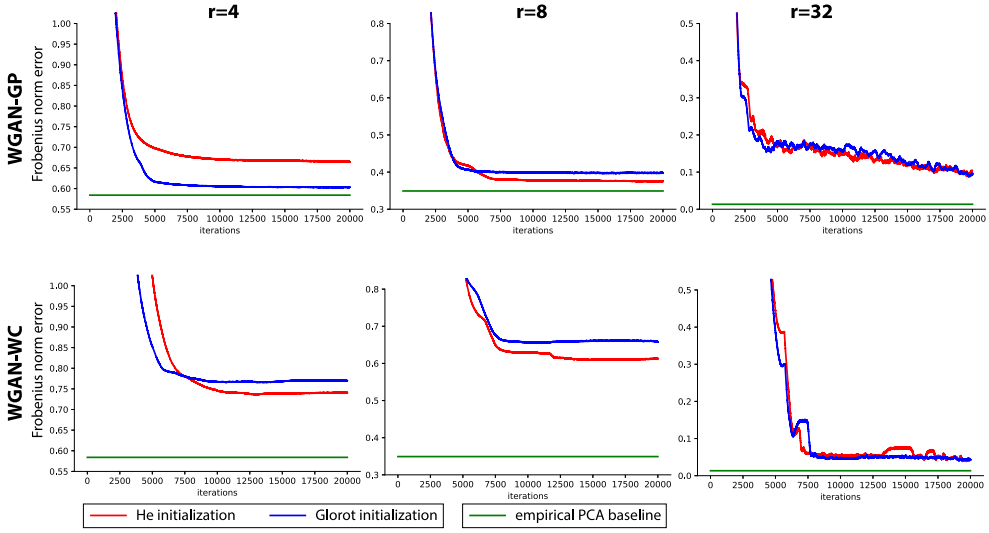

Fig. 2. A repeat of experiments of Figure 1 when the generator function is linear. A random covariance matrix is chosen instead of the identity matrix.

we still observe convergence to different bad local solutions for both WGAN+GP and WGAN+WC. Unlike Figure 1 where WGAN+WC was performing better than WGAN+GP, here the performance of WGAN+WC is significantly worst than that of the WGAN+GP. Also, unlike other cases, in WGAN+GP when $r = 4$, the Glorot initialization achieves a smaller error than that of the He initialization. These results highlight sensitivity of state-of-the-art GANs even in a simple benchmark. These empirical results motivate us to study different fundamental aspects of GANs.

III. A General Formulation for GANs

Let $\{\mathbf{y}_i\}_{i=1}^n$ be n observed data points in $\mathbb{R}^d$ drawn i.i.d. from the distribution $\mathbb{P}_Y$. Let $\mathbb{Q}_Y^n$ be the empirical distribution of these observed samples. Moreover, let $\mathbb{P}_X$ be a normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_r)$. GANs can be viewed as an optimization that minimizes a distance between the observed empirical distribution $\mathbb{Q}_Y^n$ and the generated distribution $\mathbb{P}_{\mathbf{G}(X)}$. The *population* GAN optimization replaces $\mathbb{Q}_Y^n$ with $\mathbb{P}_Y$ and is the setting we focus on in this section. The question we ask in this section is: what is a natural way of specifying a loss function ℓ for GANs and how it determines the GAN's objective? We answer the question in general and then specialize to the Gaussian benchmark by choosing an appropriate loss function for that case which is the quadratic loss function. We then show that using the resulting GAN, we obtain a good population solution (i.e., the maximum likelihood solution) under this loss function.

A. WGAN Revisited

Let us start with the WGAN optimization [4]:

$$\min_{\mathbf{G}(.)\in\mathcal{G}} \; W_1\big(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}(X)}\big), \tag{1}$$

where $\mathcal{G}$ is the set of generator functions, and the p-th order Wasserstein distance between distributions $\mathbb{P}_{Z_1}$ and $\mathbb{P}_{Z_2}$ is

defined as [22]

$$W_p^p\big(\mathbb{P}_{Z_1}, \mathbb{P}_{Z_2}\big) \coloneqq \min_{\mathbb{P}_{Z_1,Z_2}} \mathbb{E}\big[\|Z_1 - Z_2\|^p\big], \tag{2}$$

where the minimization is over all joint distributions with marginals fixed. Replacing (2) in (1), the WGAN optimization can be re-written as

$$\min_{\mathbf{G}(.)\in\mathcal{G}} \min_{\mathbb{P}_{\mathbf{G}(X),Y}} \mathbb{E}[\|Y - \mathbf{G}(X)\|]. \tag{3}$$

or equivalently:

$$\min_{\mathbb{P}_{X,Y}} \min_{\mathbf{G}(.)\in\mathcal{G}} \mathbb{E}[\|Y - \mathbf{G}(X)\|], \tag{4}$$

where the minimization is over all joint distributions $\mathbb{P}_{X,Y}$ with fixed marginals $\mathbb{P}_X$ and $\mathbb{P}_Y$.

We now connect (4) to the *supervised learning* setup. In supervised learning, the joint distribution $\mathbb{P}_{X,Y}$ is fixed and the goal is to learn a relationship between the feature variable represented by $X \in \mathbb{R}^r$, and the target variable represented by $Y \in \mathbb{R}^d$, according to the following optimization:

$$\min_{\mathbf{G}(.)\in\mathcal{G}} \mathbb{E}[\ell(Y, \mathbf{G}(X))], \tag{5}$$

where ℓ is the *loss* function. Assuming the marginal distribution of X is the same in both optimizations (4) and (5), we can connect the two optimization problems by setting $\ell(y, y') = \|y - y'\|$ in optimization (5). Note that for every fixed $\mathbb{P}_{X,Y}$, the solution of the supervised learning problem (5) yields a predictor g which is a feasible solution to the WGAN optimization problem (4). Therefore, the WGAN optimization (3) can be re-interpreted as solving the *easiest* such supervised learning problem, over all possible joint distributions $\mathbb{P}_{X,Y}$ with fixed $\mathbb{P}_X$ and $\mathbb{P}_Y$.

B. From Supervised to Unsupervised Learning

GAN is a solution to an unsupervised learning problem. What we are establishing above is a general connection between supervised and unsupervised learning problems: a good predictor $\mathbf{G}$ learnt in a supervised learning problem can be used to generate samples of the target variable Y. Hence, to solve an unsupervised learning problem for Y with distribution $\mathbb{P}_Y$, one should solve the easiest supervised learning problem $\mathbb{P}_{X,Y}$ with given marginal $\mathbb{P}_Y$ (and $\mathbb{P}_X$, the randomness generating distribution). This is in contrast to the traditional view of the unsupervised learning problem as observing the feature variable X without the label Y. (Thus in this paper we break with tradition and use Y to denote data and X as randomness for the generator in stating the GAN problem.)

This connection between supervised and unsupervised learning leads to a natural way of specifying the loss function in GANs: we simply replace the ℓ_2 in (3) with a general loss function ℓ:

$$\min_{\mathbf{G}(.)\in\mathcal{G}} \min_{\mathbb{P}_{\mathbf{G}(X),Y}} \mathbb{E}[\ell(Y, \mathbf{G}(X))]. \tag{6}$$

The inner optimization is the optimal transport problem between distributions of $\mathbf{G}(X)$ and Y [22] with general cost ℓ. This is a linear programming problem for general cost, so there is always a dual formulation under some general conditions, this leads to the Kantorovich duality [22]). The dual formulation can be interpreted as a generalized discriminator optimization problem for the cost ℓ. (For example, in the case of ℓ being the Euclidean norm, we get WGAN.) Hence, we use (6) as a formulation of GANs for general loss functions.

Note that an optimal transport view to GANs has been studied in other references (e.g., [4], [23]). Our contribution in this section is to make a *connection* between supervised and unsupervised learning problems which we will exploit to specify a proper loss function for GANs in the Gaussian model.

C. Quadratic Loss and Linear Generators

The most widely used loss function in supervised learning is the quadratic loss: $\ell(y, y') = \|y - y'\|^2$ (*squared* Euclidean norm). The quadratic loss has a strong synergy with the Gaussian model, as observed by Gauss himself. For example, under the Gaussian model and the quadratic loss in the supervised learning problem (5), the optimal g is linear, thus forming a statistical basis for linear regression. Given the connection between supervised and unsupervised learning, we use this loss function for formulating the GAN for Gaussian data . This choice of the loss function leads to the following GAN optimization which we refer to as *W2GAN*:

$$\min_{\mathbf{G}(.)\in\mathcal{G}} W_2^2\big(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}(X)}\big). \tag{7}$$

A natural choice of $\mathcal{G}$ is the set of all linear generators, from $\mathbb{R}^r$ to $\mathbb{R}^d$.

Since Wasserstein distances are weakly continuous measures in the probability space [22], similar to WGAN, the optimization of the W2GAN is well-defined even if $r < d$. The dual formulation (discriminator) for W_2^2 is [22]:

$$\begin{aligned} W_2^2\big(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}(X)}\big) = \max_{\psi(.):\text{convex}} \ & \mathbb{E}\Big[\|Y\|^2 - 2\psi(Y)\Big] \\ & - \mathbb{E}\Big[2\psi^*(\mathbf{G}(X)) - \|\mathbf{G}(X)\|^2\Big], \end{aligned} \tag{8}$$

where

$$\psi^*(\hat{\mathbf{y}}) \coloneqq \max_{\mathbf{v}}\big(\mathbf{v}^t\hat{\mathbf{y}} - \psi(\mathbf{v})\big) \tag{9}$$

is the convex-conjugate of the function $\psi(.)$. Combining (7) and (8), we obtain the minmax formulation of W2GAN:

$$\begin{aligned} \min_{\mathbf{G}\in\mathcal{G}} \max_{\psi(.):\text{convex}} \ & \mathbb{E}\Big[\|Y\|^2 - 2\psi(Y)\Big] \\ & - \mathbb{E}\Big[2\psi^*(\mathbf{G}(X)) - \|\mathbf{G}(X)\|^2\Big]. \end{aligned} \tag{10}$$

D. Population Solution: PCA

There is a simple solution to the optimization problem (7) in the population setting.

Theorem 1: Let $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{K})$ where $\mathbf{K}$ is full-rank. Let $X \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_r)$ where $r \leq d$. The optimal GAN solution in the population setting under linear generators $\mathcal{G}$ is the r-PCA solution of Y.

We say $\hat{Y}$ is the r-PCA solution of Y if $\mathbf{K}_{\hat{Y}}$ is a rank r matrix whose top r eigenvalues and eigenvectors are the same as top r eigenvalues and eigenvectors of $\mathbf{K}$. This theorem is

satisfactory as it connects GANs to PCA, one of the most basic unsupervised learning methods.

IV. Generalization of GANs

Consider the empirical version of the population W2GAN optimization problem (7):

$$\min_{\mathbf{G}(.)\in\mathcal{G}} \; W_2^2\big(\mathbb{Q}_Y^n, \mathbb{P}_{\mathbf{G}(X)}\big), \tag{11}$$

where $\mathbb{Q}_Y^n$ is the empirical distribution of the n data points $\{\mathbf{y}_i\}_{i=1}^n$. Let $\mathbf{G}_n^*$ be the optimal solution of this problem. The distance between the generated distribution $\mathbf{G}^*(X)$ and the true distribution $\mathbb{P}_Y$, $W_2^2(\mathbb{P}_Y, \mathbb{P}_{g_n^*(X)})$, converges to zero as $n \to \infty$. It was shown in [19] that if the generator class $\mathcal{G}$ is rich enough so that the generator can memorize the data and generate the empirical distribution $\mathbb{Q}_Y^n$ itself, then this rate of convergence is very slow, of the order of $n^{-2/d}$. (Strictly speaking, they have only shown it for the W_1 distance, but a very similar result holds for W_2 as well.) This is because the empirical distribution $\mathbb{Q}_Y^n$ converges very slowly to the true distribution $\mathbb{P}_Y$ in the W_2 distance. Hence, the number of samples required for convergence is exponential in the dimension d. In this section, we show that in our Gaussian setup, even if we constrain the generators to single-parameter linear functions that can generate the true distribution, the rate of convergence is still $n^{-2/d}$.

First, we define the generalization error in GANs as follows.

Definition 1: Let n be the number of observed samples from Y. Let $\hat{\mathbf{G}}(.)$ and $\mathbf{G}^*(.)$ be the optimal generators for empirical and population W2GANs respectively. Then,

$$d_{\mathcal{G}}\big(\mathbb{P}_Y, \mathbb{Q}_Y^n\big) \coloneqq W_2^2\Big(\mathbb{P}_Y, \mathbb{P}_{\hat{\mathbf{G}}(X)}\Big) - W_2^2\big(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}^*(X)}\big), \tag{12}$$

is a random variable representing the excess error of $\hat{\mathbf{G}}$ over $\mathbf{G}^*$, evaluated on the true distribution.

$d_{\mathcal{G}}(\mathbb{P}_Y, \mathbb{Q}_Y^n)$ can be viewed as a distance between $\mathbb{P}_Y$ and $\mathbb{Q}_Y^n$ which depends on $\mathcal{G}$. To have a proper generalization property, one needs to have $d_{\mathcal{G}}(\mathbb{P}_Y, \mathbb{Q}_Y^n) \to 0$ quickly as $n \to \infty$. First, we characterize this rate for an unconstrained $\mathcal{G}$. For an unconstrained $\mathcal{G}$, the second term of (12) is zero (this can be seen using a space filling generator function [24]). Moreover, $\mathbb{P}_{\hat{\mathbf{G}}(X)}$ can be arbitrarily close to $\mathbb{Q}_Y^n$. Thus, we have

$$d_{\mathcal{G}}\big(\mathbb{P}_Y, \mathbb{Q}_Y^n\big) = W_2^2\big(\mathbb{P}_Y, \mathbb{Q}_Y^n\big), \tag{13}$$

which goes to zero with high probability with the rate of $n^{-2/d}$.

The approach described for the unconstrained $\mathcal{G}$ corresponds to the *memorization* of the empirical distribution $\mathbb{Q}_Y^n$ using the trained model. Note that one can write

$$n^{-\frac{2}{d}} = 2^{-\frac{2\log(n)}{d}}.$$

Thus, to have a small $W_2^2(\mathbb{P}_Y, \mathbb{Q}_Y^n)$, the number of samples n should be exponentially large in d [25]. It is possible that for some distributions $\mathbb{P}_Y$, the convergence rate of $W_2^2(\mathbb{P}_Y, \mathbb{Q}_Y^n)$ is much faster than $n^{-2/d}$. For example, [26] shows that if $\mathbb{P}_Y$ is clusterable (i.e., Y lies in a fixed number of separate balls with fixed radii), then the convergence of $W_2^2(\mathbb{P}_Y, \mathbb{Q}_Y^n)$ is fast. However, even in that case, one optimal strategy would be to memorize observed samples, which does not capture what GANs aim for.

In supervised learning, constraining the predictor to be from a small family improves generalization. A natural question is whether constraining the family of generator functions $\mathcal{G}$ can improve the generalization of GANs. In the Gaussian setting, we are constraining the generators to be linear. To simplify calculations, we assume that $Y \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ and $d = r$. Under these assumptions, the W2GAN optimization can be re-written as

$$\min_{\mu,\mathbf{K}} \; W_2^2\big(\mathbb{Q}_Y^n, \mathcal{N}(\mu, \mathbf{K})\big), \tag{14}$$

where $\mathbf{K}$ is the covariance matrix. The optimal population solution of this optimization is $\mu_{pop}^* = \mathbf{0}$ and $\mathbf{K}_{pop}^* = \mathbf{I}$, which provides a zero Wasserstein loss with respect to the true distribution.

In the following theorem, we characterize the convergence rate of the W2GAN optimization for linear generators with single-parameters. In this case, $\mathbf{K} = s^2\mathbf{I}$ ($\mathbf{K}$ is a diagonal matrix whose diagonal elements are equal to s^2). Note that if $s = 1$, the trained model matches the population distribution.

Theorem 2: Let μ_n^* and $\mathbf{K}_n^* = (s^*)^2\mathbf{I}$ be optimal solutions for optimization (14) where $\mathbf{K}$ is restricted to $s^2\mathbf{I}$ (i.e., the generator is a single-parameter linear function). Then, $s^* \to 1$ with the rate of $n^{-2/d}$.

Now, consider a ball around the distribution $\mathbb{Q}_Y^n$ where $\mathbb{P}_Y$ lies on its surface. Note that the radius of this ball is a random variable that is concentrated around $n^{-2/d}$ [22]. This radius is large and goes to zero exponentially slow in d. If there is another Gaussian distribution inside this ball, the learner would select that distribution in the optimization rather than $\mathbb{P}_Y$. The Gaussian distribution computed in Theorem 2 satisfies this condition. Thus, in this case, one needs n to be exponentially large in d to have the error go to zero. To enhance the convergence rate of GANs, in practice, discriminators are constrained. We discuss how discriminators should be constrained properly in the Gaussian benchmark in the next section.

V. Quadratic GAN

The discriminator of the W2GAN optimization (10) is constrained over all convex functions. Since this set is non-parametric, we are unable to use gradient descent to compute a solution for this optimization. Moreover, having such a large feasible set for the discriminator function can cause poor *generalization* as explained in the previous section.

To overcome both of these issues, one option is to further constrain the *discriminator*. Ideally one would like to *properly* constrain the discriminator function such that any population solution of the constrained optimization is a population solution of the original optimization and vice versa, while at the same time allowing fast generalization. In this section, we show how we can achieve this goal for the Gaussian benchmark. This view can potentially be extended to more complex distributions as we explain in Section VII.

The following lemma characterizes the optimal solution of optimization (8) [22].

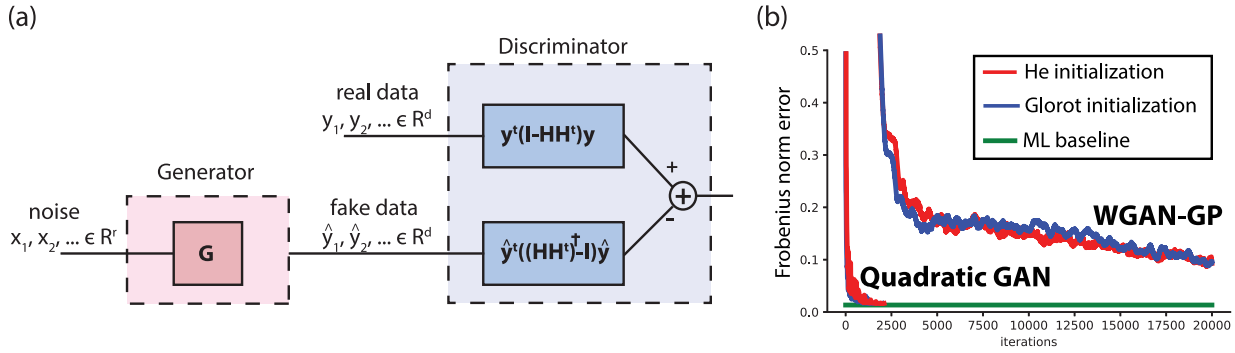

Fig. 3. (a) Quadratic GAN, with a linear generator and a quadratic discriminator. On the training data, the generator minimizes over the d by r matrix G and the adversary maximizes over the d by d matrix H. (b) Performance comparison between quadratic GAN and WGAN+GP for $d = r = 32$.

Lemma 1: Let $\mathbb{P}_Y$ be absolutely continuous whose support contained in a convex set in $\mathbb{R}^d$. For a fixed $\mathbf{G}(.) \in \mathcal{G}$, let ψ^{opt} be the optimal solution of optimization (8). This solution is unique. Moreover, we have

$$\nabla \psi^{opt}(Y) \overset{\text{dist}}{=} \mathbf{G}(X), \tag{15}$$

where $\overset{\text{dist}}{=}$ means matching distributions.

In our benchmark setup, since $\mathbf{G}(X)$ is Gaussian, $\nabla\psi^{opt}$ is a linear function. Thus, without loss of generality, $\psi(.)$ in the discriminator optimization can be constrained to quadratic functions of the form $\psi(\mathbf{y}) = \mathbf{y}^t\mathbf{A}\mathbf{y}/2$ where $\mathbf{A}$ is positive semidefinite. For the quadratic function, we have $\psi^*(\hat{\mathbf{y}}) = \hat{\mathbf{y}}^t\mathbf{A}^\dagger\hat{\mathbf{y}}/2$ when range$(\hat{Y}) \subseteq$ range$(\mathbf{A})$. Here range(.) refers to the co-domain of its input matrix.

Replacing these in optimization (10), we obtain:

$$\min_{\mathbf{G}} \max_{\mathbf{A}\succeq 0} \mathbb{E}\Big[Y^t(\mathbf{I}-\mathbf{A})Y\Big] - \mathbb{E}\Big[\hat{Y}^t\Big(\mathbf{A}^\dagger - \mathbf{I}\Big)\hat{Y}\Big] \quad \text{range}(\mathbf{G}) \subseteq \text{range}(\mathbf{A}). \tag{16}$$

Without loss of generality, we can replace the constraint range$(\mathbf{G}) \subseteq$ range$(\mathbf{A})$ with range$(\mathbf{G}) =$ range$(\mathbf{A})$. It is because for a given $\mathbf{A}$, this increases the size of the feasible set for $\mathbf{G}$ optimization, thus the objective can achieve a smaller value. For a given $\mathbf{G}$, one can decompose $\mathbf{A}$ as $\mathbf{A}_1 + \mathbf{A}_2$ where range$(\mathbf{A}_1) =$ range$(\mathbf{G})$ and range$(\mathbf{A}_2) \cap$ range$(\mathbf{G}) = \emptyset$. Note that by ignoring $\mathbf{A}_2$, the objective function does not decrease. Therefore, optimization (16) can be written as

$$\min_{\mathbf{G}} \max_{\mathbf{A}\succeq 0} \mathbb{E}\Big[Y^t(\mathbf{I}-\mathbf{A})Y\Big] - \mathbb{E}\Big[\hat{Y}^t\Big(\mathbf{A}^\dagger - \mathbf{I}\Big)\hat{Y}\Big] \quad \text{range}(\mathbf{G}) = \text{range}(\mathbf{A}). \tag{17}$$

Using the fact that trace is invariant under cyclic permutations and by replacing $\mathbf{A} = \mathbf{H}\mathbf{H}^t$, the objective function of the above optimization can be re-written as:

$$J(\mathbf{G},\mathbf{H}) = \text{Tr}\Big[\big(\mathbf{I}-\mathbf{H}\mathbf{H}^t\big)\mathbf{K}\Big] - \text{Tr}\Big[\Big(\big(\mathbf{H}\mathbf{H}^t\big)^\dagger - \mathbf{I}\Big)\mathbf{G}\mathbf{G}^t\Big]. \tag{18}$$

In practice, we apply GANs to the observed data (i.e., the empirical distribution). In that case, in the above objective function, $\mathbf{K}$ (the true covariance) should be replaced by $\tilde{\mathbf{K}}$ (the empirical covariance). This leads to the *quadratic GAN* optimization:

$$\min_{\mathbf{G}} \max_{\mathbf{H}} \text{Tr}\Big[\big(\mathbf{I}-\mathbf{H}\mathbf{H}^t\big)\tilde{\mathbf{K}}\Big] - \text{Tr}\Big[\Big(\big(\mathbf{H}\mathbf{H}^t\big)^\dagger - \mathbf{I}\Big)\mathbf{G}\mathbf{G}^t\Big]$$
$$\text{range}(\mathbf{G}) = \text{range}(\mathbf{H}). \tag{19}$$

Note that since the global optimizer of optimization (18) is PCA (Theorem 1), the global optimizer of optimization (19) is empirical PCA.

Theorem 3: Let $\tilde{\mathbf{K}}_r$ be the r-PCA of the sample covariance matrix. Let $(\mathbf{G}^*, \mathbf{H}^*)$ be a global solution for the quadratic GAN optimization (19). Then, we have $\mathbf{G}^*(\mathbf{G}^*)^t = \tilde{\mathbf{K}}_r$. I.e., quadratic GAN recovers the empirical PCA solution as the generative model.

Next, we examine the generalization error of the quadratic GAN. Consider the case where $d = r$ (the case $r < d$ is similar). The generalization error can be written as the W_2 distance between the true distribution $\mathbb{P}_Y$ and the learned distribution $\mathbb{P}_{\mathbf{G}^*(X)}$ (Section IV):

$$W_2^2\big(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}^*(X)}\big) = W_2^2\Big(\mathcal{N}(0,\mathbf{K}), \mathcal{N}(0,\tilde{\mathbf{K}})\Big). \tag{20}$$

The W_2^2 distance between two Gaussians depends only on the covariance matrices. More specifically:

$$W_2^2\Big(\mathcal{N}(0,\mathbf{K}), \mathcal{N}(0,\tilde{\mathbf{K}})\Big) = \text{Tr}(\mathbf{K}) + \text{Tr}(\tilde{\mathbf{K}}) - 2\text{Tr}\Big(\Big(\mathbf{K}^{1/2}\tilde{\mathbf{K}}\mathbf{K}^{1/2}\Big)^{1/2}\Big). \tag{21}$$

The convergence of this quantity only depends on the convergence of the empirical covariance to the population one, together with smoothness property of this function of the covariance matrices. The convergence has been established to be at a quick rate of $\tilde{\mathcal{O}}(\sqrt{d/n})$ [27].

Figure 3-a illustrates the quadratic GAN architecture.[1] Figure 3-b compares performance of quadratic GAN and WGAN+GP for $r = 32$. Quadratic GAN demonstrates stable behavior and much faster convergence to the maximum-likelihood baseline compared to WGAN. In fact, due to its simple structure, training of the Quadratic GAN takes less than 1 second on a laptop CPU which is orders of magnitudes faster than training WGAN on a GPU.

Remark 1: In the connection between unsupervised and supervised learning in Section III, we see that unsupervised learning is interpreted as optimizing both the coupling between X and Y and the predictor G in the resulting supervised

[1]For simplicity, we assume that samples have been centered to have zero means. In the general case, the generator should be an affine function.

learning problem. Thus, in addition to constraining the predictor G, another approach to improve generalization is to constrain the primal coupling as well, through regularization. One form of such regularized version of optimal transport is called Sinkhorn divergence [28]. However, our approach of constraining the discriminator in the dual form has a different interpretation. As shown in [29], the proposed quadratic constraint on the discriminator can be interpreted as weakening the original Wasserstein distance to the minimum Wasserstein distance between any two distributions sharing the same mean and covariance matrix with P_Y and $P_{G(X)}$. However, as we prove here this change in the target distance does not affect the optimal solution to the W2GAN problem under the LQG assumptions. A deeper investigation into the connection between these two approaches is an interesting research direction.

VI. Global Stability

Theorem 3 merely focuses on the quality of the global solution of the quadratic GAN's optimization, ignoring its computational aspects. One common way to solve the GAN's min-max optimization is to use alternating gradient descent with s_G gradient steps for the generator updates and s_D gradient steps for the discriminator updates. For simplicity, we refer to such a method as the (s_G, s_D)-alternating gradient descent (AGD). In this section, we analyze the global stability of the quadratic GAN under the alternating gradient descent approach. The global stability feature indicates the convergence of the AGD algorithm to the optimal solution irrespective of its initialization point.

First, we analyze the stability of the quadratic GAN under the (1, 1)-alternating GD in the full-rank case. By using variables $\mathbf{U} \coloneqq \mathbf{G}\mathbf{G}^t$ and $\mathbf{A} \coloneqq \mathbf{H}\mathbf{H}^t$, optimization (19) can be written as

$$\min_{\mathbf{U}} \max_{\mathbf{A}} \ \mathrm{Tr}\Big[(\mathbf{I}-\mathbf{A})\tilde{\mathbf{K}}\Big] - \mathrm{Tr}\Big[\Big(\big(\mathbf{A}^t\big)^\dagger - \mathbf{I}\Big)\mathbf{U}^t\Big] \quad \mathrm{range}(\mathbf{U}) = \mathrm{range}(\mathbf{A}). \tag{22}$$

For this case, we have the following result.

Theorem 4 [30]: In the quadratic GAN optimization (22), assuming full rank $\mathbf{A}$ and $r = d$, the (1, 1)-alternating gradient descent is globally stable.

Note that reference [30] discusses the stability of general convex-concave min-max optimization problems while here we aim to characterize the global stability of the proposed quadratic GAN. In particular, in the standard quadratic GAN, the alternating GD is applied on the $(\mathbf{G}, \mathbf{H})$ objective function which is not generally convex-concave. For this case, we have the following result.

Theorem 5: In the quadratic GAN optimization (19), assuming $\tilde{\mathbf{K}} = \mathbf{I}$, full rank $\mathbf{H}$ and $r = d$, the (1, 1)-alternating gradient descent is globally stable.

To prove Theorem 5, we use the following function as a Lyapunov function:

$$V(\mathbf{G}, \mathbf{H}) = \mathrm{Tr}\big[\mathbf{G}\mathbf{G}^t - \mathbf{I} - \log\big(\mathbf{G}\mathbf{G}^t\big)\big] + \mathrm{Tr}\big[\mathbf{H}\mathbf{H}^t - \mathbf{I} - \log\big(\mathbf{H}\mathbf{H}^t\big)\big]. \tag{23}$$

Each term of this function is the Von Neumann divergence. Note that log(.) of a positive-definite matrix is defined by taking the logarithm of its eigenvalues. We prove that this non-negative function is monotonically decreasing along every trajectory of the (1, 1)-alternating gradient descent and its value is zero at the global solution. This phenomena is non-trivial because the Frobenius norm distance between $\mathbf{G}\mathbf{G}^t$ and $\tilde{\mathbf{K}}$ is not monotonically decreasing along every trajectory (Appendix Figure 6).

In the low-rank case where $r < d$, however, we have the following negative result.

Theorem 6: In the quadratic GAN optimization (19), if $r < d$, the (s_G, s_D)-alternating gradient descent is *not* globally stable for any s_G and s_D.

One can think about using an equivalent optimization (19) where the constraint range($\mathbf{G}$) = range($\mathbf{H}$) is replaced by the constraint range($\mathbf{G}$) $\subseteq$ range($\mathbf{H}$) (by assuming $\mathbf{A} = \mathbf{H}\mathbf{H}^t$). For example, if $\mathbf{H}$ is full-rank, this constraint always holds. However, this does not solve the stability issue of Theorem 6. It is because in the desired saddle point, $\mathbf{H}^*$ should be a low-rank matrix whose range matches the range of $\mathbf{G}^*$. If one starts the alternating GD with a full-rank $\mathbf{H}$, the second term of the objective function (19) would decrease unboundedly when $\mathbf{H}$ loses rank in the null-space of $\mathbf{G}$ (because of the term $(\mathbf{H}\mathbf{H}^t)^\dagger\mathbf{G}\mathbf{G}^t$). Therefore, unless $\mathbf{H}$ has a matching range with $\mathbf{G}$, alternating GD will not converge to a low-rank solution for $\mathbf{H}$. Note that if the covariance matrix of the observed data itself is low rank, it makes sense to use a low rank generative model which will fall in the regime of $r < d$.

As we explained above, the main source of the instability of the quadratic GAN optimization in the low-rank case comes from the constraint range($\mathbf{G}$) = range($\mathbf{H}$), i.e., the matching column-space of the generator and the discriminator functions. One way to deal with this issue is to decouple the optimization to two parts where in one part we optimize the subspace and in the second part, we solve GAN's min-max optimization *within* that subspace. Below, we explain this approach. We denote the subspace by some orthogonal basis $\mathbf{S} \in \mathbb{R}^{d\times r}$ where $\mathbf{S}^t\mathbf{S} = \mathbf{I}$. Then, we re-write

$$\mathbf{G} \coloneqq \mathbf{S}\mathbf{G}_S, \quad \mathbf{H} \coloneqq \mathbf{S}\mathbf{H}_S, \tag{24}$$

where $\mathbf{G}_S$ and $\mathbf{H}_S$ are full-rank $r \times r$ matrices. Also, we define $\mathbf{K}_S \coloneqq \mathbf{S}^t\mathbf{K}\mathbf{S}$. Using these notation, the objective function of the quadratic GAN can be re-written as:

$$J(\mathbf{S}, \mathbf{G}_S, \mathbf{H}_S) = \mathrm{Tr}\big[\big(\mathbf{I} - \mathbf{H}_S\mathbf{H}_S^t\big)\mathbf{K}_S\big] - \mathrm{Tr}\Big[\Big(\big(\mathbf{H}_S\mathbf{H}_S^t\big)^\dagger - \mathbf{I}\Big)\mathbf{G}_S\mathbf{G}_S^t\Big] + \mathrm{Tr}[\mathbf{K} - \mathbf{K}_S]. \tag{25}$$

Note that the first two terms of this objective is the same as (18) where all variables are projected to the column-space of $\mathbf{S}$. Using the above argument, we propose the following *min-min-max* optimization:

$$\min_{\mathbf{S}} \min_{\mathbf{G}_S} \max_{\mathbf{H}_S} \ J(\mathbf{S}, \mathbf{G}_S, \mathbf{H}_S) \quad \mathbf{S}^t\mathbf{S} = \mathbf{I}. \tag{26}$$

The inner min-max optimization over $\mathbf{G}_S$ and $\mathbf{H}_S$ for a given $\mathbf{S}$ is similar to the full-rank case analysis (Theorem 5). Given the

global convergence of the (1, 1)-alternating GD in the full-rank case, the outer optimization on $\mathbf{S}$ can be re-written as

$$\max_{\mathbf{S}} \ \mathrm{Tr}\big[\mathbf{S}^t\mathbf{K}\mathbf{S}\big] \quad \mathbf{S}^t\mathbf{S} = \mathbf{I}. \tag{27}$$

Although this optimization is non-convex, it has been shown that its global optimizer, which recovers the leading eigenvectors of $\mathbf{K}$, can be computed efficiently using GD [31].

An alternative approach to solve the quadratic GAN optimization (19) is to solve the max part as a closed form and use GD to solve the min part. We analyze the convergence of this approach in Appendix Theorem 7.

VII. Discussion

Our experiments on state-of-the-art GAN architectures suggest limitations of model-free designs even when data comes from a very basic Gaussian model. This motivates us to take a model-based approach to designing GANs. In this paper, we accomplish this goal in the spacial case of Gaussian models. Even though this is for a restrictive case, we have learnt a few lessons which will be useful as we broaden our approach. We obtained a general way to specify loss functions for GANs, by connecting the unsupervised GAN learning problem to the supervised learning problem. The quadratic loss function used for the Gaussian problem is a special case of this general connection. Moreover, we learnt that by properly constraining the class of generators and the class of discriminators in a *balanced* way, we can preserve good population solution while allowing fast generalization. Finally, we saw that using a model-based design, we could analyze the global stability of different computational approaches using gradient descent. These properties are hard to come by in model-free designs. Extending these results to more complex distributions than Gaussians is an interesting direction for the future work.

Acknowledgment

The authors would like to thank Changho Suh, Fei Xia, and Jiantao Jiao for helpful discussions.

References

[1] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[2] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," 2016. [Online]. Available: arXiv:1609.04802.
[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," 2016. [Online]. Available: arXiv:1605.05396.
[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017. [Online]. Available: arXiv:1701.07875.
[5] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017. [Online]. Available: arXiv:1704.00028.
[6] M. Sanjabi, J. Ba, M. Razaviyayn, and J. D. Lee, "Solving approximate Wasserstein GANs to stationarity," 2018. [Online]. Available: arXiv:1802.08249.
[7] V. Nagarajan and J. Z. Kolter, "Gradient descent GAN optimization is locally stable," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5591–5600.
[8] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, "Stabilizing training of generative adversarial networks through regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2015–2025.
[9] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1823–1833.
[10] C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng, "Training GANs with optimism," 2017. [Online]. Available: arXiv:1711.00141.
[11] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," 2018. [Online]. Available: arXiv:1802.05957.
[12] H. Petzka, A. Fischer, and D. Lukovnicov, "On the regularization of Wasserstein GANs," 2017. [Online]. Available: arXiv:1709.08894.
[13] X. Guo, J. Hong, T. Lin, and N. Yang, "Relaxed Wasserstein with applications to GANs," 2017. [Online]. Available: arXiv:1705.07164.
[14] S. Nowozin, B. Cseke, and R. Tomioka, "F-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 271–279.
[15] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," 2015. [Online]. Available: arXiv:1505.03906.
[16] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1718–1727.
[17] X. Mao, Q. Li, H. Xie, R. Y. Lau, and Z. Wang, "Multi-class generative adversarial networks with the L2 loss function," 2016. [Online]. Available: arXiv:1611.04076.
[18] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017. [Online]. Available: arXiv:1703.10717.
[19] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (GANs)," 2017. [Online]. Available: arXiv:1703.00573.
[20] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
[21] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Stat.*, 2010, pp. 249–256.
[22] C. Villani, *Optimal Transport: Old and New*, vol. 338. New York, NY, USA: Springer, 2008.
[23] S. Liu, O. Bousquet, and K. Chaudhuri, "Approximation and convergence properties of generative adversarial learning," 2017. [Online]. Available: arXiv:1705.08991.
[24] J. W. Cannon and W. P. Thurston, "Group invariant PEANO curves," *Geometry Topol.* vol. 11, no. 3, pp. 1315–1355, 2007.
[25] G. Canas and L. Rosasco, "Learning probability measures with respect to optimal transport metrics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2492–2500.
[26] J. Weed and F. Bach, "Sharp asymptotic and finite-sample rates of convergence of empirical measures in Wasserstein distance," 2017. [Online]. Available: arXiv:1707.00087.
[27] T. Rippl, A. Munk, and A. Sturm, "Limit laws of the empirical Wasserstein distance: Gaussian distributions," *J. Multivariate Anal.*, vol. 151, pp. 90–109, Feb. 2016.
[28] A. Genevay, L. Chizat, F. Bach, M. Cuturi, and G. Peyré, "Sample complexity of Sinkhorn divergences," 2018. [Online]. Available: arXiv:1810.02733.
[29] F. Farnia and D. Tse, "A convex duality framework for GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5248–5258.
[30] Y. Freund and R. E. Schapire, "Game theory, on-line prediction and boosting," in *Proc. ACM 9th Annu. Conf. Comput. Learn. Theory*, 1996, pp. 325–332.
[31] R. Ge *et al.*, "Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2741–2750.
[32] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015. [Online]. Available: arXiv:1511.06434.