GANs Settle Scores!

Siddarth Asokan[1]
Robert Bosch Center for Cyber-Physical Systems
Indian Institute of Science (IISc)
Bengaluru - 560012, India
siddartha@iisc.ac.in

Nishanth Shetty
Department of Electrical Engineering
Indian Institute of Science
Bengaluru - 560012, India
nishanths@iisc.ac.in

Aadithya Srikanth[2]
Department of Electronics and Communication
P.E.S. University
Bengaluru - 560085, India
srikanth.aadithya@gmail.com

Chandra Sekhar Seelamantula
Department of Electrical Engineering
Indian Institute of Science
Bengaluru - 560012, India
css@iisc.ac.in

Abstract

Generative adversarial networks (GANs) comprise a generator, trained to learn the underlying distribution of the desired data, and a discriminator, trained to distinguish real samples from those output by the generator. A majority of GAN literature focuses on understanding the optimality of the discriminator through integral probability metric (IPM) or divergence based analysis. In this paper, we propose a unified approach to analyzing the generator optimization through variational approach. In f-divergence-minimizing GANs, we show that the optimal generator is the one that matches the score of its output distribution with that of the data distribution, while in IPM GANs, we show that this optimal generator matches *score-like* functions, involving the flow-field of the kernel associated with a chosen IPM constraint space. Further, the IPM-GAN optimization can be seen as one of smoothed score-matching, where the scores of the data and the generator distributions are convolved with the kernel associated with the constraint. The proposed approach serves to unify score-based training and existing GAN flavors, leveraging results from normalizing flows, while also providing explanations for empirical phenomena such as the stability of non-saturating GAN losses. Based on these results, we propose novel alternatives to f-GAN and IPM-GAN training based on score and flow matching, and discriminator-guided Langevin sampling.

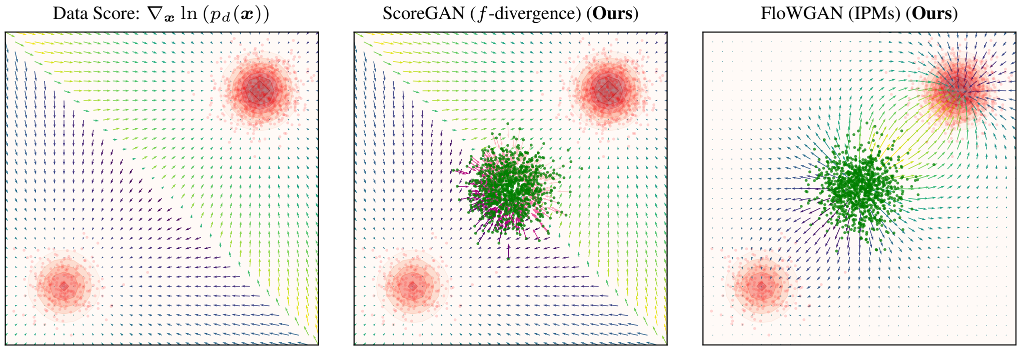
1 Introduction

Generative modeling refers to the process of learning the underlying distribution of a given dataset, either with the aim of evaluating the density, or generating new unseen samples from the underlying distribution. Generative adversarial networks (GANs, Goodfellow et al. (2014)) have become one of the most popular frameworks for image generation, owing to lower sampling times and state-of-the-art sample quality (Karras et al., 2020, 2021; Sauer et al., 2022). GANs are a two-player game between a generator network $G\colon \mathbb{R}^d \to \mathbb{R}^n$ and a discriminator network $D\colon \mathbb{R}^n \to \mathbb{R}$. In most GAN settings, $d \leq n$. The generator accepts a noise vector $\boldsymbol{z} \sim p_z$; $\boldsymbol{z} \in \mathbb{R}^d$, typically Gaussian or uniform distributed, and transforms it into a *fake* sample $G(\boldsymbol{z})$, with the push-forward distribution

[1] Corresponding Author.

[2] Work done during an internship at the Spectrum Lab, Department of Electrical Engineering, Indian Institute of Science, Bengaluru - 560012.

Preprint. Under review.


Data Score: $\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)$

ScoreGAN (f-divergence) (**Ours**)

FloWGAN (IPMs) (**Ours**)

Figure 1: (● Color online) The loss landscape of the proposed ScoreGAN and FloWGAN variants, juxtaposed against the *(Stein) score* of the target data, for a Gaussian mixture $p_d = \frac{1}{5}\mathcal{N}(-5\mathbf{1}_2, \mathbb{I}_2) + \frac{4}{5}\mathcal{N}(5\mathbf{1}_2, \mathbb{I}_2)$. The GAN generator distribution, p_g, is the standard normal Gaussian. The f-divergence GAN is an instance of ScoreGANs, where the generator is trained to minimize the error between the the score of p_g (shown in pink) and p_d. All integral probability metric (IPM) minimizing GANs are instances of FloWGANs that minimize the gradient field of the density difference $p_d - p_g$ convolved with a kernel κ. The repulsive nature of the gradient field in FloWGANs prevents vanishing gradients.

$p_g = G_{\#}(p_z)$. The discriminator accepts an input drawn either from the target distribution, $\boldsymbol{x} \sim p_d$; $\boldsymbol{x} \in \mathbb{R}^n$, or from the output of a generator, and learns *real versus fake* classifier. The objective is to learn the *optimal generator* — one that can generate realistic samples.

GANs Losses: In the standard GAN (SGAN, Goodfellow et al. (2014)), and the least-squares GAN (LSGAN, Mao et al. (2017)) formulations, the discriminator models a chosen *divergence* metric between the target and generator distributions, while the generator network is trained to minimize this divergence. For example, the Jensen-Shannon divergence in SGAN, or the Pearson-χ^2 divergence in LSGAN. Nowozin et al. (2016) generalized the formulation to account for any f-divergence, while Uehara et al. (2016) consider extension to Bregman divergences as well. Owing to the training instability of divergence-minimizing GANs on non-overlapping distributions, Arjovsky & Bottou (2017) proposed integral probability metrics (IPM) as a viable alternative. In IPM-GANs, the discriminator performs the role of a *critic*, and approximates the IPM. The choice of the metric constrains the class of functions from which the critic is drawn. For example, in Wasserstein GAN (WGAN), Arjovsky et al. (2017) consider Lipschitz-1 critics. In practice, the Lipschitz constraint is approximated through a gradient penalty enforced on the discriminator (Gulrajani et al., 2017). Mroueh et al. (2018), Adler & Lunz (2018), and Asokan & Seelamantula (2023b) consider discriminator functions drawn from Sobolev spaces, with a corresponding penalty on the energy in the gradient. Gretton et al. (2012) showed that the minimization of IPM losses can be equivalently solved through the minimization of kernel-based statistics in a reproducing-kernel Hilbert space (RHKS). Maximum-mean discrepancy GANs (MMD-GANs) (Li et al., 2017; Bińkowski et al., 2018) and Coulomb GAN (Unterthiner et al., 2018) are examples of kernel-based GANs.

Optimality in GANs: A major research focus in GAN optimization is on the optimality of the discriminator function. While Goodfellow et al. (2014) and Mao et al. (2017) considered a pointwise optimization of the discriminator, Mroueh et al. (2018); Yi et al. (2023) and Asokan & Seelamantula (2023a) consider a functional approach, and derived differential equations that govern the optimal discriminator, given the generator. Along another vertical, Pinetz et al. (2018), Stanczuk et al. (2021) and Korotin et al. (2022) showed that, in practical gradient-descent-based training, the optimal discriminator is not attained. However, a similar in-depth analysis of the optimal generator in GANs is lacking. Existing approaches rely on an empirical evaluation of the generator (Zhu et al., 2020), analyze the convergence considering infinite-width network (infinite number of nodes per layer) approximations (Franceschi et al., 2022), or derive constraints on the generator when the generator and discriminator are jointly optimized (Liang, 2021). While in most scenarios, the generator can be linked to minimizing the chosen divergence or IPM, the actual functional optimization has not been thoroughly explored. **What does the closed-form optimization of the generator lead to in GANs?** In this paper, this is the gap in literature that we seek to answer.

1.1 Our Contribution

We consider the alternating optimization in various divergence-minimizing and IPM-based GAN formulations, retaining the functional form of the optimal discriminator, and analyze the generator loss function through the lens of *variational calculus*. Considering the family of f-GANs, we show that minimizing the f-divergence results in an optimal generator which, given the optimal discriminator, minimizes the error between the score (the gradient of the log-probability) of the target data distribution, and the score of the generator's push-forward distribution. This permits interpreting the f-GANs as performing score-matching. Owing to the score-matching link, our divergence-minimizing approach is entitled *ScoreGAN*.

Considering gradient-regularized Wasserstein GAN losses, we show that the optimal generator is the one that minimizes a *smoothed* score-matching difference term, where the scores are conditioned by means of the kernel associated with the RKHS from which the IPM discriminator is drawn, akin to noise conditioned score networks (NCSN) (Song & Ermon, 2019). Futher, we show that, in IPM GANs, the *smoothed score-matching* formulation is equivalent to one of minimizing a flow induced by the gradient field of a kernel. The kernel-flow based formulation is referred to as *FloWGAN*. These results can be seen as a generalization of Sobolev descent (Mroueh et al., 2019), MMD-Flows (Arbel et al., 2019) and MonoFlows (Yi et al., 2023). A visualization of the generator loss landscape in ScoreGANs and FloWGANs, juxtaposed with the score of the data is presented in Figure 1. The results showcase a fundamental connection between the various GAN, score-based and flow-based generative models.

As a proof of concept, we validate training GANs with score-matching and flow-minimizing costs, using results from normalizing flows (Papamakarios et al., 2021) and NCSNs (Song & Ermon, 2019) on unimodal and multimodal Gaussians, and latent-space matching on image, akin to Wasserstein autoencoders (Tolstikhin et al., 2018) and latent diffusion (Rombach et al., 2022). To demonstrate the interplay between GANs and score-based approaches, we also present experiments on generating images by replacing the score network with the gradient of the kernel in a Langevin sampler.

2 Background on Scores and Flows

Score Matching: Score matching was originally proposed by Hyvärinen (2005) in the context of independent component analysis. Consider the underlying distribution of the data to be modeled, $p_d(\boldsymbol{x})$. The *(Stein) score* (Liu et al., 2016) is the gradient of logarithm of the density function with respect to the data itself, $\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)$. It generates a vector field that points in the direction where the data density grows most steeply. In score matching, the score can be approximated by a parametric function $S_\phi^{\mathcal{D}}(\boldsymbol{x})$ obtained by minimizing the Fisher divergence (Cover & Thomas, 2006):

$$\mathscr{F}(S_\phi^{\mathcal{D}}, p_d) = \frac{1}{2} \underset{\boldsymbol{x}\sim p_d}{\mathbb{E}} \left[\left\| S_\phi^{\mathcal{D}}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right) \right\|_2^2 \right]. \tag{1}$$

The output of the trained network is used to generate samples through annealed Langevin dynamics in noise-conditioned score networks (NCSN) (Song & Ermon, 2019). Recent approaches aim at either improving the approximation quality of the score network (Song et al., 2020; Ho et al., 2020; Song & Ermon, 2020; Song et al., 2021b; Gong & Li, 2021), or better discretizing the underlying differential equations to accelerate sampling (Jolicoeur-Martineau et al., 2021; Karras et al., 2022).

Normalizing Flows: Popularized by Rezende & Mohamed (2015), normalizing flows leverage the *change-of-variables* formula to learn a transformation from a parametric prior distribution to a target. The network architecture is constrained so as to facilitate easy computation of the Jacobian (Dinh et al., 2015, 2017; Kingma & Dhariwal, 2018). Recent approaches design flows based on autoregressive models (Kingma et al., 2016; Papamakarios et al., 2017; Su & Wu, 2018), or architectures motivated by the Sobolev GAN loss (Mroueh et al., 2019; Mroueh & Rigotti, 2020). Glaser et al. (2021); Ansari et al. (2021) use KL-flow to iteratively improve the noise vector input to GANs.

In the GAN context, consider the generator push-forward distribution $p_g = G_{\#}(p_z)$. For the main results of this paper, **we assume** $G\colon \mathbb{R}^n \to \mathbb{R}^n$, where the generator is a *diffeomorphism* with a well-defined inverse G^{-1}, both G and its inverse being differentiable. Therefore, $z \in \mathbb{R}^n$ is no longer the *latent* representation. Then, by the change-of-variables formula, we have:

$$p_g(\boldsymbol{x}) = p_z(\boldsymbol{z})\,|\det\ \mathrm{J}_G(\boldsymbol{z})|^{-1}, \text{ where } \boldsymbol{z} = G^{-1}(\boldsymbol{x}), \tag{2}$$

where in turn, $\mathrm{J}_G(\boldsymbol{z})$ is the Jacobian of the generator. Standard mathematical notations used in this paper, and relevant background on the *Fundamental Lemma of the Calculus of Variations* (Gel'fand & Fomin, 1964) are provided in Appendix A. We now present results discussing the optimal generator in divergence minimizing GANs.

3 Divergence Minimizing GANs

Consider the SGAN optimization: $\min_G \max_D \left\{\mathbb{E}_{\boldsymbol{x}\sim p_d}[\ln D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z}[\ln(1-D(G(\boldsymbol{z})))]\right\}$. In practice, the optimization is an alternating one, wherein the discriminator D_t is derived given the generator of the previous iteration, G_{t-1}, and the subsequent generator optimization involves computing G_t, given D_t and G_{t-1}. For simplicity, we denote the push-forward distribution at iteration t as $p_t(\boldsymbol{x}) = G_{t,\#}(p_z(\boldsymbol{z}))$. Within this formulation, the optimization becomes:

$$\mathcal{L}_D^S(D;G_{t-1}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_d}[\ln D(\boldsymbol{x})] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{t-1}}[\ln(1-D(\boldsymbol{x}))], \text{ where } D_t^*(\boldsymbol{x}) = \arg\max_D \left\{\mathcal{L}_D^S\right\} \quad (3)$$

$$\text{and } \mathcal{L}_G^S(G;D_t^*,G_{t-1}) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_z}[\ln(1-D_t^*(G(\boldsymbol{z})))], \text{ where } G_t^*(\boldsymbol{x}) = \arg\min_G \left\{\mathcal{L}_G^S\right\}. \quad (4)$$

Ideally, both the discriminator and generator would converge to the optimal solution at $t = 1$. However, in practice, through a stochastic-gradient-descent update, under mild assumptions, the alternating optimization converges to the desired optimum (Franceschi et al., 2022), *i.e.,* p_g converges to the p_d in the limit ($\lim_{t\to\infty} p_t(\boldsymbol{x}) = p_d(\boldsymbol{x})$). Optimization of the loss in Equation (3) can be carried out pointwise (Goodfellow et al., 2014), with the resulting optimum given by:

$$D_t^*(\boldsymbol{x}) = \frac{p_d(\boldsymbol{x})}{p_d(\boldsymbol{x}) + p_{t-1}(\boldsymbol{x})}. \quad (5)$$

Assume that the generator has not converged, *i.e.,* $p_{t-1}(\boldsymbol{x}) \neq p_d(\boldsymbol{x})$, and that the distribution p_d and p_{t-1} have overlapping support. Then, the following theorem gives us the optimality condition for generator (G_t^*), given D_t^*, such that $p_t(\boldsymbol{x}) = G_{t,\#}^*(p_z) = p_d(\boldsymbol{x})$.

Theorem 3.1. *Consider the generator cost in Equation* (4)*, and the optimal discriminator given by Equation* (5)*. The* ***optimal SGAN generator*** *that minimizes $\mathcal{L}_G^S$ satisfies*

$$\nabla_{\boldsymbol{x}} \ln(p_{t-1}(\boldsymbol{x}))\big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \nabla_{\boldsymbol{x}} \ln(p_d(\boldsymbol{x}))\big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})}, \quad (6)$$

where $\boldsymbol{z} \sim p_z$, and $\nabla_{\boldsymbol{x}} \ln p_{t-1}$ is the score of the push-forward generator distribution $G_{t-1,\#}(p_z)$.

The proof is provided in Appendix C.1. The optimality condition on $G_t^*(\boldsymbol{x})$ can be derived element-wise through the Fundamental Lemma of Calculus of Variations and vectorized to yield the result in Theorem 3.1. The above result is valid only for those $\boldsymbol{x}$ such that both $p_d(\boldsymbol{x})$ and $p_{t-1}(\boldsymbol{x})$ are non-zero. The implications of such a strong condition are discussed in Section 5. Owing to the score-based approach to training the generator, the proposed approach is called *ScoreGAN*. Before we explore the implications of the score-matching form of the optimal generator, we consider a generalization to all f-GANs. Nowozin et al. (2016) considered f-divergences of the form: $\mathfrak{D}_f(p_d \| p_{t-1}) = \int_{\mathcal{X}} f(r_{t-1}(\boldsymbol{x}))\, p_d(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$,, where $f\colon \mathbb{R}_+ \to \mathbb{R}$ is a convex, lower-semicontinuous function over the support $\mathcal{X}$ and satisfies $f(1) = 0$ and $r_{t-1}(\boldsymbol{x})$ is the density ratio $r_{t-1}(\boldsymbol{x}) = \frac{p_d(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x})}$. The generator loss is given by

$$\mathcal{L}_G^f(G; D_t^*, G_{t-1}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_d}[g(D^*(\boldsymbol{x}))] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{t-1}}[f^c(g(D^*(\boldsymbol{x})))], \quad (7)$$

where the real-values discriminator D_t is restricted to a desired domain by means of an activation function $g^*(\cdot)$, and f^c denotes the Fenchel conjugate of f. The following theorem presents the optimal generator transformation, given D_t^*, for f-GANs.

Theorem 3.2. *Consider the generator loss in f-GANs, given by Equation* (7)*. The* ***optimal f-GAN generator*** *satisfies the following score-matching condition:*

$$\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})\nabla_{\boldsymbol{x}}(\ln r_{t-1}(\boldsymbol{x})) = \mathbf{0}, \textit{ where} \quad (8)$$
$$\mathscr{C}(\boldsymbol{x}; p_d, p_{t-1}) = r_{t-1}(\boldsymbol{x}) g'(t)\big|_{t=D_t^*} D_t^{*\prime}(y)\big|_{y=\ln(r_{t-1})},$$

where in turn, $g'(t)$ denotes the derivative of the activation function with respect to D evaluated at D_t^*, $D_t^{*\prime}(y)$ denotes the derivative of the optimal discriminator function with respect to $y = \ln(r_{t-1}(\boldsymbol{x}))$, evaluated at $\ln(r_{t-1}(\boldsymbol{x}))$ (cf. Table 1, Appendix C.2) and $\boldsymbol{x} = G_t^*(\boldsymbol{z})$, $\boldsymbol{z} \sim p_z$.

The proof follows along the lines as that of Theorem 3.1 by simplifying the costs in Equation (7), and substituting for D_t^*. The details are discussed in Appendix C.2. While Theorem 3.2 gives the general solution for f-GANs, we remark that the solution is similar to that of the SGAN and subsumes the result of Theorem 3.1 for appropriate choices of g and f^c. Additionally, for $\boldsymbol{z}$ such that $\mathscr{C}(\boldsymbol{x};\, p_d, p_{t-1}) \neq 0$, the derived solution can further be simplified to yield the score matching cost:

$$\nabla_{\boldsymbol{x}} \ln \left(p_{t-1}(\boldsymbol{x})\right) \big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \nabla_{\boldsymbol{x}} \ln \left(p_d(\boldsymbol{x})\right) \big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})}.$$

Although the result shows that all f-GAN generators are inherently score-matching in nature, the effect of $\mathscr{C}$ accounts for the difference in training stability observed across f-GAN variants. We discuss these results in Appendix C.2. For example, $\mathscr{C}$ is unity only for reverse-KL (RKL) GANs, *i.e.,* the generator goes to zero only when the scores match exactly. This is consistent with empirical results by Nguyen et al. (2017); Shannon et al. (2020), where the relatively stabler non-saturating GAN loss considered by Goodfellow et al. (2014) was shown to approximate an RKL loss in practice.

4 The Optimal Generator in IPM GANs

Arjovsky et al. (2017) proposed Wasserstein GANs (WGANs) as an alternative to divergence minimizing GANs. Motivated by *optimal transport*, the discriminator (also called the *critic*) minimizes the Wasserstein-1 distance between p_d and p_g. The IPM GAN optimization is defined through the Kantorovich–Rubinstein duality as: $\min_{p_g} \left\{ \max_D \left\{ \mathbb{E}_{\boldsymbol{x}\sim p_d}[D(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}\sim p_g}[D(\boldsymbol{x})] + \Omega_D \right\} \right\}$, where Ω_D is an appropriately chosen regularizer. While Arjovsky et al. (2017) enforce a Lipschitz-1 discriminator by clipping the network weights, subsequent variants consider regularizers that bound the energy in the discriminator gradient (Petzka et al., 2018; Mroueh et al., 2018; Adler & Lunz, 2018; Asokan & Seelamantula, 2023b), resulting in Sobolev constraint spaces. The optimal discriminator in these variants has been shown to be the solution to partial differential equations (PDEs) (Mroueh et al., 2018; Asokan & Seelamantula, 2023b), and can be represented through convolutions with the Green's function of the PDEs. As in the f-GAN case, consider the alternating minimization involving G_{t-1}, D_t and G_t. The optimal discriminator in gradient-regularized WGANs is (Unterthiner et al., 2018; Asokan & Seelamantula, 2023b):

$$D_t^*(\boldsymbol{x}) = \mathfrak{C}_\kappa \left((p_{t-1} - p_d) * \kappa \right)(\boldsymbol{x}), \tag{9}$$

where the kernel κ is the Green's function to the differential operator governing the optimal discriminator and $\mathfrak{C}_\kappa$ is a positive constant. For example, in Poly-WGAN (Asokan & Seelamantula, 2023b), the kernel is a polyharmonic spline kernel, while in MMD-GANs, Li et al. (2017) considered Gaussian and inverse multi-quadric kernels (cf. Appendix D.1). The following theorem presents the optimality condition for the generator in kernel-based GANs:

Theorem 4.1. *Consider the generator loss given by $\mathcal{L}_G^{\kappa}(G; D_t^*, G_{t-1}) = -\,\mathbb{E}_{\boldsymbol{z}\sim p_z}[D_t^*\left(G(\boldsymbol{z})\right)]$, and the optimal discriminator given in Equation 9. The **optimal IPM-GAN generator** satisfies*

$$\mathfrak{C}_\kappa \left(\underset{\boldsymbol{y}\sim p_{t-1}}{\mathbb{E}} \left[\nabla_{\boldsymbol{y}} \ln p_{t-1}(\boldsymbol{y}) \kappa(\boldsymbol{x}-\boldsymbol{y})\right] - \underset{\boldsymbol{y}\sim p_d}{\mathbb{E}} \left[\nabla_{\boldsymbol{y}} \ln p_d(\boldsymbol{y}) \kappa(\boldsymbol{x}-\boldsymbol{y})\right] \right) \big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \mathbf{0}, \tag{10}$$

for all $\boldsymbol{x} = G_t^*(\boldsymbol{z})$, $\boldsymbol{z} \sim p_z$, where $\mathfrak{C}_\kappa$ is a non-zero constant dependent on the kernel κ.

The above theorem shows that the optimal generator in IPM GANs is also one of score-matching, where the score is conditioned by the kernel function, centered around $\boldsymbol{x}$. As in f-GANs, the above condition must be met for all $\boldsymbol{x}$, which is relatively stringent. We observe that the condition presented in Theorem 4.1 is equivalent to a condition on the kernel gradient, given by the following lemma.

Lemma 4.2. *Consider the optimality condition for the IPM generator, presented in Theorem 4.1. The condition can equivalently be written as:*

$$\mathfrak{C}_\kappa \left((p_d - p_{t-1}) * \nabla_{\boldsymbol{x}} \kappa \right)(\boldsymbol{x}) \big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \mathbf{0}, \tag{11}$$

where $\nabla_{\boldsymbol{x}}\kappa$ denotes the gradient vector of the kernel, and the convolution must be interpreted element-wise, i.e., $p_d(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})$ is convolved with each entry of $\nabla_{\boldsymbol{x}}\kappa$.

The proof of Theorem 4.1 and Lemma 4.2 follow analogously to the f-GAN scenario, and are presented in detail in Appendix D.1. The optimal IPM GAN generator can be seen as minimizing a proxy to the score – similar to the Stein score, where the gradient field induced by the kernel κ is maximized at locations where data samples are present. As observed in Coulomb GANs, these are akin to charge-potential fields, with the *attractive* data samples and *repulsive* generator samples.

5 Interpreting the Optimal Generator

The optimality condition in f-GANs (cf. Theorem 3.2) brings to light the underlying link between f-GANs and score-based models. While NCSN and its variants rely on Langevin dynamics to model transformation, the optimal generator in GANs can be interpreted as approximating these iterations one-shot. In practice, as the score is undefined when either $p_d(\boldsymbol{x})$ or $p_{t-1}(\boldsymbol{x})$ are zero, the optimality condition cannot be met pointwise, but must be approximated (cf. Section 6). On the other hand, the optimality of IPM-GANs link the generator to conditioned score-matching, and flow-based models. From Theorem 4.1, we see that the generator performs *smoothed* score-matching. As opposed to being a pointwise condition on $\boldsymbol{x}$ as in f-GANs, the generator in IPM-GANs minimizes a weighted average of the score, with the kernel inducing the weight function. This also alleviates convergences issues arising due to non-overlapping supports of p_d and p_{t-1}.

From Lemma 4.2, we see that the gradient field of the kernels convolved with the density difference, and the data score $\nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)$, serve similar purposes, which is to output an arbitrarily large value at data sample location, and low values elsewhere. Unlike the score, however, the kernel gradients produce a repulsive force at the location of generator samples, resulting in a *push-pull* framework – The target distribution creates a *pull*, while the generator distribution creates the *push*. This serves to validate why IPM GANs typically do not suffer from vanishing gradients (Arjovsky & Bottou, 2017), as opposed to the f-divergence counterparts. When $p_0(\boldsymbol{x})$ is initialized far from the target, although the *influence* of the score is weak, the repulsive force of the kernel-based loss is strong. FloWGANs can also be used to explain denoising diffusion GANs (DDGAN, Xiao et al. (2022)), wherein a GAN is trained to model the reverse diffusion process, with the generator and discriminator networks conditioned on the time index. DDGAN can be seen as a special instance of FloWGAN, with Langevin updates over the gradient field of the time-conditioned discriminator (cf. Appendix D). FloWGANs can also be viewed as generalized score matching (Lyu, 2009) where the IPM-GAN generators minimize a *generalized score*, *i.e.,* given an IPM GAN, an equivalent diffusion model exists, with the flow field induced by the kernel of the discriminator, and vice versa. We present a proof-of-concept implementation of this approach in Section 7.2.

6 Practical Considerations

Consider the score matching condition given in Theorem 3.1. We approximate the pointwise optimality condition with a least-squares cost. Given a neural network generator G_{θ_t}, where θ_t denotes the network parameters at time t, this gives rise to the *Fisher divergence* between the scores:

$$\mathcal{L}_G^{\mathrm{Sc}}(\theta) = \mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_z}\left[\left\|\nabla_{\boldsymbol{x}}\ln(p_{t-1}(\boldsymbol{x})) - \nabla_{\boldsymbol{x}}\ln(p_d(\boldsymbol{x}))\right\|_2^2\Big|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z})}\right],$$

where $\theta^* = \arg\min_\theta \mathcal{L}_G^S(\theta)$. The above loss involves computing two key terms: (i) The score of the target data; and (ii) The score of the generator distribution. For parametric distributions such as Gaussians, the score of the data can be computed by means of automatic differentiation (Abadi et al., 2016; Paszke et al., 2019). In the case of image data, a pre-trained score network $S_\phi^{\mathcal{D}}$ can be used to approximate the score of the data (Song & Ermon, 2020; Song et al., 2021a; Rombach et al., 2022).

To compute the score of the generator, when the dimensionality of the data is relatively small, say $\mathcal{O}(10^3)$, the *change of variables* formula (cf. Equation (2)) can be used. The following lemma presents the score of the generator distribution.

Lemma 6.1. *Consider the push-forward generator distribution $p_t(\boldsymbol{x}) = G_{\theta_t,\#}(p_z)$, where $p_z = \mathcal{N}(\boldsymbol{z}; \mu_z, \Sigma_z)$ and $G_{\theta_t}\colon \mathbb{R}^n \to \mathbb{R}^n$. Then, the generator score is given by:*

$$\nabla_{\boldsymbol{x}}\ln p_t(\boldsymbol{x})\Big|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z})} = -\mathrm{J}_{G_{\theta_t}}^{-\mathrm{T}}\left(\nabla_{\boldsymbol{z}}\ln|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z})| + \boldsymbol{z}\right),$$

where $\mathrm{J}_{G_{\theta_t}}$ denotes the Jacobian of the generator G_{θ_t}.

The proof is discussed in Appendix C.4. Generalizations considering $G\colon \mathbb{R}^d \to \mathbb{R}^n;\ d \ll n$ are discussed in Appendix C.5. In very high dimensions, the Jacobian computations are inefficient, and one could consider training a second score network, S_ψ^G, to approximate the score of the generator, trained jointly with the generator in a *non-adversarial* fashion. Although this is a potential alternative to circumventing evaluation of the generator score in closed-form, given the enormous computational overhead of ScoreGAN (cf. Appendices C and E), we focus on the FloWGAN formulation.

The result in Lemma 4.2 links IPM GANs to flow-based models, where the flow-field is induced by the discriminator kernel. First, we consider training the generator network with a least-squares loss, similar to ScoreGANs. The following lemma presents the loss function in FloWGAN:

Lemma 6.2. *Consider the optimality condition in Equation* (10). *Let G_{θ_t} denote the neural network generator, parametrized by θ_t. The FloWGAN generator loss is given by:*

$$\mathcal{L}_G^{\mathrm{FloW}} = \underset{\boldsymbol{z}\sim p_z}{\mathbb{E}}\left[\left\| \sum_{\boldsymbol{y}\sim p_{t-1}} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z})-\boldsymbol{y}} - \sum_{\boldsymbol{y}\sim p_d} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z})-\boldsymbol{y}} \right\|_2^2\right].$$

The proof follows akin to the approach used in Coulomb GANs (Unterthiner et al., 2018), and is given in Appendix D.2. While we use the Euclidean 2-norm to train the GAN on *Fisher-like* divergences, one could also consider our distance metrics or norms in the generator loss. Unlike in f-GANs, Lemma 6.2 hold for all input dimensionalities, *i.e.,* for $G_{\theta_t} : \mathbb{R}^d \to \mathbb{R}^n,\ \forall\ d$.

As a proof of concept, we also consider a Langevin sampling approach to generative modeling, where the score of the data is replaced with the gradient of the kernel-based discriminator. While the score of the data possesses a *strong attractive force* in regions close to the target data, it does not influence samples that are far away. On the other hand, the kernel gradients possess a repulsive term that *pushes* particles away from where they previously were, thereby accelerating convergence. We consider the following update scheme:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_t \nabla_{\boldsymbol{x}} D_t^*(\boldsymbol{x}_t) + \gamma_t \boldsymbol{z}_t, \quad \text{where} \quad \boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{0}_n, \mathbb{I}_n)$$

and the discriminator gradient is an N-sample estimate with centers consisting of data samples $\boldsymbol{d}^i \sim p_d$, and the set of samples generated at the previous iteration $\{\boldsymbol{x}_{t-1} \,|\, \boldsymbol{x}_{t-1} \sim p_{t-1}\}$, given by:

$$\nabla_{\boldsymbol{x}} D_t^*(\boldsymbol{x}_t) = \mathfrak{C}_k' \sum_{\boldsymbol{g}^j \sim \{\boldsymbol{x}_{t-1}\}} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x}_t - \boldsymbol{g}^j) - \mathfrak{C}_k' \sum_{\boldsymbol{d}^i \sim p_d} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x}_t - \boldsymbol{d}^i). \tag{12}$$

Typically, $\gamma_t = \sqrt{2\alpha_t}$, while α_t is decayed geometrically (Song & Ermon, 2019). Within this framework, the training time is *traded in* for memory overhead – We do not require a trained score network, but require random batches of samples drawn $\{\boldsymbol{d}^i \sim p_d\}$ at each sampling step.

7 Experimental Validation

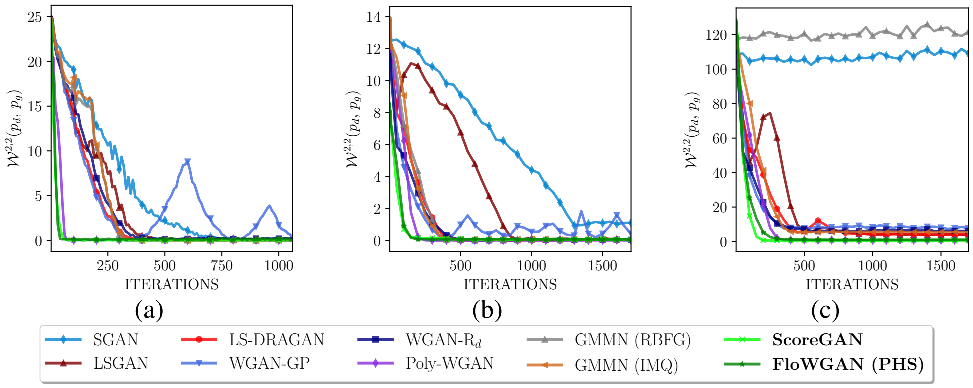
To validate the observations made in Sections 3–4, within a GAN setting, we consider synthetic experiments on learning Gaussians, and extend the FloWGAN approach to learning the latent-space of images using pre-trained autoencoder networks. Subsequently, as a proof-of-concept, we provide results on Langevin sampling with the gradient on the discriminator. While these experiments are not targeted towards outperforming state-of-the-art GANs (Sauer et al., 2022; Kang et al., 2023), they serve to illuminate the training dynamics present in these GAN variants.

7.1 Training GANs with Score-based and Flow-based Losses

As baselines, we consider SGAN (Goodfellow et al., 2014), LSGAN (Mao et al., 2017) and WGAN-GP (Gulrajani et al., 2017). In addition, we compare against gradient-regularized alternatives such as LS-DRAGAN (Kodali et al., 2017), WGAN-R$_d$ (Mescheder et al., 2018), and kernel-based generative moment matching networks (GMMNs) with the inverse multi-quadric (IMQ) and Gaussian (RBFG) kernels (Li et al., 2015), and Poly-WGAN (Asokan & Seelamantula, 2023b).

Experiments on Gaussian data: We present results on learning 2-, 16-, and 128-dimensional univariate Gaussians. The generator is a linear transformation $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}$. Additional network details are given in Appendix E. From Figure 2, we observe that in all three scenarios, GMMNs and Poly-WGAN converges faster than the baseline GANs owing to the lack of adversarial training. On low-dimensional Gaussian learning, FloWGAN is on par with Poly-WGAN, while ScoreGAN converges the fastest. As the dimensionality increases, the convergence of Poly-WGAN worsens, as the kernel decays to zero quickly, while FloWGAN, that relies on the gradient of the kernel, remains unaffected. SGAN and GMMN (RBFG) fail to converge on 128-D data, owing to vanishing gradients.

Experiments on Gaussian-mixture data: To showcase the instability of f-divergence based GANs when p_{t-1} and p_d possess non-overlapping supports, we present results on learning Gaussian mixture



(a) (b) (c)

Figure 2: (Color online) Comparisons between ScoreGAN, FloWGAN and the baselines in terms of the Wasserstein-2 distance $\mathcal{W}^{2,2}(p_d, p_g)$ on learning (a) a 2-D; (b) a 16-D; and (c) a 128-D Gaussian. ScoreGAN and FloWGAN converge an order of magnitude faster than the baseline GANs, while being on par with Poly-WGAN on 2-D Gaussians. As the data dimensionality increases, ScoreGAN and FloWGAN outpace Poly-WGAN due to the availability of gradients of larger magnitude. The baseline SGAN and GMMN (RBFG) fail to converge on 128-D Gaussian data. While ScoreGAN converges the fastest, its computational load is higher, due to computing the Jacobian of the generator.

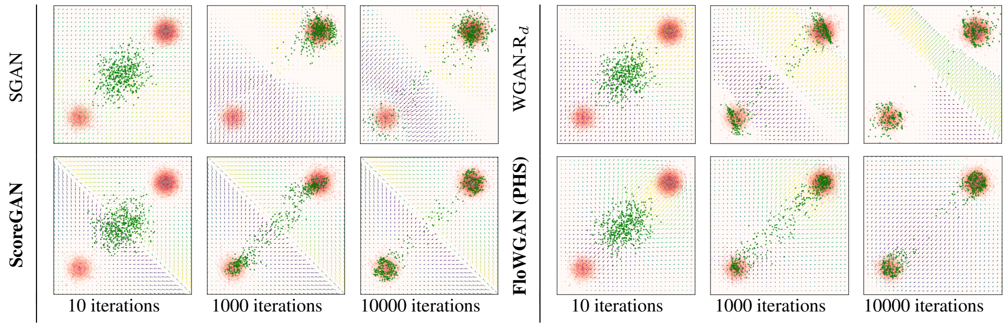
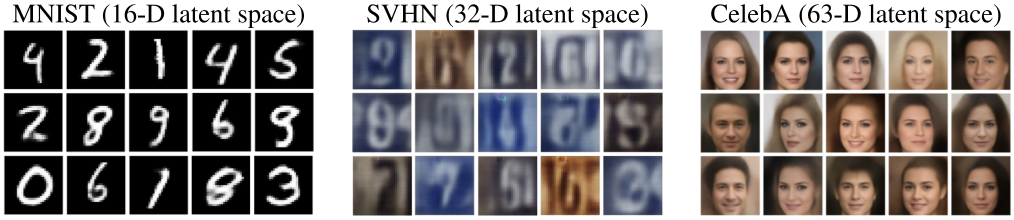


Figure 3: (Color online) Convergence of the generator samples to the target two-component Gaussian, $p_d(\boldsymbol{x}) = \frac{1}{5}\mathcal{N}(-5\mathbf{1}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(5\mathbf{1}, \mathbb{I})$. The quiver plot (best seen in the zoomed-in mode on a *.pdf*) depicts the gradient field of the discriminator on baseline variants, and the gradient of the PHS kernel convolved with the density difference in FloWGAN. While SGAN collapses to the more pronounced mode, ScoreGAN and FloWGAN converges to the target accurately.


MNIST (16-D latent space)

SVHN (32-D latent space)

CelebA (63-D latent space)

Figure 4: (Color online) Images generated by decoding latent-space representations learnt by FloWGAN generator trained with the polyharmonic-spline kernel. The target latent-space distributions are obtained from a pre-trained convolutional autoencoder. FloWGAN converges in 10^3 iterations on MNIST, and 10^4 iterations on SVHN and CelebA.

data. Similar empirical observations were made when training WGANs and SGAN on the Dirac measure (Arjovsky & Bottou, 2017). Here, we present experiments on the two-component Gaussian mixture originally considered by Song & Ermon (2019): $p_d(\boldsymbol{x}) = \frac{1}{5}\mathcal{N}(-5\mathbf{1}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(5\mathbf{1}, \mathbb{I})$. Figure 3 provides the generator and data distributions, juxtaposed with the discriminator gradient for a select few baseline GANs, and the flow field in FloWGAN. Comparisons and ablation experiments are provided in Appendix E.1. From Figure 3, we observe that, while the IPM-based GANs converge accurately to the desired target distribution, SGAN misses the less-represented mode located at $\boldsymbol{\mu} = -5\mathbf{1}$. This can be explained through Theorem 3.2 – When the generated samples are far from the data, $p_d(G_{\theta_t}(\boldsymbol{z})) \to 0$, leading to small gradients induced by the rapid decay of the score.

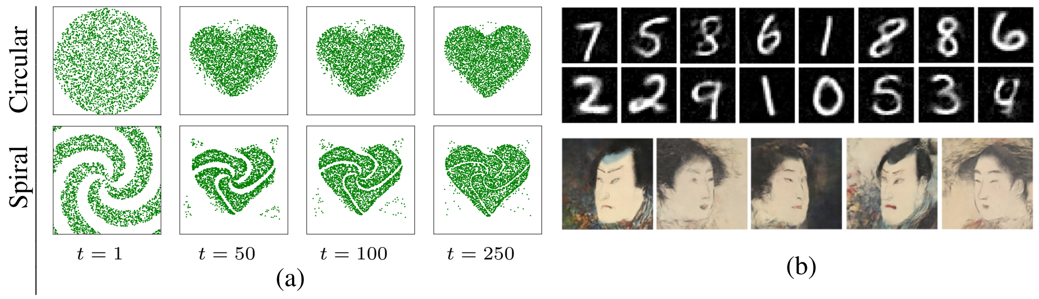


Figure 5: (● Color online) (a) Shape morphing using discriminator-guided Langevin sampler. For relatively simpler input shapes, such as the circular pattern, the sampler converges in about 100 iterations, while in the spiral case, the sampler converges in about 500 steps. (b) Images generated using the discriminator-guided Langevin sampler on MNIST and Ukiyo-E faces datasets. The score in standard diffusion models is replaced with the gradient field of the discriminator, obviating the need for a trainable neural network.

Experiments on Image Data: Although ScoreGANs possess superior convergence, the computational overhead in evaluating the Jacobian of the generator, and its sensitivity to distribution overlap, make it an unfavorable choice in practice. FloWGANs scale more favorably to higher-dimensional data. Therefore, we consider training the FloWGAN generator on the latent-space representation of standard image datasets. We pre-train an autoencoder on MNIST (LeCun et al., 1998), SVHN (Netzer et al., 2011), and CelebA (Liu et al., 2015), with 16-, 32- and 63-dimensional latent-space, respectively. Subsequently, we train the baseline GANs and FloWGAN with the polyharmonic spline (PHS) kernel to model the latent-space of the data. Figure 4 depicts the images generated by FloWGAN on various datasets, while additional comparisons provided in Appendix E.2. FloWGAN converges in 10^3 iterations on MNIST, and 10^4 iterations on SVHN and CelebA, and perform on par with Poly-WGAN.

7.2 Discriminator-guided Langevin Diffusion

To demonstrate the performance of the discriminator-guided Langevin flow, we consider shape morphing, proposed by Mroueh et al. (2019). The source and target samples are drawn uniformly from the interior regions of pre-defined shapes. Figure 10(a) depicts two such scenarios, where the target shape is a heart, and the input shapes are a disk, and a spiral, respectively. Additional combinations are presented in Appendix F. The discriminator-guided Langevin sampler converges in about 500 iterations in all the scenarios considered, compared to the 800 iterations reported in Sobolev descent (Mroueh et al., 2019; Mroueh & Rigotti, 2020), without the need for training a network to approximate the discriminator kernel.

We extend the proposed approach to images, considering MNIST, SVHN and Ukiyo-E (Pinkney & Adler, 2020) datasets. Ablation experiments on the choice of α_t and γ_t are provided in Appendix F. Figure 10(b) presents the samples generated by this discriminator-guided Langevin sampler on MNIST and 256-dimensional Ukiyo-E faces. The model converges to realistic images in as few as 300 steps of sampling, resulting in performance comparable to baseline NCSN (Song & Ermon, 2019). Subsequent iterations, akin to NCSN models, serve to *clean* the noisy images generated. Additional experiments are provided in Appendix F.

8 Discussions and Conclusion

In this paper, we proposed a novel approach to analyzing the optimal generator in divergence-minimizing and IPM-based GAN, through the perspective of variational Calculus. While our analysis covers most popular GAN flavors, the analysis can be extended to any GAN loss function (cf. Appendix C.3). We derive two core results corresponding to f-GANs and IPM-GANs — Theorems 3.1 and 3.2 show that in all f-GANs, the generator is a score-matching network. In IPM-GANs, the GAN generator performs *smoothed* score matching, with the weight function corresponding to the kernel associated with the RKHS of the discriminator constraint (Theorem 4.1). We showed that this is equivalent to minimizing a flow-field induced by the kernel gradient.

These results deepen our understanding of the optimality in GANs. For examples, the score loss in f-GANs help explain their poor performance on non-overlapping distributions. The *push-pull* nature of the IPM-GAN loss is what allows for avoiding local minima. Beyond explaining the optimality in existing GANs, we present novel training algorithm based on the score-matching, and flow-based loss functions (giving rise to *ScoreGANs* and *FloWGANs*, respectively), and discriminator-guided Langevin diffusion. While ScoreGANs suffer from the documented pitfalls of score minimization, the FloWGAN algorithm can be scaled to learn the latent-space distribution of data. ScoreGANs and FloWGANs provide a framework for deriving equivalent diffusion models, given a GAN, and *vice versa.* An in-depth analysis of discriminator-guided diffusion, cosidering alternative sampling techniques (Jolicoeur-Martineau et al., 2021; Karras et al., 2022; Rissanen et al., 2023), or latent-space models (Rombach et al., 2022) is a promising direction for future research.

Acknowledgments

Siddarth Asokan is supported by the Microsoft Research Ph.D. Fellowship 2018, Qualcomm Innovation Fellowship in 2019, 2021, and 2022 and the Robert Bosch Center for Cyber-Physical Systems Ph.D. Fellowship for 2020-2022. Nishanth Shetty is supported by the Prime Minister's Research Fellowship.

References

Abadi, M. et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint, arXiv:1603.04467*, Mar. 2016. URL `https://arxiv.org/abs/1603.04467`.

Adler, J. and Lunz, S. Banach Wasserstein GAN. In *Advances in Neural Information Processing Systems 31*, pp. 6754–6763. 2018.

Ansari, A. F., Ang, M. L., and Soh, H. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=Zbc-ue9p_rE`.

Arbel, M., Korba, A., Salim, A., and Gretton, A. Maximum mean discrepancy gradient flow. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. *arXiv preprints, arXiv:1701.04862*, 2017. URL `https://arxiv.org/abs/1701.04862`.

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

Asokan, S. and Seelamantula, C. S. Euler-Lagrange analysis of generative adversarial networks. *Journal of Machine Learning Research*, 24(126):1–100, 2023a. URL `http://jmlr.org/papers/v24/20-1390.html`.

Asokan, S. and Seelamantula, C. S. Data interpolants – That's what discriminators in higher-order gradient-regularized GANs are. *arXiv preprint, arXiv:2306.00785*, abs/2306.00785, 2023b. URL `https://arxiv.org/abs/2306.00785`.

Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. Demystifying MMD GANs. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Cover, T. and Thomas, J. *Elements of Information Theory*. Wiley-Interscience, 2006.

Dinh, L., Krueger, D., and Bengio, Y. NICE: non-linear independent components estimation. In *3rd International Conference on Learning Representations, Workshop Track Proceedings*, 2015. URL `http://arxiv.org/abs/1410.8516`.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=HkpbnH9lx`.

Ferguson, J. A brief survey of the history of the calculus of variations and its applications. *arXiv preprint, arXiv:math/0402357*, Feb. 2004. URL `https://arxiv.org/abs/math/0402357`.

Franceschi, J.-Y., De Bézenac, E., Ayed, I., Chen, M., Lamprier, S., and Gallinari, P. A neural tangent kernel perspective of GANs. In *Proceedings of the 39th International Conference on Machine Learning*, Jul 2022.

Gel'fand, I. M. and Fomin, S. V. *Calculus of Variations*. Prentice-Hall, 1964.

Glaser, P., Arbel, M., and Gretton, A. KALE flow: A relaxed KL gradient flow for probabilities with disjoint support. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Goldstine, H. H. *A History of the Calculus of Variations from the 17th Through the 19th Century*. Springer, New York, 1980.

Gong, W. and Li, Y. Interpreting diffusion score matching using normalizing flow. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2021. URL `https://openreview.net/forum?id=jxsmOXCDv9l`.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. 2014.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of Wasserstein GANs. In *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint, arXiv:2006.11239*, 2020. URL `https://arxiv.org/abs/2006.11239`.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL `http://jmlr.org/papers/v6/hyvarinen05a.html`.

Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., and Mitliagkas, I. Gotta go fast with score-based generative models. In *The Symbiosis of Deep Learning and Differential Equations*, 2021. URL `https://openreview.net/forum?id=gEoVDSASC2h`.

Jordan, R., Kinderlehrer, D., and Otto, F. The variational formulation of the Fokker–Planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Kang, M., Zhu, J.-Y., Zhang, R., Park, J., Shechtman, E., Paris, S., and Park, T. Scaling up GANs for text-to-image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems 33*, 2020.

Karras, T., Aittala, M., Aila, T., and Laine, S. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Karras, T. et al. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*, volume 34, June 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint, arXiv:1807.03039*, abs/1807.03039, 2018. URL `https://arxiv.org/abs/1807.03039`.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

Kodali, N., Abernethy, J. D., Hays, J., and Kira, Z. On convergence and stability of GANs. *arXiv preprint, arXiv:1705.07215*, May 2017. URL `http://arxiv.org/abs/1705.07215`.

Korotin, A., Kolesov, A., and Burnaev, E. Kantorovich strikes back! Wasserstein GANs are not optimal transport? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Kwon, D., Fan, Y., and Lee, K. Score-based generative modeling secretly minimizes the Wasserstein distance. In *Advances in Neural Information Processing Systems*, 2022.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Li, C. L., Chang, W. C., Cheng, Y., Yang, Y., and Poczos, B. MMD GAN: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems 30*, pp. 2203–2213. 2017.

Li, Y., Swersky, K., and Zemel, R. Generative moment matching networks. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1718–1727, Jul 2015.

Liang, T. How well generative adversarial networks learn distributions. *Journal of Machine Learning Research*, 22(228):1–41, 2021. URL `http://jmlr.org/papers/v22/20-911.html`.

Liu, Q. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

Liu, Q., Lee, J., and Jordan, M. A kernelized Stein discrepancy for goodness-of-fit tests. In *Proceedings of The 33rd International Conference on Machine Learning*, Jun 2016.

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

Lunz, S., Öktem, O., and Schönlieb, C.-B. Adversarial regularizers in inverse problems. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Lyu, S. Interpretation and generalization of score matching. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009.

Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., and Smolley, S. P. Least squares generative adversarial networks. In *Proceedings of International Conference on Computer Vision*, 2017.

Mescheder, L., Geiger, A., and Nowozin, S. Which training methods for GANs do actually converge? In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3481–3490, Stockholmsmassan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Mroueh, Y. and Nguyen, T. On the convergence of gradient descent in GANs: MMD GAN as a gradient flow. In Banerjee, A. and Fukumizu, K. (eds.), *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, Apr 2021.

Mroueh, Y. and Rigotti, M. Unbalanced Sobolev descent. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Mroueh, Y., Li, C., Sercu, T., Raj, A., and Cheng, Y. Sobolev GAN. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Mroueh, Y., Sercu, T., and Raj, A. Sobolev descent. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, Apr 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, T., Le, T., Vu, H., and Phung, D. Dual discriminator generative adversarial nets. volume 30, 2017.

Nowozin, S., Cseke, B., and Tomioka, R. f-GAN: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29*, pp. 271–279. 2016.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems 30*. 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 2021. URL http://jmlr.org/papers/v22/19-1028.html.

Paszke, A. et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, volume 32, 2019.

Petersen, K. B., Pedersen, M. S., et al. The Matrix Cookbook. *Technical University of Denmark*, 7 (15):510, 2008.

Petzka, H., Fischer, A., and Lukovnikov, D. On the regularization of Wasserstein GANs. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Pinetz, T., Soukup, D., and Pock, T. What is optimized in Wasserstein GANs? In *Proceedings of the 23rd Computer Vision Winter Workshop*, 02 2018.

Pinkney, J. N. M. and Adler, D. Resolution dependent GAN interpolation for controllable image synthesis between domains. *arXiv preprint, arXiv:2010.05334*, Oct. 2020. URL https://arxiv.org/abs/2010.05334.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the 4th International Conference on Learning Representations*, pp. 000–000, 2016.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, pp. 15301538, 2015.

Rissanen, S., Heinonen, M., and Solin, A. Generative modelling with inverse heat dissipation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4PJUBT9f2Ol.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Sauer, A., Schwarz, K., and Geiger, A. StyleGAN-XL: scaling StyleGAN to large diverse datasets. volume abs/2201.00273, 2022. URL https://arxiv.org/abs/2201.00273.

Shannon, M., Poole, B., Mariooryad, S., Bagby, T., Battenberg, E., Kao, D., Stanton, D., and Skerry-Ryan, R. Non-saturating GAN training as divergence minimization. *arXiv preprint arXiv:2010.08029*, 2020.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33*, 2020.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pp. 574–584, Jul 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL `https://openreview.net/forum?id=PxTIG12RRHS`.

Stanczuk, J., Etmann, C., Kreusser, L. M., and Schnlieb, C.-B. Wasserstein GANs work because they fail (to approximate the Wasserstein distance). *arXiv preprint, arXiv:2103.01678*, abs/2104.11222, 2021. URL `https://arxiv.org/abs/2103.01678`.

Su, J. and Wu, G. f-VAEs: Improve VAEs with conditional flows. *arXiv preprint, arXiv:1809.05861*, abs/1809.05861, 2018. URL `https://arxiv.org/abs/1809.05861`.

Tolstikhin, I. O., Bousquet, O., Gelly, S., and Schölkopf, B. Wasserstein auto-encoders. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Uehara, M., Sato, I., Suzuki, M., Nakayama, K., and Matsuo, Y. Generative adversarial nets from a density ratio estimation perspective. *arXiv preprint, arXiv:1610.02920*, abs/1610.02920, 2016. URL `https://arxiv.org/abs/1610.02920`.

Unterthiner, T., Nessler, B., Seward, C., Klambauer, G., Heusel, M., Ramsauer, H., and Hochreiter, S. Coulomb GANs: Provably optimal Nash equilibria via potential fields. In *Proceedings of the 6th International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=SkVqXOxCb`.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=JprM0p-q0Co`.

Yi, M., Zhu, Z., and Liu, S. Monoflow: Rethinking divergence GANs via the perspective of differential equations. *arXiv preprint, arXiv:2302.01075*, abs/2302.01075, 2023. URL `https://arxiv.org/abs/2302.01075`.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., Hutchinson, B., Han, W., Parekh, A., Li, X., Zhang, H., Baldridge, J., and Wu, Y. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint, arXiv:2206.10789*, abs/2206.10789, 2022. URL `https://arxiv.org/abs/2206.10789`.

Zhu, B., Jiao, J., and Tse, D. Deconstructing generative adversarial networks. *IEEE Transactions on Information Theory*, 66, 2020.

Appendix

Table of Contents

Overview of Supporting Documents

The Supporting Documents of this manuscript comprise the appendices, animations pertaining to various experiments presented, and the source code for ScoreGANs and FloWGANs. The appendices present relevant mathematical frameworks used in the proofs, the proofs of the theorems stated in the *Main Manuscript* and results of ablation experimentation on synthetic Gaussians and image learning tasks.

A Mathematical Preliminaries

Consider a vector $\boldsymbol{z} = [z_1, z_2, \ldots, z_n]^{\mathrm{T}} \in \mathbb{R}^n$ and the generator $G : \mathbb{R}^n \to \mathbb{R}^n$, *i.e,.,* $G(\boldsymbol{z}) = [G^1(\boldsymbol{z}), G^2(\boldsymbol{z}), \ldots, G^n(\boldsymbol{z})]$. The notation $\nabla_{\boldsymbol{z}} G(\boldsymbol{z})$ represents the gradient matrix associated with the generator, with entries consisting of the partial derivatives of the entries of G with respect to the entries of $\boldsymbol{z}$:

$$\nabla_{\boldsymbol{z}} G(\boldsymbol{z}) = \begin{bmatrix} \frac{\partial G^1}{\partial z_1} & \frac{\partial G^2}{\partial z_1} & \cdots & \frac{\partial G^n}{\partial z_1} \\ \frac{\partial G^1}{\partial z_2} & \frac{\partial G^2}{\partial z_2} & \cdots & \frac{\partial G^n}{\partial z_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial G^1}{\partial z_n} & \frac{\partial G^2}{\partial z_n} & \cdots & \frac{\partial G^n}{\partial z_n} \end{bmatrix}.$$

The Jacobian J can be thought of as *measuring* the transformation that the function imposes locally near the point of evaluation, and is is defined to be the transpose of the gradient, *i.e.,* $\nabla_{\boldsymbol{z}} G(\boldsymbol{z}) = \mathrm{J}_G^{\mathrm{T}}(\boldsymbol{z})$.

Calculus of Variations: Our analysis centers around deriving the optimal generator in the functional sense, leveraging the *Fundamental Lemma of the Calculus of Variations* (Goldstine, 1980; Ferguson, 2004). Consider an integral cost $\mathcal{L}$, to be optimized over a function h:

$$\mathcal{L}\left(h, h'\right) = \int\limits_{\mathcal{X}} \mathcal{F}\left(\boldsymbol{x}, h(\boldsymbol{x}), h'(\boldsymbol{x})\right)\,\mathrm{d}\boldsymbol{x}\,, \tag{13}$$

where h is assumed to be continuously differentiable or at least possess a piecewise-smooth derivative $h'(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. If $h^*(\boldsymbol{x})$ denotes the optimum, The *first variation* of $\mathcal{L}$, evaluated at h^*, is defined as the derivative $\delta\mathcal{L}(h^*; \eta) = \frac{\partial \mathcal{L}_\epsilon(h^*)}{\partial \epsilon}$ evaluated at $\epsilon = 0$, where $\mathcal{L}_\epsilon(h^*)$ denotes an ϵ-perturbation of the argument h about the optimum h^*, given by

$$\mathcal{L}_{h,\epsilon}(\epsilon) = \mathcal{L}\left(h^*(\boldsymbol{x}) + \epsilon\,\eta(\boldsymbol{x}), h^{*\prime}(\boldsymbol{x}) + \epsilon\,\eta'(\boldsymbol{x})\right)$$

where, in turn, $\eta(\boldsymbol{x})$ is a family of *perturbations* that are compactly supported, infinitely differentiable functions, and vanishing on the boundary of $\mathcal{X}$. Then, the optimizer of the cost $\mathcal{L}$ satisfies the following first-order condition:

$$\left.\frac{\partial \mathcal{L}_{h,\epsilon}(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} = 0$$

Another core concept in deriving functional optima is the *Fundamental Lemma of Calculus of Variations*, which states that, if a function $g(\boldsymbol{x})$ satisfies the condition

$$\int_{\mathcal{X}} g(\boldsymbol{x})\,\eta(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0$$

for all compactly supported, infinitely differentiable functions $\eta(\boldsymbol{x})$, then g must be identically zero almost everywhere in $\mathcal{X}$. Together, these results are used to derive the condition that the optimal generator transformation satisfies, within various GAN formulations.

B An Overview of Related Works

Links between diffusion and flows can be traced back to the work of Jordan et al. (1998), where the Fokker-Planck equation was shown to lead to a Kullback-Leibler (KL) flow, discretized to give rise to the Langevin Monte Carlo algorithm. However, an analysis under **KL flow** or **Stein flow** (Liu, 2017) for GANs is infeasible, as this requires the analytical form of the target density. Recently, Gong & Li (2021) showed that diffusion score matching can be interpreted as normalizing flows. Our results, in a similar vein, link GAN generator optimization to both flows, and score matching. Mroueh & Nguyen (2021) leverage **MMD flow** (Arbel et al., 2019) to analyze the convergence in MMD-GANs. Recently, Kwon et al. (2022) showed that the score-matching networks in fact solve for the **Wasserstein flow** between p_d and p_g.

The closest approach to ours is that of **MonoFlow**, proposed by Yi et al. (2023), who showed that the divergence-minimizing discriminator can be seen as approximating the vector field of a gradient flow in the Wasserstein space, induced by a monotonically increasing function of the density ratio. Our results can be seen as a generalization of those considered in MonoFlow, relating both divergence-minimizing, and IPM-based GANs. Liang (2021) optimize IPM-based generator and discriminator networks jointly, and show that additional regularization on the space of the generator functions is necessary in IPM GANs to attain the optimum. Franceschi et al. (2022) propose **NTK-GANs**, a unifying theory for the optimality of GANs considering neural-network discriminators, and show that the generator in and GAN can be seen as minimizing a cost related to the NTK associated with an infinite-width discriminator.

C Optimality of Divergence-minimizing GANs

We now present the proofs for the optimality conditions derived for the divergence minimizing GANs. We also consider an extension, pertaining to a GAN variant that does not directly correspond to the f-divergence, the LSGAN formulation with arbitrarily chosen class labels.

C.1 Optimality of SGAN (Proof of Theorem 3.1)

Recall the SGAN generator optimization problem:

$$\mathcal{L}_G^S(G;\, D_t^*, G_{t-1}) = \underset{\boldsymbol{x}\sim p_d}{\mathbb{E}}[\ln(D_t^*(\boldsymbol{x}))] + \underset{\boldsymbol{z}\sim p_z}{\mathbb{E}}[\ln(1-D_t^*(G(\boldsymbol{z})))]$$

$$G_t^*(\boldsymbol{x}) = \arg\min_G \left\{\mathcal{L}_G^S(G;\, D_t^*, G_{t-1})\right\}, \quad \text{where} \quad D_t^*(\boldsymbol{x}) = \frac{p_d(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x}) + p_d(\boldsymbol{x})}.$$

The optimal discriminator was originally derived through a point-wise optimization by Goodfellow et al. (2014), but later shown to be consistent with the functional form of optimization by Asokan & Seelamantula (2023a). Since the expectation with respect to the data term in $\mathcal{L}_G^S$ does not involve the generator samples at the current iteration t, it can be ignored with respect to the optimization. A similar approach was considered by Franceschi et al. (2022) in analyzing the NTK-GAN formulation. Expanding the integral, and substitute in for the optimal discriminator D_t^* yields:

$$\mathcal{L}_G^S(G) = \int_{\mathcal{Z}} \ln\left(\frac{p_{t-1}(G(\boldsymbol{z}))}{p_{t-1}(G(\boldsymbol{z})) + p_d(G(\boldsymbol{z}))}\right) p_z(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z},$$

where $\mathcal{Z}$ denotes the support of the input distribution p_z. The optimization of $\mathcal{L}_G^S$ is a functional one, and can be found by computing the first variation, and setting it to zero under the *Fundamental Lemma of Calculus of Variations.* Let the optimal solution be denoted by

$$G_t^*(\boldsymbol{z}) = [G_t^{1^*}(\boldsymbol{z}), G_t^{2^*}(\boldsymbol{z}),\ \ldots,\ G_t^{i^*}(\boldsymbol{z}),\ \ldots,\ G_t^{n^*}(\boldsymbol{z})]^{\mathrm{T}},$$

where $G_t^{i^*}$ denotes the optimum along the i^{th} dimension. Let $\mathcal{L}_{G,i,\epsilon}$ be the loss considering an epsilon perturbation of the i^{th} entry about the optimum, given by:

$$G_{t,i,\epsilon}^*(\boldsymbol{z}) = [G_t^{1^*}(\boldsymbol{z}), G_t^{2^*}(\boldsymbol{z}),\ \ldots,\ G_t^{i^*}(\boldsymbol{z}) + \epsilon\eta(\boldsymbol{z}),\ \ldots,\ G_t^{n^*}(\boldsymbol{z})]^{\mathrm{T}},$$

where $\eta(\boldsymbol{z})$ is drawn from a family of compactly supported, infinitely differentiable functions. The loss can now be written as a function of ϵ as:

$$\mathcal{L}_{G,i,\epsilon}^S(\epsilon) = \int_{\mathcal{Z}} \ln\left(\frac{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right) + p_d\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}\right) p_z(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z},$$

Differentiating $\mathcal{L}_{G,i,\epsilon}$ with respect to epsilon and equating it to zero at $\epsilon = 0$ yields:

$$\begin{aligned}\left.\frac{\partial \mathcal{L}_{G,i,\epsilon}^S(\epsilon)}{\partial\epsilon}\right|_{\epsilon=0} &= \int_{\mathcal{Z}} \left.\left(\frac{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right) + p_d\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}\right)\right|_{\epsilon=0} \cdot \\ &\qquad \underbrace{\frac{\partial}{\partial\epsilon}\left(\frac{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}{p_{t-1}\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right) + p_d\left(G_{t,i,\epsilon}^*(\boldsymbol{z})\right)}\right)}_{\mathrm{T}_1} p_z(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} \\ &= 0.\end{aligned}$$

Table 1: Various f-GANs (Nowozin et al., 2016), given the activation function g and the Fenchel conjugate f^c. The corresponding optimal discriminator $D_t^*(\boldsymbol{x})$, derived via pointwise optimization, and the corresponding coefficient function $\mathscr{C}$.

f-divergence	$g(D)$	$f^c(T)$	$D_t^*(\ln(r_{t-1}))$	$\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})$
Kullback-Leibler (KL)	D	e^{T-1}	$1 + \ln\left(r_{t-1}(\boldsymbol{x})\right)$	$r_{t-1}(\boldsymbol{x})$
Reverse KL	$-e^{-D}$	$-1-\ln(-T)$	$\ln\left(r_{t-1}(\boldsymbol{x})\right)$	1
Pearson-χ^2	D	$\frac{1}{4}T^2 + T$	$2\left(r_{t-1}(\boldsymbol{x}) - 1\right)$	$2r_{t-1}^2(\boldsymbol{x})$
Squared-Hellinger	$1 - e^{-D}$	$\frac{T}{1-T}$	$\frac{1}{2}\ln\left(r_{t-1}(\boldsymbol{x})\right)$	$\frac{1}{2}\sqrt{r_{t-1}(\boldsymbol{x})}$
SGAN	$-\ln\left(1 + e^{-D}\right)$	$-\ln\left(1 - e^{T}\right)$	$\ln\left(r_{t-1}(\boldsymbol{x})\right)$	$r_{t-1}^2(\boldsymbol{x})\left(r_{t-1}(\boldsymbol{x}) + 1\right)^{-1}$

Let $\boldsymbol{x} = G_{t,i,\epsilon}^*(\boldsymbol{z})\big|_{\epsilon=0} = G_t^*(\boldsymbol{z})$. The term T_1 can be simplified as:

$$\begin{aligned}
\mathrm{T}_1 &= \left.\left(\frac{p_d(G_{t,i,\epsilon}^*(\boldsymbol{z}))\frac{\partial p_{t-1}(\boldsymbol{y})}{\partial y_i}\big|_{\boldsymbol{y}=G_{t,i,\epsilon}^*(\boldsymbol{z})} - p_{t-1}(G_{t,i,\epsilon}^*(\boldsymbol{z}))\frac{\partial p_d(\boldsymbol{y})}{\partial y_i}\big|_{\boldsymbol{y}=G_{t,i,\epsilon}^*(\boldsymbol{z})}}{\left(p_d(G_{t,i,\epsilon}^*(\boldsymbol{z})) + p_{t-1}(G_{t,i,\epsilon}^*(\boldsymbol{z}))\right)^2} \right)\right|_{\epsilon=0} \eta(\boldsymbol{z}) \\
&= \left(\frac{p_d(\boldsymbol{x})\frac{\partial p_{t-1}(\boldsymbol{x})}{\partial x_i} - p_{t-1}(\boldsymbol{x})\frac{\partial p_d(\boldsymbol{x})}{\partial x_i}}{\left(p_d(\boldsymbol{x}) + p_{t-1}(\boldsymbol{x})\right)^2} \right) \eta(\boldsymbol{z}).
\end{aligned}$$

Substituting back for T_1 in $\frac{\partial \mathcal{L}_{G,i,\epsilon}^S(\epsilon)}{\partial \epsilon}$ and simplifying yields:

$$\left.\frac{\partial \mathcal{L}_{G,i,\epsilon}^S(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} = \int_{\mathcal{Z}} p_z(\boldsymbol{z}) \left.\left(\frac{p_d(\boldsymbol{x})p_{t-1,i}'(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})p_{d,i}'(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x})\left(p_d(\boldsymbol{x}) + p_{t-1}(\boldsymbol{x})\right)} \right) \eta(\boldsymbol{z}) \right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = 0,$$

where $p_{t-1,i}'$ and $p_{d,i}'$ denote the derivative of p_{t-1} and p_d, respectively, with respect to x_i, the i^{th} entry of the argument $\boldsymbol{x}$. Then, from the *Fundamental Lemma of the Calculus of Variations*, we have:

$$p_z(\boldsymbol{z}) \left.\left(\frac{p_d(\boldsymbol{x})p_{t-1,i}'(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})p_{d,i}'(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x})\left(p_d(\boldsymbol{x}) + p_{t-1}(\boldsymbol{x})\right)} \right)\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = 0, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

Since $p_z(\boldsymbol{z})$ is non-zero over its support $\mathcal{Z}$, and if $p_{t-1}(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x} = G_t^*(\boldsymbol{z})$ (which is a reasonable assumption to make, since p_{t-1} is the push-forward distribution of the generator), the optimality condition becomes:

$$\left. p_d(\boldsymbol{x})p_{t-1,i}'(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})p_{d,i}'(\boldsymbol{x}) \right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = 0, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

Rearranging and simplifying yields:

$$\left.\frac{\partial \ln p_{t-1}(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \left.\frac{\partial \ln p_d(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})}, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

Since the analysis holds identically for all i, we have:

$$\nabla_{\boldsymbol{x}} \ln\left(p_{t-1}(\boldsymbol{x})\right)\big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \nabla_{\boldsymbol{x}} \ln\left(p_d(\boldsymbol{x})\right)\big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})}, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

which is the desired result of Theorem 3.1.

C.2 Optimality of f-GAN (Proof of Theorem 3.2)

We now derive the optimality condition for f-GANs in general, and subsequently analyze each variant considered by Nowozin et al. (2016) (cf. Table 1). Recall the f-GAN optimization:

$$\begin{aligned}
\mathcal{L}_D^f(D;\, G_{t-1}) &= -\mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_d}\left[T(\boldsymbol{x})\right] + \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{t-1}}\left[f^c(T(\boldsymbol{x}))\right] \\
\mathcal{L}_G^f(G; D_t^*, G_{t-1}) &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_d}\left[T^*(\boldsymbol{x})\right] - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{t-1}}\left[f^c(T^*(\boldsymbol{x}))\right],
\end{aligned}$$

where $T(\cdot) = g(D(\cdot))$ is the output of the discriminator $D \in \mathbb{R}$, restricted to a desired domain by mean of an activation function $g(\cdot)$, f^c denotes the Fenchel conjugate of f, and $T^*(\cdot) = g(D^*(\cdot))$. The discriminator optimization in f-GANs has been well studied by Nowozin et al. (2016); Asokan & Seelamantula (2023a) and Yi et al. (2023). For completeness, we summarize the result here. Consider the integral form of the discriminator cost:

$$\mathcal{L}_D^f(D;\, G_{t-1}) = \int_{\mathcal{X}} \underbrace{f^c(T(\boldsymbol{x}))\, p_{t-1}(\boldsymbol{x}) - T(\boldsymbol{x})\, p_d(\boldsymbol{x})}_{\mathcal{F}D}\, \mathrm{d}\boldsymbol{x}$$

The integrand $\mathcal{F}$ can be optimized pointwise with respect to T, to derive the optimality condition for the discriminator:

$$\left.\frac{\partial f^c(T)}{\partial T}\right|_{T=T^*(\boldsymbol{x})} = \frac{p_d(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x})} = r_{t-1}(\boldsymbol{x}), \qquad \text{where} \qquad T^*(\boldsymbol{x}) = g(D^*(\boldsymbol{x})). \tag{14}$$

The above can be solved for various choices of $g(\cdot)$ and $f^c(\cdot)$, giving rise to the optimal discriminator $D_t^*(\boldsymbol{x})$ in f-GANs. For convenience, we recall the results in Table 1 of the Appendix. Since the optimal discriminator is always a function of the logarithm of the density ratio, we denote the solution as $D_t^*(\ln(r_{t-1}))$.

Consider the f-GAN generator optimization, given D_t^*. As in the SGAN case, only the term involving the generator samples affects the optimization. The integral form of the loss is given by:

$$\begin{aligned} \mathcal{L}_G^f(G) &= \int_{\mathcal{Z}} f^c(T^*(G(\boldsymbol{z})))\, p_z(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\ \Rightarrow \mathcal{L}_{G,i,\epsilon}^f(\epsilon) &= \int_{\mathcal{Z}} f^c(T^*(G_{t,i,\epsilon}^*(\boldsymbol{z})))\, p_z(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}, \end{aligned}$$

where the perturbed loss $\mathcal{L}_{G,i,\epsilon}^f$ is defined as in the case of SGANs (cf. Appendix C.1). Leveraging the chain rule and computing the derivative with respect to ϵ yields:

$$\begin{aligned} \left.\frac{\partial \mathcal{L}_{G,i,\epsilon}^f(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} = \int_{\mathcal{Z}} &\left.\frac{\partial f^c}{\partial T}\right|_{T=T^*(G_{t,i,\epsilon}^*)} \left.\frac{\partial T^*}{\partial D}\right|_{D=D_t^*(\ln r_{t-1})} \cdot \\ &\left.\frac{\partial D_t^*}{\partial y}\right|_{y=\ln r_{t-1}(G_{t,i,\epsilon}^*)} \frac{\partial}{\partial \epsilon} \left.\big(\ln\big(r_{t-1}(G_{t,i,\epsilon}^*)(\boldsymbol{z})\big)\big)\; p_z(\boldsymbol{z})\right|_{\epsilon=0} \mathrm{d}\boldsymbol{z}. \end{aligned}$$

From the optimality condition given in Equation (14), and relations in Table 1, we have:

$$\begin{aligned} \left.\frac{\partial \mathcal{L}_{G,i,\epsilon}^f(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} &= \int_{\mathcal{Z}} r_{t-1}(\boldsymbol{x}) g'\left(D_t^*(\ln r_{t-1}(\boldsymbol{x}))\right) \left.\frac{\partial D_t^*}{\partial y}\right|_{y=\ln r_{t-1}} \cdot \\ &\qquad \left.\frac{\partial}{\partial x_i}\left(\ln(r_{t-1}(\boldsymbol{x}))\right)\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} \frac{\partial G_{t,i,\epsilon}^*}{\partial \epsilon}\, p_z(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} \\ &= \int_{\mathcal{Z}} \underbrace{r_{t-1}(\boldsymbol{x}) g'\left(D_t^*(\ln r_{t-1}(\boldsymbol{x}))\right) D_t^{*\prime}(y)\big|_{y=\ln r_{t-1}}}_{\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})} \cdot \\ &\qquad \left.\frac{\partial}{\partial x_i}\left(\ln(r_{t-1}(\boldsymbol{x}))\right)\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} \eta(\boldsymbol{z})\, p_z(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} = 0, \end{aligned}$$

where $g'(t)$ denotes the derivatives of the activation function with respect to D evaluated at D_t^*, and $D_t^{*\prime}(y)$ denotes the derivative of the optimal discriminator function with respect to $y = \ln(r_{t-1}(\boldsymbol{x}))$, evaluated at $\ln(r_{t-1}(\boldsymbol{x}))$. As in the case of SGANs, from the *Fundamental Lemma of Calculus of Variations*, we have:

$$\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1}) \left.\frac{\partial}{\partial x_i}\left(\ln(r_{t-1}(\boldsymbol{x}))\right)\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = 0, \qquad \forall\, \boldsymbol{z} \in \mathcal{Z}.$$

The above can be vectorized over all i, giving rise to the result provided in Theorem 3.2:

$$\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})\nabla_{\boldsymbol{x}}\left(\ln r_{t-1}(\boldsymbol{x})\right)\bigg|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \mathbf{0}, \qquad \forall\, \boldsymbol{z} \in \mathcal{Z}.$$

The coefficient $\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})$ can be derived for each f-GAN (given in Table 1), we discuss the results here.

KL divergence: Consider the f-GAN with the Kullback-Leibler divergence. We have $g'(D_t^*) = 1$ and $D_t^{*\prime}(\boldsymbol{x}) = 1$, which gives us $\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1}) = r_{t-1}(\boldsymbol{x})$. Recall that the density ratio is given by $r_{t-1}(G_t^*(\boldsymbol{z})) = \frac{p_d(G_t^*(\boldsymbol{z}))}{p_{t-1}(G_t^*(\boldsymbol{z}))}$. Since p_{t-1} denotes the push-forward distribution of the generator of the previous iteration, for sufficiently small learning rates, the generator samples at time t are sufficiently close to those at $t-1$, and $p_{t-1}(G_t^*(\boldsymbol{z}))$ is non-zero. However, if the generated samples are far from the data density, $p_d(G_t^*(\boldsymbol{z}))$ is close to zero, resulting in vanishing gradients while training — Even if the scores do not match, the training loss is zero, as $p_d(G_{\theta_t}(\boldsymbol{z})) \to 0$.

Reverse-KL (RKL) divergence: For the reverse-KL-based f-GAN, we have $g'(D_t^*) = r_{t-1}^{-1}(\boldsymbol{x})$ and $D_t^{*\prime}(\boldsymbol{x}) = 1$. As a consequence, the coefficient $\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1})$ is unity. Therefore, when trained with the RKL loss, it is clear that the generator would not suffer from vanishing gradients. This observation is consistent with the literature, as Nguyen et al. (2017) and Shannon et al. (2020) have both observed that the non-saturating GAN loss can be seen as a *smoothened* RKL loss.

Pearson-χ^2 divergence: The Pearson-χ^2 GAN can be seen as a special case of LSGAN. Here, we have $g'(D_t^*) = 1$ and $D_t^{*\prime}(\boldsymbol{x}) = r_{t-1}(\boldsymbol{x})$. The coefficient $\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1}) = r_{t-1}^2(\boldsymbol{x})$ grows quadratically in p_d, resulting in vanishing gradients in a more pronounced manner than in KL-GANs. We discuss the effect of choosing alternative class labels in LSGAN, those that do not lead to the Pearson-χ^2 GAN, in Appendix C.3.

Squared-Hellinger divergence: In the Squared-Hellinger GAN formulation, we have $g'(D_t^*) = r_{t-1}^{-\frac{1}{2}}(\boldsymbol{x})$ and $D_t^{*\prime}(\boldsymbol{x}) = \frac{1}{2}$, which yields $\mathscr{C}(\boldsymbol{x};\, p_d,\, p_{t-1}) = \frac{1}{2}r_{t-1}^{\frac{1}{2}}(\boldsymbol{x})$. As the coefficient only decays as the square-root of p_d, we expect that the Squared-Hellinger GAN is relatively more stable, compared to the Pearson-χ^2-divergence based counterpart. This was observed empirically by Nowozin et al. (2016).

C.3 Non-divergence-miniming GAN Formulations

In this section, we consider an example GAN formulation that does not lie within the divergence minimization framework. The results serve to show that the proposed approach can by applied to any existing GAN variant.

Least-squares GAN: Consider the LSGAN formulation presented by Mao et al. (2017) with the discriminator and generator loss given by:

$$\mathcal{L}_D^{LS}(D;\, G_{t-1}) = \underset{\boldsymbol{x}\sim p_d}{\mathbb{E}}\left[(D(\boldsymbol{x}) - b)^2\right] + \underset{\boldsymbol{x}\sim p_{t-1}}{\mathbb{E}}\left[(D(\boldsymbol{x}) - a)^2\right] \text{ and}$$
$$\mathcal{L}_G^{LS}(G;\, D_t^*,\, G_{t-1}) = \underset{\boldsymbol{x}\sim p_d}{\mathbb{E}}\left[(D_t^*(\boldsymbol{x}) - c)^2\right] + \underset{\boldsymbol{z}\sim p_z}{\mathbb{E}}\left[(D_t^*(G(\boldsymbol{z})) - c)^2\right],$$

respectively, where a and b are the class-labels assigned by the discriminator to real and fake samples, respectively. The generator is trained to create samples such that they are classified as c by the discriminator. The discriminator optimization can be carried out pointwise, giving rise to the optimal discriminator:

$$D_t^*(\boldsymbol{x}) = \frac{a p_{t-1}(\boldsymbol{x}) + b p_d(x)}{p_{t-1}(\boldsymbol{x}) + p_d(x)}. \tag{15}$$

As in the case of SGANs, the generator loss can be expanded into the integral form, and evaluated at perturbed location about the optimal solution $G^*_{t,i,\epsilon}$, which yields:

$$\begin{aligned}
\mathcal{L}^{LS}_{G,i,\epsilon}(\epsilon) &= \int_{\mathcal{Z}} \left(D^*_t(G^*_{t,i,\epsilon}(\boldsymbol{z})) - c\right)^2 \, p_z(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \\
\Rightarrow \left.\frac{\partial \mathcal{L}^{LS}_{G,i,\epsilon}(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} &= \int_{\mathcal{Z}} 2\left(D^*_t(G^*_{t,i,\epsilon}(\boldsymbol{z})) - c\right)\Big|_{\epsilon=0} \left.\frac{\partial D^*_t(G^*_{t,i,\epsilon}(\boldsymbol{z}))}{\partial \epsilon}\right|_{\epsilon=0} p_z(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} \\
&= \int_{\mathcal{Z}} 2\left(D^*_t(\boldsymbol{x}) - c\right)\Big|_{\boldsymbol{x}=G^*_t(\boldsymbol{z})} \left.\frac{\partial D^*_t(\boldsymbol{x})}{\partial x_i}\right|_{x=G^*_{t,i,\epsilon}(\boldsymbol{z})} \eta(\boldsymbol{z}) p_z(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = 0 \qquad (16)
\end{aligned}$$

Given the optimal LSGAN discriminator in Equation (15), we have:

$$\begin{aligned}
(D^*_t(\boldsymbol{x}) - c) &= \frac{(a-c)p_{t-1}(\boldsymbol{x}) + (b-c)p_d(\boldsymbol{x})}{p_{t-1}(\boldsymbol{x}) + p_d(\boldsymbol{x})} \quad \text{and} \\
\frac{\partial D^*_t(\boldsymbol{x})}{\partial x_i} &= \frac{(b-a)\left(p_d(\boldsymbol{x})p'_{t-1,i}(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})p'_{d,i}(\boldsymbol{x})\right)}{(p_{t-1}(\boldsymbol{x}) + p_d(\boldsymbol{x}))^2}
\end{aligned}$$

Substituting for the above into Equation (16) yields:

$$\int_{\mathcal{Z}} \underbrace{\frac{(b-a)\left((a-c)p_{t-1}(\boldsymbol{x}) + (b-c)p_d(\boldsymbol{x})\right)}{\left(p_{t-1}(\boldsymbol{x}) + p_d(\boldsymbol{x})\right)^3}}_{\mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c)} \cdot$$
$$\left(p_d(\boldsymbol{x})p'_{t-1,i}(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})p'_{d,i}(\boldsymbol{x})\right)\Big|_{x=G^*_{t,i,\epsilon}(\boldsymbol{z})} \eta(\boldsymbol{z})p_z(\boldsymbol{z}) \, \mathrm{d}\boldsymbol{z} = 0,$$

where $\mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c)$ is the coefficient term, similar to the one seen in the f-GAN formulation, that also depends on the choice of class-labels (a, b, c). From the *Fundamental Lemma of Calculus of Variations*, we have:

$$\mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c)\left(p_d(\boldsymbol{x})\, p'_{t-1,i}(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})\, p'_{d,i}(\boldsymbol{x})\right)\Big|_{x=G^*_{t,i,\epsilon}(\boldsymbol{z})} = 0,$$

As in the case of f-GANs, we see that the least-squares GAN also results in a score-matching loss, when $\mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c)$ is non-zero. Mao et al. (2017) propose two choices of class labels – (i) $(a, b, c) = (-1, 0, 1)$, which satisfy the conditions that $b - c = 1$ and $a - c = -1$, resulting in the Pearson-χ^2 divergence-based GANs; and (2) $(a, b, c) = (0, 1, 1)$, which leads to stabler training. From the solution above, we see that,

$$\begin{aligned}
&\text{When } \ (a,b,c) = (-1,0,1), \ \text{ we have } \ \mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c) = \frac{1}{(p_{t-1}(\boldsymbol{x}) - p_d(\boldsymbol{x}))^2}, \ \text{ and} \\
&\text{When } \ (a,b,c) = (0,1,1), \ \text{ we have } \ \mathcal{C}(\boldsymbol{x};\, p_{t-1}, p_d, a, b, c) = \frac{-p_{t-1}(\boldsymbol{x})}{(p_{t-1}(\boldsymbol{x}) - p_d(\boldsymbol{x}))^3}.
\end{aligned}$$

For either case, for sufficiently small learning rates, the updated sample $\boldsymbol{x} = G^*_t(\boldsymbol{z})$ is sufficiently close to the sample generated at the previous iteration, and we have $p_{t-1}(\boldsymbol{x}) > 0$. As a result, we see that even when the loss does not correspond to a divergence minimizing cost, the class label (a, b, c) can be chosen such that the LSGAN generator optimization results in a score-matching cost.

C.4 Computing the Score of the Generator (Proof of Lemma 6.1)

Consider the push-forward generator distribution at time t, given by $p_t(\boldsymbol{x}) = G_{\theta_t,\#}(p_z)$, where $p_z = \mathcal{N}(\boldsymbol{z}; \mu_z, \Sigma_z)$. We assume that the generator $G_{\theta_t} : \mathbb{R}^n \to \mathbb{R}^n$ is an invertible function, with the inverse given by $G^{-1}_{\theta_t}$. Then, by the change-of-variables formula, we have:

$$p_t(\boldsymbol{x}) = p_z(G^{-1}_{\theta_t}(\boldsymbol{x})) \left|\det \mathrm{J}_{G^{-1}_{\theta_t}}(\boldsymbol{x})\right|.$$

If the generator is invertible, we have,

$$p_t(\boldsymbol{x}) = p_z(G_{\theta_t}^{-1}(\boldsymbol{x}))\,\left|\det \mathrm{J}_{G_{\theta_t}}^{-1}(G_{\theta_t}^{-1}(\boldsymbol{x}))\right| = p_z(G_{\theta_t}^{-1}(\boldsymbol{x}))\,\left|\det \mathrm{J}_{G_{\theta_t}}(G_{\theta_t}^{-1}(\boldsymbol{x}))\right|^{-1}.$$

Then, the score of the generator is given by:

$$\begin{aligned}\nabla_{\boldsymbol{x}} \ln\left(p_t(\boldsymbol{x})\right) &= \nabla_{\boldsymbol{x}} \ln\left(p_z(G_{\theta_t}^{-1}(\boldsymbol{x}))\,\left|\det \mathrm{J}_{G_{\theta_t}}(G_{\theta_t}^{-1}(\boldsymbol{x}))\right|^{-1}\right)\\ &= \nabla_{\boldsymbol{x}}\left(\ln\left(p_z(G_{\theta_t}^{-1}(\boldsymbol{x}))\right) - \ln\left|\det \mathrm{J}_{G_{\theta_t}}(G_{\theta_t}^{-1}(\boldsymbol{x}))\right|\right)\end{aligned}$$

Then, given the transformation $\boldsymbol{x} = G_{\theta_t}(\boldsymbol{z})$, we have

$$\nabla_{\boldsymbol{x}} \ln\left(p_t(\boldsymbol{x})\right) = \mathrm{J}_{G_{\theta_t}}^{-\mathrm{T}}(\boldsymbol{z})\left(\nabla_{\boldsymbol{z}}\left(\ln\left(p_z(\boldsymbol{z})\right) - \ln\left|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z})\right|\right)\right)$$

In most GAN frameworks, p_z is set to be the standard Gaussian $\mathcal{N}(\boldsymbol{z};\,\mathbf{0},\mathbb{I}$. Simplifying for the score of the Gaussian, we get

$$\nabla_{\boldsymbol{x}} \ln\left(p_t(\boldsymbol{x})\right) = \mathrm{J}_{G_{\theta_t}}^{-\mathrm{T}}(\boldsymbol{z})\Bigg(-\boldsymbol{z} - \underbrace{\nabla_{\boldsymbol{z}} \ln\left|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z})\right|}_{\mathrm{T}_1}\Bigg),$$

which is the desired result of Lemma 6.1. In practice, the Jacobian of the generator can be computed using automatic differentiation in standard libraries such as TensorFlow (Abadi et al., 2016) or PyTorch (Paszke et al., 2019). The term T_1 can further be simplified though well-known matrix differentiation properties. Consider the following:

$$\begin{aligned}\mathrm{T}_1 &= \nabla_{\boldsymbol{z}} \ln\left|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z})\right|\\ &= \nabla_{\boldsymbol{z}}\mathrm{J}_{G_{\theta_t}} \otimes \nabla_{\mathrm{J}_{G_{\theta_t}}} \ln\left|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z})\right|,\end{aligned}$$

where $\nabla_{\boldsymbol{z}}\mathrm{J}_{G_{\theta_t}}$ denotes a Hessian tensor in $\mathbb{R}^{n\times n\times d}$, with the $(i,j,k)^{th}$ entry given by $\left[\nabla_{\boldsymbol{z}}\mathrm{J}_{G_{\theta_t}}\right]_{i,j,k} = \frac{\partial[\mathrm{J}_{G_{\theta_t}}]_{i,j}}{\partial z_k}$. Applying the matrix identity $\nabla_{\boldsymbol{M}} \ln|\det \boldsymbol{M}| = \boldsymbol{M}^{-\mathrm{T}}$ (Petersen et al., 2008) yields:

$$\mathrm{T}_1 = \left(\nabla_{\boldsymbol{z}}\mathrm{J}_{G_{\theta_t}} \otimes \mathrm{J}_{G_{\theta_t}}^{-\mathrm{T}}\right)(\boldsymbol{z}),$$

$$\text{with entries given by}\quad [\mathrm{T}_1]_i = \sum_{j,k}\left[\nabla_{\boldsymbol{z}}\mathrm{J}_{G_{\theta_t}}\right]_{i,j,k}\cdot\left[\mathrm{J}_{G_{\theta_t}}^{-\mathrm{T}}\right]_{j,k};\quad i = 1,2,\ldots,d,$$

where the Hessian tensor can be computed either through automatic differential, or approximated by the Jacobian outer product.

C.5 ScoreGANs with Rectangular Jacobian Matrices

We now extend the results of Appendix C.4 to the scenario when $G_{\theta_t}:\mathbb{R}^d \to \mathbb{R}^n;\ d \ll n$. Papamakarios et al. (2021) showed that when the data $\boldsymbol{x} \in \mathbb{R}^n$ is assumed to lie in a low, d-dimensional manifold by means of the mapping (G_{θ_t}), we can define the metric $\mathrm{M}(\boldsymbol{z})$ induced on the space $\mathcal{X}$ as:

$$\mathrm{M}(\boldsymbol{z}) = \mathrm{J}_{G_{\theta_t}}^{\mathrm{T}}(\boldsymbol{z})\mathrm{J}_{G_{\theta_t}}(\boldsymbol{z}).$$

Then, the change-of-variables formula for the transformation of random variables with measures defined over $\mathcal{X}$ is:

$$p_t(\boldsymbol{x}) = p_z(G_{\theta_t}^{-1}(\boldsymbol{x}))\,\left(\det \mathrm{M}(G_{\theta_t}^{-1}(\boldsymbol{x}))\right)^{-\frac{1}{2}}.$$

An analysis similar to the one provided in Appendix C.4 can now be applied to derive the following approximation:

$$\nabla_{\boldsymbol{x}} \ln\left(p_t(\boldsymbol{x})\right) \approx \mathrm{J}_{G_{\theta_t}}^{\dagger^{\mathrm{T}}}(\boldsymbol{z})\Bigg(-z - \underbrace{\frac{1}{2}\nabla_{\boldsymbol{z}} \ln\det\left(\mathrm{J}_{G_{\theta_t}}^{\mathrm{T}}\mathrm{J}_{G_{\theta_t}}\right)}_{\mathrm{T}_1}\Bigg),$$

where $\mathrm{J}^{\dagger}_{G_{\theta_t}}$ denotes the pseudoinverse of the Jacobian matrix. Further, simplifying T_1 using the standard matrix identity $\nabla_{\boldsymbol{A}} \ln|\det \boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}| = 2\boldsymbol{A}^{\dagger^{\mathrm{T}}}$ (Petersen et al., 2008) yields

$$\begin{aligned} &\mathrm{T}_1 = \nabla_{\boldsymbol{z}} \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z}) \otimes \mathrm{J}^{\dagger^{\mathrm{T}}}_{G_{\theta_t}}(\boldsymbol{z}), \\ \text{with entries given by} \quad &[\mathrm{T}_1]_i = \sum_{j,k} \left[\nabla_{\boldsymbol{z}} \mathrm{J}_{G_{\theta_t}}\right]_{i,j,k} \cdot \left[\mathrm{J}^{\dagger^{\mathrm{T}}}_{G_{\theta_t}}\right]_{j,k}; \quad i = 1, 2, \ldots, d. \end{aligned}$$

While the above result provides a closed-form approximation to the generator density in the most general sense, additional constrained can be enforced on the generator network architecture, as in the case of normalizing flows (Papamakarios et al., 2021) to further simplify computation.

D Optimality of IPM-based GANs

We now derive the IPM GAN counterparts to the proofs presented in Appendix C

D.1 Optimality of Kernel-based IPM-GANs (Proofs of Theorem 4.1 and Lemma 4.2)

Mroueh et al. (2018), in the context of SobolevGAN, showed that IPM-GANs with a gradient-based constraint defined with respect to a base density $\mu(\boldsymbol{x})$ results in the optimal discriminator solving the Fokker-Planck partial differential equation (PDE), given by:

$$\mathrm{div.}\left(\mu\ \nabla D\right)\Big|_{D=D^*_t(\boldsymbol{x})} = \mathrm{c}\left(p_d(\boldsymbol{x}) - p_{t-1}(\boldsymbol{x})\right),$$

where div denotes the divergence operator and c is some constant. For the particular case when the base measure is the uniform, Asokan & Seelamantula (2023a) showed that the PDE simplifies to a Poisson equation, while in the case of higher-order gradient penalties (Adler & Lunz, 2018; Asokan & Seelamantula, 2023b), the optimal discriminator solves an iterated Laplacian, and can be seen as a generalization of SobolevGAN. The optimal discriminator that satisfies the iterated-Laplacian operator was shown to be (Asokan & Seelamantula, 2023b):

$$D^*_t(\boldsymbol{x}) = \mathfrak{C}_\kappa\left((p_{t-1} - p_d) * \kappa\right)(\boldsymbol{x})$$

where $\mathfrak{C}_\kappa = \frac{(-1)^{m+1}\varrho}{2\lambda}$ and ϱ are positive constants, and the kernel κ is the Green's function associated with the differential operator. In Poly-WGAN, the kernel corresponds to the family of polyharmonic splines, given by

$$\kappa(\boldsymbol{x}) = \begin{cases} \|\boldsymbol{x}\|^k & \text{if } k < 0 \text{ or } n \text{ is odd}, \\ \|\boldsymbol{x}\|^k \ln(\|\boldsymbol{x}\|) & \text{if } k \geq 0 \text{ and } n \text{ is even}, \end{cases}$$

where in turn, $k = 2m - n$. The above was also shown to be an m^{th}-order generalization to the Plummer kernel considered in Coulomb GANs (Unterthiner et al., 2018). Given the optimal discriminator, consider the generator optimization. As always, only the terms involving $G(\boldsymbol{z})$ influence the alternating optimization in practice, and the other terms can be ignored. Then, the cost is given by:

$$\mathcal{L}^\kappa_G(G; D^*_t, G_{t-1}) = -\mathop{\mathbb{E}}_{\boldsymbol{z}\sim p_z}\left[D^*_t\left(G(\boldsymbol{z})\right)\right] = -\int_{\mathcal{Z}} D^*_t(G(\boldsymbol{z}))\, p_z(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z}$$

As in the case of f-GANs, we consider the perturbed optimal generator $G^*_{t,i,\epsilon}(\boldsymbol{z})$, and the corresponding cost $\mathcal{L}_{G,i,\epsilon}(\epsilon)$. Substituting in for D^*_t and expanding the convolution integral yields:

$$\mathcal{L}^\kappa_{G,i,\epsilon}(\epsilon) = -\int_{\mathcal{Z}} \mathfrak{C}_\kappa\, p_z(\boldsymbol{z}) \int_{\mathcal{Y}} \left(p_{t-1}(G^*_{t,i,\epsilon}(\boldsymbol{z}) - \boldsymbol{y}) - p_d(G^*_{t,i,\epsilon}(\boldsymbol{z}) - \boldsymbol{y})\right) \kappa(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}\, \mathrm{d}\boldsymbol{z}, \tag{17}$$

where $\mathcal{Y}$ can be viewed as the union of the supports of p_d and p_{t-1} when overlapping, and the convex hull of their supports when non-overlapping. Differentiating the above with respect to ϵ and evaluating at $\epsilon = 0$ gives:

$$\begin{aligned}
\left.\frac{\partial \mathcal{L}^{\kappa}_{G,i,\epsilon}(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} &= -\int_{\mathcal{Z}} \mathfrak{C}_{\kappa}\, p_z(\boldsymbol{z}) \int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \left.\frac{\partial \kappa(G^*_{t,i,\epsilon}(\boldsymbol{z}) - \boldsymbol{y})}{\partial \epsilon}\right|_{\epsilon=0} \mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{z} \\
&= -\int_{\mathcal{Z}} \mathfrak{C}_{\kappa}\, p_z(\boldsymbol{z}) \int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \left.\frac{\partial \kappa(\boldsymbol{w})}{\partial x_i}\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} \frac{\partial [G^*_{t,i,\epsilon}(\boldsymbol{z})]_i}{\partial \epsilon}\, \mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{z} \\
&= -\int_{\mathcal{Z}} \mathfrak{C}_{\kappa}\, p_z(\boldsymbol{z}) \int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \left.\frac{\partial \kappa(\boldsymbol{w})}{\partial w_i}\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} \eta(\boldsymbol{z})\, \mathrm{d}\boldsymbol{y}\,\mathrm{d}\boldsymbol{z} = 0.
\end{aligned}$$

The inner integral once again represents a convolution, given by

$$\left.\frac{\partial \mathcal{L}^{\kappa}_{G,i,\epsilon}(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} = -\mathfrak{C}_{\kappa} \int_{\mathcal{Z}} \left.\left((p_{t-1} - p_d) * \kappa'_i\right)(\boldsymbol{x})\right|_{\boldsymbol{x}=G^*_t(\boldsymbol{z})} p_z(\boldsymbol{z})\eta(\boldsymbol{z})\, \mathrm{d}\boldsymbol{z} = 0,$$

where κ'_i is the derivative of the kernel κ with respect to its i^{th} entry. From the *Fundamental Lemma of Calculus of Variations*, we have

$$\mathfrak{C}_{\kappa} \left.\left((p_{t-1} - p_d) * \kappa'_i\right)(\boldsymbol{x})\right|_{\boldsymbol{x}=G^*_t(\boldsymbol{z})} = 0, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}. \tag{18}$$

Since the above holds for all i, the above can be written mode compactly as

$$\mathfrak{C}_{\kappa} \left.\left((p_{t-1} - p_d) * \nabla_{\boldsymbol{x}}\kappa\right)(\boldsymbol{x})\right|_{\boldsymbol{x}=G^*_t(\boldsymbol{z})} = \boldsymbol{0}, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z},$$

where the convolution between a scalar- and vector-valued function must be interpreted element-wise with respect to the vector. This completes the proof for Lemma 4.2. Table 2 lists a few common kernels used across GAN variants, and their corresponding gradient vectors.

Proof of Theorem 4.1: An alternative approach to solving the aforementioned optimization, is to leverage the properties of convolution in Equation (18). Consider the convolution integral:

$$\begin{aligned}
\left((p_{t-1} - p_d) * \kappa'_i\right)(\boldsymbol{w}) &= \left.\int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \frac{\partial \kappa(\boldsymbol{w})}{\partial w_i}\, \mathrm{d}\boldsymbol{y}\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} \\
&= \left.\frac{\partial}{\partial w_i}\left(\int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \kappa(\boldsymbol{w})\, \mathrm{d}\boldsymbol{y}\right)\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} = 0, \forall\ \boldsymbol{z} \in \mathcal{Z}.
\end{aligned}$$

From the property of convolutions, we have:

$$\begin{aligned}
\left((p_{t-1} - p_d) * \kappa'_i\right)(\boldsymbol{w}) &= \left.\frac{\partial}{\partial w_i}\left(\int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{w}) - p_d(\boldsymbol{w})\right) \kappa(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}\right)\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} \\
&= \left.\left(\int_{\mathcal{Y}} \left(\frac{\partial p_{t-1}(\boldsymbol{w})}{\partial w_i} - \frac{\partial p_d(\boldsymbol{w})}{\partial w_i}\right) \kappa(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}\right)\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} = 0, \forall\ \boldsymbol{z} \in \mathcal{Z}.
\end{aligned}$$

Using the identity $\dfrac{\partial p(\boldsymbol{w})}{\partial w_i} = p(\boldsymbol{w})\dfrac{\partial \ln\left(p(\boldsymbol{w})\right)}{\partial w_i}$, we obtain:

$$\begin{aligned}
\left((p_{t-1} - p_d) * \kappa'_i\right)(\boldsymbol{w}) &= \left.\left(\int_{\mathcal{Y}} \left(\frac{\partial p_{t-1}(\boldsymbol{w})}{\partial w_i} - \frac{\partial p_d(\boldsymbol{w})}{\partial w_i}\right) \kappa(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}\right)\right|_{\boldsymbol{w}=G^*_t(\boldsymbol{z})-\boldsymbol{y}} \\
&= \left(\int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y})\frac{\partial \ln(p_{t-1}(\boldsymbol{y}))}{\partial y_i} - p_d(\boldsymbol{y})\frac{\partial \ln(p_d(\boldsymbol{y}))}{\partial y_i}\right) \kappa(\boldsymbol{x} - \boldsymbol{y})\, \mathrm{d}\boldsymbol{y}\right) = 0,
\end{aligned}$$

for all $\boldsymbol{z} \in \mathcal{Z}$ and $\boldsymbol{x} = G^*_t(\boldsymbol{z})$. Rewriting the integrals as expectations yields

$$\mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_{t-1}}\left[\frac{\partial \ln(p_{t-1}(\boldsymbol{y}))}{\partial y_i}\kappa(G^*_t(\boldsymbol{z}) - \boldsymbol{y})\right] - \mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_d}\left[\frac{\partial \ln(p_d(\boldsymbol{y}))}{\partial y_i}\kappa(G^*_t(\boldsymbol{z}) - \boldsymbol{y})\right] = 0, \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

Table 2: Standard kernels considered in the GAN literature, and their associated gradient fields. While the kernels decay to zero either exponentially, or in polynomial time, the gradient fields induced by them are relatively more stable, owing to the conditioning by $\boldsymbol{x}$. As a result the FloWGAN approaches are stable even in high-dimensional cases, unlike their base-kernel counterparts.

Kernel	$\kappa(\boldsymbol{x})$	Gradient $\nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x})$
Radial basis function Gaussian (RBFG) ($\sigma > 0$)	$\exp\left(-\frac{1}{\sigma^2}\|\boldsymbol{x}\|^2\right)$	$-\frac{1}{\sigma^2}\boldsymbol{x}\exp\left(-\frac{1}{\sigma^2}\|\boldsymbol{x}\|^2\right)$
Mixture of Gaussians (MoG) $\left(\{\sigma_i > 0\}_{i=1}^{\ell}\right)$	$\sum_{\sigma_i}\exp\left(-\frac{1}{\sigma_t^2}\|\boldsymbol{x}\|^2\right)$	$-\boldsymbol{x}\left(\sum_{\sigma_i}\frac{1}{\sigma_i^2}\exp\left(-\frac{1}{\sigma_i^2}\|\boldsymbol{x}\|^2\right)\right)$
Inverse multi-quadric (IMQ) ($c > 0$)	$(\|\boldsymbol{x}\|^2 + c)^{-\frac{1}{2}}$	$-\frac{1}{2}\boldsymbol{x}\,(\|\boldsymbol{x}\|^2 + c)^{-\frac{3}{2}}$
Polyharmonic spline (PHS) ($k < 0$ or n is odd)	$\|\boldsymbol{x}\|^k$	$(k-2)\boldsymbol{x}\|\boldsymbol{x}\|^{k-2}$
Polyharmonic spline (PHS) ($k \geq 0$ and n is even)	$\|\boldsymbol{x}\|^k\ln(\|\boldsymbol{x}\|)$	$\boldsymbol{x}\|\boldsymbol{x}\|^{k-2}\left((k-2)\ln(\|\boldsymbol{x}\|) + 1\right)$

Stacking the above, for all i, as a vector, we obtain:

$$\underset{\boldsymbol{y}\sim p_{t-1}}{\mathbb{E}}\left[\nabla_{\boldsymbol{y}}\ln(p_{t-1}(\boldsymbol{y}))\kappa(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\right] - \underset{\boldsymbol{y}\sim p_d}{\mathbb{E}}\left[\nabla_{\boldsymbol{y}}\ln(p_d(\boldsymbol{y}))\kappa(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\right] = \mathbf{0}, \qquad \forall\ \boldsymbol{z}\in\mathcal{Z}.$$

This completes the proof of Theorem 4.1.

Explaining Denoising Diffusion GANs: To derive a general solution to IPM-GANs (both network-based, or otherwise), consider the discriminator given at iteration t, $D_t(\boldsymbol{x})$. Then, the generator optimization is given by:

$$\mathcal{L}_G^{IPM}(G; D_t, G_{t-1}) = -\underset{\boldsymbol{z}\sim p_z}{\mathbb{E}}\left[D_t\left(G(\boldsymbol{z})\right)\right] = -\int_{\mathcal{Z}} D_t(G(\boldsymbol{z}))\, p_z(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z}$$

The loss defined about the perturbed optimal generator is then given by:

$$\mathcal{L}_{G,i,\epsilon}^{IPM}(\epsilon) = -\int_{\mathcal{Z}} D_t(G_{t,i,\epsilon}^*(\boldsymbol{z}))\,\mathrm{d}\boldsymbol{z}$$

$$\Rightarrow \qquad \left.\frac{\partial\mathcal{L}_{G,i,\epsilon}^{IPM}(\epsilon)}{\partial\epsilon}\right|_{\epsilon=0} = \int_{\mathcal{Z}}\left.\frac{\partial D_t(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} p_z(\boldsymbol{z})\eta(\boldsymbol{z})\,\mathrm{d}\boldsymbol{z} = 0.$$

A similar approach, as in the case of kernel-based IPM-GANs, to simplifying the above for all i, results in the following optimality condition:

$$\nabla_{\boldsymbol{x}} D_t(\boldsymbol{x})\big|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = \mathbf{0}, \quad \forall\,\boldsymbol{z}\in p_z.$$

While the above condition appears trivial in the context of gradient-descent-based training of GANs (as the condition derived is essentially one of gradient descent over the discriminator), it can be used to explain the optimality of GAN based diffusion models such as Denoising Diffusion GANs (DDGAN, Xiao et al. (2022)). In DDGAN, a GAN is trained to approximate the reverse diffusion process, with time-embedding-conditioned discriminator and generator networks. While the approach results in superior sampling speeds as one only needs to sample from the sequence of generators, the underlying transformations that the generated images undergo, can be seen as the flow through the gradient field of the time-dependent discriminator as obtained above.

D.2 Sample Estimate of the FloWGAN Cost (Proof of Lemma 6.2)

The proof follows closely, the approach used in Asokan & Seelamantula (2023b). Consider the optimality condition in FloWGAN along a give dimension i. We have:

$$\left.\mathfrak{C}_\kappa\left((p_{t-1} - p_d) * \kappa_i'\right)(\boldsymbol{x})\right|_{\boldsymbol{x}=G_t^*(\boldsymbol{z})} = 0, \qquad \forall\ \boldsymbol{z}\in\mathcal{Z}.$$

Expanding the convolution integral yields

$$\begin{aligned}
\mathfrak{C}_\kappa \int_{\mathcal{Y}} \left(p_{t-1}(\boldsymbol{y}) - p_d(\boldsymbol{y})\right) \kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\, \mathrm{d}\boldsymbol{y} = 0, &\qquad \forall\ \boldsymbol{z} \in \mathcal{Z} \\
\Rightarrow \int_{\mathcal{Y}} p_{t-1}(\boldsymbol{y})\, \kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\, \mathrm{d}\boldsymbol{y} - \int_{\mathcal{Y}} p_d(\boldsymbol{y})\, \kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\, \mathrm{d}\boldsymbol{y} = 0, &\qquad \forall\ \boldsymbol{z} \in \mathcal{Z} \\
\Rightarrow \mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_{t-1}} \left[\kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\right] - \mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_d} \left[\kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y})\right] = 0, &\qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.
\end{aligned}$$

The above expectation can be replaced with their sample estimates to yield

$$\sum_{\boldsymbol{y}_\ell \sim p_{t-1}} \kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y}_\ell) = \sum_{\boldsymbol{y}_\ell \sim p_d} \kappa_i'(G_t^*(\boldsymbol{z}) - \boldsymbol{y}_\ell), \qquad \forall\ \boldsymbol{z} \in \mathcal{Z}.$$

The above equivalence can be enforced over multiple samples $\boldsymbol{z} \sim p_z$, and for derivatives κ_i', $\forall i$. Replacing the functional form of the generator with a parameterized neural network G_{θ_t} yields

$$\sum_{\boldsymbol{z}_k \sim p_z} \left(\mathrm{dist} \left(\sum_{\boldsymbol{y}_\ell \sim p_d} \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z}_k)-\boldsymbol{y}_\ell}, \sum_{\boldsymbol{y}_\ell \sim p_{t-1}} \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z}_k)-\boldsymbol{y}_\ell} \right), \right)$$

where dist is any chosen distance metric, such as the 1-norm or 2-norm. When we select the 2-norm, we obtain:

$$\sum_{\boldsymbol{z}_k \sim p_z} \left[\left\| \sum_{\boldsymbol{y}_\ell \sim p_{t-1}} \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z}_k)-\boldsymbol{y}_\ell} - \sum_{\boldsymbol{y}_\ell \sim p_d} \nabla_{\boldsymbol{x}} \kappa(\boldsymbol{x})|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z}_k)-\boldsymbol{y}_\ell} \right\|_2^2 \right],$$

which concludes the proof of Lemma 6.2.

D.3 Convergence of Discriminator-guided Langevin Diffusion

An in-depth analysis of the convergence of discriminator-guided Langevin diffusion, from the point of view of stochastic differential equations (SDEs), is out of score for this paper. However, (Lunz et al., 2018), in the context of adversarial regularization for inverse problems, have extensively analyzed the following iterative algorithm:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \nabla_{\boldsymbol{x}} D_{t,\theta}^*(\boldsymbol{x}),$$

where η is the learning rate, and $D_{t,\theta}^*(\boldsymbol{x})$ denotes the optimal discriminator at time t parameterized by θ. In particular, they show that (Lunz et al. (2018), Theorem 1):

$$\frac{\partial}{\partial \eta} \mathcal{W}(p_d, p_t) = - \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p_{t-1}} \left[\|\nabla_{\boldsymbol{x}} D_{t,\theta}^*(\boldsymbol{x})\|_2^2 \right],$$

where $\mathcal{W}$ denotes the Wasserstein-1 (earth mover's) distance. This shows that, the updated distribution p_t is closer in Wasserstein distance to the target distribution p_d, in comparison to p_{t-1}. For functions with $\|\nabla_{\boldsymbol{x}} D_{t,\theta}^*(\boldsymbol{x})\| = 1$, which is the condition under which the gradient-regularized GANs have been optimized, we have the decay $\frac{\partial}{\partial \eta}\mathcal{W}(p_d, p_t) = -1$. While we consider the updates

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_t \nabla_{\boldsymbol{x}} D_t^*(\boldsymbol{x}_t) + \gamma_t \boldsymbol{z}_t$$

in discriminator-guided Langevin diffusion, we will show, experimentally, that the update scheme $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_0 \nabla_{\boldsymbol{x}} D_t^*(\boldsymbol{x}_t)$ indeed performs the best, on image datasets (cf. Appendix F).

E Additional Experimentation on Score- and Flow-matching GANs

In this appendix, we present additional results and ablation experiments from training ScoreGAN and FloWGAN on Gaussian and Image data.

E.1 Additional Experimental Results on Gaussian Learning

Training Parameters: All models are trained using the TensorFlow (Abadi et al., 2016) library. On the unimodal Gaussian experiments, the generator is a linear transformation $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{z} + \boldsymbol{b}$. The target Gaussian is $p_d = \mathcal{N}(5\mathbf{1}_2, 0.75\mathbb{I}_2)$ in 2-D and $p_d = \mathcal{N}(0.7\mathbf{1}_n, 0.02\mathbb{I}_n)$ in the n-D case for $n > 2$. In baseline GAN variants with a network-based discriminator, we use a four-layer perceptron architecture, with 128, 32, 16, and 1 node(s), respectively in each layer. The Leaky-ReLU activation is used across all layers. The networks are trained with the Adam (Kingma & Ba, 2015) optimizer. A batch size of 500 is used. The models are compared using the Wasserstein-2 distance between the target and source Gaussians $\mathcal{W}^{2,2}(p_d, p_g) = \|\boldsymbol{\mu}_d - \boldsymbol{\mu}_g\|_2^2 + \mathrm{Trace}\left(\Sigma_d + \Sigma_g - 2\sqrt{\Sigma_d\Sigma_g}\right)$. On the Gaussian-mixture model (GMM) learning tasks, the generator is a three-layer perceptron architecture, with 32, 16, and 2 node(s), respectively in each layer. The input dimensionality is 100 for all the baseline variants and FloWGAN. For ScoreGAN, we compare against both a 2-D input (resulting in a square, invertible Jacobian), and a 100-D input (resulting in a rectangular Jacobian matrix).

Additional results on Gaussian and GMM learning: On the GMM learning task, we consider ablation experiments on training ScoreGAN with, and without, the rectangular Jacobian. In the scenario where the input and output dimensions match, ScoreGAN fails to converge, and the network has insufficient capacity to map an unimodal Gaussian to a multimodal one. Figure 6 presents the generator and data distributions, superimposed on the gradient field over which the generator is optimized, for various baseline variants, ScoreGAN and FloWGAN. In the case of the baselines, this corresponds to the gradient of the discriminator, while in ScoreGAN, it is the score of the target dataset. In FloWGANs, it is the gradient of the polyharmonic spline kernel. Table 3 compares the *Batch Compute Time* between generator updates for the baseline GANs, ScoreGAN and FloWGAN. ScoreGANs are more compute-intensive compared to FloWGANs due to the need for computing the score of the generator network in each update step. The computational complexity of FloWGAN is on par with that of kernel-based models such as generative moment-matching networks (GMMNs) or Poly-WGAN

Choice of the FloWGAN kernel: Besides the PHS kernel, we also consider the radial basis function Gaussian (RBFG) and inverse multi-quadric kernels, as described in Table 2. As noted in the case of MMD-GANs (Li et al., 2017), the Gaussian kernel is sensitive to the scale parameter. Therefore, we consider two scenarios: (a) A single Gaussian kernel with $\sigma = 1$; and (2) A mixture of five kernels with scale parameters $\sigma \in \{0.5, 1, 2, 4, 8\}$. Figure 7 depicts the target and generated samples overlaid on the gradient field. FloWGAN with a single Gaussian kernel collapses to a region about the mean of the dominating mode of the target. While the gradients in the IMQ kernel decay in regions far away from both p_d and p_g, the gradient fields of the PHS and the *mixture of Gaussians* kernels is comparable. Since the polyharmonic function is not sensitive to a scale parameter, it converges to the target reliably for any input dynamic range. We therefore consider the PHS kernel in all experiments presented in Section 7 and Appendices E.2 and F.

E.2 Additional Experimental Results on Image Learning

We present two sets of results: (a) Image space learning with FloWGANs; (b) Additional results on latent-space learning with FloWGANs.

Training Parameters: For image-space learning, we consider the standard deep convolutional GAN (DCGAN, Radford et al. (2016)) architecture, trained on the MNIST and CelebA datasets. The CelebA images are center-cropped to $140 \times 140 \times 3$, and resized to $32 \times 32 \times 3$ though bilinear interpolation. For the latent-space learning tasks, similar to the approach considered in Poly-WGAN (Asokan & Seelamantula, 2023b), we train a deep convolutional autoencoder to learn an n_d latent-space distribution of MNIST, SVHN, and CelebA datasets. We consider a 16-D latent space for MNIST, 32-D latent space of SVHN, and 63-D latent space of CelebA. The generator in the baseline GANs and FloWGAN are subsequently trained to learn a mapping for a 100-D Gaussian to the latent-space of the images. The generators employ a four-layer perceptron architecture, with 512, 256, 128 nodes

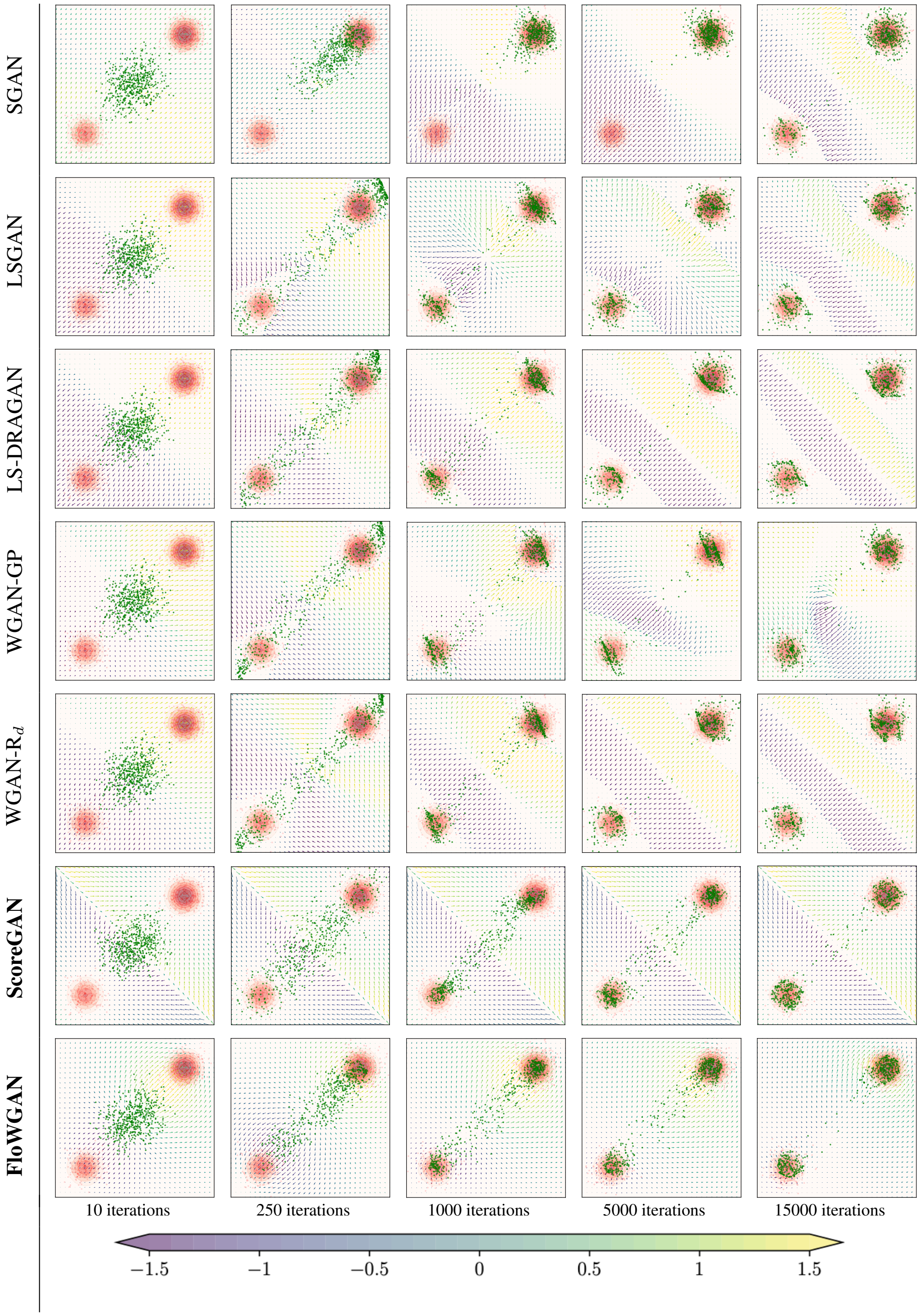


Figure 6: (Color online) Convergence of the generator samples (shown in green) to the target two-component Gaussian (shown in red), $p_d(\boldsymbol{x}) = \frac{1}{5}\mathcal{N}(\boldsymbol{x}; -5\mathbf{1}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(\boldsymbol{x}; 5\mathbf{1}, \mathbb{I})$. The quiver plot depicts the gradient field of the discriminator on baseline variants, and the gradient of the PHS kernel convolved with the density difference in FloWGAN and the score of the dataset in the case of ScoreGAN. While SGAN collapses to the more pronounced mode, FloWGAN and ScoreGAN converges to both the modes accurately, faster than the baseline counterparts.

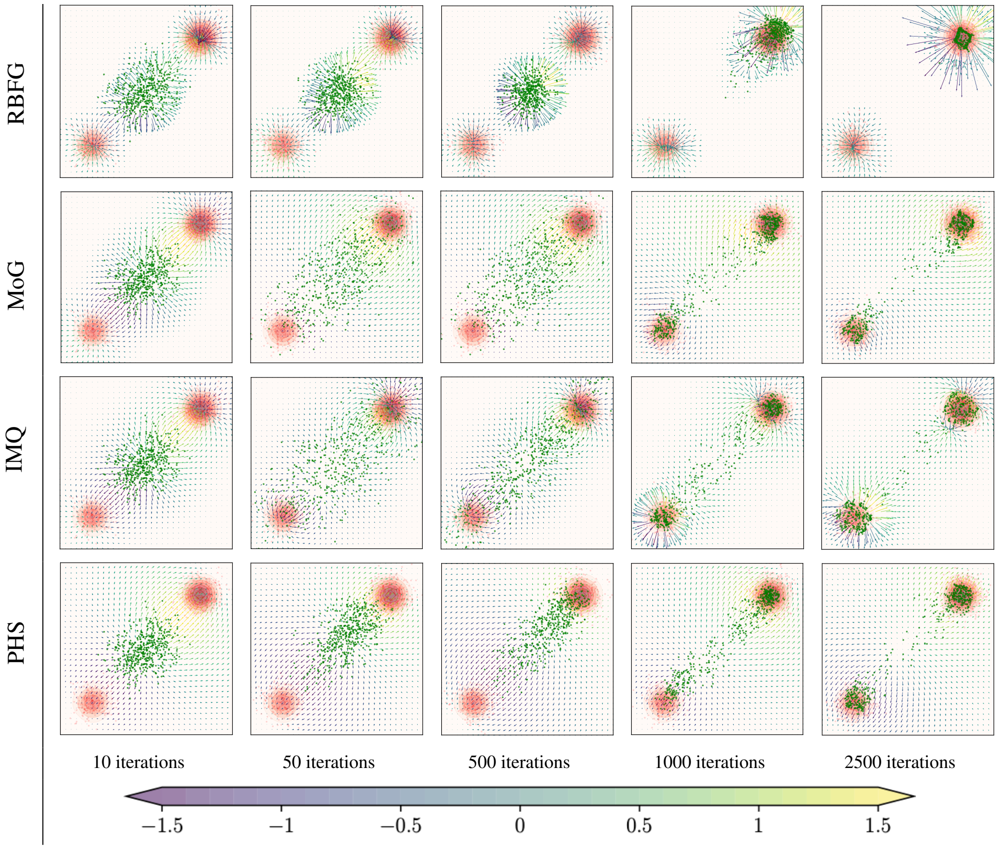


Figure 7: (Color online) Convergence of the generator samples (shown in green) to the target two-component Gaussian (shown in red), $p_d(\boldsymbol{x}) = \frac{1}{5}\mathcal{N}(\boldsymbol{x}; -5\mathbf{1}, \mathbb{I}) + \frac{4}{5}\mathcal{N}(\boldsymbol{x}; 5\mathbf{1}, \mathbb{I})$ considering various choices of the kernel function in FloWGAN. The quiver plot depicts the gradient field of the kernel convolved with the density difference. The single-component Gaussian kernel (RBFG) performs poorly if the chosen scale does not match the scale of the data. The mixture of Gaussians (MoG) kernel (Li et al., 2017) alleviates this issue. FloWGANs with the MoG, inverse multiquadric (IMQ) and Polyharmonic spline (PHS) kernel converge to the target data accurately.

in the first three layers, and n_d nodes in the output layer. The batch size is set to 100 across all experiments, while the Adam optimizer is used to train the networks. The discriminator in baseline GANs is identical to the Gaussian learning case.

Experimental Results: Figure 8 presents the images generated by FloWGAN when trained on MNIST and CelebA. We observe that FloWGAN is capable of learning on the image-space for relatively simple distribution such as MNIST. The performance of FloWGAN is superior to kernel-based counterparts such as GMMNs and Poly-WGAN. On CelebA, since the base Gaussian and PHS kernels do not scale well with the data dimensionality, the baselines fail to generate realistic images. The gradient of the PHS kernel considered in FloWGAN scales more favorably. While the kernels decay to zero either exponentially, or in polynomial time, the gradient fields induced by them are relatively more stable, owing to the conditioning by $\boldsymbol{x}$. Therefore, FloWGANs are stable even in high-dimensional cases. However, all kernel-based methods are sub-par, compared to the baseline DCGAN. We attribute this to the *curse of dimensionality* – the number of samples required to approximate the convolution grows as $\mathcal{O}(n)$ for data $\boldsymbol{x} \in \mathbb{R}^n$.

Table 4 presents the FID achieved by the best-case models when trained on the latent-space of the images. On low-dimensional latent spaces such as MNIST, FloWGAN is on par with Poly-WGAN, while outperforming the baseline GANs. However, on higher-dimensional latent spaces as in the case of CelebA, FloWGAN is superior to the baselines. To generate more realistic images, one could consider a flipped scenario, wherein Langevin update steps are used to generate realistic images based on the gradient field considered in FloWGAN. This is a promising direction for future research.

Table 3: A comparison of baseline GAN variants and FloWGAN in terms of their training time (measured in seconds per batch) on Gaussian learning tasks. ScoreGANs are more compute-intensive compared to FloWGANs due to the need for computing the score of the generator network in each update step. For image data, and DCGAN architecture, the *change-of-variables* approach to computing the generator score becomes impractical. The computational complexity of FloWGAN is on par with that of kernel-based models such as generative moment-matching networks (GMMNs) or Poly-WGAN, as only the norm-based kernel computations are required.

GAN Variant	Batch Compute Time (seconds/batch)	
	2-D data	128-D data
	Batch size 500	Batch size 100
SGAN	0.4651 ± 0.023	0.2031 ± 0.023
LSGAN	0.4622 ± 0.021	0.1973 ± 0.031
LS-DRAGAN	0.4854 ± 0.020	0.2066 ± 0.019
WGAN-GP	0.4553 ± 0.031	0.1849 ± 0.031
WGAN-R$_d$	0.4427 ± 0.032	0.1932 ± 0.022
Poly-WGAN	0.2316 ± 0.012	0.1571 ± 0.020
GMMN (RBFG)	0.2015 ± 0.020	0.1881 ± 0.031
GMMN (IMQ)	0.1981 ± 0.021	0.1579 ± 0.019
ScoreGAN (Ours)	0.3222 ± 0.022	1.1178 ± 0.015
FloWGANs (Ours)	$\mathbf{0.1152 \pm 0.011}$	$\mathbf{0.1379 \pm 0.015}$

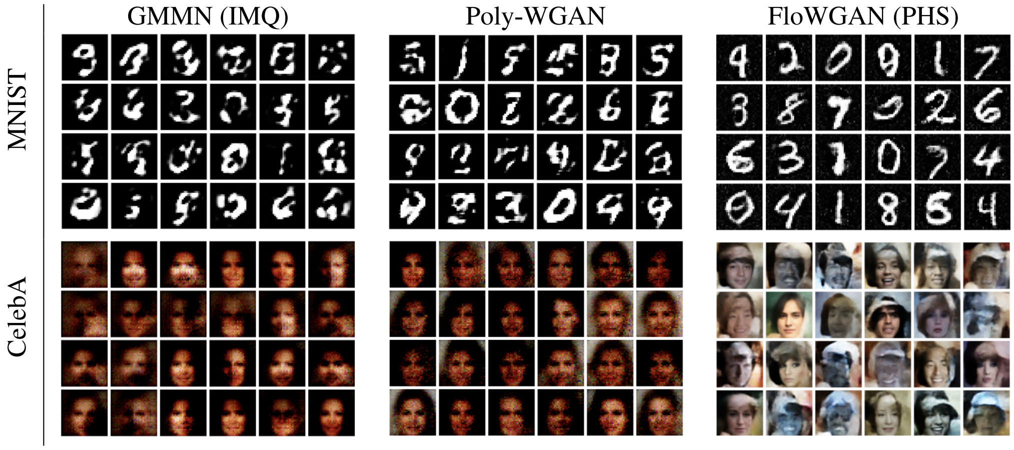


Figure 8: Images generated by GMMN-IMQ and Poly-WGAN and the proposed FloWGAN (PHS) on MNIST and CelebA image-space learning task. While FloWGAN is superior to the baselines in MNIST generation, the generated images are relatively noisy (best seen when zoomed-in on the *.pdf*). On CelebA, since the base Gaussian and PHS kernels do not scale well with the data dimensionality, the baselines fail to generate realistic images. The gradient of the PHS kernel considered in FloWGAN scales more favorably. However, the quality of images generated are sub-par compared to baseline GANs. We attribute this to the *curse of dimensionality* – the number of samples required to approximate the convolution grows as $\mathcal{O}(n)$ for data $\boldsymbol{x} \in \mathbb{R}^n$.

E.3 Training Algorithm

The training procedure for ScoreGANs and FloWGANs are presented in Algorithms 1 and 2, respectively. ScoreGANs are computationally more intensive to train, due to the need to compute the Jacobian of the generator at each training step. Consequently, ScoreGANs do not scale to high-dimensions. FloWGAN train the generator on the kernel-based loss given in

Table 4: A comparison of Baseline GAN variants and FloWGAN in terms of the FID metric. Poly-WGAN implements the optimal polyharmonic spline (PHS) discriminator by means of a radial basis function networks, while the other baselines train a discriminator network. Poly-WGAN outperforms FloWGAN with the PHS kernel on MNIST. However, the FloWGAN approach scales more favorably to higher-dimensions, results in superior performance on SVHN and CelebA

WGAN flavor	MNIST (16-D)	SVHN (32-D)	CelebA (63-D)
SGAN	21.24	53.561	49.840
WGAN-GP	19.441	51.241	49.840
WGAN-LP	17.825	50.342	50.694
WGAN-R$_d$	17.948	49.231	48.064
WGAN-R$_g$	18.498	52.321	51.104
Poly-WGAN	**17.397**	48.341	45.886
FloWGAN **(Ours)**	17.492	**47.980**	**42.263**

Algorithm 1: ScoreGAN – Training the GAN generator trained to minimize the distance between its score and the score of the data.

```
Input: Training data $\boldsymbol{x} \sim p_d$, Gaussian prior distribution $p_z = \mathcal{N}(\mu_{\boldsymbol{z}}, \Sigma_{\boldsymbol{z}})$, Max training
        iterations $T$.
Parameters: Batch size $M$, optimizer learning rate $\eta$, number of radial basis function (RBF)
centers $N$, discriminator kernel $\kappa$, kernel order $k$.
Models: Generator: $\mathrm{G}_\theta$; Data score model: $S^d_\phi = \nabla_{\boldsymbol{x}} \ln\left(p_d(\cdot\,;\,\phi)\right)$.
while $t$ = 1,2,...,$T$ do
    Sample: $\boldsymbol{z}_\ell \sim p_z$ – A batch of $M$ noise samples.
    Sample: $\boldsymbol{x}_\ell = \mathrm{G}_{\theta_t}(\boldsymbol{z}_\ell)$ – Generator output samples.
    Compute: $\mathrm{J}_{G_{\theta_t}}(\boldsymbol{z}_\ell)$ – Jacobian of the generator evaluated at $\boldsymbol{z}_\ell$.
    Compute: $\nabla_{\boldsymbol{x}} \ln p_t$ – Score of the generator evaluated at $G_{\theta_t}(\boldsymbol{z}_\ell)$ (cf. Lemma 3.3):
        $\nabla_{\boldsymbol{x}} \ln p_t(\boldsymbol{x})|_{\boldsymbol{x}=\boldsymbol{x}_\ell} = -\mathrm{J}^{-\mathrm{T}}_{G_{\theta_t}}\left(\nabla_{\boldsymbol{z}} \ln|\det \mathrm{J}_{G_{\theta_t}}(\boldsymbol{z}_\ell)| + \boldsymbol{z}_\ell\right)$,
    Compute: Score-matching-based generator loss (cf. Section 6):
        $\mathcal{L}^{\mathrm{Score}}_G(\theta_t) = \sum_{\boldsymbol{x}_\ell} \nabla_{\boldsymbol{x}} \| \ln\left(p_t(\boldsymbol{x}_\ell)\right) - S^d_\phi(\boldsymbol{x}_\ell)\|^2_2$.
    Update: Generator $\mathrm{G}_{\theta_{t+1}} : \theta_{t+1} = \eta \nabla_\theta [\mathcal{L}^{\mathrm{Score}}_G(\theta)]|_{\theta=\theta_t}$ – Generator at $\theta_{t+1}$ is the one
    that minimizes the score matching loss of the generator at $\theta_t$
Output: Samples output by the Generator: $\boldsymbol{x} = \mathrm{G}_{\theta_T}(\boldsymbol{z})$
```

F Additional Experimentation on Discriminator-guided Langevin Sampling

We present additional experimental results on generating 2-D shapes, and image data using the discriminator-guided Langevin sampler.

F.1 Additional Experimental Results on Synthetic Data Learning

On the 2-D learning task, we present additional combinations on the *shape morphing experiment*.

Training Parameters: All samplers are implemented using the TensorFlow (Abadi et al., 2016) library. The discriminator gradient is built as a custom radial basis function network, whose weights and centers are assigned at each iteration. At $t = 0$, the centers $\boldsymbol{g}^j \sim p_{t-1}$ are sampled from the unit Gaussian, *i.e.,* $p_{-1} = \mathcal{N}(\mathbf{0}, \mathbb{I})$. In subsequent iterations, the batch of samples from time instant $t-1$ serve as the centers for D^*_t. Based on experiments presented in Appendix F.2, we set $\gamma_t = 0$ and $\alpha_t = 1 \ \forall\ t$. The input and target distributions are created following the approach presented by (Mroueh & Rigotti, 2020). Given an 8-bit grayscale input (output) image $\mathrm{I}(p, q)$ (cf. Figure 9), the input (output) dataset consists of points drawn uniformly from the regions of the image where $\mathrm{I}(p, q) < 128$.

Algorithm 2: FloWGAN − GAN with the generator trained to minimize the flow-filed induced by the gradient of the discriminator kernel.

```
Input: Training data $\boldsymbol{x} \sim p_d$, Gaussian prior distribution $p_z = \mathcal{N}(\mu_{\boldsymbol{z}}, \Sigma_{\boldsymbol{z}})$, Max training
        iterations $T$.
Parameters: Batch size $M$, optimizer learning rate $\eta$, number of radial basis function (RBF)
centers $N$, discriminator kernel $\kappa$, kernel order $k$.
Models: Generator: $\mathrm{G}_\theta$.
while $t$ = 1,2,...,$T$ do
    Stored Data: Samples from the Generator at $t-1$:
    $\{\boldsymbol{y}_i \sim p_{t-1}\ ;\ \boldsymbol{y}_i = G_{\theta_{t-1}}(\boldsymbol{z}_i),\ \boldsymbol{z}_i \sim p_z\}$
    Sample: $\boldsymbol{z}_\ell \sim p_z$ – A batch of $M$ noise samples.
    Sample: $\mathrm{G}_{\theta_t}(\boldsymbol{z}_\ell)$ – Generator output samples.
    Sample: $\tilde{\boldsymbol{y}}_j \sim p_d$ – A batch of $N$ target data reference locations.
    Sample: $\boldsymbol{z}_i \sim p_z$ – A batch of $M$ reference noise samples.
    Compute: Kernel-gradient-based generator loss (cf. Lemma 4.2):
        $\mathcal{L}_G^{\mathrm{FloW}}(\theta_t) = \mathbb{E}_{\boldsymbol{z}\sim p_z}\left[ \left\| \sum_{\boldsymbol{y}\sim p_{t-1}} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x}) - \sum_{\boldsymbol{y}\sim p_d} \nabla_{\boldsymbol{x}}\kappa(\boldsymbol{x}) \right\|_2^2 \Bigg|_{\boldsymbol{x}=G_{\theta_t}(\boldsymbol{z})-\boldsymbol{y}} \right]$ .
    Update: Generator $\mathrm{G}_{\theta_{t+1}} : \theta_{t+1} = \theta_t + \eta\nabla_\theta[\mathcal{L}_G^{\mathrm{FloW}}(\theta)]|_{\theta=\theta_t}$ – Generator at $\theta_{t+1}$ is the
    one whose samples are pulled towards $p_d$, and pushed away from $p_{t-1}$
Output: Samples output by the Generator: $\boldsymbol{x} = G_{\theta_T}(\boldsymbol{z})$
```

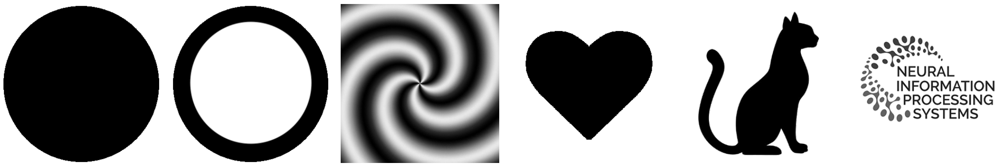


Figure 9: (● Color online) Images considered in generating the source and target in the *Shape morphing* experiment.

Experimental Results: We consider the *Heart* and *Cat* shapes as the target, while considering various input shapes, corresponding to varying levels of difficulty in matching to the target. In the case of learning the *Heart* shape, for input shapes that do not contain *holes*, the convergence is relatively quick, and the shape matching occurs in about 100 to 250 iteration. For more challenging input shapes, such as the *Cat* or the *NeurIPS* logo, the discriminator-guided Langevin sampler converges in about 500 iterations. This is superior to the reported 800 iterations in the Unbalanced Sobolev descent formulation. The results are similar in the case where the *Cat* image is the target (cf. Figure 10).

F.2 Additional Experimental Results on Image Learning

We present ablation experiments on generating images with the discriminator-guided Langevin sampler to determine the choice of α_t and γ_t in the update regime. We also provide additional images pertaining to the experiments presented in the *Main Manuscript*

Choice of coefficients α_t ***and*** γ_t: For the ablation experiments, we consider MNIST, SVHN, and 64-dimensional CelebA images. Based on the ablation experiments presented in Appendix E.1, we consider the kernel-based discriminator with the polyharmonic spline kernel in all subsequent experiments. Recall the update scheme:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-1} - \alpha_t \nabla_{\boldsymbol{x}} D_t^*(\boldsymbol{x}_t;\, p_{t-1}, p_d) + \gamma_t \boldsymbol{z}_t, \quad \text{where} \quad \boldsymbol{z}_t \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$$

Based on the observations made by Karras et al. (2022), to ascertain the optimal choice of the coefficients, we consider the scenarios:

- **The ordinary differential equation (ODE) formulation**, wherein the noise perturbations are ignored, giving rise to and ODE that the samples are evolved through. Here $\gamma_t = 0,\ \forall\, t$.
- **The stochastic differential equation (SDE) formulation**, wherein we retain the noise perturbations. Based on the links between score-based approaches and the GANs, we

consider the approach presented in noise-conditioned score networks (NCSNv1) (Song & Ermon, 2019), with $\gamma_t = \sqrt{2\alpha_t}$.

Within these two scenarios, we further consider the following subcases:

- **Unadjusted Langevin dynamics (ULD)**, wherein α_t is made static, *i.e.,* $\alpha_t = \alpha_0,\ \forall\, t$.
- **Annealed Langevin dynamics (ALD)**, wherein α_t is decayed. While various approaches have been proposed for scaling (Song & Ermon, 2019, 2020; Song et al., 2021b; Jolicoeur-Martineau et al., 2021; Karras et al., 2022), we consider the geometric decay considered in NCSNv1 (Song & Ermon, 2019).

For either case, we present results considering $\alpha_0 \in \{100, 10, 1\}$.

Figures 11–13 present the images generated by the discriminator-guided Langevin sampler on MNIST, SVHN and CelebA, respectively, for the various scenarios considered. Across all datasets, we observe that annealing the coefficients results in poor convergence. We attribute this to the fact that the polyharmonic kernel, being a distance function, decays *automatically* as the iterates converse, *i.e.,* as p_t converges to p_d. Consequently, the magnitude of the discriminator gradient, in the case when α_t is decays, is too small to significantly move the particles along the discriminator gradient field. Next, we observe that for relatively small $\alpha_0 \leq 10$, the samplers converge to realistic images. When α_0 is large, the resulting *gradient explosion* during the initial steps of the sampler results in *mode-collapse* in all scenarios. Thirdly, in choosing $\boldsymbol{z}_t$, the experimental results indicate that the model converges to visually superior images when $\boldsymbol{z}_t = 0$. For the scenarios where α_t as the coefficient of $\nabla_{\boldsymbol{x}} D_t^*$ is kept constant, the coefficient $\gamma_t = \sqrt{2\alpha_t}$ continues to be decays. When $\boldsymbol{z}_t$ is non-zero, the generated images are noisy. We attribute the convergence of the discriminator-guided Langevin sampler to unique samples even in scenarios when $\boldsymbol{z}_t$ is zero, to the implicit randomness (of RBF centers) introduced by the sample-estimation of convolution in the discriminator D_t^*.

The superior convergence of the proposed approach is further validated by the *iterate convergence* presented in Figure 18. We compare discriminator-guided Langevin sampler, with $\alpha_t = \alpha_0 = 10$, with and without noise perturbations $\boldsymbol{z}_t$, against the base NCSN model, owing to the links to the score-based results derived in ScoreGANs and FloWGANs. We plot $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_2^2$ as a function of iteration t for the MNIST learning task. In NCSN, the iterates converge at each noise level, and subsequently, when the noise level drops, the sample quality improved. This is consistent with the observations made by Song & Ermon (2020), who showed that the score network S_θ implicitly scales its output by the noise variance σ. The proposed approach, with $\boldsymbol{z}_t = 0$, performs the best.

Uniqueness of generated images: As the kernel-based discriminator directly operates on the target data, drawing batches of samples as centers in the RBF interpolator, an obvious question to ask is whether the discriminator-guided Langevin iterations convergence to unique samples *not present in the dataset*. To verify this, we perform a k-nearest neighbor analysis, considering $k = 9$. Figures 14–16 present the top-k neighbors of samples generated by the proposed images from each digit class of MNIST, SVHN, and CelebA datasets. The neighbors are found across all *digit* classes in the case of MNIST and SVHN. We observe that the proposed approach **does not** memorize the target data. In the case of SVHN, considering the sample generated from *digit class 5* of *digit class 9*, we observe that the nearest neighbor is from a different class, indicative of the sampler's ability to interpolate between the classes seen as part of discriminator centers during sampling.

Details on the experiment present in Section 7.2 of the Main Manuscript: Figure 17 presented the images, considering the Langevin sampler with $\alpha_t = \alpha_0 = 10$ with $\boldsymbol{z}_t = 0$. Across all three datasets, we observe that the models converge to nearly realists samples in about $t = 500$ iterations, while subsequent iterations serve to *denoise* the images. Animations pertaining to these iterations are provided as part of the supplementary material.

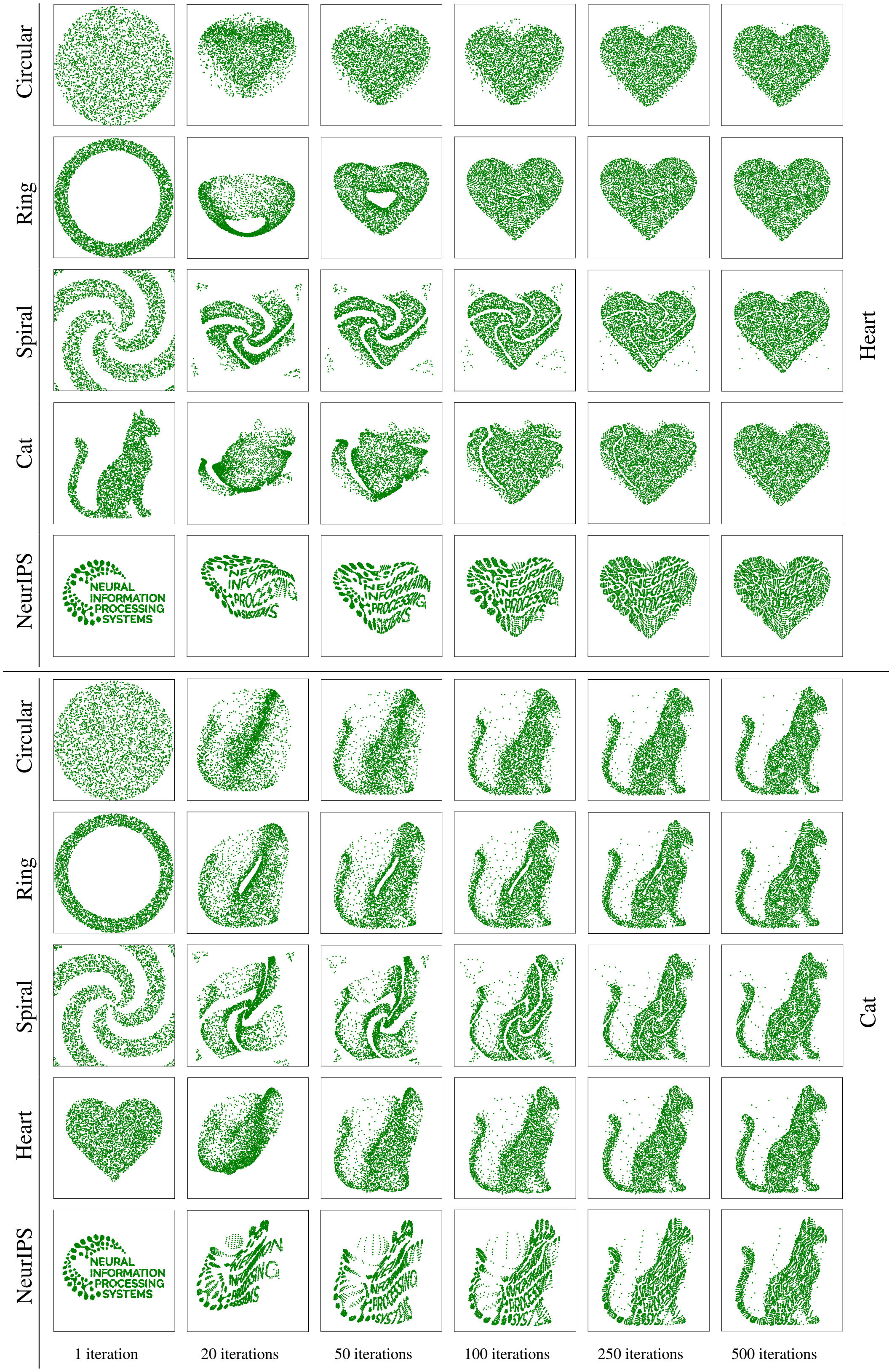


Figure 10: (Color online) Samples across iteration for the discriminator-guided Langevin sampler, considering various shapes of the initial uniform distributions, given a target uniform distribution shaped like a *Heart*, or a *Cat*. For relatively simpler input shapes, such as the circular pattern, the sampler converges in about 100 iterations, while in the spiral case, the sampler converges in about 250 steps.

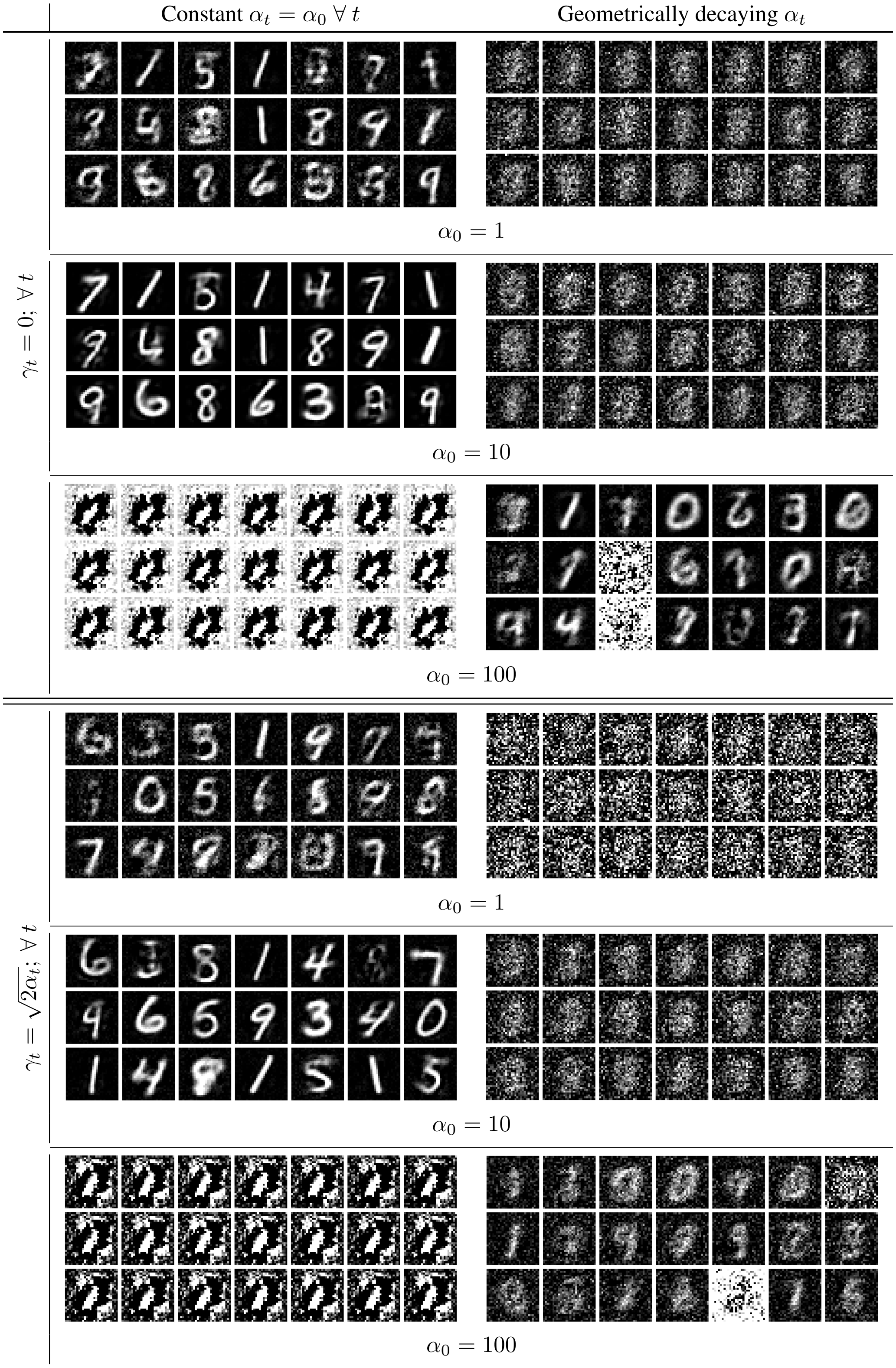

Constant $\alpha_t = \alpha_0 \; \forall \, t$
Geometrically decaying α_t
$\gamma_t = 0; \; \forall \, t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$
$\gamma_t = \sqrt{2\alpha_t}; \; \forall \, t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$

Figure 11: (Color online) Images generated using the discriminator-guided Langevin sampler when samples using the MNIST dataset as the target. The model fails to converge when α_t decays, for small $\alpha_0 \leq 10$. For the case when $\alpha_0 = 100$, some samples diverge as a consequence of gradient explosion. We observe that $\alpha_0 = 10$, with $\boldsymbol{z}_t = 0$ yields the best performance.

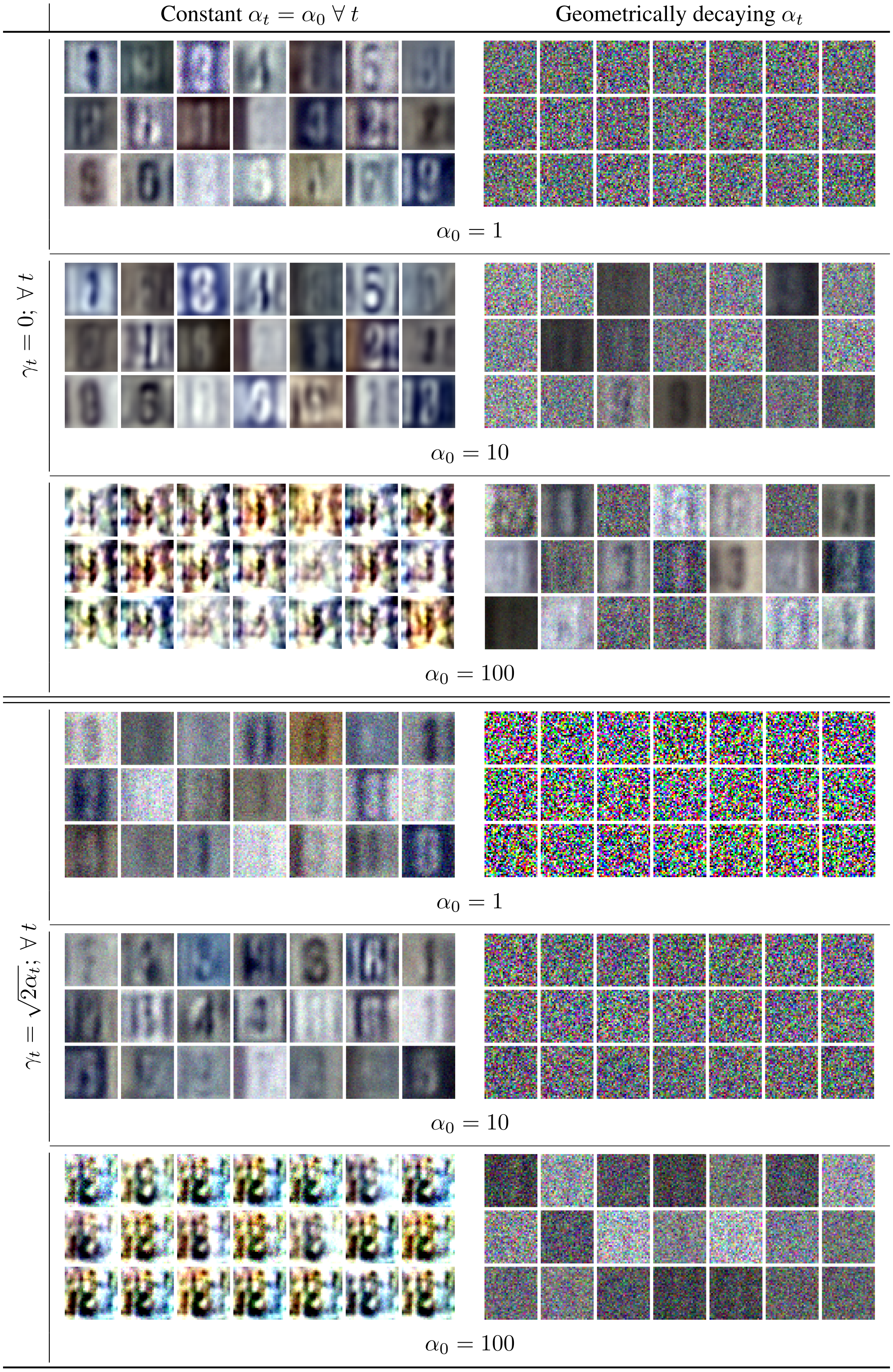

Constant $\alpha_t = \alpha_0\ \forall\ t$
Geometrically decaying α_t
$\gamma_t = 0;\ \forall\ t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$
$\gamma_t = \sqrt{2\alpha_t};\ \forall\ t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$

Figure 12: (Color online) Images generated using the discriminator-guided Langevin sampler when samples using the SVHN dataset as the target. The model fails to converge when geometrically decaying α_t decays, or when $\boldsymbol{z}_t$ is non-zero. As in the MNIST case observe that $\alpha_0 = 10$, with $\boldsymbol{z}_t = 0$ yields the best performance. Setting $\alpha_0 = 1$ with $\boldsymbol{z}_t = 0$ results in slow convergence.

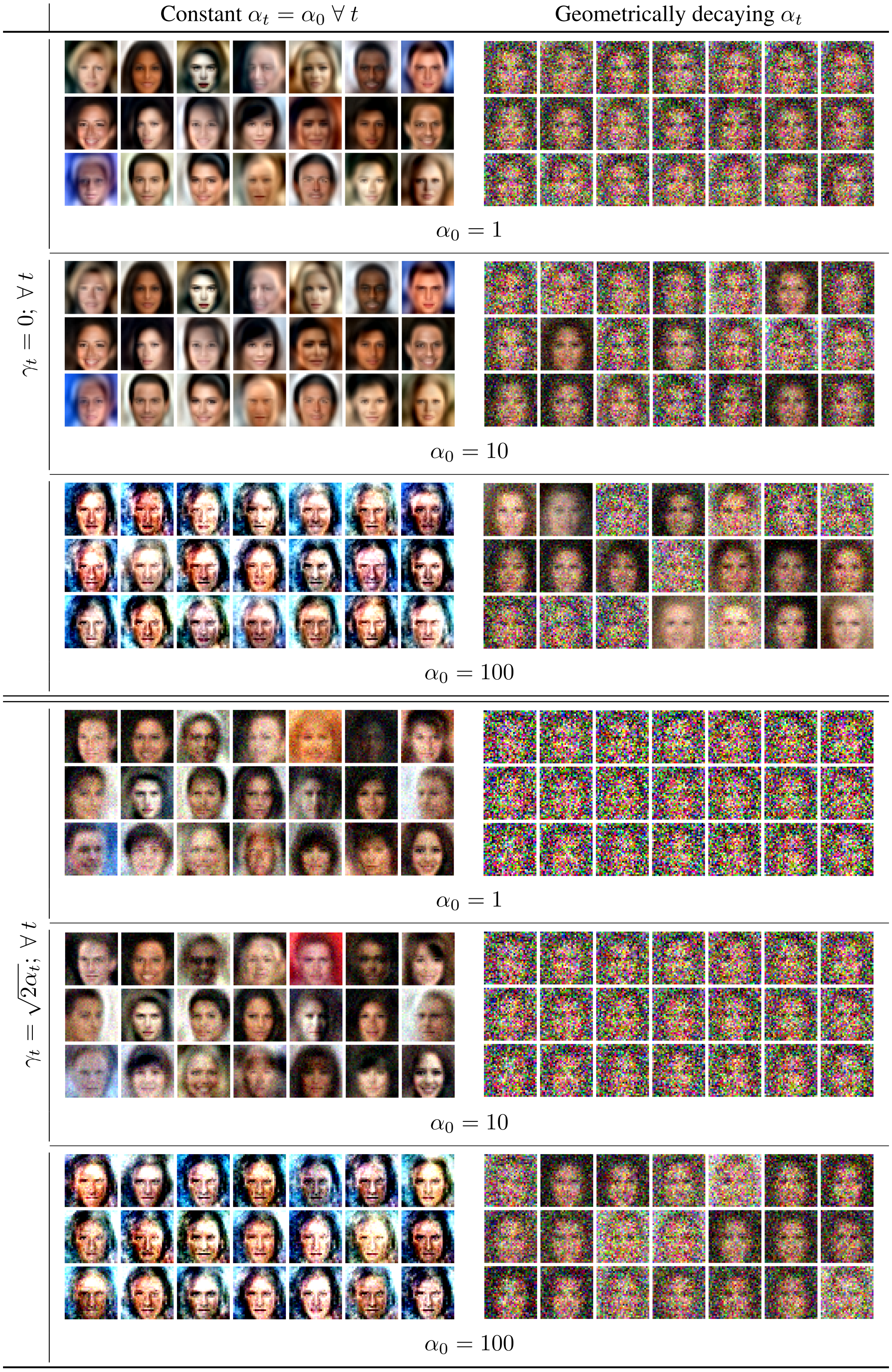

Constant $\alpha_t = \alpha_0 \, \forall \, t$
Geometrically decaying α_t
$\gamma_t = 0; \, \forall \, t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$
$\gamma_t = \sqrt{2\alpha_t}; \, \forall \, t$
$\alpha_0 = 1$
$\alpha_0 = 10$
$\alpha_0 = 100$

Figure 13: (Color online) Images generated using the discriminator-guided Langevin sampler when samples using the CelebA dataset as the target. The model fails to converge when geometrically decaying α_t decays, or when $\boldsymbol{z}_t$ is non-zero. Setting $\alpha_0 \in [1, 10]$, with $\boldsymbol{z}_t = 0$ results in the sampler generating realistic images. For these choices of α_0, when $\boldsymbol{z}_t$ is non-zero, the generated images are noisy.

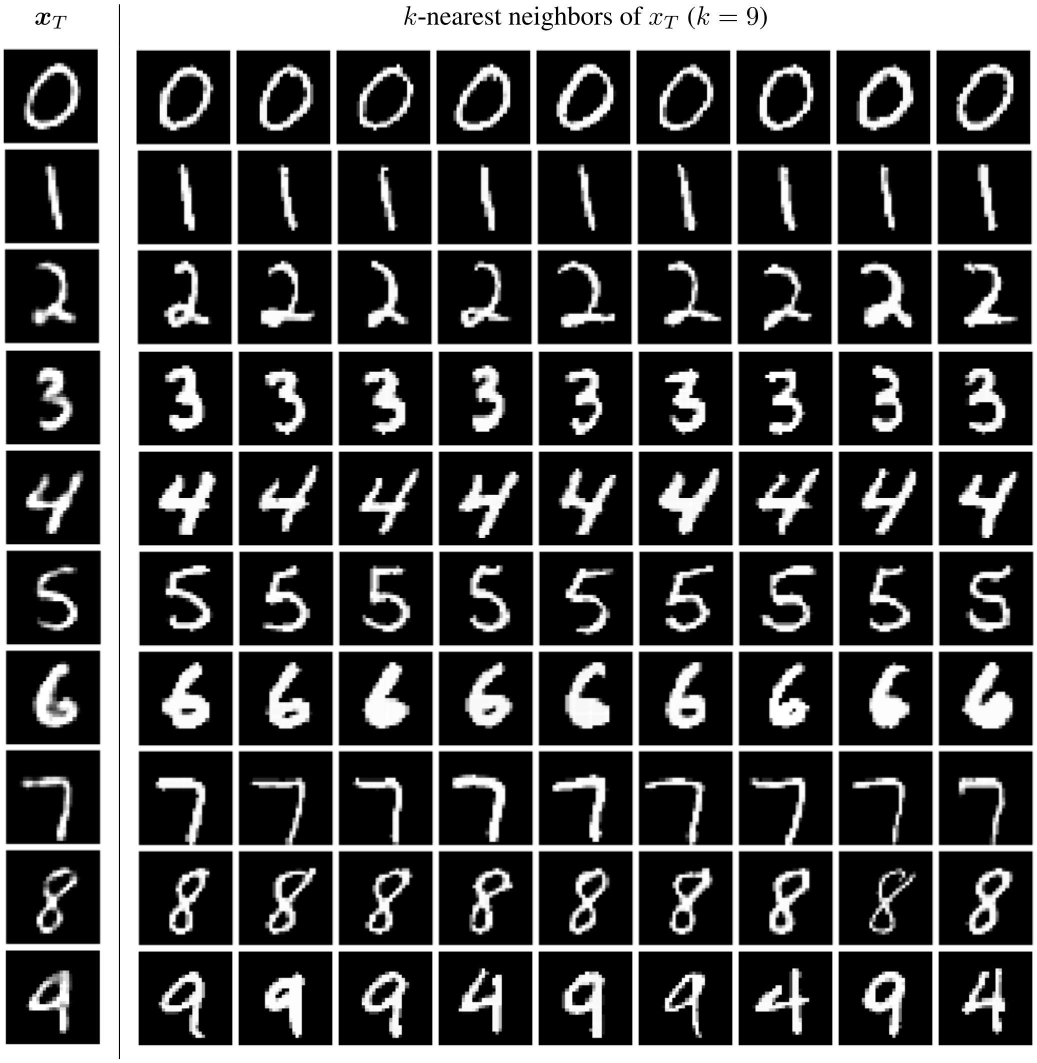


Figure 14: (Color online) The k-nearest neighbor (k-NN) test performed on images generated by the discriminator-guided Langevin sampler, when $\alpha_t = \alpha_0 = 10$ and $\boldsymbol{z}_t = 0$, on the MNIST dataset. We observe that the generated images are unique, compared to the top-9 neighbors drawn from the target dataset, indicating that the sampler **does not memorize** the images seen as part of the interpolating RBF discriminator's centers.

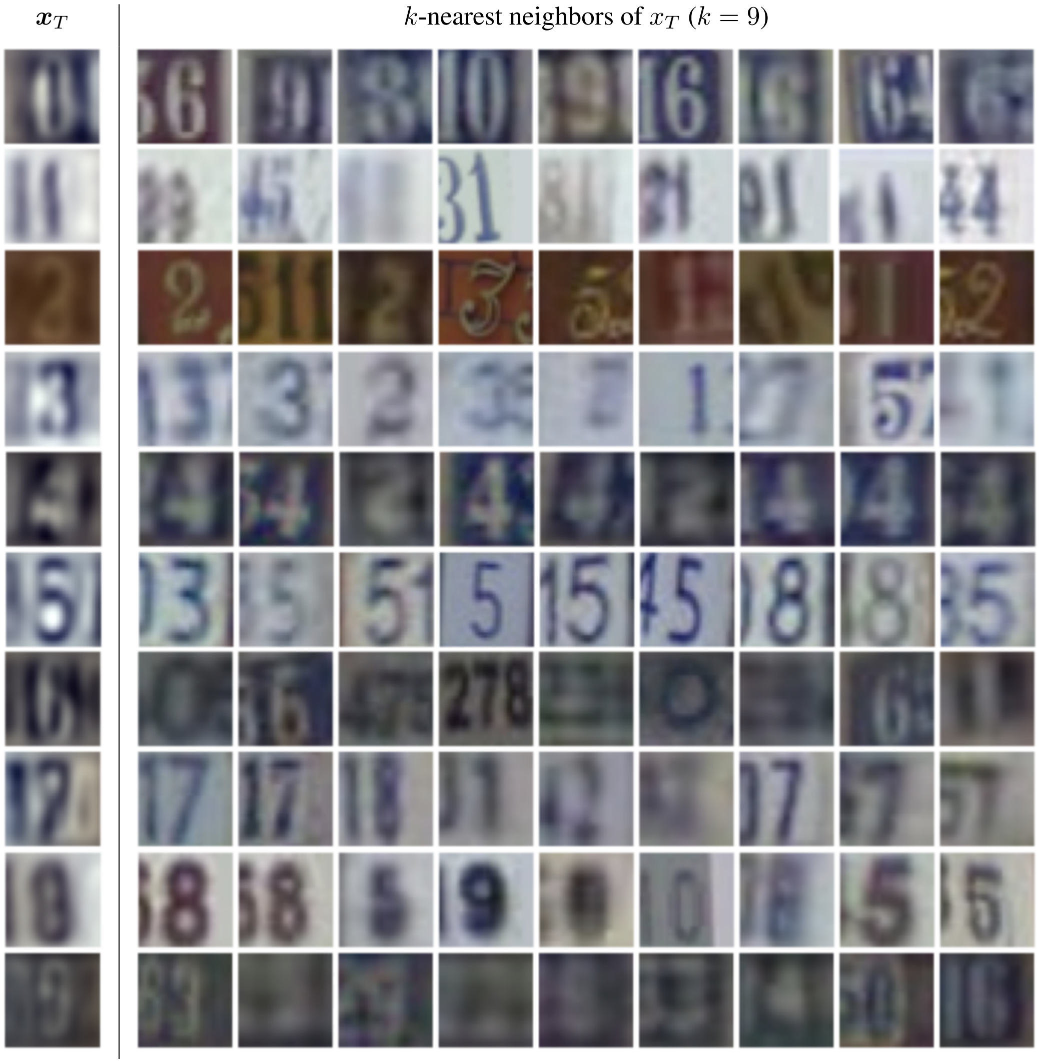


Figure 15: (Color online) The k-nearest neighbor (kNN) test performed on images generated by the discriminator-guided Langevin sampler, when $\alpha_t = \alpha_0 = 10$ and $\boldsymbol{z}_t = 0$, on the SVHN dataset. We observe that the generated images are unique, compared to the top-9 neighbors drawn from the target dataset. For generated samples such as the *digit 9* or *digit 5*, we observe that the top k-NN images are from classes different from that of the generated image, indicative of the model's ability to interpolate between the classes seen as part of discriminator centers during sampling.

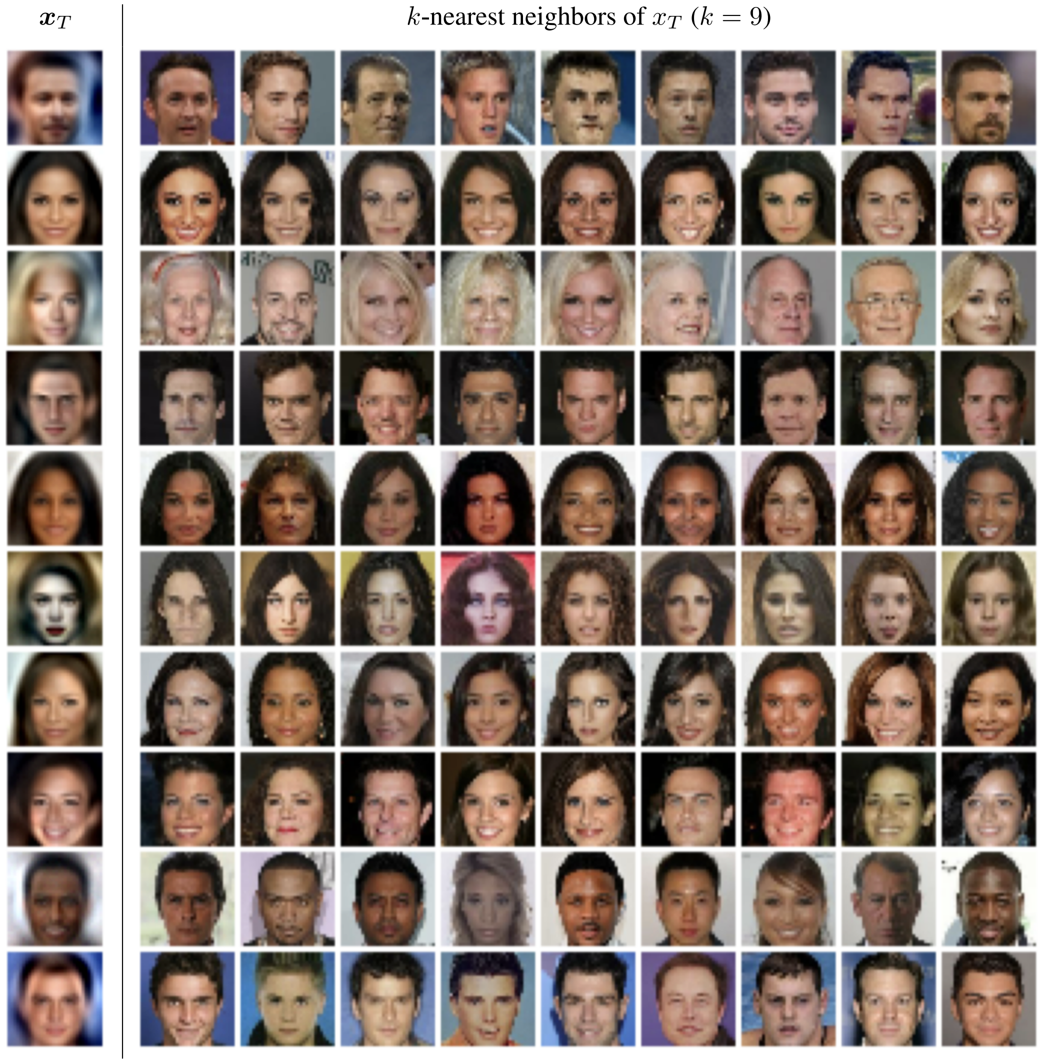


Figure 16: (Color online) The k-nearest neighbor (kNN) test performed on images generated by the discriminator-guided Langevin sampler, when $\alpha_t = \alpha_0 = 10$ and $\boldsymbol{z}_t = 0$, on the CelebA dataset. The generated images are unique, compared to the top-9 neighbors drawn from the target dataset, which suggests that the proposed approach does not learn to memorize samples.

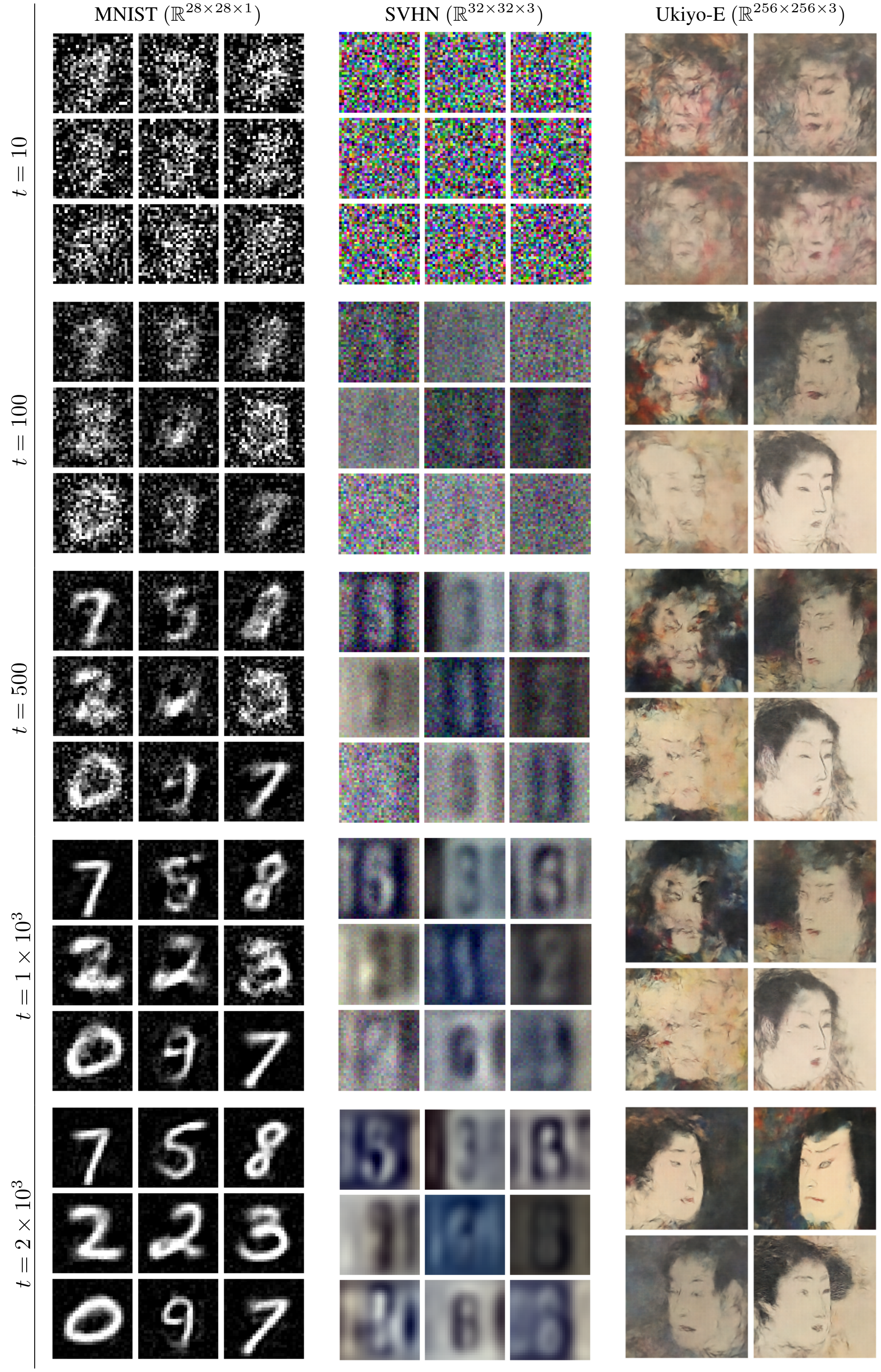

MNIST ($\mathbb{R}^{28\times28\times1}$)
SVHN ($\mathbb{R}^{32\times32\times3}$)
Ukiyo-E ($\mathbb{R}^{256\times256\times3}$)
$t = 10$
$t = 100$
$t = 500$
$t = 1\times10^3$
$t = 2\times10^3$

Figure 17: (Color online) Images generated using the discriminator-guided Langevin sampler. The score in standard diffusion models is replaced with the gradient field of the discriminator, obviating the need for any trainable neural network, while generating realistic samples.

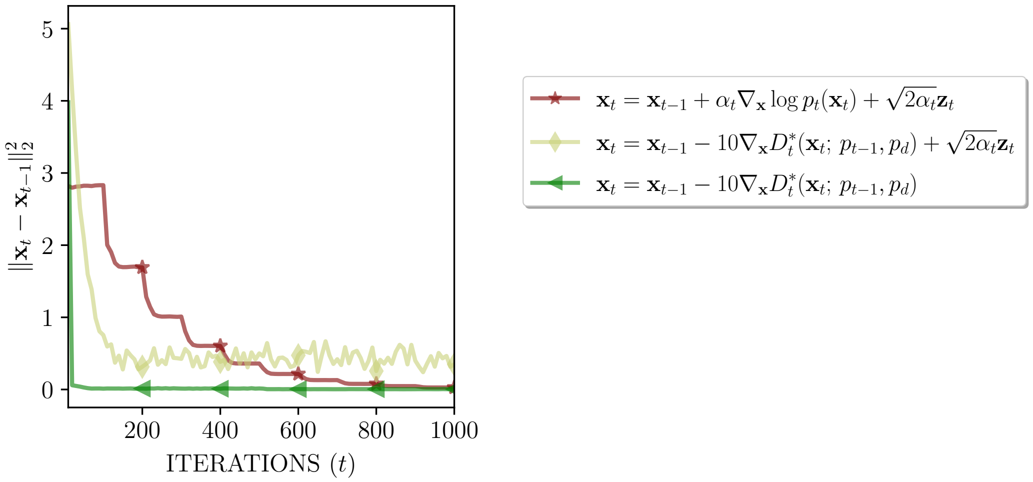


Figure 18: (● Color online) Plot comparing the *iterate convergence* of the discriminator-guided Langevin diffusion model, compared against the baseline NCSNv1 (Song & Ermon, 2019) model. The score in NCSN is replaced with the output of a score network S_θ. The norm of the iterate-differences decays as the noise-scale in the case of NCSN. This is consistent with the observations made by Song & Ermon (2020), who showed that the score network S_θ implicitly scales its output by the noise variance σ. In discriminator-guided Langevin diffusion, adding noise results in poorer performance, while the unadjusted Langevin sampler performs the best.

G Computational Resources

All experiments were carried out using a TensorFlow 2.0 (Abadi et al., 2016) backend. Experiments on NCSN were built atop a publicly available implementation (URL: `https://github.com/Xemnas0/NCSN-TF2.0`). Experiments were performed on SuperMicro workstations with 256 GB of system RAM comprising two NVIDIA GTX 3090 GPUs, each with 24 GB of VRAM.

H Source Code and Animations

The TF 2.0 (Abadi et al., 2016) based source code for implementing ScoreGANs, FloWGANs, and discriminator-guided Langevin diffusion have been included as part of the *Supplementary Material* and are accessible at `https://github.com/DarthSid95/ScoreFloWGANs`. Additionally, we have also provided animations corresponding to the *Shape Morphing* experiments presented in Figure 10, and the images generated in Figures 11– 13 and Figure 17. Full-resolution versions of images presented in the paper are accessible in the GitHub Repository.

Broader Impact

The main goal of this paper is to introduce a unifying theory for GANs and score-based models, both of which are classes of generative modeling schemes. In recent years, the advancements made in the context of image-to-image (Karras et al., 2021; Kang et al., 2023) or text-to-image (Yu et al., 2022) based generative models have brought to light, a need to better explain their inner workings. Consequently, this work aims to demystify various aspect of GAN and score-based models, linking both their causes for success, or their failures. Whether these insights are used to improve desired features such as the better diversity and reduced bias, or to exemplify their negative qualities, the choice lies in the hands of the model engineers!