How Useful is the Density Learned by GANs for Computer Vision?

Roy Friedman, Yair Weiss
The Hebrew University of Jerusalem
roy.friedman@mail.huji.ac.il

Abstract

Generative Adversarial Networks (GANs) are trained to minimize the distance between the model density, $p_\theta(x)$, and the distribution of the data, $p_{data}(x)$. In this paper we ask: is the prior learned by GANs, $p_\theta(x)$, useful for downstream computer vision tasks such as classification, outlier detection or image restoration? To answer this question, we conduct a series of experiments on state-of-the-art GANs trained on various datasets with varying optimization criteria. We show that even though these GANs can generate remarkably realistic images and perform well under the task of image generation, using the learned prior $p_\theta(x)$ to solve computer vision tasks yields poor performance. We explain the origin of these results by showing that $p_\theta(x)$ has almost perfect correlation with a model which was trained on small image patches.

1. Introduction

Generative models fundamentally aim to approximate the distribution of training data, $p_{\text{data}}(x)$ [7]. When such models approximate the data distribution well, they can be used to generate realistic images, a capability that has driven much recent research in image generation [10, 15, 28]. However, having access to the distribution from which natural images are sampled holds tremendous potential beyond image synthesis. Theoretically, it is well known that optimal performance in tasks such as image recognition and restoration is possible when we have access to the true distribution of the training data [7]. This suggests a compelling possibility: rather than developing specialized models for individual tasks, a single well-trained generative model could address multiple inference challenges at once. In particular, if the model's density, $p_\theta(x)$, closely approximates the data distribution, $p_{\text{data}}(x)$, then numerous inference tasks can be solved using only the model's density $p_\theta(x)$.

Among the various generative modeling frameworks, Generative Adversarial Networks (GANs) [8] have gained particular prominence. When training a GAN, the distance between the model's density $p_\theta(x)$ and that of the data $p_{\text{data}}(x)$ is minimized [1, 8, 21, 32], through a joint optimization of a generator network $G_\theta(\cdot)$ and a discriminator. In this work we ask: to what extent is the prior learned by the GAN, $p_\theta(x)$, useful for downstream tasks?

When analyzed in this way, we find that the GAN prior is inefficient for solving downstream tasks such as classification, outlier detection and image restoration. This limitation persists across GANs with different architectures, under a wide range of optimization criteria and for different datasets. Moreover, even GANs that are state-of-the-art in terms of FID [10] (such as StyleGAN-XL [28]) exhibit this deficiency, and the ability of the different GANs to solve downstream tasks seems independent from their sample generation quality as measured by FID. This is especially upsetting in the context of image restoration, where the ability to generate realistic images can be considered an important factor [25].

To better understand this behavior we analyze the densities assigned by the GANs. Consistently, we find that the images whose density is highest are those with strong correlations in small regions within the image, such as images of a single color or SVHN images. This suggests that the GAN density is dominated by local statistics inside the image. To test this hypothesis, we compare the GAN prior to a model trained only on local features: a Gaussian mixture model (GMM) trained on non-overlapping, 8×8 patches. Remarkably, we find that the density assigned by different GANs and the simple patch model are highly correlated, frequently with a correlation above 90%. This high correlation of the GAN density to simplified patch models explains the poor performance of GANs in tasks such as classification and outlier detection.

2. Inference with GANs

2.1. Calculating the GAN density

To use the prior learned by the model in order to solve downstream tasks, the log-density $\log p_\theta(x)$ needs to be calculated. In contrast to many other generative models, the GAN log-density cannot be calculated explicitly. Worse, as the GAN defines a transformation from a low-dimensional

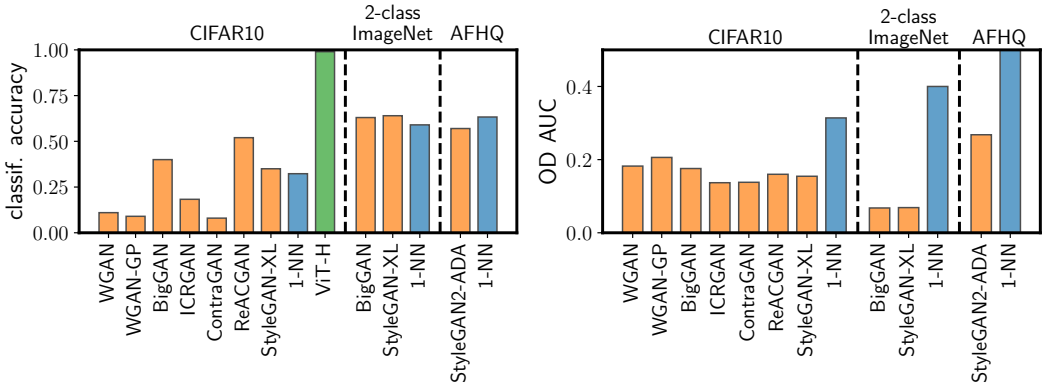


Figure 1. **Left:** average accuracy of GANs as generative classifiers. **Right:** AUC when using GANs for outlier detection.

latent space, $\mathcal{Z}$, to the data space $\mathcal{X}$, naively any sample off of the manifold defined by the GAN's generator will have a density of 0.

To overcome this limitation and meaningfully apply GAN priors, we adopt an approach similar to variational auto-encoders (VAEs, [18]). While VAEs are also frequently defined as a transformation from a low-dimensional space to the high-dimensional data space, they assign densities larger than 0 to points outside their manifold. This is made possible by assuming that each sample x is observed after some noise has been added to it. Thus, it is assumed that each sample was created under the following process:

$$x = G_\theta(z) + \eta \tag{1}$$

where $G_\theta(\cdot)$ is the mapping from the latent space to the sample space and $\eta \sim p_{\text{noise}}$ represents observation noise, sampled independently from the latent code z. Adding an observation model in this way defines a conditional distribution for each sample given the latent code, $p_\theta(x|z)$, determined by the distribution of the observation noise. In this work, we consider GANs analogously to VAEs, with an additive Gaussian noise model with variance σ^2, which is common for many other low-dimensional latent generative models [17, 27, 31]. While we primarily use this Gaussian model, our experiments with a more perceptual distance yielded qualitatively similar results.

Under this noise model, the log-density is given by the following equation:

$$\log p_\theta(x) = \log \int p_\theta(x|z) \cdot p_\theta(z) dz \tag{2}$$

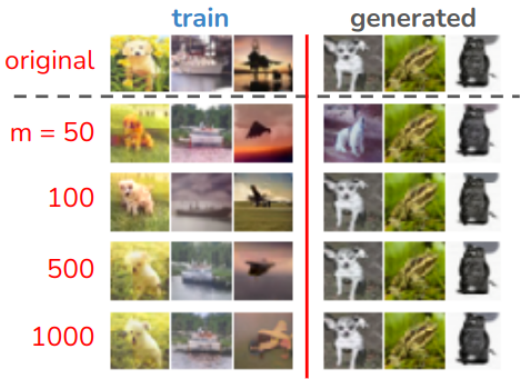
for unconditional GANs. When using conditional GANs, the log-density is defined per-class $\log p_\theta(x|z;c)$, using the generator's conditioning on the class c. As the above integral is intractable, we follow Wu et al. [33] and use annealed importance sampling (AIS, Neal 24) to estimate the log-density. In all of our experiments, we use AIS with 500 steps, 4 chains, a Hamiltonian Monte Carlo (HMC) kernel and 10 leap-frogs to estimate the log-density of a sample. These hyperparameters were chosen to mitigate the high computational load of running AIS while still extracting accurate estimates of the log-likelihood. To set the variance of the observation model, σ^2, we used the maximum likelihood estimator, which is the mean variance of the reconstruction error of training samples. In all cases, the signal-to-noise ratio of the images was high; that is, the variance of the observation noise used was comparatively small.

2.2. Results

To ascertain if the priors learned by modern GANs are effective for computer vision tasks, we run a series of experiments on three different datasets: CIFAR10, two classes of ImageNet, and AFHQ. We use a wide range of GANs, which vary in terms of training procedure and architecture: StyleGAN-XL [28], ReACGAN [12], BigGAN [5] and BigGAN-DiffAug [34], StyleGAN2-ADA [14], ICRGAN [13, 35], WGAN [13, 22], ContraGAN [11, 13] and MHGAN [13, 16].

Under these different datasets and GANs, we consider three standard computer vision tasks: classification, outlier detection and compressed sensing. For the tasks of outlier detection and classification, we additionally compare the performance of the different GANs to a simple ℓ_2 nearest-neighbor (1-NN) estimator.

Classification: Generative classification is carried out by calculating the log-density of the sample x under a class-specific model. The sample is then categorized to the class whose paired model gives highest log-density: $\hat{c}(x) = \arg\max_c p_\theta(x;c)$. When $p_\theta(x;c) = p_{\text{data}}(x;c)$, this classification scheme can be shown to be optimal in terms of test accuracy [7]. In these experiments, we consider classification of CIFAR10, a subset of ImageNet for binary classification, and all three classes of AFHQ (cat, dog, and wild). The results of using GANs as generative classifiers can be seen



model	data	m =50	100	500	1000
WGAN [2]	train	15.5	17.2	20.5	20.8
	generated	19.3	26.2	33.6	35.0
WGAN-GP [9]	train	16.5	17.5	21.0	21.8
	generated	17.4	19.8	39.6	45.8
BigGAN [5]	train	16.6	18.3	20.9	21.3
	generated	20.0	23.3	29.7	29.2
ICRGAN [35]	train	15.8	17.7	20.2	20.5
	generated	23.0	34.5	45.5	45.5
ContraGAN [11]	train	15.4	17.3	20.2	20.3
	generated	19.1	26.9	41.9	43.1
ReACGAN [12]	train	16.0	17.8	20.6	20.9
	generated	21.6	32.4	44.3	44.4
StyleGAN-XL [28]	train	16.2	17.2	18.2	18.1
	generated	28.9	31.7	33.0	32.3
Gaussian prior	train	17.9	19.6	25.9	28.8

Figure 2. **Left:** examples of reconstructions by StyleGAN-XL for training and images that StyleGAN-XL itself generated, under different numbers of measurements. For training images, the GAN captures the overall color information of the image but is never able to reconstruct the main subject, even when there are numerous measurements. On the other hand, the GAN is adept at reconstructing generated images, which are truly from the distribution of the GAN, meaning the reconstruction algorithm itself is not the origin of the poor performance. **Right:** a table of average PSNRs of different GANs on training images and images that they themselves generated. The numbers underlined have the highest PSNR on training images. The GANs are systematically worse than a Gaussian prior at solving compressed sensing.

in Figure 1 (left). All of the GANs, even those that are able to generate extremely realistic samples such as StyleGAN-XL [28] with an FID of 1.88 on CIFAR10, learn poor priors for the task of classification. As a comparison, the GANs perform on par with the simple 1-NN baseline classifier using ℓ_2 in pixel space.

Outlier detection: One way generative models can be used to detect outliers is by defining some threshold τ and deeming any sample x with an assigned density lower than this threshold, $\log p_\theta(x) < \tau$, as an outlier [3]. In a similar way, for the 1-NN baseline a sample is determined to be an outlier if its distance from the nearest training sample is larger than a threshold τ. The performance of the model as an outlier detector is evaluated using the area under the ROC curve (AUC), calculated over different choices for the threshold τ. When the AUC is lower than $1/2$, this means that the estimator is worse than random at detecting outliers. Under this framework, we evaluate GANs as outlier detectors, where the outliers considered are images of a single color or scaled images from SVHN. For all of the GANs considered in this work, we find that their AUC is worse than random, as can be seen in Figure 1(right). In comparison, the simple 1-NN baseline always detects outliers better than the GANs.

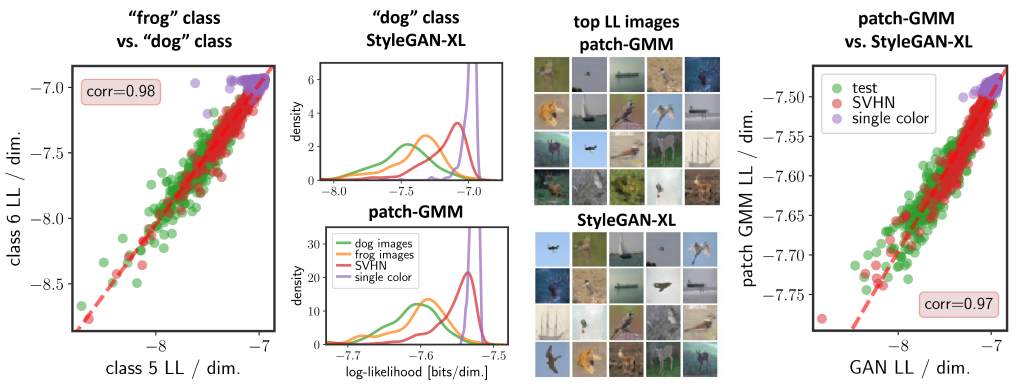
Compressed sensing: Finally, we test whether the GAN prior is effective for solving image restoration problems. Sampling from the GAN prior results in images that seem to come from the true data distribution, potentially making them well suited to the task of image restoration in general, and compressed sensing in particular [4, 6, 26]. In compressed sensing, an image $x \in \mathbb{R}^d$ needs to be reconstructed from a small number of (noisy) measurements of the signal, $y = Mx + \epsilon$. Here it is assumed that the number of observations is much smaller than the true dimensionality of the data $m \ll d$, and the matrix $M \in \mathbb{R}^{m \times d}$ projects the full image onto this low-dimensional space. $\epsilon \in \mathbb{R}^m$ is a noise vector. This task can be solved by optimizing the posterior distribution to find the maximum a-priori (MAP) estimate. When $\epsilon \sim \mathcal{N}\left(0, I\lambda^2\right)$, this is equivalent to optimizing:

$$\hat{z} = \arg\max_z \{\log p_\theta(z) + \log p_\theta(y|z)\} \tag{3}$$

$$= \arg\max_z \left\{ \frac{1}{\lambda^2} \|y - MG_\theta(z)\|^2 + \log p_\theta(z) \right\} \tag{4}$$

Once $\hat{z}$ is found, the estimate for the restored image is given by $\hat{x} = G_\theta(\hat{z})$. This method for image restoration was suggested by Bora et al. and directly uses the GAN prior to solve the problem of compressed sensing. We use a probabilistic principal component analysis (pPCA, [31]) prior as a baseline, fitted to training images from CIFAR10 with 500 components. In essence, this baseline is a Gaussian prior over CIFAR10 images.

In our experiments, we follow common practice and sample the elements of the measurement matrix to be i.i.d. Gaussian, $M_{ij} \sim \mathcal{N}(0, 1/m)$, where m is the number of measurements [4]. On top of this, the added noise is Gaussian with a standard deviation of $\lambda = 0.01$. The latent code $\hat{z}$ was optimized using Adam from multiple different initializations, of which we report the peak signal-to-noise ratio (PSNR) of the best restoration. Figure 2 (left) qualitatively shows that while images the GAN itself generated (i.e. are from $p_\theta(x)$) can be easily restored, the GAN is unable to restore images that *were seen during training*. These results are further backed by the average PSNRs achieved by different GANs on this task (Figure 2, right). Moreover, the PSNR achieved by the GANs on training images is worse



GAN	corr.	FID
CIFAR10		
WGAN	0.8	107.7
WGAN-GP	0.84	53.9
BigGAN	0.95	8.7
ICRGAN	0.96	7.4
ContraGAN	0.96	6
ReACGAN	0.96	3.9
StyleGAN-XL	0.96	1.9
ImageNet		
BigGAN	0.87	8.5
StyleGAN-XL	0.92	1.9

Figure 3. **Left:** a scatter plot of the LLs assigned by StyleGAN-XL conditioned on the "frog" class versus the same in the "dog" class. **Middle left:** histograms for the likelihoods assigned by StyleGAN-XL (top) and the patch-GMM (bottom) trained on the "dog" class of CIFAR10. The ranking of different groups of images is retained between both models. **Middle:** the 20 CIFAR10 test images that were assigned the highest likelihoods according to the patch-GMM (top) and StyleGAN-XL (bottom). While the models are different from each other, the top images for each are essentially shuffled versions of the same set of images. **Middle right:** scatter plot of the log-likelihoods of a patch-GMM trained on 8x8 independent patches as a function of the log-likelihoods assigned by StyleGAN-XL. **Right:** table of the correlations of different GANs with the path-GMM averaged across classes, together with their respective FID.

than the Gaussian prior, for any number of measurements.

3. The GAN prior

The above results paint a grim picture; priors of modern GANs, those that are state-of-the-art in terms of image generation, are ineffective at solving other computer vision tasks. Analyzing the densities assigned to images can give us a partial understanding as to why this is the case. In classification, different classes were assigned the same log-densities (Figure 3, left), resulting in subpar classification performance. On the other hand, images such as those of a single color or SVHN images, were assigned high-density relative to the train images (Figure 3), explaining the poor outlier detection performance. Previous works have shown that other deep generative models, whose likelihoods can be explicitly calculated such as VAEs and normalizing flows, also suffer from some of these problems [19, 20, 23, 29, 30]. As an explanation, Schirrmeister et al. posited that the prior implied by such models is general to all natural images, and is thus dominated by the statistics of small regions within the image.

To test whether this is also the case for GANs, we compare the likelihoods assigned by GANs to those assigned by a model explicitly trained on local statistics - small, independent, patches of the image. To do so, we train a Gaussian mixture model (GMM) on 8×8 non-overlapping patches of the data, models which we call patch-GMMs. We use GMMs, instead of GANs, to model small patches in the images for two reasons: (1) it was shown in previous works that these GMMs model small patches very well [36], and (2) calculating the log-likelihood assigned by GMMs can be done in closed form, unlike for GANs.

Qualitatively, we see that the patch-GMM likelihoods behave similarly to those of the GANs (Figure 3, middle left). In fact, the same set of 20 CIFAR10 images have the highest likelihoods out of the test images under both models (Figure 3, middle right). Even more surprisingly, for many of the GANs we analyzed we found that the correlation between the patch-GMM density and the GAN density is very high, frequently above 95%.

4. Discussion

One of the motivations for designing and training generative models is to facilitate the solution of downstream tasks. In this work, we asked whether the priors of GANs with exceptional image generation abilities are useful in various inference tasks and found that the answer is overwhelmingly in the negative. We then showed that this poor performance can be attributed to the strong local bias inherent to the priors learned by many GANs.

Even though this work has presented an overall negative view of GANs, they are still extremely powerful data samplers. Indeed, this fact was taken advantage of in domains such as image manipulation, style transfer and data augmentation, to great effect. However, our work shows that a more cautious stance should be taken when attempting to use GANs as priors for the true data distribution.

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks, 2017. 1

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 3

[3] Christopher M Bishop. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing*, 141(4):217–222, 1994. 3

[4] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *International conference on machine learning*, pages 537–546. PMLR, 2017. 3

[5] 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

[6] Giannis Daras, Joseph Dean, Ajil Jalal, and Alexandros G Dimakis. Intermediate layer optimization for inverse problems using deep generative models. *arXiv preprint arXiv:2102.07364*, 2021. 3

[7] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification and scene analysis*. Wiley New York, 1973. 1, 2

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 1

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017. 3

[10] 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. *Advances in neural information processing systems*, 30, 2017. 1

[11] Minguk Kang and Jaesik Park. Contragan: Contrastive learning for conditional image generation. *Advances in Neural Information Processing Systems*, 33:21357–21369, 2020. 2, 3

[12] Minguk Kang, Woohyeon Shim, Minsu Cho, and Jaesik Park. Rebooting acgan: Auxiliary classifier gans with stable training. *Advances in neural information processing systems*, 34:23505–23518, 2021. 2, 3

[13] Minguk Kang, Joonghyuk Shin, and Jaesik Park. Studiogan: a taxonomy and benchmark of gans for image synthesis. *arXiv preprint arXiv:2206.09479*, 2022. 2

[14] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *Advances in neural information processing systems*, 33:12104–12114, 2020. 2

[15] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

[16] Ilya Kavalerov, Wojciech Czaja, and Rama Chellappa. A multi-class hinge loss for conditional gans. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1290–1299, 2021. 2

[17] Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021. 2

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. *Advances in neural information processing systems*, 33:20578–20589, 2020. 4

[20] Ryen Krusinga, Sohil Shah, Matthias Zwicker, Tom Goldstein, and David Jacobs. Understanding the (un) interpretability of natural image distributions using generative models. *arXiv preprint arXiv:1901.01499*, 2019. 4

[21] Tengyuan Liang. How well generative adversarial networks learn distributions. *Journal of Machine Learning Research*, 22(228):1–41, 2021. 1

[22] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *arXiv preprint arXiv:1802.05637*, 2018. 2

[23] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018. 4

[24] Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11:125–139, 2001. 2

[25] Yohan Poirier-Ginter and Jean-François Lalonde. Robust unsupervised stylegan image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22292–22301, 2023. 1

[26] Ankit Raj, Yuqi Li, and Yoram Bresler. Gan-based projector for faster recovery with convergence guarantees in linear inverse problems. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5602–5611, 2019. 3

[27] Eitan Richardson and Yair Weiss. On gans and gmms. *Advances in neural information processing systems*, 31, 2018. 2

[28] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 1, 2, 3

[29] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. *Advances in Neural Information Processing Systems*, 33:21038–21049, 2020. 4

[30] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019. 4

[31] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999. 2, 3

[32] Ananya Uppal, Shashank Singh, and Barnabás Póczos. Nonparametric density estimation & convergence rates for gans under besov ipm losses. *Advances in neural information processing systems*, 32, 2019. 1

[33] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016. 2

[34] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33:7559–7570, 2020. 2

[35] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11033–11041, 2021. 2, 3

[36] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 international conference on computer vision*, pages 479–486. IEEE, 2011. 4