Polarity Sampling: Quality and Diversity Control of Pre-Trained Generative Networks via Singular Values

Ahmed Imtiaz Humayun [†*], Randall Balestriero [‡*], Richard Baraniuk[†]
Rice University[†], *Meta AI Research*[‡]

Abstract

We present Polarity Sampling, a theoretically justified plug-and-play method for controlling the generation quality and diversity of any pre-trained deep generative network (DGN). Leveraging the fact that DGNs are, or can be approximated by, continuous piecewise affine splines, we derive the analytical DGN output space distribution as a function of the product of the DGN's Jacobian singular values raised to a power ρ. We dub ρ the **polarity** *parameter and prove that ρ focuses the DGN sampling on the modes ($\rho < 0$) or anti-modes ($\rho > 0$) of the DGN output-space probability distribution. We demonstrate that nonzero polarity values achieve a better precision-recall (quality-diversity) Pareto frontier than standard methods, such as truncation, for a number of state-of-the-art DGNs. We also present quantitative and qualitative results on the improvement of overall generation quality (e.g., in terms of the Fréchet Inception Distance) for a number of state-of-the-art DGNs, including StyleGAN3, BigGAN-deep, NVAE, for different conditional and unconditional image generation tasks. In particular, Polarity Sampling redefines the state-of-the-art for StyleGAN2 on the FFHQ Dataset to FID 2.57, StyleGAN2 on the LSUN Car Dataset to FID 2.27 and StyleGAN3 on the AFHQv2 Dataset to FID 3.95. Colab Demo.*

1. Introduction

Deep Generative Networks (DGNs) have emerged as the go-to framework for generative modeling of high-dimensional datasets, such as natural images. Within the realm of DGNs, different frameworks can be used to produce an approximation of the data distribution, e.g., Generative Adversarial Networks (GANs) [18], Variational AutoEncoders (VAEs) [31] or flow-based models [40]. But despite the different training settings and losses that each of these frameworks aim to minimize, the evaluation metric of choice that is used to characterize the overall quality of generation is the *Fréchet Inception Distance* (FID) [22].

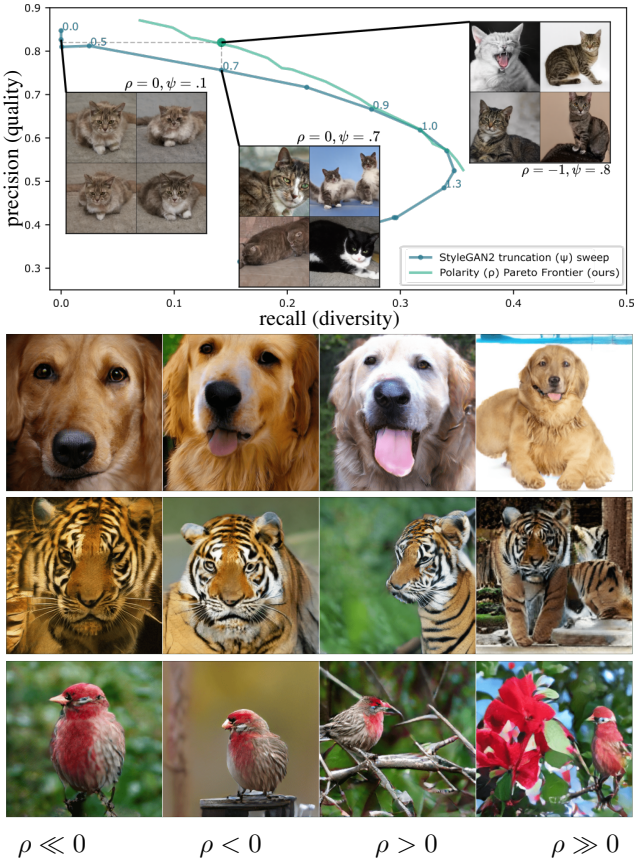

Figure 1. **First row**: Evolution of generation quality and diversity for varying truncation [29] ψ and polarity ρ. Polarity Sampling achieves a better Pareto trade-off than truncation, e.g., polarity can be used to achieve a specified recall at higher precision or a specified precision at higher recall, compared to truncation. For additional Pareto examples, see Fig. 3. **Second, Third, and Fourth row:** Samples obtained from BigGAN-deep on *Golden Retriever*, *Tiger* and *House Finch* classes of Imagenet with samples of greater quality ($\rho < 0$) and greater diversity ($\rho > 0$). For examples with LSUN [54], see Fig. 4.

The FID is obtained by taking the Fréchet Distance in the InceptionV3 [48] embedding space between two distributions; the distributions are usually taken to be the training dataset and samples from a DGN trained on the dataset.

*equal contribution

It has been established in prior work [45] that FID non-linearly combines measures of quality and diversity of the samples, which has inspired further research into disentanglement of these quantities as *precision* and *recall* [32, 45] metrics respectively.

Recent state-of-the-art DGNs such as BigGAN [8], StyleGAN2/3 [28, 30], and NVAE [53], have reached FIDs nearly as low as one could obtain when comparing subsets of real data with themselves. This has led to the deployment of DGNs in a variety of applications, such as real-world high-quality content generation and data-augmentation. However, it is clear that, **depending on the domain of application, generating samples from the best FID model could be suboptimal**. For example, realistic content generation might benefit more from high-quality (precision) samples, while data-augmentation might benefit more from samples of high-diversity (recall), even if in each case, the overall FID slightly diminishes [16, 25]. Therefore, a number of state-of-the-art DGNs have introduced a controllable parameter to trade-off between the precision and recall of the generated samples, e.g., truncated latent space sampling [8], interpolating truncation [29, 30]. However, these methods do not always work "out-of-the-box" [8], e.g., BigGAN requires orthogonal regularization of the DGN's parameters during training. These methods also lack a clear theoretical understanding which can limit their deployment for sensitive applications.

In this paper, we propose a principled solution to control the quality (precision) and diversity (recall) of DGN samples that does not require retraining nor specific conditioning of model training. Our method, termed *Polarity Sampling*, builds on our previous work on the analytical form of the learned DGN sample distribution [24] and introduces a new hyperparameter, that we dub the *polarity* $\rho \in \mathbb{R}$, that adapts the latent space distribution for post-training control. **The polarity parameter provably forces the latent distribution to concentrate on the modes of the DGN distribution, i.e., regions of high probability ($\rho < 0$), or on the anti-modes, i.e., regions of low-probability ($\rho > 0$); with $\rho = 0$ recovering the original DGN distribution.** The Polarity Sampling process depends only on the top singular values of the DGN's output Jacobian matrices evaluated at each input sample and can be implemented to perform online sampling. A crucial benefit of Polarity Sampling lies in its theoretical derivation from the analytical DGN data distribution [24] where the product of the DGN Jacobian matrices singular values – raised to the power ρ – provably controls the DGN samples distribution as desired. See Fig. 1 for an initial example of Polarity Sampling in action.

Our main contributions are as follows:

[C1] We first provide the theoretical derivation of Polarity Sampling based on the singular values of the generator Jacobian matrix. We provide pseudocode for Polarity Sampling and an approximation scheme to control its computational complexity as desired (Sec. 3).

[C2] We demonstrate on a range of DGNs and datasets that Polarity Sampling not only enables one to move on the precision-recall Pareto frontier (Sec. 4.1), i.e., it controls the quality and diversity efficiently, but it also reaches improved FID scores for each model (Sec. 4.2).

[C3] We leverage the fact that negative Polarity Sampling provides access to the modes of the learned DGN distribution, which enables us to explore several timely and important questions regarding DGNs. We provide visualization of the modes of trained GANs and VAEs (Sec. 5.1) and assess the perceptual smoothness around the modes (Sec. 5.2).

2. Related Work

Deep Generative Networks as Piecewise-Linear Mappings. In most DGN settings, once training has been completed, sampling new data points is performed by first sampling latent space samples $\boldsymbol{z}_i \in \mathbb{R}^K$ from a latent space distribution $\boldsymbol{z}_i \sim p_{\boldsymbol{z}}$ and then processing those samples throughout a DGN $G : \mathbb{R}^K \mapsto \mathbb{R}^D$ to obtain the sample $\boldsymbol{x}_i \triangleq G(\boldsymbol{z}_i), \forall i$. One recent line of research that we will rely on through our study consists in formulating DGNs as Continuous Piecewise Affine (CPA) mappings [3, 35], that be expressed as

$$G(\boldsymbol{z}) = \sum_{\omega \in \Omega} (\boldsymbol{A}_\omega \boldsymbol{z} + \boldsymbol{b}_\omega) 1_{\{\boldsymbol{z} \in \omega\}}, \tag{1}$$

where Ω is the input space partition induced by the DGN architecture, ω is a partition-region where z resides, and $\boldsymbol{A}_\omega, \boldsymbol{b}_\omega$ are the corresponding slope and offset parameters. The CPA formulation of Eq. (1) either represents the exact DGN mapping, when the nonlinearities are CPA e.g. (leaky-)ReLU, max-pooling, or represents a first-order approximation of the DGN mapping. For more background on CPA networks, see [4]. *The key result from [12] that we will leverage is that Eq. (1) is either exact, or can be made close enough to the true mapping G, to be considered exact for practical purposes.*

Post-Training Improvement of a DGN's Latent Distribution. The idea that the training-time latent distribution $p_{\boldsymbol{z}}$ might be suboptimal for test-time evaluation has led to multiple research directions to improve the quality of samples post-training. [10,49] proposed to optimize the samples $\boldsymbol{z} \sim p_{\boldsymbol{z}}$ based on a Wasserstein discriminator, leading to the *Discriminator Optimal Transport* (DOT) method. That is, after sampling a latent vector $\boldsymbol{z}$, the latter is repeatedly updated such that the produced datum has greater quality. [50] proposes to simply remove the samples that produce data out of

the true data manifold. This can be viewed as a binary rejection decision of any new sample $\boldsymbol{z} \sim p_{\boldsymbol{z}}$. [2] were the first to formally introduce rejection sampling based on a discriminator providing a quality estimate used for the rejection sampling of candidate vectors $\boldsymbol{z} \sim p_{\boldsymbol{z}}$. Replacing rejection sampling with the Metropolis-Hasting algorithm [21] led to the method of [52], coined MH-GAN. An improvement made by [19] was to use the *Sampling-Importance-Resampling* (SIR) algorithm [43]. [26] proposes *latentRS* which consists in training a WGAN-GP [20] on top of any given DGN to learn an improved latent space distribution producing higher-quality samples. [26] also proposes *latentRS+GA*, where the generated samples from that learned distribution are further improved through gradient ascent.

Truncation of the Latent Distribution. Latent space truncation was introduced for high-resolution face image generation by [33] as a method of removing generated artifacts. The authors employed a latent prior of $\boldsymbol{z} \sim \mathcal{U}[-1, 1]$ during training and $\boldsymbol{z} \sim \mathcal{U}[-0.5, 0.5]$ for qualitative improvement during evaluation. The "truncation trick" was formally introduced by [8] where the authors propose resampling latents z if they exceed a specified threshold for truncation. The authors also use weight orthogonalization during training to make truncation amenable. Style-based architectures [29, 30] introduce a linear interpolation based truncation in the style-space, which is also designed to converge to the average of the dataset [29]. Ablations for truncation in style-based generators are provided in [32].

3. Introducing The Polarity Parameter From First Principles

In this section, we introduce *Polarity Sampling*, a method that enables us to control the generation quality and diversity of DGNs. We will proceed by first expressing the analytical form of DGNs' output distribution (Sec. 3.1), and parametrizing the latent space distribution by the singular values of its Jacobian matrix and our *polarity* parameter (Sec. 3.2). We provide pseudo-code and an approximation strategy that enables fast sampling (Sec. 3.3).

3.1. Analytical Output-Space Density Distribution

Given a DGN G, samples are obtained by sampling $G(\boldsymbol{z})$ with a given latent space distribution, as in $\boldsymbol{z} \sim p_{\boldsymbol{z}}$. This produces samples that will lie on the *image* of G, the distribution of which is subject to $p_{\boldsymbol{z}}$, the DGN latent space partition Ω and per-region affine parameters $\boldsymbol{A}_\omega, \boldsymbol{b}_\omega$. We denote the DGN output space distribution as p_G. Under an injective DGN mapping assumption ($g(z) = g(z') \implies z = z'$) (which holds for various architectures, see, e.g., [41]) it is possible to obtain the analytical form of the DGN output distribution by p_G [24]. For a reason that will become clear in the next section, we focus here on the case $\boldsymbol{z} \sim U(\mathcal{D})$ i.e., using a Uniform latent space distribution over the domain $\mathcal{D}$. Leveraging the Moore-Penrose pseudo inverse [51] $\boldsymbol{A}^\dagger \triangleq (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T$, we obtain the following.

Theorem 1. *For $\boldsymbol{z} \sim U(\mathcal{D})$, the probability density $p_G(\boldsymbol{x})$ is given by*

$$p_G(\boldsymbol{x}) \propto \sum_{\omega \in \Omega} \det(\boldsymbol{A}_\omega^T \boldsymbol{A}_\omega)^{-\frac{1}{2}} \mathbb{1}_{\{\boldsymbol{A}_\omega^\dagger(\boldsymbol{x}-\boldsymbol{b}_\omega) \in \omega \cap \mathcal{D}\}}, \quad (2)$$

where det *is the pseudo-determinant, i.e., the product of the nonzero eigenvalues of $\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega$. (Proof in Appendix B.1.)*

Note that one can also view $\det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega)^{1/2}$ as the product of the nonzero singular values of $\boldsymbol{A}_\omega$. Theorem 1 is crucial to our development since it demonstrates that the probability of a sample $\boldsymbol{x} = g(\boldsymbol{z})$ is proportional to the change in volume $(\det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega)^{1/2})$ produced by the coordinate system $\boldsymbol{A}_\omega$ of the region ω in which $\boldsymbol{z}$ lies in (recall Eq. (1)). If a region $\omega \in \Omega$ has a slope matrix $\boldsymbol{A}_\omega$ that contracts the space ($\det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega) < 1$) then the output density on that region — mapped to the output space region $\{\boldsymbol{A}_\omega\boldsymbol{u} + \boldsymbol{b}_\omega : \boldsymbol{u} \in \omega\}$ — is increased, as opposed to other regions that either do not contract the space as much, or even expand it ($\det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega) > 1$). Hence, the concentration of samples in each output space region depends on how that region's slope matrix contracts or expands the space, relative to all other regions.

3.2. Controlling the Density Concentration with a Single Parameter

From Theorem 1 we can directly obtain an explicit parametrization of $p_{\boldsymbol{z}}$ that enables us to control the distribution of samples in the output space, i.e., to control p_G. In fact, note that one can sample from the mode of the DGN distribution by employing $\boldsymbol{z} \sim U(\omega^*), \omega^* = \arg\min_{\omega \in \Omega} \det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega)$. Alternatively, one can sample from the region of lowest probability, i.e., the anti-mode, by employing $\boldsymbol{z} \sim U(\omega^*), \omega^* = \arg\max_{\omega \in \Omega} \det(\boldsymbol{A}_\omega^T\boldsymbol{A}_\omega)$. This directly leads to our Polarity Sampling method that adapts the latent space distribution based on the per-region pseudo-determinants.

Corollary 1. *The latent space distribution*

$$p_\rho(\boldsymbol{z}) \propto \sum_{\omega \in \Omega} \det(\boldsymbol{A}_\omega^T \boldsymbol{A}_\omega)^{\frac{\rho}{2}} \mathbb{1}_{\{\boldsymbol{z} \in \omega\}}, \quad (3)$$

where $\rho \in \mathbb{R}$ is the polarity parameter, produces the DGN output distribution

$$p_G(\boldsymbol{x}) \propto \sum_{\omega \in \Omega} \det(\boldsymbol{A}_\omega^T \boldsymbol{A}_\omega)^{\frac{\rho-1}{2}} \mathbb{1}_{\{\boldsymbol{A}_\omega^\dagger(\boldsymbol{x}-\boldsymbol{b}_\omega) \in \omega \cap \mathcal{D}\}}, \quad (4)$$

which falls back to the standard DGN distribution for $\rho = 0$, to sampling of the mode(s) for $\rho \to -\infty$ and to sampling of the anti-mode(s) for $\rho \to \infty$. (Proof in Appendix B.2.)

Polarity Sampling consists of using the latent space distribution Eq. (3) with a polarity parameter ρ, that is either negative, concentrating the samples toward the mode(s) of the DGN distribution p_G, positive, concentrating the samples towards the anti-modes(s) of the DGN distribution p_G or zero, which removes the effect of polarity. Note that Polarity Sampling changes the output density in a continuous fashion. Its practical effect, as we will see in Sec. 4.1, is to control the quality and diversity of the obtained samples.

3.3. Approximation and Implementation

We now provide the details and pseudocode for the Polarity Sampling procedure that implements Corollary 1.

Computing the $\boldsymbol{A}_\omega$ Matrix. The per-region slope matrix as in Eq. (1), can be obtained given any DGN by first sampling a latent vector $\boldsymbol{z} \in \omega$, and then obtaining the Jacobian matrix of the DGN $\boldsymbol{A}_\omega = \boldsymbol{J}_{\boldsymbol{z}} G(\boldsymbol{z}), \forall \boldsymbol{z} \in \omega$. This has the benefit of directly employing automatic differentiation libraries and thus does not require any exhaustive implementation nor derivation. Computing $\boldsymbol{J}_{\boldsymbol{z}} G(\boldsymbol{z})$ of a generator is not uncommon in practice, e.g., it is employed during path length regularization of StyleGAN2 [30].

Discovering the Regions $\omega \in \Omega$. As per Eq. (3), we need to obtain the singular values of $\boldsymbol{A}_\omega$ (see next paragraph) for each region $\omega \in \Omega$. This is often a complicated task, especially for state-of-the-art DGNs that can have a partition Ω whose number of regions grows with the architecture depth and width [36]. Furthermore, checking if $\boldsymbol{z} \in \omega$ requires one to solve a linear program [15], which is expensive. As a result, we develop an approximation that consists of sampling many $\boldsymbol{z} \sim U(\mathcal{D})$ vectors from the latent space (hence our uniform prior assumption in Corollary 1), and computing their corresponding matrices $\boldsymbol{A}_{\omega(\boldsymbol{z})}$. This way, we are guaranteed that $\boldsymbol{A}_{\omega(\boldsymbol{z})}$ corresponds to the slope of the region ω in which $\boldsymbol{z}$ falls in, removing the need to check whether $\boldsymbol{z} \in \omega$. We do so over N samples obtained uniformly from the DGN latent space (based on the original latent space domain). Selection of N can impact performance as this exploration needs to discover as many regions from Ω as possible.

Singular Value Computation. Computing the singular values of $\boldsymbol{A}_\omega$ is an $\mathcal{O}(\min(K, D)^3)$ operation [17]. However, not all singular values might be relevant, e.g., the smallest singular values that are nearly constant across regions ω can be omitted without altering Corollary 1. Hence, we employ only the top-k singular values of $\boldsymbol{A}_\omega$ to speed up singular value computation to $\mathcal{O}(Dk^2)$, details provided in Appendix A.3. (Further approximation could be employed if needed, e.g., power iteration [39]).

While the required number of latent space samples N and the number of top singular values k might seem to be a limitation of Polarity Sampling, we have found in practice that N and k for state-of-the-art DGNs can be set at $N \approx 200K$, $k \in [30, 100]$. We conduct a careful ablation study and demonstrate the impact of different choices for N and k in Fig. 2 and Tabs. 3 and 4 in Appendix A.2. Computation times and software/hardware details are provided in Appendix A.3. To reduce round-off errors that can occur for extreme values of ρ, we compute the product of singular values in log-space, as shown in Algorithm 1.

We summarize how to obtain S samples using the above steps in the pseudocode given in Algorithm 1 and provide an efficient solution to reduce the memory requirement incurred when computing the large matrix $\boldsymbol{A}_\omega$ in Appendix A.4. We also provide an implementation that enables online sampling in Algorithm 2 (Appendix A.1). It is also possible to control the DGN prior $p_{\boldsymbol{z}}$ with respect to a different space than the data-space e.g. inception-space, or with a different input space than the latent-space e.g. style-space in StyleGAN2/3. This incurs no changes in Algorithm 1 except that the DGN is now considered to be either a subset of the original one, or to be composed with a VGG/InceptionV3 network. We provide the implementation details for style-space, VGG-space, and Inception-

Algorithm 1 Polarity Sampling procedure with polarity ρ; online version and 2D examples in Appendix. Algorithm 2 and Fig. 11. For implementation details, see Sec. 3.3.

Input: $K > 0, S > 0, N \gg S, G, \mathcal{D}, \rho \in \mathbb{R}$
  $\mathcal{Z}, \mathcal{S}, \mathcal{R} \leftarrow [], [], []$
  for $n = 1, \dots, N$ **do**
    $\boldsymbol{z} \sim U(\mathcal{D})$
    $\sigma = \text{SingularValues}(\boldsymbol{J}_{\boldsymbol{z}} G(\boldsymbol{z}), \text{decreasing} = \text{True})$
    $\mathcal{Z}.\text{append}(\boldsymbol{z})$
    $\mathcal{S}.\text{append}(\rho \sum_{k=1}^{K} \log(\sigma[k] + \epsilon))$
  for $n = 1, \dots, S$ **do**
    $i \sim \text{Categorical}(\text{prob} = \text{softmax}(\mathcal{S}))$
    $\mathcal{R}.\text{append}(\mathcal{Z}[i])$
Output: $\mathcal{R}$

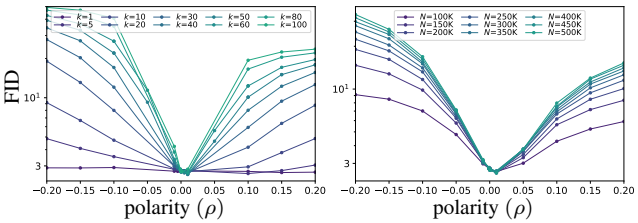

Figure 2. Effect of Polarity Sampling on FID of a StyleGAN2-F model pretrained on FFHQ for varying number of top-k singular values (**left**) and varying number of latent space samples N used to obtain per-region slope matrix $\boldsymbol{A}_\omega$ singular values (**right**) (recall Sec. 3.3 and Algorithm 1). The trend in FIDs to evaluate the impact of ρ stabilizes when using around $k = 40$ singular values and $N \approx$200,000 latent space samples. For the effect of k and N on precision and recall, see Fig. 9.

space in Appendix A.5. In those cases, the partition Ω and the per-region mapping parameters $\boldsymbol{A}_\omega, \boldsymbol{b}_\omega$ are the ones of the corresponding sub-network or composition of networks (recall Eq. (1)). Polarity Sampling adapts the DGN prior distribution to obtain the modes or anti-modes with respect to the considered output spaces.

4. Controlling Precision, Recall, and FID via Polarity

We now provide empirical validation of Polarity Sampling with an extensive array of experiments. Since calculation of distribution metrics such as FID, precision, and recall are sensitive to image processing nuances, we use each model's original code repository except for BigGAN-deep on ImageNet [13], for which we use the evaluation pipeline specified for ADM [14]. For NVAE (trained on colored-MNIST [1]), we use a modified version of the StyleGAN3 evaluation pipeline. Precision and recall metrics are all based on the implementation of [32]. Metrics in Tab. 2 are calculated for 50K training samples to be able to compare with existing latent reweighing methods. For all other results, the metrics are calculated using $\min\{N_D, 100K\}$ training samples, where N_D is the number of samples in the dataset.

4.1. Polarity Efficiently Parametrizes the Precision-Recall Pareto Frontier

As we have discussed above, Polarity Sampling can explicitly sample from the modes or anti-modes of any learned DGN distribution. Since the DGN is trained to fit the training distribution, sampling from the modes and anti-modes correspond to sampling from regions of the data manifold that are approximated better/worse by the DGN . Therefore, Polarity Sampling is an efficient parameterization of the trade-off between precision and recall of generation [32] since regions with higher precision are regions where the manifold approximation is more accurate.

As experimental proof, we provide in Fig. 3 the precision-recall trade-off when sweeping polarity, and compare it with truncation [29] for pretrained StyleGAN{2,3} architectures. We see that Polarity Sampling offers a competitive alternative to truncation for controlling the precision-recall trade-off of DGNs across datasets and models. For any given precision, the ρ parameter allows us to reach greater recall than what is possible via latent space truncation [29]. And conversely, for any given recall, it is possible to reach a higher precision than what can be attained using latent space truncation. We see that diversity collapses rapidly for latent truncation compared to Polarity Sampling, across all architectures, which is a major limitation. In addition to that, controlling both truncation and polarity allows us to further extend the Pareto frontier for all of our experiments.

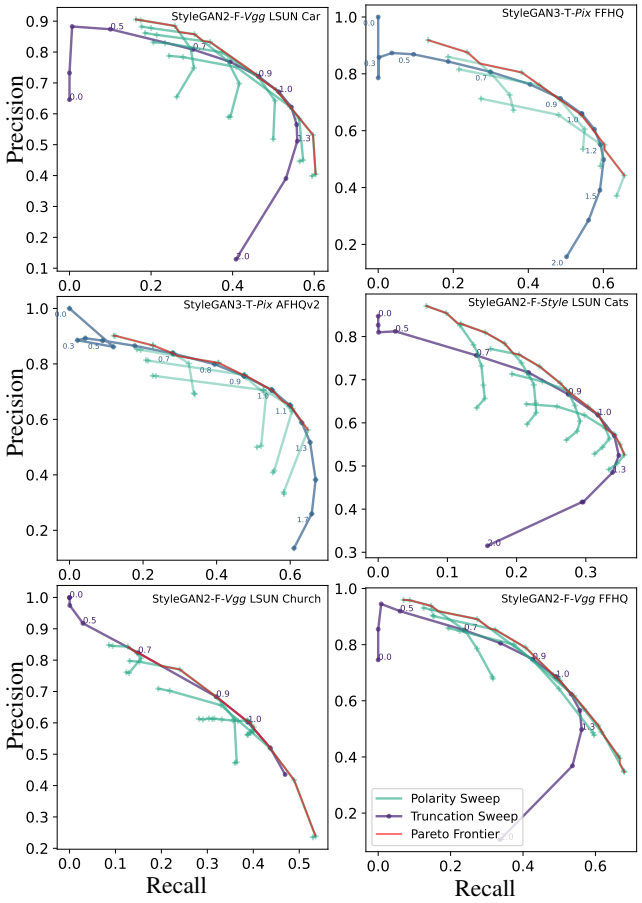

Figure 3. Pareto frontier of the precision-recall metrics can be obtained solely by varying the polarity parameter, for any given truncation level. We depict here six different models and datasets. Results for additional models and datasets are provided in Fig. 1 and Fig. 8.

Apart from the results presented here, we also see that polarity can be used to effectively control the precision-recall trade-off for BigGAN-deep [8] and ProGAN [27]. ProGAN unlike BigGAN and StyleGAN, is not compatible with truncation based methods, i.e., latent space truncation has negligible effect on precision-recall. Hence, polarity offers a great benefit over those existing solutions: Polarity Sampling can be applied regardless of training or controllability factors that are preset in the DGN design. We provide additional results in Appendix C.

4.2. Polarity Improves Any DGN's FID

We saw in Sec. 4.1 that polarity can be used to control quality versus diversity in a meaningful and controllable manner. In this section, we connect the effect of polarity with FID. Recall that the FID metric nonlinearly combines quality and diversity [45] into a distribution distance measure. Since polarity allows us to control the output distri-

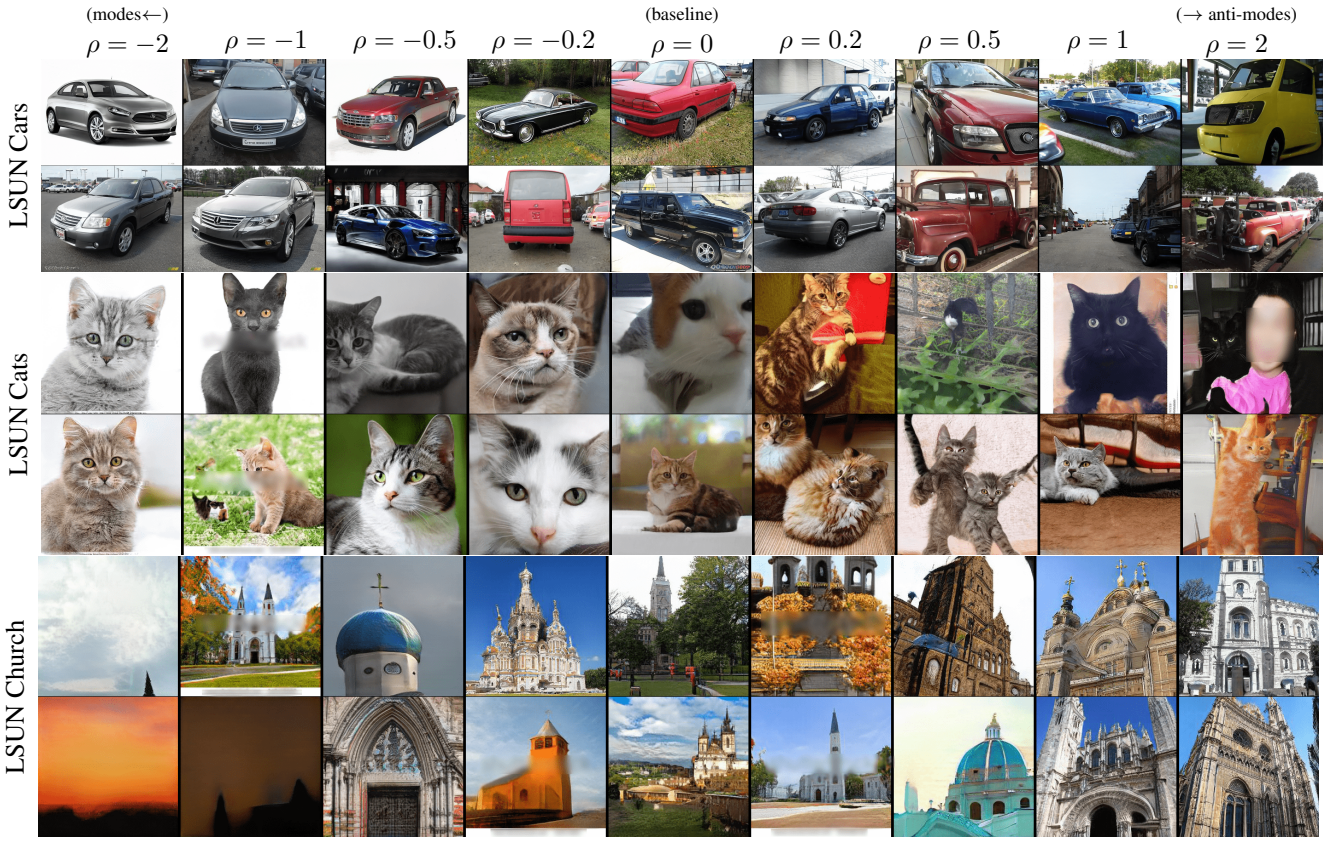

Figure 4. Curated samples of cars and cats for Polarity Sampling in style-space, and church for Polarity Sampling in pixel-space. (Qualitative comparison with truncation sweep in Fig. 10 and nearest training samples in Fig. 12 in the Appendix.) None of the images correspond to training samples, as we discuss in Sec. 5.1.

bution of the DGN, an indirect result of polarity is the reduction of FID by matching the inception embedding distribution of the DGN with that of the training set distribution. Recall that $\rho = 0$ recovers the baseline DGN sampling; for all the state-of-the-art methods in question, we reach lower (better) FID by using a nonzero polarity. In Tab. 1, we compare Polarity Sampling with state-of-the-art solutions that propose to improve FID by learning novel DGN latent space distributions, as were discussed in Sec. 2. We see that for a StyleGAN2 pre-trained on the LSUN church [54] dataset, by increasing the diversity ($\rho = 0.2$) of the VGG embedding distribution, Polarity Sampling surpasses the FID of methods reported in literature that post-hoc improves quality of generation.

In Tab. 2, we present for LSUN {Church, Car, Cat} [54], ImageNet [13], FFHQ [29], and AFHQv2 [11,28] improved FID obtained *solely by changing the polarity* ρ of a state-of-the-art DGN. This implies that Polarity Sampling provides an efficient solution to adapt the DGN latent space.

We observe that, given any specific setting, $\rho \neq 0$ always improves a model's FID. We see that in a case specific manner, both positive and negative ρ improves the FID.For

	LSUN Church 256×256		
StyleGAN2 variant	FID ↓	Prec ↑	Recall ↑
Standard	6.29	.60	.51
SIR† [43]	7.36	.61	**.58**
DOT† [49]	6.85	.67	.48
latentRS† [26]	6.31	.63	.58
latentRS+GA† [26]	6.27	**.73**	.43
ρ-sampling 0.2	**6.02**	.57	.53

Table 1. Comparison of Polarity Sampling with latent reweighting techniques from literature. FID, Precision and Recall is calculated using 50,000 samples. †Metrics reported from papers due to unavailability of code. †Precision-recall is calculated with 1024 samples only.

StyleGAN2-F trained on FFHQ, *increasing the diversity* of the inception space embedding distribution helps reach a new state-of-the-art FID. By *increasing the precision* of StyleGAN3-T via Polarity Sampling in the Vgg space, we are able to surpass the FID of baseline StyleGAN2-F [28]. We observe that controlling the polarity of the InceptionV3 embedding distribution of StyleGAN2-F gives the most significant gains in terms of FID. This is due to the fact that the

Model	FID ↓	Precision ↑	Recall ↑	Model	FID ↓	Precision ↑	Recall ↑
		LSUN Church 256×256				**LSUN Cat 256×256**	
DDPM† [23]	7.86	-	-	ADM (dropout)†	**5.57**	0.63	**0.52**
StyleGAN2	3.97	0.59	0.39	StyleGAN2	6.49	0.62	0.32
+ ρ-sampling Vgg 0.001	3.94	0.59	0.39	+ ρ-sampling Pix 0.01	6.44	0.62	0.32
+ ρ-sampling Pix -0.001	**3.92**	**0.61**	**0.39**	+ ρ-sampling Sty -0.1	6.39	**0.64**	0.32
		LSUN Car 512×384				**FFHQ 1024×1024**	
StyleGAN†	3.27	**0.70**	0.44	StyleGAN2-E	3.31	**0.71**	0.45
StyleGAN2	2.34	0.67	0.51	Projected GAN† [46]	3.08	0.65	0.46
+ ρ-sampling Vgg -0.001	2.33	0.68	0.51	StyleGAN3-T	2.88	0.65	0.53
+ ρ-sampling Sty 0.01	**2.27**	0.68	**0.51**	+ ρ-sampling Vgg -0.01	2.71	0.66	**0.54**
+ ρ-sampling Pix 0.01	2.31	0.68	0.50				
		ImageNet 256×256		StyleGAN2-F	2.74	0.68	0.49
DCTransformer† [37]	36.51	0.36	0.67	+ ρ-sampling Ic3 0.01	**2.57**	0.67	0.5
VQ-VAE-2† [42]	31.11	0.36	0.57	+ ρ-sampling Pix 0.01	2.66	0.67	0.5
SR3 † [44]	11.30	-	-				
IDDPM† [38]	12.26	0.70	0.62			**AFHQv2 512×512**	
ADM† [14]	10.94	0.69	**0.63**	StyleGAN2†	4.62	-	-
ICGAN+DA† [9]	7.50	-	-	StyleGAN3-R†	4.40	-	-
BigGAN-deep	6.86	0.85	0.29	StyleGAN3-T	4.05	0.70	0.55
+ ρ-sampling Pix 0.0065	6.82	**0.86**	0.29	+ ρ-sampling Vgg -0.001	**3.95**	**0.71**	**0.55**
ADM+classifier guidance	**4.59**	0.82	0.52				

Table 2. †Paper reported metrics. We observe that moving away from $\rho = 0$, Polarity Sampling improves FID across models and datasets, empirically validating that the top singular values of a DGN's Jacobian matrices contain meaningful information to improve the overall quality of generation

.

Frechet distance between real and generated distributions is directly affected while performing Polarity Sampling in the Inception space. We provide generated samples in Fig. 4 varying the style-space ρ for LSUN cars and LSUN cats, whereas varying the pixel-space ρ for LSUN Church. It is clear that $\rho < 0$ i.e. sampling closer to the DGN distribution modes produce samples of high visual quality, while $\rho > 0$ i.e. sampling closer to the regions of low-probability produce samples of high-diversity, with some samples which are off the data manifold due to the approximation quality of the DGN in that region. Using Polarity Sampling, we are able to advance the state-of-the-art performance on three different settings: for StyleGAN2 on the FFHQ [29] Dataset to FID 2.57, StyleGAN2 on the LSUN [54] Car Dataset to FID 2.27, and StyleGAN3 on the AFHQv2 [28] Dataset to FID 3.95. For additional experiments with ProGAN, and NVAE under controlled training and reference dataset distribution shift, see Appendix C.

5. New Insights into DGN Distributions

In Sec. 4 we demonstrated that Polarity Sampling is a practical method to manipulate DGN output distributions to control their quality and diversity. We now demonstrate that Polarity Sampling has more foundational theoretical applications as well. In particular, we dive into several timely questions regarding DGNs that can be probed using our framework.

5.1. Are GAN/VAE Modes Training Samples?

Mode collapse [5, 34, 47] has complicated GAN training for many years. It consists of the entire DGN collapsing to generate a few different samples or modes. For VAEs, modes can be expected to be related to the modes of the empirical dataset distribution, as reconstruction is part of the objective. But this might not be the case with GANs e.g., the modes can correspond to parts of the space where the discriminator is the least good at differentiating between true and fake samples. There has been no reported methods in literature that allows us to observe the modes of a trained GAN. Existing visualization techniques focus on finding the role of each DGN unit [6] or finding images that GANs cannot generate [7]. Using Polarity Sampling, we can visualize the modes of DGNs for the first time. In Fig. 5, we present samples from the modes of BigGAN-deep trained on ImageNet, StyleGAN3 trained on AFHQv2, and NVAE trained on colored-MNIST. We observe that BigGAN modes tend to reproduce the unique features of the class, removing the background and focusing more on the object that the class is assigned to. AFHQv2 modes on the other hand, focus on younger animal faces and smoother textures. NVAE mode sampling predominately produce the digit '1' which corresponds to the dataset mode (digit with the least intra-class

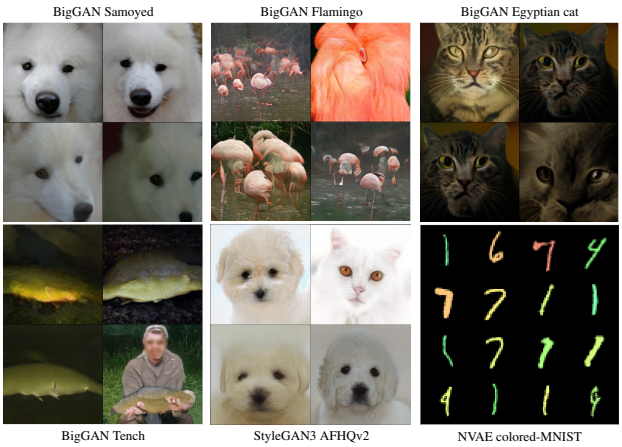

Figure 5. Modes for BigGAN-deep, StyleGAN3-T and NVAE obtained via $\rho \ll 0$ Polarity Sampling. **This is, to the best of our knowledge, the first visualization of the modes of DGNs in pixel space.**

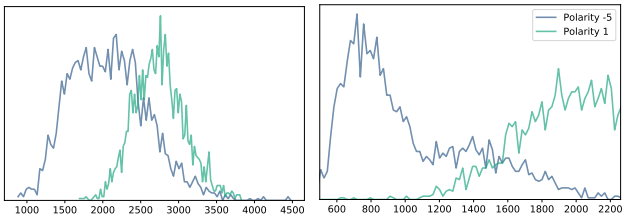

Figure 6. Distribution of l_2 distance to 3 training set nearest neighbors at 32×32 resolution, for 1000 generated samples from LSUN Church StyleGAN2 **(left)** and colored-MNIST NVAE **(right)**. Samples closer to the modes ($\rho < 0$) have a significant shift in the distribution closer to the training samples for NVAE, while for StyleGAN2 the distribution shift is minimal with significant overlap. This behavior is expected as **VAE models are encouraged to position their modes on the training samples, as opposed to GANs whose modes depend on the discriminator**.

variation). We also provide in Fig. 6 the distribution of the l_2 distances between generated samples and their 3 nearest training samples for modal ($\rho = -5$) and anti-modal ($\rho = 1$) polarity. We see that even after reducing the polarity, StyleGAN2 nearest neighbor distributions have overlap whereas for NVAE the modes move significantly closer to the training samples. In Appendix. Fig. 15 we observe a similar effect for WGAN and NVAE trained on MNIST.

5.2. Perceptual Path Length Around Modes

Perceptual Path Length (PPL) is the distance between the Vgg space image of two latent space points. It has previously been proposed as a measure of perceptual distance [30]. In Fig. 7, we report the PPL of a StyleGAN2-F trained on FFHQ, for an interpolation step of length 10^{-4} between endpoints from the latent/style space. We sample points using Polarity Sampling varying $\rho \in [1, -1]$, es-

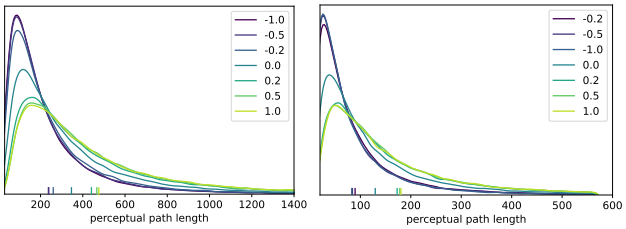

Figure 7. Distribution of PPL for StyleGAN2-F trained on FFHQ with varying Polarity Sampling (in VGG space) setting (ρ given in the legend) for endpoints in the input latent space (**left**) and endpoints in style-space (**right**). The means of the distributions (PPL score) are provided as markers on the horizontal axis.

sentially measuring the PPL for regions of the data manifold with increasing density as we increase ρ. We see that for negative values of polarity, we have significantly lower PPL compared to positive polarity or even baseline sampling ($\rho = 0$). This result shows that for StyleGAN2, there are smoother perceptual transitions closer to modes. While truncation also reduces the PPL, it essentially does so by sampling points closer to the style space mean [29], see Appendix C.5 for comparisons. Polarity Sampling in the Vgg space, can be used to directly sample from Vgg modes, making it the first method that can be used to explicitly sample regions that are perceptually smoother. It can therefore be used to develop sophisticated interpolation methods where, the interpolation is done along a high-likelihood path on a feature space manifold.

6. Conclusions

We have proposed a new parameterization of the DGN prior $p_{\boldsymbol{z}}$ in terms of a single parameter – the polarity ρ– to force the DGN samples to be concentrated on the distribution modes or anti-modes (Sec. 3). As a byproduct, for a range of DGNs, we improve the state-of-the-art FID performance. On the theoretical side, Polarity Sampling's guarantee that it samples from the modes of a DGN enabled us to explore some timely open questions, including the relation between distribution modes and training samples (Sec. 5.1), and the effect of going from mode to anti-mode generation on the perceptual path length (Sec. 5.2). We show that Polarity sampling can also be performed on feature space distributions of classifiers appended with a generator, which can be possibly used for fair attribute generation, out-of-distribution synthetic data generation and much more.

Acknowledgements

Humayun and Baraniuk were supported by NSF grants CCF-1911094, IIS-1838177, and IIS-1730574; ONR grants N00014-18-12571, N00014-20-1-2534, and MURI N00014-20-1-2787; AFOSR grant FA9550-22-1-0060; and a Vannevar Bush Faculty Fellowship, ONR grant N00014-18-1-2047.

References

[1] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 5

[2] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. *arXiv preprint arXiv:1810.06758*, 2018. 3

[3] Randall Balestriero and Richard Baraniuk. A spline theory of deep learning. In *ICML*, pages 374–383, 2018. 2

[4] Randall Balestriero and Richard Baraniuk. Mad max: Affine spline insights into deep learning. *Proceedings of the IEEE*, 109(5):704–727, 2020. 2, 3

[5] Duhyeon Bang and Hyunjung Shim. Mggan: Solving mode collapse using manifold-guided training. In *ICCV*, pages 2347–2356, 2021. 7

[6] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018. 7

[7] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *ICCV*, pages 4502–4511, 2019. 7

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2, 3, 5, 4

[9] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero. Instance-conditioned GAN. In *NeurIPS*, 2021. 7

[10] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020. 2

[11] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *CVPR*, pages 8188–8197, 2020. 6

[12] Ingrid Daubechies, Ronald A. DeVore, Nadav Dym, Shira Faigenbaum-Golovin, Shahar Z. Kovalsky, Kung-Ching Lin, Josiah Park, Guergana Petrova, and Barak Sober. Neural network approximation of refinable functions. *CoRR*, abs/2107.13191, 2021. 2

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 5, 6

[14] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *arXiv preprint arXiv:2105.05233*, 2021. 5, 7

[15] Matteo Fischetti and Jason Jo. Deep neural networks as 0-1 mixed integer linear programs: A feasibility study. *arXiv preprint arXiv:1712.06174*, 2017. 4

[16] Edoardo Giacomello, Pier Luca Lanzi, and Daniele Loiacono. Doom level generation using generative adversarial networks. In *Games, Entertainment, Media Conference*, pages 316–323, 2018. 2

[17] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Linear algebra*, pages 134–151. Springer, 1971. 4

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[19] Aditya Grover, Jiaming Song, Alekh Agarwal, Kenneth Tran, Ashish Kapoor, Eric Horvitz, and Stefano Ermon. Bias correction of learned generative models using likelihood-free importance weighting. *arXiv preprint arXiv:1906.09531*, 2019. 3

[20] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. 3

[21] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. 3

[22] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 1

[23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020. 7

[24] Ahmed Imtiaz Humayun, Randall Balestriero, and Richard Baraniuk. Magnet: Uniform sampling from deep generative network manifolds without retraining. In *ICLR*, 2022. 2, 3

[25] Zubayer Islam, Mohamed Abdel-Aty, Qing Cai, and Jinghui Yuan. Crash data augmentation using variational autoencoder. *Accident Analysis & Prevention*, 151:105950, 2021. 2

[26] Thibaut Issenhuth, Ugo Tanielian, David Picard, and Jeremie Mary. Latent reweighting, an almost free improvement for gans. *arXiv preprint arXiv:2110.09803*, 2021. 3, 6

[27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 5, 4, 7

[28] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *arXiv preprint arXiv:2106.12423*, 2021. 2, 6, 7

[29] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2, 3, 5, 6, 7, 8

[30] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, pages 8110–8119, 2020. 2, 3, 4, 8

[31] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1

[32] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *arXiv preprint arXiv:1904.06991*, 2019. 2, 3, 5

[33] Marco Marchesi. Megapixel size image creation using generative adversarial networks. *arXiv preprint arXiv:1706.00082*, 2017. 3

[34] Luke Metz, Ben Poole, David Pfau, and Jascha Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016. 7

[35] Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. *arXiv preprint arXiv:1402.1869*, 2014. 2

[36] Guido Montúfar, Yue Ren, and Leon Zhang. Sharp bounds for the number of regions of maxout networks and vertices of minkowski sums. *arXiv preprint arXiv:2104.08135*, 2021. 4

[37] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021. 7

[38] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021. 7

[39] Andy Packard, Michael KH Fan, and John C Doyle. A power method for the structured singular value. Technical report, 1988. 4

[40] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*, pages 3617–3621. IEEE, 2019. 1

[41] Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten de Hoop. Globally injective relu networks. *arXiv preprint arXiv:2006.08464*, 2020. 3

[42] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, pages 14866–14876, 2019. 7

[43] Donald B Rubin. Using the sir algorithm to simulate posterior distributions. *Bayesian statistics*, 3:395–402, 1988. 3, 6

[44] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *arXiv preprint arXiv:2104.07636*, 2021. 7

[45] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018. 2, 5

[46] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *arXiv preprint arXiv:2111.01007*, 2021. 7

[47] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *NeurIPS*, pages 3310–3320, 2017. 7

[48] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1

[49] Akinori Tanaka. Discriminator optimal transport. *arXiv preprint arXiv:1910.06832*, 2019. 2, 6

[50] Ugo Tanielian, Thibaut Issenhuth, Elvis Dohmatob, and Jérémie Mary. Learning disconnected manifolds: a no gan's land. In *ICML*, pages 9418–9427, 2020. 2

[51] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997. 3

[52] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *ICML*, pages 6345–6353, 2019. 3

[53] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv preprint arXiv:2007.03898*, 2020. 2, 4, 6

[54] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 1, 6, 7