Article

A Unifying Generator Loss Function for Generative Adversarial Networks

Justin Veiner [1], Fady Alajaji [1,*] and Bahman Gharesifard [2]

[1] Department of Mathematics and Statistics, Queen's University, Kingston, ON K7L 3N6, Canada; justin.veiner@queensu.ca
[2] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095, USA; gharesifard@ucla.edu
[*] Correspondence: fa@queensu.ca

Abstract: A unifying α-parametrized generator loss function is introduced for a dual-objective generative adversarial network (GAN) that uses a canonical (or classical) discriminator loss function such as the one in the original GAN (VanillaGAN) system. The generator loss function is based on a symmetric class probability estimation type function, $\mathcal{L}_\alpha$, and the resulting GAN system is termed $\mathcal{L}_\alpha$-GAN. Under an optimal discriminator, it is shown that the generator's optimization problem consists of minimizing a Jensen-f_α-divergence, a natural generalization of the Jensen-Shannon divergence, where f_α is a convex function expressed in terms of the loss function $\mathcal{L}_\alpha$. It is also demonstrated that this $\mathcal{L}_\alpha$-GAN problem recovers as special cases a number of GAN problems in the literature, including VanillaGAN, least squares GAN (LSGAN), least kth-order GAN (LkGAN), and the recently introduced (α_D, α_G)-GAN with $\alpha_D = 1$. Finally, experimental results are provided for three datasets—MNIST, CIFAR-10, and Stacked MNIST—to illustrate the performance of various examples of the $\mathcal{L}_\alpha$-GAN system.

Keywords: generative adversarial networks; deep learning; parameterized loss functions; f-divergence; Jensen-f-divergence

1. Introduction

Generative adversarial networks (GANs), first introduced by Goodfellow et al. in 2014 [1], have a variety of applications in media generation [2], image restoration [3], and data privacy [4]. GANs aim to generate synthetic data that closely resemble the original real data with (unknown) underlying distribution $P_\mathbf{x}$. The GAN is trained such that the distribution of the generated data, $P_\mathbf{g}$, approximates $P_\mathbf{x}$ well. More specifically, low-dimensional random noise is fed to a generator neural network G to produce synthetic data. Real data and the generated data are then given to a discriminator neural network D that scores the data between 0 and 1, with a score close to 1 meaning that the discriminator thinks the data belong to the real dataset. The discriminator and generator play a minimax game, where the aim is to minimize the generator's loss and maximize the discriminator's loss.

Since its initial introduction, several variants of GAN have been proposed. Deep convolutional GAN (DCGAN) [5] utilizes the same loss functions as VanillaGAN (the original GAN) while combining GANs with convolutional neural networks, which are helpful when applying GANs to image data as they extract visual features from the data. DCGANs are more stable than the baseline model but can suffer from mode collapse, which occurs when the generator learns that a select number of images can easily fool the discriminator, resulting in the generator only generating those images. Another notable issue with VanillaGAN is the tendency for the generator network's gradients to vanish. In the early stages of training, the discriminator lacks confidence and assigns generated data values close to zero. Therefore, the objective function tends to zero, resulting in small

gradients and a lack of learning. To mitigate this issue, a non-saturating generator loss function was proposed in [1] so that gradients do not vanish early on in training.

In the original (VanillaGAN) problem setup, the objective function, expressed as a negative sum of two Shannon cross-entropies, is to be minimized by the generator and maximized by the discriminator. It is demonstrated that if the discriminator is fixed to be optimal (i.e., as a maximizer of the objective function), the GAN's minimax game can be reduced to minimizing the Jensen-Shannon divergence (JSD) between the real and generated data's probability distributions [1]. An analogous result was proven in [6] for RényiGANs, a dual-objective GAN using distinct discriminator and generator loss functions. More specifically, under a canonical discriminator loss function (as in [1]) and a generator loss function expressed in terms of two Rényi cross-entropies, it is shown that the RényiGAN optimization problem reduces to minimizing the Jensen-Rényi divergence, hence extending VanillaGAN's results.

Nowozin et al. generalized VanillaGAN by formulating a class of loss functions in [7] parametrized by a lower semicontinuous convex function f, devising f-GAN. More specifically, the f-GAN problem consists of minimizing an f-divergence between the true data distribution and the generator distribution via a minimax optimization of a Fenchel conjugate representation of the f-divergence, where the VanillaGAN discriminator's role (as a binary classifier) is replaced by a variational function estimating the ratio of the true data and generator distributions. The f-GAN loss function may be tedious to derive, as it requires computation of the Fenchel conjugate of f. It can be shown that f-GAN can interpolate between VanillaGAN and HellingerGAN, among others [7].

More recently, α-GAN was presented in [8], for which the aim is to derive a class of loss functions parameterized by $\alpha > 0$ and expressed in terms of a class probability estimation (CPE) loss between a real label $y \in \{0,1\}$ and predicted label $\hat{y} \in [0,1]$ [8]. The ability to control α as a hyperparameter is beneficial to be able to apply one system to multiple datasets, as two datasets may be optimal under different α values. This work was further analyzed in [9] and expanded in [10] by introducing the dual-objective (α_D, α_G)-GAN, which allowed for the generator and discriminator loss functions to have distinct α parameters with the aim of improving training stability. When $\alpha_D = \alpha_G$, the α-GAN optimization reduces to minimizing an Arimoto divergence, as originally derived in [8]. Note that α-GAN can recover several f-GANs, such as HellingerGAN, VanillaGAN, WassersteinGAN, and total variation GAN [8]. Furthermore, in their more recent work [11] that unifies [8–10], the authors establish, under some conditions, a one-to-one correspondence between CPE-loss-based GANs (such as α-GANs) and f-GANs that use a symmetric f-divergence (see Theorems 4–5 and Corollary 1 in [11]). They also prove various generalization and estimation error bounds for (α_D, α_G)-GANs and illustrate their ability to mitigate training instability for synthetic Gaussian data as well as the Celeb-A and LSUN Classroom image datasets. The various (α_D, α_G)-GAN equilibrium results do not provide an analogous result to JSD and Jensen-Rényi divergence minimization for the VanillaGAN [1] and RényiGAN [6] problems, respectively, as they do not involve a *Jensen-type divergence*. More specifically given a divergence measure $\mathcal{D}(p\|q)$ between distributions p and q (i.e., a positive-definite bivariate function: $\mathcal{D}(p\|q) \geq 0$ with equality if and only if (iff) $p = q$ almost everywhere (a.e.)), a *Jensen-type divergence of $\mathcal{D}$* is given by

$$\frac{1}{2}\mathcal{D}\left(p\|\frac{p+q}{2}\right) + \frac{1}{2}\mathcal{D}\left(q\|\frac{p+q}{2}\right);$$

i.e., it is the arithmetic average of two $\mathcal{D}$-divergences: one between p and the mixture $(p+q)/2$ and the other between q and $(p+q)/2$.

The main objective of our work is to present a unifying approach that provides an axiomatic framework to encompass several existing GAN generator loss functions so that GAN optimization can be simplified in terms of a Jensen-type divergence. In particular, our framework classifies the set of α-parameterized CPE-based loss functions $\mathcal{L}_\alpha$, generalizing the α-loss function in [8–11]. We then propose $\mathcal{L}_\alpha$-GAN: a dual-objective GAN that uses

a function from this class for the generator and uses any canonical discriminator loss function that admits the same optimizer as VanillaGAN [1]. We show that under some regularity (convexity/concavity) conditions on $\mathcal{L}_\alpha$, the minimax game played with these two loss functions is equivalent to the minimization of a Jensen-f_α-divergence: a Jensen-type divergence and another natural extension of the Jensen-Shannon divergence (in addition to the Jensen-Rényi divergence [6]), where the generating function f_α of the divergence is directly computed from the CPE loss function $\mathcal{L}_\alpha$. This result recovers various prior dual-objective GAN equilibrium results, thus unifying them under one parameterized generator loss function. The newly obtained Jensen-f_α-divergence, which is noted to belong to the class of symmetric f-divergences with different generating functions (see Remark 1), is a useful measure of dissimilarity between distributions as it requires a convex function f with a restricted domain given by the interval $[0, 2]$ (see Remark 2) in addition to its symmetry and finiteness properties.

The rest of the paper is organized as follows. In Section 2, we review f-divergence measures and introduce the Jensen-f-divergence as an extension of the Jensen-Shannon divergence. In Section 3, we establish our main result regarding the optimization of our unifying generator loss function (Theorem 1) and show that it can be applied to a large class of known GANs (Lemmas 2–4). We conduct experiments in Section 4 by implementing different manifestations of $\mathcal{L}_\alpha$-GAN on three datasets: MNIST, CIFAR-10, and Stacked MNIST. Finally, we conclude the paper in Section 5.

2. Preliminaries

We begin by presenting key information measures used throughout the paper. Let $f : [0, \infty) \to (-\infty, \infty]$ be a convex continuous function that is strictly convex at 1 (i.e., $f(\lambda u_1 + (1 - \lambda)u_2) < \lambda f(u_1) + (1 - \lambda)f(u_2)$ for all $u_1, u_2 \geq 0$, $u_1 \neq u_2$, and $\lambda \in (0, 1)$ such that $\lambda u_1 + (1 - \lambda)u_2 = 1$) and satisfying

$$f(1) = 0.$$

Note that the convexity of f already implies its continuity on $(0, \infty)$. Here, the continuity of f at 0 is extended, setting $f(0) = \lim_{u \downarrow 0} f(u)$, which may be infinite. Otherwise, $f(u)$ is assumed to be finite for $u > 0$.

Definition 1 ([12–14]). *The **f-divergence** between two probability densities p and q with common support $\mathcal{R} \subseteq \mathbb{R}^d$ on the Lebesgue measurable space $(\mathcal{R}, \mathcal{B}(\mathcal{R}), \mu)$ is denoted by $D_f(p\|q)$ and given by*

$$D_f(p\|q) = \int_{\mathcal{R}} q\, f\left(\frac{p}{q}\right) d\mu, \tag{1}$$

where we have used the shorthand $\int_{\mathcal{R}} g\, d\mu := \int_{\mathcal{R}} g(x)\, d\mu(x)$, where g is a measurable function; we follow this convention from now on. Here, f is referred to as the generating function of $D_f(p\|q)$.

For simplicity, we consider throughout densities with common supports. A comprehensive definition of f-divergence for arbitrary distributions can be found in Section III of [15]. We require that f is strictly convex around 1 and that it satisfies the normalization condition $f(1) = 0$ to ensure positive-definiteness of the f-divergence, i.e., $D_f(p\|q) \geq 0$ with equality holding iff $p = q$ (a.e.). We present examples of f-divergences under various choices of their generating function f in Table 1. We will be invoking these divergence measures in different parts of the paper.

Table 1. Examples of f-divergences.

f-**Divergence**	**Symbol**	**Formula**	$f(u)$
Kullback–Leiber [16]	KL	$\int_{\mathcal{R}} p \log\left(\frac{p}{q}\right) d\mu$	$u \log u$
Jensen-Shannon [17]	JSD	$\frac{1}{2}\text{KL}\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}\text{KL}\left(q\middle\|\frac{p+q}{2}\right)$	$\frac{1}{2}\left(u \log u - (u+1)\log\frac{u+1}{2}\right)$
Pearson χ^2 [18]	χ^2	$\int_{\mathcal{R}} \frac{(q-p)^2}{p} d\mu$	$\left(\sqrt{x} - \frac{1}{\sqrt{x}}\right)^2$
Pearson–Vajda ($k > 1$) [18]	$|\chi|^k$	$\int_{\mathcal{R}} \frac{|q-p|^k}{p^{k-1}} d\mu$	$u^{1-k}|1-u|^k$
Arimoto ($\alpha > 0, \alpha \neq 1$) [15,19,20]	$\mathcal{A}_\alpha$	$\frac{\alpha}{\alpha-1}\left(\int_{\mathcal{R}}(p^\alpha + q^\alpha)^{\frac{1}{\alpha}} d\mu - 2^{\frac{1}{\alpha}}\right)$	$\frac{\alpha}{\alpha-1}\left((1+u)^{\frac{1}{\alpha}} - (1+u) - 2^{\frac{1}{\alpha}} + 2\right)$
Hellinger ($\alpha > 0, \alpha \neq 1$) [15,21,22]	$\mathcal{H}_\alpha$	$\frac{1}{\alpha-1}\left(\int_{\mathcal{R}} p^\alpha q^{1-\alpha} d\mu - 1\right)$	$\frac{u^\alpha - 1}{\alpha - 1}$

The Rényi divergence of order α ($\alpha > 0$, $\alpha \neq 1$) between densities p and q with common support $\mathcal{R}$ is used in [6] in the RényiGAN problem; it is given by [23,24]

$$D_\alpha(p\|q) = \frac{1}{\alpha - 1}\log\left(\int_{\mathcal{R}} p^\alpha q^{1-\alpha} d\mu\right). \tag{2}$$

Note that the Rényi divergence is not an f-divergence; however, it can be expressed as a transformation of the Hellinger divergence (which is itself an f-divergence):

$$D_\alpha(p\|q) = \frac{1}{\alpha - 1}\log(1 + (\alpha - 1)\mathcal{H}_\alpha(p\|q)). \tag{3}$$

We now introduce a new measure, the Jensen-f-divergence, which is analogous to the Jensen-Shannon and Jensen-Rényi divergences.

Definition 2. *The* ***Jensen-f-divergence*** *between two probability distributions p and q with common support $\mathcal{R} \subseteq \mathbb{R}^d$ on the Lebesgue measurable space $(\mathcal{R}, \mathcal{B}(\mathcal{R}), \mu)$ is denoted by $\text{JD}_f(p\|q)$ and given by*

$$\text{JD}_f(p\|q) = \frac{1}{2}D_f\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}D_f\left(q\middle\|\frac{p+q}{2}\right), \tag{4}$$

where $D_f(\cdot\|\cdot)$ is the f-divergence.

We next verify that the Jensen-Shannon divergence is a Jensen-f-divergence.

Lemma 1. *Let p and q be two densities with common support $\mathcal{R} \subseteq \mathbb{R}^d$, and consider the function $f : [0, \infty) \to (-\infty, \infty]$ given by $f(u) = u \log u$. Then we have that*

$$\text{JD}_f(p\|q) = \text{JSD}(p\|q). \tag{5}$$

Proof. As f is convex (and continuous) on its domain with $f(1) = 0$, we have that

$$\begin{aligned}
\text{JSD}(p\|q) &= \frac{1}{2}\text{KL}\left(p\middle\|\frac{p+q}{2}\right) + \frac{1}{2}\text{KL}\left(q\middle\|\frac{p+q}{2}\right) \\
&= \frac{1}{2}\int_{\mathcal{R}} p\log\left(\frac{2p}{p+q}\right) d\mu + \frac{1}{2}\int_{\mathcal{R}} q\log\left(\frac{2q}{p+q}\right) d\mu \\
&= \frac{1}{2}\int_{\mathcal{R}} \frac{p+q}{2}\left(\frac{2p}{p+q}\log\left(\frac{2p}{p+q}\right)\right) d\mu \\
&\quad + \frac{1}{2}\int_{\mathcal{R}} \frac{p+q}{2}\left(\frac{2q}{p+q}\log\left(\frac{2q}{p+q}\right)\right) d\mu \\
&= \text{JD}_f(p\|q).
\end{aligned}$$

$\square$

Remark 1 (*Jensen-f-divergence is a symmetric f-divergence*). *Note that* $\mathrm{JD}_f(p\|q)$ *is itself a symmetric f-divergence (with a modified generating function). Indeed, given the continuous convex function f that is strictly convex around 1 with $f(1) = 0$, consider the functions*

$$f_1(u) := \frac{u+1}{2} f\left(\frac{2u}{u+1}\right), \qquad u \geq 0,$$

and

$$f_2(u) := \frac{u+1}{2} f\left(\frac{2}{u+1}\right), \qquad u \geq 0,$$

which are both continuous convex, strictly convex around 1, and satisfy $f_1(1) = f_2(1) = 0$. Now, direct calculations yield that

$$D_f\left(p \,\middle\|\, \frac{p+q}{2}\right) = D_{f_1}(p\|q)$$

and

$$D_f\left(q \,\middle\|\, \frac{p+q}{2}\right) = D_{f_2}(p\|q).$$

Thus,

$$\mathrm{JD}_f(p\|q) = \frac{1}{2}D_{f_1}(p\|q) + \frac{1}{2}D_{f_2}(p\|q) = D_{\bar{f}}(p\|q),$$

where $\bar{f} := \frac{1}{2}(f_1 + f_2)$, i.e.,

$$\bar{f}(u) = \frac{u+1}{4}\left(f\left(\frac{2u}{u+1}\right) + f\left(\frac{2}{u+1}\right)\right), \qquad u \geq 0, \tag{6}$$

is also continuous convex, strictly convex around 1, and satisfies $\bar{f}(1) = 0$. Since by (4)*,*

$$\mathrm{JD}_f(p\|q) = \mathrm{JD}_f(q\|p),$$

we conclude that the Jensen-f-divergence is a symmetric $\bar{f}$-divergence. An equivalent argument is to note that $\bar{f} = \bar{f}^\star$, where $\bar{f}^\star(u) := u\bar{f}(\frac{1}{u})$, $u \geq 0$ (with $\bar{f}^\star(0) = \lim_{t\to\infty} \bar{f}(t)/t$), which is a necessary and sufficient condition for the $\bar{f}$-divergence to be symmetric (see p. 4399 in [15]*).*

Remark 2 (*Domain of f*). *Examining* (4)*, we note that the Jensen-f-divergence between p and q involves the f-divergences between either p or q and their mixture $(p+q)/2$. In other words, to determine $\mathrm{JD}_f(p\|q)$, we only need $f\left(\frac{2p}{p+q}\right)$ and $f\left(\frac{2q}{p+q}\right)$ when taking the expectations in* (1)*. Thus, it is sufficient to restrict the domain of the convex function f to the interval $[0,2]$.*

3. Main Results

We now present our main theorem that unifies various generator loss functions under a CPE-based loss function $\mathcal{L}_\alpha$ for a dual-objective GAN, $\mathcal{L}_\alpha$-GAN, with a canonical discriminator loss function that is optimized as in [1]. Under some regularity conditions on the loss function $\mathcal{L}_\alpha$, we show that under the optimal discriminator, our generator loss becomes a Jensen-f-divergence.

Let $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \mu)$ be the measured space of $n \times n \times m$ images (where $m = 1$ for black and white images and $m = 3$ for RGB images), and let $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}), \mu)$ be a measured space such that $\mathcal{Z} \subseteq \mathbb{R}^d$. The discriminator neural network is given by $D : \mathcal{X} \to [0,1]$, and the generator neural network is given by $G : \mathcal{Z} \to \mathcal{X}$. The generator's noise input is sampled from a multivariate Gaussian distribution $P_{\mathbf{z}} : \mathcal{Z} \to [0,1]$. We denote the probability distribution of the real data by $P_{\mathbf{x}} : \mathcal{X} \to [0,1]$ and the probability distribution of the generated data by $P_{\mathbf{g}} : \mathcal{X} \to [0,1]$. We also set $P_{\mathbf{x}}$ and $P_{\mathbf{g}}$ as the densities corresponding to $P_{\mathbf{x}}$ and $P_{\mathbf{g}}$, respectively. We begin by introducing the $\mathcal{L}_\alpha$-GAN system.

Definition 3. *Fix $\alpha \in \mathcal{A} \subseteq \mathbb{R}$ and let $\mathcal{L}_\alpha : \{0,1\} \times [0,1] \to [0,\infty)$ be a loss function such that $\hat{y}\mathcal{L}_\alpha\left(1, \frac{\hat{y}}{2}\right)$ is a continuous function that is either convex or concave in $\hat{y} \in [0,2]$ with strict convexity (respectively, strict concavity) around $\hat{y} = 1$ and such that $\mathcal{L}_\alpha$ is symmetric in the sense that*

$$\mathcal{L}_\alpha(1, \hat{y}) = \mathcal{L}_\alpha(0, 1 - \hat{y}), \qquad \hat{y} \in [0,1]. \tag{7}$$

Then the $\mathcal{L}_\alpha$-GAN system is defined by $(V_D, V_{\mathcal{L}_\alpha, G})$, where $V_D : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is the discriminator loss function, and $V_{\mathcal{L}_\alpha} : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ is the generator loss function, which is given by

$$V_{\mathcal{L}_\alpha, G}(D, G) = \mathbb{E}_{A \sim P_\mathbf{x}}[-\mathcal{L}_\alpha(1, D(A))] + \mathbb{E}_{B \sim P_\mathbf{g}}[-\mathcal{L}_\alpha(0, D(B))]. \tag{8}$$

Moreover, the $\mathcal{L}_\alpha$-GAN problem is defined by

$$\sup_D V_D(D, G) \tag{9}$$

$$\inf_G V_{\mathcal{L}_\alpha, G}(D, G). \tag{10}$$

We now present our main result about the $\mathcal{L}_\alpha$-GAN optimization problem.

Theorem 1. *For a fixed $\alpha \in \mathcal{A} \subseteq \mathbb{R}$ and $\mathcal{L}_\alpha : \{0,1\} \times [0,1] \to [0,\infty)$, let $(V_D, V_{\mathcal{L}_\alpha, G})$ be the loss functions of $\mathcal{L}_\alpha$-GAN and consider joint optimization in (9)–(10). If V_D is a canonical loss function in the sense that it is maximized at $D = D^*$, where*

$$D^* = \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}, \tag{11}$$

then (10) reduces to

$$\inf_G V_{\mathcal{L}_\alpha, G}(D^*, G) = \inf_G 2a\mathrm{JD}_{f_\alpha}(P_\mathbf{x} \| P_\mathbf{g}) - 2ab, \tag{12}$$

where $\mathrm{JD}_{f_\alpha}(\cdot \| \cdot)$ is the Jensen-f_α-divergence, and $f_\alpha : [0,2] \to \mathbb{R}$ is a continuous convex function that is strictly convex around 1 and is given by

$$f_\alpha(u) = -u\left(\frac{1}{a}\mathcal{L}_\alpha\left(1, \frac{u}{2}\right) - b\right), \tag{13}$$

where a and b are real constants chosen so that $f_\alpha(1) = 0$ with $a < 0$ (respectively, $a > 0$) if $u\mathcal{L}_\alpha\left(1, \frac{u}{2}\right)$ is convex (respectively, concave). Finally, (12) is minimized when $P_\mathbf{x} = P_\mathbf{g}$ (a.e.).

Proof. Under the assumption that V_D is maximized at $D^* = \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}$, we have that

$$
\begin{aligned}
V_{\mathcal{L}_\alpha, G}(D^*, G) &= \mathbb{E}_{\mathbf{A} \sim P_\mathbf{x}}[-\mathcal{L}_\alpha(1, D^*(\mathbf{A}))] + \mathbb{E}_{\mathbf{B} \sim P_\mathbf{g}}[-\mathcal{L}_\alpha(0, D^*(\mathbf{B}))] \\
&= -\int_\mathcal{X} P_\mathbf{x} \mathcal{L}_\alpha(1, D^*)\, d\mu - \int_\mathcal{X} P_\mathbf{g} \mathcal{L}_\alpha(0, D^*)\, d\mu \\
&= -\int_\mathcal{X} P_\mathbf{x} \mathcal{L}_\alpha\left(1, \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu - \int_\mathcal{X} P_\mathbf{g} \mathcal{L}_\alpha\left(0, \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu \\
&= -2\int_\mathcal{X} \left(\frac{P_\mathbf{x} + P_\mathbf{g}}{2}\right) \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}} \mathcal{L}_\alpha\left(1, \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu \\
&\quad - 2\int_\mathcal{X} \left(\frac{P_\mathbf{x} + P_\mathbf{g}}{2}\right) \frac{P_\mathbf{g}}{P_\mathbf{x} + P_\mathbf{g}} \mathcal{L}_\alpha\left(0, \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu \\
&\overset{(a)}{=} -2\int_\mathcal{X} \left(\frac{P_\mathbf{x} + P_\mathbf{g}}{2}\right) \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}} \mathcal{L}_\alpha\left(1, \frac{P_\mathbf{x}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu \\
&\quad - 2\int_\mathcal{X} \left(\frac{P_\mathbf{x} + P_\mathbf{g}}{2}\right) \frac{P_\mathbf{g}}{P_\mathbf{x} + P_\mathbf{g}} \mathcal{L}_\alpha\left(1, \frac{P_\mathbf{g}}{P_\mathbf{x} + P_\mathbf{g}}\right) d\mu
\end{aligned}
$$

$$\overset{(b)}{=} -2 \int_{\mathcal{X}} \left(\frac{P_{\mathbf{x}} + P_{\mathbf{g}}}{2} \right) \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \left(\frac{-a f_\alpha \left(\frac{2 P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \right)}{\frac{2 P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}} + ab \right) d\mu$$

$$- 2 \int_{\mathcal{X}} \left(\frac{P_{\mathbf{x}} + P_{\mathbf{g}}}{2} \right) \frac{P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \left(\frac{-a f_\alpha \left(\frac{2 P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \right)}{\frac{2 P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}} + ab \right) d\mu$$

$$= 2a \left(\frac{1}{2} \int_{\mathcal{X}} \frac{P_{\mathbf{x}} + P_{\mathbf{g}}}{2} f_\alpha \left(\frac{2 P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \right) d\mu \right.$$

$$+ \frac{1}{2} \int_{\mathcal{X}} \frac{P_{\mathbf{x}} + P_{\mathbf{g}}}{2} f_\alpha \left(\frac{2 P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}} \right) d\mu \right) - 2ab$$

$$= 2a \, \mathrm{JD}_{f_\alpha}(P_{\mathbf{x}} \| P_{\mathbf{g}}) - 2ab,$$

where:

- (a) holds since $\mathcal{L}_\alpha(1, u) = \mathcal{L}_\alpha(0, 1 - u)$ by (7), where $u = \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$.
- (b) holds by solving for $\mathcal{L}_\alpha(1, u)$ in terms of $f_\alpha(2u)$ in (13), where $u = \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$ in the first term and $u = \frac{P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$ in the second term.

The constants a and b are chosen so that $f_\alpha(1) = 0$. Finally, the continuity and convexity of f_α (as well as its strict convexity around 1) directly follow from the corresponding assumptions imposed on the loss function $\mathcal{L}_\alpha$ in Definition 3 and on the condition imposed on the sign of a in the theorem's statement. □

Remark 3. *Note that not only D^* given in (11) is an optimal discriminator of the (original) VanillaGAN discriminator loss function, but it also optimizes the LSGAN/LkGAN discriminators loss functions when their discriminators' labels for fake and real data, γ and β, respectively satisfy $\gamma = 1$ and $\beta = 0$ (see Section 3.3).*

We next show that the $\mathcal{L}_\alpha$-GAN of Theorem 1 recovers as special cases a number of well-known GAN generator loss functions and their equilibrium points (under an optimal classical discriminator D^*).

3.1. VanillaGAN

VanillaGAN [1] uses the same loss function V_{VG} for the both generator and discriminator, which is

$$V_{\mathrm{VG}}(D, G) = \mathbb{E}_{\mathbf{A} \sim P_{\mathbf{x}}}[-\log D(\mathbf{A})] + \mathbb{E}_{\mathbf{B} \sim P_{\mathbf{g}}}[-\log(1 - D(\mathbf{B}))] \tag{14}$$

and can be cast as a saddle point optimization problem:

$$\inf_G \sup_D V_{\mathrm{VG}}(D, G). \tag{15}$$

It is shown in [1] that the optimal discriminator for (15) is given by $D^* = \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$, as in (11). When $D = D^*$, the optimization reduces to minimizing the Jensen-Shannon divergence:

$$\inf_G V_{\mathrm{VG}}(D^*, G) = \inf_G 2\mathrm{JSD}(P_{\mathbf{x}} \| P_{\mathbf{g}}) - 2 \log 2. \tag{16}$$

We next show that (16) can be obtained from Theorem 1.

Lemma 2. *Consider the optimization of VanillaGAN given in (15). Then we have that*

$$V_{\mathrm{VG}}(D^*, G) = 2\mathrm{JSD}(P_{\mathbf{x}} \| P_{\mathbf{g}}) - 2 \log 2 = V_{\mathcal{L}_\alpha, G}(D^*, G),$$

where $\mathcal{L}_\alpha(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$ for all $\alpha \in \mathcal{A} = \mathbb{R}$.

Proof. For any fixed $\alpha \in \mathbb{R}$, let the function $\mathcal{L}_\alpha$ in (8) be as defined in the statement:

$$\mathcal{L}_\alpha(y, \hat{y}) = -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}).$$

Note that $\mathcal{L}_\alpha$ is symmetric, since for $\hat{y} \in [0, 1]$, we have that

$$\mathcal{L}_\alpha(1, \hat{y}) = -\log(\hat{y}) = \mathcal{L}_\alpha(0, 1 - \hat{y}).$$

Instead of showing the continuity and convexity/concavity conditions imposed on $\hat{y} \mathcal{L}_\alpha\left(1, \frac{\hat{y}}{2}\right)$ in Definition 3, we implicitly verify them by directly deriving f_α from $\mathcal{L}_\alpha$ using (13) and showing that it is continuous convex and strictly convex around 1. Setting $a = 1$ and $b = \log 2$, we have that

$$\begin{aligned} f_\alpha(u) &= -u \left(\frac{1}{a} \mathcal{L}_\alpha\left(1, \frac{u}{2}\right) - b \right) \\ &= -u \left(-\log \frac{u}{2} - \log 2 \right) = u \log u. \end{aligned}$$

Clearly, f is convex (actually strictly convex on $(0, \infty)$ and hence strictly convex around 1) and continuous on its domain (where $f(0) = \lim_{u \to 0} u \log(u) = 0$). It also satisfies $f(1) = 0$. By Lemma 1, we know that under the generating function $f(u) = u \log(u)$, the Jensen-f divergence reduces to the Jensen-Shannon divergence. Therefore, by Theorem 1, we have that

$$\begin{aligned} V_{\mathcal{L}_\alpha, G}(D^*, G) &= 2a \mathrm{JD}_{f_\alpha}(P_{\mathbf{x}} \| P_{\mathbf{g}}) - 2ab \\ &= 2\mathrm{JSD}(P_{\mathbf{x}} \| P_{\mathbf{g}}) - 2\log 2 \\ &= V_{\mathrm{VG}}(D^*, G), \end{aligned}$$

which finishes the proof. $\square$

3.2. α-GAN

The notion of α-GANs is introduced in [8] as a way to unify several existing GANs using a parameterized loss function. We describe α-GANs next.

Definition 4 ([8]). *Let $y \in \{0, 1\}$ be a binary label, $\hat{y} \in [0, 1]$, and fix $\alpha > 0$. The **α-loss** between y and $\hat{y}$ is the map $\ell_\alpha : \{0, 1\} \times [0, 1] \to [0, \infty)$ given by*

$$\ell_\alpha(y, \hat{y}) = \begin{cases} \frac{\alpha}{\alpha - 1} \left(1 - y \hat{y}^{\frac{\alpha - 1}{\alpha}} + (1 - y)(1 - \hat{y})^{\frac{\alpha - 1}{\alpha}} \right), & \alpha \in (0, 1) \cup (1, \infty) \\ -y \log \hat{y} - (1 - y) \log(1 - \hat{y}), & \alpha = 1. \end{cases} \tag{17}$$

Definition 5 ([8]). *For $\alpha > 0$, the **α-GAN loss function** is given by*

$$V_\alpha(D, G) = \mathbb{E}_{A \sim P_{\mathbf{x}}}[-\ell_\alpha(1, D(A))] + \mathbb{E}_{B \sim P_{\mathbf{g}}}[-\ell_\alpha(0, D(B))]. \tag{18}$$

Joint optimization of the α-GAN problem is given by

$$\inf_G \sup_D V_\alpha(D, G). \tag{19}$$

It is known that α-GAN recovers several well-known GANs by varying the α parameter: notably, VanillaGAN ($\alpha = 1$) [1] and HellingerGAN ($\alpha = \frac{1}{2}$) [7]. Furthermore, as $\alpha \to \infty$, V_α recovers a translated version of the WassersteinGAN loss function [25]. We now present the solution to the joint optimization problem presented in (19).

Proposition 1 ([8]). *Let $\alpha > 0$ and consider joint optimization of the α-GAN presented in (19). The discriminator D^* that maximizes the loss function is given by*

$$D^* = \frac{P_{\mathbf{x}}{}^{\alpha}}{P_{\mathbf{x}}{}^{\alpha} + P_{\mathbf{g}}{}^{\alpha}}. \tag{20}$$

Furthermore, when $D = D^*$ is fixed, the problem in (19) reduces to minimizing an Arimoto divergence (as defined in Table 1) when $\alpha \neq 1$:

$$\inf_{G} V_{\alpha}(D^*, G) = \inf_{G} \mathcal{A}_{\alpha}(P_{\mathbf{x}}\|P_{\mathbf{g}}) + \frac{\alpha}{\alpha - 1}\left(2^{\frac{1}{\alpha}} - 2\right) \tag{21}$$

and a Jensen-Shannon divergence when $\alpha = 1$:

$$\inf_{G} V_1(D^*, G) = \inf_{G} \mathrm{JSD}(P_{\mathbf{x}}\|P_{\mathbf{g}}) - 2\log 2, \tag{22}$$

where (21) and (22) achieve their minima iff $P_{\mathbf{x}} = P_{\mathbf{g}}$ (a.e.).

Recently, α-GAN was generalized in [10] to implement a dual-objective GAN, which we describe next.

Definition 6 ([10]). *For $\alpha_D > 0$ and $\alpha_G > 0$, the $(\boldsymbol{\alpha_D}, \boldsymbol{\alpha_G})$-**GAN's optimization** is given by*

$$\sup_{D} V_{\alpha_D}(D, G) \tag{23}$$

$$\inf_{G} V_{\alpha_G}(D, G) \tag{24}$$

where V_{α_D} and V_{α_G} are defined in (18), with α replaced by α_D and α_G, respectively.

Proposition 2 ([10]). *Consider the joint optimization in (23) and (24). Let parameters $\alpha_D, \alpha_G > 0$ satisfy*

$$\left(\alpha_D \leq 1, \alpha_G > \frac{\alpha_D}{\alpha_D + 1}\right) \text{ or } \left(\alpha_D > 1, \frac{\alpha_D}{2} < \alpha_G \leq \alpha_D\right). \tag{25}$$

The discriminator D^* that maximizes V_{α_D} is given by

$$D^* = \frac{P_{\mathbf{x}}{}^{\alpha_D}}{P_{\mathbf{x}}{}^{\alpha_D} + P_{\mathbf{g}}{}^{\alpha_D}}. \tag{26}$$

Furthermore, when $D = D^*$ is fixed, the minimization of V_{α_G} in (24) is equivalent to the following f-divergence minimization:

$$\inf_{G} V_{\alpha_G}(D^*, G) = \inf_{G} D_{f_{\alpha_D, \alpha_G}}(P_{\mathbf{x}}\|P_{\mathbf{g}}) + \frac{\alpha}{\alpha - 1}\left(2^{\frac{1}{\alpha}} - 2\right), \tag{27}$$

where $f_{\alpha_D, \alpha_G} : [0, \infty) \to \mathbb{R}$ is given by

$$f_{\alpha_D, \alpha_G}(u) = \frac{\alpha_G}{\alpha_G - 1}\left(\frac{u^{\alpha_D\left(1 - \frac{1}{\alpha_G}\right) + 1} + 1}{(u^{\alpha_D} + 1)^{1 - \frac{1}{\alpha_G}}}\right). \tag{28}$$

We now apply the (α_D, α_G)-GAN to our main result in Theorem 1 by showing that (12) can recover (27) when $\alpha_D = 1$ (which corresponds to a VanillaGAN discriminator loss function).

Lemma 3. *Consider the (α_D, α_G)-GAN given in Definition 6. Let $\alpha_D = 1$ and $\alpha_G = \alpha > \frac{1}{2}$. Then, the solution to (24) presented in Proposition 2 is equivalent to minimizing a Jensen-f_{α}-divergence: specifically, if D^* is the optimal discriminator given by (26), which is equivalent to (11) when $\alpha_D = 1$, then $V_{\alpha, G}(D^*, G)$ in (27) satisfies*

$$V_{\alpha,G}(D^*,G) = 2^{\frac{1}{\alpha}} \mathrm{JD}_{f_\alpha}(P_\mathbf{x}\|P_\mathbf{g}) + \frac{\alpha}{\alpha-1}(2^{\frac{1}{\alpha}}-2) = V_{\mathcal{L}_\alpha,G}(D^*,G), \tag{29}$$

where $\mathcal{L}_\alpha(y,\hat{y}) = \ell_\alpha(y,\hat{y})$, and

$$f_\alpha(u) = \frac{\alpha}{\alpha-1}\left(u^{2-\frac{1}{\alpha}}-u\right), \quad u \geq 0. \tag{30}$$

Proof. We show that Theorem 1 recovers Proposition 2 by setting $\mathcal{L}_\alpha(y,\hat{y}) = \ell_\alpha(y,\hat{y})$. Note that ℓ_α is symmetric since

$$\ell_\alpha(1,\hat{y}) = \frac{\alpha}{\alpha-1}(1-\hat{y}^{1-\frac{1}{\alpha}}) = \ell_\alpha(0,1-\hat{y}).$$

As in the proof of Lemma 2, instead of proving the conditions imposed on $\hat{y}\mathcal{L}_\alpha\left(1,\frac{\hat{y}}{2}\right)$ in Definition 3, we derive f_α directly from $\mathcal{L}_\alpha$ using (13) and show that it is continuous convex and strictly convex around 1. From Lemma 2, we know that when $\alpha = 1$, $f_\alpha(u) = u\log u$ (which is strictly convex and continuous). For $\alpha \in (0,1) \cup (1,\infty)$, setting $a = 2^{\frac{1}{\alpha}-1}$ and $b = \frac{\alpha}{\alpha-1}\left(2^{1-\frac{1}{\alpha}}-1\right)$ in (13), we have that

$$\begin{aligned}
f_\alpha(u) &= -u\left(\frac{1}{a}\mathcal{L}_\alpha\left(1,\frac{u}{2}\right)-b\right) \\
&= -u\left(2^{1-\frac{1}{\alpha}}\frac{\alpha}{\alpha-1}\left(1-\left(\frac{u}{2}\right)^{1-\frac{1}{\alpha}}\right) - \frac{\alpha}{\alpha-1}(2^{1-\frac{1}{\alpha}}-1)\right) \\
&= \frac{\alpha}{\alpha-1}(-u)[2^{1-\frac{1}{\alpha}}-u^{1-\frac{1}{\alpha}}-(2^{1-\frac{1}{\alpha}}-1)] \\
&= \frac{\alpha}{\alpha-1}(u^{2-\frac{1}{\alpha}}-u).
\end{aligned}$$

Clearly, $f_\alpha(1) = 0$. Furthermore for $\alpha \neq 1$, we have that

$$f''_\alpha(u) = \frac{(2\alpha-1)u^{\frac{-1}{\alpha}}}{\alpha}, \quad u \geq 0,$$

which is positive for $\alpha > \frac{1}{2}$, and f_α is convex for $\alpha > \frac{1}{2}$ (as well as continuous on its domain and strictly convex around 1). Thus, by Theorem 1, we have that

$$\begin{aligned}
V_{\mathcal{L}_\alpha,G}(D^*,G) &= 2a\mathrm{JD}_{f_\alpha}(P_\mathbf{x}\|P_\mathbf{g}) - 2ab \\
&= 2\cdot 2^{\frac{1}{\alpha}-1}\mathrm{JD}_{f_\alpha}(P_\mathbf{x}\|P_\mathbf{g}) - 2\frac{\alpha}{\alpha-1}2^{\frac{1}{\alpha}-1}(2^{1-\frac{1}{\alpha}}-1) \\
&= 2^{\frac{1}{\alpha}}\mathrm{JD}_{f_\alpha}(P_\mathbf{x}\|P_\mathbf{g}) + \frac{\alpha}{\alpha-1}(2^{\frac{1}{\alpha}}-2).
\end{aligned}$$

We now show that the above Jensen-f_α-divergence is equal to the $f_{1,\alpha}$-divergence originally derived for the $(1,\alpha)$-GAN problem of Proposition 2 (note from Proposition 2 that if $\alpha_D = 1$, then $\alpha_G = \alpha > \frac{1}{2}$, so the range of α concurs with the range required above for the convexity of f_α). For any two distributions p and q with common support $\mathcal{X}$, we have that

$$\begin{aligned}
D_{f_{1,\alpha}}(p\|q) &= \frac{\alpha}{\alpha-1}\int_\mathcal{X} q\frac{\left(\frac{p}{q}\right)^{2-\frac{1}{\alpha}}+1}{\left(\frac{p}{q}+1\right)^{1-\frac{1}{\alpha}}}d\mu - \frac{\alpha}{\alpha-1}2^{\frac{1}{\alpha}} \\
&= \frac{\alpha}{\alpha-1}\int_\mathcal{X} q\frac{\left(\frac{p}{q}\right)^{2-\frac{1}{\alpha}}+1}{\left(\frac{p+q}{q}\right)^{1-\frac{1}{\alpha}}}d\mu - \frac{\alpha}{\alpha-1}2^{\frac{1}{\alpha}}
\end{aligned}$$

$$= \frac{\alpha}{\alpha-1} \int_{\mathcal{X}} \left((p+q)\left(\frac{p}{p+q}\right)^{2-\frac{1}{\alpha}} + (p+q)\left(\frac{q}{p+q}\right)^{2-\frac{1}{\alpha}} \right) d\mu$$

$$- \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}}$$

$$= \frac{\alpha}{\alpha-1} \frac{2}{2^{2-\frac{1}{\alpha}}} \int_{\mathcal{X}} \left(\frac{p+q}{2}\left(\frac{2p}{p+q}\right)^{2-\frac{1}{\alpha}} + \frac{p+q}{2}\left(\frac{2q}{p+q}\right)^{2-\frac{1}{\alpha}} \right) d\mu$$

$$- \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}}$$

$$= \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}-1} \int_{\mathcal{X}} \left(\frac{p+q}{2}\left(\left(\frac{2p}{p+q}\right)^{2-\frac{1}{\alpha}} - \frac{2p}{p+q}\right) + p \right) d\mu$$

$$+ \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}-1} \int_{\mathcal{X}} \left(\frac{p+q}{2}\left(\left(\frac{2q}{p+q}\right)^{2-\frac{1}{\alpha}} - \frac{2q}{p+q}\right) + q \right) d\mu$$

$$- \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}}$$

$$= \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}} \frac{1}{2} \left(\int_{\mathcal{X}} \frac{p+q}{2}\left(\left(\frac{2p}{p+q}\right)^{2-\frac{1}{\alpha}} - \frac{2p}{p+q}\right) d\mu + 1 \right)$$

$$+ \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}} \frac{1}{2} \left(\int_{\mathcal{X}} \frac{p+q}{2}\left(\left(\frac{2q}{p+q}\right)^{2-\frac{1}{\alpha}} - \frac{2q}{p+q}\right) d\mu + 1 \right)$$

$$- \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}}$$

$$= 2^{\frac{1}{\alpha}} \mathrm{JD}_{f_\alpha}(p\|q) + \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}-1}(2) - \frac{\alpha}{\alpha-1} 2^{\frac{1}{\alpha}}$$

$$= 2^{\frac{1}{\alpha}} \mathrm{JD}_{f_\alpha}(p\|q).$$

Therefore, $V_{\mathcal{L}_\alpha,G}(D^*,G) = V_\alpha(D^*,G)$. $\square$

Note that this lemma generalizes Lemma 2; VanillaGAN is a special case of $(1,\alpha)$-GAN for $\alpha = 1$.

3.3. Shifted LkGANs and LSGANs

Least squares GAN (LSGAN) was proposed in [26] to mitigate the vanishing gradient problem with VanillaGAN and to stabilize training performance. LSGAN's loss function is derived from the squared error distortion measure, whereby we aim to minimize the distortion between the data samples and a target value we want the discriminator to assign the samples to. LSGAN was generalized with LkGAN in [6] by replacing the squared error distortion measure with an absolute error distortion measure of order $k \geq 1$, therefore introducing an additional degree of freedom to the generator's loss function. We first state the general LkGAN problem. We then apply the result of Theorem 1 to the loss functions of LSGAN and LkGAN.

Definition 7 ([6]). *Let γ, β, $c \in [0,1]$, and let $k \geq 1$. **LkGAN's loss functions**, denoted by $V_{LSGAN,D}$ and $V_{k,G}$, are given by*

$$V_{LSGAN,D}(D,G) = -\frac{1}{2}\mathbb{E}_{A \sim P_\mathbf{x}}[(D(A)-\beta)^2] - \frac{1}{2}\mathbb{E}_{B \sim P_\mathbf{g}}[(D(B)-\gamma)^2] \tag{31}$$

$$V_{k,G}(D,G) = \mathbb{E}_{A \sim P_\mathbf{x}}[|D(A)-c|^k] + \mathbb{E}_{B \sim P_\mathbf{g}}[|D(B)-c|^k]. \tag{32}$$

*The **LkGAN problem** is the joint optimization*

$$\sup_D V_{LSGAN,D}(D,G) \tag{33}$$

$$\inf_G V_{k,G}(D,G). \tag{34}$$

We next recall the solution to (33), which is a minimization of the Pearson–Vajda divergence $|\chi|^k(\cdot\|\cdot)$ of order k (as defined in Table 1).

Proposition 3 ([6])**.** *Consider the joint optimization for LkGAN presented in* (33). *Then the optimal discriminator D^* that maximizes $V_{LSGAN,D}$ in* (31) *is given by*

$$D^* = \frac{\gamma P_{\mathbf{x}} + \beta P_{\mathbf{g}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}. \tag{35}$$

Furthermore, if $D = D^*$ and $\gamma - \beta = 2(c - \beta)$, the minimization of $V_{k,G}$ in (32) *reduces to*

$$\inf_G V_{k,G}(D,G) = \inf_G |c - \beta|^k |\chi|^k (P_{\mathbf{x}} + P_{\mathbf{g}} \| 2P_{\mathbf{g}}). \tag{36}$$

Note that LSGAN [26] is a special case of LkGAN, as we recover LSGAN when $k = 2$ [6].

By scrutinizing Proposition 3 and Theorem 1, we observe that the former cannot be recovered from the latter. However, we can use Theorem 1 by slightly modifying the LkGAN generator's loss function. First, for the dual-objective GAN proposed in Theorem 1, we need $D^* = \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$. By (35), this is achieved for $\gamma = 1$ and $\beta = 0$. Then, we define the intermediate loss function

$$\tilde{V}_{k,G}(D,G) = \mathbb{E}_{\mathbf{A}\sim P_{\mathbf{x}}}[|D(\mathbf{A}) - c_1|^k] + \mathbb{E}_{\mathbf{B}\sim P_{\mathbf{g}}}[|D(\mathbf{B}) - c_2|^k]. \tag{37}$$

Comparing the above loss function with (8), we note that setting $c_1 = 0$ and $c_2 = 1$ in (37) satisfies the symmetry property of $\mathcal{L}_\alpha$. Finally, to ensure the generating function f_α satisfies $f_\alpha(1) = 0$, we shift each term in (37) by 1. Putting these changes together, we propose a revised generator loss function denoted by $\hat{V}_{k,G}$ and given by

$$\hat{V}_{k,G}(D,G) = \mathbb{E}_{\mathbf{A}\sim P_{\mathbf{x}}}[|D(\mathbf{A})|^k - 1] + \mathbb{E}_{\mathbf{B}\sim P_{\mathbf{g}}}[|1 - D(\mathbf{B})|^k - 1]. \tag{38}$$

We call a system that uses (38) as a generator loss function a **Shifted LkGAN (SLkGAN)**. If $k = 2$, we have a shifted version of the LSGAN generator loss function, which we call **Shifted LSGAN (SLSGAN)**. Note that none of these modifications alter the gradients of $V_{k,G}$ in (32), since the first term is independent of G, the choice of c_1 is irrelevant, and translating a function by a constant does not change its gradients. However, from Proposition 3, for $\gamma = 0$, $\beta = 1$, and $c = 1$, we do not have that $\gamma - \beta = 2(c - \beta)$, and as a result, this modified problem does not reduce to minimizing a Pearson–Vajda divergence. Consequently, we can relax the condition on k in Definition 7 to just $k > 0$. We now show how Theorem 1 can be applied to $\mathcal{L}_\alpha$-GAN using (38).

Lemma 4. *Let $k > 0$. Let V_D be a discriminator loss function, and let $\hat{V}_{k,G}$ be the generator's loss function defined in* (38)*. Consider the joint optimization*

$$\sup_D V_D(D,G) \tag{39}$$

$$\inf_G \hat{V}_{k,G}(D,G) \tag{40}$$

If V_D is optimized at $D^* = \frac{P_{\mathbf{x}}}{P_{\mathbf{x}} + P_{\mathbf{g}}}$ (i.e., V_D is canonical), then we have that

$$\hat{V}_{k,G}(D^*,G) = \frac{1}{2^{k-1}} \mathrm{JD}_{f_k}(P_{\mathbf{x}}\|P_{\mathbf{g}}) + \frac{1}{2^{k-1}} - \frac{1}{2},$$

where f_k is given by

$$f_k(u) = u(u^k - 1), \qquad u \geq 0.$$

Examples of $V_D(D, G)$ that satisfy the requirements of Lemma 4 include the LkGAN discriminator loss function given by (31) with $\gamma = 1$ and $\beta = 0$ and the VanillaGAN discriminator loss function given by (14).

Proof. Let $k > 0$. We can restate SLkGAN's generator loss function in (38) in terms of $V_{\mathcal{L}_\alpha, G}$ in (8): we have that $V_{\mathcal{L}_\alpha, G}(D^*, G) = \hat{V}_{k,G}(D^*, G)$, where $\alpha = k$, and $\mathcal{L}_k : \{0, 1\} \times [0, 1] \to [0, \infty)$ is given by

$$\mathcal{L}_k(y, \hat{y}) = -(y(\hat{y}^k - 1) + (1 - y)((1 - \hat{y})^k - 1)). \tag{41}$$

We have that $\mathcal{L}_k$ is symmetric, since

$$\mathcal{L}_k(1, \hat{y}) = -(\hat{y}^k - 1) = \mathcal{L}_k(0, 1 - \hat{y}).$$

We derive f_α from $\mathcal{L}_\alpha$ via (13) and directly check that it is continuous convex and strictly convex around 1. Setting $a = \frac{1}{2^k}$ and $b = 2^k - 1$ in (13), we have that

$$\begin{aligned}
f_k(u) &= -u\left(\frac{1}{a}\mathcal{L}_k\left(1, \frac{u}{2}\right) - b\right) \\
&= -u\left(2^k\left(1 - \left(\frac{u}{2}\right)^k\right) - (2^k - 1)\right) \\
&= -u(2^k - u^k - 2^k + 1) \\
&= u(u^k - 1).
\end{aligned}$$

We clearly have that $f_k(1) = 0$ and that f_k is continuous. Furthermore, we have that $f_k''(u) = k(k+1)u$, which is non-negative for $u \geq 0$. Therefore, f_k is convex (as well as strictly convex around 1). As a result, by Theorem 1, we have that

$$\begin{aligned}
\hat{V}_{k,G}(D^*, G) &= \frac{1}{2^{k-1}}\mathrm{JD}_{f_k}(P_\mathbf{x}\|P_\mathbf{g}) - \frac{1}{2^{k-1}}(2^k - 1) \\
&= \frac{1}{2^{k-1}}\mathrm{JD}_{f_k}(P_\mathbf{x}\|P_\mathbf{g}) + \frac{1}{2^{k-1}} - \frac{1}{2}.
\end{aligned}$$

□

We conclude this section by emphasizing that Theorem 1 serves as a unifying result recovering the existing loss functions in the literature and, moreover, provides a way for generalizing new ones. Our aim in the next section is to demonstrate the versatility of this result in experimentation.

4. Experiments

We perform two experiments on three different image datasets that we describe below.

Experiment 1: In the first experiment, we compare (α, α)-GAN with $(1, \alpha)$-GAN while controlling the value of α. Recall that $\alpha_D = 1$ corresponds to the canonical VanillaGAN (or DCGAN) discriminator. We aim to verify whether or not replacing an α-GAN discriminator with a VanillaGAN discriminator stabilizes or improves the system's performance depending on the value of α. Note that the result of Theorem 1 only applies to the (α_D, α_G)-GAN for $\alpha_D = 1$. We herein confine the comparison of $(1, \alpha)$-GAN with (α, α)-GAN only so that both systems have the same tunable free parameter α. The results obtained in [10] for the Stacked MNIST dataset show that (α_D, α_G)-GAN provides consistently robust performance when $\alpha_D = \alpha_G$. Other experiments illustrating the performance of (α_D, α_G)-GAN with $\alpha_D \neq 1$ are carried for the Celeb-A and LSUN Classroom image datasets in [11] and show improved training stability for $\alpha_D < 1$ values.

Experiment 2: We train two variants of SL*k*GAN with the generator loss function as described in (38) and parameterized by $k > 0$. We then utilize two different canonical discriminator loss functions to align with Theorem 1. The first is the VanillaGAN discriminator loss given by (14); we call the resulting dual-objective GAN **Vanilla-SL*k*GAN**. The second is the L*k*GAN discriminator loss given by (31), where we set $\gamma = 1$ and $\beta = 0$ such that the optimal discriminator is given by (11). We call this system **L*k*-SL*k*GAN**. We compare the two variants to analyze how the value of *k* and choice of discriminator loss impacts the system's performance.

4.1. Experimental Setup

We run both experiments on three image datasets: MNIST [27], CIFAR-10 [28], and Stacked MNIST [29]. The MNIST dataset is a dataset of black and white handwritten digits between 0 and 9 and with a size of $28 \times 28 \times 1$. The CIFAR-10 dataset is an RGB dataset of small images of common animals and modes of transportation with a size of $32 \times 32 \times 3$. The Stacked MNIST dataset is an RGB dataset derived from the MNIST dataset and constructed by taking three MNIST images, assigning each to one of the three color channels, and stacking the images on top of each other. The resulting images are then padded so that each one of them has a size of $32 \times 32 \times 3$.

For Experiment 1, we use α values of 0.5, 5.0, 10.0, and 20.0. For each value of α, we train (α, α)-GAN and $(1, \alpha)$-GAN. We additionally train DCGAN, which corresponds to $(1, 1)$-GAN. For Experiment 2, we use *k* values of 0.25, 1.0, 2.0, 7.5, and 15.0. Note that when $k = 2$, we recover LSGAN. For the MNIST dataset, we run 10 trials with the random seeds 123, 500, 1600, 199,621, 60,677, 20,435, 15,859, 33,764, 79,878, and 36,123 and train each GAN for 250 epochs. For the RGB datasets (CIFAR-10 and Stacked MNIST), we run five trials with the random seeds 123, 1600, 60,677, 15,859, and 79,878 and train each GAN for 500 epochs. All experiments utilize an Adam optimizer for the stochastic gradient descent algorithm with a learning rate of 2×10^{-4} and parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$ [30]. We also experiment with the addition of a gradient penalty (GP); we add a penalty term to the discriminator's loss function to encourage the discriminator's gradient to have a unit norm [31].

The MNIST experiments were run on one 6130 2.1 GHz 1xV100 GPU, 8 CPUs, and 16 GB of memory. The CIFAR-10 and Stacked MNIST experiments were run on one Epyc 7443 2.8 GHz GPU, 8 CPUs, and 16 GB of memory. For each experiment, we report the best overall Fréchet inception distance (FID) score [32], the best average FID score amongst all trials and its variance, and the average epoch the best FID score occurs and its variance. The FID score for each epoch was computed over 10,000 images. For each metric, the lowest numerical value corresponds to the model with the best metric (indicated in bold in the tables). We also report how many trials we include in our summary statistics, as it is possible for a trial to collapse and not train for the full number of epochs. The neural network architectures used in our experiments are presented in Appendix A. The training algorithms are presented in Appendix B.

4.2. Experimental Results

We report the FID metrics for Experiment 1 in Tables 2–4 and for Experiment 2 in Tables 5–7. We report only on those experiments that produced meaningful results. Models that utilize a simplified gradient penalty have the suffix "-GP". For (α_D, α_G)-GANs, we display the output of the best-performing systems in Figure 1 and plot the trajectories of the FID scores throughout the training epochs in Figure 2. Similarly for SL*K*GANs, outputs of the best-performing systems and FID scores vs. epochs trajectories are provided in Figures 3 and 4, respectively.
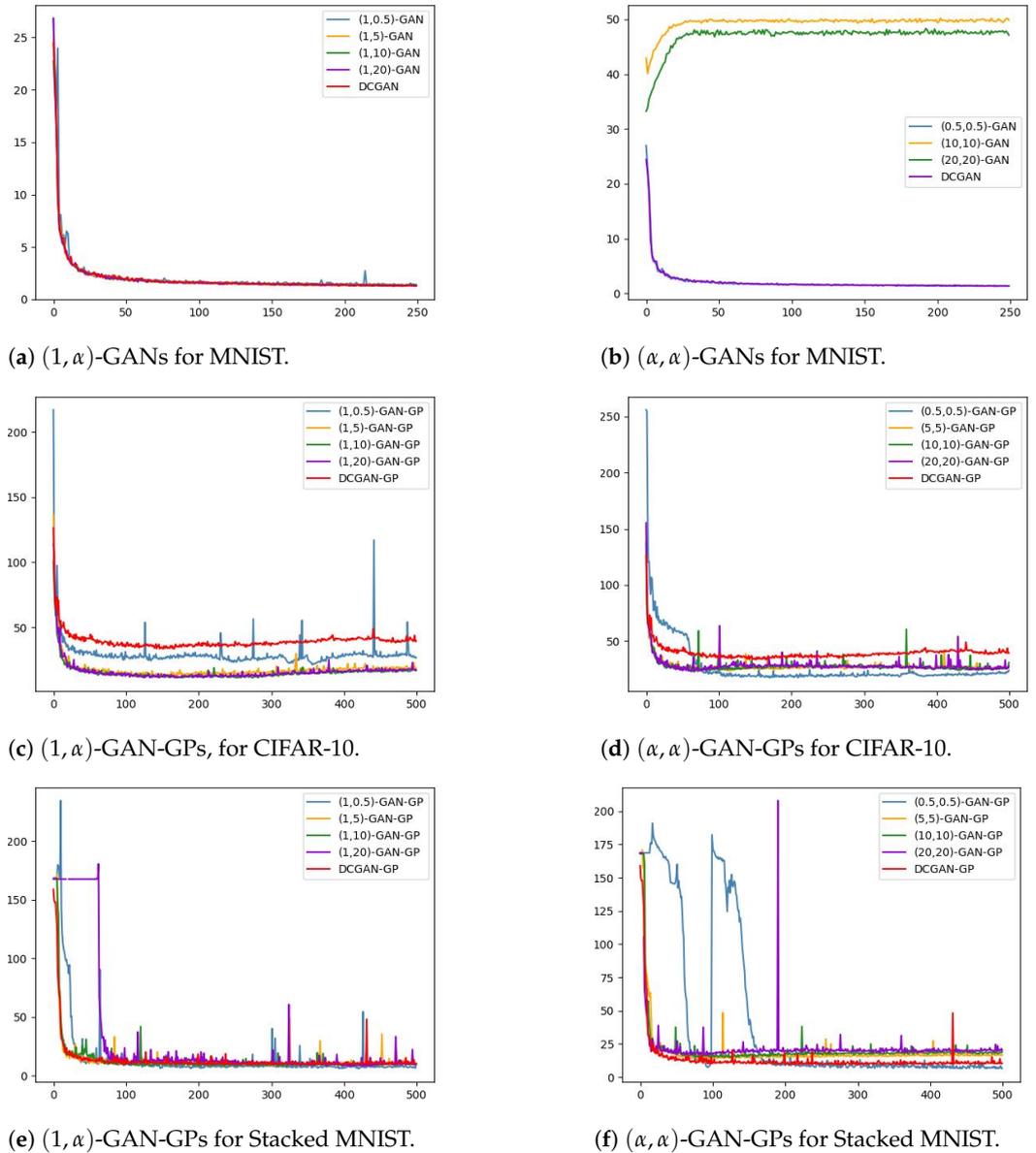
(a)

(b)

(c)
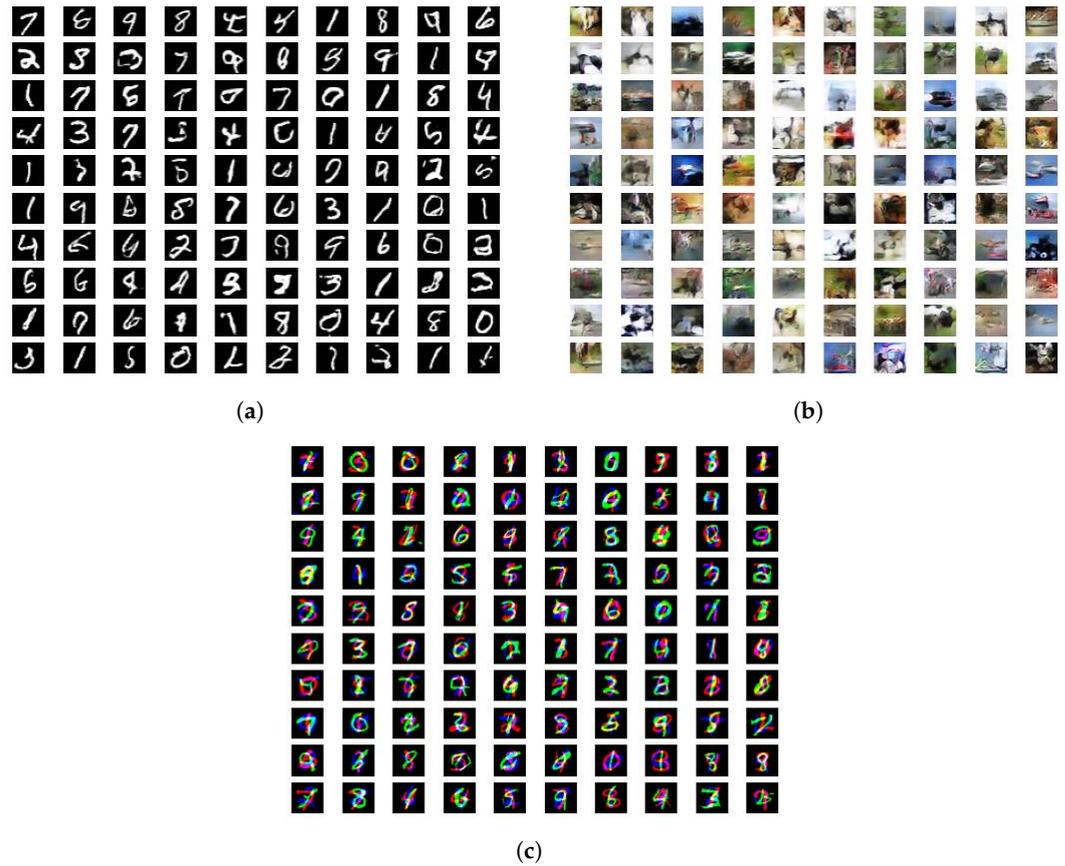
Figure 1. Generated images for the best-performing (α_D, α_G)-GANs. (**a**) (α_D, α_G)-GAN for MNIST, $\alpha_D = 1.0$, $\alpha_G = 5.0$, FID: 1.125. (**b**) (α_D, α_G)-GAN-GP for CIFAR-10, $\alpha_D = 1.0$, $\alpha_G = 20.0$, FID = 8.466. (**c**) (α_D, α_G)-GAN-GP for Stacked MNIST, $\alpha_D = 1.0$, $\alpha_G = 0.5$, FID = 4.833.

Table 2. (α_D, α_G)-GAN results for MNIST.

(α_D, α_G)-GAN	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/10)
(1,0.5)-GAN	1.264	1.288	2.979×10^{-4}	227.25	420.25	4
(0.5,0.5)-GAN	1.209	1.265	0.001	234.5	156.7	6
(1,5)-GAN	**1.125**	1.17	8.195×10^{-4}	230.3	617.344	10
(1,10)-GAN	1.147	**1.165**	7.984×10^{-4}	225.6	253.156	10
(10,10)-GAN	36.506	39.361	16.312	1.5	0.5	2
(1,20)-GAN	1.135	1.174	0.001	237.5	274.278	10
(20,20)-GAN	33.23	33.23	**0.0**	**1.0**	**0.0**	1
DCGAN	1.154	1.208	0.001	231.3	357.122	10

Table 3. (α_D, α_G)-GAN results for CIFAR-10.

(α_D, α_G)-GAN	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/5)
(1,0.5)-GAN-GP	10.551	14.938	12.272	326.2	1808.7	5
(0.5,0.5)-GAN-GP	13.734	14.93	0.517	223.6	11,378.3	5
(1,5)-GAN-GP	10.772	11.635	0.381	132.0	1233.5	5
(5,5)-GAN-GP	20.79	21.72	0.771	**84.8**	1527.2	5
(1,10)-GAN-GP	9.465	**10.187**	**0.199**	182.6	**1096.3**	5
(10,10)-GAN-GP	19.99	21.095	0.434	131.8	13,374.7	5
(1,20)-GAN-GP	**8.466**	10.217	1.479	216.2	6479.7	5
(20,20)-GAN-GP	19.378	21.216	2.315	138.2	29,824.2	5
DCGAN-GP	25.731	28.378	3.398	158.0	2510.5	5

(**a**) $(1, \alpha)$-GANs for MNIST.

(**b**) (α, α)-GANs for MNIST.

(**c**) $(1, \alpha)$-GAN-GPs, for CIFAR-10.

(**d**) (α, α)-GAN-GPs for CIFAR-10.

(**e**) $(1, \alpha)$-GAN-GPs for Stacked MNIST.

(**f**) (α, α)-GAN-GPs for Stacked MNIST.

Figure 2. Average FID scores vs. epochs for various (α_D, α_G)-GANs.

Table 4. (α_D, α_G)-GAN results for Stacked MNIST.

(α_D, α_G)-GAN	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/5)
(1,0.5)-GAN-GP	**4.833**	**4.997**	0.054	311.5	23,112.5	2
(0.5,0.5)-GAN-GP	6.418	6.418	**0.0**	479.0	**0.0**	1
(1,5)-GAN-GP	7.98	7.988	1.357×10^{-4}	379.5	11,704.5	2
(5,5)-GAN-GP	12.236	12.836	0.301	**91.5**	387.0	4
(1,10)-GAN-GP	7.502	7.528	0.001	326.5	14,280.5	2
(10,10)-GAN-GP	14.22	14.573	0.249	95.0	450.0	2
(1,20)-GAN-GP	8.379	8.379	**0.0**	427.0	**0.0**	1
(20,20)-GAN-GP	16.584	16.584	**0.0**	94.0	**0.0**	1
DCGAN-GP	7.507	7.774	0.064	303.4	11,870.8	5

(a)

(b)

(c)

Figure 3. Generated images for best-performing SL*k*GANs. (**a**) Vanilla-SL*k*GAN-0.25 for MNIST, FID = 1.112. (**b**) Vanilla-SL*k*GAN-2.0 for CIFAR-10, FID = 4.58. (**c**) Vanilla-SL*k*GAN-15.0-GP for Stacked MNIST, FID = 3.836.

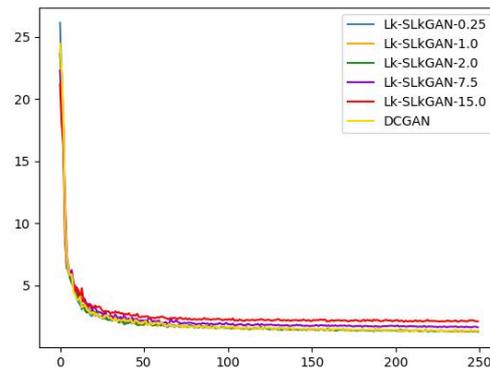**Table 5.** SL*k*GAN results for MNIST.

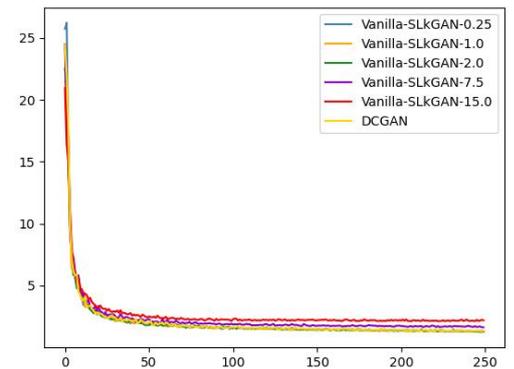| Variant-SL*k*GAN-*k*	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/10)
L*k*-SL*k*GAN-0.25	1.15	1.174	6.298×10^{-4}	224.3	940.9	10
Vanilla-SL*k*GAN-0.25	**1.112**	**1.162**	0.001	237.0	**124.0**	10
L*k*-SL*k*GAN-1.0	1.122	1.167	8.857×10^{-4}	233.0	124.0	10
Vanilla-SL*k*GAN-1.0	1.126	1.17	9.218×10^{-4}	226.2	1182.844	10
L*k*-SL*k*GAN-2.0	1.148	1.198	5.248×10^{-4}	237.2	288.4	10
Vanilla-SL*k*GAN-2.0	1.124	1.184	8.933×10^{-4}	237.8	138.4	10
L*k*-SL*k*GAN-7.5	1.455	1.498	4.422×10^{-4}	229.0	322.222	10
Vanilla-SL*k*GAN-7.5	1.439	1.511	0.001	212.2	1995.067	10
L*k*-SL*k*GAN-15.0	1.733	1.872	0.005	198.8	1885.733	10
Vanilla-SL*k*GAN-15.0	1.773	1.876	0.005	**171.6**	3122.267	10
DCGAN	1.154	1.208	0.001	231.3	357.122	10

Table 6. SL*k*GAN results for CIFAR-10.

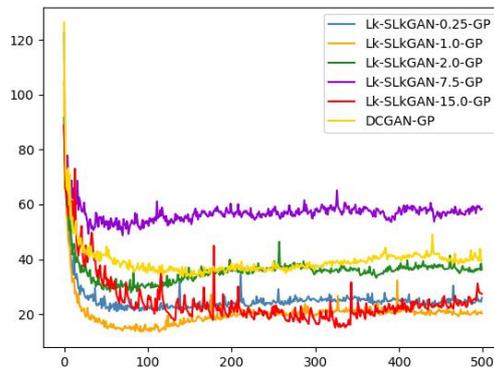| Variant-SL*k*GAN-*k*	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/5)
L*k*-SL*k*GAN-1.0	4.727	118.242	10,914.643	**60.8**	1897.2	5
Vanilla-SL*k*GAN-1.0	4.821	5.159	**0.092**	88.0	506.5	5
L*k*-SL*k*GAN-2.0	4.723	145.565	7492.26	73.2	3904.2	5
Vanilla-SL*k*GAN-2.0	**4.58**	**5.1**	0.261	105.4	740.8	5
L*k*-SL*k*GAN-7.5	6.556	155.497	7116.521	254.6	18,605.3	5
Vanilla-SL*k*GAN-7.5	6.384	48.905	8698.195	72.2	1711.7	5
L*k*-SL*k*GAN-15.0	8.576	145.774	5945.097	263.0	36,463.0	5
Vanilla-SL*k*GAN-15.0	7.431	50.868	8753.002	82.6	3106.8	5
DCGAN	4.753	5.194	0.117	88.6	**462.8**	5

Table 6. *Cont.*

Variant-SL*k*GAN-*k*	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/5)
L*k*-SL*k*GAN-0.25-GP	17.366	18.974	2.627	87.8	1897.2	5
Vanilla-SL*k*GAN-0.25-GP	16.013	17.912	1.961	189.0	9487.5	5
L*k*-SL*k*GAN-1.0-GP	10.771	12.567	1.083	77.8	**239.2**	5
Vanilla-SL*k*GAN-1.0-GP	8.569	9.588	**0.749**	197.6	2690.3	5
L*k*-SL*k*GAN-2.0-GP	23.11	25.013	1.924	**75.4**	658.8	5
Vanilla-SL*k*GAN-2.0-GP	28.215	29.69	1.242	232.0	20,438.5	5
L*k*-SL*k*GAN-7.5-GP	33.304	41.48	49.187	82.8	1081.2	5
Vanilla-SL*k*GAN-7.5-GP	33.085	34.799	1.597	290.8	12,714.7	5
L*k*-SL*k*GAN-15.0-GP	9.157	12.504	3.839	310.4	6976.8	5
Vanilla-SL*k*GAN-15.0-GP	**7.283**	**8.568**	1.535	185.6	5978.3	5
DCGAN-GP	25.731	28.378	3.398	158.0	2510.5	5

(**a**) L*k*-SL*k*GANs for MNIST.

(**b**) Vanilla-SL*k*GANs for MNIST.
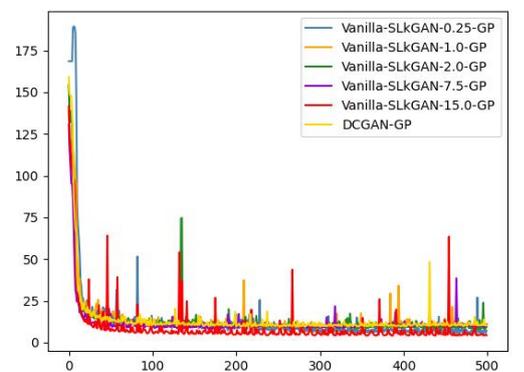

(**c**) L*k*-SL*k*GAN-GPs for CIFAR-10.
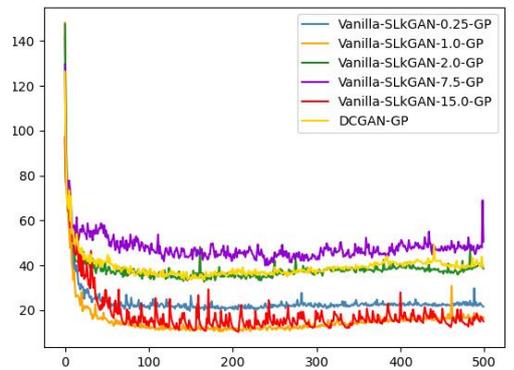
(**d**) Vanilla-SL*k*GAN-GPs for CIFAR-10.
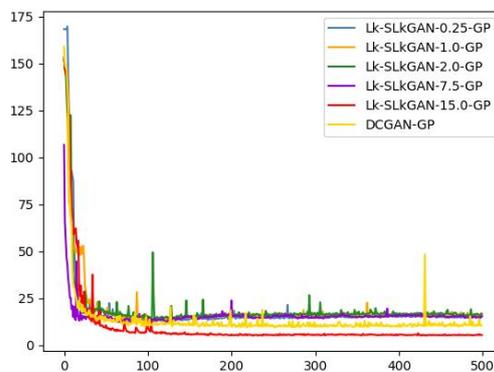
(**e**) L*k*-SL*k*GAN-GPs for Stacked MNIST.

(**f**) Vanilla-SL*k*GAN-GPs, Stacked MNIST.

Figure 4. FID scores vs. epochs for various SL*k*GANs.

Table 7. SL*k*GAN results for Stacked MNIST.

Variant-SL*k*GAN-*k*	Best FID Score	Average Best FID Score	Best FID Score Variance	Average Epochs	Epoch Variance	Number of Successful Trials (/5)
L*k*-SL*k*GAN-0.25-GP	10.541	11.824	0.678	113.6	356.3	5
Vanilla-SL*k*GAN-0.25-GP	5.197	5.197	**0.0**	496.0	**0.0**	1
L*k*-SL*k*GAN-1.0-GP	11.545	12.046	0.291	**89.0**	238.5	5
Vanilla-SL*k*GAN-1.0-GP	7.475	7.626	0.045	177.0	3528.0	2
L*k*-SL*k*GAN-2.0-GP	10.682	12.782	2.12	180.2	28,484.7	5
Vanilla-SL*k*GAN-2.0-GP	6.023	7.096	0.991	416.667	12,244.333	3
L*k*-SL*k*GAN-7.5-GP	8.912	9.906	0.577	239.0	35,663.5	5
Vanilla-SL*k*GAN-7.5-GP	6.074	6.43	0.164	238.0	21,729.5	5
L*k*-SL*k*GAN-15.0-GP	4.458	4.74	0.029	253.4	11,512.3	5
Vanilla-SL*k*GAN-15.0-GP	**3.836**	**3.873**	0.002	485.0	354.667	4
DCGAN-GP	7.507	7.774	0.064	303.4	11,870.8	5

4.3. Discussion

4.3.1. Experiment 1

From Table 2, we note that 37 of the 90 trials collapse before 250 epochs have passed without a gradient penalty. The (5,5)-GAN collapses for all five trials, and hence, it is not displayed in Table 2. This behavior is expected, as (α,α)-GAN is more sensitive to exploding gradients when α does not tend to 0 or $+\infty$ [8]. The addition of a gradient penalty could mitigate the discriminator's gradients diverging in the (5,5)-GAN by encouraging gradients to have a unit norm. Using a VanillaGAN discriminator with an α-GAN generator (i.e., $(1,\alpha)$-GAN) produces better quality images for all tested values of α compared to when both networks utilize an α-GAN loss function. The (1,10)-GAN achieves excellent stability, converging in all 10 trials, and also achieves the lowest average FID score. The (1,5)-GAN achieves the lowest FID score overall, marginally outperforming DCGAN. Note that when the average best FID score is very close to the best FID score, the resulting best FID score variance is quite small (of the order of 10^{-3}), indicating little statistical variability over the trials.

Likewise, for the CIFAR-10 and Stacked MNIST datasets, $(1,\alpha)$-GAN produces lower FID scores than (α,α)-GAN (see Tables 3 and 4). However, both models are more stable with the CIFAR-10 dataset. With the exception of DCGAN, no model converged to its best FID score for all five trials with the Stacked MNIST dataset. Comparing the trials that did converge, both (α,α)-GAN and $(1,\alpha)$-GAN performed better on the Stacked MNIST dataset than the CIFAR-10 dataset. For CIFAR-10, the (1,10)- and (1,20)-GANs produced the best overall FID score and the best average FID score, respectively. On the other hand, the (1,0.5)-GAN produced the best overall FID score and the best average FID score for the Stacked MNIST dataset. We also observe a tradeoff between speed and performance for the CIFAR-10 and Stacked MNIST datasets: the $(1,\alpha)$-GANs arrive at their lowest FID scores later than their respective (α,α)-GANs but achieve lower FID scores overall.

Comparing Figure 2c and Figure 2d, we observe that (α,α)-GAN-GP provides more stability than $(1,\alpha)$-GAN for lower values of α (i.e., $\alpha = 0.5$), while $(1,\alpha)$-GAN-GP exhibits more stability for higher α values ($\alpha = 10$ and $\alpha = 20$). Figure 2e,f show that the two α-GANs trained on the Stacked MNIST dataset exhibit unstable behavior earlier into training when $\alpha = 0.5$ or $\alpha = 20$. However, both systems stabilize and converge to their lowest FID scores as training progresses. The (0.5,0.5)-GAN-GP system in particular exhibits wildly erratic behavior for the first 200 epochs then finishes training with a stable trajectory that outperforms DCGAN-GP.

A future direction is to explore how the complexity of an image dataset influences the best choice of α. For example, the Stacked MNIST dataset might be considered to be less complex than CIFAR-10, as images in the Stacked MNIST dataset only contain four unique colors (black, red, green, and blue), while the CIFAR-10 dataset utilizes significantly more colors.

4.3.2. Experiment 2

We see from Table 5 that all Lk-LkGANs and Vanilla-SLkGANs have FID scores comparable to the DCGAN. When $k = 15$, Vanilla-SLkGAN and Lk-SLkGAN arrive at their lowest FID scores slightly earlier than DCGAN and other SLkGANs.

The addition of a simplified gradient penalty is necessary for Lk-SLkGAN to achieve overall good performance on the CIFAR-10 dataset (see Table 6). Interestingly, Vanilla-SLkGAN achieves lower FID scores without a gradient penalty for lower k values ($k = 1, 2$) and with a gradient penalty for higher k values ($k = 7.5, 15$). When $k = 0.25$, both SLkGANs collapsed for all five trials without a gradient penalty.

Table 7 shows that Vanilla-SLkGANs achieve better FID scores than their respective Lk-LkGAN counterparts. However, Lk-LkGANs are more stable, as no single trial collapsed, while 10 of the 25 Vanilla-SLkGAN trials collapsed before 500 epochs had passed. While all Vanilla-SLkGANs outperform the DCGAN with a gradient penalty, Lk-SLkGAN-GP only outperforms DCGAN-GP when $k = 15$. Except for when $k = 7.5$, we observe that the Lk-SLkGAN system takes fewer epochs to arrive at its lowest FID score. Comparing Figure 4e and Figure 4f, we observe that Lk-SLkGANs exhibit more stable FID score trajectories than their respective Vanilla-SLkGANs. This makes sense, as the LkGAN loss function aims to increase the GAN's stability compared to DCGAN [6].

5. Conclusions

We introduced a parameterized CPE-based generator loss function for a dual-objective GAN termed $\mathcal{L}_\alpha$-GAN that, when used in tandem with a canonical discriminator loss function that achieves its optimum in (11), minimizes a Jensen-f_α-divergence. We showed that this system can recover VanillaGAN, $(1, \alpha)$-GAN, and LkGAN as special cases. We conducted experiments with the three aforementioned $\mathcal{L}_\alpha$-GANs on three image datasets. The experiments indicate that $(1, \alpha)$-GAN exhibits better performance than (α, α)-GAN with $\alpha > 1$. They also show that the devised SLkGAN system achieves lower FID scores with a VanillaGAN discriminator compared with an LkGAN discriminator.

Future work consists of unveiling more examples of existing GANs that fall under our result as well as applying $\mathcal{L}_\alpha$-GAN to novel, judiciously designed CPE losses $\mathcal{L}_\alpha$ and evaluating the performance (in terms of both quality and diversity of generated samples) and the computational efficiency of the resulting models. Another interesting and related direction is to study $\mathcal{L}_\alpha$-GAN within the context of f-GANs, given that the Jensen-f-divergence is itself an f-divergence (see Remark 1), by systematically analyzing different Jensen-f-divergences and the role they play in improving GAN performance and stability. Other worthwhile directions include incorporating the proposed $\mathcal{L}_\alpha$ loss into state-of-the-art GAN models, such as, among others, BigGAN [33], StyleGAN [34], and CycleGAN [35], for high-resolution data generation and image-to-image translation applications and conducting a meticulous analysis of the sensitivity of the models' performance to different values of the α parameter and providing guidelines on how best to tune α for different types of datasets.

Author Contributions: Conceptualization, investigation and manuscript preparation, all authors; formal analysis, all authors; software development and simulation, J.V. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: All codes used in our experiments can be found at this https://github.com/justin-veiner/MASc, accessed on 20 February 2024.

Appendix A. Neural Network Architectures

We outline the architectures used for the generator and discriminator. For the MNIST dataset, we use the architectures of [6]. For the CIFAR-10 and Stacked MNIST datasets, we

base the architectures on [5]. We summarize some aliases for the architectures in Table A1. For all models, we use a batch size of 100 and a noise size of 784 for the generator input.

Table A1. Summary of aliases used to describe neural network architectures.

Alias	Definition
FC	Fully Connected
UpConv2D	Deconvolutional Layer
Conv2D	Convolutional Layer
BN	Batch Normalization
LeakyReLU	Leaky Rectified Linear Unit

We omit the bias in the convolutional and deconvolutional layers to decrease the number of parameters being trained, which in turn decreases computation times. We initialize our kernels using a normal distribution with zero mean and variance 0.01. We present the MNIST architectures in Tables A2 and A3 and the CIFAR-10 and Stacked MNIST architectures in Tables A4 and A5.

Table A2. Discriminator architecture for the MNIST dataset.

Layer	Output Size	Kernel	Stride	BN	Activation
Input	$28 \times 28 \times 1$	No			
Conv2D	$14 \times 14 \times 64$	5×5	2	No	LeakyReLU (0.3)
Dropout (0.3)				No	
Conv2D	$7 \times 7 \times 128$	5×5	2	No	LeakyReLU (0.3)
Dropout(0.3)				No	
FC	1			No	Sigmoid

Table A3. Generator architecture for the MNIST dataset.

Layer	Output Size	Kernel	Stride	BN	Activation
Input	784				
FC	$7 \times 7 \times 256$				
UpConv2D	$7 \times 7 \times 128$	5×5	1	Yes	LeakyReLU (0.3)
UpConv2D	$14 \times 14 \times 64$	5×5	2	Yes	LeakyReLU (0.3)
UpConv2D	$28 \times 28 \times 1$	5×5	2	No	Tanh

Table A4. Discriminator architecture for the CIFAR-10 and Stacked MNIST datasets.

Layer	Output Size	Kernel	Stride	BN	Activation
Input	$32 \times 32 \times 3$				
Conv2D	$16 \times 16 \times 128$	3×3	2	No	LeakyReLU (0.2)
Conv2D	$8 \times 8 \times 128$	3×3	2	No	LeakyReLU (0.2)
Conv2D	$4 \times 4 \times 256$	3×3	2	No	LeakyReLU (0.2)
Dropout (0.4)				No	
FC	1				Sigmoid

Table A5. Generator architecture for the CIFAR-10 and Stacked MNIST datasets.

Layer	Output Size	Kernel	Stride	BN	Activation
Input	784				
FC	$4 \times 4 \times 256$				
UpConv2D	$8 \times 8 \times 128$	4×4	2	Yes	LeakyReLU (0.2)
UpConv2D	$16 \times 16 \times 128$	4×4	2	Yes	LeakyReLU (0.2)
UpConv2D	$32 \times 32 \times 128$	4×4	2	Yes	LeakyReLU (0.2)
Conv2D	$32 \times 32 \times 3$	3×3	1	No	Tanh

Appendix B. Algorithms

We outline the algorithms used to train our models in Algorithms A1–A3.

Algorithm A1 Overview of (α_D, α_G)-GAN training

Require α_D, α_G, number of epochs n_e, batch size B, learning rate η
Initialize generator G with parameters $\boldsymbol{\theta}_G$, discriminator D with parameters $\boldsymbol{\theta}_D$.
for $i = 1$ to n_e **do**
     Sample batch of real data $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_B\}$ from dataset
     Sample batch of Gaussian noise vectors $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_B\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
     Update the discriminator's parameters using an Adam optimizer with learning rate
η by descending the gradient:

$$\nabla_{\boldsymbol{\theta}_D}\left(-\frac{1}{B}\sum_{i=1}^{B}\left(-\ell_\alpha(1, D(\mathbf{x}_i)) - \ell_\alpha(0, D(G(\mathbf{z}_i)))\right)\right)$$

     or **update** the discriminator's parameters with a simplified GP:

$$\nabla_{\boldsymbol{\theta}_D}\left(-\frac{1}{B}\sum_{i=1}^{B}\left(-\ell_\alpha(1, D(\mathbf{x}_i)) - \ell_\alpha(0, D(G(\mathbf{z}_i)))\right)\right.$$
$$\left. +5\left(\sum_{i=1}^{B}\left\|\nabla_{\mathbf{x}}\log\left(\frac{D(\mathbf{x})}{1-D(\mathbf{x})}\right)\right\|_2^2\right)\right)$$

     Update the generator's parameters using an Adam optimizer with learning rate η
and descending the gradient:

$$\nabla_{\boldsymbol{\theta}_G}\left(\frac{1}{B}\sum_{i=1}^{B}\ell_\alpha(0, D(G(\mathbf{z}_i)))\right)$$

end for

Algorithm A2 Overview of Lk-SLkGAN training

Require k, number of epochs n_e, batch size B, learning rate η
Initialize generator G with parameters $\boldsymbol{\theta}_G$, discriminator D with parameters $\boldsymbol{\theta}_D$.
for $i = 1$ to n_e **do**
     Sample batch of real data $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_B\}$ from dataset
     Sample batch of Gaussian noise vectors $\mathbf{z} = \{\mathbf{z}_1, \dots, \mathbf{z}_B\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
     Update the discriminator's parameters using an Adam optimizer with learning rate
η by descending the gradient:

$$\nabla_{\boldsymbol{\theta}_D}\left(\frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{2}(D(\mathbf{x}_i)-1)^2 + \frac{1}{2}(D(G(\mathbf{z}_i))^2)\right)\right)$$

     or **update** the discriminator's parameters with a simplified GP:

$$\nabla_{\boldsymbol{\theta}_D}\left(\frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{2}(D(\mathbf{x}_i)-1)^2 + \frac{1}{2}(D(G(\mathbf{z}_i))^2)\right)\right.$$
$$\left. +5\left(\sum_{i=1}^{B}\left\|\nabla_{\mathbf{x}}\log\left(\frac{D(\mathbf{x})}{1-D(\mathbf{x})}\right)\right\|_2^2\right)\right)$$

     Update the generator's parameters using an Adam optimizer with learning rate η
and descending the gradient:

$$\nabla_{\boldsymbol{\theta}_G}\left(\frac{1}{B}\sum_{i=1}^{B}\frac{1}{2}\left(|1 - D(G(\mathbf{z}_i))|^k - 1\right)\right)$$

end for

Algorithm A3 Overview of Vanilla-SLkGAN training

Require k, number of epochs n_e, batch size B, learning rate η

Initialize generator G with parameters $\boldsymbol{\theta}_G$, discriminator D with parameters $\boldsymbol{\theta}_D$.

for $i = 1$ to n_e **do**

    Sample batch of real data $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_B\}$ from dataset

    Sample batch of noise vectors $\mathbf{z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_B\} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

    Update the discriminator's parameters using an Adam optimizer with learning rate η by descending the gradient:

$$\nabla_{\boldsymbol{\theta}_D}\left(-\frac{1}{B}\sum_{i=1}^{B}(\log(D(\mathbf{x}_i)) + \log(1 - D(G(\mathbf{z}_i))))\right)$$

    or **update** the discriminator's parameters with a simplified (GP):

$$\nabla_{\boldsymbol{\theta}_D}\left(-\frac{1}{B}\sum_{i=1}^{B}(\log(D(\mathbf{x}_i)) + \log(1 - D(G(\mathbf{z}_i))))\right.$$
$$\left. +5\left(\sum_{i=1}^{B}\left\|\nabla_{\mathbf{x}}\log\left(\frac{D(\mathbf{x})}{1 - D(\mathbf{x})}\right)\right\|_2^2\right)\right)$$

    Update the generator's parameters using an Adam optimizer with learning rate η and descending the gradient:

$$\nabla_{\boldsymbol{\theta}_G}\left(\frac{1}{B}\sum_{i=1}^{B}\frac{1}{2}(|1 - D(G(\mathbf{z}_i))|^k - 1)\right)$$

end for

References

1. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27, pp. 2672–2680.
2. Kwon, Y.H.; Park, M.G. Predicting future frames using retrospective cycle GAN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
3. Pan, X.; Zhan, X.; Dai, B.; Lin, D.; Loy, C.C.; Luo, P. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7474–7489. [CrossRef] [PubMed]
4. Jordon, J.; Yoon, J.; Van Der Schaar, M. PATE-GAN: Generating synthetic data with differential privacy guarantees. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
5. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In Proceedings of the 9th International Conference on Image and Graphics, Shanghai, China, 13–15 September 2017; pp. 97–108.
6. Bhatia, H.; Paul, W.; Alajaji, F.; Gharesifard, B.; Burlina, P. Least kth-order and Rényi generative adversarial networks. *Neural Comput.* **2021**, *33*, 2473–2510. [CrossRef] [PubMed]
7. Nowozin, S.; Cseke, B.; Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2016; Volume 29.
8. Kurri, G.R.; Sypherd, T.; Sankar, L. Realizing GANs via a tunable loss function. In Proceedings of the IEEE Information Theory Workshop (ITW), Virtual, 17–21 October 2021; pp. 1–6.
9. Kurri, G.R.; Welfert, M.; Sypherd, T.; Sankar, L. α-GAN: Convergence and estimation guarantees. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, 26 June–1 July 2022; pp. 276–281.
10. Welfert, M.; Otstot, K.; Kurri, G.R.; Sankar, L. (α_D, α_G)-GANs: Addressing GAN training instabilities via dual objectives. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Taipei, Taiwan, 25–30 June 2023.
11. Welfert, M.; Kurri, G.R.; Otstot, K.; Sankar, L. Addressing GAN training instabilities via tunable classification losses. *arXiv* **2023**, arXiv:2310.18291.
12. Csiszar, I. Eine Informationstheoretische Ungleichung und ihre Anwendung auf den Bewis der Ergodizitat on Markhoffschen Ketten. *Publ. Math. Inst. Hung. Acad. Sci. Ser. A* **1963**, *8*, 85–108.
13. Csiszár, I. Information-type measures of difference of probability distributions and indirect observations. *Stud. Sci. Math. Hung.* **1967**, *2*, 299–318.

14. Ali, S.M.; Silvey, S.D. A general class of coefficients of divergence of one distribution from another. *J. R. Stat. Soc. Ser. (Methodol.)* **1966**, *28*, 131–142. [CrossRef]
15. Liese, F.; Vajda, I. On divergences and informations in statistics and information theory. *IEEE Trans. Inf. Theory* **2006**, *52*, 4394–4412. [CrossRef]
16. Kullback, S.; Leibler, R.A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
17. Nielsen, F. On a generalization of the Jensen–Shannon divergence and the Jensen–Shannon centroid. *Entropy* **2020**, *22*, 221. [CrossRef] [PubMed]
18. Nielsen, F.; Nock, R. On the chi square and higher-order chi distances for approximating f-divergences. *IEEE Signal Process. Lett.* **2013**, *21*, 10–13. [CrossRef]
19. Arimoto, S. Information-theoretical considerations on estimation problems. *Inf. Control.* **1971**, *19*, 181–194. [CrossRef]
20. Österreicher, F. On a class of perimeter-type distances of probability distributions. *Kybernetika* **1996**, *32*, 389–393.
21. Hellinger, E. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *J. Reine Angew. Math.* **1909**, *1909*, 210–271. [CrossRef]
22. Sason, I. On f-divergences: Integral representations, local behavior, and inequalities. *Entropy* **2018**, *20*, 383. [CrossRef] [PubMed]
23. Rényi, A. On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*; University of California Press: Berkeley, CA, USA, 1961; Volume 4, pp. 547–562.
24. Van Erven, T.; Harremos, P. Rényi divergence and Kullback-Leibler divergence. *IEEE Trans. Inf. Theory* **2014**, *60*, 3797–3820. [CrossRef]
25. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein generative adversarial networks. In Proceedings of the International Conference on Machine Learning, PMLR, Sydney, Australia, 6–11 August 2017; pp. 214–223.
26. Mao, X.; Li, Q.; Xie, H.; Lau, R.Y.; Wang, Z.; Paul Smolley, S. Least squares generative adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
27. Deng, L. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **2012**, *29*, 141–142. [CrossRef]
28. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images 2009. Available online: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (accessed on 22 February 2024).
29. Lin, Z.; Khetan, A.; Fanti, G.; Oh, S. PacGAN: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; Volume 31, pp. 1–10.
30. Kingma, D.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
31. Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; Courville, A.C. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30, pp. 1–11.
32. Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Hochreiter, S. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30, pp. 6626–6637.
33. Brock, A.; Donahue, J.; Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. *arXiv* **2018**, arXiv:1809.11096 .
34. Karras, T.; Laine, S.; Aila, T. A style-based generator architecture for generative adversarial networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4401–4410.
35. Almahairi, A.; Rajeshwar, S.; Sordoni, A.; Bachman, P.; Courville, A. Augmented CycleGAN: Learning many-to-many mappings from unpaired data. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 195–204.