Addressing GAN Training Instabilities via Tunable Classification Losses

Monica Welfert, *Graduate Student Member, IEEE*, Gowtham R. Kurri, *Member, IEEE*, Kyle Otstot, and Lalitha Sankar, *Senior Member, IEEE*

***Abstract*—Generative adversarial networks (GANs), modeled as a zero-sum game between a generator (G) and a discriminator (D), allow generating synthetic data with formal guarantees. Noting that D is a classifier, we begin by reformulating the GAN value function using class probability estimation (CPE) losses. We prove a two-way correspondence between CPE loss GANs and f-GANs which minimize f-divergences. We also show that all symmetric f-divergences are equivalent in convergence. In the finite sample and model capacity setting, we define and obtain bounds on estimation and generalization errors. We specialize these results to α-GANs, defined using α-loss, a tunable CPE loss family parametrized by $\alpha \in (0, \infty]$. We next introduce a class of dual-objective GANs to address training instabilities of GANs by modeling each player's objective using α-loss to obtain (α_D, α_G)-GANs. We show that the resulting non-zero sum game simplifies to minimizing an f-divergence under appropriate conditions on (α_D, α_G). Generalizing this dual-objective formulation using CPE losses, we define and obtain upper bounds on an appropriately defined estimation error. Finally, we highlight the value of tuning (α_D, α_G) in alleviating training instabilities for the synthetic 2D Gaussian mixture ring as well as the large publicly available Celeb-A and LSUN Classroom image datasets.**

***Index Terms*—Generative adversarial networks, CPE loss formulation, estimation error, training instabilities, dual objectives.**

I. Introduction

GENERATIVE adversarial networks (GANs) have become a crucial data-driven tool for generating synthetic data. GANs are generative models trained to produce samples from an unknown (real) distribution using a finite number of training data samples. They consist of two modules, a generator G and a discriminator D, parameterized by vectors $\theta \in \Theta \subset \mathbb{R}^{n_g}$ and $\omega \in \Omega \subset \mathbb{R}^{n_d}$, respectively, which play an adversarial game with each other. The generator G_θ maps noise $Z \sim P_Z$ to a data sample in $\mathcal{X}$ via the mapping $z \mapsto G_\theta(z)$ and aims to mimic data from the real distribution P_r. The discriminator D_ω takes as input $x \in \mathcal{X}$ and classifies it as real or generated by computing a score $D_\omega(x) \in [0, 1]$ which reflects the probability that x comes from P_r (real) as opposed to P_{G_θ} (synthetic). For a chosen value function $V(\theta, \omega)$, the adversarial game between G and D can be formulated as a zero-sum min-max problem given by

$$\inf_{\theta \in \Theta} \sup_{\omega \in \Omega} V(\theta, \omega). \tag{1}$$

Goodfellow et al. [1] introduce the vanilla GAN for which

$$V_{\mathrm{VG}}(\theta, \omega) = \mathbb{E}_{X \sim P_r}\big[\log D_\omega(X)\big] + \mathbb{E}_{X \sim P_{G_\theta}}\big[\log\left(1 - D_\omega(X)\right)\big].$$

For this V_{VG}, they show that when the discriminator class $\{D_\omega\}_{\omega \in \Omega}$ is rich enough, (1) simplifies to minimizing the Jensen-Shannon divergence [2] between P_r and P_{G_θ}.

Various other GANs have been studied in the literature using different value functions, including f-divergence based GANs called f-GANs [3], integral probability metric (IPM) based GANs [4], [5], [6], etc. Observing that the discriminator is a classifier, recently, in [7], we show that the value function in (1) can be written using a class probability estimation (CPE) loss $\ell(y, \hat{y})$ whose inputs are the true label $y \in \{0, 1\}$ and predictor $\hat{y} \in [0, 1]$ (soft prediction of y) as

$$V(\theta, \omega) = \mathbb{E}_{X \sim P_r}[-\ell(1, D_\omega(X))] + \mathbb{E}_{X \sim P_{G_\theta}}[-\ell(0, D_\omega(X))].$$

We further introduce α-GAN in [7] using the tunable CPE loss α-loss [8], [9], defined for $\alpha \in (0, \infty]$ as

$$\ell_\alpha(y, \hat{y}) \coloneqq \frac{\alpha}{\alpha - 1}\left(1 - y\hat{y}^{\frac{\alpha-1}{\alpha}} - (1 - y)\left(1 - \hat{y}\right)^{\frac{\alpha-1}{\alpha}}\right), \tag{2}$$

and show that this α-GAN formulation recovers various f-divergence based GANs including the Hellinger GAN [3] ($\alpha = 1/2$), the vanilla GAN [1] ($\alpha = 1$), and the Total Variation (TV) GAN [3] ($\alpha = \infty$). Furthermore, for a large enough discriminator class, we also show that the min-max optimization for α-GAN in (1) simplifies to minimizing the Arimoto divergence [10], [11]. In [12], we also show that the resulting Arimoto divergences are equivalent in convergence.

While each of the abovementioned GANs have distinct advantages, they continue to suffer from one or more types of training instabilities, including vanishing/exploding gradients, mode collapse, and sensitivity to hyperparameter tuning.

Manuscript received 11 October 2023; revised 30 April 2024; accepted 10 June 2024. Date of publication 19 June 2024; date of current version 30 July 2024. This work was supported in part by the National Science Foundation (NSF) under Grant CIF-1901243, Grant CIF-1815361, Grant CIF-2007688, Grant DMS-2134256, and Grant SCH-2205080. Some parts of this work have been presented at the 2021 Information Theory Workshop, the 2022 International Symposium on Information Theory, and the 2023 International Symposium on Information Theory. *(Monica Welfert and Gowtham R. Kurri contributed equally to this work.) (Corresponding author: Monica Welfert.)*

Monica Welfert, Kyle Otstot, and Lalitha Sankar are with the School of Electrical, Computer, and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: mwelfert@asu.edu; kotstot@asu.edu; lsankar@asu.edu).

Gowtham R. Kurri was with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85281 USA. He is now with the Signal Processing and Communications Research Centre, International Institute of Information Technology, Hyderabad 500032, India (e-mail: gowtham.kurri@iiit.ac.in).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSAIT.2024.3415670, provided by the authors.

Digital Object Identifier 10.1109/JSAIT.2024.3415670

In [1], Goodfellow et al. note that the generator's objective in the vanilla GAN can *saturate* early in training (due to the use of the sigmoid activation) when D can easily distinguish between the real and synthetic samples, i.e., when the output of D is near zero for all synthetic samples, leading to vanishing gradients. Furthermore, a confident D induces a steep gradient at samples close to the real data, thereby preventing G from learning such samples due to exploding gradients. To alleviate these problems, [1] proposes a *non-saturating* (NS) generator objective:

$$V_{\mathrm{VG}}^{\mathrm{NS}}(\theta,\omega)=\mathbb{E}_{X\sim P_{G_\theta}}\big[-\log D_\omega(X)\big]. \tag{3}$$

This NS version of the vanilla GAN may be viewed as involving different objective functions for the two players (in fact, with two versions of the $\alpha = 1$ CPE loss, i.e., log-loss, for D and G). However, it continues to suffer from mode collapse [13], [14] due to failure to converge and sensitivity to hyperparameter initialization (e.g., learning rate) because of large gradients. While other dual-objective GANs have also been proposed (e.g., Least Squares GAN (LSGAN) [15], RényiGAN [16], NS f-GAN [3], hybrid f-GAN [17]), few have successfully addressed the landscape of training instabilities.

Recent results have shown that α-loss demonstrates desirable gradient behaviors for different α values [9]. These results also assure learning robust classifiers that can reduce the confidence of D (a classifier); this, in turn, can allow G to learn without gradient issues. More broadly, by using different loss-based value functions for D and G, we can fully exploit this varying gradient behavior. To this end, in [18] we introduce a different α-loss objective[1] for each player and propose a tunable dual-objective (α_D, α_G)-GAN, where the value functions of D and G are written in terms of α-loss with parameters $\alpha_D \in (0, \infty]$ and $\alpha_G \in (0, \infty]$, respectively.

This paper ties together and significantly enhances our prior results investigating single-objective CPE loss-based GANs including α-GAN [7], [12] and dual-objective GANs including (α_D, α_G)-GANs [18]. One of the most important contributions of this work is taking a loss function perspective of GANs, be it single- or dual-objective. CPE loss based GANs can be easier to implement, provide a more intuitive formulation, and also provide a unifying framework for analyzing convergence guarantees in addition to generalization and estimation error bounds. Moreover, compared to f-GANs, this formulation more clearly emphasizes the connection between the discriminator as a classifier and the divergence being minimized. We list below all our contributions (while highlighting novelty relative to [7], [12], [18]) for both single- and dual-objective GANs.

A. Our Contributions

Single-objective GANs:

- We review CPE loss GANs and include a two-way correspondence between CPE loss GANs and f-divergences (Theorem 1) previously published in [12]. We note that we include a more comprehensive proof of this result here. We review α-GANs, originally proposed in [7], and present the optimal strategies for G and D, provided they have sufficiently large capacity and infinite samples (Theorem 2). We also include a result from [7] showing that α-GAN interpolates between various f-GANs including vanilla GAN ($\alpha = 1$), Hellinger GAN [3] ($\alpha = 1/2$), and Total Variation GAN [3] ($\alpha = \infty$) by tuning α (Theorem 3).
- A novel contribution of this work is proving an equivalence between a CPE loss GAN and a corresponding f-GAN (Theorem 5). We specialize this for α-GANs and f_α-GANs to show that one can go between the two formulations using a bijective activation function (Theorem 4).
- We study *convergence* properties of CPE loss GANs in the presence of sufficiently large number of samples and discriminator capacity. We show that all symmetric f-divergences are *equivalent* in convergence (Theorem 6) generalizing an equivalence proven in our prior work [12] for Arimoto divergences. We remark that the proof techniques used here give rise to a conceptually simpler proof of equivalence between Jensen-Shannon divergence and total variation distance proved earlier by Arjovsky et al. [4, Th. 2(1)].
- In the setting of finite training samples and limited capacity for the generator and discriminator models, we extend the definition of generalization, first introduced by Arora et al. [19], to CPE loss GANs. We do so by introducing a refined neural net divergence and prove that it indeed generalizes with increasing number of training samples (Theorem 7).
- To conclude our results on single-objective GANs, we review the definition of estimation error for CPE loss GANs introduced in [12], present an upper bound on the error originally proven in [12] (Theorem 8), and a matching lower bound under additional assumptions for α-GANs previously proven in [18] (Theorem 9).

Dual-objective GANs:

- We begin by reviewing (α_D, α_G)-GANs, originally proposed in [18], and the corresponding optimal strategies for D and G for appropriate (α_D, α_G) values (Theorem 10). We also review the non-saturating version of (α_D, α_G)-GANs, also proposed in [18], and present its Nash equilibrium strategies for D and G (Theorem 11).
- A novel contribution of this work is a gradient analysis highlighting the effect of tuning (α_D, α_G) on the magnitude of the gradient of the generator's loss for both the saturating and non-saturating versions of the (α_D, α_G)-GAN formulation (Theorem 12).
- We introduce a dual-objective CPE loss GAN formulation generalizing our dual-objective (α_D, α_G)-GAN formulation in [18]. For this non-zero sum game, we present the optimal strategies for D and G and prove that for the optimal D_{ω^*}, G minimizes an f-divergence under certain conditions (Proposition 1).
- We generalize the definition of estimation error we introduced in [18] for (α_D, α_G)-GANs to dual-objective CPE loss GANs. We present an upper bound on the error

[1] Throughout the paper, we use the terms objective and value function interchangeably.

(Theorem 13), and show that this result subsumes that for (α_D, α_G)-GANs in [18].

- Focusing on (α_D, α_G)-GANs, we demonstrate empirically that tuning α_D and α_G significantly reduces vanishing and exploding gradients and alleviates mode collapse on a synthetic 2D-ring dataset (originally published in [18]). For the high-dimensional Celeb-A and LSUN Classroom datasets, we show that our tunable approach is more robust in terms of the Fréchet Inception Distance (FID) to the choice of GAN hyperparameters, including number of training epochs and learning rate, relative to both vanilla GAN and LSGAN.
- Finally, throughout the paper, we illustrate the effect of tuning (α_D, α_G) on training instabilities including vanishing and exploding gradients, as well as model oscillation and mode collapse.

B. Related Work

GANs face several challenges that threaten their training stability [1], [20], [21], [22], such as vanishing/exploding gradients, mode collapse, sensitivity to hyperparameter initialization, and model oscillation, which occurs when the generated data oscillates around modes in real data due to large gradients. Many GAN variants have been proposed to stabilize training by changing the objective optimized [1], [3], [4], [15], [16], [17], [23], [24], [25], [26], [27] or the architecture design [28], [29], [30], [31]. Since we focus on tuning the objective, we restrict discussions and comparisons to similar approaches. Approaches modifying the objective can be categorized as single-objective or dual-objective variants. For the single objective setting, arguing that vanishing gradients are due to the sensitivity of f-divergences to mismatch in distribution supports, Arjovsky et al. [4] proposed Wasserstein GAN (WGAN) using a "weaker" Euclidean distance between distributions. However, this formulation requires a Lipschitz constraint on D, which in practice is achieved either via clipping model weights or using a computationally expensive gradient penalty method [25]. More generally, a broader class of GANs based on IPM distances have been proposed, including MMD GANs [32], [33], Sobolev GANs [34], (surveyed in [6]), and total variation GANs [35]. Our work focuses on classifier based GANs, and does not require clipping or penalty methods, thus limiting meaningful comparisons with IPM-based GANs. Finally, for single-objective GANs, many theoretical approaches to GANs assume that a particular divergence is minimized and study the role of regularization methods [36], [37]. Our work goes beyond these approaches by explicitly analyzing the value function optimizations of both D and G, thereby enabling understanding and addressing training instabilities.

Noting the benefit of using different objectives for the D and G, various dual-objective GANs, beyond the NS vanilla GAN, have been proposed. Mao et al. [15] proposed Least Squares GAN (LSGAN) where the objectives for D and G use different linear combinations of squared loss-based measures. LSGANs can be viewed as state of the art in highlighting the effect of objective in GAN performance; therefore, in addition to vanilla GAN, we contrast our results to this work, as it allows for a fair comparison when choosing the same hyperparameters including model architecture, learning rate, initialization, optimization methodology, etc. for both approaches. Dual objective variants including RényiGAN [16], least kth-order GANs [16], NS f-GAN [3], and hybrid f-GAN [17] have also been proposed. Recently, [38] attempts to unify a variety of divergence-based GANs (including special cases of both our (α_D, α_G)-GANs and LSGANs) via $\mathcal{L}_\alpha$-GANs. However, our work is distinct in highlighting the role of GAN objectives in reducing training instabilities. Finally, it is worth mentioning that dual objectives have been shown to be essential in the context of learning models robust to adversarial attacks [39].

Generalization for single-objective GANs was first introduced by Arora et al. [19]. Our work is the first to extend the definition of generalization to incorporate CPE losses. There is a growing interest in studying and constructing bounds on the estimation error in training GANs [6], [40], [41]. Estimation error evaluates the performance of a limited fixed capacity generator (e.g., a class of neural networks) learned with finite samples relative to the best generator. The results in [6], [40], [41] study estimation error using a specific formulation that does not take into account the loss used and also define estimation error only in the single-objective setting. In this work, we study the impact of the loss used as well as the dual-objective formulation on the estimation error guarantees. To the best of our knowledge, this is the first result of this kind for dual-objective GANs.

The remainder of the paper is organized as follows. We review various GANs in the literature, classification loss functions, particularly α-loss, and GAN training instabilities in Section II. In Section III, we present and analyze the loss function perspective of GANs and introduce tunable α-GANs. In Section IV, we propose and analyze dual-objective (α_D, α_G)-GANs and introduce a dual-objective CPE-loss GAN formulation. Finally, in Section V, we highlight the value of tuning (α_D, α_G) for (α_D, α_G)-GANs on several datasets. Proof sketches for our results are included in this manuscript; detailed proofs and additional experimental results can be found in the accompanying supplementary material (Appendices A-P).

II. Preliminaries: Overview of GANs and Loss Functions for Classification

A. Background on GANs

We begin by presenting an overview of GANs in the literature. Let P_r be a probability distribution over $\mathcal{X} \subset \mathbb{R}^d$, which the generator wants to learn *implicitly* by producing samples by playing a competitive game with a discriminator in an adversarial manner. We parameterize the generator G and the discriminator D by vectors $\theta \in \Theta \subset \mathbb{R}^{n_g}$ and $\omega \in \Omega \subset \mathbb{R}^{n_d}$, respectively, and write G_θ and D_ω (θ and ω are typically the weights of neural network models for the generator and the discriminator, respectively). The generator G_θ takes as input a $d'(\ll d)$-dimensional latent noise $Z \sim P_Z$ and maps it to a data point in $\mathcal{X}$ via the mapping $z \mapsto G_\theta(z)$. For an input

$x \in \mathcal{X}$, the discriminator outputs $D_\omega(x) \in [0, 1]$, an estimate of the probability that x comes from P_r (real) as opposed to P_{G_θ} (synthetic). The generator and the discriminator play a two-player min-max game with a value function $V(\theta, \omega)$, resulting in a saddle-point optimization problem given by

$$\inf_{\theta\in\Theta} \sup_{\omega\in\Omega} V(\theta, \omega). \tag{4}$$

Goodfellow et al. [1] introduced the vanilla GAN using

$$\begin{aligned} V_{\mathrm{VG}}(\theta, \omega) &= \mathbb{E}_{X\sim P_r}\big[\log D_\omega(X)\big] \\ &\quad + \mathbb{E}_{Z\sim P_Z}\big[\log\left(1 - D_\omega(G_\theta(Z))\right)\big] \\ &= \mathbb{E}_{X\sim P_r}\big[\log D_\omega(X)\big] + \mathbb{E}_{X\sim P_{G_\theta}}\big[\log\left(1 - D_\omega(X)\right)\big], \end{aligned} \tag{5}$$

for which they showed that when the discriminator class $\{D_\omega\}$, parametrized by ω, is rich enough, (4) simplifies to finding $\inf_{\theta\in\Theta} 2D_{\mathrm{JS}}(P_r||P_{G_\theta}) - \log 4$, where $D_{\mathrm{JS}}(P_r||P_{G_\theta})$ is the Jensen-Shannon divergence [2] between P_r and P_{G_θ}. This simplification is achieved, for any G_θ, by choosing the optimal discriminator

$$D_{\omega^*}(x) = \frac{p_r(x)}{p_r(x) + p_{G_\theta}(x)}, \quad x \in \mathcal{X}, \tag{6}$$

where p_r and p_{G_θ} are the corresponding densities of the distributions P_r and P_{G_θ}, respectively, with respect to a base measure dx (e.g., Lebesgue measure).

Generalizing this by leveraging the variational characterization of f-divergences [42], Nowozin et al. [3] introduced f-GANs via the value function

$$V_f(\theta, \omega) = \mathbb{E}_{X\sim P_r}[D_\omega(X)] + \mathbb{E}_{X\sim P_{G_\theta}}\big[-f^*(D_\omega(X))\big], \tag{7}$$

where[2] $D_\omega : \mathcal{X} \to \mathbb{R}$ and $f^*(t) := \sup_u\{ut - f(u)\}$ is the Fenchel conjugate of a convex lower semicontinuous function f defining an f-divergence[3] $D_f(P_r||P_{G_\theta}) := \int_{\mathcal{X}} p_{G_\theta}(x) f(\frac{p_r(x)}{p_{G_\theta}(x)})dx$ [44], [45]. In particular, $\sup_{\omega\in\Omega} V_f(\theta, \omega) = D_f(P_r||P_{G_\theta})$ when there exists $\omega^* \in \Omega$ such that $D_{\omega^*}(x) = f'(\frac{p_r(x)}{p_{G_\theta}(x)})$. In order to respect the domain $\mathrm{dom}(f^*)$ of the conjugate f^*, Nowozin et al. further decomposed (7) by assuming the discriminator D_ω can be represented in the form $D_\omega(x) = g_f(Q_\omega(x))$, yielding the value function

$$\widetilde{V}_f(\theta, \omega) = \mathbb{E}_{X\sim P_r}\big[g_f(Q_\omega(x))\big] + \mathbb{E}_{X\sim P_{G_\theta}}\big[-f^*(g_f(Q_\omega(x)))\big], \tag{8}$$

where $Q_\omega : \mathcal{X} \to \mathbb{R}$ and $g_f : \mathbb{R} \to \mathrm{dom}(f^*)$ is an output activation function specific to the f-divergence used.

Highlighting the problems with the continuity of various f-divergences (e.g., Jensen-Shannon, KL, reverse KL, total variation) over the parameter space Θ [13], Arjovsky et al. [4] proposed Wasserstein-GAN (WGAN) using the following Earth Mover's (also called Wasserstein-1) distance:

$$W(P_r, P_{G_\theta}) = \inf_{\Gamma_{X_1X_2}\in\Pi(P_r,P_{G_\theta})} \mathbb{E}_{(X_1,X_2)\sim\Gamma_{X_1X_2}} \|X_1 - X_2\|_2, \tag{9}$$

where $\Pi(P_r, P_{G_\theta})$ is the set of all joint distributions $\Gamma_{X_1X_2}$ with marginals P_r and P_{G_θ}. WGAN employs the Kantorovich-Rubinstein duality [46] using the value function

$$V_{\mathrm{WGAN}}(\theta, \omega) = \mathbb{E}_{X\sim P_r}[D_\omega(X)] - \mathbb{E}_{X\sim P_{G_\theta}}[D_\omega(X)], \tag{10}$$

where the functions $D_\omega : \mathcal{X} \to \mathbb{R}$ are all 1-Lipschitz, to simplify $\sup_{\omega\in\Omega} V_{\mathrm{WGAN}}(\theta, \omega)$ to $W(P_r, P_{G_\theta})$ when the class Ω is rich enough. Although various GANs have been proposed in the literature, each of them exhibits their own strengths and weaknesses in terms of convergence, vanishing/exploding gradients, mode collapse, computational complexity, etc., leaving the problem of addresing GAN training instabilities unresolved [14].

B. Background on Loss Functions for Classification

The ideal loss function for classification is the Bayes loss, also known as the 0-1 loss. However, the complexity of implementing such a non-convex loss has led to much interest in seeking surrogate loss functions for classification. Several surrogate losses with desirable properties have been proposed to train classifiers; the most oft-used and popular among them is log-loss, also referred to as cross-entropy loss. However, enhancing robustness of classifier has broadened the search for better surrogate losses or families of losses; one such family is the class probability estimator (CPE) losses that operate on a soft probability or risk estimate. Recently, it has been shown that a large class of known CPE losses can be captured by a tunable loss family called α-loss, which includes the well-studied exponential loss ($\alpha = 1/2$), log-loss ($\alpha = 1$), and soft 0-1 loss, i.e., the probability of error ($\alpha = \infty$). Formally, α-loss is defined as follows.

Definition 1 (Sypherd et al. [9]): For a set of distributions $\mathcal{P}(\mathcal{Y})$ over $\mathcal{Y}$, α-loss $\ell_\alpha : \mathcal{Y} \times \mathcal{P}(\mathcal{Y}) \to \mathbb{R}_+$ for $\alpha \in (0, 1) \cup (1, \infty)$ is defined as

$$\ell_\alpha(y, \hat{P}) = \frac{\alpha}{\alpha - 1}\left(1 - \hat{P}(y)^{\frac{\alpha-1}{\alpha}}\right). \tag{11}$$

By continuous extension, $\ell_1(y, \hat{P}) = -\log \hat{P}(y)$, $\ell_\infty(y, \hat{P}) = 1 - \hat{P}(y)$, and $\ell_0(y, \hat{P}) = \infty$.

Note that $\ell_{1/2}(y, \hat{P}) = \hat{P}(y)^{-1} - 1$, which is related to the exponential loss, particularly in the margin-based form [9]. Also, α-loss is convex in the probability term $\hat{P}(y)$. Regarding the history of (11), Arimoto first studied α-loss in finite-parameter estimation problems [47], and later Liao et al. [48] independently introduced and used α-loss to model the inferential capacity of an adversary to obtain private attributes. Most recently, Sypherd et al. [9] studied α-loss extensively in the classification setting, which is an impetus for this work.

C. Background on GAN Training Instabilities

GANs face several challenges during training. Imbalanced performance between the generator and discriminator often coincides with the presence of exploding and vanishing gradients. When updating the generator weights during the

[2]This is a slight abuse of notation in that D_ω does not map to $[0, 1]$ here. However, we chose this for consistency in notation of discriminator across various GANs.

[3]In the classical information theory literature, two more conditions on f are considered: $f(1) = 0$ and f strictly convex at 1, which are regularity conditions. The condition $f(1) = 0$ is mainly required for non-negativity of $D_f(\cdot||\cdot)$; otherwise, $D_f(\cdot||\cdot) \geq f(1)$. The condition of f being strictly convex at 1 is required for $D_f(P||Q) = 0 \implies P = Q$; however, for our most general analysis in Theorems 1, 4, and 5, we relax these two conditions in line with [43].

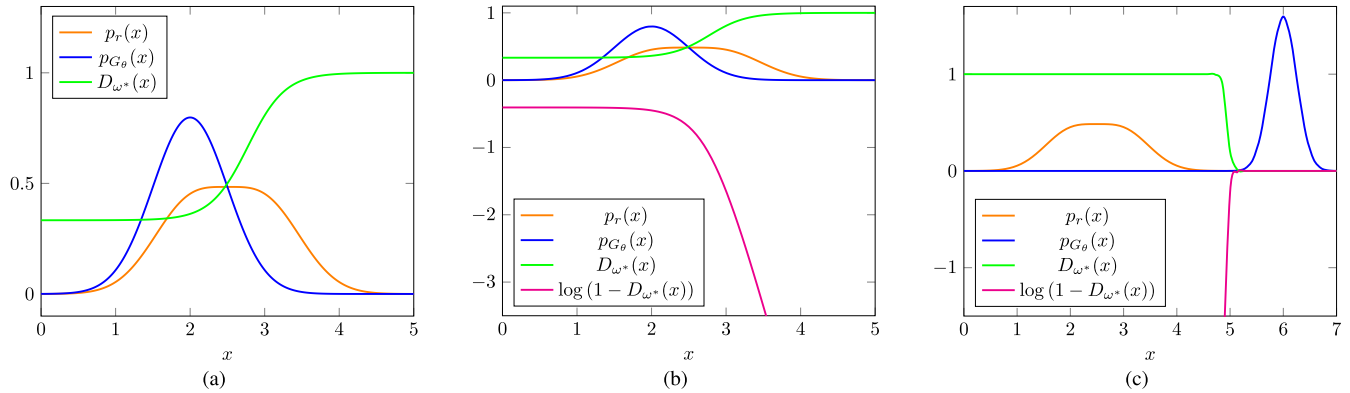


Fig. 1. A toy example of the vanilla GAN illustrating vanishing and exploding gradients, where the real distribution $P_r = 0.5\mathcal{N}(2, 0.5^2) + 0.5\mathcal{N}(3, 0.5^2)$ (orange curve) and the assumed initial generated distribution $P_{G_\theta} = \mathcal{N}(2, 0.5^2)$ (blue curve). (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (6) (green curve). (b) A plot of the generator's saturating loss $\log(1 - D_{\omega^*}(x))$ (pink curve). The rightmost generated samples receive steep gradients (exploding gradients) which causes the generated data to overshoot the real data mode toward the $D_{\omega^*}(x) \approx 1$ region. (c) For this saturating generator loss setting, following the generator's update using (b), when the discriminator updates, the generated samples now receive flat gradients (vanishing gradients), thus freezing P_{G_θ}.

backward pass of the network $G_\theta \circ D_\omega$, the gradients are computed by propagating the gradient of the value function from the output layer of D_ω to the input layer of G_θ, following the chain rule of derivatives. Each layer contributes to the gradient update by multiplying the incoming gradient with the local gradient of its activation function, and passing it to the preceding layer. When the gradients become large, the successive multiplication of these gradients across the layers can result in an exponential growth, known as *exploding gradients*. Conversely, small gradients can lead to an exponential decay, referred to as *vanishing gradients*. In both cases, networks with multiple hidden layers are particularly susceptible to unstable weight updates, causing extremely large or small values that may overflow or underflow the numerical range of computations, respectively.

In the context of the vanilla GAN, exploding gradients can occur when the generator successfully produces samples that are severely misclassified (close to 1) by the discriminator. During training, the generator is updated using the loss function $\log(1 - D_\omega(x))$, which diverges to $-\infty$ as the discriminator output $D_\omega(x)$ approaches 1. Consequently, the gradients for the generator weights fail to converge to non-zero values, leading to the generated data potentially overshooting the real data in any direction. This is illustrated in Fig. 1(b), relative to an initial starting point in Fig. 1(a). In severe cases of exploding gradients, the weight update can push the generated data towards a region far from the real data. As a result, the discriminator can easily assign scores close to zero to the generated data and close to one to the real data. As the discriminator output approaches zero, the generator's loss function saturates, causing the gradients of the generator weights to gradually vanish. This is shown in Fig. 1(c). The conflation of these two phenomena can prevent the generator from effectively correcting itself and improving its performance over time.

To alleviate the issues of exploding and vanishing gradients, Goodfellow et al. [1] proposed a *non-saturating* (NS) generator objective:

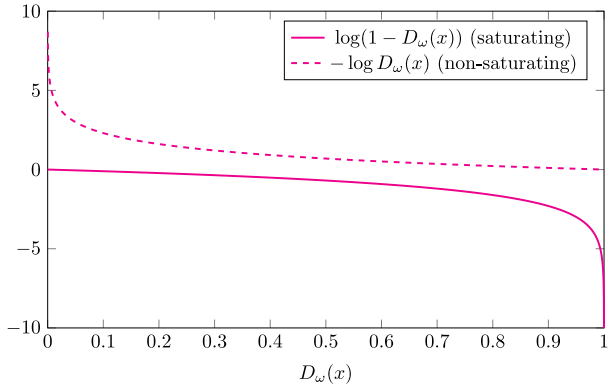


Fig. 2. A plot of the vanilla GAN generator's saturating loss $\log(1 - D_{\omega^*}(x))$ and non-saturating loss $-\log(D_{\omega^*}(x))$.

$$V_{\text{VG}}^{\text{NS}}(\theta, \omega) = \mathbb{E}_{X \sim P_{G_\theta}}\big[-\log D_\omega(X)\big]. \tag{12}$$

The use of this non-saturating objective provides a more intuitive optimization trajectory that allows the generated distribution P_{G_θ} to converge to the real distribution P_r. As the discriminator output $D_\omega(x)$ for a sample x approaches 1, the generator loss $-\log D_\omega(x)$ approaches zero, indicating that the generated data is closer to the real distribution Additionally, with a high-performing discriminator, the generator receives steep gradients (as opposed to vanishing gradients) during the update process; this occurs because the generator loss diverges to $+\infty$ as the discriminator output approaches zero (see Fig. 2). As we show in the sequel, using α-loss based value functions allow modulating the magnitude of the gradient (and therefore, how steeply it rises), thereby improving over the vanilla GAN performance.

While the non-saturating vanilla GAN (an industry standard) incorporates two different objective functions for the generator and discriminator in order to combat vanishing and exploding gradients, it can still suffer from mode collapse and oscillations [13], [14]. These issues often arise due to the sensitivity of the GAN to hyperparameter initialization. The

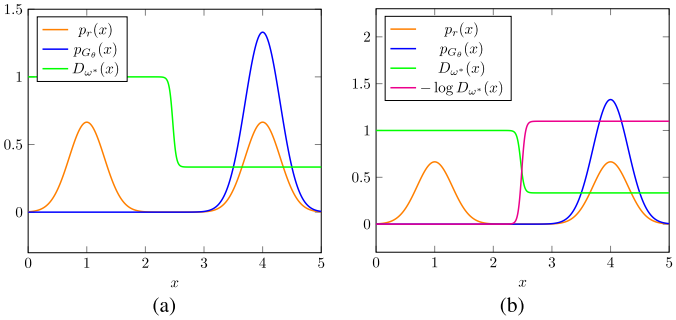


Fig. 3. A toy example of the vanilla GAN illustrating mode collapse, where $P_r = 0.5\mathcal{N}(1, 0.3^2) + 0.5\mathcal{N}(4, 0.3^2)$ (orange curve) and $P_{G_\theta} = \mathcal{N}(4, 0.3^2)$ (blue curve). (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (6). The discriminator output is flat in the dense p_{G_θ} region. (b) A plot of the generator's non-saturating loss $-\log(D_{\omega^*}(x))$. The loss is also flat in the dense p_{G_θ} region, causing the generator to receive near-zero gradients, thus appearing to "collapse" on the real data mode.

problem of *mode collapse* occurs when the generator produces samples that closely resemble only a limited subset of the real data. In such cases, the generator lacks the incentive to capture the remaining modes since the discriminator struggles to effectively differentiate between the real and generated samples. One possible explanation for this phenomenon, as depicted in Fig. 3, is that the generator and/or discriminator become trapped in a local minimum, impeding the necessary adjustments to mitigate mode collapse. In Fig. 3(a), the generated distribution approaches a single mode of the real distribution, which causes the optimal discriminator to have uniform predicted probabilities in this region; as a result, when the discriminator landscape is sufficiently flat in the mode neighborhood, the generator will get stuck and will not move out of the mode. We note that an extreme case of complete mode collapse is captured in Fig. 1(c) where the generator is stuck in a non-mode region. As we show in the sequel, α-loss based dual objective GANs can resolve such mode collapse issues which result from vanishing and exploding gradients.

Yet another potential cause of mode collapse is *model oscillation*. This occurs when a generator training with the non-saturating value function $V_{\mathrm{VG}}^{\mathrm{NS}}$ fails to converge due to the influence of a generated outlier data sample, as illustrated in Fig. 4. In Fig. 4(a), most of the generated data is situated at a real data mode, while some are outliers and are situated very far from the real distribution. The discriminator very confidently classifies such outlier data as fake but is less sure about the generated data that is close to the real data. As shown in Fig. 4(b), the outlier data consequently receive gradients of very large magnitude while the generated data closer to the real data receive gradients of much smaller magnitude. The generator then prioritizes directing the outlier data toward the real data over keeping the data close to the real data in place; as a result, the generator update reflects a compromise in Fig. 4(c), where the outliers are resolved at the expense of moving the other data away from the real data mode. Although the generator succeeds at bringing down the average loss by eliminating these outliers, the discriminator is now able to confidently distinguish between the distributions, leading to near-zero scores assigned to the generated data. In turn, as shown in Fig. 4(d), the generated samples all receive very large gradients which may result in oscillations around the real data. For this setting as well, in the sequel, we show that choosing value functions that allow modulating the role of the outliers such as via α-loss, can be very beneficial in addressing mode oscillation. We begin our analysis by first introducing a loss function perspective of GANs.

III. Loss Function Perspective on GANs

Noting that a GAN involves a classifier (i.e., discriminator), it is well known that the value function $V_{\mathrm{VG}}(\theta, \omega)$ in (5) considered by Goodfellow et al. [1] is related to binary cross-entropy loss. We first formalize this loss function perspective of GANs. In [19], Arora et al. observed that the log function in (5) can be replaced by any (monotonically increasing) concave function $\phi(x)$ (e.g., $\phi(x) = x$ for WGANs). In the context of using classification-based losses, we show that one can write $V(\theta, \omega)$ in terms of *any* class probability estimation (CPE) loss $\ell(y, \hat{y})$ whose inputs are the true label $y \in \{0, 1\}$ and predictor $\hat{y} \in [0, 1]$ (soft prediction of y). For a GAN, we have $(X|y = 1) \sim P_r$, $(X|y = 0) \sim P_{G_\theta}$, and $\hat{y} = D_\omega(x)$. With this, we define a value function

$$\begin{aligned} V(\theta, \omega) &= \mathbb{E}_{X|y=1}\big[-\ell(y, D_\omega(X))\big] + \mathbb{E}_{X|y=0}\big[-\ell(y, D_\omega(X))\big] \\ &= \mathbb{E}_{X\sim P_r}[-\ell(1, D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[-\ell(0, D_\omega(X))]. \end{aligned} \tag{13}$$

For binary cross-entropy loss, i.e., $\ell_{\mathrm{CE}}(y, \hat{y}) := -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$, notice that the expression in (13) is equal to V_{VG} in (5). For the value function in (13), we consider a GAN given by the min-max optimization problem:

$$\inf_{\theta\in\Theta} \sup_{\omega\in\Omega} V(\theta, \omega). \tag{14}$$

Let $\phi(\cdot) := -\ell(1, \cdot)$ and $\psi(\cdot) := -\ell(0, \cdot)$ in the sequel. The functions ϕ and ψ are assumed to be monotonically increasing and decreasing functions, respectively, so as to retain the intuitive interpretation of the vanilla GAN (that the discriminator should output high values to real samples and low values to the generated samples). These functions should also satisfy the constraint

$$\phi(t) + \psi(t) \le \phi\left(\frac{1}{2}\right) + \psi\left(\frac{1}{2}\right), \text{ for all } t \in [0, 1], \tag{15}$$

so that the optimal discriminator guesses uniformly at random (i.e., outputs a constant value $1/2$ irrespective of the input) when $P_r = P_{G_\theta}$. A loss function $\ell(y, \hat{y})$ is said to be *symmetric* [49] if $\psi(t) = \phi(1-t)$, for all $t \in [0, 1]$. Notice that the value function considered by Arora et al. [19] is a special case of (13), i.e., (13) recovers the value function in [19, Equation (2)] when the loss function $\ell(y, \hat{y})$ is symmetric. For symmetric losses, concavity of the function ϕ is a sufficient condition for satisfying (15), but not a necessary condition.

A. CPE Loss GANs and f-Divergences

We now establish a precise correspondence between the family of GANs based on CPE loss functions and a family of f-divergences. We do this by building upon a

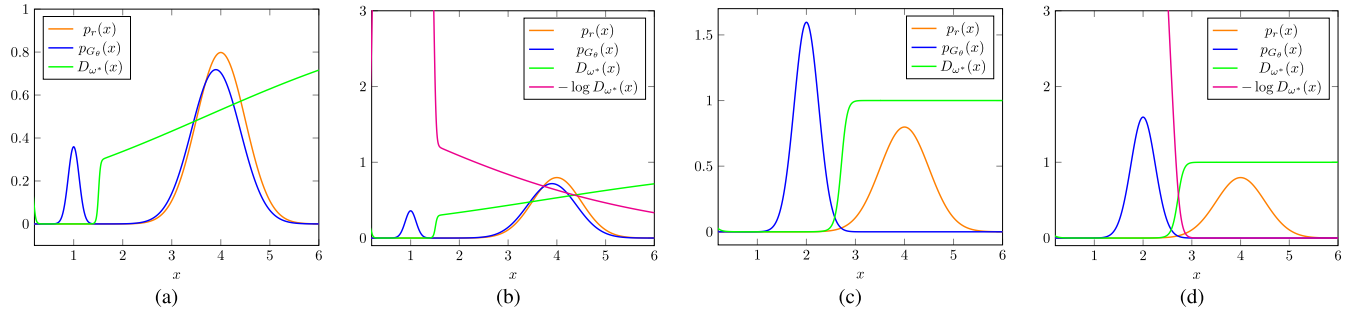


Fig. 4. A toy example of the vanilla GAN illustrating model oscillation, where the real distribution $P_r = \mathcal{N}(4, 0.5^2)$ (orange curve) and the assumed initial generated distribution $P_{G_\theta} = 0.1\mathcal{N}(1, (1/9)^2) + 0.9\mathcal{N}(3.9, 0.5^2)$ (blue curve). (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (6). The discriminator confidently classifies "outlier" generated data and gives cautious predictions for remaining data. (b) A plot of the generator's non-saturating loss $-\log(D_{\omega^*}(x))$. The outlier generated data receive very large gradients while generated data close to the real data receive relatively small gradients, which causes the generator to prioritize correcting the outlier data at the expense of preserving the proximity of the generated data close to the real data. (c) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (6) after the generator and discriminator both update. The discriminator now confidently distinguishes the generated data from the real data. (d) A plot of the generator's non-saturating loss $-\log(D_{\omega^*}(x))$ after the updates in (c). The generated samples now receive very large gradients, which may lead to oscillations around the real mode.

relationship between margin-based loss functions [50] and f-divergences first demonstrated by Nguyen et al. [43] and leveraging our CPE loss function perspective of GANs given in (13). This complements the connection established by Nowozin et al. [3] between the variational estimation approach of f-divergences [42] and f-divergence based GANs. We call a CPE loss function $\ell(y, \hat{y})$ *symmetric* [49] if $\ell(1, \hat{y}) = \ell(0, 1-\hat{y})$ and an f-divergence $D_f(\cdot\|\cdot)$ *symmetric* [51], [52] if $D_f(P\|Q) = D_f(Q\|P)$. We assume GANs with sufficiently large number of samples and ample discriminator capacity.

Theorem 1: For any symmetric CPE loss GAN with a value function in (13), the min-max optimization in (4) reduces to minimizing an f-divergence. Conversely, for any GAN designed to minimize a symmetric f-divergence, there exists a (symmetric) CPE loss GAN minimizing the same f-divergence.

Proof Sketch: Let ℓ be the symmetric CPE loss of a given CPE loss GAN; note that ℓ has a bivariate input $(y, \hat{y})$ (e.g., in (2)), where $y \in \{0, 1\}$ and $\hat{y} \in [0, 1]$. We define an associated margin-based loss function $\tilde{\ell}$ using a bijective link function (satisfying a mild regularity condition); note that a margin-based loss function has a univariate input $z \in \mathbb{R}$ (e.g., the logistic loss $\tilde{l}^{\log}(z) = \log(1 + e^{-z})$) and the bijective link function maps $z \to \hat{y}$ (see [49], [50] for more details). We show after some manipulations that the inner optimization of the CPE loss GAN reduces to an f-divergence with

$$f(u) := -\inf_{t \in \mathbb{R}} \left(\tilde{\ell}(-t) + u\tilde{\ell}(t) \right). \tag{16}$$

For the converse, given a symmetric f-divergence, using [43, Corollary 3 and Th. 1(b)], note that there exists a margin-based loss $\tilde{\ell}$ such that (16) holds. The rest of the argument follows from defining a symmetric CPE loss ℓ from this margin-based loss $\tilde{\ell}$ via the *inverse* of the same link function. See Appendix A for the detailed proof. ∎

A consequence of Theorem 1 is that it offers an interpretable way to design GANs and connect a desired measure of divergence to a corresponding loss function, where the latter is easier to implement in practice. Moreover, CPE loss based GANs inherit the intuitive and compelling interpretation of vanilla GANs that the discriminator should assign higher likelihood values to real samples and lower ones to generated samples.

We now specialize the loss function perspective of GANs to the GAN obtained by plugging in α-loss. We first write α-loss in (11) in the form of a binary classification loss to obtain

$$\ell_\alpha(y, \hat{y}) := \frac{\alpha}{\alpha - 1} \left(1 - y\hat{y}^{\frac{\alpha-1}{\alpha}} - (1-y)(1-\hat{y})^{\frac{\alpha-1}{\alpha}} \right), \tag{17}$$

for $\alpha \in (0, 1) \cup (1, \infty)$. Note that (17) recovers ℓ_{CE} as $\alpha \to 1$. Now consider a *tunable* α*-GAN* with the value function

$$\begin{aligned} V_\alpha(\theta, \omega) &= \mathbb{E}_{X \sim P_r}[-\ell_\alpha(1, D_\omega(X))] \\ &\quad + \mathbb{E}_{X \sim P_{G_\theta}}[-\ell_\alpha(0, D_\omega(X))] \\ &= \frac{\alpha}{\alpha - 1} \times \Big(\mathbb{E}_{X \sim P_r}\left[D_\omega(X)^{\frac{\alpha-1}{\alpha}} \right] \\ &\quad + \mathbb{E}_{X \sim P_{G_\theta}}\left[(1 - D_\omega(X))^{\frac{\alpha-1}{\alpha}} \right] - 2 \Big). \end{aligned} \tag{18}$$

We can verify that $\lim_{\alpha \to 1} V_\alpha(\theta, \omega) = V_{\text{VG}}(\theta, \omega)$, recovering the value function of the vanilla GAN. Also, notice that

$$\lim_{\alpha \to \infty} V_\alpha(\theta, \omega) = \mathbb{E}_{X \sim P_r}[D_\omega(x)] - \mathbb{E}_{X \sim P_{G_\theta}}[D_\omega(x)] - 1 \tag{19}$$

is the value function (modulo a constant) used in IPM based GANs,[4] e.g., WGAN, McGan [27], Fisher GAN [26], and Sobolev GAN [34]. The resulting min-max game in α-GAN is given by

$$\inf_{\theta \in \Theta} \sup_{\omega \in \Omega} V_\alpha(\theta, \omega). \tag{20}$$

The following theorem provides the min-max solution, i.e., Nash equilibrium, to the two-player game in (20) for the non-parametric setting, i.e., when the discriminator set Ω is large enough.

Theorem 2: For $\alpha \in (0, 1) \cup (1, \infty)$ and a generator G_θ, the discriminator D_{ω^*} optimizing the sup in (20) is

$$D_{\omega^*}(x) = \frac{p_r(x)^\alpha}{p_r(x)^\alpha + p_{G_\theta}(x)^\alpha}, \quad x \in \mathcal{X}, \tag{21}$$

[4]Note that IPMs do not restrict the function D_ω to map to $[0, 1]$.

where p_r and p_{G_θ} are the corresponding densities of the distributions P_r and P_{G_θ}, respectively, with respect to a base measure dx (e.g., Lebesgue measure). For this D_{ω^*}, (20) simplifies to minimizing a non-negative symmetric f_α-divergence $D_{f_\alpha}(\cdot||\cdot)$ to obtain

$$\inf_{\theta\in\Theta} D_{f_\alpha}(P_r||P_{G_\theta}) + \frac{\alpha}{\alpha-1}\left(2^{\frac{1}{\alpha}}-2\right), \tag{22}$$

where

$$f_\alpha(u) = \frac{\alpha}{\alpha-1}\left(\left(1+u^\alpha\right)^{\frac{1}{\alpha}} - (1+u) - 2^{\frac{1}{\alpha}} + 2\right), \tag{23}$$

for $u \geq 0$ and[5]

$$D_{f_\alpha}(P||Q) = \frac{\alpha}{\alpha-1}\left(\int_{\mathcal{X}} \left(p(x)^\alpha + q(x)^\alpha\right)^{\frac{1}{\alpha}} dx - 2^{\frac{1}{\alpha}}\right), \tag{24}$$

which is minimized iff $P_{G_\theta} = P_r$.

A detailed proof of Theorem 2 is in Appendix B.

Remark 1: As $\alpha \to 0$, note that (21) implies a more cautious discriminator, i.e., if $p_{G_\theta}(x) \geq p_r(x)$, then $D_{\omega^*}(x)$ decays more slowly from 1/2, and if $p_{G_\theta}(x) \leq p_r(x)$, $D_{\omega^*}(x)$ increases more slowly from 1/2. Conversely, as $\alpha \to \infty$, (21) simplifies to $D_{\omega^*}(x) = \mathbb{1}\{p_r(x) > p_{G_\theta}(x)\} + \frac{1}{2}\mathbb{1}\{p_r(x) = p_{G_\theta}(x)\}$, where the discriminator implements the Maximum Likelihood (ML) decision rule, i.e., a hard decision whenever $p_r(x) \neq p_{G_\theta}(x)$. In other words, (21) for $\alpha \to \infty$ induces a very confident discriminator. Regarding the generator's perspective, (22) implies that the generator seeks to minimize the discrepancy between P_r and P_{G_θ} according to the geometry induced by D_{f_α}. Thus, the optimization trajectory traversed by the generator during training is strongly dependent on the practitioner's choice of $\alpha \in (0, \infty)$. Please refer to Fig. 11 in Appendix C for an illustration of this observation. Figure 5 illustrates this effect of tuning α on the optimal D and the corresponding loss of the generator for a toy example.

It is worth noting that the divergence $D_{f_\alpha}(\cdot||\cdot)$ in (24) that naturally emerges from the analysis of α-GAN was first proposed by Österriecher [10] in a statistical context of measures and was later referred to as the *Arimoto divergence* by Liese and Vajda [11]. Next, we show that α-GAN recovers various well known f-GANs.

Theorem 3: α-GAN recovers vanilla GAN, Hellinger GAN (H-GAN) [3], and Total Variation GAN (TV-GAN) [3] as $\alpha \to 1$, $\alpha = \frac{1}{2}$, and $\alpha \to \infty$, respectively.

Proof Sketch: We show the following: (i) as $\alpha \to 1$, (22) equals $\inf_{\theta\in\Theta} 2D_{\mathrm{JS}}(P_r||P_{G_\theta}) - \log 4$ recovering the vanilla GAN; (ii) for $\alpha = \frac{1}{2}$, (22) gives $2\inf_{\theta\in\Theta} D_{\mathrm{H}^2}(P_r||P_{G_\theta}) - 2$ recovering Hellinger GAN (up to a constant); and (iii) as $\alpha \to \infty$, (22) equals $\inf_{\theta\in\Theta} D_{\mathrm{TV}}(P_r||P_{G_\theta}) - 1$ recovering TV-GAN (modulo a constant). A detailed proof is in Appendix C. ■

Next, we present an equivalence between f_α-GAN defined using the value function in (8) and α-GAN. Let $\overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$ denote the set of extended real numbers. Two optimization problems $\sup_{v\in A} g(v)$ and $\sup_{t\in B} h(t)$ are said to be equivalent [54], [55] if there exists a bijective function $k : A \to B$ such that

$$g(v) = h(k(v)) \text{ and } h(t) = g(k^{-1}(t)), \text{ for all } v \in A, t \in B. \tag{25}$$

In other words, two optimization problems are equivalent if a change of variable via the function k can transform one into the other.

Theorem 4: For any $\alpha \in (0, 1)\cup(1, \infty)$, let $\tilde{f}_\alpha$ be a slightly modified version of (23) defined as

$$\tilde{f}_\alpha(u) = \frac{\alpha}{\alpha-1}\left(\left(1+u^\alpha\right)^{\frac{1}{\alpha}} - (1+u)\right), \quad u \geq 0, \tag{26}$$

with continuous extensions at $\alpha = 1$ and $\alpha = \infty$. Let $\tilde{f}_\alpha^*$ be the convex conjugate of $\tilde{f}_\alpha$ given by

$$\tilde{f}_\alpha^*(t) = \frac{\alpha}{\alpha-1}\left(1 - (1-s(t))^{\frac{\alpha-1}{\alpha}}\right), \tag{27}$$

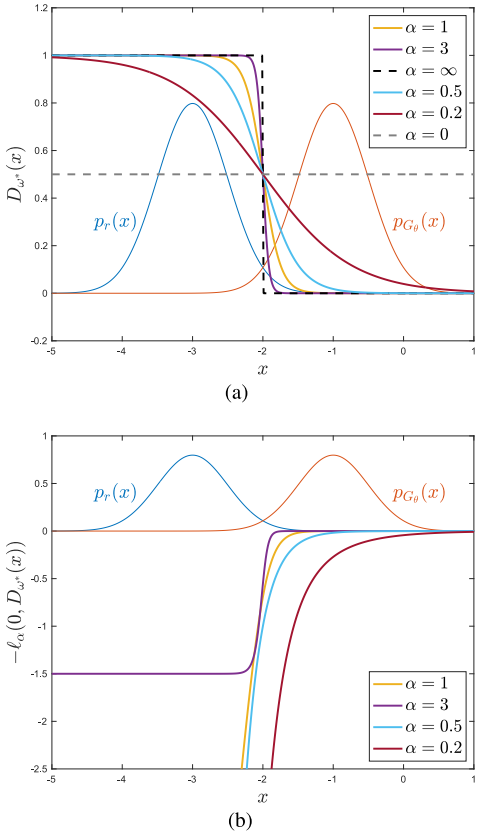


Fig. 5. A toy example of α-GAN, where $P_r = \mathcal{N}(-2, 0.5^2)$ (blue curve) and the assumed initial $P_{G_\theta} = \mathcal{N}(2, 0.5^2)$ (orange curve). (a) A plot of the optimal $D_{\omega^*}(x)$ for $\alpha \in \{0, 0.2, 0.5, 1, 3, \infty\}$. As α decreases, D_{ω^*} becomes increasingly less confident in its predictions until it outputs 1/2 for all x when $\alpha \to 0$. Conversely, as α increases, D_{ω^*} becomes increasingly more confident until it implements the Maximum Likelihood decision rule when $\alpha \to \infty$. (b) A plot of the generator's corresponding loss $-\ell_\alpha(0, D_{\omega^*}(x))$ for $\alpha \in \{0.2, 0.5, 1, 3\}$. As α decreases, the magnitude of the gradients of the loss increases, while increasing α saturates the gradients. Early in training, if the discriminator is very confident and outputs values close to 0 for the generated data, the generator will not have much gradient to continue learning, resulting in vanishing gradients. Decreasing α reduces the discriminator's confidence and provides more gradient for the generator to learn.

[5]We note that the divergence D_{f_α} has been referred to as *Arimoto divergence* in the literature [10], [11], [53].

where

$$s(t) = \left(1 + \frac{\alpha - 1}{\alpha} t\right)^{\frac{\alpha}{\alpha - 1}}. \tag{28}$$

Let $g_{f_\alpha} : \overline{\mathbb{R}} \to \text{dom}(\tilde{f}_\alpha^*)$ be a bijective output activation function. Then the optimization problems involved in $\tilde{f}_\alpha$-GAN (using (8) with $f = \tilde{f}_\alpha$) and α-GAN (using (18)) are equivalent in the sense of (25) for

$$g(Q_w) = \mathbb{E}_{X \sim P_r}\Big[g_{f_\alpha}\big(Q_\omega(X)\big)\Big] + \mathbb{E}_{X \sim P_{G_\theta}}\Big[-\tilde{f}_\alpha^*\big(g_{f_\alpha}(Q_\omega(X))\big)\Big] \tag{29}$$

with $A = \{Q_\omega : \mathcal{X} \to \overline{\mathbb{R}}\}$ and

$$h(D_\omega) = \mathbb{E}_{X \sim P_r}\Big[-\ell_\alpha\big(1, D_\omega(X)\big)\Big] + \mathbb{E}_{X \sim P_{G_\theta}}\Big[-\ell_\alpha\big(0, D_\omega(X)\big)\Big] \tag{30}$$

with $B = \{D_\omega : \mathcal{X} \to [0, 1]\}$ when $k : A \to B$ is chosen as

$$k(v) = s(g_{f_\alpha}(v)) = \left(1 + \left(\frac{\alpha - 1}{\alpha}\right) g_{f_\alpha}(v)\right)^{\frac{\alpha}{\alpha - 1}}. \tag{31}$$

Proof Sketch: The proof relies on a lemma that is proved in the appendix showing that there exists a mapping between the terms involved in the optimization of both GAN formulations. Proof details are in Appendix D. ∎

The following theorem generalizes the equivalence demonstrated above between $\tilde{f}_\alpha$-GAN and α-GAN to an equivalence between f-GANs (using the original value function in (7)) and CPE loss based GANs.

Theorem 5: For any given symmetric f-divergence, the optimization problems involved in f-GAN and the CPE loss based GAN minimizing the same f-divergence are equivalent under the following regularity conditions on f:

- there exists a strictly convex and differentiable CPE (partial) loss function ℓ such that

$$f(u) = \sup_{t \in [0,1]} -u\ell(t) - \ell(1 - t) \tag{32}$$

  (note that this condition without the requirement of strict convexity of ℓ is indeed guaranteed by [42, Th. 2] for any convex function f resulting in a symmetric divergence) and $-u\ell(t) - \ell(1-t)$ has a local maximum in t for every $u \in \mathbb{R}_+$, and
- the function mapping $u \in \mathbb{R}_+$ to unique optimizer in (32) is bijective.

Proof sketch: Observing that the inner optimization problem in the CPE loss GAN formulation reduces to the pointwise optimization (32) and that of the f-GAN formulation reduces to the pointwise optimization

$$f(u) = \sup_{v \in \text{dom} f^*} uv - f^*(v), \tag{33}$$

it suffices to show that the variational forms of f in (32) and (33) are equivalent. We do this by showing that (32) is equivalent to the optimization problem

$$f(u) = \sup_{v \in \mathbb{R}_+} uf'(v) - \big[vf'(v) - f(v)\big], \tag{34}$$

which has been shown to be equivalent to (33) [56]. A detailed proof is in Appendix E. ∎

Remark 2: Since α-loss, $\ell_\alpha(p) = \frac{\alpha}{\alpha - 1}(1 - p^{\frac{\alpha - 1}{\alpha}})$, $p \in [0, 1]$, is strictly convex for $\alpha \in (0, \infty)$, and the function mapping $u \in \mathbb{R}_+$ to unique optimizer in (32) with α-loss, i.e., $\frac{u^\alpha}{1 + u^\alpha}$, is bijective, Theorem 5 implies that α-GAN is equivalent to $\tilde{f}_\alpha$-GAN with $\tilde{f}_\alpha$ defined in (26).

Remark 3: Though the CPE loss GAN and f-GAN formulations are equivalent, the following aspects differentiate the two:

- The f-GAN formulation focuses on the generator minimizing an f-divergence with no explicit emphasis on the role of the discriminator as a binary classifier in relation to the function f. With the CPE loss GAN formulation, we bring into the foreground the connection between the binary classification performed by the discriminator and the f-divergence minimization done by the generator.
- More importantly, the CPE loss function perspective of GANs allows us to prove convergence properties (Theorem 6), generalization error bounds (Theorem 7), and estimation error bounds (Theorem 8) as detailed in the following sections.

B. Convergence Guarantees for CPE Loss GANs

Building on the above one-to-one correspondence, we now present *convergence* results for CPE loss GANs, including α-GAN, thereby providing a unified perspective on the convergence of a variety of f-divergences that arise when optimizing GANs. Here again, we assume a sufficiently large number of samples and ample discriminator capacity. In [57], Liu et al. address the following question in the context of convergence analysis of any GAN: For a sequence of generated distributions (P_n), does convergence of a divergence between the generated distribution P_n and a fixed real distribution P to the global minimum lead to some standard notion of distributional convergence of P_n to P? They answer this question in the affirmative provided the sample space $\mathcal{X}$ is a compact metric space.

Liu et al. [57] formally define any divergence that results from the inner optimization of a general GAN in (4) as an *adversarial divergence* [57, Definition 1], thus broadly capturing the divergences used by a number of existing GANs, including vanilla GAN [1], f-GAN [3], WGAN [4], and MMD-GAN [32]. Indeed, the divergence that results from the inner optimization of a CPE loss GAN (including α-GAN) in (14) is also an adversarial divergence. For *strict adversarial divergences* (a subclass of the adversarial divergences where the minimizer of the divergence is uniquely the real distribution), Liu et al. [57] show that convergence of the divergence to its global minimum implies weak convergence of the generated distribution to the real distribution. Interestingly, this also leads to a structural result on the class of strict adversarial divergences [57, Figure 1 and Corollary 12] based on a notion of *relative strength* between adversarial divergences. We note that the Arimoto divergence D_{f_α} in (24) is a strict adversarial divergence. We briefly summarize the following terminology from Liu et al. [57] to present our results on convergence

properties of CPE loss GANs. Let $\mathcal{P}(\mathcal{X})$ be the probability simplex of distributions over $\mathcal{X}$.

Definition 2 ([57, Definition 11]): A strict adversarial divergence τ_1 is said to be stronger than another strict adversarial divergence τ_2 (or τ_2 is said to be weaker than τ_1) if for any sequence of probability distributions (P_n) and target distribution P (both in $\mathcal{P}(\mathcal{X})$), $\tau_1(P\|P_n) \to 0$ as $n \to \infty$ implies $\tau_2(P\|P_n) \to 0$ as $n \to \infty$. We say τ_1 is equivalent to τ_2 if τ_1 is both stronger and weaker than τ_2.

Arjovsky et al. [4] proved that the Jensen-Shannon divergence (JSD) is equivalent to the total variation distance (TVD). Later, Liu et al. showed that the squared Hellinger distance is equivalent to both of these divergences, meaning that all three divergences belong to the same equivalence class (see [57, Fig. 1]). Noticing that the squared Hellinger distance, JSD, and TVD correspond to Arimoto divergences $D_{f_\alpha}(\cdot||\cdot)$ for $\alpha = 1/2$, $\alpha = 1$, and $\alpha = \infty$, respectively, it is natural to ask the question: Are Arimoto divergences for all $\alpha > 0$ equivalent? We answer this question in the affirmative in Theorem 6. In fact, we prove that all symmetric f-divergences, including D_{f_α}, are equivalent in convergence.

Theorem 6: Let $f_i : [0, \infty) \to \mathbb{R}$ be a convex function which is continuous at 0 and strictly convex at 1 such that $f_i(1) = 0$, $uf_i(\frac{1}{u}) = f_i(u)$, and $f_i(0) < \infty$, for $i \in \{1, 2\}$. Then for a sequence of probability distributions $(P_n)_{n\in\mathbb{N}} \in \mathcal{P}(\mathcal{X})$ and a fixed distribution $P \in \mathcal{P}(\mathcal{X})$, we have $D_{f_1}(P_n||P) \to 0$ as $n \to \infty$ if and only if $D_{f_2}(P_n||P) \to 0$ as $n \to \infty$.

Proof sketch: Note that it suffices to show that $D_f(\cdot\|\cdot)$ is equivalent to $D_{\mathrm{TV}}(\cdot\|\cdot)$ for any function f satisfying the conditions in the theorem. To show this, we employ an elegant result by Feldman and Österreicher [58, Corollary 1] which gives lower and upper bounds on a symmetric f-divergence, with the function f satisfying the conditions in the theorem statement, in terms of TVD as

$$\gamma_f(D_{\mathrm{TV}}(P||Q)) \le D_f(P||Q) \le \gamma_f(1)D_{\mathrm{TV}}(P||Q), \tag{35}$$

for an appropriately defined well-behaved (continuous, invertible, and bounded) function $\gamma_f : [0, 1] \to [0, \infty)$. We use the lower and upper bounds in (35) to show that $D_f(\cdot\|\cdot)$ is stronger than $D_{\mathrm{TV}}(\cdot\|\cdot)$, and $D_f(\cdot\|\cdot)$ is weaker than $D_{\mathrm{TV}}(\cdot\|\cdot)$, respectively. Proof details are in Appendix F. ■

Remark 4: We note that the proof techniques used in proving Theorem 6 give rise to a conceptually simpler proof of equivalence between JSD ($\alpha = 1$) and TVD ($\alpha = \infty$) proved earlier by Arjovsky et al. [4, Th. 2(1)], where measure-theoretic analysis was used. In particular, our proof of equivalence relies on the fact that TVD upper bounds JSD [2, Th. 3]. See Appendix G.

Theorems 1 through 6 hold in the ideal setting of sufficient samples and discriminator capacity. In practice, however, GAN training is limited by both the number of training samples as well as the choice of G_θ and D_ω. In fact, recent results by Arora et al. [19] show that under such limitations, convergence in divergence does not imply convergence in distribution, and have led to new metrics for evaluating GANs. To address these limitations, we consider two measures to evaluate the performance of GANs, namely generation and estimation errors, as detailed below.

C. Generalization and Estimation Error Bounds for CPE Loss GANs

Arora et al. [19] defined *generalization* in GANs as the scenario when the divergence between the real distribution and the generated distribution is well-captured by the divergence between their empirical versions. In particular, a divergence or distance[6] $d(\cdot, \cdot)$ between distributions *generalizes* with m training samples and error $\epsilon > 0$ if, for the learned distribution P_{G_θ}, the following holds with high probability:

$$\left|d\big(P_r, P_{G_\theta}\big) - d\Big(\hat{P}_r, \hat{P}_{G_\theta}\Big)\right| \le \epsilon, \tag{36}$$

where $\hat{P}_r$ and $\hat{P}_{G_\theta}$ are the empirical versions of the real (with m samples) and the generated (with a polynomial number of samples) distributions, respectively. Arora et al. [19, Lemma 1] show that the Jensen-Shannon divergence and Wasserstein distance do not generalize with any polynomial number of samples. However, they show that generalization can be achieved for a new notion of divergence, the *neural net divergence*, with a moderate number of training examples [19, Th. 3.1]. To this end, they consider the following optimization problem

$$\inf_{\theta\in\Theta} d_{\mathcal{F}}\big(P_r, P_{G_\theta}\big), \tag{37}$$

where $d_{\mathcal{F}}(P_r, P_{G_\theta})$ is the neural net divergence defined as

$$\begin{aligned} &d_{\mathcal{F}}\big(P_r, P_{G_\theta}\big) \\ &\quad= \sup_{\omega\in\Omega}\Big(\mathbb{E}_{X\sim P_r}[\phi(D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[\phi(1 - D_\omega(X))]\Big) \\ &\qquad- 2\phi(1/2) \end{aligned} \tag{38}$$

such that the class of discriminators $\mathcal{F} = \{D_\omega{:}\omega \in \Omega\}$ is L-Lipschitz with respect to the parameters ω, i.e., there exists a constant $L \ge 1$ such that for every $x \in \mathcal{X}$, $|D_{\omega_1}(x) - D_{\omega_2}(x)| \le L||\omega_1 - \omega_2||$, for all $\omega_1, \omega_2 \in \Omega$, and the function ϕ takes values in $[-\Delta, \Delta]$ and is L_ϕ-Lipschitz. Let p be the discriminator capacity (i.e., number of parameters) and $\epsilon > 0$. For these assumptions, in [19, Th. 3.1], Arora et al. prove that (39) generalizes. We summarize their result as follows: for the empirical versions $\hat{P}_r$ and $\hat{P}_{G_\theta}$ of two distributions P_r and P_{G_θ}, respectively, with at least m random samples each, there exists a universal constant c such that when $m \ge \frac{cp\Delta^2 \log(LL_\phi p/\epsilon)}{\epsilon^2}$, with probability at least $1 - \exp(-p)$ (over the randomness of samples),

$$\left|d_{\mathcal{F}}\big(P_r, P_{G_\theta}\big) - d_{\mathcal{F}}\Big(\hat{P}_r, \hat{P}_{G_\theta}\Big)\right| \le \epsilon. \tag{39}$$

Our first contribution is to show that we can generalize (39) and [19, Th. 3.1] to incorporate any partial losses ϕ and ψ (not just those that are symmetric). To this end, we first define the *refined neural net divergence* as

$$\begin{aligned} &\tilde{d}_{\mathcal{F}}\big(P_r, P_{G_\theta}\big) \\ &\quad= \sup_{\omega\in\Omega}\Big(\mathbb{E}_{X\sim P_r}[\phi(D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[\psi(D_\omega(X))]\Big) \\ &\qquad- \phi(1/2) - \psi(1/2), \end{aligned} \tag{40}$$

[6]For consistency with other works on generalization and estimation error, we refer to a semi-metric as a distance.

where the discriminator class is same as the above and the functions ϕ and ψ take values in $[-\Delta, \Delta]$ and are L_ϕ- and L_ψ-Lipschitz, respectively. Note that the functions ϕ and ψ should also satisfy (15) so as to respect the optimality of the uniformly random discriminator when $P_r = P_{G_\theta}$. The following theorem shows that the refined neural net divergence generalizes with a moderate number of training examples, thus extending [19, Th. 3.1].

Theorem 7: Let $\hat{P}_r$ and $\hat{P}_{G_\theta}$ be empirical versions of two distributions P_r and P_{G_θ}, respectively, with at least m random samples each. For $\Delta, p, L, L_\phi, L_\psi, \epsilon > 0$ defined above, there exists a universal constant c such that when $m \geq \frac{cp\Delta^2 \log(L\max\{L_\phi, L_\psi\}p/\epsilon)}{\epsilon^2}$, we have that with probability at least $1 - \exp(-p)$ (over the randomness of samples),

$$\left|\tilde{d}_{\mathcal{F}}(P_r, P_{G_\theta}) - \tilde{d}_{\mathcal{F}}\left(\hat{P}_r, \hat{P}_{G_\theta}\right)\right| \leq \epsilon. \tag{41}$$

The proof of Theorem 7 is provided in Appendix H. When $\phi(t) = t$ and $D_\omega = f_\omega$ can take values in $\mathbb{R}$ (not just in $[0, 1]$), (39) yields the so-called *neural net* (*nn*) *distance*[7] [19], [41], [59] given by

$$d_{\mathcal{F}_{nn}}(P_r, P_{G_\theta}) = \sup_{\omega\in\Omega}\left(\mathbb{E}_{X\sim P_r}[f_\omega(X)] - \mathbb{E}_{X\sim P_{G_\theta}}[f_\omega(X)]\right), \tag{42}$$

where the discriminator[8] and generator $f_\omega(\cdot)$ and $G_\theta(\cdot)$, respectively, are neural networks. Using (42), Ji et al. [41] defined and studied the notion of *estimation error*, which quantifies the effectiveness of the generator (for a corresponding optimal discriminator model) in learning the real distribution with limited samples. In order to define estimation error for CPE-loss GANs (including α-GAN), we first introduce a *loss-inclusive neural net divergence*[9] $d^{(\ell)}_{\mathcal{F}_{nn}}$ to highlight the effect of the *loss* on the error. For training samples $S_x = \{X_1, \ldots, X_n\}$ and $S_z = \{Z_1, \ldots, Z_m\}$ from P_r and P_Z, respectively, we begin with the following minimization for GAN training:

$$\inf_{\theta\in\Theta} d^{(\ell)}_{\mathcal{F}_{nn}}\left(\hat{P}_r, \hat{P}_{G_\theta}\right), \tag{43}$$

where $\hat{P}_r$ and $\hat{P}_{G_\theta}$ are the empirical real and generated distributions estimated from S_x and S_z, respectively, and

$$\begin{aligned} &d^{(\ell)}_{\mathcal{F}_{nn}}\left(\hat{P}_r, \hat{P}_{G_\theta}\right) \\ &\quad= \sup_{\omega\in\Omega}\left(\mathbb{E}_{X\sim\hat{P}_r}[\phi(D_\omega(X))] + \mathbb{E}_{X\sim\hat{P}_{G_\theta}}[\psi(D_\omega(X))]\right) \\ &\qquad - \phi(1/2) - \psi(1/2), \end{aligned} \tag{44}$$

where for brevity we henceforth use $\phi(\cdot) := -\ell(1, \cdot)$ and $\psi(\cdot) := -\ell(0, \cdot)$. As proven in Theorem 3, for $\ell = \ell_\alpha$ and $\alpha = \infty$, (45) reduces to the neural net total variation distance.

As a step towards obtaining bounds on the estimation error, we consider the following setup, analogous to that in [41]. For $x \in \mathcal{X} := \{x \in \mathbb{R}^d : ||x||_2 \leq B_x\}$ and $z \in \mathcal{Z} := \{z \in \mathbb{R}^p : ||z||_2 \leq B_z\}$, we consider discriminators and generators as neural network models of the form:

$$D_\omega : x \mapsto \sigma\left(\mathbf{w}_k^{\mathsf{T}} r_{k-1}(\mathbf{W}_{k-1} r_{k-2}(\ldots r_1(\mathbf{W}_1 x)))\right) \tag{45}$$

$$G_\theta : z \mapsto \mathbf{V}_l s_{l-1}(\mathbf{V}_{l-1} s_{l-2}(\ldots s_1(\mathbf{V}_1 z))), \tag{46}$$

where $\mathbf{w}_k$ is a parameter vector of the output layer; for $i \in [1 : k-1]$ and $j \in [1 : l]$, $\mathbf{W}_i$ and $\mathbf{V}_j$ are parameter matrices; $r_i(\cdot)$ and $s_j(\cdot)$ are entry-wise activation functions of layers i and j, i.e., for $\mathbf{a} \in \mathbb{R}^t$, $r_i(\mathbf{a}) = [r_i(a_1), \ldots, r_i(a_t)]$ and $s_i(\mathbf{a}) = [s_i(a_1), \ldots, s_i(a_t)]$; and $\sigma(\cdot)$ is the sigmoid function given by $\sigma(p) = 1/(1 + \mathrm{e}^{-p})$ (note that σ does not appear in the discriminator in [41, eq. (7)] as the discriminator considered in the neural net distance is not a soft classifier mapping to [0, 1]). We assume that each $r_i(\cdot)$ and $s_j(\cdot)$ are R_i- and S_j-Lipschitz, respectively, and also that they are positive homogeneous, i.e., $r_i(\lambda p) = \lambda r_i(p)$ and $s_j(\lambda p) = \lambda s_j(p)$, for any $\lambda \geq 0$ and $p \in \mathbb{R}$. Finally, as modelled in [41], [60], [61], [62], we assume that the Frobenius norms of the parameter matrices are bounded, i.e., $||\mathbf{W}_i||_F \leq M_i$, $i \in [1{:}k-1]$, $||\mathbf{w}_k||_2 \leq M_k$, and $||\mathbf{V}_j||_F \leq N_j$, $j \in [1{:}l]$.

We define the estimation error for a CPE loss GAN as

$$d^{(\ell)}_{\mathcal{F}_{nn}}\left(P_r, P_{G_{\hat{\theta}^*}}\right) - \inf_{\theta\in\Theta} d^{(\ell)}_{\mathcal{F}_{nn}}(P_r, P_{G_\theta}), \tag{47}$$

where $\hat{\theta}^*$ is the minimizer of (43) and present the following upper bound on the error. We also specialize these bounds for α-GANs, relying on the Rademacher complexity of this loss class to do so.

Theorem 8: For the setting described above, additionally assume that the functions $\phi(\cdot)$ and $\psi(\cdot)$ are L_ϕ- and L_ψ-Lipschitz, respectively. Then, with probability at least $1 - 2\delta$ over the randomness of training samples $S_x = \{X_i\}_{i=1}^n$ and $S_z = \{Z_j\}_{j=1}^m$, we have

$$\begin{aligned} &d^{(\ell)}_{\mathcal{F}_{nn}}\left(P_r, P_{G_{\hat{\theta}^*}}\right) - \inf_{\theta\in\Theta} d^{(\ell)}_{\mathcal{F}_{nn}}(P_r, P_{G_\theta}) \\ &\quad\leq \frac{L_\phi B_x U_\omega \sqrt{3k}}{\sqrt{n}} + \frac{L_\psi U_\omega U_\theta B_z \sqrt{3(k+l-1)}}{\sqrt{m}} \\ &\qquad + U_\omega \sqrt{\log\frac{1}{\delta}}\left(\frac{L_\phi B_x}{\sqrt{2n}} + \frac{L_\psi B_z U_\theta}{\sqrt{2m}}\right), \end{aligned} \tag{48}$$

where $U_\omega := M_k \prod_{i=1}^{k-1}(M_i R_i)$ and $U_\theta := N_l \prod_{j=1}^{l-1}(N_j S_j)$.

In particular, when this bound is specialized to the case of α-GAN by letting $\phi(p) = \psi(1-p) = \frac{\alpha}{\alpha-1}(1 - p^{\frac{\alpha-1}{\alpha}})$, the resulting bound is nearly identical to the terms in the RHS of (48), except for substitutions $L_\phi \leftarrow 4C_{Q_x}(\alpha)$ and $L_\psi \leftarrow 4C_{Q_z}(\alpha)$, where $Q_x := U_\omega B_x$, $Q_z := U_\omega U_\theta B_z$, and

$$C_h(\alpha) := \begin{cases} \sigma(h)\sigma(-h)^{\frac{\alpha-1}{\alpha}}, & \alpha \in (0, 1] \\ \left(\frac{\alpha-1}{2\alpha-1}\right)^{\frac{\alpha-1}{\alpha}} \frac{\alpha}{2\alpha-1}, & \alpha \in (1, \infty). \end{cases} \tag{49}$$

Proof sketch: Our proof involves the following steps:

- Building upon the proof techniques of Ji et al. [41, Th. 1], we bound the estimation error in terms of Rademacher complexities of *compositional* function classes involving the CPE loss function.

[7] This term was first introduced in [19] but with a focus on a discriminator D_ω taking values in [0, 1]. Ji et al. [41], [59] generalized it to $D_\omega = f_\omega$ taking values in $\mathbb{R}$.

[8] In [41], f_ω indicates a discriminator function that takes values in $\mathbb{R}$.

[9] We refer to this measure as a divergence since it may not be a semi-metric for all choices of the loss ℓ.

- We then upper bound these Rademacher complexities leveraging a contraction lemma for Lipschitz loss functions [63, Lemma 26.9]. We remark that this differs considerably from the way the bounds on Rademacher complexities in [41, Corollary 1] are obtained because of the explicit role of the loss function in our setting.
- For the case of α-GAN, we extend a result by Sypherd et al. [9] where they showed that α-loss is Lipschitz for a logistic model with (79). Noting that similar to the logistic model, we also have a sigmoid in the outer layer of the discriminator, we generalize the preceding observation by proving that α-loss is Lipschitz when the input is equal to a sigmoid function acting on a *neural network* model. This is the reason behind the dependence of the Lipschitz constant on the neural network model parameters (in terms of Q_x and Q_z). Note that (79) is monotonically decreasing in α, indicating the bound saturates. However, one is not able to make definitive statements regarding the estimation bounds for relative values of α because the LHS in (48) is *also* a function of α. Proof details are in Appendix I. ■

We now focus on developing lower bounds on the estimation error. Due to the fact that oft-used techniques to obtain min-max lower bounds on the quality of an estimator (e.g., LeCam's methods, Fano's methods, etc.) require a semi-metric distance measure, we restrict our attention to a particular α-GAN, namely that for $\alpha = \infty$, to derive a matching lower bound on the estimation error. We consider the loss-inclusive neural net divergence in (44) with $\ell = \ell_\alpha$ for $\alpha = \infty$, which, for brevity, we henceforth denote as $d^{\ell_\infty}_{\mathcal{F}_{nn}}(\cdot,\cdot)$ As in [41], suppose the generator's class $\{G_\theta\}_{\theta\in\Theta}$ is rich enough such that the generator G_θ can learn the real distribution P_r and that the number m of training samples in S_z scales faster than the number n of samples in S_x.[10] Then $\inf_{\theta\in\Theta} d^{\ell_\infty}_{\mathcal{F}_{nn}}(P_r, P_{G_\theta}) = 0$, so the estimation error simplifies to the single term $d^{\ell_\infty}_{\mathcal{F}_{nn}}(P_r, P_{G_{\hat{\theta}^*}})$. Furthermore, the upper bound in (48) reduces to $O(c/\sqrt{n})$ for some constant c (note that, in (49), $C_h(\infty) = 1/4$). In addition to the above assumptions, also assume the activation functions r_i for $i \in [1{:}k-1]$ are either strictly increasing or ReLU. For the above setting, we derive a matching min-max lower bound (up to a constant multiple) on the estimation error.

Theorem 9: For the setting above, let $\hat{P}_n$ be an estimator of P_r learned using the training samples $S_x = \{X_i\}_{i=1}^n$. Then,

$$\inf_{\hat{P}_n}\sup_{P_r\in\mathcal{P}(\mathcal{X})} \mathbb{P}\left\{ d^{\ell_\infty}_{\mathcal{F}_{nn}}\left(\hat{P}_n, P_r\right) \geq \frac{C(\mathcal{P}(\mathcal{X}))}{\sqrt{n}}\right\} > 0.24,$$

where the constant $C(\mathcal{P}(\mathcal{X}))$ is given by

$$C(\mathcal{P}(\mathcal{X})) = \frac{\log(2)}{20}\Big[\sigma(M_k r_{k-1}(\dots r_1(M_1 B_x)) - \sigma(M_k r_{k-1}(\dots r_1(-M_1 B_x))\Big]. \tag{50}$$

Proof sketch: To obtain min-max lower bounds, we first prove that $d^{\ell_\infty}_{\mathcal{F}_{nn}}$ is a semi-metric. The remainder of the proof is similar to that of [41, Th. 2], replacing $d_{\mathcal{F}_{nn}}$ with $d^{\ell_\infty}_{\mathcal{F}_{nn}}$. Finally, we note that the additional sigmoid activation function after the last layer in D satisfies the monotonicity assumption as detailed in Appendix J. A challenge that remains to be addressed is to verify if $d^{\ell_\alpha}_{\mathcal{F}_{nn}}$ is a semi-metric for $\alpha < \infty$. ■

[10] Since the noise distribution P_Z is known, one can generate an arbitrarily large number m of noise samples.

IV. Dual-Objective GANs

As illustrated in Fig. 5, tuning $\alpha < 1$ provides more gradient for the generator to learn early in training when the discriminator more confidently classifies the generated data as fake, alleviating vanishing gradients, and also creates a smooth landscape for the generated data to descend towards the real data, alleviating exploding gradients. However, tuning $\alpha < 1$ may provide too large of gradients for the generator when the generated samples approach the real samples, which can result in too much movement of the generated data, potentially repelling it from the real data. The following question therefore arises: Can we combine a less confident discriminator with a more stable generator loss? We show that we can do so by using different objectives for the discriminator and generator, resulting in (α_D, α_G)-GANs.

A. (α_D, α_G)-GANs

We propose a dual-objective (α_D, α_G)-GAN with different objective functions for the generator and discriminator in which the discriminator maximizes $V_{\alpha_D}(\theta,\omega)$ while the generator minimizes $V_{\alpha_G}(\theta,\omega)$, where

$$V_\alpha(\theta,\omega) = \mathbb{E}_{X\sim P_r}[-\ell_\alpha(1, D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[-\ell_\alpha(0, D_\omega(X))], \tag{51}$$

for $\alpha = \alpha_D, \alpha_G \in (0,\infty]$ with $\ell_\alpha(\cdot,\cdot)$ given in (17). We recover the α-GAN [7], [12] value function when $\alpha_D = \alpha_G = \alpha$. The resulting (α_D, α_G)-GAN is given by

$$\sup_{\omega\in\Omega} V_{\alpha_D}(\theta,\omega) \tag{52a}$$

$$\inf_{\theta\in\Theta} V_{\alpha_G}(\theta,\omega). \tag{52b}$$

We maintain the same ordering as the original min-max GAN formulation for this non-zero sum game, wherein for a set of chosen parameters for both players, the discriminator plays first, followed by the generator. The following theorem presents the conditions under which the optimal generator learns the real distribution P_r when the discriminator set Ω is large enough.

Theorem 10: For the game in (52) with $(\alpha_D, \alpha_G) \in (0,\infty]^2$, given a generator G_θ, the discriminator optimizing (52a) is

$$D_{\omega^*}(x) = \frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D} + p_{G_\theta}(x)^{\alpha_D}}, \quad x \in \mathcal{X}. \tag{53}$$

For this D_{ω^*} and the function $f_{\alpha_D,\alpha_G} : \mathbb{R}_+ \to \mathbb{R}$ defined as

$$f_{\alpha_D,\alpha_G}(u) = \frac{\alpha_G}{\alpha_G - 1}\left(\frac{u^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)+1}+1}{(u^{\alpha_D}+1)^{1-\frac{1}{\alpha_G}}} - 2^{\frac{1}{\alpha_G}}\right), \tag{54}$$

(52b) simplifies to minimizing a non-negative symmetric f_{α_D,α_G}-divergence $D_{f_{\alpha_D,\alpha_G}}(\cdot||\cdot)$ as

$$\inf_{\theta\in\Theta} D_{f_{\alpha_D,\alpha_G}}\big(P_r||P_{G_\theta}\big)+\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right), \tag{55}$$

which is minimized iff $P_{G_\theta}=P_r$ for (α_D,α_G) such that $(\alpha_D\leq 1,\ \alpha_G>\frac{\alpha_D}{\alpha_D+1})$ or $(\alpha_D>1,\ \frac{\alpha_D}{2}<\alpha_G\leq\alpha_D)$.

Proof sketch: We substitute the optimal discriminator of (52a) into the objective function of (52b) and write the resulting expression in the form

$$\int_{\mathcal{X}} p_{G_\theta}(x) f_{\alpha_D,\alpha_G}\left(\frac{p_r(x)}{p_{G_\theta}(x)}\right)dx+\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right). \tag{56}$$

We then find the conditions on α_D and α_G for f_{α_D,α_G} to be strictly convex so that the first term in (56) is an f-divergence. Figure 12(a) in Appendix K illustrates the feasible (α_D,α_G)-region. A detailed proof can be found in Appendix K. See Fig. 6 for a toy example illustrating the value of tuning $\alpha_D<1$ and $\alpha_G\geq 1$. ∎

Noting that α-GAN recovers various well-known GANs, including the vanilla GAN, which is prone to saturation, the (α_D,α_G)-GAN formulation using the generator objective function in (51) can similarly saturate early in training, potentially causing vanishing gradients. We propose the following NS alternative to the generator's objective in (51):

$$V^{\text{NS}}_{\alpha_G}(\theta,\omega)=\mathbb{E}_{X\sim P_{G_\theta}}\big[\ell_{\alpha_G}(1,D_\omega(X))\big], \tag{57}$$

thereby replacing (52b) with

$$\inf_{\theta\in\Theta} V^{\text{NS}}_{\alpha_G}(\theta,\omega). \tag{58}$$

Comparing (52b) and (58), note that the additional expectation term over P_r in (51) results in (52b) simplifying to a symmetric divergence for D_{ω^*} in (53), whereas the single term in (57) will result in (58) simplifying to an asymmetric divergence. The optimal discriminator for this NS game remains the same as in (53). The following theorem provides the solution to (58) under the assumption that the optimal discriminator can be attained.

Theorem 11: For the same D_{ω^*} in (53) and the function $f^{\text{NS}}_{\alpha_D,\alpha_G}:\mathbb{R}_+\to\mathbb{R}$ defined as

$$f^{\text{NS}}_{\alpha_D,\alpha_G}(u)=\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}-1}-\frac{u^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)}}{\left(u^{\alpha_D}+1\right)^{1-\frac{1}{\alpha_G}}}\right), \tag{59}$$

(52b) simplifies to minimizing a non-negative asymmetric $f^{\text{NS}}_{\alpha_D,\alpha_G}$-divergence $D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(\cdot||\cdot)$ as

$$\inf_{\theta\in\Theta} D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}\big(P_r||P_{G_\theta}\big)+\frac{\alpha_G}{\alpha_G-1}\left(1-2^{\frac{1}{\alpha_G}-1}\right), \tag{60}$$

which is minimized iff $P_{G_\theta}=P_r$ for $(\alpha_D,\alpha_G)\in(0,\infty]^2$ such that $\alpha_D+\alpha_G>\alpha_G\alpha_D$.

The proof mimics that of Theorem 10 and is detailed in Appendix L. Figure 12(b) in Appendix L illustrates the feasible (α_D,α_G)-region; in contrast to the saturating setting of Theorem 10, the NS setting constrains $\alpha\leq 2$ when $\alpha_D=\alpha_G=\alpha$. See Fig. 6(c) for a toy example illustrating how tuning $\alpha_D<1$ and $\alpha_G\geq 1$ can also alleviate training instabilities in the NS setting.

We note that the input to the discriminator is a random variable X which can be viewed as being sampled from a mixture distribution, i.e., $X\sim\delta P_r+(1-\delta)P_{G_\theta}$ where $\delta\in(0,1)$. Without loss of generality, we assume $\delta=1/2$ but the analysis that follows can be generalized for arbitrary δ. We use the Bernoulli random variable $Y\in\{0,1\}$ to indicate that $X=x$ is from the real $(Y=1)$ or generated $(Y=0)$ distributions. Therefore, the marginal probabilities of the two classes are $P_Y(1)=1-P_Y(0)=\delta=1/2$. Thus, one can then compute the true posterior $P_{Y|X}(1|x)$ and its tilted version $P^{(\alpha_D)}_{Y|X}(1|x)$ as follows:

$$P_{Y|X}(1|x)=\frac{p_r(x)}{p_r(x)+p_{G_\theta}(x)}, \tag{61a}$$

$$P^{(\alpha_D)}_{Y|X}(1|x)=\frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D}+p_{G_\theta}(x)^{\alpha_D}}, \tag{61b}$$

where both expressions simplify to the optimal discriminator of the vanilla GAN in (6) for $\alpha_D=1$.

We now present a theorem to quantify precisely the effect of tuning α_D and α_G. To this end, we begin by first taking a closer look at the gradients induced by the generator's loss during training. To simplify our analysis, we assume that at every step of training, the discriminator can achieve its optimum, D_{ω^*}.[11] For any sample $x=G_\theta(z)$ generated by G, we can write the gradient of the generator's loss for an (α_D,α_G)-GAN *w.r.t.* its weight vector θ as

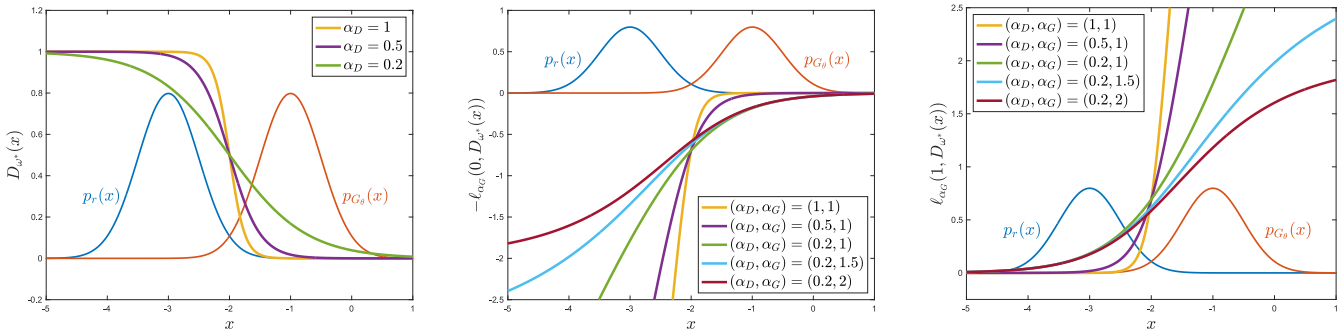
$$\begin{aligned}-\frac{\partial\ell_{\alpha_G}(0,D_{\omega^*}(x))}{\partial\theta}&=-\frac{\partial\ell_{\alpha_G}(0,D_{\omega^*}(x))}{\partial x}\times\frac{\partial x}{\partial\theta}\\&=-\frac{\partial\ell_{\alpha_G}(0,D_{\omega^*}(x))}{\partial D_{\omega^*}(x)}\times\frac{\partial D_{\omega^*}(x)}{\partial x}\times\frac{\partial x}{\partial\theta}.\end{aligned} \tag{62}$$

We note that while we cannot explicitly analyze the term $\frac{\partial x}{\partial\theta}$ in (62), we assume that by using models satisfying boundedness and Lipschitz assumptions,[12] this term will not be unbounded. We thus focus on the first two terms on the right side of (62) for any α_G. For $\alpha_D=1$, from (53), we see that in regions densely populated by the generated but not the real data, $D_{\omega^*}(x)\to 0$. Further, the first term in (53) is bounded thus causing the gradient in (62) to vanish. On the other hand, when $\alpha_D<1$, D_{ω^*} increases (resp. decreases) in areas denser in generated (resp. real) data, thereby providing more gradients for G. This is clearly illustrated in Fig. 6(a) and 6(b) and reveals how strongly dependent the optimization trajectory traversed by G during training is on the practitioner's choice of $(\alpha_D,\alpha_G)\in(0,\infty]^2$. In fact, this holds irrespective of the saturating or the NS (α_D,α_G)-GAN. In the following theorem, we offer deeper insights into how such an optimization trajectory is influenced by tuning α_D and α_G.

Theorem 12: For a given P_r and P_{G_θ}, let x be a sample generated according to P_{G_θ}, and D_{ω^*} be optimal with respect to $V_{\alpha_D}(\theta,\omega)$. Then

[11] We note that a related gradient analysis was considered by Shannon [56, Sec. 3.1] for f-GANs assuming an optimal discriminator.

[12] These assumptions match practical settings.



(a) (b) (c)

Fig. 6. (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (53) for several values of $\alpha_D \leq 1$ for the same toy example as in Figure 5. Tuning $\alpha_D < 1$ reduces the confidence of the optimal discriminator D_{ω^*}. (b) A plot of the generator's loss $-\ell_{\alpha_G}(0, D_{\omega^*}(x))$ for several values of $(\alpha_D \leq 1, \alpha_G \geq 1)$. Tuning $\alpha_D < 1$ and $\alpha_G = 1$ provides larger gradients for the generated data far from the real data, thereby alleviating vanishing gradients, and also provides smaller gradients for generated data close to the real data, helping to combat exploding gradients. Tuning $\alpha_G \geq 1$ yields a quasiconcave objective, further reducing the magnitude of the gradients for generated data approaching the real data. (c) A plot of the generator's NS loss $\ell_{\alpha_G}(1, D_{\omega^*}(x))$ for several values of $(\alpha_D \leq 1, \alpha_G \geq 1)$. Tuning $\alpha_D < 1$ and $\alpha_G = 1$ reduces the magnitude of the gradients for generated data far from the real data, which can help stabilize training by decreasing sensitivity to hyperparameter initialization and alleviating model oscillation; tuning $\alpha_G > 1$ yields a quasiconvex generator objective, which can potentially further improve training stability.

(a) the **saturating** and **non-saturating** gradients, $-\partial \ell_{\alpha_G}(0, D_{\omega^*}(x))/\partial x$ and $\partial \ell_{\alpha_G}(1, D_{\omega^*}(x))/\partial x$, respectively, demonstrate the following behavior:

$$-\frac{\partial \ell_{\alpha_G}(0, D_{\omega^*}(x))}{\partial x} = C_{x,\alpha_D,\alpha_G}\left(\frac{1}{p_{G_\theta}(x)}\frac{\partial p_{G_\theta}}{\partial x} - \frac{1}{p_r(x)}\frac{\partial p_r}{\partial x}\right) \quad (63)$$

$$\frac{\partial \ell_{\alpha_G}(1, D_{\omega^*}(x))}{\partial x} = C^{\mathrm{NS}}_{x,\alpha_D,\alpha_G}\left(\frac{1}{p_{G_\theta}(x)}\frac{\partial p_{G_\theta}}{\partial x} - \frac{1}{p_r(x)}\frac{\partial p_r}{\partial x}\right), \quad (64)$$

where using the tilted probability $P^{(\alpha_D)}_{Y|X}(1|x)$ as written in (61),

$$C_{x,\alpha_D,\alpha_G} := \alpha_D P^{(\alpha_D)}_{Y|X}(1|x)\left(1 - P^{(\alpha_D)}_{Y|X}(1|x)\right)^{1-1/\alpha_G}, \quad (65)$$

$$C^{\mathrm{NS}}_{x,\alpha_D,\alpha_G} := \alpha_D\left(1 - P^{(\alpha_D)}_{Y|X}(1|x)\right)P^{(\alpha_D)}_{Y|X}(1|x)^{1-1/\alpha_G}; \quad (66)$$

(b) the gradients in both (63) and (65) have directions that are independent of α_D and α_G.

Remark 5: One can view the results in Theorem 12 above as a one-shot (in any iteration) analysis of the gradients of the generator's loss, and thus, we fix P_{G_θ}. Doing so allows us to ignore the implicit dependence on (α_D, α_G) of the P_{G_θ} learned up to this iteration, thus allowing us to obtain tractable expressions for any iteration.

A detailed proof of Theorem 12 can be found in Fig. 13 in Appendix M. Focusing first on saturating (α_D, α_G)-GANs, in Appendix M, we plot C_{x,α_D,α_G} as a function of $P_{Y|X}(1|x)$ defined in (61) for five (α_D, α_G) combinations. In the $(1, 1)$ case (i.e., vanilla GAN), $C_{x,1,1} \approx 0$ for generated samples far away from the real data (where $P_{Y|X}(1|x) \approx 0$). As discussed earlier, this optimization strategy is troublesome when the real and generated data are fully separable, since the sample gradients are essentially zeroed out by the scalar, leading to vanishing gradients. To address this issue, Fig. 13(a) shows that tuning α_D below 1 (e.g., 0.6) ensures that samples most likely to be "generated" ($P_{Y|X}(1|x) \approx 0$) receive sufficient gradient for updates that direct them closer to the real distribution.

The vanilla GAN also suffers from convergence issues since generated samples close to the real data (when $P_{Y|X}(1|x) \approx 1$) receive gradients large in magnitude ($C_{x,1,1} \approx 1$). Ideally, these generated samples should not be instructed to move since they convincingly pass as real to D_{ω^*}. As explained in Section II-B, an excessive gradient can push the generated data away from the real data, which ultimately separates the distributions and forces the GAN to restart training. Although the $(0.6, 1)$-GAN in Fig. 13(a) appears to decrease C_{x,α_D,α_G} for samples close to the real data ($P_{Y|X}(1|x) \approx 1$), tuning $\alpha_G > 1$ allows this gradient to converge to zero as desired (see Fig. 6(b)).

Although tuning the saturating (α_D, α_G)-GAN formulation away from vanilla GAN promotes a more favorable optimization trajectory for G, this approach continues to suffer from the problem of providing small gradients for generated samples far from P_r. This suggests looking at the behavior of the NS (α_D, α_G)-GAN formulation. Figure Fig. 13(b) in Appendix M illustrates the relationship between the gradient scalar $C^{\mathrm{NS}}_{x,\alpha_D,\alpha_G}$ and $P_{Y|X}(1|x)$ defined in (61) for several values of (α_D, α_G). In the vanilla $(1, 1)$-GAN case, we observe a negative linear relationship, i.e., the samples least likely to be real ($P_{Y|X}(1|x) \approx 0$) receive large gradients ($C^{\mathrm{NS}}_{x,1,1} \approx 1$) while the samples most likely to be real receive minimal gradients ($C^{\mathrm{NS}}_{x,1,1} \approx 0$). While this seems desirable, unfortunately, the vanilla GAN's optimization strategy often renders it vulnerable to model oscillation, a common GAN failure detailed in Section II-B, as a result of such large gradients of the outlier (far from real) samples causing the generated data to oscillate around the real data modes. By tuning α_D below 1, as shown in Fig. 13(b), one can slightly increase (resp. decrease) $C^{\mathrm{NS}}_{x,\alpha_D,\alpha_G}$ for the generated samples close to (resp. far from) the real modes. As a result, the generated samples are more robust to outliers and therefore more likely to converge to the real modes. Finally, tuning α_G above 1 can further improve this robustness. A caveat here is the fact that $C^{\mathrm{NS}}_{x,\alpha_D,\alpha_G} \approx 0$ when

$P_{Y|X}(1|x) \approx 0$ can potentially be problematic since the near-zero gradients may immobilize generated data far from the real distribution. This is borne out in our results for several large image datasets in Section V where choosing $\alpha_G = 1$ yields the best results. The cumulative effects of tuning (α_D, α_G) are further illustrated in Fig. 6(c).

B. CPE Loss Based Dual-Objective GANs

Similarly to the single-objective loss function perspective in Section III, we can generalize the (α_D, α_G)-GAN formulation to incorporate general CPE losses. To this end, we introduce a dual-objective loss function perspective of GANs in which the discriminator maximizes $V_{\ell_D}(\theta, \omega)$ while the generator minimizes $V_{\ell_G}(\theta, \omega)$, where

$$V_\ell(\theta, \omega) = \mathbb{E}_{X\sim P_r}[-\ell(1, D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[-\ell(0, D_\omega(X))], \tag{67}$$

for any CPE losses $\ell = \ell_D, \ell_G$. The resulting CPE loss dual-objective GAN is given by

$$\sup_{\omega\in\Omega} V_{\ell_D}(\theta, \omega) \tag{68a}$$

$$\inf_{\theta\in\Theta} V_{\ell_G}(\theta, \omega). \tag{68b}$$

The CPE losses ℓ_D and ℓ_G can be completely different losses, the same loss but with different parameter values, or the same loss with the same parameter values, in which case the above formulation reduces to the single-objective formulation in (14). For example, choosing $\ell_D = \ell_G = \ell_\alpha$, we recover the α-GAN formulation in (20); choosing $\ell_D = \ell_{\alpha_D}$ and $\ell_G = \ell_{\alpha_G}$, we obtain the (α_D, α_G)-GAN formulation in (52). Note that ℓ_D should satisfy the constraint in (15) so that the optimal discriminator outputs $1/2$ for any input when $P_r = P_{G_\theta}$. We once again maintain the same ordering as the original min-max GAN formulation and present the conditions under which the optimal generator minimizes a symmetric f-divergence when the discriminator set Ω is large enough in the following proposition.

Proposition 1: Let ℓ_D and ℓ_G be symmetric CPE loss functions with $\ell_D(1, \cdot)$ also differentiable with derivative $\ell_D'(1, \cdot)$ and strictly convex. Then the optimal discriminator D_{ω^*} optimizing (68a) satisfies the implicit equation, provided it has a solution,

$$\ell_D'(1, 1-D_{\omega^*}(x)) = \frac{p_r(x)}{p_{G_\theta}(x)}\ell_D'(1, D_{\omega^*}(x)), \quad x \in \mathcal{X}. \tag{69}$$

If (69) does not have a solution for a particular $x \in \mathcal{X}$, then $D_{\omega^*}(x) = 0$ or $D_{\omega^*}(x) = 1$. Let $A(\frac{p_r(x)}{p_{G_\theta}(x)}) := D_{\omega^*}(x)$. For this D_{ω^*}, (68b) simplifies to minimizing a symmetric f-divergence $D_f(P_r||P_{G_\theta})$ if the function $f : \mathbb{R}_+ \to \mathbb{R}$ is convex, where f is defined as

$$f(u) = -u\ell_G(1, A(u)) - \ell_G(1, 1-A(u)) + 2\ell_G(1, 1/2). \tag{70}$$

Proof sketch: The proof involves a straightforward application of KKT conditions when optimizing (68a) and substituting in (68b). A detailed proof can be found in Appendix N. ∎

As it is difficult to come up with conditions without having the explicit forms of the losses ℓ_D and ℓ_G, Proposition 1 provides a broad outline of what the optimal strategies will look like. The assumption of the losses being symmetric can be relaxed, in which case the resulting f-divergence will no longer be guaranteed to be symmetric. Theorem 10 is a special case of Proposition 1 for $\ell_D = \ell_{\alpha_D}$ and $\ell_G = \ell_{\alpha_G}$. Proposition 1 also recovers [38, Th. 1] when $\ell_D(y, \hat{y}) = \ell_{\text{CE}}(y, \hat{y}) := -y\log\hat{y} - (1-y)\log(1-\hat{y})$ and $\ell_G = \mathcal{L}_\alpha$ as defined in [38, Def. 3]. As another example, consider the following square loss based CPE losses[13]:

$$\ell_D(y, \hat{y}) = \frac{1}{2}\left[y(\hat{y}-1)^2 + (1-y)\hat{y}^2\right] \tag{71}$$

$$\ell_G(y, \hat{y}) = -\frac{1}{2}\left[y\left(\hat{y}^2-1\right) + (1-y)\left(\left(1-\hat{y}\right)^2-1\right)\right]. \tag{72}$$

Note that (71) and (72) are both symmetric and $\ell_D(1, \cdot)$ is both convex (and therefore ℓ_D satisfies (15)) and differentiable with $\ell_D'(1, \hat{y}) = \hat{y}-1$. The implicit equation in (69) then becomes

$$(1-D_{\omega^*}(x))-1 = u(D_{\omega^*}(x)-1),$$

where

$$D_{\omega^*}(x) = \frac{u}{u+1} = \frac{p_r(x)}{p_r(x)+p_{G_\theta}(x)}.$$

The corresponding f in (70) is $f(u) = [3(1-u)]/[4(u+1)]$, which is convex. Therefore, the dual-objective CPE loss GAN using (71) and (72) minimizes a symmetric f-divergence.

C. Estimation Error for CPE Loss Dual-Objective GANs

In order to analyze what occurs in practice when both the number of training samples and model capacity are usually limited, we now consider the same setting as in Section III-C with finite training samples $S_x = \{X_1, \ldots, X_n\}$ and $S_z = \{Z_1, \ldots, Z_m\}$ from P_r and P_Z, respectively, and with neural networks chosen as the discriminator and generator models. The sets of samples S_x and S_z induce the empirical real and generated distributions $\hat{P}_r$ and $\hat{P}_{G_\theta}$, respectively. A useful quantity to evaluate the performance of GANs in this setting is again that of the estimation error. In Section III-C, we define estimation error for CPE loss GANs. However, such a definition requires a common value function for both discriminator and generator, and therefore, does not directly apply to the dual-objective setting we consider here.

Our definition relies on the observation that estimation error inherently captures the effectiveness of the generator (for a corresponding optimal discriminator model) in learning with limited samples. We formalize this intuition below.

Since CPE loss dual-objective GANs use different objective functions for the discriminator and generator, we start by defining the optimal discriminator ω^* for a generator model G_θ as

$$\omega^*(P_r, P_{G_\theta}) := \arg\max_{\omega\in\Omega} V_{\ell_D}(\theta, \omega)\big|_{P_r, P_{G_\theta}}, \tag{73}$$

where the notation $|_{\cdot,\cdot}$ allows us to make explicit the distributions used in the value function. In keeping with the literature where the value function being minimized is referred to as the neural net (NN) distance (since D and G are modeled

[13]Note that these losses were considered in [38] and were shown to result in a special case of a shifted LSGAN minimizing a certain Jensen-f-divergence.

as neural networks) [12], [19], [41], we define the generator's NN distance $d_{\omega^*(P_r,P_{G_\theta})}$ as

$$d_{\omega^*(P_r,P_{G_\theta})}(P_r,P_{G_\theta}) \coloneqq V_{\ell_G}(\theta,\omega^*(P_r,P_{G_\theta}))\big|_{P_r,P_{G_\theta}}. \quad (74)$$

The resulting minimization for training the CPE-loss dual-objective GAN using finite samples is

$$\inf_{\theta\in\Theta} d_{\omega^*\left(\hat{P}_r,\hat{P}_{G_\theta}\right)}\left(\hat{P}_r,\hat{P}_{G_\theta}\right). \quad (75)$$

Denoting $\hat{\theta}^*$ as the minimizer of (75), we define the estimation error for CPE loss dual-objective GANs as

$$d_{\omega^*\left(P_r,P_{G_{\hat{\theta}^*}}\right)}\left(P_r,P_{G_{\hat{\theta}^*}}\right)-\inf_{\theta\in\Theta} d_{\omega^*(P_r,P_{G_\theta})}(P_r,P_{G_\theta}). \quad (76)$$

We use the same notation as in Section III-C, detailed again in the following for easy reference. For $x\in\mathcal{X}\coloneqq\{x\in\mathbb{R}^d : ||x||_2\le B_x\}$ and $z\in\mathcal{Z}\coloneqq\{z\in\mathbb{R}^p : ||z||_2\le B_z\}$, we model the discriminator and generator as k- and l-layer neural networks, respectively, such that D_ω and G_θ can be written as:

$$D_\omega : x\mapsto \sigma\left(\mathbf{w}_k^{\mathsf{T}} r_{k-1}(\mathbf{W}_{k-1}r_{k-2}(\dots r_1(\mathbf{W}_1x)))\right) \quad (77)$$

$$G_\theta : z\mapsto \mathbf{V}_l s_{l-1}(\mathbf{V}_{l-1}s_{l-2}(\dots s_1(\mathbf{V}_1z))), \quad (78)$$

where (i) $\mathbf{w}_k$ is a parameter vector of the output layer; (ii) for $i\in[1:k-1]$ and $j\in[1:l]$, $\mathbf{W}_i$ and $\mathbf{V}_j$ are parameter matrices; (iii) $r_i(\cdot)$ and $s_j(\cdot)$ are entry-wise activation functions of layers i and j, respectively, i.e., for $\mathbf{a}\in\mathbb{R}^t$, $r_i(\mathbf{a})=[r_i(a_1),\dots,r_i(a_t)]$ and $s_i(\mathbf{a})=[s_i(a_1),\dots,s_i(a_t)]$; and (iv) $\sigma(\cdot)$ is the sigmoid function given by $\sigma(p)=1/(1+\mathrm{e}^{-p})$. We assume that each $r_i(\cdot)$ and $s_j(\cdot)$ are R_i- and S_j-Lipschitz, respectively, and also that they are positive homogeneous, i.e., $r_i(\lambda p)=\lambda r_i(p)$ and $s_j(\lambda p)=\lambda s_j(p)$, for any $\lambda\ge 0$ and $p\in\mathbb{R}$. Finally, as is common in such analysis [41], [60], [61], [62], we assume that the Frobenius norms of the parameter matrices are bounded, i.e., $||\mathbf{W}_i||_F\le M_i$, $i\in[1:k-1]$, $||\mathbf{w}_k||_2\le M_k$, and $||\mathbf{V}_j||_F\le N_j$, $j\in[1:l]$. We now present an upper bound on (76) in the following theorem.

Theorem 13: For the setting described above, additionally assume that the functions $\phi(\cdot)\coloneqq-\ell_G(1,\cdot)$ and $\psi(\cdot)\coloneqq-\ell_G(0,\cdot)$ are L_ϕ- and L_ψ-Lipschitz, respectively. Then, with probability at least $1-2\delta$ over the randomness of training samples $S_x=\{X_i\}_{i=1}^n$ and $S_z=\{Z_j\}_{j=1}^m$, we have

$$d_{\omega^*\left(P_r,P_{G_{\hat{\theta}^*}}\right)}\left(P_r,P_{G_{\hat{\theta}^*}}\right)-\inf_{\theta\in\Theta} d_{\omega^*(P_r,P_{G_\theta})}(P_r,P_{G_\theta}) \le \frac{L_\phi B_x U_\omega\sqrt{3k}}{\sqrt{n}}+\frac{L_\psi U_\omega U_\theta B_z\sqrt{3(k+l-1)}}{\sqrt{m}} U_\omega\sqrt{\log\frac{1}{\delta}}\left(\frac{L_\phi B_x}{\sqrt{2n}}+\frac{L_\psi B_z U_\theta}{\sqrt{2m}}\right), \quad (79)$$

where $U_\omega\coloneqq M_k\prod_{i=1}^{k-1}(M_iR_i)$ and $U_\theta\coloneqq N_l\prod_{j=1}^{l-1}(N_jS_j)$.

In particular, when specialized to the case of (α_D,α_G)-GANs by letting $\phi(p)=\psi(1-p)=\frac{\alpha_G}{\alpha_G-1}(1-p^{\frac{\alpha_G-1}{\alpha_G}})$, the resulting bound is nearly identical to the terms in the RHS of (79), except for substitutions $L_\phi\leftarrow 4C_{Q_x}(\alpha_G)$ and $L_\psi\leftarrow 4C_{Q_z}(\alpha_G)$, where $Q_x\coloneqq U_\omega B_x$, $Q_z\coloneqq U_\omega U_\theta B_z$, and

$$C_h(\alpha)\coloneqq\begin{cases}\sigma(h)\sigma(-h)^{\frac{\alpha-1}{\alpha}}, & \alpha\in(0,1]\\ \left(\frac{\alpha-1}{2\alpha-1}\right)^{\frac{\alpha-1}{\alpha}}\frac{\alpha}{2\alpha-1}, & \alpha\in(1,\infty).\end{cases} \quad (80)$$

The proof is similar to that of Theorem 8 (and also [41, Th. 1]). We observe that (80) does not depend on ℓ_D, an artifact of the proof techniques used, and is therefore most likely not the tightest bound possible. See Appendix O for proof details.

V. Illustration of Results

We illustrate the value of (α_D,α_G)-GAN as compared to the vanilla GAN (i.e., the (1, 1)-GAN). Focusing on DCGAN architectures [28], we compare against LSGANs [15], one of the current state-of-the-art (SOTA) dual-objective approach.[14] While WGANs [4] have also been proposed to address the training instabilities, their training methodology is distinctly different and uses a different optimizer (RMSprop), requires gradient clipping or penalty, and does not leverage batch normalization, all of which make meaningful comparisons difficult.

We evaluate our approach on three datasets: (i) a synthetic dataset generated by a two-dimensional, ring-shaped Gaussian mixture distribution (2D-ring) [65]; (ii) the 64×64 Celeb-A image dataset [66]; and (iii) the 112×112 LSUN Classroom dataset [67]. For each dataset and pair of GAN objectives, we report several metrics that encapsulate the stability of GAN training over hundreds of random seeds. This allows us to clearly showcase the potential for tuning (α_D,α_G) to obtain stable and robust solutions for image generation.

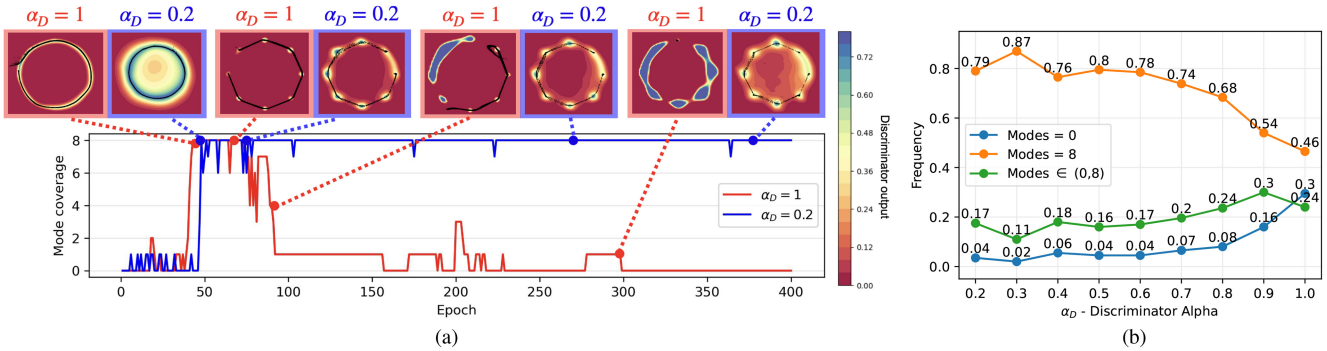
A. 2D Gaussian Mixture Ring

The 2D-ring is an oft-used synthetic dataset for evaluating GANs. We draw samples from a mixture of 8 equal-prior Gaussian distributions, indexed $i\in\{1,2,\dots,8\}$, with a mean of $(\cos(2\pi i/8),\ \sin(2\pi i/8))$ and variance 10^{-4}. We generate 50,000 training and 25,000 testing samples and the same number of 2D latent Gaussian noise vectors, where each entry is a standard Gaussian.

Both the D and G networks have 4 fully-connected layers with 200 and 400 units, respectively. We train for 400 epochs with a batch size of 128, and optimize with Adam [68] and a learning rate of 10^{-4} for both models. We consider three distinct settings that differ in the objective functions as: **(i)** (α_D,α_G)-GAN in (52); **(ii)** NS (α_D,α_G)-GAN's in (52a), (58); **(iii)** LSGAN with the 0-1 binary coding scheme (see Appendix P for details).

For every setting listed above, we train our models on the 2D-ring dataset for 200 random state seeds, where each seed contains different weight initializations for D and G. Ideally, a stable method will reflect similar performance across randomized initializations and also over training epochs; thus, we

[14]We acknowledge that LSGAN may not be SOTA for every architecture or dataset, as shown in [64], but for reasons of fair comparisons without regularizers, we restrict our comparisons to settings where we can evaluate the effect of loss functions.



(a) (b)

Fig. 7. (a) Plot of mode coverage over epochs for (α_D, α_G)-GAN training with the **saturating** objectives in (52). Fixing $\alpha_G = 1$, we compare $\alpha_D = 1$ (vanilla GAN) with $\alpha_D = 0.2$. Placed above this plot are 2D visuals of the generated samples (in black) at different epochs; these show that both GANs successfully capture the ring-like structure, but the vanilla GAN fails to maintain the ring over time. We illustrate the discriminator output in the same visual as a heat map to show that the $\alpha_D = 1$ discriminator exhibits more confident predictions (tending to 0 or 1), which in turn subjects G to vanishing and exploding gradients when its objective $\log(1-D)$ saturates as $D \to 0$ and diverges as $D \to 1$, respectively. This combination tends to repel the generated data when it approaches the real data, thus freezing any significant weight update in the future. In contrast, the less confident predictions of the (0.2, 1)-GAN create a smooth landscape for the generated output to descend towards the real data. (b) Plot of success and failure rates over 200 seeds vs. α_D with $\alpha_G = 1$ for the **saturating** (α_D, α_G)-GAN on the 2D-ring, which underscores the stability of $(\alpha_D < 1, \alpha_G)$-GANs relative to vanilla GAN.

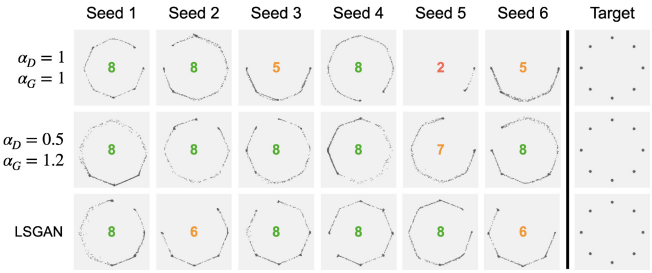


Fig. 8. Generated samples from two (α_D, α_G)-GANs trained with the **NS** objectives in (52a), (58), as well as LSGAN. We provide 6 seeds to illustrate the stability in performance for each GAN across multiple runs.

explore how GAN training performance for each setting varies across seeds and epochs. Our primary performance metric is *mode coverage*, defined as the number of Gaussians (0-8) that contain a generated sample within 3 standard deviations of its mean. A score of 8 conveys successful training, while a score of 0 conveys a significant GAN failure; on the other hand, a score in between 0 and 8 may be indicative of common GAN issues, such as mode collapse or failure to converge.

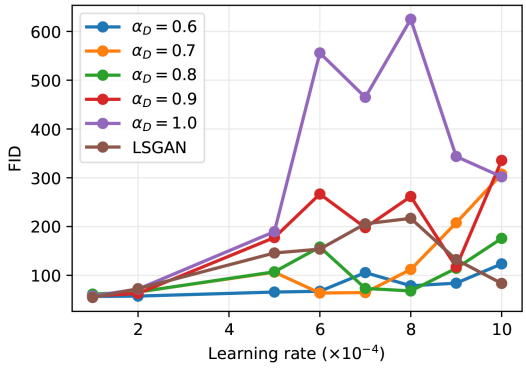
For the saturating setting, the improvement in stability of the (0.2, 1)-GAN relative to the vanilla GAN is illustrated in Fig. 7 as detailed in the caption. Vanilla GAN fails to converge to the true distribution 30% of the time while succeeding only 46% of the time. In contrast, the (α_D, α_G)-GAN with $\alpha_D < 1$ learns a more stable G due to a less confident D (see also Fig. 7(a)). For example, the (0.3, 1)-GAN success and failure rates improve to 87% and 2%, respectively. For the NS setting in Fig. 8, we find that tuning α_D and α_G yields more consistently stable outcomes than vanilla and LSGANs. Mode coverage rates over 200 seeds for saturating (Tables I and II) and NS (Table III) are in Appendix P.

B. Celeb-A & LSUN Classroom

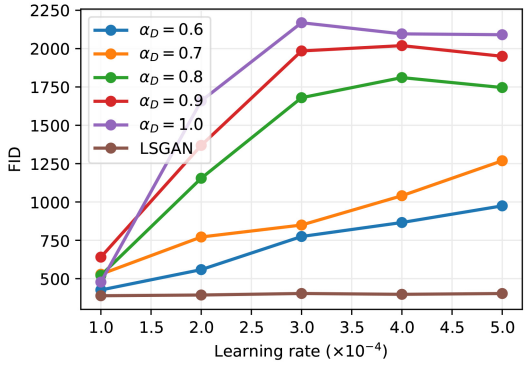
The Celeb-A dataset [66] is a widely recognized large-scale collection of over 200,000 celebrity headshots, encompassing images with diverse aspect ratios, camera angles, backgrounds, lighting conditions, and other variations. Similarly, the LSUN Classroom dataset [67] is a subset of the comprehensive Large-scale Scene Understanding (LSUN) dataset; it contains over 150,000 classroom images captured under diverse conditions and with varying aspect ratios. To ensure consistent input for the discriminator, we follow the standard practice of resizing the images to 64×64 for Celeb-A and 112×112 for LSUN Classroom. For both experiments, we randomly select 80% of the images for training and leave the remaining 20% for validation (evaluation of goodness metrics). Finally, for the generator, for each dataset, we generate a similar 80%-20% training-validation split of 100-dimensional latent Gaussian noise vectors, where each entry is a standard Gaussian, for a total matching the size of the true data.

For training, we employ the DCGAN architecture [28] that leverages deep convolutional neural networks (CNNs) for both D and G. In Appendix P, detailed descriptions of the D and G architectures can be found in Tables IV and V for the Celeb-A and LSUN Classroom datasets, respectively. Following SOTA methods, we focus on the non-saturating setting, utilizing appropriate objectives for vanilla GAN, (α_D, α_G)-GAN, and LSGAN. We consider a variety of learning rates, ranging from 10^{-4} to 10^{-3}, for Adam optimization. We evaluate our models every 10 epochs up to a total of 100 epochs and report the Fréchet Inception Distance (FID), an unsupervised similarity metric between the real and generated feature distributions extracted by InceptionNet-V3 [69]. For both datasets, we train each combination of objective function, number of epochs, and learning rate for 50 seeds. In the following subsections, we empirically demonstrate the dependence of the FID on learning rate and number of epochs for the vanilla GAN, (α_D, α_G)-GAN, and LSGAN. Achieving robustness to hyperparameter initialization is especially desirable in the unsupervised GAN setting as the choices that facilitate steady model convergence are not easily determined *a priori*.

1) Celeb-A Results: In Fig. 9(a), we examine the relationship between learning rate and FID for each GAN trained



(a)



(b)

Fig. 9. (a) Plot of **Celeb-A** FID scores averaged over 50 seeds vs. learning rates for 6 different GANs, trained for 100 epochs. (b) Plot of **LSUN Classroom** FID scores averaged over 50 seeds vs. learning rates for 6 different GANs, trained for 100 epochs.

for 100 epochs on the Celeb-A dataset. When using learning rates of 1×10^{-4} and 2×10^{-4}, all GANs consistently perform well. However, when the learning rate increases, the vanilla $(1, 1)$-GAN begins to exhibit instability across the 50 seeds. As the learning rate surpasses 5×10^{-4}, the performance of the vanilla GAN becomes even more erratic, underscoring the importance of GANs being robust to the choice of learning rate. Figure 9(a) also demonstrates that the GANs with $\alpha_D < 1$ perform on par with, if not better than, the SOTA LSGAN. For instance, the $(0.6, 1)$-GAN consistently achieves low FIDs across all tested learning rates.

In Fig. 10(a), for different learning rates, we compare the dependence on the number of training epochs (hyperparameter) of the vanilla $(1, 1)$-GAN, $(0.6, 1)$-GAN, and LSGAN by plotting their FIDs every 10 epochs, up to 100 epochs, for two similar learning rates: 5×10^{-4} and 6×10^{-4}. We discover that the vanilla $(1, 1)$-GAN performs significantly worse for the higher learning rate and deteriorates over time for both learning rates. Conversely, both the $(0.6, 1)$-GAN and LSGAN consistently exhibit favorable FID performance for both learning rates. However, the $(0.6, 1)$-GAN converges to a low FID, while the FID of the LSGAN slightly increases as training approaches 100 epochs. Finally, Fig. 10(b) displays a grid of generated Celeb-A faces, randomly sampled over 8 seeds for three GANs trained for 100 epochs with a learning rate of 5×10^{-4}. Here, we observe that the faces generated by the $(0.6, 1)$-GAN and LSGAN exhibit a comparable level of quality to the rightmost column images, which are randomly sampled from the real Celeb-A dataset. On the other hand, the vanilla $(1, 1)$-GAN shows clear signs of performance instability, as some seeds yield high-quality images while others do not.

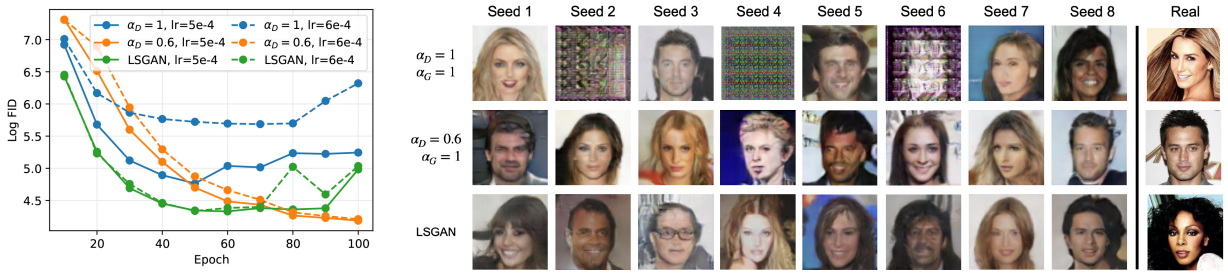
2) LSUN Classroom Results: In Fig. 9(b), we illustrate the relationship between learning rate and FID for GANs trained on the LSUN dataset for 100 epochs. In fact, when all GANs are trained with a learning rate of 1×10^{-4}, they consistently deliver satisfactory performance. However, increasing it to 2×10^{-4} leads to instability in the vanilla $(1, 1)$-GAN across 50 seeds.

On the other hand, we observe that $\alpha_D < 1$ contributes to stabilizing the FID across the 50 seeds even when trained with slightly higher learning rates. In Fig. 9(b), we see that as α_D is tuned down to 0.6, the mean FIDs consistently decrease across all tested learning rates. These lower FIDs can be attributed to the increased stability of the network. Despite the gains in GAN stability achieved by tuning down α_D, Fig. 9 demonstrates a noticeable disparity between the best (α_D, α_G)-GAN and the SOTA LSGAN. This suggests that there is still room for improvement in generating high-dimensional images with (α_D, α_G)-GANs.

In Appendix P, Fig. 14(a), we illustrate the average FID throughout the training process for three GANs: $(1, 1)$-GAN, $(0.6, 1)$-GAN, and LSGAN, using two different learning rates: 1×10^{-4} and 2×10^{-4}. These findings validate that the vanilla $(1, 1)$-GAN performs well when trained with the lower learning rate, but struggles significantly with the higher learning rate. In contrast, the $(0.6, 1)$-GAN exhibits less sensitivity to learning rate, while the LSGAN achieves nearly identical scores for both learning rates. In Fig. 14(b), we showcase the image quality generated by each GAN at epoch 100 with the higher learning rate. This plot highlights that the vanilla $(1, 1)$-GAN frequently fails during training, whereas the $(0.6, 1)$-GAN and LSGAN produce images that are more consistent in mimicking the real distribution. Finally, we present the FID vs. learning rate results for both datasets in Table VI in Appendix P. This allows yet another way to evaluate performance by comparing the percentage (out of 50 seeds) of FID scores below a desired threshold for each dataset, as detailed in the Appendix.

VI. Conclusion

Building on our prior work introducing CPE loss GANs and α-GANs, we have introduced new results on the equivalence of CPE loss GANs and f-GANs, convergence properties of the symmetric f-divergences induced by CPE loss GANs under certain conditions, and the generalization and estimation error for CPE loss GANs including α-GANs. We have introduced a dual-objective GAN formulation, focusing in particular on using α-loss with potentially different α values for both players' objectives. GANs offer an alternative to diffusion models in being faster to train but training instabilities stymie such advantages. In this context, our results highlight how tuning α can not only alleviate training instabilities



(a) (b)

Fig. 10. (a) Log-scale plot of **Celeb-A** FID scores over training epochs in steps of 10 up to 100 total, for three specific (α_D, α_G)-GANs – (1, 1)-GAN (vanilla), (0.6, 1)-GAN, and LSGAN– and for two similar learning rates– 5×10^{-4} and 6×10^{-4}. Results show that the vanilla GAN performance is sensitive to learning rate choice, while the other two GANs achieve consistently low FIDs. (b) Generated Celeb-A faces from the same three GANs over 8 seeds when trained for 100 epochs with a learning rate of 5×10^{-4}. These samples show that the vanilla (1, 1)-GAN training is sensitive to random model weight initializations, while the other two GANs demonstrate both robustness to random weight initializations as well as realistic face generation.

but also enhance robustness to learning rates and training epochs, hyperparameters whose optimal values are generally not known *a priori*. A natural extension to our work is to define and study generalization of dual-objective GANs. An equally important problem is to evaluate if our observations hold more broadly, including, when the training data is noisy [70].

While different f-divergence based GANs have been introduced, no principled reasons have been proposed thus far for choosing a specific f-divergence measure and corresponding loss functions to optimize. Even in the more practical finite sample and model capacity settings, different choices of objectives, as shown earlier, lead to different neural network divergence measures. Using tunable losses, our work has the advantage of motivating the choice of appropriate loss functions and the resulting f-divergence/neural network divergence from the crucial viewpoint of avoiding training instabilities. This connection between loss functions and divergences to identify the appropriate measure of goodness can be of broader interest both to the IT and ML communities.

REFERENCES

[1] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[2] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.

[3] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, vol. 70, 2017, pp. 214–223.

[5] B. K. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. R. Lanckriet, "On the empirical estimation of integral probability metrics," *Electron. J. Statist.*, vol. 6, pp. 1550–1599, 2012.

[6] T. Liang, "How well generative adversarial networks learn distributions," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 10366–10406, 2021.

[7] G. R. Kurri, T. Sypherd, and L. Sankar, "Realizing GANs via a tunable loss function," in *Proc. IEEE Inf. Theory Workshop (ITW)*, 2021, pp. 1–6.

[8] T. Sypherd, M. Diaz, L. Sankar, and P. Kairouz, "A tunable loss function for binary classification," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2019, pp. 2479–2483.

[9] T. Sypherd, M. Diaz, J. K. Cava, G. Dasarathy, P. Kairouz, and L. Sankar, "A tunable loss function for robust classification: Calibration, landscape, and generalization," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 6021–6051, Sep. 2022.

[10] F. Österreicher, "On a class of perimeter-type distances of probability distributions," *Kybernetika*, vol. 32, no. 4, pp. 389–393, 1996.

[11] F. Liese and I. Vajda, "On divergences and informations in statistics and information theory," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4394–4412, Oct. 2006.

[12] G. R. Kurri, M. Welfert, T. Sypherd, and L. Sankar, "α-GAN: Convergence and estimation guarantees," in *Proc. IEEE ISIT*, 2022, pp. 276–281.

[13] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017, *arXiv:1701.04862*.

[14] M. Wiatrak, S. V. Albrecht, and A. Nystrom, "Stabilizing generative adversarial networks: A survey," 2019, *arXiv:1910.00927*.

[15] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 1–9.

[16] H. Bhatia, W. Paul, F. Alajaji, B. Gharesifard, and P. Burlina, "Least kth-order and Rényi generative adversarial networks," *Neural Comput.*, vol. 33, no. 9, pp. 2473–2510, 2021.

[17] B. Poole, A. A. Alemi, J. Sohl-Dickstein, and A. Angelova, "Improved generator objectives for GANs," 2016, *arXiv:1612.02780*.

[18] M. Welfert, K. Otstot, G. R. Kurri, and L. Sankar, "(α_D, α_G)-GANs: Addressing GAN training instabilities via dual objectives," in *Proc. IEEE ISIT*, 2023, pp. 1–12.

[19] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang, "Generalization and equilibrium in generative adversarial nets (GANs)," in *Proc. 34th ICML*, vol. 70, 2017, pp. 224–232.

[20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–10.

[21] S. Zhao, H. Ren, A. Yuan, J. Song, N. Goodman, and S. Ermon, "Bias and generalization in deep generative models: An empirical study," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–16.

[22] F. Huszár, "How (not) to train your generative model: Scheduled sampling, likelihood, adversary?" 2015, *arXiv:1511.05101*.

[23] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos, "MMD GAN: Towards deeper understanding of moment matching network," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[24] D. Berthelot, T. Schumm, and L. Metz, "BEGAN: Boundary equilibrium generative adversarial networks," 2017, *arXiv:1703.10717*.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–20.

[26] Y. Mroueh and T. Sercu, "Fisher GAN," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–29.

[27] Y. Mroueh, T. Sercu, and V. Goel, "McGan: Mean and covariance feature matching GAN," in *Proc. 34th ICML*, vol. 70, 2017, pp. 2527–2535.

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

[29] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*.

[30] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*.

[31] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 1–10.

[32] G. K. Dziugaite, D. M. Roy, and Z. Ghahramani, "Training generative neural networks via maximum mean discrepancy optimization," 2015, *arXiv:1505.03906*.
[33] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *Proc. 32nd ICML*, vol. 37, 2015, pp. 1718–1727.
[34] Y. Mroueh, C.-L. Li, T. Sercu, A. Raj, and Y. Cheng, "Sobolev GAN," 2017, *arXiv:1711.04894*.
[35] Z. Lin, A. Khetan, G. Fanti, and S. Oh, "PacGAN: The power of two samples in generative adversarial networks," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 324–335, May 2020.
[36] D. Reshetova, Y. Bai, X. Wu, and A. Özgür, "Understanding Entropic Regularization in GANs," in *Proc. IEEE ISIT*, 2021, pp. 825–830.
[37] D. A. Mesa, J. Tantiongloc, M. Mendoza, S. Kim, and T. P. Coleman, "A distributed framework for the construction of transport maps," *Neural Comput.*, vol. 31, no. 4, pp. 613–652, 2019.
[38] J. Veiner, F. Alajaji, and B. Gharesifard, "A unifying generator loss function for generative adversarial networks," 2023, *arXiv:2308.07233*.
[39] A. Robey, F. Latorre, G. J. Pappas, H. Hassani, and V. Cevher, "Adversarial training should be cast as a non-zero-sum game," 2023, *arXiv:2306.11035*.
[40] D. Zhou, P. Zhang, Q. Liu, T. Xu, and X. He, "On the discrimination-Generalization tradeoff in GANs," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–26.
[41] K. Ji, Y. Zhou, and Y. Liang, "Understanding estimation and Generalization error of generative adversarial networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 5, pp. 3114–3129, May 2021.
[42] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.
[43] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "On surrogate loss functions and f-divergences," *Ann. Statist.*, vol. 37, no. 2, pp. 876–904, 2009.
[44] I. Csiszár, "Information-type measures of difference of probability distributions and indirect observation," *Studia Scientiarum Mathematicarum Hungarica*, vol. 2, pp. 229–318, 1967.
[45] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *J. Roy. Stat. Soc. Ser. B*, vol. 28, no. 1, pp. 131–142, 1966.
[46] C. Villani, *Optimal Transport: Old and New*, vol. 338. Berlin, Germany: Springer, 2008.
[47] S. Arimoto, "Information-theoretical considerations on estimation problems," *Inf. Control*, vol. 19, no. 3, pp. 181–194, 1971.
[48] J. Liao, O. Kosut, L. Sankar, and F. P. Calmon, "A tunable measure for information leakage," in *Proc. IEEE ISIT*, 2018, pp. 701–705.
[49] M. D. Reid and R. C. Williamson, "Composite binary losses," *J. Mach. Learn. Res.*, vol. 11, pp. 2387–2422, 2010.
[50] P. L. Bartlett, M. I. Jordan, and J. D. Mcauliffe, "Convexity, classification, and risk bounds," *J. Amer. Stat. Assoc.*, vol. 101, no. 473, pp. 138–156, 2006.
[51] F. Liese and I. Vajda, *Convex Statistical Distances* (Teubner-Texte zur Mathematik). Leipzig, Germany: Teubner, 1987.
[52] I. Sason, "Tight bounds for symmetric divergence measures and a new inequality relating f-divergences," in *Proc. IEEE ITW*, 2015, pp. 1–5.
[53] F. Österreicher and I. Vajda, "A new class of metric divergences on probability spaces and its applicability in statistics," *Ann. Inst. Stat. Math.*, vol. 55, no. 3, pp. 639–653, 2003.
[54] A. Ruderman, M. Reid, D. García-García, and J. Petterson, "Tighter variational representations of f-divergences via restriction to probability measures," 2012, *arXiv:1206.4664*.
[55] M. I. Belghazi et al., "MINE: Mutual information neural estimation," in *Proc. 35th ICML*, vol. 80, 2018, pp. 531–540.
[56] M. Shannon, "Properties of f-divergences and f-GAN training," 2020, *arXiv:2009.00757*.
[57] S. Liu, O. Bousquet, and K. Chaudhuri, "Approximation and convergence properties of generative adversarial learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–9.
[58] D. Feldman and F. Österreicher, "A note on f-divergences," *Studia Scientiarum Mathematicarum Hungarica*, vol. 24, no. 2, pp. 191–200, 1989.
[59] K. Ji and Y. Liang, "Minimax estimation of neural net distance," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.
[60] B. Neyshabur, R. Tomioka, and N. Srebro, "Norm-based capacity control in neural networks," in *Proc. Conf. Learn. Theory*, 2015, pp. 1376–1401.
[61] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.
[62] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Proc. Conf. Learn. Theory*, 2018, pp. 297–299.
[63] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ., 2014.
[64] Z. Li et al., "A systematic survey of regularization and normalization in GANs," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, 2023.
[65] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "VEEGAN: Reducing mode collapse in GANs using implicit variational learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[66] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proc. IEEE ICCV*, 2015, pp. 3730–3738.
[67] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015, *arXiv:1506.03365*.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[69] 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," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 6629–6640.
[70] S. Nietert, Z. Goldfeld, and R. Cummings, "Outlier-robust optimal transport: Duality, structure, and statistical analysis," in *Proc. 25th Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2022, pp. 11691–11719.