

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

## Supplementary Materials

Codes available at [github.com/AhmedImtiazPrio/magnet-polarity](https://github.com/AhmedImtiazPrio/magnet-polarity)

Google Colab demo [bit.ly/polarity-samp](https://bit.ly/polarity-samp)

We provide the following supplementary materials (SMs) as support of our theoretical and empirical claims. This SM is organized as follows.

Appendix A provides all the implementation details. We first provide the pseudocode for the online sampling algorithm in Appendix A.1 that allows for polarity sampling to be performed rapidly for real-time applications. Along it, we provide further details on the effect of the two hyper-parameters of polarity sampling that are  $N$  and  $k$ , namely the number of samples used to estimate the per-region singular values of  $\mathbf{A}_\omega$  for as many  $\omega$  i.e.  $N$  of them, and the number  $k$  of top-singular values to utilize (Appendix A.2). We then describe the computation times we observed on our hardware/software (Appendix A.3).

Appendix B provides the proofs for Theorem 1 and Corollary 1. Appendix C supports our claims with additional experiments on various dataset and models. First, Appendix C.1 studies how polarity sampling can help under distribution shift between the training distribution and a target distribution, this is done through colored-MNIST and NVAE. Then Appendix C.2 proposes to study the effect of polarity sampling on ProGAN which is crucial as ProGAN does not allow for truncation based control of its samples.

We conclude with Appendix D and Appendix E that provide descriptions of the datasets and additional qualitative samples from the empirical experiments performed in the main part of the paper, respectively.

## A. Implementation Details and Online Sampling Solution

### A.1. Online Algorithm

One important aspect of polarity sampling, as summarized in Algorithm 1 is the need to first sample the DGN latent space to obtain the top singular values of as many per-region slope matrices  $\mathbf{A}_\omega$  as possible. This might seem as a bottleneck if one wants to repeatedly apply polarity sampling on a same DGN. However, this is to provide an estimate of the DGN per-region change of variables, and as the DGN is not retrained nor fine-tuned, it only needs to be done once. Furthermore, this allows for an online sampling algorithm that we provide in Algorithm 2. In short, one first perform this task of estimating as many per-region top singular values as possible, once this is completed, only sampling of latent vectors  $z$  and rejection sampling based

on the corresponding  $\mathbf{A}_\omega$  matrix is done online,  $\mathbf{A}_\omega$  of the sampled  $z$  being obtained easily via  $\mathbf{A}_\omega = \mathbf{JG}(z)$ .

---

### Algorithm 2 Online Rejection Sampling Algorithm

---

**Input:** Latent space domain,  $\mathcal{D}$ ; Generator  $\mathbf{G}$ ;  $N$  change of volume scalars  $\{\sigma_1, \sigma_2, \dots, \sigma_N\}$ ; Number of singular values  $K$ ;

**while** True **do**

$z \sim U(\mathcal{D})$

$\alpha \sim U[0, 1]$

$\mathbf{A} = \mathbf{JG}(z)$

$\sigma_z = \prod_{k=1}^K K\text{-SingularValues}(\mathbf{A}, K)$

**if**  $\frac{\sigma_z^p}{\sigma_z^p + \sum_{i=1}^N \sigma_i^p} \geq \alpha$  **then**

$x \leftarrow \mathbf{G}(z)$

**return**  $x$

---

### A.2. Effect of $N$ and $k$

One important aspect of our algorithm comes from the two hyper-parameters  $N$  and  $k$ . They represent respectively the number of latent space samples to use to estimate as much  $\mathbf{A}_\omega$  as possible (recall Algorithm 1), and the number of top singular values to compute. Both represent a trade-off between exact polarity sampling, and computation complexity. We argue that in practice,  $N \approx 150K$  and  $k \approx 100$  is enough to obtain a good estimate of the polarity sampling distribution (Eq. (4)). To demonstrate that, we first provide an ablation study of the number of  $N$  and  $k$  used for polarity sampling in Tab. 3 and Tab. 4. We also present a visual inspection of the impact of  $N$  and  $k$  on the precision and recall in Fig. 9.

### A.3. Computation Times and Employed Software/Hardware

All the experiments were run on a Quadro RTX 8000 GPU, which has 48 GB of high-speed GDDR6 memory and 576 Tensor cores. For the software details we refer the reader to the provided codebase. In short, we employed TF2 (2.4 at the time of writing), all the usual Python scientific libraries such as NumPy and PyTorch. We employed the official repositories of the various models we employed with official pre-trained weights. As a note, most of the architectures can not be run on GPUs with less or equal to 12 GB of memory.

$N$	FFHQ 1024×1024			LSUN Cat 256×256		
	FID (lowest)	Precision (max)	Recall (max)	FID (lowest)	Precision (max)	Recall (max)
100K	2.63	0.80	0.59	6.38	0.69	0.31
200K	2.62	0.82	0.63	6.38	0.71	0.32
250K	2.59	0.84	0.64	6.39	0.74	0.31
300K	2.61	0.87	0.65	6.37	0.75	0.33
500K	2.58	0.90	0.67	6.34	0.77	0.33

Table 3. Ablation of  $N$  and its effect on best FID, Precision and Recall values that can be obtained by a StyleGAN2 ( $\psi = 1$ ) on the FFHQ and LSUN Cat dataset. We vary the polarity in the VGG space for FFHQ dataset, and style space for LSUN Cats, for number of singular values  $k = 30$ .

$K$	FFHQ 1024×1024		
	FID (lowest)	Precision (max)	Recall (max)
10	2.71	0.89	0.65
20	2.67	0.90	0.66
40	2.57	0.90	0.67
60	2.62	0.90	0.66
80	2.67	0.90	0.66
100	2.70	0.90	0.67

Table 4. Ablation of  $K$  and its effect on the best FID, Precision and Recall values that can be obtained by a StyleGAN2 ( $\psi = 1$ ) on the FFHQ dataset. We vary the polarity in the inception space for FFHQ dataset

We report here the Jacobian computation times for Tensorflow 2.5 with CUDA 11 and Cudnn 8 on an NVIDIA Titan RTX GPU. For StyleGAN2 pixel space, 5.03s/it; StyleGAN2 style-space, 1.12s/it; BigGAN 5.95s/it; ProgGAN 3.02s/it. For NVAE on Torch 1.6 it takes 20.3s/it. Singular value calculation for StyleGAN2 pixel space takes .005s/it, StyleGAN2 style space .008s/it, BigGAN .001s/it, ProgGAN .004s/it and NVAE .02s/it on NumPy. According to this, for StyleGAN2-e,  $N=250,000$  requires 14 days to obtain. This only needs to be done once, and it is also possible to perform online sampling once it is calculated. The time required for this is relatively small compared to the training time required for only one set of hyperparameters, which is 35 days and 11 hours<sup>1</sup>. We have added pseudocode for MaGNET sampling and online sampling in Appendix G.

**Computational Complexity.** We are computing the top- $k$  singular values of the  $D \times K$  Jacobian matrix. This can be performed in  $\mathcal{O}(DKk + Dk^2)$ . In fact, one has to project  $k$   $K$ -dimensional vectors onto the  $D \times K$  Jacobian’s matrix:  $\mathcal{O}(DKk)$  and then perform QR-decomposition of the  $D \times k$  matrix:  $\mathcal{O}(Dk^2)$ . Then,  $k$   $D$ -dimensional vectors are pro-

jected onto the transpose of the Jacobian matrix:  $\mathcal{O}(DKk)$  followed by their QR-decomposition:  $\mathcal{O}(Kk^2)$ , dominated by  $\mathcal{O}(Dk^2)$  (full SVD runs in  $\mathcal{O}(DK^2)$ ).

#### A.4. Reducing Memory Requirements

The core of polarity sampling relies on computing the top-singular values of the possibly large matrix  $A_\omega$  for a variety of regions  $\omega \in \Omega$ , discovered through latent space sampling (recall Algorithm 1). One challenge for state-of-the-art DGNs lies in the size of the matrices  $A_\omega$ . Multiple solutions exist, such as computing the top singular values through block power iterations. Doing so, the matrices  $A_\omega$  do not need to be computed entirely, only the matrix-matrix product  $A_\omega W$  and  $A_\omega^T V$  needs to be performed repeatedly (interleaved with QR decompositions). After many iterations,  $W$  estimate the top right-singular vectors of  $A_\omega$ , and  $V$  the corresponding top left-singular vectors from which the singular values can be obtained. However, we found this solution to remain computationally extensive, and found that in practice, a simpler approximation that we now describe provided sufficiently accurate estimates.

Instead of the above iterative estimation, one can instead compute the top-singular values of  $WA_\omega$  with  $W$  a semi-orthogonal matrix of shape  $D' \times D$  with  $D' < D$  (recall that  $A_\omega$  is of shape  $D \times K$ ). Doing so, we are now focusing on the singular values of  $A_\omega$  whose left-singular vectors are not orthogonal with the right singular vectors of  $W$ . While this possibly incurs an approximation error, we found that the above was sufficient to provide polarity sampling and adequate precision-recall control.

#### A.5. Applying Polarity Sampling in Style, VGG and Inception Space

We call the ambient space of the images the pixel-space, because each dimension in this space corresponds to individual pixels of the images. Apart from controlling the density of the pixel-space manifold, polarity can also be used to control the density of the style-space manifold for style based architectures such as StyleGAN{1,2,3} [28–30]. We also extend the idea of intermediate manifolds to feature space manifolds such as VGG or InceptionV3 space, which can be assumed continuous mappings of the pixel space to the corresponding models’ bottleneck embedding space. In Fig. 8-left we present comparisons between Style, Pixel, VGG and Inception space precision-recall curves for StyleGAN2-F FFHQ with  $\psi = 1$ , top- $k = 30$  and  $\rho = [-2, 2]$ . We see that the VGG and InceptionV3 curves trace almost identically. This is expected behavior since both these feature spaces correspond to perceptual features, therefore the transform they induce on the pixel space distribution is almost identical. On the other hand, the pixel space distribution saturates at high polarity at almost equal values. The point of equal precision and

<sup>1</sup><https://github.com/NVlabs/stylegan2>

recall for both the Inception and VGG spaces, occurs at a polarity of 0.1. It's clear from the figures that feature space polarity changes have a larger effect on precision and recall compared to pixel-space and style-space has the least effect on precision and recall. This could be due to the number of density transforms the style-space distribution undergoes until the VGG space, where precision and recall is calculated. In Fig. 8-right we present the polarity characteristics for StyleGAN2-E, StyleGAN2-F and StyleGAN3. For each model, we choose the best space w.r.t the pareto frontier, VGG and Inception space for StyleGAN2-E and StyleGAN2-F, and pixel-space for StyleGAN3. Notice that StyleGAN3 exceeds the recall of the other two models for negative polarity, while matching the precision for StyleGAN2-E.

## B. Proofs

The proofs of the two main claims of the paper heavily rely on the spline form of the DGN input-output mapping from Eq. (1). For more background on the form of the latent space partition  $\Omega$ , the per-region affine mappings  $\mathbf{A}_\omega, \mathbf{b}_\omega, \forall \omega \in \Omega$  and further discussion on how to deal with DGN including smooth activation functions, we refer the reader to [4], and in particular to [24] for DGN specific results.

### B.1. Proof of Theorem 1

*Proof.* We will be doing the change of variables  $\mathbf{z} = (\mathbf{A}_\omega^T \mathbf{A}_\omega)^{-1} \mathbf{A}_\omega^T (\mathbf{x} - \mathbf{b}_\omega) \triangleq \mathbf{A}_\omega^\dagger (\mathbf{x} - \mathbf{b}_\omega)$ , also notice that  $J_{G^{-1}}(\mathbf{x}) = \mathbf{A}_\omega^\dagger$ . First, we know that  $P_{G(\mathbf{z})}(\mathbf{x} \in w) = P_{\mathbf{z}}(\mathbf{z} \in G^{-1}(w)) = \int_{G^{-1}(w)} p_{\mathbf{z}}(\mathbf{z}) d\mathbf{z}$  which is well defined based on our full rank assumptions. We then proceed by

$$\begin{aligned} P_G(\mathbf{x} \in w) &= \sum_{\omega \in \Omega} \int_{\omega \cap w} p_{\mathbf{z}}(G^{-1}(\mathbf{x})) \\ &\quad \times \sqrt{\det(J_{G^{-1}}(\mathbf{x})^T J_{G^{-1}}(\mathbf{x}))} d\mathbf{x} \\ &= \sum_{\omega \in \Omega} \int_{\omega \cap w} p_{\mathbf{z}}(G^{-1}(\mathbf{x})) \\ &\quad \times \sqrt{\det((\mathbf{A}_\omega^\dagger)^T \mathbf{A}_\omega^\dagger)} d\mathbf{x} \\ &= \sum_{\omega \in \Omega} \int_{\omega \cap w} p_{\mathbf{z}}(G^{-1}(\mathbf{x})) \frac{1}{\sqrt{\det(\mathbf{A}_\omega^T \mathbf{A}_\omega)}} d\mathbf{x}, \end{aligned}$$

where the second to third equality follows by noticing that  $\sigma_i(\mathbf{A}^\dagger) = (\sigma_i(\mathbf{A}))^{-1}$  which can be showed easily by replacing  $\mathbf{A}_\omega$  with its SVD and unrolling the product of matrices. Now considering a uniform latent distribution case on a bounded domain  $U$  in the DGN latent space we obtain

by substitution in the above result

$$p_G(\mathbf{x}) = \frac{\sum_{\omega \in \Omega} \mathbb{1}_{\mathbf{x} \in \omega} \det(\mathbf{A}_\omega^T \mathbf{A}_\omega)^{-\frac{1}{2}}}{\text{Vol}(U)}, \quad (5)$$

leading to the desired result.  $\square$

### B.2. Proof of Corollary 1

*Proof.* The proof of this result largely relies on Theorem 1. Taking back our previous result, we know that

$$p_G(\mathbf{x}) = \sum_{\omega \in \Omega} p_{\mathbf{z}}(G^{-1}(\mathbf{x})) \mathbb{1}_{\{G^{-1}(\mathbf{x}) \in \omega\}} \frac{1}{\sqrt{\det(\mathbf{A}_\omega^T \mathbf{A}_\omega)}} d\mathbf{x}. \quad (6)$$

However, recall that polarity sampling leverages the prior probability given by

$$p_\rho(\mathbf{z}) = \frac{1}{\kappa} \sum_{\omega \in \Omega} \det(\mathbf{A}_\omega^T \mathbf{A}_\omega)^{\frac{\rho}{2}} \mathbb{1}_{\{\mathbf{z} \in \omega\}}, \quad (7)$$

which, after replacing  $G^{-1}(\mathbf{x})$  with its corresponding  $\mathbf{z}$  becomes

$$p_G(\mathbf{x}) = \sum_{\omega \in \Omega} \frac{1}{\kappa} \frac{\det(\mathbf{A}_\omega^T \mathbf{A}_\omega)^{\frac{\rho}{2}}}{\sqrt{\det(\mathbf{A}_\omega^T \mathbf{A}_\omega)}} \mathbb{1}_{\{\mathbf{z} \in \omega'\}} d\mathbf{x}, \quad (8)$$

and simplifies to

$$p_G(\mathbf{x}) = \sum_{\omega \in \Omega} \frac{1}{\kappa} \det(\mathbf{A}_\omega^T \mathbf{A}_\omega)^{\frac{\rho-1}{2}} \mathbb{1}_{\{\mathbf{z} \in \omega'\}} d\mathbf{x}, \quad (9)$$

leading to the desired result. Note that when  $\rho = 1$  then the density is uniform onto the DGN manifold, when  $\rho = 0$ , one recovers the original DGN density onto the manifold, and in the extreme cases, only the region with highest or lowest probability would be sampled i.e. the modes or anti-modes.  $\square$

## C. Extra experiments

### C.1. Polarity Helps Under Distribution Shift

A last benefit of polarity sampling is to adapt a sampling distribution to a reference distribution that suffered a distribution shift. For example, this could occur when training a DGN on a training set, and using it for content generation with a slightly different type target samples. In fact, as long as the distribution shift remains reachable by the model i.e. in the support of  $p_G$ , altering the value of  $\rho$  will help to shift the sampling distribution, possible to better match the target one. In all generality, there is no guarantee that any benefit would happen for  $\rho \neq 0$ , however, for the particular case where the distribution shift only changes the way the samples are distributed (on the same domain), we observe in Fig. 14 that  $\rho \neq 0$  can provide benefit. To control the

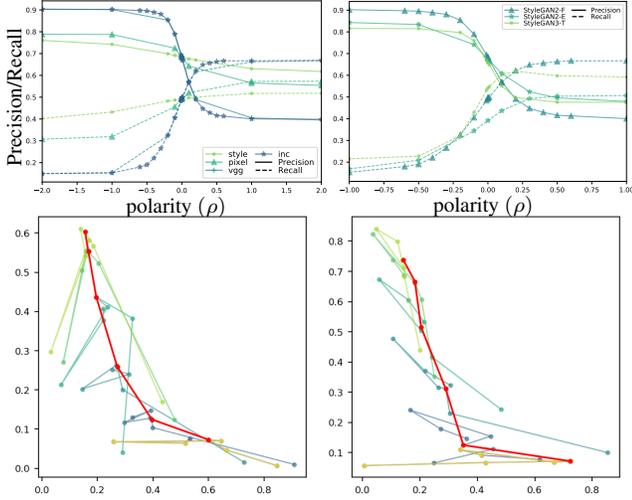


Figure 8. Top: Precision Recall tradeoff for polarity sweep on VGG, Inception and Pixel space distributions. Bottom: BigGAN-deep Imagenet Pareto curves obtained for a few classes, in red is the baseline while each scatter point can be reached by varying truncation and  $\rho$ . Calculated with 1300 real and generated samples.

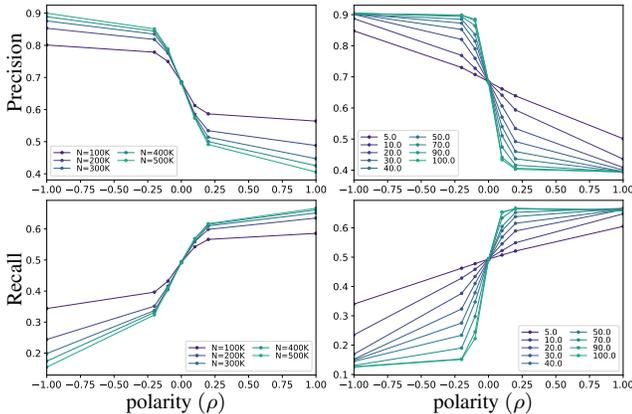


Figure 9. Effect of Polarity Sampling on Precision (top) and Recall (bottom) of a StyleGAN2-F model pretrained on FFHQ for varying number of top- $k$  singular values (right) and varying number of latent space samples  $N$  (left) used to obtain per-region slope matrix  $\mathbf{A}_\omega$  singular values (recall Sec. 3.3 and Algorithm 1). The trend in metrics stabilizes when using around  $N \approx 300,000$  latent space samples. Increasing the number of top- $k$  singular values to use, amplifies the effect of polarity, saturating at around  $k = 50$ .

experimental setting, we took the colored-MNIST dataset and the NVAE DGN model [53] and produce a training set with a Gaussian hue distribution favoring blue and two test set, one with same hue distribution and one with uniform hue distribution. We observe that  $\rho$  can provide a beneficial distribution shift to go from the biased-hue samples to the uniform-hue one.

## C.2. ProGAN Polarity Sweep

Previously in Sec. 4.1, we have drawn note to the fact that ProGAN [27], an architecture which is widely used, but does not incorporate truncation, can also be controlled via polarity sampling. In Tab. 5 we present precision-recall characteristics for polarity sweep on ProGAN. As control, we also perform latent space truncation as in [8] by sampling a truncated gaussian distribution, parameterized by its support  $[-\beta, \beta]$ . We change  $\beta$  between  $[10^{-10}, 10]$  and notice that for  $\beta$  smaller than  $10^{-4}$ , the generator collapses to 0 precision and recall. Other than that, it maintains a precision of 0.72 and recall of 0.34. Using polarity sweep, we also exceed the baseline FID on CelebA HQ 1024x1024 attained by ProGAN; polarity of  $-.01$  in pixel-space reduces the FID from 7.37 to 7.28.

## C.3. FID for truncated models

While the FID improvement for some of the methods we present are not significantly large, we see that for truncated models, i.e., models with  $\psi < 1$ ,  $\rho > 0$  provides significant FID improvements by increasing diversity, e.g., for StyleGAN2-FFHQ with  $\psi = \{.9, .7, .5\}$ , increasing  $\rho > 0$  improves FID by  $\{.69, 8.11, 11.1\}$  points. Such truncation is commonly used in practice for qualitative experiments, making polarity sampling particularly relevant in such settings. Since truncation reduces the range of the generator and polarity increases the diversity of sampling within the range, both can be combined to achieve greater FID improvements.

## C.4. NVAE Negative Log-Likelihood for Varying $\rho$

To validate the effect of  $\rho$  on the likelihood of generated samples, we estimate the negative log-likelihood of samples generated via an NVAE trained on colored-MNIST while varying  $\rho$ . We generate 5000 samples each for  $\rho = \{-5, -1, -.5, -.1, 0, .1, .5, 1, 5\}$  which yields negative log-likelihood values of  $\{3.0, 3.2, 3.5, 3.8, 3.9, 4.1, 4.3, 4.6, 5.1\} \times 10^{-2}$  bits/dim. This shows that decreasing  $\rho < 0$  samples high-likelihood points while increasing  $\rho > 0$  samples lower-likelihood points compared to standard sampling ( $\rho = 0$ ).

## C.5. Perceptual Path Length for Constant Latent Shifts

In Sec. 5.2 we present the PPL variation for a  $10^{-4}$  interpolation step from a latent space point towards another random latent space point. To evaluate the perceptual smoothness around regions of the latent space, we also calculate the PPL for paths of length .3 starting from individual latent space points towards random directions. We see that for both StyleGAN2-FFHQ and BigGAN-Imagenet, PPL decreases monotonically with  $\rho < 0$  whereas first de-



Figure 10. Modes for BigGAN-deep trained on Imagenet, and conditioned on a specific class: “pug” (top left), “lion” (top right), “cheeseburger” (bottom left) and “pomerian” (bottom right). We observe that the modes correspond to nearly aligned faces with little to no background. Variation of colors and sizes can be seen across the modes. The same observation can be made for the cheeseburger, nearly no background is present, and the shape is consistent to a typical cheeseburger “template”. See Fig. 19 for additional classes.

creases and then increases for  $\psi < 1$ . For StyleGAN2-FFHQ we acquire  $\text{PPL} = \{281, 316\}$  for  $\psi = \{0.1, 0\}$  and  $\text{PPL} = \{274, 271\}$  for  $\rho = \{-2, -10\}$ . For BigGAN-GoldenRetriever we get PPL 35.9 for  $\psi = 0$  and 0.20 for

$\rho = -2$ . This possibly indicates that decreasing truncation might not always lead to perceptually smoother regions whereas decreasing polarity does.

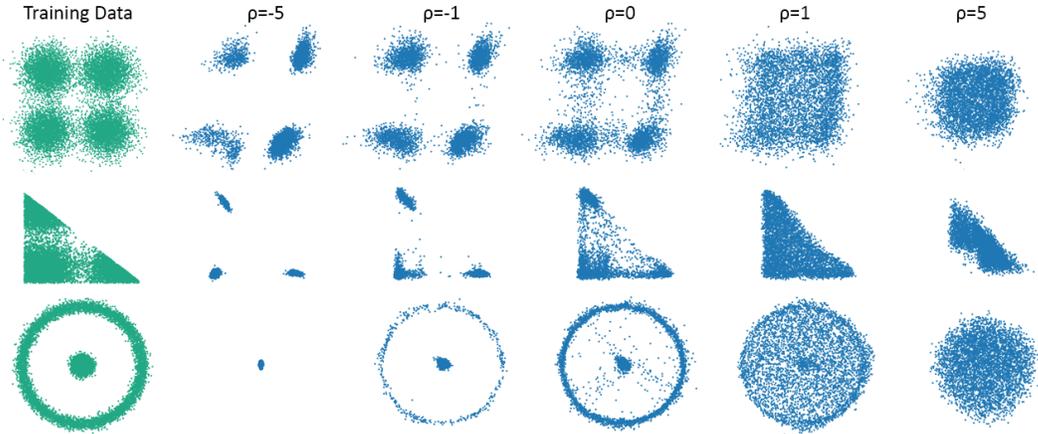


Figure 11. Polarity sweep for a WGAN trained on 2D toy datasets with 4 gaussians (top row), 4 gaussians with triangular domain (middle row) and two circles (bottom row).

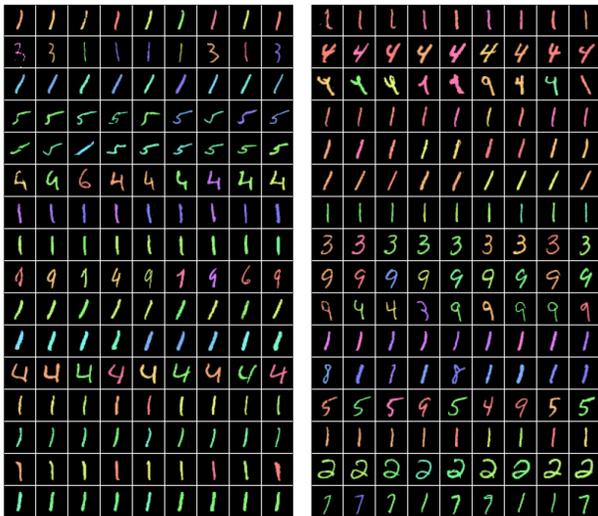


Figure 12. Modes for a VAE trained on colored-MNIST with 8 nearest neighbors. The leftmost column for each figure contains generated samples. Notice the higher prevalence of digit 1. Due to low pixel variations, digit 1 samples have high density on the manifold.

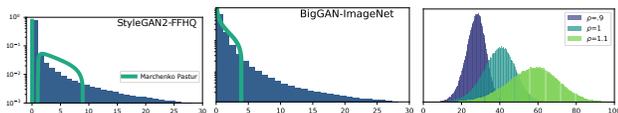


Figure 13. Singular value distribution and Marchenko Pastur distribution fit for StyleGAN2-FFHQ (left) and BigGAN-ImageNet (middle). Log-sigma distribution for StyleGAN2-FFHQ with varying  $\rho$  (right).

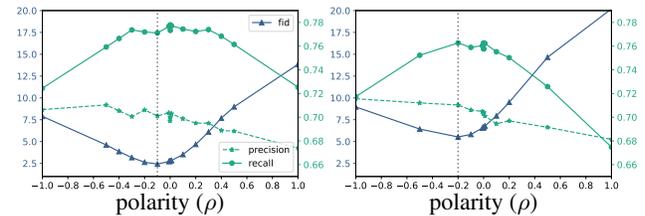


Figure 14. FID, precision and recall for an NVAE trained on colored-MNIST with hue bias. Metrics are calculated for a test dataset with hue distribution **Left**: identical to training and **Right**: uniformly distributed across digit classes. **Polarity allows adapting the DGN output distribution to balance possible distribution shifts.**

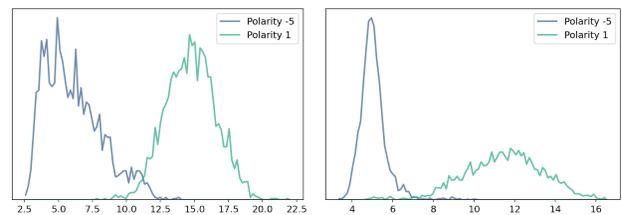


Figure 15. Distribution of  $l_2$  distance to 3 MNIST training set nearest neighbors, for 1000 generated MNIST samples from WGAN (left) and NVAE (right). For  $\rho = -5$  we see that both distributions have a peak around 5. For WGAN the distribution has a significantly longer tail compared to NVAE, indicating that the WGAN modes don't necessarily coincide with training points.

## D. Dataset Description

### D.1. colored-MNIST

We perform controlled experiments on NVAE [53] by training on datasets with and without controllable distribution shifts. To control the shift, we colorize MNIST with hue ranging  $[0, \pi]$  by 1) uniformly sampling the hue and 2)

CelebAHQ 1024×1024						
$\rho$	$\rho \leq 0$			$\rho > 0$		
	FID	Precision	Recall	FID	Precision	Recall
0	7.37	.73	.34	-	-	-
0.01	<b>7.28</b>	.73	.34	7.45	.73	.35
0.1	7.41	.76	.31	8.95	.68	.38
1	12.65	.85	.19	17.96	.58	.48
2	13.09	<b>.86</b>	.19	18.54	.58	<b>.48</b>

Table 5. FID, Precision and Recall metrics of ProGAN [27] with polarity sweep in the pixel space.

sampling the hue for each image from a truncated normal distribution, with a truncation scale of 2.

## D.2. LSUN Dataset

We use the LSUN dataset [54] available at the official website<sup>2</sup>. We preprocess the dataset using the StyleGAN2 repository<sup>3</sup>.

## D.3. AFHQv2 and FFHQ

We use the version 2 of AFHQ that was released prepackaged with StyleGAN3 for our experiments. For FFHQ we use also use TFRecords provided with StyleGAN2.

## D.4. License

The majority of Polarity-Sampling is licensed under CC-BY-NC, however portions of the project are available under separate license terms: NVAE, StyleGAN2 and StyleGAN3 are licensed under the NVIDIA license; guided-diffusion is licensed under the MIT license.<sup>4</sup>

## E. Qualitative results

We provide in the following pages, Figs. 16 to 19 that correspond to LSUN Cats, LSUN Cars, AFHQv2 samples with varying  $\rho$  values and 800 Imagenet modes. For LSUN Cars and LSUN Cats we draw comparisons between varying truncation and varying polarity independently.

<sup>2</sup><https://www.yf.io/p/lsun>

<sup>3</sup><https://github.com/NVlabs/stylegan2>

<sup>4</sup><https://github.com/openai/guided-diffusion/blob/main/LICENSE>

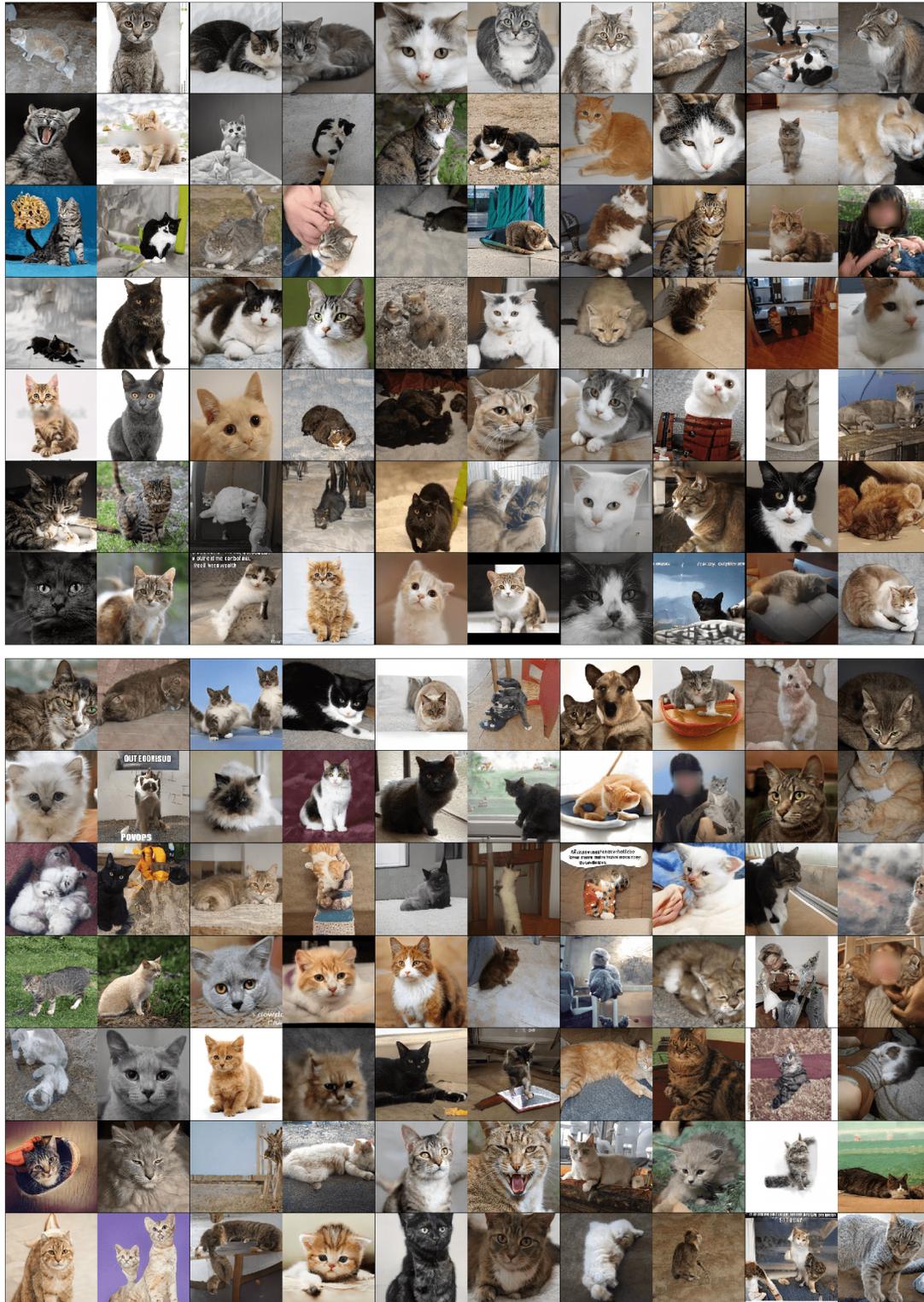


Figure 16. Uncensored samples of LSUN Cats using (top)  $\rho = \{-1, -.5, -.2, -.1, 0\}$ ,  $\psi = .8$  and (bottom)  $\psi = \{.7, .73, .75, .77, .8\}$ ; both representing regions with roughly an equal span of recall score on Fig. 3. Notice the significant precision of the left-most columns of top compared to the left-most of bottom, where at equal diversity, top has significantly higher precision score.



Figure 17. Uncurated samples of LSUN Cars using (top)  $\rho = \{-.1, -.075, -.05, -.025, 0\}$ ,  $\psi = .7$  and (bottom)  $\psi = \{.5, .55, .6, .65, .7\}$ ; both representing regions with roughly an equal span of precision score on Fig. 3 Notice the significant diversity in (top) especially in the leftmost columns, where the recall score is significantly higher than that of the leftmost column of bottom left, with equal precision.



Figure 18. Uncurated samples of AFHQv2 using  $\rho = \{-2, -1, -.5, -.2, 0, .01, .1, .2, .5\}$  and  $\psi = .9$  in pixel-space. As we move right from baseline (middle column) we see an increase in texture diversity of images, whereas, moving left, we see images with smoother textures.

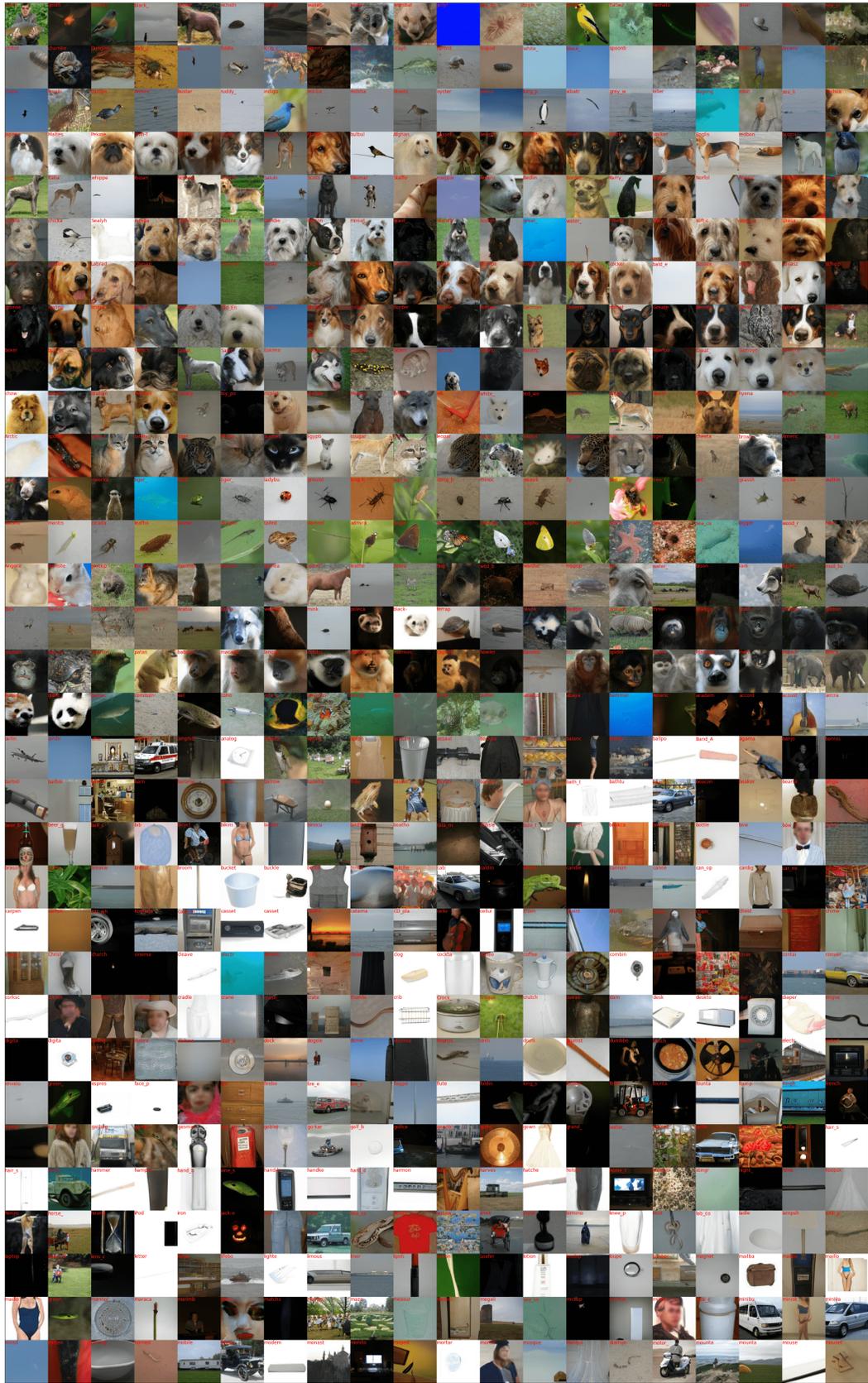


Figure 19. Depiction of a single mode (large negative  $\rho$ ) for each class of the first 800 Imagenet classes.