REPRODUCIBLE OR NOT REPRODUCIBLE, THAT IS THE QUESTION

Harry Cooper, Alexander Llewellyn, Christopher Teakle
{hc3g17,al11g17,ct5g16}@soton.ac.uk

ABSTRACT

As part of the reproducibility challenge for COMP6248, we attempted to reproduce some of the experiments and results from RealnessGAN, an adjustment to a traditional generative model, where realness is instead treated as a random distribution rather than a scalar value. Our results confirm that RealnessGANs do give an advantage over traditional generative models. We verify this using the same data-set as the authors, and a new data-set to make sure the hypothesis holds. All code used is available on the COMP6248 Reproducibility Challenge GitHub[1].

1 INTRODUCTION

The paper titled "REAL OR NOT REAL THAT IS THE QUESTION", published at the International Conference on Representation Learning 2020, proposes a RealnessGAN to generalise the standard framework of generative adversarial networks (GAN) (Goodfellow et al. (2014)) "by treating realness as a random variable, represented as a distribution rather than a single scalar" (Xiangli et al. (2020)). This report investigates how reproducible the experiments and conclusions of this paper are. Included are the results from reproducing the experiments when trained on the synthetic uniform Gaussian data-set, the results from reproducing the experiments when trained on the CelebA data-sets, and comparing the Sliced-Wasserstein Distance (SWD) results when trained on the MNIST data-set, using our own implementation as opposed to the CelebA and CIFAR10 data-sets used in the paper, in order to establish the effectiveness of the RealnessGAN.

2 REALNESS GAN OVERVIEW

A standard implementation of a generative adversarial network (GAN) (Goodfellow et al. (2014)) makes use of a scalar value to estimate the realness of a generated image. The RealnessGAN paper (Xiangli et al. (2020)) instead changes the realness metric to be a distribution, in order to mimic humans perceiving multiple criteria of an image. The objective function for the generator is shown in Equation 1 (Equation 19, Xiangli et al. (2020))

$$\min_G \mathbb{E}_{\boldsymbol{x}\sim p_{data},\, \boldsymbol{z}\sim p_{\boldsymbol{z}}}[D_{KL}(D(\boldsymbol{x})||D(G(\boldsymbol{z}))] - \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}}[D_{KL}(A_0||D(G(\boldsymbol{z}))] \tag{1}$$

Figure 1 (Figures 4b., 4d. Xiangli et al. (2020)) shows the sliced-Wasserstein distance (SWD) per epoch of the RealnessGAN compared to other types of GANs on the CelebA and CIFAR10 data-sets. Importantly, the RealnessGAN outperforms all other GAN implementations including the standard GAN (Std-GAN).

Sliced-Wasserstein distance measures the difference in distribution of one set of images from another, in this case, the original images and the generated fake images.

A lower SWD value indicates that the images are closer in distribution, and are hence better fakes (Rabin et al. (2011)).

[1] https://github.com/COMP6248-Reproducability-Challenge/Reproducible-Or-Not-Reproducible-That-Is-The-Question

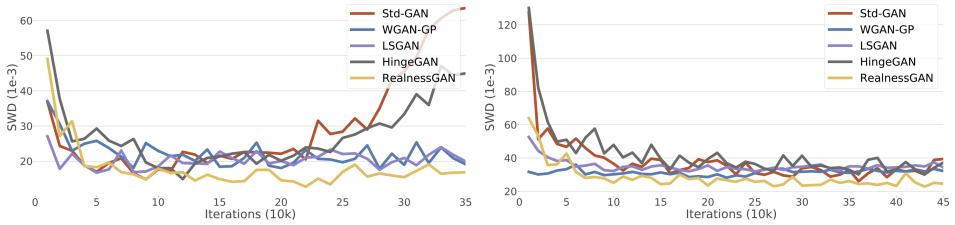

Figure 1: SWD plots from the original paper using: left) CelebA, right) CIFAR10

3 UNIFORM GAUSSIAN

Xiangli et al. (2020) first test the RealnessGAN's mode coverage ability on a synthetic data-set. The synthetic data-set consists of 100,000 2D points that are sampled from 9 isotropic Gaussian distributions with the means of the distributions arranged in a 3 by 3 grid.

To test the output that the paper outlines we wrote a Python program to plot the outputs from the script that were being saved in the pickle file format (the author did not provide such a program). This program also printed out the percentage of 'high quality' samples and the number of 'recovered modes', using the method described in the paper. A sample is regarded as high quality if it is within 4σ of the μ of the nearest Gaussian. A mode is considered recovered if it is assigned with more than 100 high quality samples.

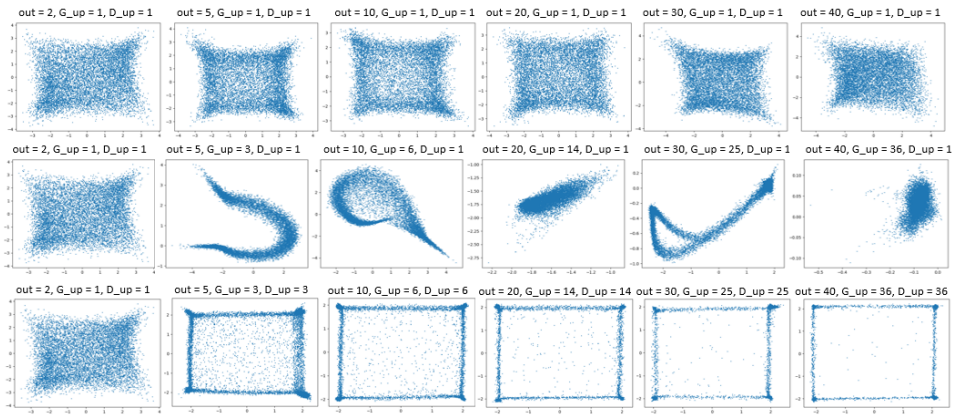

Figure 2: Reproduced results, top row) when G_updates and D_updates were fixed at 1, middle row) when G_updates was incremented accordingly and D_updates fixed, bottom row) when G_updates and D_updates incremented

We used the method that Xiangli et al. (2020) outlined to test how reproducible the results are. Figure (2) shows the results produced and Figure (3) shows the plots of percentage of high quality samples and number of recovered modes for the different settings used. We can see that the results when G_updates and D_updates were fixed to 1 are similar to the results that the paper outlines (Figure (3) Xiangli et al. (2020)) and agree with the author's statement that "in general G recovers less modes as the number of outcomes grows". They state this is a "direct result of D becoming increasingly rigorous and imposing more constraints on G". However, when we incremented G_updates with D_updates being fixed, our results were very different to the paper, with the number of recovered modes that we observed being 1 with the number of outcomes set to 40, as opposed to 8 modes in the paper.

We conducted further tests identifying settings that could produce better results. Increasing D_updates together with G_updates produced better results, although still worse than what the paper outlines. Under these settings the number of high quality samples increased as number of outcomes increased with 4 modes being recovered. Increasing the number of total_iters (number of iterations) past the 500 that the paper states also improved the results, with 1000 total_iters recovering 4 modes for 2 number of outcomes. Therefore, we propose that the results using the synthetic data-set are not

fully reproducible using the settings outlined in the paper, but are likely achievable using different settings.

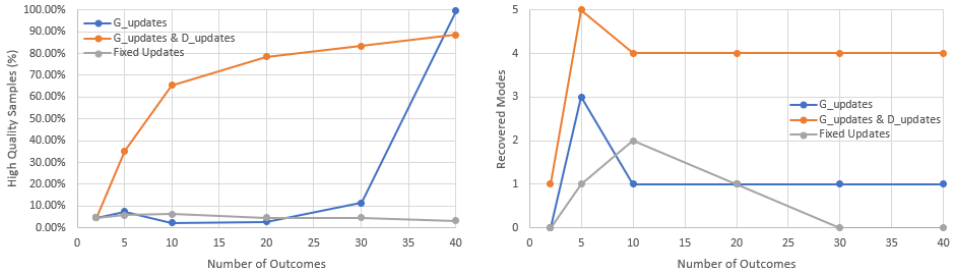

Figure 3: Plots of results,left) showing the percentage of high quality samples as the number of outcomes increases, right) showing the number of recovered modes as the number of outcomes increases (Grey when G_updates and D_updates were fixed at 1, Blue when G_updates was incremented accordingly and D_updates fixed, Orange when G_updates and D_updates incremented)

4 CelebA Data-Set

To verify the results of the RealnessGAN paper the RealnessDCGAN was trained on the CelebA data-set for parameters specified in the RealnessGAN paper (Xiangli et al. (2020)). It was quickly found that the authors code could not run on a Windows OS due to file system management, therefore the provided code was altered allowing code to run. However, minor bugs still persisted that would lead to code crashing occasionally, this could be due to code being run on a virtual machine with checkpoint results being saved to a network drive. Model could be recovered from saved checkpoints and for the total 520K training iterations it took the model 4 days to train on a computer using a RTX 2070, this training time could be reduced to potentially 2-3 days if the occasional bugs do not occur.

The results of the RealnessDCGAN are shown by Figures (4, 5, 6, 7) which contain a sample of fake images of the CelebA dataset at the stated iterations. Figures 6 and 7 show that the generated fake images are reasonably similar indicating that model has mostly converged, this can be viewed by the SWD plots from Figure 1 (Figures 4b., 4d. Xiangli et al. (2020)). Figures (4) and (5) show the largest differences from Figure 7 showing that the model is still training at these points.

While these figures show RealnessDCGAN producing images of human faces, at a closer inspection by eye, for some of the generated images it can easily be discerned that the images are fake. This seems to be due to the model not understanding human hair particularly well or specific background textures that end up merged with the hair or human face.

Figure 4: Reconstruction of CelebA images using author's implementation at iteration 1k

Figure 5: Reconstruction of CelebA images using author's implementation at iteration 50k

5 Realness GAN on MNIST Data-set

In order to test the hypothesis that using a RealnessGAN produces better fake images than other types of GAN, we re-implemented the paper without using the author's code, using the MNIST data-set to compare the models. The reasoning for using the MNIST data-set was because training the model on the CelebA data-set would take approximately 2 days. However, since the MNIST data-set

Figure 6: Reconstruction of CelebA images using author's implementation at iteration 200k

Figure 7: Reconstruction of CelebA images using author's implementation at iteration 520k

only makes use of 28x28 gray-scale images, the training time could be reduced to approximately 10 minutes, as a significantly lower number of epochs are required for the train losses to plateau.

In order to calculate the SWD score at each epoch, we made use of the `swd-pytorch`[2] package, comparing 32 MNIST train images to 32 produced fakes at the end of every epoch.

The results from the re-implementation experiments are shown in Figure 8. The results show a considerable amount of noise, however the underlying trend shows that the RealnessGAN consistently produced better fakes than the standard GAN.

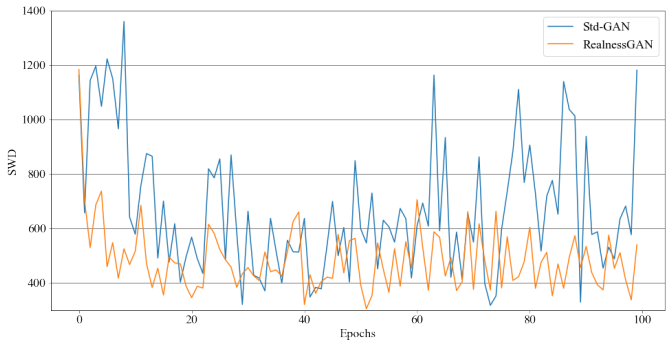

Figure 8: Reproduced SWD results on MNIST data-set

6 Conclusion

For the synthetic data-set the results in the paper were not fully reproducible using the settings described in the paper, but would likely be achievable under different settings. The results of the RealnessGAN on the real-world CelebA and MNIST data-sets were reproducible, with the RealnessGAN creating better fakes than a standard GAN. However, in order to keep computational time and power to a minimum, a smaller, less complex data-set was used. Furthermore, due to development effort constraints, the GANs presented were not compared against a Wasserstein GAN (WGAN), Least-Square GAN (LSGAN) or a HingeGAN, as seen in the original paper (Xiangli et al. (2020)).

References

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. volume 8, pp. 435–446, 05 2011. ISBN 978-3-642-24784-2. doi: 10.1007/978-3-642-24785-9_37.

Yuanbo Xiangli, Yubin Deng, Bo Dai, Chen Change Loy, and Dahua Lin. Real or not real, that is the question, 2020.

[2]https://github.com/koshian2/swd-pytorch