Can We Train Dogs and Humans at the Same Time?
GANs and Hidden Distributions

Bryan Cheong, Hubert Teo
Stanford University
bcheong@stanford.edu, hteo@stanford.edu

Abstract

This paper is a series of experiments on InfoGAN, exploring the extent to which an InfoGAN is able to generate the hidden distribution behind a sparse initial dataset. We observe from our experiments that the infoGAN would prefer to memorise a sparse dataset if it is not sufficiently complex, and even when sufficiently complex will not generalise over unpopulated regions of the data distribution. As a result, the InfoGAN does not extrapolate over mixed distributions.

1. Introduction

In this paper, we aim to hone in on the limitations of generative models for image generation. In particular, we evaluate the limitations of generative adversarial networks (GANs) on an image extrapolation task.

Our research question concerns the nature of the types of distributions that can be modelled by GANs. Recently, GANs have seen remarkable empirical success in their application to unsupervised learning, generating sharp and realistic images after training only on unlabelled data [18]. They accomplish this by modelling the underlying data distribution in a way that allows them to be sampled from. However, there is still much to be understood about the kinds of distributions that are modelled well by GANs, as well as their generalizing power. In the context of images, what classes of transformations can be captured by GANs? Can GAN models extrapolate between mixtures of distributions to produce convincing images that interpolate between two datasets? In particular, given a combined dataset of images of dog and human faces, can GANs produce an acceptable morph between dog and human faces? Our experiments are an approach to test the extent to which an unsupervised approach can produce explanatory factors, and extrapolate between these factors, as a complement to supervised representation-learning algorithms [2].

These questions have broad implications on the extent to which GAN-based approaches to image generation are possible. Furthermore, we hope they will yield a better understanding of the nature of the distributions generated thereby.

2. Related Work

Within the family of GANs, there have been many different architectures that reportedly have different properties. We explored the different types of GANs and summarise their developments here. The Conditional GAN or CGAN is a variation on the GAN by Mirza et al. which feeds latent variables that condition the data distribution into the GAN as an additional layer in the GAN structure, so that both the G and D distributions are conditioned on the latent variables. Their experiment of the CGAN consisted of generating MNIST numerals conditioned on their class labels and concluded that the CGAN was a viable model that could capture multimodal labelling. [16]

Since our project is explores the limitations of GANs around limited and sparse datasets, we also consulted the literature on DeLiGANs, or Generative Adversarial Networks for Diverse and Limited Data [9]. Gurumurthy et al. implemented an architecture that tried learning a mapping from a simple latent distribution to a more complicated data distribution, in order to train a GAN for when the original dataset is limited but has a diverse and sparse modality by drawing on a reparamaterized mixture of Gaussians instead of over the distribution of the latent variable directly (which is a single Gaussian). This is a so-called reparameterization trick. [5] They experimented on datasets such as MNIST and freeform drawn samples, and demonstrate that they are able to actively avoid the low-probability regions. The DeLiGAN handles a low probability void between two modes of high information in a dataset distribution by absorbing the void into its own latent distribution and produces no samples from this low-probability region. Indeed, much of the literature is concerned about the stability of GANs when presented with non-ideal training data. For example, many slightly different architectures, objective functions or formulations to alleviate GAN training in-stability have been researched, including the Wasserstein GAN (WGAN) [1], unrolled GAN [15], and even ensemble methods [10]. There is also significant literature exploring best GAN training practices that seek to optimize stability and prevent mode collapse [19] [8]. Our experiments in this paper, however, seek not to avoid covering these low-probability regions, but instead explore how a GAN might treat these low-probability regions when forced to do so.

Apart from GANs, other generative approaches to unsupervised or semi-supervised image generation have also been explored. Siddharth et al. introduced the generalised variational autoencoder (VAE) [20] model that is reportedly able to disentangle representations that encode distinct aspects of the data into separate variables. To this end, they used partially-specified graphical model structures to construct a recognisable disentangled space, and demonstrated their model's ability to do so for faces and multi-MNIST. Similarly to GANs, this general framework also admits many architectural variants such as conditional VAEs [21] [13], hard-regularization approaches like lossy VAEs [4], to name a few. Crucial to both VAEs and GANs is the idea of the latent variable that implicitly parametrizes the data distribution and understanding the behaviour of and modifying the underlying distribution of these latent variables is an area of heavy research [22] [24]. We wish to explore with our experiments if a more unsupervised InfoGAN approach can likewise encode a disentangled space within its latent variables, and not only disentangle but also populate the full space between the modes of the dataset distribution.

Since our experiments is essentially an exploration on the 'creativity' of GANs in their ability to construct distributions, we also take caution that our GAN implementation will not memorise the original dataset distribution, which would defeat the purpose of our experiments. We consulted the literature on latent geometry and memorisation in GANs by Matt D. Feiszli. [6] We have kept in mind his paper's understanding of how a GAN can learn an output distribution that is concentrated on a finite number of examples, and have chosen our methods and visualisations to be conscientious of catching this memorisation if it indeed does occur.

3. Methods

In this paper, we perform three experiments that share a similar structure. Each of the three experiments begin with a dataset composed of a mix of two sets of images. Then, we train GANs on the dataset and use the trained model to perform various tasks. We keep the GAN model architecture constant (save for hyperparameters and image channels) in order to evaluate its performance across the three datasets.

3.1. Generative Adversarial Networks (GANs)

A generative adversarial network is an unsupervised learning technique that estimates a joint distribution between latent variables z and data x via an adversarial process. Generative models learn distributions not by attempting to estimate the probability that a particular data point is drawn from a distribution, but instead by modelling the distributions themselves. GANs achieve this goal by training a generator and a discriminator adversarially. The generator models $p(x|z)$, approximating the distribution of data given incompressible random variables z as input. The discriminator's goal is instead to distinguish between data points sampled from the generator's distribution and real data points from the data set. Goodfellow [7] defined the following minimax game that optimizes the generator and discriminator in alternating phases:

$$\min_G \max_D V(G,D) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

3.2. InfoGAN

An InfoGAN is a GAN variant that makes latent variables c explicitly different from noise variables z [3]. The generator G is allowed to parametrize the data distribution $p(x|z,c)$ with both incompressible noise z and meaningful latent variables c. Another network Q is introduced to approximate the posterior distribution $p(c|x)$. The objective function is also modified to include a mutual information term $\mathcal{L}_I(G,Q)$ which gives a lower bound on the true mutual information between the data distribution and the latent variables:

$$\mathcal{L}_I(G,Q) = \mathbb{E}_{c\sim p_c(c), x\sim G(c,z)}[\log Q(c|x)]$$

This mutual information lower bound is then included in the minimax objective so that it is maximized by the G:

$$\min_{G,Q} \max_D V(G,D) - \lambda \mathcal{L}_I(G,Q)$$

In practice, D and Q are made to share the same network up to the last embedding layer before branching off. Hence, we treat λ as a regularizing term that encourages the generator to use the latent variables c meaningfully, such that the auxiliary network Q is able to infer c from its output.

Most relevantly to our research question, Xi Chen et. al. concluded that GANs can approximately model the latent variable distribution of a given x drawn from the dataset. In other words, they produce an embedding of the dataset into a meaningful latent variable space, from which images can be generated. Our research is a way to test the generalizing power of this latent variable space by attempting to generate images from outside the support of latent variable distribution associated with the dataset.

3.3. Architecture

The GAN architecture we use is taken directly from the seminal InfoGAN paper [3], and is essentially a deep con-

G	D	Q
deconv-512-4-1-0	conv-64-4-1-2	
deconv-256-4-2-1	conv-128-4-1-2	
deconv-128-4-2-1	conv-256-4-1-2	
deconv-64-4-2-1	conv-512-4-1-2	
conv-64-3-1-1	conv-σ-1-4-1-0	conv-C-4-1-0
conv-tanh-3-4-2-1		

Table 1. GAN architecture

volutional GAN (DCGAN) [18] with an additional mutual information objective.

Our implementation is based off the PyTorch DCGAN example [17], but with a slightly modified architecture to produce smaller images for speed. (Table 1.) deconv-L-K-S-P denotes a deconvolutional layer producing L features with kernel size K, stride S and padding P, followed by batch normalization and LeakyReLU(0.2). conv-L-K-S-P describes an analogous convolutional layer. The first layer of G is fed the both the noise and latent variables. Note that G's output volume and D's input volume are both $3 \times 64 \times 64$. The final layers of both G and D skip normalization and use their respective non-linearities. The final layer of Q instead predicts the C latent variables from the last embedding layer it shares with D. To support greyscale images in experiment 3, the above network architecture is also modified minimally to produce $1 \times 64 \times 64$ images corresponding to only a single luminance channel.

3.4. Datasets

3.4.1 F dataset

The objective of this dataset is to be as clean and simple as possible while still being a pure additive mixture of two (potentially orthogonal, potentially overlapping) distributions. We plan to use this clean and small dataset to estimate a set of suitable hyperparameters and a viable training schedule for our GAN on the larger dataset.

To this end, this dataset is synthetic and consists of black, san-serif capital Fs on a white background, translated (but not rotated) across the 64×64 image. There are 128 data points in total, corresponding to 64 Fs translated at different pixel locations along the horizontal axis but vertically centered, and another 64 Fs translated along a centered vertical axis. Notably, there are no Fs translated along a diagonal offset from the center.

Additionally, since the images are entirely synthetic with latent variables (vertical and horizontal offset) that are completely transparent, we also generated four Fs diagonally offset from the center, corresponding to the four quadrants. These diagonal Fs can be used to probe the trained generative models for their ability to generate them.

3.5. Dogs & Humans

This dataset is again a pure additive mixture of two distributions. The motivation for including a more complex distribution is that the two distributions presumably have a larger variance and spread in latent space. To account for this more diffuse distribution, we hope that the discriminator should penalize outliers less, giving the generator more leeway to produce images that are far away from both distributions.

Hence, we produced a dataset combining dog faces and human faces. These two distributions have similar structure: both have eyes, noses and mouths, and consist of a roughly circular shape on some background. However, there is no overlap (there are no dog faces that are also human faces). The hope is that the GAN will be able to interpolate between the two distributions while preserving the spatial structure of faces, creating a plausible morph between a dog and a human face.

To create this dataset, we extracted 1517 images from the Stanford Dogs Dataset [11] and hand-aligned them so that the dog faces were roughly registered in the center of the images and with a similar scale relative to the image size. These images were mixed among pre-aligned human faces from the CelebA dataset [14]. In a bid to leverage the well-procured human faces from CelebA while not biasing our GAN towards either distribution, we combined these datasets dynamically at training time. Human faces are randomly sampled at each epoch to match the number of dog faces in our dataset, and the two sets of images are combined so that each batch has an equal number of dog and human faces. In this way, this dataset effectively has a size of 3034 and has an equal number of dog and human faces, but the human faces are instantiated from CelebA on demand.

3.6. Experiment 1: Image Extrapolation

In this experiment, we train GANs on the F dataset, varying several critical hyper-parameters. Training is performed with stochastic batch gradient descent (SGD) using the Adam [12] update rule ($\beta_1 = 0.5, \beta_2 = 0.999$). In each batch iteration, Z random normal variables $z_i \sim N(0, 1)$ are sampled and C latent variables $c_i \sim U(-1, 1)$ are sampled as input to the generator, and the discriminator update step is performed before the generator update step. Training proceeds for a total of 1024 epochs.

A total of 27 models were trained, corresponding to a cross product between the following hyperparameter choices: the SGD learning rate ($lr = 5 \times 10^{-4}, 1 \times 10^{-3}, 2 \times 10^{-3}$), mutual information constant or regularization ($\lambda = 1, 10, 100$), and three configurations for the number of noise and latent variables ($(Z, C) = (2, 2), (16, 2), (16, 16)$). These choices were chosen because of the relative simplicity of the F dataset, where the optimal number of latent variables is theoretically only 2. The MSE loss of the resulting hyperparameters can be found in Fig. 1

Following training, we perform the following reverse optimization problem to estimate the optimal noise and latent variables to produce a given target image:

$$\min_{z,c} \|x - G(z,c)\|_2^2$$

We thus find the generator input that minimizes the L2 norm between the generator output and a target image. The target images are the four diagonal Fs described above. For each model, we then calculate the average loss for the diagonal Fs as a way to measure the ability of the model to generalize and produce examples outside of the given data distribution.

3.7. Experiment 2: Image Interpolation

For the next experiment, we trained a GAN using similar hyperparameters on the dogs and human faces dataset, except with 128 noise variables and 128 latent variables due to the more complicated dataset. These values are selected to match the InfoGAN paper, which also uses more than 200 input variables in its GAN architecture for larger datasets.

Next, we sample a set of random line segments that vary entirely in the noise space and interpolate between them. Similarly, we do the same for line segments varying only in latent variable space. By sampling points along these line segments and using the generator network to produce generated images, we obtain interpolated images that characterize the differences between variation in noise space and variation in latent space.

Furthermore, we also obtain the generator inputs that produce each data point in the dataset via the reverse optimization problem shown above, and linearly interpolate between pairs of them. At each linearly-interpolated point in the latent distribution space, we again use the generator network to produce generated images. Since there is no ground truth for a dog-human face morph, we then evaluate the generated images qualitatively.

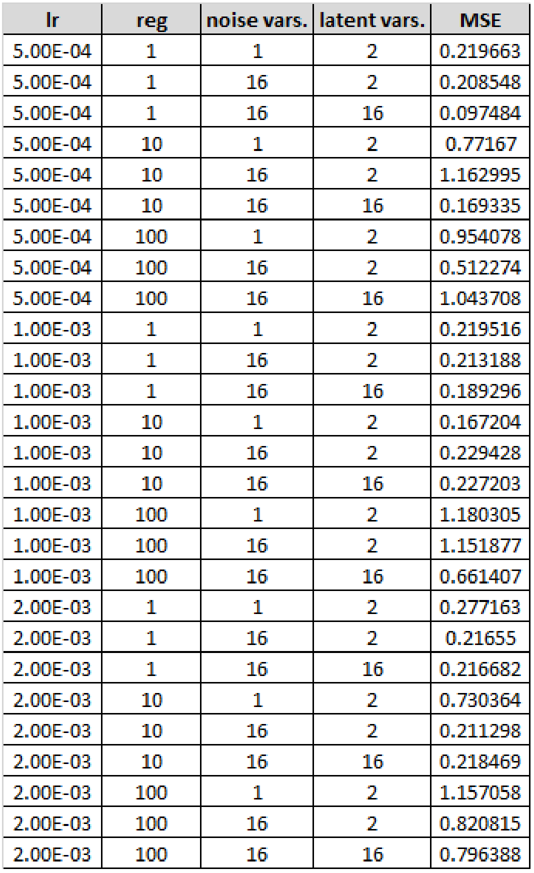
3.8. Experiment 3: Greyscale

Finally, we repeat the training and interpolation process for greyscale versions of both of the above datasets. This is to determine if colour plays a significant role in the ability of GAN to generalize and extrapolate between mixture distributions.

4. Results

4.1. Experiment 1: Image Extrapolation

The images drawn for these results are from the output that is quantitatively assessed to have the lowest MSE loss for its output. The distribution over the hyperparameters tested can be seen in Fig. 1. The generation of the diagonally translated F image exists so far outside the input space bounds of the latent variable that we can observe that the InfoGAN effectively does not extrapolate the two translations into the hidden distribution of the diagonal translation. (Fig. 2). From our samples of the generated output at epoch 1024 after training the GAN (Fig. 3), we observe that the output generally looks like the original dataset of vertically and horizontally translated F images.



lr	reg	noise vars.	latent vars.	MSE
5.00E-04	1	1	2	0.219663
5.00E-04	1	16	2	0.208548
5.00E-04	1	16	16	0.097484
5.00E-04	10	1	2	0.77167
5.00E-04	10	16	2	1.162995
5.00E-04	10	16	16	0.169335
5.00E-04	100	1	2	0.954078
5.00E-04	100	16	2	0.512274
5.00E-04	100	16	16	1.043708
1.00E-03	1	1	2	0.219516
1.00E-03	1	16	2	0.213188
1.00E-03	1	16	16	0.189296
1.00E-03	10	1	2	0.167204
1.00E-03	10	16	2	0.229428
1.00E-03	10	16	16	0.227203
1.00E-03	100	1	2	1.180305
1.00E-03	100	16	2	1.151877
1.00E-03	100	16	16	0.661407
2.00E-03	1	1	2	0.277163
2.00E-03	1	16	2	0.21655
2.00E-03	1	16	16	0.216682
2.00E-03	10	1	2	0.730364
2.00E-03	10	16	2	0.211298
2.00E-03	10	16	16	0.218469
2.00E-03	100	1	2	1.157058
2.00E-03	100	16	2	0.820815
2.00E-03	100	16	16	0.796388

Figure 1. The MSE loss for generating diagonal effs over the hyperparameters of the translated effs experiment.

Keeping in mind that the space of translated Fs is relatively small, we were wary of the InfoGAN memorising the output translations, and thus not producing well-distributed generated images. To observe this, we interpolated between the latent variables of the InfoGAN (Fig. 4) and also over the noise variables (Fig. 5). There is very little movement of the Fs, and there is no difference observable between the interpolation over the latent variables or the noise. This suggests that this dataset has effectively been memorised by the InfoGAN, and so produces output images that are clustered around only a few examples, rather than learning the vertical or horizontal translations. It is not a surprise then that the diagonal F translations were also not produced, since

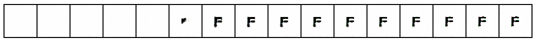


Figure 2. Interpolation between of diagonal Fs. The leftmost and rightmost images are the closest generated images to top-left and bottom-right diagonal Fs. Since they are not in any way off-horizontal or off-vertical, the images are essentially not generated.

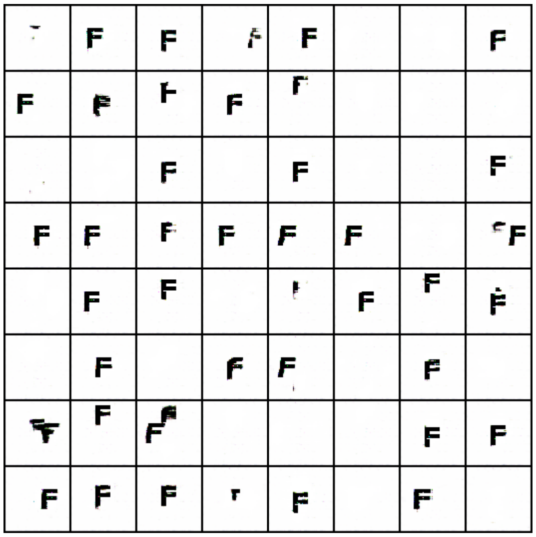


Figure 3. Samples of the generated translations of the letter F at epoch 1024 after training the InfoGAN

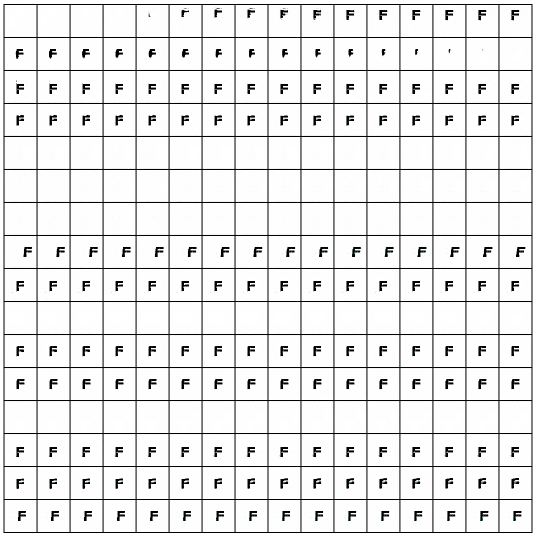


Figure 4. The interpolation over the latent variables of the InfoGAN for translated Fs.

the vertical and horizontal translations were effectively not learned but instead memorized.

4.2. Experiment 2: Image Interpolation

In the case of the facial images of dogs and humans, the InfoGAN in this case was observed not to have memorised the output, producing a good amount of variation in the generated images. We can observe from the output that the faces of both humans and dog are generalized, but unfortunately when we interpolate the images that are generated closer to a dog image and the images that are generated closer to a human face, there is no preservation of the facial features in the mapping. This can be observed with the figure of our batch approximation of the generated output images (Fig. 7). In fact, a lot of the mapping between the images seems to be between the colours of the various images, suggesting that the colour information is dominating the learning of the InfoGAN. This is another motivation for our third experiment, in which we repeat this experiment with only the luminance channel of the same dataset.

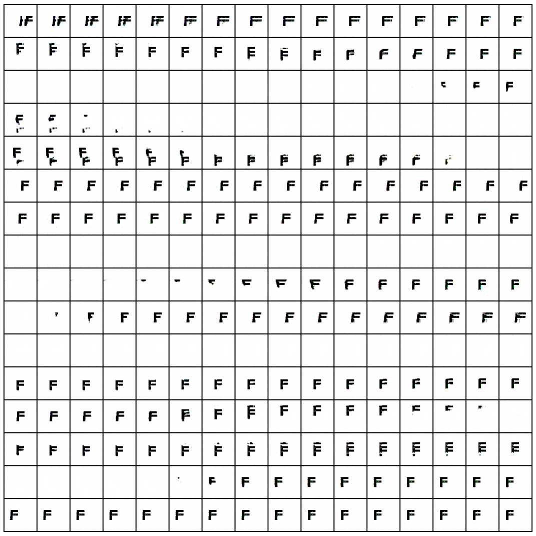


Figure 5. The interpolation over the noise variables of the InfoGAN for translated Fs.

Even though the target images produced via back-solving for dataset samples are largely imperfectly produced, the generation of dog faces and human faces seems quite robust. They are able to produce dog and human faces separately, albeit without a plausible structure-preserving correspondence between these two outputs. This is a somewhat surprising result as the dataset we used to generate these images is very noisy, since it is a mixture of two different distributions. In this case, we can also infer that mode-collapse did not occur even with a clearly bimodal mixture distribution, which is encouraging. This can also be observed in the t-distributed stochastic neighbor embedding (t-SNE) [23] image of our latent variables in Fig. 6.

Another observation that we can make when interpolating between latent variables (Fig. 8) and the noise variables (Fig. 9) is that interpolation between latent variables varies details, while varying between noise variables varies large-scale features such as the overall colour distribution in the image. This suggests that in this case the latent variables are indeed capturing the meaningful variation within the dataset (facial features, geometry, texture) as opposed to the irrel-

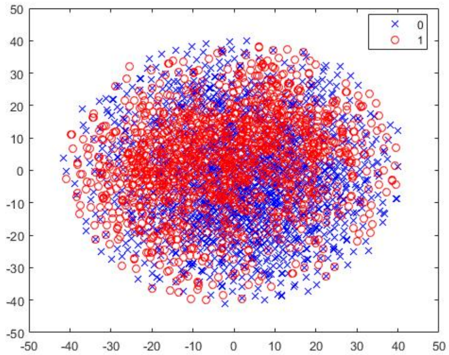


Figure 6. t-SNE visualisation of dogs and humans over latent variables. Dogs are labelled 0 and humans with 1.

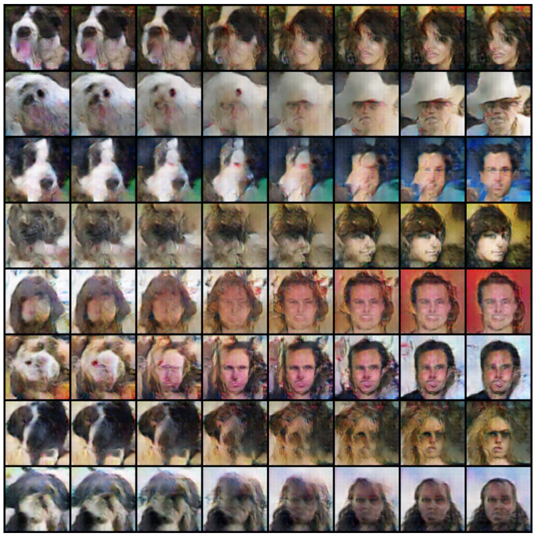


Figure 7. The batch approximation when interpolating between dogs and humans generated by the InfoGAN.

evant details (background colour, shading, and face position). This is also strong evidence that the GAN has indeed learned a meaningful representation of the latent distribution space, and is no longer memorising the distribution as it was in the case of the F dataset.

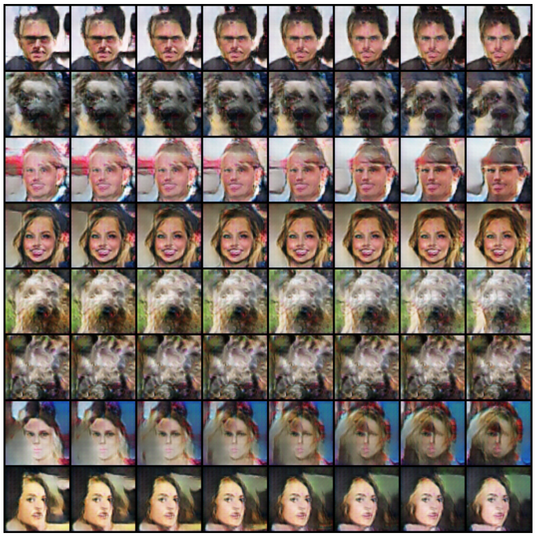


Figure 8. The interpolation over the latent variables of the InfoGAN for dog and human faces.

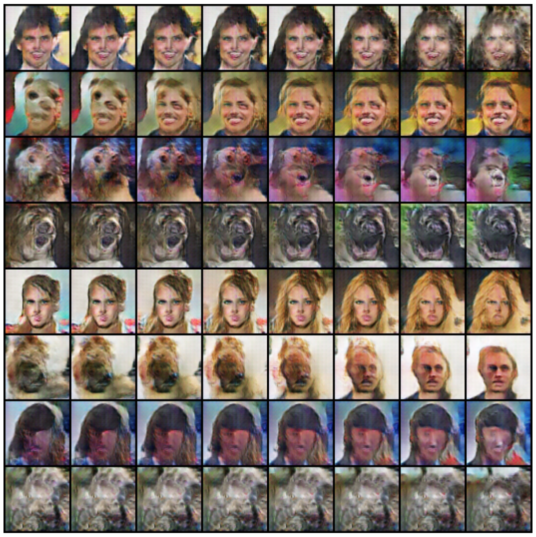


Figure 9. The interpolation over the noise variables of the InfoGAN for dog and human faces.

4.3. Experiment 3: Greyscale

From our batch approximation of the greyscale faces of dogs and humans, this interpolation seems to have a more consistent mapping between the generated output dog and human faces, producing a reasonable transition that largely preserves the features, although there appears to still be some exceptions. (Fig. 11) We can observe then that by restricting the number of channels we did manage to reduce the dominance of colour over other features of the output. This might be the case because the colour features of dogs vary very substantially over the colour variations of humans, and so the dominating feature being translated from dogs to humans is their colour, instead of the visual features that we desired. Even in this case, a t-SNE visualisation of the latent variables shows that the space over which the human faces are distributed is smaller than that of dog faces, suggesting that there is indeed more variation in dog faces even when colour information has been removed and the channels have been restricted. (Fig. 10) However, like the coloured dog and human faces in experiment 2, most of the interpolated images do not seem to display any translation-like movements, and there seems to be a preference for just locally morphing between the initial and final image. This is similar to our experiment with the F dataset, where the

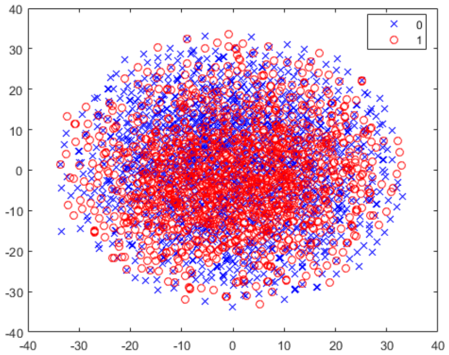


Figure 10. The t-SNE visualisation of greyscale dogs and human face latent variables. The red labels, 1 are human while the blue 0 labels are dogs.

GAN was reluctant to learn to translate an identical image horizontally and vertically, instead choosing to memorize all instances of Fs in each offset position.

On the other hand, in the case of the translated F images, restricting the number of channels does not appear to have a significant impact on preventing the memorisation of images. We quantified the MSE of the generated output over the number of latent variables that we gave the InfoGAN (Fig. 14), and while we observe that the MSE decreases with increasing number of latent parameters, this is likely not because the InfoGAN is learning anything significant from the distribution. The distribution is so simple that such a large number of latent variables is likely not necessary. Indeed when we compare the interpolated output images over the latent variables (Fig. 12) and over the noise variable (Fig. 13) we essentially observe no qualitative difference between these interpolations and the output we observed from our first experiment, which had more channels. We can conclude then that the dataset in this case is so sparse that changing the number of channels does not affect the way that the InfoGAN learns the distribution.

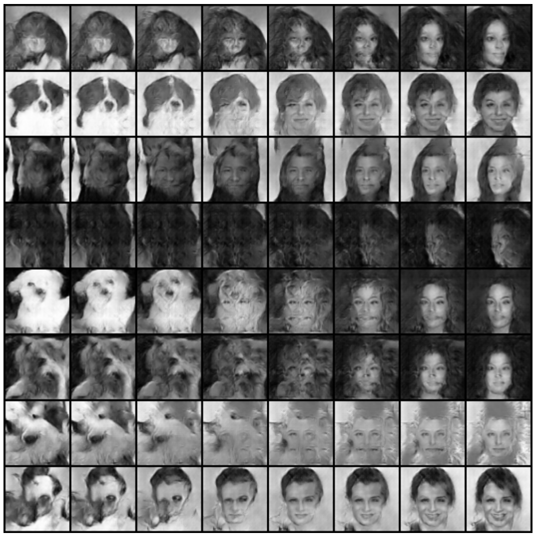


Figure 11. The batch approximation interpolating the output generated by the InfoGAN for greyscale dog and human faces.

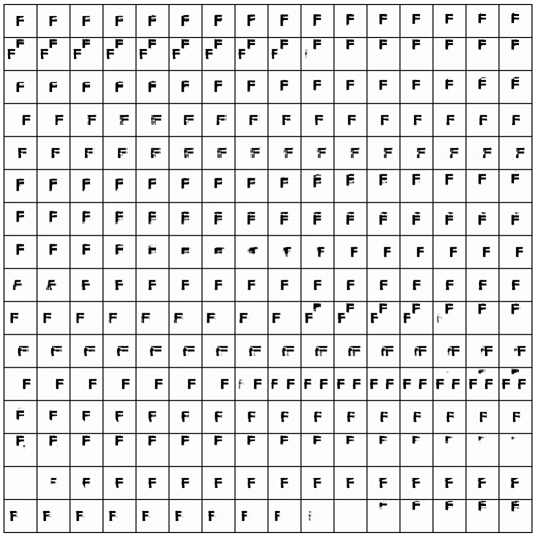


Figure 12. The interpolation over the latent variables of the InfoGAN for greyscale dog and human faces.

5. Conclusion & Future Work

Our conclusion is that while InfoGAN is able to disentangle representations, there are limits to this disentanglement. We see that there is a qualitative difference between the distribution captured by noise variables and latent variables, even when the GAN is trained on a very noisy data distribution. Sparse datasets with biased gathering that do not represent some parts of the population cannot be re-created through InfoGAN. If the dataset is sufficiently sparse, the InfoGAN will instead memorise the output distribution, and not accord any significant learning to the latent variables. In this sense, we see that the noisy distribution is actually beneficial to the InfoGAN's generalizing power, since the widened support and variance reduces the discriminator's ability to penalize the generator for not exactly matching images in the data distribution. There might, however, also be inherent limitations in our current DCGAN-based framework for applying GANs to the image generation task. As we have seen in all three experiments, DCGANs are reluctant to capture macro-scale image transformations like translation.

Further work needs to be done to pursue exactly how creative GANs are, in the human sense. In particular, it would be beneficial to characterize exactly the class of image-space transformations that are easily modelled by GANs and the class of transformations that are not. Additionally, other GAN variants and even other generative models

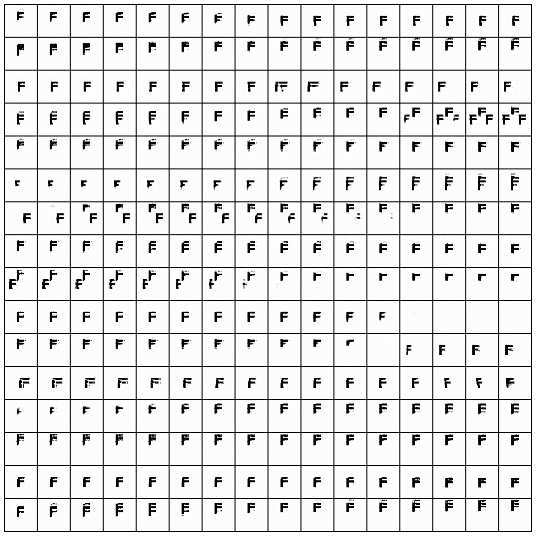


Figure 13. The interpolation over the noise variables of the InfoGAN for greyscale dog and human faces.

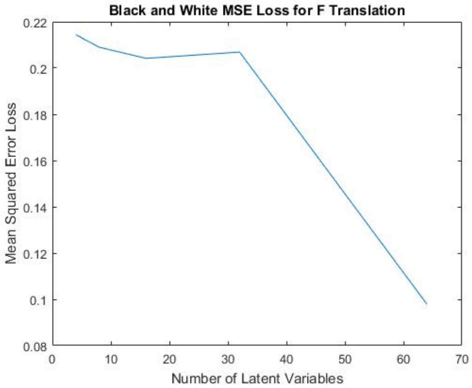


Figure 14. The MSE loss of the greyscale translated F images over the hyperparameter of number of latent variables used.

(based on VAEs or otherwise) should be evaluated according to their ability to interpolate between mixture distributions or extrapolate into regions of low density. This has broader implications for how a GAN may be used to populate datasets that are restricted over certain easier-to-gather modes, or even infer useful information from sparsely sampled datasets. In essence, these are the boundaries of the creative abilities of generative image models, and the quality of the distributions that they learn.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan, 2017. 2

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug 2013. 1

[3] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *CoRR*, abs/1606.03657, 2016. 2

[4] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *CoRR*, abs/1611.02731, 2016. 2

[5] L. Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th Conference on Winter Simulation*, WSC '86, pages 260–265, New York, NY, USA, 1986. ACM. 1

[6] M. Feiszli. Latent geometry and memorization in generative models, 2017. 2

[7] I. J. Goodfellow. NIPS 2016 tutorial: Generative adversarial networks. *CoRR*, abs/1701.00160, 2017. 2

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017. 2

[9] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu. Deligan : Generative adversarial networks for diverse and limited data. In *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition*. 1

[10] F. Juefei-Xu, V. N. Boddeti, and M. Savvides. Gang of gans: Generative adversarial networks with maximum margin ranking. *CoRR*, abs/1704.04865, 2017. 2

[11] A. Khosla, N. Jayadevaprakash, B. Yao, and F. fei Li. L.: Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR (2011*. 3

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 3

[13] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. *CoRR*, abs/1406.5298, 2014. 2

[14] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015. 3

[15] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *CoRR*, abs/1611.02163, 2016. 2

[16] M. Mirza and S. Osindero. Conditional generative adversarial nets. *CoRR*, abs/1411.1784, 2014. 1

[17] PyTorch. Pytorch. `https://github.com/pytorch/pytorch`, 2017. 3

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015. 1, 3

[19] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. 2

[20] N. Siddharth, B. Paige, A. Desmaison, J.-W. V. de Meent, F. Wood, N. D. Goodman, P. Kohli, and P. H. S. Torr. Inducing interpretable representations with variational autoencoders, 2016. 2

[21] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and

R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3483–3491. Curran Associates, Inc., 2015. 2

[22] K. Sun and X. Zhang. Coarse grained exponential variational autoencoders. *CoRR*, abs/1702.07904, 2017. 2

[23] L. van der Maaten and G. Hinton. Visualizing high-dimensional data using t-sne. 2008. 5

[24] S. Zhao, J. Song, and S. Ermon. Towards deeper understanding of variational autoencoding models. *CoRR*, abs/1702.08658, 2017. 2