Tunable Dual-Objective GANs for Stable Training

Monica Welfert [1] **Kyle Otstot** [1] **Gowtham R. Kurri** [2] **Lalitha Sankar** [1]

Abstract

In an effort to address the training instabilities of GANs, we introduce a class of dual-objective GANs with different value functions (objectives) for the generator (G) and discriminator (D). In particular, we model each objective using α-loss, a tunable classification loss, to obtain (α_D, α_G)-GANs, parameterized by $(\alpha_D, \alpha_G) \in (0, \infty]^2$. For sufficiently large number of samples and capacities for G and D, we show that the resulting non-zero sum game simplifies to minimizing an f-divergence under appropriate conditions on (α_D, α_G). We highlight the value of tuning (α_D, α_G) in alleviating training instabilities for the synthetic 2D Gaussian mixture ring, the CelebA, and the LSUN Classroom datasets.

1. 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

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

[1]Arizona State University, USA [2]IIIT Hyderabad, India. Correspondence to: Monica Welfert <mwelfert@asu.edu>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

Goodfellow *et al.* (Goodfellow et al., 2014) introduce the vanilla GAN for which

$$V_{\text{VG}}(\theta,\omega) = \mathbb{E}_{X\sim P_r}[\log D_\omega(X)] + \mathbb{E}_{X\sim P_{G_\theta}}[\log(1-D_\omega(X))]. \tag{2}$$

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 (Lin, 1991) between P_r and P_{G_θ}. This simplification is achieved, for any G_θ, by the discriminator $D_{\omega^*}(x)$ maximizing (2) which has the form

$$D_{\omega^*}(x) = \frac{p_r(x)}{p_r(x)+p_{G_\theta}(x)}, \tag{3}$$

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).

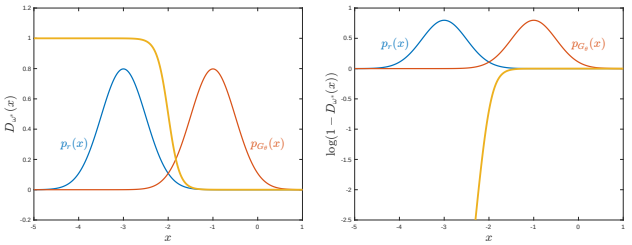
Various other GANs have been studied in the literature using different value functions, including f-divergence based GANs called f-GANs (Nowozin et al., 2016), IPM based GANs (Arjovsky et al., 2017; Sriperumbudur et al., 2012; Liang, 2018), etc. Observing that the discriminator is a classifier, recently, Kurri *et al.* (Kurri et al., 2021; 2022) 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

$$V(\theta,\omega) = \mathbb{E}_{X\sim P_r}[-\ell(1,D_\omega(X))] + \mathbb{E}_{X\sim P_{G_\theta}}[-\ell(0,D_\omega(X))]. \tag{4}$$

Using this approach, they introduce α-GAN using the tunable CPE loss α-loss (Sypherd et al., 2019; 2022), defined for $\alpha\in(0,\infty]$ as

$$\ell_\alpha(y,\hat{y}) := \frac{\alpha}{\alpha-1}\left(1-y\hat{y}^{\frac{\alpha-1}{\alpha}}-(1-y)(1-\hat{y})^{\frac{\alpha-1}{\alpha}}\right). \tag{5}$$

They show that the α-GAN formulation recovers various f-divergence based GANs including the Hellinger GAN (Nowozin et al., 2016) ($\alpha=1/2$), the vanilla GAN (Goodfellow et al., 2014) ($\alpha=1$), and the Total Variation (TV) GAN (Nowozin et al., 2016) ($\alpha=\infty$). Further, for a large enough discriminator class, the min-max optimization for α-GAN in (1) simplifies to minimizing the Arimoto divergence (Österreicher, 1996; Liese & Vajda,



(a) (b)

Figure 1. A toy example of the vanilla GAN, where the real distribution $P_r=\mathcal{N}(-2,0.5^2)$ (blue curve) and the assumed initial generated distribution $P_{G_\theta}=\mathcal{N}(2,0.5^2)$ (orange curve). (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (3). (b) A plot of the generator's saturating loss $\log(1-D_{\omega^*}(x))$.

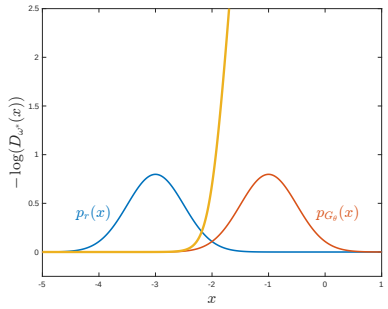


Figure 2. A plot of the vanilla GAN generator's non-saturating loss $-\log(D_{\omega^*}(x))$ for the same toy example as in Figure 1.

2006). While each of the abovementioned GANs have some advantages, they continue to suffer from one or more types of training instabilities, including vanishing/exploding gradients, mode collapse, and sensitivity to hyperparameter tuning. In (Goodfellow et al., 2014), 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 (see Figure 1). Further, 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 (see again Figure 1). To alleviate these, (Goodfellow et al., 2014) propose a *non-saturating* (NS) generator objective:

$$V_{\text{VG}}^{\text{NS}}(\theta,\omega)=\mathbb{E}_{X\sim P_{G_\theta}}[-\log D_\omega(X)]. \tag{6}$$

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 (Arjovsky & Bottou, 2017; Wiatrak et al., 2019) due to failure to converge and sensitivity to hyperparameter initialization because of large gradients (see Figure 2). While other dual-objective GANs have also been proposed (e.g., Least Squares GAN (LSGAN) (Mao et al., 2017), RényiGAN (Bhatia et al., 2021), NS f-GAN (Nowozin et al., 2016), hybrid f-GAN (Poole et al., 2016)), few have had success fully addressing training instabilities. Recent results have shown that α-loss demonstrates desirable gradient behaviors for different α values (Sypherd et al., 2022). It also assures learning robust classifiers that can reduce the confidence of D (a classifier) thereby allowing G to learn without gradient issues. To this end, we introduce a different α-loss objective for each player to address training instabilities. We propose a tunable dual-objective (α_D,α_G)-GAN, where the objective 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. Our key contributions are:

- For this non-zero sum game, we show that a Nash equilibrium exists. For appropriate (α_D,α_G) values, we derive the optimal strategies for D and G and prove that for the optimal D_{ω^*}, G minimizes an f-divergence and can therefore learn the real distribution P_r.
- Since α-GAN captures various GANs, including the vanilla GAN, it can potentially suffer from vanishing gradients due to a saturation effect. We address this by introducing a non-saturating version of the (α_D,α_G)-GAN and present its Nash equilibrium strategies for D and G.
- Finally, 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. 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.

2. Main Results

2.1. (α_D,α_G)-GAN

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

$$\begin{aligned}&V_\alpha(\theta,\omega)\\&=\mathbb{E}_{X\sim P_r}[-\ell_\alpha(1,D_\omega(X))]+\mathbb{E}_{X\sim P_{G_\theta}}[-\ell_\alpha(0,D_\omega(X))],\end{aligned} \tag{7}$$

for $\alpha=\alpha_D,\alpha_G\in(0,\infty]$. We recover the α-GAN (Kurri et al., 2021; 2022) value function when $\alpha_D=\alpha_G=\alpha$. The resulting (α_D,α_G)-GAN is given by

$$\sup_{\omega\in\Omega} V_{\alpha_D}(\theta,\omega) \tag{8a}$$

$$\inf_{\theta\in\Theta} V_{\alpha_G}(\theta,\omega). \tag{8b}$$

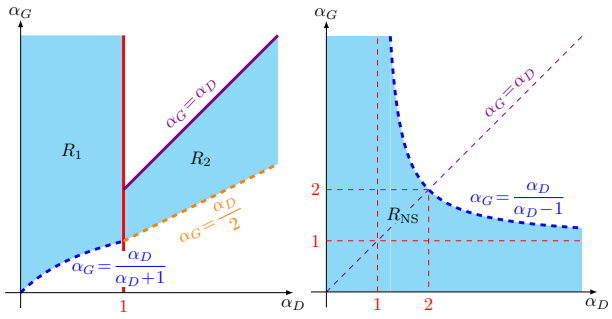


Figure 3. (a) Plot of regions for which f_{α_D,α_G} is strictly convex. (b) Plot of region for which $f^{\text{NS}}_{\alpha_D,\alpha_G}$ is strictly convex.

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 2.1. *For a fixed generator G_θ, the discriminator optimizing (8a) is given by*

$$D_{\omega^*}(x)=\frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D}+p_{G_\theta}(x)^{\alpha_D}}, \tag{9}$$

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

$$f_{\alpha_D,\alpha_G}(u)=\frac{\alpha_G}{\alpha_G-1}\left(\frac{u^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)+1}+1}{\left(u^{\alpha_D}+1\right)^{1-\frac{1}{\alpha_G}}}-2^{\frac{1}{\alpha_G}}\right), \tag{10}$$

(8b) simplifies to minimizing a non-negative symmetric f_{α_D,α_G}-divergence $D_{f_{\alpha_D,\alpha_G}}(\cdot||\cdot)$ as

$$\inf_{\theta\in\Theta} D_{f_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})+\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right), \tag{11}$$

which is minimized iff $P_{G_\theta}=P_r$ for $(\alpha_D,\alpha_G)\in(0,\infty]^2$ such that $\left(\alpha_D\le 1,\alpha_G>\frac{\alpha_D}{\alpha_D+1}\right)$ or $\left(\alpha_D>1,\frac{\alpha_D}{2}<\alpha_G\le\alpha_D\right)$.

Proof sketch. We substitute the optimal discriminator of (8a) into the objective function of (8b) and translate it into the form in (11) by finding the appropriate conditions on α_D and α_G for f_{α_D,α_G} to be a strictly convex function. Figure 3(a) illustrates the feasible (α_D,α_G)-region. A detailed proof can be found in Appendix A. See Figure 4 for a toy example illustrating the value of tuning $\alpha_D<1$ and $\alpha_G\ge 1$.

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 (7) can similarly saturate early in training, potentially causing vanishing gradients. Thus, we propose the following NS alternative to the generator's objective in (7):

$$V^{\text{NS}}_{\alpha_G}(\theta,\omega)=\mathbb{E}_{X\sim P_{G_\theta}}[\ell_{\alpha_G}(1,D_\omega(X))], \tag{12}$$

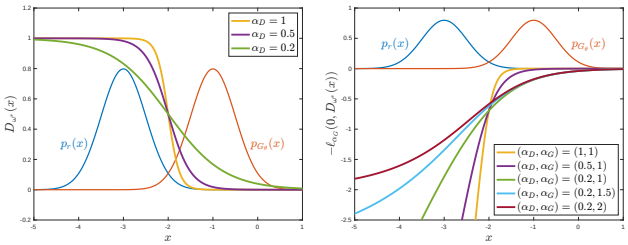


Figure 4. (a) A plot of the optimal discriminator output $D_{\omega^*}(x)$ in (9) for several values of $\alpha_D\le 1$ for the same toy example as in Figure 1. Tuning $\alpha_D<1$ decreases the confidence of the optimal discriminator D_{ω^*}. (b) A plot of the generator's loss $-\ell_{\alpha_G}(0,D_{\omega^*}(x))$ for several values of $(\alpha_D\le 1,\alpha_G\ge 1)$. Tuning $\alpha_D<1$ provides more gradient for the generator to learn early in training when the discriminator more confidently classifies the generated data as fake, thereby alleviating vanishing gradients, while tuning $\alpha_G\ge 1$ creates a smooth landscape for the generated data to descend towards the real data, alleviating exploding gradients.

thereby replacing (8b) with

$$\inf_{\theta\in\Theta} V^{\text{NS}}_{\alpha_G}(\theta,\omega). \tag{13}$$

Comparing (8b) and (13), note that the additional expectation term over P_r in (7) results in (8b) simplifying to a symmetric divergence for D_{ω^*} in (9), whereas the single term in (12) will result in (13) simplifying to an asymmetric divergence. The optimal discriminator for this NS game remains the same as in (9). The following theorem provides the solution to (13) under the assumption that the optimal discriminator can be attained.

Theorem 2.2. *For the same D_{ω^*} in (9) and the function $f^{NS}_{\alpha_D,\alpha_G}:\mathbb{R}_+\to\mathbb{R}$ defined as*

$$f^{NS}_{\alpha_D,\alpha_G}(u)=\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}-1}-\frac{u^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)}}{\left(u^{\alpha_D}+1\right)^{1-\frac{1}{\alpha_G}}}\right), \tag{14}$$

(8b) simplifies to minimizing a non-negative asymmetric $f^{NS}_{\alpha_D,\alpha_G}$-divergence $D_{f^{NS}_{\alpha_D,\alpha_G}}(\cdot||\cdot)$ as

$$\inf_{\theta\in\Theta} D_{f^{NS}_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})+\frac{\alpha_G}{\alpha_G-1}\left(1-2^{\frac{1}{\alpha_G}-1}\right), \tag{15}$$

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 2.1 and is detailed in Appendix B. Figure 3(b) illustrates the feasible (α_D,α_G)-region; in contrast to the saturating setting of Theorem 2.1, the NS setting constrains $\alpha\le 2$ when $\alpha_D=\alpha_G=\alpha$. See Figure 5 for a toy example illustrating how tuning $\alpha_D<1$ and $\alpha_G\ge 1$ can also alleviate training instabilities in the NS setting.

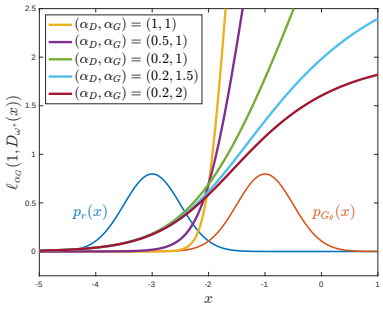


Figure 5. A plot of the generator's NS loss $\ell_{\alpha_G}(1,D_{\omega^*}(x))$ for several values of $(\alpha_D \leq 1, \alpha_G \geq 1)$. Tuning $\alpha_D < 1$ and $\alpha_G = 1$ makes the loss less convex, which can help stabilize training by decreasing sensitivity to hyperparameter initialization and alleviating mode collapse; tuning $\alpha_G > 1$ results in a quasiconvex generator objective, which can further improve training stability.

3. 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 (Radford et al., 2015), we compare against LSGANs (Mao et al., 2017), the current state-of-the-art (SOTA) dual-objective approach. While WGANs (Arjovsky et al., 2017) have also been proposed to address the training instabilities, their training methodology is distinctly different and involves a different optimizer (RMSprop), lack of batch normalization, and gradient clipping or penalty, 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) (Srivastava et al., 2017), (ii) the 64×64 Celeb-A image dataset (Liu et al., 2015), and (iii) the 112×112 LSUN Classroom dataset (Yu et al., 2015). 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.

3.1. 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 $i\in\{1,2,...,8\}$, 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.

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 (Kingma & Ba, 2014) 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 (8); **(ii)** NS (α_D,α_G)-GAN's in (8a), (13); **(iii)** LSGAN with the 0-1 binary coding scheme (see Appendix C for details).

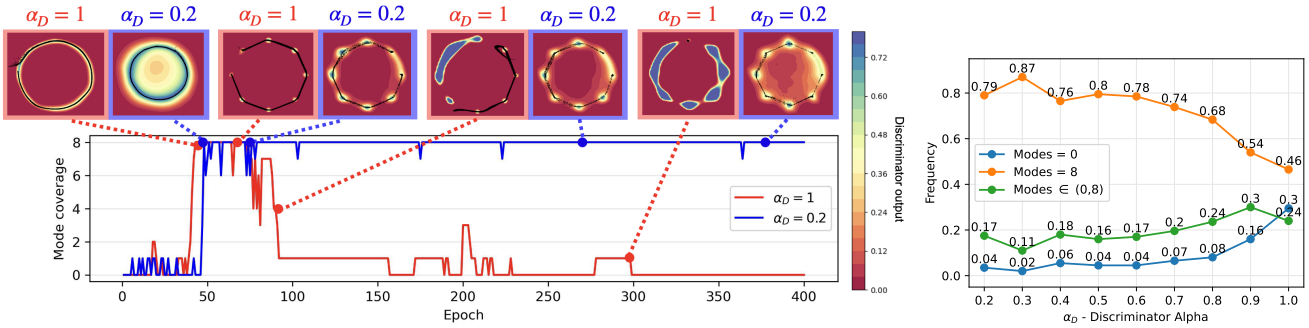
For every setting listed above, we train our models on the 2D-ring dataset for 200 random state seeds, where each seed contains different weight initializations for D and G. Ideally, a stable method will reflect similar performance across randomized initializations and also over training epochs; thus, we 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 Figure 6 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 Figure 6(a)). For example, the $(0.3,1)$-GAN success and failure rates improve to 87% and 2%, respectively. For the NS setting in Figure 7, 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 1 and 2) and NS (Table 3) are in Appendix C.

3.2. Celeb-A & LSUN Classroom

The Celeb-A dataset (Liu et al., 2015) 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 (Yu et al., 2015) 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, for a total matching the size of the true data.

For training, we employ the DCGAN architecture (Radford et al., 2015) that leverages deep convolutional neural networks (CNNs) for both D and G. In Appendix C, detailed descriptions of the D and G architectures can be found in Tables 4 and 5 for the Celeb-A and LSUN Classroom



(a) (b)

Figure 6. (a) Plot of mode coverage over epochs for (α_D,α_G)-GAN training with the **saturating** objectives in (8). 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.

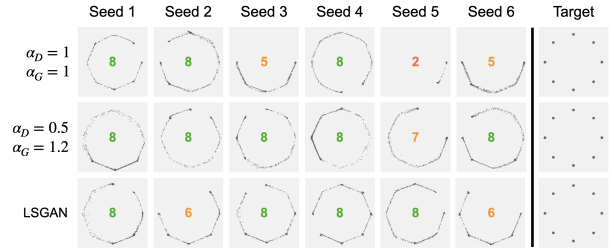
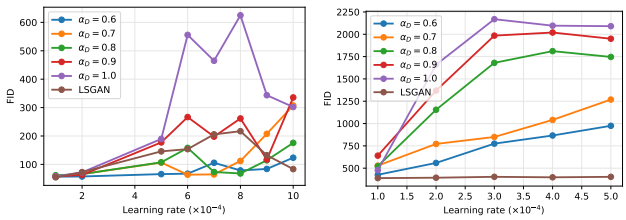


Figure 7. Generated samples from two (α_D,α_G)-GANs trained with the **NS** objectives in (8a), (13), as well as the LSGAN. We provide 6 seeds to illustrate the stability in performance for each GAN across multiple runs.



(a) (b)

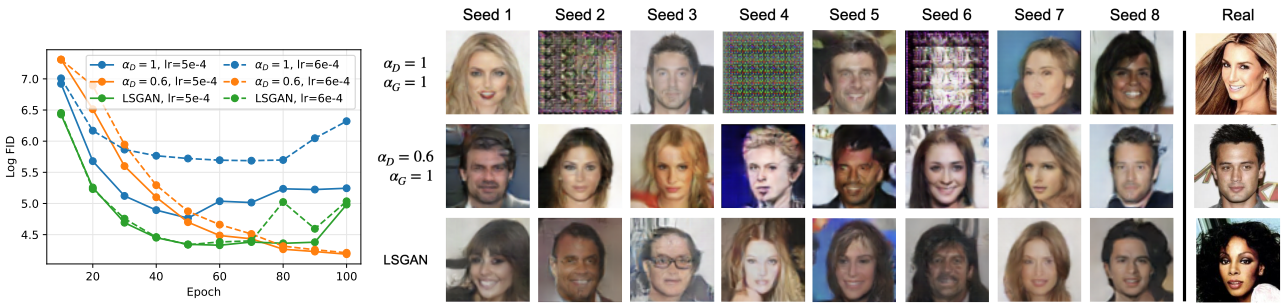
Figure 8. (a) Plot of **Celeb-A** FID scores averaged over 50 seeds vs. learning rates for 6 different GANs, trained for 100 epochs. (b) Plot of **LSUN Classroom** FID scores averaged over 50 seeds vs. learning rates for 6 different GANs, trained for 100 epochs.

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 (Heusel et al., 2017). 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*.

3.2.1. Celeb-A Results

In Figure 8(a), we examine the relationship between learning rate and FID for each GAN trained 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 8(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 Figure 9(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



(a) (b)

Figure 9. (a) Log-scale plot of **Celeb-A** FID scores over training epochs in steps of 10 up to 100 total, for three noteworthy 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.

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. 9(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.

3.2.2. LSUN Classroom Results

In Figure 8(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 Figure 8(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, Figure 8 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 C, Figure 10(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 Figure 10(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 6 in Appendix C. 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.

4. Concluding Remarks

We have introduced a dual-objective GAN formulation, focusing in particular on using the tunable α-loss, with different α values for each player's objective. Our results highlight the value of tuning α in alleviating GAN training instabilities and enhancing robustness to learning rates and training epochs, key hyperparameters whose optimal values are often unknown *a priori*. The limited range guided by theory for α_D makes tuning for the best α_D,α_G easier. An interesting problem is to evaluate if our results hold more broadly, including when the training data is noisy (Nietert et al., 2022).

References

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

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 214–223, 2017.

Bhatia, H., Paul, W., Alajaji, F., Gharesifard, B., and Burlina, P. Least kth-order and Rényi generative adversarial networks. *Neural Computation*, 33(9):2473–2510, 2021.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2672–2680, 2014.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a Nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kurri, G. R., Sypherd, T., and Sankar, L. Realizing GANs via a tunable loss function. In *IEEE Information Theory Workshop (ITW)*, pp. 1–6, 2021.

Kurri, G. R., Welfert, M., Sypherd, T., and Sankar, L. α-GAN: Convergence and estimation guarantees. In *IEEE International Symposium on Information Theory (ISIT)*, pp. 276–281, 2022.

Liang, T. How well generative adversarial networks learn distributions. *arXiv preprint arXiv:1811.03179*, 2018.

Liese, F. and Vajda, I. On divergences and informations in statistics and information theory. *IEEE Transactions on Information Theory*, 52(10):4394–4412, 2006.

Lin, J. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

Nietert, S., Goldfeld, Z., and Cummings, R. Outlier-robust optimal transport: Duality, structure, and statistical analysis. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 11691–11719, 2022.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 271–279, 2016.

Österreicher, F. On a class of perimeter-type distances of probability distributions. *Kybernetika*, 32(4):389–393, 1996.

Poole, B., Alemi, A. A., Sohl-Dickstein, J., and Angelova, A. Improved generator objectives for gans. *arXiv preprint arXiv:1612.02780*, 2016.

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

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. On the empirical estimation of integral probability metrics. *Electronic Journal of Statistics*, 6:1550–1599, 2012.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. VEEGAN: Reducing mode collapse in GANs using implicit variational learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Sypherd, T., Diaz, M., Sankar, L., and Kairouz, P. A tunable loss function for binary classification. In *IEEE International Symposium on Information Theory*, pp. 2479–2483, 2019.

Sypherd, T., Diaz, M., Cava, J. K., Dasarathy, G., Kairouz, P., and Sankar, L. A tunable loss function for robust classification: Calibration, landscape, and generalization. *IEEE Transactions on Information Theory*, 68(9):6021–6051, 2022.

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

Yu, F., Zhang, Y., Song, S., Seff, A., and Xiao, J. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*, abs/1506.03365, 2015. URL http://arxiv.org/abs/1506.03365.

A. Proof of Theorem 2.1

The proof to obtain (9) is the same as that for (Kurri et al., 2021)[Theorem 2], where $\alpha=\alpha_D$. The generator's optimization problem in (8b) with the optimal discriminator in (9) can be written as $\inf_{\theta\in\Theta}V_{\alpha_G}(\theta,\omega^*)$, where

$$\begin{aligned}
V_{\alpha_G}(\theta,\omega^*)&=\frac{\alpha_G}{\alpha_G-1}\left[\int_{\mathcal{X}}\left(p_r(x)D_{\omega^*}(x)^{\frac{\alpha_G-1}{\alpha_G}}+p_{G_\theta}(x)(1-D_{\omega^*}(x))^{\frac{\alpha_G-1}{\alpha_G}}\right)dx-2\right]\\
&=\frac{\alpha_G}{\alpha_G-1}\left[\int_{\mathcal{X}}\left(p_r(x)\left(\frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D}+p_{G_\theta}(x)^{\alpha_D}}\right)^{\frac{\alpha_G-1}{\alpha_G}}+p_{G_\theta}(x)\left(\frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D}+p_{G_\theta}(x)^{\alpha_D}}\right)^{\frac{\alpha_G-1}{\alpha_G}}\right)dx-2\right]\\
&=\frac{\alpha_G}{\alpha_G-1}\left(\int_{\mathcal{X}}p_{G_\theta}(x)\left(\frac{(p_r(x)/p_{G_\theta}(x))^{\alpha_D(1-1/\alpha_G)+1}+1}{((p_r(x)/p_{G_\theta}(x))^{\alpha_D}+1)^{1-1/\alpha_G}}\right)dx-2\right)\\
&=\int_{\mathcal{X}}p_{G_\theta}(x)f_{\alpha_D,\alpha_G}\left(\frac{p_r(x)}{p_{G_\theta}(x)}\right)dx+\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right),
\end{aligned}$$

where f_{α_D,α_G} is as defined in (10). Note that if f_{α_D,α_G} is strictly convex, the first term in the last equality above equals an f-divergence which is minimized if and only if $P_r=P_{G_\theta}$. Define the regions R_1 and R_2 as follows:

$$R_1:=\left\{(\alpha_D,\alpha_G)\in(0,\infty]^2\,\middle|\,\alpha_D\leq 1,\alpha_G>\frac{\alpha_D}{\alpha_D+1}\right\}$$

and

$$R_2:=\left\{(\alpha_D,\alpha_G)\in(0,\infty]^2\,\middle|\,\alpha_D>1,\frac{\alpha_D}{2}<\alpha_G\leq\alpha_D\right\}.$$

In order to prove that f_{α_D,α_G} is strictly convex for $(\alpha_D,\alpha_G)\in R_1\cup R_2$, we take its second derivative, which yields

$$f''_{\alpha_D,\alpha_G}(u)=A_{\alpha_D,\alpha_G}(u)\left[(\alpha_G+\alpha_D\alpha_G-\alpha_D)\left(u+u^{\alpha_D+\frac{\alpha_D}{\alpha_G}}\right)+(\alpha_G-\alpha_D\alpha_G)\left(u^{\frac{\alpha_D}{\alpha_G}}+u^{\alpha_D+1}\right)\right],\tag{16}$$

where

$$A_{\alpha_D,\alpha_G}(u)=\frac{\alpha_D}{\alpha_G}u^{\alpha_D-\frac{\alpha_D}{\alpha_G}-2}(1+u^{\alpha_D})^{\frac{1}{\alpha_G}-3}.\tag{17}$$

Note that $A_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$ and $\alpha_D,\alpha_G\in(0,\infty]$. Therefore, in order to ensure $f''_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$ it is sufficient to have

$$\alpha_G+\alpha_D\alpha_G-\alpha_D>\alpha_G(\alpha_D-1)B_{\alpha_D,\alpha_G}(u),\tag{18}$$

where

$$B_{\alpha_D,\alpha_G}(u)=\frac{u^{\frac{\alpha_D}{\alpha_G}}+u^{\alpha_D+1}}{u+u^{\alpha_D+\frac{\alpha_D}{\alpha_G}}}\tag{19}$$

for $u>0$. Since $B_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$, the sign of the RHS of (18) is determined by whether $\alpha_D\leq 1$ or $\alpha_D>1$. We look further into these two cases in the following:

Case 1: $\alpha_D\leq 1$. Then $\alpha_G(\alpha_D-1)B_{\alpha_D,\alpha_G}(u)\leq 0$ for all $u>0$ and $(\alpha_D,\alpha_G)\in(0,\infty]^2$. Therefore, we need

$$\alpha_G(1+\alpha_D)-\alpha_D>0\Leftrightarrow\alpha_G>\frac{\alpha_D}{\alpha_D+1}.\tag{20}$$

Case 2: $\alpha_D>1$. Then $\alpha_G(\alpha_D-1)B_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$ and $(\alpha_D,\alpha_G)\in(0,\infty]^2$. In order to obtain conditions on α_D and α_G, we determine the monotonicity of B_{α_D,α_G} by finding its first derivative as follows:

$$B'_{\alpha_D,\alpha_G}(u)=\frac{(\alpha_G-\alpha_D)(u^{2\alpha_D}-1)+\alpha_D\alpha_G\left(u^{\alpha_D-\frac{\alpha_D}{\alpha_G}+1}-u^{\alpha_D+\frac{\alpha_D}{\alpha_G}-1}\right)}{\alpha_G u^{-\frac{\alpha_D}{\alpha_G}}\left(u+u^{\alpha_D+\frac{\alpha_D}{\alpha_G}}\right)^2}.$$

Since the denominator of B'_{α_D,α_G} is positive for all $u>0$ and $(\alpha_D,\alpha_G)\in(0,\infty]^2$, we just need to check the sign of the numerator.

Case 2a: $\alpha_D>\alpha_G$. For $u\in(0,1)$,

$$u^{2\alpha_D}-1<0 \quad \text{and} \quad u^{\alpha_D-\frac{\alpha_D}{\alpha_G}+1}-u^{\alpha_D+\frac{\alpha_D}{\alpha_G}-1}>0,$$

so $B'_{\alpha_D,\alpha_G}(u)>0$. For $u>1$,

$$u^{2\alpha_D}-1>0 \quad \text{and} \quad u^{\alpha_D-\frac{\alpha_D}{\alpha_G}+1}-u^{\alpha_D+\frac{\alpha_D}{\alpha_G}-1}<0,$$

so $B'_{\alpha_D,\alpha_G}(u)<0$. For $u=1$, $B'_{\alpha_D,\alpha_G}(u)=0$. Hence, B'_{α_D,α_G} is strictly increasing for $u\in(0,1)$ and strictly decreasing for $u\geq 1$. Therefore, B_{α_D,α_G} attains a maximum value of 1 at $u=1$. This means B_{α_D,α_G} is bounded, i.e. $B_{\alpha_D,\alpha_G}\in(0,1]$ for all $u>0$. Thus, in order for (18) to hold, it suffices to ensure that

$$\alpha_G+\alpha_D\alpha_G-\alpha_D>\alpha_G(\alpha_D-1)\Leftrightarrow\alpha_G>\frac{\alpha_G}{2}. \tag{21}$$

Case 2b: $\alpha_D<\alpha_G$. For $u\in(0,1)$, $u^{2\alpha_D}-1<0$ and $u^{\alpha_D-\frac{\alpha_D}{\alpha_G}+1}-u^{\alpha_D+\frac{\alpha_D}{\alpha_G}-1}<0$, so $B'_{\alpha_D,\alpha_G}(u)<0$. For $u>1$, $u^{2\alpha_D}-1>0$ and $u^{\alpha_D-\frac{\alpha_D}{\alpha_G}+1}-u^{\alpha_D+\frac{\alpha_D}{\alpha_G}-1}>0$, so $B'_{\alpha_D,\alpha_G}(u)>0$. Hence, B'_{α_D,α_G} is strictly decreasing for $u\in(0,1)$ and strictly increasing for $u\geq 1$. Therefore, B_{α_D,α_G} attains a minimum value of 1 at $u=1$. This means that B_{α_D,α_G} is not bounded above, so it is not possible to satisfy (18) without restricting the domain of B_{α_D,α_G}.

Thus, for $(\alpha_D,\alpha_G)\in R_1\cup R_2$,

$$V_{\alpha_G}(\theta,\omega^*)=D_{f_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})+\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right).$$

This yields (11). Note that $D_{f_{\alpha_D,\alpha_G}}(P||Q)$ is symmetric since

$$\begin{aligned}
D_{f_{\alpha_D,\alpha_G}}(Q||P)&=\int_{\mathcal{X}}p(x)f_{\alpha_D,\alpha_G}\left(\frac{q(x)}{p(x)}\right)dx\\
&=\frac{\alpha_G}{\alpha_G-1}\left(\int_{\mathcal{X}}p(x)\left(\frac{(p(x)/q(x))^{-\alpha_D\left(1-\frac{1}{\alpha_G}\right)-1}+1}{((p(x)/q(x))^{-\alpha_D}+1)^{1-\frac{1}{\alpha_G}}}\right)dx-2^{\frac{1}{\alpha_G}}\right)\\
&=\frac{\alpha_G}{\alpha_G-1}\left(\int_{\mathcal{X}}p(x)\left(\frac{q(x)/p(x)+(p(x)/q(x))^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)}}{(1+(p(x)/q(x))^{\alpha_D})^{1-\frac{1}{\alpha_G}}}\right)dx-2^{\frac{1}{\alpha_G}}\right)\\
&=\frac{\alpha_G}{\alpha_G-1}\left(\int_{\mathcal{X}}q(x)\left(\frac{1+(p(x)/q(x))^{\alpha_D\left(1-\frac{1}{\alpha_G}\right)}}{(1+(p(x)/q(x))^{\alpha_D})^{1-\frac{1}{\alpha_G}}}\right)dx-2^{\frac{1}{\alpha_G}}\right)\\
&=D_{f_{\alpha_D,\alpha_G}}(P||Q).
\end{aligned}$$

Since f_{α_D,α_G} is strictly convex and $f_{\alpha_D,\alpha_G}(1)=0$, $D_{f_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})\geq 0$ with equality if and only if $P_r=P_{G_\theta}$. Thus, we have $V_{\alpha_G}(\theta,\omega^*)\geq\frac{\alpha_G}{\alpha_G-1}\left(2^{\frac{1}{\alpha_G}}-2\right)$ with equality if and only if $P_r=P_{G_\theta}$.

B. Proof of Theorem 2.2

The generator's optimization problem in (8b) with the optimal discriminator in (9) can be written as $\inf_{\theta\in\Theta}V^{\text{NS}}_{\alpha_G}(\theta,\omega^*)$, where

$$\begin{aligned}
V^{\text{NS}}_{\alpha_G}(\theta,\omega^*)&=\frac{\alpha_G}{\alpha_G-1}\left[1-\int_{\mathcal{X}}\left(p_{G_\theta}(x)D_{\omega^*}(x)^{\frac{\alpha_G-1}{\alpha_G}}\right)dx\right]\\
&=\frac{\alpha_G}{\alpha_G-1}\left[1-\int_{\mathcal{X}}p_{G_\theta}(x)\left(\frac{p_r(x)^{\alpha_D}}{p_r(x)^{\alpha_D}+p_{G_\theta}(x)^{\alpha_D}}\right)^{\frac{\alpha_G-1}{\alpha_G}}dx\right]\\
&=\frac{\alpha_G}{\alpha_G-1}\left[1-\int_{\mathcal{X}}p_{G_\theta}(x)\frac{(p_r(x)/p_{G_\theta}(x))^{\alpha_D(1-1/\alpha_G)}}{((p_r(x)/p_{G_\theta}(x))^{\alpha_D}+1)^{1-1/\alpha_G}}dx\right]
\end{aligned}$$

$$=\int_{\mathcal{X}} p_{G_\theta}(x) f^{\text{NS}}_{\alpha_D,\alpha_G}\left(\frac{p_r(x)}{p_{G_\theta}(x)}\right)dx+\frac{\alpha_G}{\alpha_G-1}\left(1-2^{\frac{1}{\alpha_G}-1}\right),$$

where $f^{\text{NS}}_{\alpha_D,\alpha_G}$ is as defined in (14). In order to prove that $f^{\text{NS}}_{\alpha_D,\alpha_G}$ is strictly convex for $(\alpha_D,\alpha_G)\in R_{\text{NS}}=\{(\alpha_D,\alpha_G)\in(0,\infty]^2 | \alpha_D>\alpha_G(\alpha_D-1)\}$, we take its second derivative, which yields

$$f''_{\alpha_D,\alpha_G}(u)=A_{\alpha_D,\alpha_G}(u)\left[(\alpha_G-\alpha_D\alpha_G+\alpha_D)+\alpha_G(1+\alpha_D)u^{\alpha_D}\right], \tag{22}$$

where A_{α_D,α_G} is defined as in (17). Since $A_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$ and $(\alpha_D,\alpha_G)\in(0,\infty]^2$, to ensure $f''_{\alpha_D,\alpha_G}(u)>0$ for all $u>0$ it suffices to have

$$\frac{\alpha_G-\alpha_D\alpha_G+\alpha_D}{\alpha_G(1+\alpha_D)}>-u^{\alpha_D}$$

for all $u>0$. This is equivalent to

$$\frac{\alpha_G-\alpha_D\alpha_G+\alpha_D}{\alpha_G(1+\alpha_D)}>0,$$

which results in the condition

$$\alpha_D>\alpha_G(\alpha_D-1)$$

for $(\alpha_D,\alpha_G)\in(0,\infty]^2$. Thus, for $(\alpha_D,\alpha_G)\in R_{\text{NS}}$,

$$V^{\text{NS}}_{\alpha_G}(\theta,\omega^*)=D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})+\frac{\alpha_G}{\alpha_G-1}\left(1-2^{\frac{1}{\alpha_G}-1}\right).$$

This yields (15). Note that $D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(P||Q)$ is not symmetric since $D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(P||Q)\neq D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(Q||P)$. Since $f^{\text{NS}}_{\alpha_D,\alpha_G}$ is strictly convex and $f^{\text{NS}}_{\alpha_D,\alpha_G}(1)=0$, $D_{f^{\text{NS}}_{\alpha_D,\alpha_G}}(P_r||P_{G_\theta})\geq 0$ with equality if and only if $P_r=P_{G_\theta}$. Thus, we have $V^{\text{NS}}_{\alpha_G}(\theta,\omega^*)\geq \frac{\alpha_G}{\alpha_G-1}\left(1-2^{\frac{1}{\alpha_G}-1}\right)$ with equality if and only if $P_r=P_{G_\theta}$.

C. Additional Experimental Results

C.1. Brief Overview of LSGAN

The Least Squares GAN (LSGAN) is a dual-objective min-max game introduced in (Mao et al., 2017). The LSGAN objective functions, as the name suggests, involve squared loss functions for D and G which are written as

$$\begin{aligned}&\inf_{\omega\in\Omega}\frac{1}{2}\left(\mathbb{E}_{X\sim P_r}[(D_\omega(X)-b)^2]+\mathbb{E}_{X\sim P_{G_\theta}}[(D_\omega(X)-a)^2]\right)\\&\inf_{\theta\in\Theta}\frac{1}{2}\left(\mathbb{E}_{X\sim P_r}[(D_\omega(X)-c)^2]+\mathbb{E}_{X\sim P_{G_\theta}}[(D_\omega(X)-c)^2]\right).\end{aligned} \tag{23}$$

For appropriately chosen values of the parameters a, b, and c, (23) reduces to minimizing the Pearson χ^2-divergence between $P_r+P_{G_\theta}$ and $2P_{G_\theta}$. As done in the original paper (Mao et al., 2017), we use $a=0$, $b=1$ and $c=1$ for our experiments to make fair comparisons. The authors refer to this choice of parameters as the 0-1 binary coding scheme.

C.2. 2D Gaussian Mixture Ring

In Tables 1 and 2, we report the success (8/8 mode coverage) and failure (0/8 mode coverage) rates over 200 seeds for a grid of (α_D,α_G) combinations for the *saturating* setting. Compared to the vanilla GAN performance, we find that tuning α_D below 1 leads to a greater success rate and lower failure rate. However, in this saturating loss setting, we find that tuning α_G away from 1 has no significant impact on GAN performance.

In Table 3, we detail the success rates for the NS setting. We note that for this dataset, no failures, and therefore, no vanishing/exploding gradients, occurred in the NS setting. In particular, we find that the $(0.5,1.2)$-GAN doubles the success rate of the vanilla $(1,1)$-GAN, which is more susceptible to mode collapse as illustrated in Figure 7. We also find that LSGAN achieves a success rate of 32.5%, which is greater than vanilla GAN but less than the best-performing (α_D,α_G)-GAN.

Table 1. Success rates for 2D-ring with the saturating (α_D,α_G)-GAN over 200 seeds, with top 4 combinations emboldened.

% OF SUCCESS (8/8 MODES)		α_D					
		0.5	0.6	0.7	0.8	0.9	1.0
α_G	0.9	73	79	69	60	46	34
	1.0	**80**	**79**	74	68	54	47
	1.1	**79**	77	68	70	59	47
	1.2	**75**	74	71	65	57	46

Table 2. Failure rates for 2D-ring with the saturating (α_D,α_G)-GAN over 200 seeds, with top 3 combinations emboldened.

% OF FAILURE (0/8 MODES)		α_D					
		0.5	0.6	0.7	0.8	0.9	1.0
α_G	0.9	11	10	12	13	29	49
	1.0	**5**	**5**	7	8	16	30
	1.1	7	9	13	12	13	26
	1.2	9	**5**	9	12	17	31

Table 3. Success rates for 2D-ring with the NS (α_D,α_G)-GAN over 200 seeds, with top 5 combinations emboldened.

% OF SUCCESS (8/8 MODES)		α_D							
		0.5	0.6	0.7	0.8	0.9	1.0	1.1	1.2
α_G	0.8	35	24	19	19	14	16	18	10
	0.9	**39**	37	19	22	16	20	19	21
	1.0	34	35	29	28	26	22	20	32
	1.1	**40**	36	31	22	24	15	23	25
	1.2	**45**	38	34	25	26	28	20	22
	1.3	**44**	**39**	26	28	28	25	31	29

C.3. Celeb-A & LSUN Classroom

The discriminator and generator architectures used for the Celeb-A and LSUN Classroom datasets are described in Tables 4 and 5 respectively. Each architecture consists of four CNN layers, with parameters such as kernel size (i.e., size of the filter, denoted as "Kernel"), stride (the amount by which the filter moves), and the activation functions applied to the layer outputs. Zero padding is also assumed. In both tables, "BN" represents batch normalization, a technique that normalizes the inputs to each layer using a batch of samples during model training. Batch normalization is commonly employed in deep learning to prevent cumulative floating point errors and overflows, and to ensure that all features remain within a similar range. This technique serves as a computational tool to address vanishing and/or exploding gradients.

In Table 6, we collate the FID results for both datasets as a function of the learning rates. This table captures the percentage (out of 50 seeds) of FID scores below a desired threshold, which is 80 for the CELEB-A dataset and 800 for the LSUN Classroom dataset.

We first focus on the CELEB-A dataset: Table 6 demonstrates that for a learning rate of 1×10^{-4}, all GANs (vanilla, different (α_D,α_G)-GANs, and LSGANs) achieve an FID score below 80 at least 93% of the time. However, the instability of vanilla GAN is also evident in Table 6, where for a slightly higher learning rate of 6×10^{-4}, the $(1,1)$-GAN achieves an FID score below 80 only 60% of the time whereas at least one $(\alpha_D,\alpha_G=1)$-GAN consistently performs better than 76% over all chosen learning rates. We observe that tuning α_D below 1 contributes to stabilizing the FID scores over the 50 seeds while maintaining relatively low scores on average. This stability is emphasized in Table 6, in particular for the $(0.7,1)$-GAN, as it achieves an FID score below 80 at least 80% of the time for 7 out of the 10 the learning rates.

Table 6 also illustrates similar results for the LSUN Classroom dataset. However, increasing it to 2×10^{-4} leads to instability in the vanilla $(1,1)$-GAN across 50 seeds.

Table 4. Discriminator and generator architectures for Celeb-A. The final sigmoid activation layer is removed for the LSGAN discriminator.

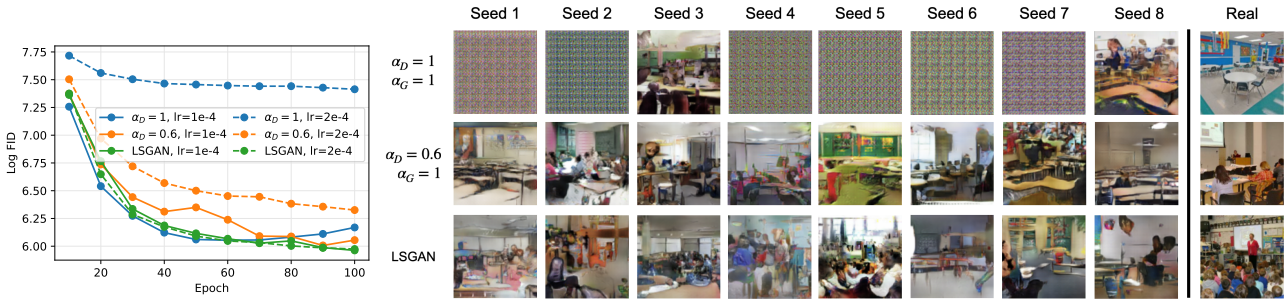
Discriminator						Generator					
Layer	Output size	Kernel	Stride	BN	Activation	Layer	Output size	Kernel	Stride	BN	Activation
Input	3×64×64				Leaky ReLU	Input	100×1×1				ReLU
Convolution	64×32×32	4×4	2	Yes	Leaky ReLU	ConvTranspose	512×4×4	4×4	2	Yes	ReLU
Convolution	128×16×16	4×4	2	Yes	Leaky ReLU	ConvTranspose	256×8×8	4×4	2	Yes	ReLU
Convolution	256×8×8	4×4	2	Yes	Leaky ReLU	ConvTranspose	128×16×16	4×4	2	Yes	ReLU
Convolution	512×4×4	4×4	2	Yes	Leaky ReLU	ConvTranspose	64×32×32	4×4	2	Yes	ReLU
Convolution	1×1×1	4×4	2		Sigmoid	ConvTranspose	3×64×64	4×4	2		Tanh

Table 5. Discriminator and generator architectures for LSUN Classroom. The final sigmoid activation layer is removed for the LSGAN discriminator.

Discriminator						Generator					
Layer	Output size	Kernel	Stride	BN	Activation	Layer	Output size	Kernel	Stride	BN	Activation
Input	3×112×112				Leaky ReLU	Input	100×1×1				ReLU
Convolution	64×56×56	4×4	2	Yes	Leaky ReLU	ConvTranspose	512×7×7	7×7	2	Yes	ReLU
Convolution	128×28×28	4×4	2	Yes	Leaky ReLU	ConvTranspose	256×14×14	4×4	2	Yes	ReLU
Convolution	256×14×14	4×4	2	Yes	Leaky ReLU	ConvTranspose	128×28×28	4×4	2	Yes	ReLU
Convolution	512×7×7	4×4	2	Yes	Leaky ReLU	ConvTranspose	64×56×56	4×4	2	Yes	ReLU
Convolution	1×1×1	7×7	2		Sigmoid	ConvTranspose	3×112×112	4×4	2		Tanh

Table 6. Percentage out of 50 seeds of FID scores below 80 (Celeb-A) or 800 (LSUN Classroom) for each combination of (α_D,α_G)-GAN and learning rate, trained for 100 epochs. Best results for each dataset and learning rate are **emboldened**.

GAN	Celeb-A								LSUN Classroom				
	Learning rate ($\times 10^{-4}$)												
(α_D,α_G)	1	2	5	6	7	8	9	10	1	2	3	4	5
(1,1)	**100**	93.2	82.6	59.5	58.5	39.0	53.7	54.8	92.0	36.2	12.5	13.0	12.2
(0.9,1)	**100**	95.2	78.3	72.3	81.4	66.7	74.4	46.5	76.0	53.1	22.2	17.0	22.2
(0.8,1)	97.8	**97.6**	**88.9**	82.2	81.4	72.1	68.4	75.6	88.5	60.8	36.2	27.9	29.2
(0.7,1)	**100**	90.7	**88.9**	**91.5**	**86.4**	**81.2**	67.6	**80.0**	90.2	80.4	78.4	67.4	55.1
(0.6,1)	97.8	93.0	88.4	76.6	84.6	75.6	**76.9**	69.2	**95.7**	**90.4**	**85.1**	**78.3**	**66.0**



(a) (b)

Figure 10. (a) Log-scale plot of **LSUN Classroom** FID scores over training epochs in steps of 10 up to 100 total, for three noteworthy GANs– (1,1)-GAN (vanilla), (0.6,1)-GAN, and LSGAN– and for two similar learning rates– 1×10^{-4} and 2×10^{-4}. Results show that the vanilla GAN performance is very sensitive to learning rate choice as the difference between training with 1×10^{-4} and 2×10^{-4} is drastic. On the other hand, the other two GANs achieve consistently lower FIDs, with the LSGAN performing the best. (b) Generated LSUN Classroom images from the same three GANs over 8 seeds when trained for 100 epochs with a learning rate of 2×10^{-4}. These samples show that the vanilla (1,1)-GAN training fails for most of seeds while the other two GANs perform fairly well across all seeds, thus exhibiting robustness to random weight initializations.