UNDERSTANDING THE (UN)INTERPRETABILITY OF NATURAL IMAGE DISTRIBUTIONS USING GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Probability density estimation is a classical and well studied problem, but standard density estimation methods have historically lacked the power to model complex and high-dimensional image distributions. More recent generative models leverage the power of neural networks to implicitly learn and represent probability models over complex images. We describe methods to extract explicit probability density estimates from GANs, and explore the properties of these image density functions. We perform sanity check experiments to provide evidence that these probabilities are reasonable. However, we also show that density functions of natural images are difficult to interpret and thus limited in use. We study reasons for this lack of interpretability, and suggest that we can get better interpretability by doing density estimation on latent representations of images.

1 INTRODUCTION

Researchers have long sought to estimate the probability density functions (PDFs) of images. The resulting generative models can be used in image synthesis, outlier detection, image restoration, and in classification. There have been some impressive successes, including building generative models of textures for texture synthesis, and using low-level statistical models for image denoising. However, building accurate densities for full, complex images remains challenging.

Recently there has been a flurry of activity in building deep generative models of complex images, including the use of generative adversarial networks (GANs) (Goodfellow et al., 2014) to generate stunningly realistic complex images. While some deep models, like VAEs, focus explicitly on building probability densities of images, we focus on GANs, leveraging their rapid improvements. Implicitly, these GANs also encode probability densities. In this paper we explore whether these implicit densities capture the intuition of a probable image. We show that in some sense the answer is "no". But, we suggest that by computing PDFs over latent representations of images, we can do better.

We first propose some methods for extracting probability densities from GANs. It is well known that when a bijective function maps one density to another, the relationship between the two densities can be understood using the determinant of the Jacobian of the function. GANs are not bijective, and map a low-dimensional latent space to a high-dimensional image space. In this case, we modify the standard formula so that we can extract the probability density value of an image given its latent representation. This allows us to compute densities of images generated by the GAN, which we then use to train a regressor that computes densities of arbitrary images.

We perform sanity checks to ensure that GANs do indeed produce reasonable densities on images. We show that GANs produce similar densities for training images and for held out test images from the same distribution. We also show that when we compute the density of either real or generated images, the most likely (highest density value) images are of low complexity, and the least likely images are of high complexity. An example of this last result is shown in Figure 1, which displays the images with highest and lowest densities among samples generated by a StackGAN (Zhang et al., 2017) and a StyleGAN (Karras et al., 2018). The StackGAN images are conditioned on two different captions, and the StyleGAN images are from models trained on two different datasets.

Unfortunately, we also show that probability densities learned on images are difficult to interpret and have unintuitive behaviors. The strong influence of visual complexity on the learned PDF causes irrelevant background details to dominate the shape of the distribution; we see that the most likely images tend to contain small objects with large, simple backgrounds, while images with complex backgrounds are deemed unlikely despite being otherwise sensible. For example, for a GAN trained on MNIST, all of the most likely digits are 1, despite each type of digit occurring in equal proportion in the training set. If we exclude 1s from the training data and then compute the densities of all MNIST digits under this altered distribution, the most likely digits are still 1s, even though the GAN never saw them during training. In fact, even if we train a GAN on CIFAR images of real objects, the GAN will produce higher densities for MNIST images of 1s than for most of the CIFAR images. Theoretically, this is not surprising: high-dimensional density functions tend to have peaks of very large probability density away from "typical" points. Consider the example of a high-dimensional Gaussian with an identity covariance matrix, which has large density values at its center, though most sampled points lie near the unit sphere. In practice, this becomes a problem when real images inhabit these high-density peaks, because . We investigate these unintuitive properties of density functions in detail, and explore reasons for this lack of interpretability.

We propose to mitigate this problem by doing probability density estimation on the latent representations of the images, rather than their pixel representations. With this approach we obtain probability distributions with inliers and outliers that seem to coincide more closely with our intuition. In the Gaussian latent space, the problem of natural images lying near high-density peaks is mitigated: natural images correspond to latent codes near the unit sphere, putting them on more equal footing with one another. Outliers can then be detected by finding images with density values that are lower or higher than expected.

In parallel to our work, Nalisnick et al. (2018) also addresses the interpretability of density functions over images, claiming that seemingly uninterpretable density estimates result from inaccurate estimation on out-of-sample images (Nalisnick et al., 2018). Our thesis is different, as we argue that density estimation is often accurate even for unusual images, but the true underlying density function (even if known exactly) is fundamentally difficult to interpret.

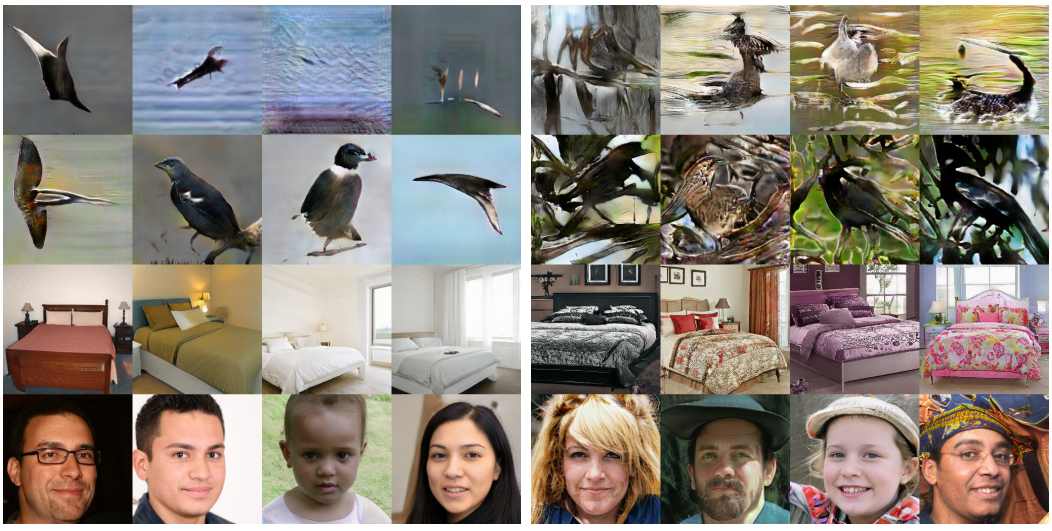


Figure 1: The images with the highest and lowest densities among 100 samples from a StackGAN (top two rows) and a StyleGAN (bottom two rows). **Left:** Images with highest density. **Right:** Images with lowest density. **Top row:** Samples from StackGAN trained on the CUB-200 dataset (Welinder et al., 2010), conditioned on the caption "A bird with a very long wing span and a long pointed beak." **Second row:** Samples from StackGAN conditioned on the caption "This bird has a white eye with a red round shaped beak." **Third row:** Samples from a StyleGAN model pretrained on the LSUN Bedroom dataset (Yu et al., 2015). **Bottom row:** Samples from a StyleGAN model pretrained on the Flickr-Faces-HQ dataset (Karras et al., 2018).

2 BACKGROUND

There are many classical models for density estimation in low-dimensional spaces. Non-parametric methods such as Kernel density estimation (i.e., Parzan windows (Parzen, 1962; Heidenreich et al., 2013)) can model simple distributions with light tails, and nearest-neighbor classifiers (eg., Boiman et al. (2008)) implicitly use this representation. Directed graphical models (eg., Chow-Liu trees and related models (Chow & Liu, 1968)) have also been used for classification (Mattar & Learned-Miller). However, these models do not scale up to the complexity or dimensionality of image distributions.

There is a long history of approximating the PDFs of images using simple statistical models. These approaches succeed at estimating some low-dimensional marginal distribution of the true image density. Modeling the complete, high-dimensional distribution of complex images is a substantially more difficult problem. For example, Olshausen & Field (1996) models the low-level statistics of natural images. Portilla et al. (2003) uses conditional models on the wavelet coefficients of images and shows that these models can improve image denoising. Roth & Black (2005) learns and applies image priors based on Fields of Experts. Markov models have also been used to synthesize textures with impressive realism (De Bonet & Viola, 1998; Efros & Leung, 1999).

Neural networks have been used to build generative models of images. Park (2016) and Timofte et al. (2012) do so assuming independence of pixels or patches. Restricted Boltzmann Machines Smolensky (1986) and Deep Boltzmann machines (Salakhutdinov & Larochelle, 2010) also model image densities. However these methods suffer from complex training and sampling procedures due to mean field inference and expensive Markov Chain Monte Carlo methods (Salimans et al., 2015). In another approach, Variational Autoencoders (Kingma & Welling, 2013) simultaneously learn a generative model and an approximate inference, and offer a powerful approach to modeling image densities. However, they tend to produce blurry samples and are limited in application to low-dimensional deep representations.

Recently, GANs (Goodfellow et al., 2014) have presented a powerful new way of building generative models of images with remarkably realistic results (Brock et al., 2018). Generative adversarial networks are neural network models trained adversarially to learn a data distribution. They consist of a generator $G_\theta : \mathbb{R}^n \to \mathbb{R}^m$ and a discriminator $D_\phi : \mathbb{R}^m \to \mathbb{R}$, where n is the dimension of a latent space with probability distribution P_z and m is the dimension of the data distribution P_d, which is equal to width $\times$ height $\times$ #colors in the case of images. In the original GAN, the discriminator produces a probability estimate as output, and the GAN is trained to reach a saddle point via the learning objective

$$\min_\theta \max_\phi \mathbb{E}_{x\sim P_d}[\log D_\phi(x)] + \mathbb{E}_{z\sim P_z}[\log(1 - D_\phi(G_\theta(z))], \tag{1}$$

which incentivizes the generator to produce samples that the discriminator classifies as likely to be real, and the discriminator to assign high probability values to real points and low values to fake points. Unfortunately, GANs don't produce explicit density models – the GAN is capable of sampling the density, but not evaluating the density function directly.

A major limitation of GANs is that they are not invertible. So, given an image, one does not have access to its latent representation, which could be used to calculate the image's density value. To overcome this problem, Real Non-Volume-Preserving transformations (Real NVP) (Dinh et al., 2016) learn an invertible transformation from the latent space to images. This yields an explicit probability distribution in which exact density values can be computed. Real NVP can be trained using either maximum likelihood methods or adversarial methods, or a combination of both, as in FlowGAN (Grover et al., 2017). Both of these models have proven effective at generating high-quality images. (See also: Dinh et al. (2014), Papamakarios et al. (2017)).

In this paper, we choose to focus on the use of non-invertible GANs to estimate image density. An alternative approach would be to use VAEs, but we feel that the widespread use of GANs makes them an interesting target for study. One issue with invertible GANs is that the latent space must be of the same dimension as the image space, which becomes problematic for large, high-dimensional images. Also, non-invertible GANs currently produce higher quality images than invertible GANs like FlowGAN, suggesting that they might implicitly represent the most accurate probability distributions. Furthermore, non-invertible GANs use simpler network architectures and training procedures than invertible GANs. The standard DCGAN (Radford et al., 2015), for example, consists of basic

convolutional layers with batch norm and ReLU transformations. By contrast, Real NVP requires a scheme of coupling, masking, reshaping, and factoring over variables. Our proposed methods can be applied to any GAN, so that they can leverage any improvements made in new GAN architectures.

Extracting density estimates from GANs presents several challenges. A (non-invertible) GAN learns an embedding of a lower-dimensional latent space (the random codes) into a much higher dimensional space (the space of all possible images of a certain size). Thus, the probability distribution that it learns is restricted to a low-dimensional manifold within the higher-dimensional space. Exact densities for images can be computed via the Jacobian if the latent code is known, as we will show in the next section, but densities are technically zero for images that are not exactly generated by any latent code. Extending densities meaningfully beyond the data manifold requires either incorporating an explicit noise model, such as in the recent Entropic GAN (Balaji et al., 2018), or learning a projection from images to latent codes, such as in BiGAN (Donahue et al., 2016).

In this paper, we avoid these complexities by creating a simple regressor network that accepts an image and returns its estimated probability density. Training such a regressor network is easy if one has a large dataset of images labeled with their probability densities. In section 3 we describe a simple method for obtaining such a dataset.

3 Extracting probability densities

A GAN generator G takes a random variable Z with a known latent distribution P_z and produces an image $G(Z)$ from an implicit learned distribution P_d. But what is P_d? If G is differentiable and bijective, then for $x = G(z)$ the change of variables formula (Munkres, 2018) yields $P_d(x) = P_z(G^{-1}(x))|\det \partial G^{-1}(x)|$, where $\partial G^{-1}(x)$ is the Jacobian of the inverse function at x.

But most GAN generators are not bijective; they map a low-dimensional latent space to a high-dimensional pixel space, so the Jacobian is not square and we cannot compute a determinant. The solution is to perform calculations not on the codomain, but on the low-dimensional manifold consisting of the image of the latent space under G. If G is differentiable and injective, then this manifold has the same intrinsic dimensionality as the latent space, and we can consider how a unit cube in the n-dimensional latent space distorts as it maps onto the (also n-dimensional) image manifold. The resulting modified formula is

$$P_d(x) = P_z(z)|\det \partial G^{\mathrm{T}}(z)\partial G(z)|^{-\frac{1}{2}}. \tag{2}$$

This formula uses the fact that $\det(M^{-1}) = (\det M)^{-1}$ for any square matrix M. It also uses the fact that the squared volume of a parallelpiped in a linear subspace is computed by projecting to subspace coordinates via the transpose of the coordinate matrix, resulting in the square matrix $\partial G^{\mathrm{T}}\partial G$ (an expression which is known as a metric tensor), and then taking the determinant.

The Jacobian ∂G can be computed analytically from the network computation graph, or numerically via a finite difference approximation (we found that the latter approach was much faster and did not change the qualitative results). Once computed, we can find the above determinant via a QR decomposition. If $\partial G = Q \cdot R$, where Q is an $m \times n$ matrix with orthonormal columns and R is an $n \times n$ upper-triangular matrix, then $\det \partial G^{\mathrm{T}}\partial G = \det R^{\mathrm{T}}Q^{\mathrm{T}}QR = \det(R)^2 = (\prod_{i=1}^{n} r_{ii})^2$. Substituting back into equation (2), we obtain the probability formula $P_d(x) = P_z(z)\prod_{i=1}^{n}|r_{ii}|^{-1}$. In practice, we use the log-densities to avoid numerical over/underflow.

To generalize probability predictions to novel images, we train a separate regressor network on samples from G, which are labeled with their log-probability densities. This regressor predicts densities directly from images. We will refer to this as the pixel regressor. This regressor does not truly learn a probability distribution, but is a reasonably accurate proxy.

Our basic generative model was a DCGAN (Radford et al., 2015), and the structure of our pixel regressor was modified from the discriminator. We describe the models and experimental methods in detail in the supplementary material, which will be included in the (currently non-anonymous) Github repository for this project.

Figure 2: From left to right: random samples from a GAN trained on MNIST, samples of lowest probability density according to the pixel regressor, random samples from a GAN trained on CIFAR, samples of lowest density according to the pixel regressor.

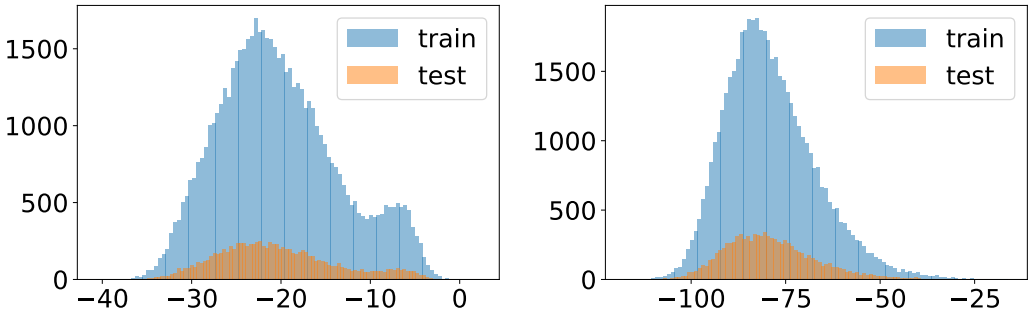


Figure 3: **Left:** histogram of log probability densities of MNIST train and test data as predicted by a pixel regressor for an MNIST GAN. **Right:** histogram of log densities of CIFAR train and test data as predicted by a pixel regressor for a CIFAR GAN.

4 Sanity check: do GANs yield reasonable probability estimates?

The accuracy of GAN-based density estimation depends on the accuracy of the generated density labels, and the ability of the regression network to generalize to unseen data. In this section, we investigate whether the obtained probability densities are meaningful. We do this quantitatively by comparing histograms of predicted densities in the train and test datasets, and also qualitatively by examining how probability density correlates with image quality.

4.1 Comparing histograms

The GAN and regressor model can be inaccurate because of under-fitting (e.g., missing modes), or overfitting (assigning excessively high density to individual images). We test for these problems by plotting histograms for the probability densities on both the train and test data to validate that these distributions have high levels of similarity.

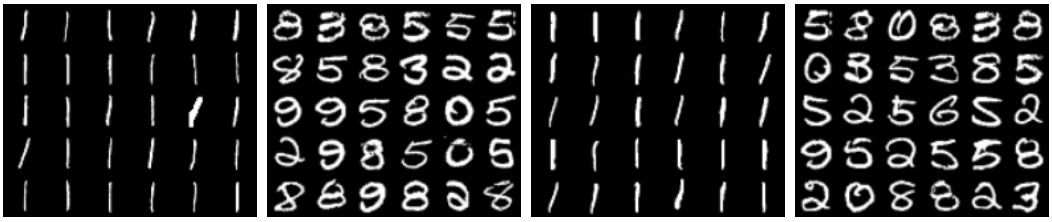
Results are shown in Figure 3. The test histograms appear as a scaled-down version of the train histograms because the test sets contain fewer samples (we did not normalize the histograms by number of samples because this difference in scale helps in seeing both distributions on the same figure). For both MNIST and CIFAR, we see a very high degree of similarity between test and train distributions, indicating a good model fit (without over-fitting).

4.2 Visualizing typical and low density images

We get a stronger sense for what the density estimator is doing by visualizing "outliers" that have low probability density. Figure 2 shows typical samples produced by the GAN models for MNIST and CIFAR. We see that the GANs fit the distributions nicely, as typical samples reflect what we want these images to look like. However, the lowest density outliers (selected from 50,000 GAN random samples) are extremely irregular and clearly lie away from the modes of the distribution. When we use more sophisticated GANs, such as StackGAN or the recent StyleGAN (Figure 1), the low density images always contain more complex textures and varied features, while the high density images are very uniform (as we will discuss further in the next section).

These visualizations suggest that GAN-based density estimators make reasonable density predictions. However, we will see in the next section that even highly accurate density estimation can have unreasonable consequences for some tasks.

5 BE CAREFUL WHAT YOU WISH FOR: THE DIFFICULTIES OF INTERPRETING IMAGE DENSITIES



Most likely with 1s | Least likely with 1s | Most likely without 1s | Least likely without 1s

Figure 4: Highest and lowest density real MNIST digits as predicted by a pixel regressor for a GAN trained on MNIST with and without 1s, and tested on MNIST with 1s.

Although in some respects the learned probability densities correlate sensibly with image complexity and quality, we now show that these distributions are also highly irregular and non-uniform – a characteristic that makes them difficult to interpret. In particular, the densities do not correlate well with human intuitions about semantic categories, such as object class or digit type.

5.1 EVERYBODY LOVES 1S

We saw in Section 4.1 that image densities could be used to discern certain kinds of visual outliers from an image distribution. But what about the inliers? In this section we dive further into what image characteristics most strongly determine image density.

The left two images of Figure 4 show the most likely and least likely real images from the MNIST dataset. We see that all of the most likely images are 1s, while all of the least likely images are more "loopy" digits. This may seem unintuitive at first: 1's are just as likely to occur as any other digits, so why should they have higher probability density? However, this preference for 1s is in fact the

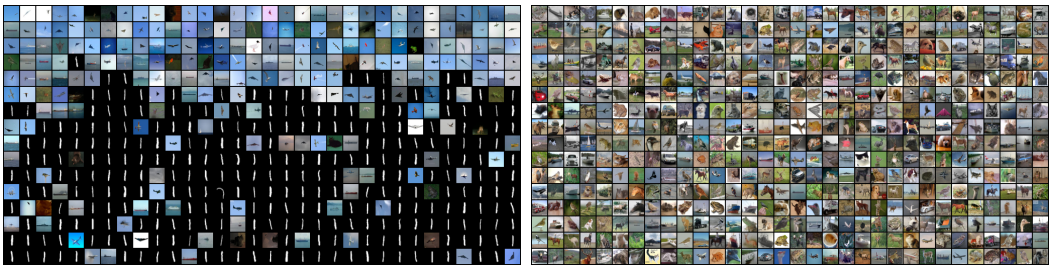
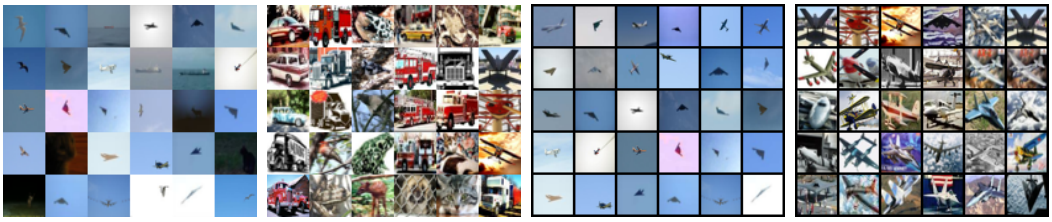


Figure 5: **Left:** Highest density 512 images from CIFAR and MNIST combined, as predicted by a pixel regressor trained on CIFAR. **Right:** Highest density 512 images for the combined data as predicted by a code regressor for a GAN trained on CIFAR.



Top CIFAR | Bottom CIFAR | Top Airplanes | Bottom Airplanes

Figure 6: From left to right: Highest density real CIFAR images according to a pixel regressor trained on CIFAR, lowest density CIFAR images for the same distribution, highest density real airplanes, lowest density airplanes.

result of correct density estimation; images of 1s in the MNIST dataset are all very well aligned and similar, and when interpreted as vectors in a high-dimensional space they cluster closely together. As a result, the 1s define an extremely high-density mode in the image distribution. This can be seen prominently in Figure 3, where we found that the bump on the right of the MNIST histogram was almost exclusively comprised of 1s.

To better assess the severity of this problem, we train the density estimator on all images except 1s. Intuitively, the 1s should now be outliers from the distribution. However, the density function still thinks the opposite. When the density of this incomplete distribution is evaluated on all MNIST test data (including the 1s), the most likely digits are still 1s (Figure 4, second image from the right).

This effect is likely because of the constant black background in images of 1s. Most pixels in these images are black (the most common value), and so these images lie relatively close (in the Euclidean sense) to many other MNIST images, making them inliers rather than outliers.

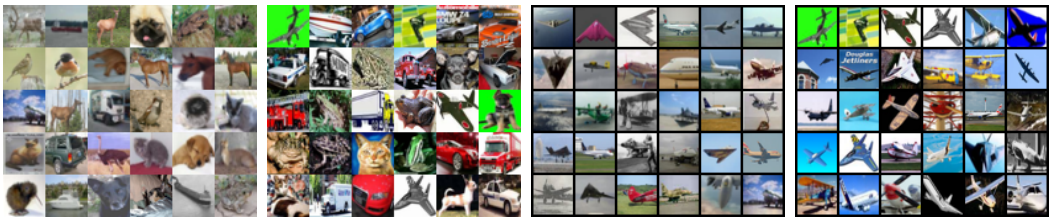
A similar problem manifests in the CIFAR dataset (Figure 6). In this case, the most likely images contain a simple blue background. This is likely because the "airplane" class contains many images with a smooth background of a similar blue color, and so these images lie close together in Euclidean distance, defining a high-density mode. Furthermore, in images of high density, the actual object of interest is extremely small and the background is dominant. Images with large foreground objects or complex backgrounds contain high-frequency features that do not correlate as well, so they lie far apart in Euclidean space and have relatively low densities as in Figure 6.

5.2 Are CIFAR images outliers in their own distribution?

Intuitively, one might expect to use density estimates for outlier detection; outliers from other, highly distinct distributions should have extremely low densities compared to inliers. We saw in Section 4.2 that densities were able to detect irregular/outlier images sampled from the learned distribution.

We study whether MNIST images are outliers from the CIFAR distribution. To this end, we train a density model on only CIFAR, and evaluate the density function on both CIFAR and MNIST images. The most likely images from the combined CIFAR/MNIST dataset are shown in the left image of Figure 5. The set of most likely images is dominated by MNIST digits, with a small number of extremely simple CIFAR images in the top as well. Histograms of these densities are depicted in the leftmost image of Figure 8, and we see that MNIST is indeed far more likely than CIFAR.

This result is consistent with the experiments above – smooth, geometrically structured images lie far to the right of the distribution. The MNIST images apparently lie in an extremely high density mode. However, in the CIFAR distribution, highly structured images of this type seldom appear. This indicates that the high density region occupies an extremely small volume and thus very small probability mass. Meanwhile, the lower-density outlying region (which contains the vast majority of the CIFAR images) comprises nearly all the probability mass.



Top CIFAR | Bottom CIFAR | Top Airplanes | Bottom Airplanes

Figure 7: Highest and lowest density real CIFAR images, as predicted by a latent code regressor trained on CIFAR, and highest and lowest density CIFAR airplanes as predicted by a latent code regressor.

6 Making density functions interpretable

The experiments in Section 5 indicate that probability densities on complex image datasets have a structure that corresponds more to certain geometric properties of images than human-recognizable categories. Fairly "typical" images often lie far from the modes of the distribution, their probability mass spread thinly. This lack of interpretability is a consequence of a well known problem; the Euclidean distance between images does not capture an intuitive or semantically meaningful concept

of similarity. "Outliers" of a distribution are points that lie far from the modes in a Euclidean sense, and so we should expect the relationship between density and semantic structure to be tenuous.

To make density estimates interpretable, we need to embed images into a space where Euclidean distance has semantic meaning. We do this by embedding images into a deep feature space. In deep feature space, nearby images have similar semantic structure, and well separated images are semantically different. This enables distributions to have interpretable modes and outliers.

There are many options to choose from when selecting a deep embedding. In the unsupervised setting where we already have a GAN at our disposal, the simplest choice for a feature embedding is to associate images with their latent representation z, the pre-image of the GAN. This embedding is particularly simple because the density function in this space is simply the Gaussian density, which can be evaluated in closed form. We learn this density mapping by associated each image with the density of its pre-image z, without accounting for the Jacobian of the mapping (using a GAN trained with a slightly modified, InfoGAN-based loss (Chen et al., 2016) as described in the supplementary material, to impose more structure on the latent space). For this approach, we train a regressor to predict the density of the z code that generated an image; we refer to this as the "code regressor."

6.1 Images are now inliers in their own distribution

We show the most and least likely CIFAR images under the deep feature model in Figure 7. Unlike the pixel-space model, there is now diversity in the most likely images, and the distribution is not dominated by blue sky. The deep model also produces much more uniform densities than the pixel model, as shown in the rightmost plot of Figure 8, where 1s no longer completely dominate the far right – this is expected since the MNIST dataset is itself fairly uniform with few semantic outliers. We saw in Section 5.2 that MNIST digits were inliers with respect to the CIFAR distribution, and many CIFAR images were outliers in their own distribution (when estimating densities in the pixel space). Density in the deep feature space captures a more intuitive notion of outliers. To show this, we train a deep feature density estimator on the CIFAR distribution only, and then infer densities on the combined CIFAR and MNIST dataset. The middle plot of Figure 8 shows the histogram of estimated densities. We see that CIFAR images now occupy high density regions close to the distribution modes, and MNIST images occupy the low density "outlier" regions. This is visually depicted in the right image of Figure 5, which shows the most probable CIFAR and MNIST images with respect to the CIFAR distribution. Unlike the left image in the same figure, we now see that all of the most likely images are from the CIFAR distribution. Recall our discussion from the introduction about high-dimensional density functions. In the latent space, images from the learned distribution are more clustered around the unit sphere; this explains why the rightmost histogram in Figure 8 is more clustered than the pixel-space histogram on the left (although there is still a small tendency for ones to have higher density value). It also explains why the images on the right of Figure 5 are more typical CIFAR images, as opposed to images of uniform background.

6.2 Deep densities depend on image content rather than smoothness

Unlike the pixel-space density estimator depicted in Figure 6, the deep feature model (Figure 7) favors images where the foreground object is well-defined and occupies a large fraction of the image. The least likely images contain many objects in unusual configurations or strange backgrounds (e.g., airplanes with a green sky). For the category of airplanes shown in the two rightmost images of Figure 7, the densities seem to no longer depend strongly on the image complexity, but rather on the image content and color.

7 Conclusion

Using the power of GANs, we explored the density functions of complex image distributions. Unfortunately, inliers and outliers of these density functions cannot be readily interpreted as typical and atypical images, at least according to human intuition. However, we suggest that this lack of interpretability could be mitigated by considering the probability densities not of the images themselves, but of the latent codes that produced them. We postulate that such feature embeddings avoid the problems of pixel-space densities (which are too dependent on pixel-level image properties such as background uniformity), and instead allow for representations that are more semantically meaningful. There are a host of potential applications for the resulting image PDFs, including detecting outliers and domain shift that will be explored in future work.

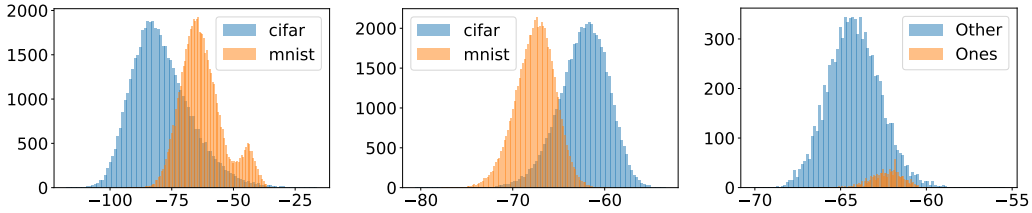


Figure 8: **Left:** histogram of log probability densities of MNIST and CIFAR, predicted using a pixel-space density estimator for CIFAR. **Middle:** histogram of log densities of MNIST and CIFAR, predicted using the latent code regressor for a GAN trained on CIFAR. **Right:** histogram of log densities of MNIST, as predicted by a latent code regressor for a GAN trained on MNIST. Note that the log density values are much more clustered than in pixel space, though they are still near the top of the distribution.

REFERENCES

Yogesh Balaji, Hamed Hassani, Rama Chellappa, and Soheil Feizi. Entropic gans meet vaes: A statistical approach to compute sample likelihoods in gans. arXiv preprint arXiv:1810.04147, 2018. 4

Oren Boiman, Eli Shechtman, and Michal Irani. In defense of nearest-neighbor based image classification. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE, 2008. 3

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096, 2018. 3

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in neural information processing systems, pp. 2172–2180, 2016. 8

C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. IEEE transactions on Information Theory, 14(3):462–467, 1968. 3

Jeremy S De Bonet and Paul Viola. Texture recognition using a non-parametric multi-scale statistical model. In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pp. 641–647. IEEE, 1998. 3

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516, 2014. 3

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. arXiv preprint arXiv:1605.08803, 2016. 3

Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016. 4

Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In iccv, pp. 1033. IEEE, 1999. 3

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pp. 2672–2680, 2014. 1, 3

Aditya Grover, Manik Dhar, and Stefano Ermon. Flow-gan: Bridging implicit and prescribed learning in generative models. arXiv preprint, 2017. 3

Nils-Bastian Heidenreich, Anja Schindler, and Stefan Sperlich. Bandwidth selection for kernel density estimation: a review of fully automatic selectors. AStA Advances in Statistical Analysis, 97(4):403–433, 2013. 3

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. arXiv preprint arXiv:1812.04948, 2018. 1, 2

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013. 3

Marwan A Mattar and Erik G Learned-Miller. Improved generative models for continuous image features through tree-structured non-parametric distributions. 3

James R Munkres. Analysis on manifolds. CRC Press, 2018. 4

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. 2

Bruno A Olshausen and David J Field. Natural image statistics and efficient coding. Network: computation in neural systems, 7(2):333–339, 1996. 3

George Papamakarios, Iain Murray, and Theo Pavlakou. Masked autoregressive flow for density estimation. In Advances in Neural Information Processing Systems, pp. 2338–2347, 2017. 3

Dong-Chul Park. Image classification using naïve bayes classifier. Int. J. Comp. Sci. Electron. Eng.(IJCSEE), 4, 2016. 3

Emanuel Parzen. On estimation of a probability density function and mode. The annals of mathematical statistics, 33(3):1065–1076, 1962. 3

Javier Portilla, Vasily Strela, Martin J Wainwright, and Eero P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image processing, 12(11):1338–1351, 2003. 3

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434, 2015. 3, 4

Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 2, pp. 860–867. IEEE, 2005. 3

Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp. 693–700, 2010. 3

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In International Conference on Machine Learning, pp. 1218–1226, 2015. 3

Paul Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986. 3

Radu Timofte, Tinne Tuytelaars, and Luc Van Gool. Naive bayes image classification: beyond nearest neighbors. In Asian Conference on Computer Vision, pp. 689–703. Springer, 2012. 3

P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 2

Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. LSUN: construction of a large-scale image dataset using deep learning with humans in the loop. CoRR, abs/1506.03365, 2015. URL http://arxiv.org/abs/1506.03365. 2

Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. arXiv preprint, 2017. 1