Generative Adversarial Networks: A Likelihood Ratio Approach

Kalliopi Basioti
Department of Computer Science
Rutgers University
Piscataway, New Jersey, USA
kalliopi.basioti@rutgers.edu

George V. Moustakides
Department of Electrical and Computer Engineering
University of Patras
Patras, Greece
moustaki@ece.upatras.gr

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533584

***Abstract*—We are interested in the design of generative networks. The training of these mathematical structures is mostly performed with the help of adversarial (min-max) optimization problems. We propose a simple methodology for constructing such problems assuring, at the same time, consistency of the corresponding solution. We give characteristic examples developed by our method, some of which can be recognized from other applications, and some are introduced here for the first time. We present a new metric, the likelihood ratio, that can be employed online to examine the convergence and stability during the training of different Generative Adversarial Networks (GANs). Finally, we compare various possibilities by applying them to well-known datasets using neural networks of different configurations and sizes.**

***Index Terms*—Generative Networks, Generative Adversarial Networks, Likelihood Ratio**

I. Introduction

The problem we are interested in, can be summarized as follows: We are given two collections of training data $\{\mathbf{z}_j\}$ and $\{\mathbf{x}_i\}$. In the first set the samples follow the origin probability density $h(\mathbf{z})$ and in the second the target density $f(\mathbf{x})$. The target density $f(\mathbf{x})$ is considered unknown while $h(\mathbf{z})$ can either be known with the possibility to produce samples $\mathbf{z}_j$ every time it is necessary or unknown in which case we have a second fixed training set $\{\mathbf{z}_j\}$. Our goal is to design a deterministic transformation $G(\mathbf{z})$ so that the data $\mathbf{y}_j$ produced by applying the transformation $\mathbf{y} = G(\mathbf{z})$ onto $\mathbf{z}_j$ follow the target density $f(\mathbf{y})$.

Of course one may wonder whether the proposed problem enjoys any solution, namely, whether there indeed exists a transformation $G(\mathbf{z})$ capable of transforming $\mathbf{z}$ into $\mathbf{y}$ with the former following the origin density $h(\mathbf{z})$ and the latter the target density $f(\mathbf{y})$. The problem of transforming random vectors has been analyzed in [1] where existence is shown under general conditions. Computing, however, the actual transformation is a completely different challenge with one of the possible solutions relying on adversarial approaches applied to neural networks.

The most well known usage of this result is the possibility to generate synthetic data that follow the unknown target density $f(\mathbf{x})$. In this case $h(\mathbf{z})$ is selected to be simple (e.g. i.i.d. standard Gaussian or i.i.d. uniform) so that generating realizations from $h(\mathbf{z})$ is straightforward. As mentioned, the adversarial approach can be applied even if the origin density $h(\mathbf{z})$ is unknown provided that we have a dataset $\{\mathbf{z}_j\}$ with data following the origin density.

It was [2] that first introduced the idea of adversarial (min-max) optimization and demonstrated that it results in the determination of the desired transformation $G(\mathbf{z})$ (consistency). Alternative adversarial approaches were subsequently suggested by [3], [4] and also shown to deliver the correct transformation $G(\mathbf{z})$.

In the work of [5] a class of min-max optimizations, f-GANs, was defined to design generator/discriminator pairs. Then, [6] defined the adversarial divergences class of objective functions, which further combined f-GANs, MMD-GAN [7], Wasserstein GANs (WGANs) [3], WGAN with Gradient Penalty [8], and entropic regularized optimal transport problems. Next, the work of [9] connected f-GANs, and WGANs, and later [10] generalized the results by introducing the $(f, \Gamma)-$divergencies, which allowed to bridge f-divergencies and integral probability metrics. A drawback of the f-GANs based families is that the derivation of a loss function requires the solution of an additional optimization problem, making it challenging to discover new GANs losses. Our work will show that our methods provide an abundance of adversarial problems capable of identifying the appropriate transformation $G(\mathbf{z})$ without additional optimization to find new losses. Instead, we will provide a simple recipe as to how we can successfully construct such problems.

Arguing along the same lines of the existing min-max formulations: We would like to optimally specify a vector transformation $G(\mathbf{z})$, the generator, and a scalar function $D(\mathbf{x})$, the discriminator. To achieve this, for each combination $\{G(\mathbf{z}), D(\mathbf{x})\}$ we define the cost function

$$J(G, D) = \mathbb{E}_{\mathbf{x}\sim f}[\phi(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim h}[\psi(D(G(\mathbf{z})))] \quad (1)$$

where $\phi(z), \psi(z)$ are two scalar functions of the scalar z. The optimum combination generator/discriminator is then identified by solving the following min-max problem

$$\min_{G(\mathbf{z})} \max_{D(\mathbf{x})} J(G, D) \quad (2)$$

This work was supported by the US National Science Foundation, Grant CIF 1513373, through Rutgers University.

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

We must point out that our goal is not to solve (2), but rather find a class of functions $\phi(z), \psi(z)$ so that the transformation $G(\mathbf{z})$ that will come out of the solution of equation 2 is such that $\mathbf{y} = G(\mathbf{z})$ follows the target density $f(\mathbf{y})$ when $\mathbf{z}$ follows the origin density $h(\mathbf{z})$.

If $\mathbf{z}$ is random following $h(\mathbf{z})$ then $\mathbf{y} = G(\mathbf{z})$ is also random and we denote with $g(\mathbf{y})$ its corresponding probability density. Clearly, there exists a correspondence between transformations $G(\mathbf{z})$ and densities $g(\mathbf{y})$ when the density $h(\mathbf{z})$ of $\mathbf{z}$ is fixed. Since we can write $\mathbb{E}_{\mathbf{z}\sim h}[\psi(D(G(\mathbf{z})))] = \mathbb{E}_{\mathbf{y}\sim g}[\psi(D(\mathbf{y}))]$ this allows us to argue that the min-max problem in (2) is equivalent to

$$\min_{g(\mathbf{y})} \max_{D(\mathbf{x})} \{\mathbb{E}_{\mathbf{x}\sim f}[\phi(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{y}\sim g}[\psi(D(\mathbf{y}))]\} \tag{3}$$

It is now possible to combine the two expectations by applying a change of measure and a change of variables and equivalently write (3) as follows:

$$\begin{aligned}
&\min_{g(\mathbf{y})} \max_{D(\mathbf{x})} \left\{ \mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x})\big)\big] + \int \psi\big(D(x)\big) \frac{g(x)}{f(x)} f(x) dx \right\} \\
&= \min_{g(\mathbf{x})} \max_{D(\mathbf{x})} \left\{ \mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x})\big)\big] + \mathbb{E}_{\mathbf{x}\sim f}\big[r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\big] \right\} \\
&= \min_{g(\mathbf{x})} \max_{D(\mathbf{x})} \mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\big]
\end{aligned} \tag{4}$$

where $r(\mathbf{x}) = g(\mathbf{x})/f(\mathbf{x})$ denotes the corresponding likelihood ratio. Since $f(\mathbf{x})$ is also fixed, there is again a correspondence between $r(\mathbf{x})$ and $g(\mathbf{x})$, hence the previous min-max problem becomes equivalent to

$$\min_{r(\mathbf{x})\in\mathcal{L}_f} \max_{D(\mathbf{x})} \mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\big] \tag{5}$$

Here $\mathcal{L}_f$ denotes the class of all likelihood ratios $r(\mathbf{x})$ with respect to the density $f(\mathbf{x})$, namely, all the functions $r(\mathbf{x})$ that satisfy

$$\mathcal{L}_f = \left\{ r(\mathbf{x}) : r(\mathbf{x}) \geq 0, \int r(\mathbf{x}) f(\mathbf{x}) d\mathbf{x} = 1 \right\} \tag{6}$$

Using these definitions, let us define the cost

$$J(r, D) = \mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\big] \tag{7}$$

and according to (4), we are interested in the following min-max problem

$$\min_{r(\mathbf{x})\in\mathcal{L}_f} \max_{D(\mathbf{x})} J(r, D) \tag{8}$$

As mentioned, our actual goal is not to solve the adversarial problem. Instead, we would like to properly identify pairs of functions $\{\phi(z), \psi(z)\}$ so that (8) accepts as solution the function $r(\mathbf{x}) = 1$. Indeed, if $r(\mathbf{x}) = 1$ is the solution of (8), this means that $g(\mathbf{x}) = f(\mathbf{x})$ is the solution to (3) and finally the optimum $G(\mathbf{x})$ obtained from (1) is such that $\mathbf{y} = G(\mathbf{x})$ follows $g(\mathbf{y}) = f(\mathbf{y})$ which, of course, is our original objective. Even though the min-max problem in (1) is what we attempt to solve, it is through (8) that we understand what its solution entails. In the next section we focus on (7), (8) and propose a simple design method (recipe) for the two functions $\phi(z), \psi(z)$ that assures that the solution (8) is indeed $r(\mathbf{x}) = 1$.

Before we discuss the details of our work, we would like to summarize this paper's contribution.

- We design a family of GANs problems using a likelihood ratio approach. In this class, all optimization problems have the desired property that the generator output follows the target distribution of the random vector of interest, $\mathbf{x}$, in other words, that the likelihood ratio of the two distributions is equal to one.
- We propose a straightforward *recipe* to explore the GANs family. With this methodology, we were able to identify subclasses in the GANs family characterized by specific transformations of the likelihood ratio. We identified known GANs (such as vanilla [2] and Wasserstein GANs) and discovered novel ones in these subclasses.
- We propose a new *online* metric, the likelihood ratio, for evaluating the performance of GANs during training.
- Our experiments provide insights for the different GANs' objective functions' behavior, with some novel objective functions performing better than the already known GANs.

II. A class of Functions $\phi(z), \psi(z)$

Suppose that $\omega(r)$ is a *strictly increasing and (left and right) differentiable* scalar function of the nonnegative scalar r, i.e. $r \in [0, \infty)$. Denote with $J_\omega = \omega\big([0, \infty)\big)$ the range of values of $\omega(r)$ and let $\omega^{-1}(z)$ be the inverse function of $\omega(r)$ which exists and is defined for $z \in J_\omega$. Let $\rho(z) > 0$ be a positive scalar function also defined for $z \in J_\omega$ then, using $\omega(r)$ and $\rho(z)$, we propose the following pair $\phi(z), \psi(z)$

$$\phi'(z) = -\omega^{-1}(z)\rho(z), \quad \psi'(z) = \rho(z), \tag{9}$$

where "$'$" denotes derivative. Since $\omega(r)$ and $\rho(z)$ are arbitrary (provided they satisfy the strict increase and positivity constraint respectively), the class of pairs defined by (9) is very rich allowing for a multitude of choices. We show next that *any* such pair $\{\phi(z), \psi(z)\}$ gives rise to a min-max problem, as in (8), that accepts $r(\mathbf{x}) = 1$ as its unique solution. We prove this claim in two steps. The first, involves a theorem where we consider a simplified version of the min-max problem.

Theorem 1: Let $\omega(r), \phi(z), \psi(z)$ and J_ω be defined as above with the additional constraint $\psi\big(\omega(1)\big) = 0$. Fix $r \geq 0$ and consider $\phi(D) + r\psi(D)$ as a function of the scalar D. Then, for any $D \in J_\omega$, we have that

$$\phi(D) + r\psi(D) \leq \phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big), \tag{10}$$

with equality if and only if $D = \omega(r)$.

Consider next the minimization with respect to r of the maximal value in (10). It is then true that

$$\min_{r\geq 0} \left\{ \phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big) \right\} = \phi\big(\omega(1)\big), \tag{11}$$

with equality if and only if $r = 1$.

Proof: We note that the constraint $\psi\big(\omega(1)\big) = 0$ does not affect the generality of our class of functions since from (9) we

have that $\psi(z)$, after integration, is defined up to an arbitrary additive constant. We can always select this constant so that the constraint is satisfied. We would also like to emphasize that this constraint is needed only for the proof of this theorem and it is not necessary for the corresponding min-max problem defined in (8).

For fixed r, to find the maximum of $\phi(D) + r\psi(D)$ we consider the derivative with respect to D which, using (9), takes the form

$$\phi'(D) + r\psi'(D) = \big(r - \omega^{-1}(D)\big)\rho(D). \tag{12}$$

The strict increase of $\omega(r)$ is inherited by its inverse function $\omega^{-1}(z)$ which, combined with the positivity of $\rho(z)$, implies that the previous expression has the same sign as $r-\omega^{-1}(D)$ or $\omega(r)-D$. Consequently $D=\omega(r)$ is the *only critical point* of $\phi(D)+r\psi(D)$ which is a global maximum. Of course there are possibilities for extrema at the two end points of J_ω but they can only be (local) minima.

Let us now focus on the resulting function $\phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big)$. Taking its derivative with respect to r yields

$$\begin{aligned}\big\{\phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big)\big\}' = \big\{\phi'\big(\omega(r)\big) + r\psi'\big(\omega(r)\big)\big\}\,\omega'(r)\\ + \psi\big(\omega(r)\big) = \psi\big(\omega(r)\big),\end{aligned} \tag{13}$$

where the last equality is due to the specific definition of the two functions $\phi(z), \psi(z)$ in (9). Since $\psi'(z)=\rho(z)>0$, this implies that $\psi(z)$ is strictly increasing, being also the integral of $\rho(z)$ it is continuous in z. If we combine this property with the strict increase and continuity (as a result of left and right differentiability) of $\omega(r)$ we conclude that $\psi\big(\omega(r)\big)$ is also strictly increasing and continuous in r. We recall that $\psi(z)$ is selected to satisfy $\psi\big(\omega(1)\big)=0$, consequently for $r=1$ the function $\phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big)$ has a unique minimum which is global and no other critical points. Of course it can still exhibit extrema at $r=0$ and/or $r\to\infty$ but they can only be (local) maxima. ∎

A consequence of Theorem 1 is the next corollary, which constitutes the second and final step in proving that the adversarial problem defined in (8) has as unique solution the function $r(\mathbf{x})=1$.

Corollary 1: If the functions $\phi(z), \psi(z)$ satisfy (9) and $\omega(r)$ is strictly increasing and left and right differentiable, then in the adversarial problem defined in (8) the maximizer is $D(\mathbf{x}) = \omega\big(r(\mathbf{x})\big)$ and the minimizer is $r(\mathbf{x})=1$, while the resulting min-max value is equal to

$$\begin{aligned}\min_{r(\mathbf{x})\in\mathcal{L}_f}\max_{D(\mathbf{x})}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\right]\\ = \phi\big(\omega(1)\big) + \psi\big(\omega(1)\big).\end{aligned} \tag{14}$$

Proof: First, we observe that

$$\begin{aligned}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\right]\\ = \mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\tilde{\psi}\big(D(\mathbf{x})\big)\right] + \psi\big(\omega(1)\big)\end{aligned} \tag{15}$$

with the last equality being true since $\mathbb{E}_{\mathbf{x}\sim f}[r(\mathbf{x})] = 1$ and where $\tilde{\psi}(z) = \psi(z) - \psi\big(\omega(1)\big)$. We start with the maximization problem. Since $D(\mathbf{x})$ is a function of $\mathbf{x}$ we have

$$\begin{aligned}\max_{D(\mathbf{x})}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\tilde{\psi}\big(D(\mathbf{x})\big)\right]\\ = \mathbb{E}_{\mathbf{x}\sim f}\left[\max_{D(\mathbf{x})}\left\{\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\tilde{\psi}\big(D(\mathbf{x})\big)\right\}\right].\end{aligned} \tag{16}$$

The maximization under the expectation can be performed for each fixed $\mathbf{x}$. However, when we fix $\mathbf{x}$ then $r(\mathbf{x})$ becomes a constant and the result of the maximization depends only on the actual value of $r(\mathbf{x})$. This suggests that we can limit ourselves to functions of the form $D(\mathbf{x}) = D\big(r(\mathbf{x})\big)$. After this observation we can drop the dependence on $\mathbf{x}$ and perform, equivalently, the maximization $\max_D\left\{\phi\big(D(r)\big) + r\tilde{\psi}\big(D(r)\big)\right\}$ for each fixed r. The pair $\{\phi(z), \tilde{\psi}(z)\}$ satisfies the assumptions of Theorem 1, therefore maximization is achieved for $D(r)=\omega(r)$. This implies that

$$\begin{aligned}\max_{D(\mathbf{x})}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\psi\big(D(\mathbf{x})\big)\right] =\\ \mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(\omega\big(r(\mathbf{x})\big)\big) + r(\mathbf{x})\tilde{\psi}\big(\omega\big(r(\mathbf{x})\big)\big)\right] + \psi\big(\omega(1)\big).\end{aligned} \tag{17}$$

We can now continue in a similar way for the minimization problem. Specifically

$$\begin{aligned}&\min_{r(\mathbf{x})\in\mathcal{L}_f}\max_{D(\mathbf{x})}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(D(\mathbf{x})\big) + r(\mathbf{x})\tilde{\psi}\big(D(\mathbf{x})\big)\right]\\ &\quad= \min_{r(\mathbf{x})\in\mathcal{L}_f}\mathbb{E}_{\mathbf{x}\sim f}\left[\phi\big(\omega\big(r(\mathbf{x})\big)\big) + r(\mathbf{x})\tilde{\psi}\big(\omega\big(r(\mathbf{x})\big)\big)\right]\\ &\quad\geq \mathbb{E}_{\mathbf{x}\sim f}\left[\min_{r(\mathbf{x})\in\mathcal{L}_f}\left\{\phi\big(\omega\big(r(\mathbf{x})\big)\big) + r(\mathbf{x})\tilde{\psi}\big(\omega\big(r(\mathbf{x})\big)\big)\right\}\right]\\ &\quad\geq \mathbb{E}_{\mathbf{x}\sim f}\left[\min_{r}\left\{\phi\big(\omega(r)\big) + r\tilde{\psi}\big(\omega(r)\big)\right\}\right] = \phi\big(\omega(1)\big)\end{aligned} \tag{18}$$

with the last inequality being true since the minimization that follows is unconstrained and the last equality being a consequence of Theorem 1. The final lower bound is clearly attained by $r(\mathbf{x})=1$, which is also a legitimate solution of the constrained minimization, since $r(\mathbf{x})=1$ belongs to the class $\mathcal{L}_f$ of likelihood ratios. Consequently $r(\mathbf{x})=1$ is the solution to the min-max problem. Returning to the original min-max setup with $\psi(z)$ replacing $\tilde{\psi}(z)$, we can clearly see that it satisfies (14). This completes the proof. ∎

Remark 1: The adversarial problem is defined with the help of the two functions $\phi(z), \psi(z)$ which, according to (9), can be obtained by integrating the corresponding derivatives. However, this integration might not always be possible, analytically. As we will have the chance to confirm in Section III, in an actual optimization algorithm (e.g. of gradient type) that solves (2), the exact form of $\phi(z), \psi(z)$ is not necessary. Instead, what is required is their derivatives which are analytically available from (9).

We must emphasize that there already exists the significant work by [5] that addresses a similar problem as our current work, namely the definition of a class of min-max optimizations that can be used to design the generator/discriminator

pair. The class in [5] is defined in terms of a convex function $f(r)$ which can be shown to correspond to the outcome of our maximization, namely the function $\phi\big(\omega(r)\big) + r\psi\big(\omega(r)\big)$. This establishes a one-to-one correspondence between the two methods under the ideal (non data-driven) setup. However, we believe that, our approach enjoys certain significant advantages:

First, the definition of the two functions $\phi(z), \psi(z)$ in (3) is straightforward while in [5] requires the solution of an optimization problem.

Second, in our case we have complete control over the result of the maximization problem that defines the discriminator. In other words we can decide what transformation $\omega(r)$ of the likelihood ratio r, the discriminator must estimate. In [5] such flexibility does not exist.

Controlling the function we estimate with the discriminator plays a significant role in the implementation of our method. Indeed when we use a neural network to approximate the optimum discriminator, this affects the overall quality of the resulting generator/discriminator pair. We should also note that there are important applications in Statistics where one is interested in estimating only the transformation of the likelihood ratio, with the most common cases being the likelihood ratio itself, its logarithm (log-likelihood ratio), or the ratio $\frac{r}{1+r}$ which plays the role of the posterior probability between two densities. In other words, there are applications where one is interested only in the "max" part of the min-max problem. In fact, in the next section we give examples of various choices of $\omega(r)$ and mention problems where the discriminator function becomes the actual target and not the generator.

A. Subclasses of the GAN Family

Subclass A: $\omega(r) = r^{\alpha}$

The first subclass we examine is the simplest one, consisting of just powers of the likelihood ratio. We should mention that this is the first work proposing objective functions from this class. To find the pairs $\{\phi(z), \psi(z)\}$ we proceed as follows.

We have that $\omega^{-1}(z) = z^{\frac{1}{\alpha}}$ and $J_{\omega} = [0, \infty)$. According to (9), for $z \in [0, \infty)$ we must define $\phi'(z) = -z^{\frac{1}{\alpha}}\rho(z), \;\; \psi'(z) = \rho(z)$. The likelihood ratio with respect to the Discriminator function and the parameter a is $r = D^{-a}$. The following examples can be shown to satisfy these equations.

A1) If we select $\rho(z) = z^{\beta}$, with $\beta \neq -1, -1 - \frac{1}{\alpha}$, this yields $\phi(z) = -\frac{z^{1+\frac{1}{\alpha}+\beta}}{1+\frac{1}{\alpha}+\beta}$ and $\psi(z) = \frac{z^{1+\beta}}{1+\beta}$. For $\beta = -1$, $\rho(z) = z^{-1}$, $\phi(z) = -\alpha z^{\frac{1}{\alpha}}$, $\psi(z) = \log z$. For $\beta = -1 - \frac{1}{\alpha}$, $\rho(z) = z^{-1-\frac{1}{\alpha}}$, $\phi(z) = -\log z$, $\psi(z) = -\alpha z^{-\frac{1}{\alpha}}$.

A2) If we select $\alpha = 1$, $\rho(z) = \frac{1}{(1+z)}$ then, $\phi(z) = -(1+z)$ and $\psi(z) = -(1 + z^{-1})$.

A3) If we select $\alpha = 1$, $\rho(z) = \frac{1}{(1+z)z}$ then, $\phi(z) = -\log(1+z)$ and $\psi(z) = -\log(1 + z^{-1})$.

For the particular selection $\omega(r) = r$ (corresponding to $\alpha = 1$) we can show that the resulting cost is equivalent to the Bregman cost [11]. In fact there is a one-to-one correspondence between our $\rho(z)$ function and the function that defines the Bregman cost. This correspondence however is lost once we switch to a different α or a different $\omega(r)$ function, suggesting that the proposed class of pairs $\{\phi(z), \psi(z)\}$, is far richer than the class induced by the Bregman cost.

We should mention that in A1) the selection $\alpha = 1, \beta = 0$ is known as the mean square error criterion and if we apply only the maximization problem then this corresponds to a likelihood ratio estimation technique proposed in the literature by [12]. We will refer to this case as the *MSE* GAN.

Subclass B: $\omega(r) = \alpha^{-1}\log r$

This subclass considers one of the most popular transformations of the likelihood ratio, the log-likelihood ratio. As in the first subclass, for the first time, the next examples are presented. They can be used either under a min-max setting, for the determination of the generator/discriminator pair, or under a pure maximization setting for the direct estimation of the log-likelihood ratio function $\log r(\mathbf{x})$.

We have $\omega^{-1}(z) = e^{\alpha z}$ and $J_{\omega} = \mathbb{R}$. As before $\rho(z)$ must be strictly positive and, according to (9), for all real z we must define $\phi'(z) = -e^{\alpha z}\rho(z), \;\; \psi'(z) = \rho(z)$. Then the likelihood ratio is given by $r = e^{aD}$. The following examples satisfy these equations.

B1) If $\rho(z) = e^{-\beta z}$ with $\beta \neq 0, \alpha$, this produces $\phi(z) = -\frac{e^{(\alpha-\beta)z}}{\alpha-\beta}, \;\; \psi(z) = -\frac{e^{-\beta z}}{\beta}$. If $\beta = 0$ then $\rho(z) = 1$, $\phi(z) = -\frac{e^{\alpha z}}{\alpha}$, $\psi(z) = z$. If $\beta = \alpha$ then $\rho(z) = e^{-\alpha z}$, $\phi(z) = -z$ and $\psi(z) = -\frac{e^{-\alpha z}}{\alpha}$. We call the $\alpha = 1, \beta = 0.5$ case the *Exponential* GAN.

B2) If $\alpha = 1$, $\rho(z) = \frac{1}{1+e^{z}}$ then, $\phi(z) = -\log(1 + e^{z})$ and $\psi(z) = -\log(1 + e^{-z})$.

Subclass C: $\omega(r) = \frac{r}{r+1}$

As we already mentioned, this is another important transform of the likelihood ratio. Interestingly, in this subclass belongs the first introduced GAN [2] the *Cross Entropy* GAN.

When $\omega(r) = \frac{r}{r+1}$ we have $\omega^{-1}(z) = \frac{z}{1-z}$ and $J_{\omega} = [0, 1]$. For $\rho(z) > 0, z \in [0, 1]$ we must define the functions $\phi(z), \psi(z)$ according to (9) $\phi'(z) = -\frac{z}{1-z}\rho(z), \;\; \psi'(z) = \rho(z)$. In this case the likelihood ratio is $r = \frac{D}{1-D}$. The next set of examples can be seen to satisfy these equations.

C1) If we select $\rho(z) = \frac{1}{z}$, this yields $\phi(z) = \log(1 - z)$ and $\psi(z) = \log z$.

C2) Selecting $\rho(z) = (1 - z)^{\alpha}$, with $\alpha \neq 0, -1$, yields $\phi(z) = -\frac{1}{1+\alpha}(1-z)^{\alpha+1} + \frac{1}{\alpha}(1-z)^{\alpha}$ and $\psi(z) = -\frac{1}{1+\alpha}(1 - z)^{1+\alpha}$. For $\alpha = 0$, we have $\rho(z) = 1$ and $\phi(z) = z + \log(1-z)$, $\psi(z) = z$, while for $\alpha = -1$ we have $\rho(z) = \frac{1}{1-z}$ and $\phi(z) = -\log(1 - z) - \frac{1}{1-z}$, $\psi(z) = -\log(1 - z)$.

In C1) we recognize the functions used in the original article by [2]. C2) appears for the first time.

Subclass D: $\omega(r) = \text{sign}(\log r)$

This is a special case of $\omega(r)$ with the corresponding function not being strictly increasing. It turns out that we can still come up with optimization problems, two of which are known and used in practice, by considering $\omega(r)$ as a *limit* of a sequence of strictly increasing functions.

Monotone Loss: As a first approximation we propose $\text{sign}(z) \approx \tanh(\frac{c}{2}z)$ where $c > 0$ a parameter. We note that $\lim_{c\to\infty} \tanh(\frac{c}{2}z) = \text{sign}(z)$. Using this approximation we can write

$$\text{sign}(\log r) \approx \tanh\left(\frac{c}{2}\log r\right) = \frac{r^c - 1}{r^c + 1} = \omega(r). \qquad (19)$$

As we mentioned, we have exact equality for $c \to \infty$. Let us perform our analysis by assuming that c is finite. We note that $\omega^{-1}(z) = (\frac{1+z}{1-z})^{\frac{1}{c}}$ and $J_\omega = [-1, 1]$. Consequently, if $\rho(z) > 0$ for $z \in [-1, 1]$, we must define $\phi'(z) = -\left(\frac{1+z}{1-z}\right)^{\frac{1}{c}} \rho(z),\ \psi'(z) = \rho(z)$. *We notice that since $c \to \infty$ we cannot find the likelihood ratio in terms of the Discriminator function.*

D1) If we let $c \to \infty$ in order to converge to the desired sign function, this yields $\phi'(z) = -\rho(z)$ and $\psi'(z) = \rho(z)$. This suggests that $\phi(z) = -\int^z \rho(x)dx$ is decreasing and $\psi(z) = \int^z \rho(x)dx = -\phi(z)$ is increasing. In fact any strictly increasing function $\psi(z)$ can be adopted provided we select $\phi(z) = -\psi(z)$.

There is a popular combination that falls under Case D1). In particular, the selection $\psi(z) = z = -\phi(z)$ reminds us of Wasserstein GAN [3], with two differences, in our case z should lie in $[-1, 1]$ and the discriminator is not constrained to be a Lipschitz function.

Hinge Loss: As a second approximation we use the expression $\text{sign}(z) \approx \text{sign}(z)|z|^{\frac{1}{c}},\quad c > 0$, which is strictly increasing, continuous and converges to $\text{sgn}(z)$ as $c \to \infty$. This suggests that

$$\text{sign}(\log r) \approx \text{sign}(\log r)|\log r|^{\frac{1}{c}} = \omega(r), \qquad (20)$$

and $\omega^{-1}(z) = e^{z^c}$. Since $\omega(r)$ can assume any real value we conclude that $J_\omega = \mathbb{R}$ which, clearly, differs from the previous approximation where we had $J_\omega = [-1, 1]$. If $\rho(z) > 0, z \in \mathbb{R}$ then, according to (9) we must define $\phi'(z) = -e^{z^c}\rho(z),\ \psi'(z) = \rho(z)$. We present the following case that leads to a very well known pair from a completely different application.

D2) If we select $\psi'(z) = \rho(z) = \{e^{-|z|^{\frac{1}{c}}} + \mathbb{1}_{z<-1}\} > 0$ then $\phi'(z) = -e^{z^{\frac{1}{c}}}\{e^{-|z|^{\frac{1}{c}}} + \mathbb{1}_{z<-1}\}$. If we now let $c \to \infty$, we obtain the limiting form for the derivatives which become $\psi'(z) = -\mathbb{1}_{z<1}$ and $\phi'(z) = \mathbb{1}_{z>-1}$. By integrating we arrive at $\phi(z) = -\max\{1 + z, 0\}$ and $\psi(z) = -\max\{1 - z, 0\}$. The cost based on this particular pair is called the *hinge loss* [13] and it is very popular in binary classification where one is interested only in the maximization problem. The corresponding method is known to exhibit an overall performance which in practice is considered among the best [14], [15]. Here, as in [16], we propose the hinge loss as a means to perform adversarial optimization for the design of the generator $G(\mathbf{x})$.

This completes our presentation of examples. However, we must emphasize, that these are only a few illustrations of possible pairs $\{\phi(z), \psi(z)\}$ one can construct. Indeed combining, as dictated by (9), any strictly increasing function $\omega(r)$ with any positive function $\rho(z)$ generates a legitimate pair $\{\phi(z), \psi(z)\}$ and a corresponding min-max problem (8) that enjoys the desired solution $r(\mathbf{x}) = 1$. Finally, in Table I we summarize some of the GANs presented above which we will use later in our experiments.

TABLE I
OPTIMIZATION PROBLEMS FOR GANS

GAN	$\phi(z)$	$\psi(z)$	J_ω
A1a	$-z$	$\log(z)$	$[0, \infty)$
A1b	$-\log(z)$	$-z^{-1}$	$[0, \infty)$
A2	$-(1 + z)$	$-(1 + z^{-1})$	$[0, \infty)$
A3	$-\log(1 + z)$	$-\log(1 + z^{-1})$	$[0, \infty)$
MSE	$-0.5z^2$	z	$[0, \infty)$
B1a	$-e^z$	e^z	$\mathbb{R}$
B1b	$-z$	$-e^{-z}$	$\mathbb{R}$
Exponential	$-e^{0.5z}$	$-e^{-0.5z}$	$\mathbb{R}$
B2	$-\log(1 + e^z)$	$-\log(1 + e^{-z})$	$\mathbb{R}$
Cross Entropy	$\log(1 - z)$	$\log(z)$	$[0, 1]$
C2	$z + \log(1 - z)$	z	$[0, 1]$
Hinge	$-(1 + z)_+$	$-(1 - z)_+$	$\mathbb{R}$
Wasserstein	z	$-z$	$\mathbb{R}$

III. Data-Driven Setup and Neural Networks

Let us now consider the data-driven version of the problem. As mentioned, the target density $f(\mathbf{x})$ is unknown. Instead we are given a collection of realizations $\{\mathbf{x}_i\}$ that follow $f(\mathbf{x})$ and a second collection $\{\mathbf{z}_j\}$ that follows the origin density $h(\mathbf{z})$. These data constitute our training set. Regarding the second set $\{\mathbf{z}_j\}$ it can either become available "on the fly" when $h(\mathbf{z})$ is known by generating realizations every time they are needed, or it can be considered fixed from the start exactly as $\{\mathbf{x}_i\}$, if $h(\mathbf{z})$ is also unknown. As we pointed out in Section I, we are interested in designing a generator $G(\mathbf{z})$ so that when we apply it onto the data $\mathbf{z}_j$, that is, $\mathbf{y}_j = G(\mathbf{z}_j)$ the resulting $\mathbf{y}_j$ will follow a density that matches the target density $f(\mathbf{x})$. Since we are now considering the data-driven version of the problem, we are going to limit $G(\mathbf{z}), D(\mathbf{x})$ to be the outputs of corresponding neural networks. Therefore the generator is replaced by $G(\mathbf{z}, \theta)$ while the discriminator by $D(\mathbf{x}, \vartheta)$ where θ, ϑ summarize the parameters of the two neural networks.

Once we have selected our favorite $\omega(r)$ and $\rho(z)$ functions we can compute from (9) the functions $\phi(z), \psi(z)$ that enter into the min-max problem defined in (2). This problem, after limiting the generator and discriminator to neural networks, can be rewritten as follows

$$\min_\theta \max_\vartheta J(\theta, \vartheta) = \min_\theta \max_\vartheta \big\{\mathbb{E}_{\mathbf{x}\sim f}\big[\phi\big(D(\mathbf{x}, \vartheta)\big)\big] + \mathbb{E}_{\mathbf{z}\sim h}\big[\psi\big(D(G(\mathbf{z}, \theta), \vartheta)\big)\big]\big\}. \qquad (21)$$

If θ_o, ϑ_o are the corresponding optimum parameter values, and the structure of the two networks is sufficiently rich, we expect that $G(\mathbf{z}, \theta_o), D(\mathbf{x}, \vartheta_o)$ will approximate the optimum functions $D(\mathbf{x}), G(\mathbf{z})$ of the ideal problem in (2) respectively. In particular for θ_o, the generator $G(\mathbf{z}, \theta_o)$, whenever applied onto any $\mathbf{z}_j$ that follows $h(\mathbf{z})$, it will result in a $\mathbf{y}_j = G(\mathbf{z}_j, \theta_o)$ that follows a density which is expected to be close to the target density $f(\mathbf{y})$.

TABLE II
KID AND FID SCORES

GANs	CARS		CELEBA		CIFAR10		MNIST	
	KID	FID	KID	FID	KID	FID	KID	FID
	$\times 10^{-6} \pm \times 10^{-12}$		$\times 10^{-6} \pm \times 10^{-12}$		$\times 10^{-6} \pm \times 10^{-12}$		$\times 10^{-4} \pm \times 10^{-8}$	
A1a	4.03 ± 2.45	23.30 ± 0.10	2.57 ± 5.05	7.53 ± 0.02	2.75 ± 8.47	9.67 ± 0.01	7.80 ± 5.86	2.18 ± 0.01
A1b	3.33 ± 1.89	24.22 ± 0.22	2.67 ± 8.35	7.68 ± 0.04	2.36 ± 5.86	9.69 ± 0.04	6.30 ± 4.32	2.13 ± 0.01
A2	4.36 ± 3.18	24.40 ± 0.04	3.38 ± 9.68	7.62 ± 0.04	3.09 ± 8.35	9.53 ± 0.04	8.79 ± 4.65	2.17 ± 0.01
A3	4.42 ± 2.49	23.64 ± 0.23	7.29 ± 5.48	8.50 ± 0.06	2.40 ± 4.95	9.72 ± 0.04	8.53 ± 7.35	2.15 ± 0.01
MSE	3.79 ± 1.71	23.99 ± 0.04	2.55 ± 9.34	7.66 ± 0.07	2.09 ± 5.63	9.61 ± 0.03	8.64 ± 7.51	2.15 ± 0.01
B1a	6.17 ± 3.62	23.60 ± 0.12	5.56 ± 9.78	8.06 ± 0.02	2.35 ± 5.52	9.67 ± 0.05	8.05 ± 3.30	2.13 ± 0.01
B1b	5.24 ± 2.27	23.91 ± 0.07	7.32 ± 1.18	8.07 ± 0.06	2.63 ± 8.42	9.67 ± 0.04	6.85 ± 3.71	2.18 ± 0.01
Exponential	5.49 ± 2.86	23.52 ± 0.06	10.05 ± 6.10	9.15 ± 0.03	2.81 ± 8.34	9.79 ± 0.03	6.88 ± 3.41	2.13 ± 0.01
B2	7.06 ± 3.40	23.46 ± 0.12	12.48 ± 7.38	9.39 ± 0.06	2.47 ± 5.42	9.79 ± 0.07	8.82 ± 3.69	2.09 ± 0.01
Cross-Entropy	13.09 ± 8.95	25.39 ± 0.16	3.53 ± 12.65	$7.53 + 0.03$	2.53 ± 5.58	9.72 ± 0.09	6.16 ± 5.37	2.08 ± 0.01
C2	8.45 ± 2.92	24.36 ± 0.12	3.11 ± 9.72	7.47 ± 0.01	4.17 ± 11.03	9.75 ± 0.03	7.72 ± 3.36	2.10 ± 0.01
Hinge	4.97 ± 3.50	22.88 ± 0.14	26.65 ± 16.81	10.91 ± 0.05	3.97 ± 9.42	9.99 ± 0.03	8.33 ± 4.16	2.16 ± 0.01
Wasserstein	4.92 ± 2.70	23.99 ± 0.04	20.59 ± 28.38	10.99 ± 0.04	1.74 ± 5.22	9.74 ± 0.02	7.33 ± 3.33	2.16 ± 0.01

Remark 2: When replacing $D(\mathbf{x}), G(\mathbf{z})$ with neural networks we must take special care of the corresponding outputs. Basically, we must guarantee that they are of the correct form. This is particularly important in the case of the scalar output $D(\mathbf{x}, \vartheta)$ of the discriminator. We recall that the optimum discriminator is $D(\mathbf{x}) = \omega\big(r(\mathbf{x})\big)$. This implies that we need to assure that $D(\mathbf{x}, \vartheta)$ takes values in J_ω (the range of $\omega(r)$). Consequently, we must apply the proper nonlinearity in the output of the discriminator that will guarantee this fact.

IV. EXPERIMENTS

In this section, we want to examine the performance of the GANs objectives presented in Table I for different datasets. For that reason we tested their performance on four different datasets, namely MNIST [17], CelebA [18], CIFAR-10 datasets [19], and Stanford Cars [20]. We recall that GANs are notorious for their nonrobust behavior [21]–[23]. For stabilizing the training process, we used the maximum gradient-penalty methodology, which was generalized to a class of Lipschitz GANs in [24].

For the generator, we used a four-layer neural network where the first layer is linear and the remaining deconvolutional; with ReLU activation functions between the layers except the final layer where we used the hyperbolic tangent function since the output is an image with pixel values in the range $[-1, 1]$. The image range was changed from $[0, 1]$ to $[-1, 1]$ because the hyperbolic tangent speeds the convergence of GANs compared to sigmoid (used for the $[0, 1]$ range). The generator input is a standard i.i.d. normal vector with dimension 64 for MNIST and 128 for CelebA, CIFAR-10, and Stanford Cars.

For the discriminator, we used a four-layer neural network with three convolutional layers followed by a linear layer. We applied Leaky ReLUs between the layers except for the final layer, where we adopted proper functions based on the range J_ω. For the training of the two neural networks we applied the Adam algorithm [25] with $\beta_1 = 0.5$, $\beta_2 = 0.9$, learning rate 10^{-4} and batch size 50 for MNIST and 64 for CelebA and CIFAR-10.

In Table II we present the final attained Frechet Inception Distances (FID) [26] and Kernel Inception Distances (KID) [4] scores after training for 2×10^5 Generator iterations. We make this distinction since the Discriminator performs five critic iterations for each Generator iteration. Our results indicate that some losses have inferior performance compared with the others. The losses that seem to attain poor FID/KID scores are the B2, Cross-Entropy, and C2 GANs in Stanford Cars and Exponential, B2, Hinge, and Wasserstein GANs in CelebA dataset. On the other hand, the A1a, A1b, A2, and Mean Square have excellent performance concerning all examined datasets. Furthermore, we notice that for simpler datasets (MNIST), the various GANs losses perform similarly. Still, when the dataset complexity increases (CelebA, CIFAR-10 Stanford Cars) and a small number of examples are available (only ~ 8000 samples available in the Stanford Cars dataset), visible performance variations emerge for different GANs.

To further investigate the poor performance of the GANs mentioned above, we tried different values for the hyperparameter λ of the maximum gradient penalty. As in [24] we tested the values $\{0.01, 0.1, 1, 10\}$. *In Table II, we used $\lambda = 10$ for all GANs since we obtained faster convergence for this value.* Indeed for $\lambda < 10$, the scores are improved for the B2, Cross-Entropy, and C2 GANs in Stanford Cars and Exponential, B2,

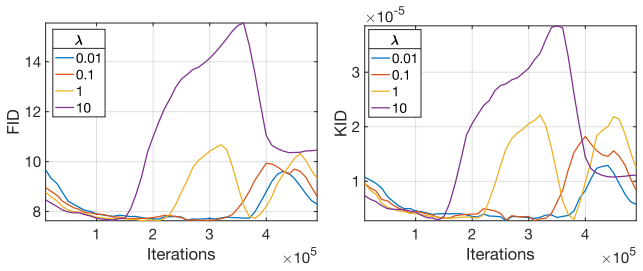

Fig. 1. Wasserstein GANs FID/KID scores during training for CelebA dataset.

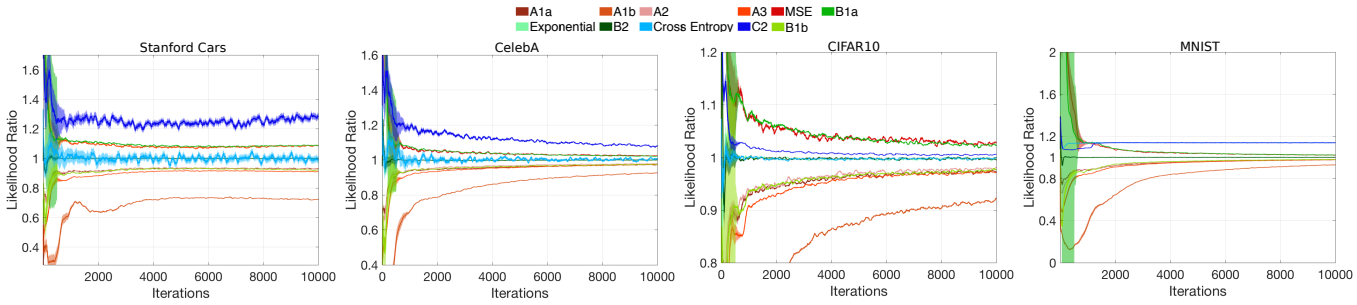

Fig. 2. Estimated Likelihood Ratio during training.

Fig. 3. Wasserstein GANs generated images for a - 110000, b - 210000, c - 310000 iterations for CelebA dataset.

Hinge, and Wasserstein GANs in CelebA dataset. But after continuing our training for more than 2×10^5 iterations, we experienced the same behavior. So the choice of λ was not diminishing this divergent behavior but rather postponing it. Fig. 1 shows an example of the previously described behavior for Wasserstein GANs with the CelebA dataset. Here for $\lambda = 10$ the divergent behavior starts a little bit earlier than 150000 iterations, for $\lambda = 1$ around 210000 iterations, for $\lambda = 0.1$ at 300000 iterations and for $\lambda = 0.01$ around 380000 iterations. Another interesting fact is that for the different values of λ, the best FID/KID scores are almost the same. So if we had to perform some early stopping in our training, it would be better to choose $\lambda = 10$ and avoid the additional iterations needed for $\lambda < 10$ to reach the same score. To qualitatively understand the effect of this behavior in Fig. 3 we present some synthetic examples of the Generator before the algorithm starts to diverge at 110000 iterations and later at 210000 and 310000 iterations (for $\lambda = 10$) where we notice the generation of blurry, distorted images.

As we mention in II-A we can compare the different GAN losses under the same quantity, the likelihood ratio function. In other words, we can examine if they converge to the optimal value, being equal to one. *Unfortunately, as we show in II-A for Hinge and Wasserstein GANs, we cannot estimate the likelihood ratio in terms of the Discriminator function.* For the other losses, it is possible to evaluate the likelihood function from the Discriminator's output employing dataset and Generator samples during training. In Fig. 2 we see the evolution of the likelihood function during the first 10000 Generator iterations where we used the training batches for its estimation. In all cases, we notice that B2, Exponential, Cross-Entropy (except MNIST) GANs converge faster to the optimum value. Interestingly additional conclusions are derived from these figures; although Cross-Entropy reaches the optimal value, it has higher variance around it than B1 and Exponential GAN (Stanford Cars and CelebA datasets). For instance, we could reduce the learning rate of the Cross-Entropy GAN to decrease the variance further.

Our simulations indicate that the Subclass A objectives A1a, A1b, A2, MSE have the best performance in terms of the computed metrics (hence image generation quality) and stability during training. Furthermore, the convergence to the optimal likelihood ratio between the dataset and the generator output can be estimated online and used as a metric to decide which GAN loss and learning rate to employ.

V. Conclusion

In this paper, we provided and demonstrated a straightforward methodology to determine loss functions that solve the generative adversarial problem. Our results suggest that there is no single loss function that achieves the best performance in terms of the examined metrics for all different datasets. This performance variation among loss functions becomes evident as the increasing complexity of the datasets that convolutes the generation task is better addressed by some loss functions that clearly outperform others. Specifically, in simpler datasets, such as MNIST, the evaluated loss functions yield very similar performance, whereas, in more intricate datasets like CelebA, CIFAR-10, and Stanford Cars, performance "gaps" between the different loss functions, and different subclasses, emerges. Our findings also propose that in every generation task, unexplored loss functions outperformed the previously proposed ones. Consequently, this function class is worth-exploring to identify new loss functions that can be used and evaluated in different applications. Our method provides a versatile tool that can be exploited in that direction.

REFERENCES

[1] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv:1406.2661*, 2014.

[3] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[4] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," *arXiv:1801.01401*, 2018.

[5] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," *arXiv:1606.00709*, 2016.

[6] S. Liu, O. Bousquet, and K. Chaudhuri, "Approximation and convergence properties of generative adversarial learning," *arXiv:1705.08991*, 2017.

[7] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "Mmd gan: Towards deeper understanding of moment matching network," *arXiv:1705.08584*, 2017.

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv:1704.00028*, 2017.

[9] J. Song and S. Ermon, "Bridging the gap between f-gans and wasserstein gans," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9078–9087.

[10] J. Birrell, P. Dupuis, M. A. Katsoulakis, Y. Pantazis, and L. Rey-Bellet, "(f, γ)-divergences: Interpolating between f-divergences and integral probability metrics," *arXiv:2011.05953*, 2020.

[11] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[12] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

[13] Y. Tang, "Deep learning using linear support vector machines," *arXiv:1306.0239*, 2013.

[14] L. Rosasco, E. D. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural computation*, vol. 16, no. 5, pp. 1063–1076, 2004.

[15] K. Janocha and W. M. Czarnecki, "On loss functions for deep neural networks in classification," *arXiv:1702.05659*, 2017.

[16] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv:1609.03126*, 2016.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[19] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[20] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *Proceedings of the IEEE international conference on computer vision workshops*, 2013, pp. 554–561.

[21] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 437–478.

[22] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[23] L. Mescheder, S. Nowozin, and A. Geiger, "The numerics of gans," *arXiv:1705.10461*, 2017.

[24] Z. Zhou, J. Liang, Y. Song, L. Yu, H. Wang, W. Zhang, Y. Yu, and Z. Zhang, "Lipschitz generative adversarial nets," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7584–7593.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[26] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *arXiv:1706.08500*, 2017.