A QUANTITATIVE MEASURE OF GENERATIVE ADVERSARIAL NETWORK DISTRIBUTIONS

Dan Hendrycks*
University of Chicago
dan@ttic.edu

Steven Basart*
University of Chicago
xksteven@uchicago.edu

ABSTRACT

We introduce a new measure for evaluating the quality of distributions learned by Generative Adversarial Networks (GANs). This measure computes the Kullback-Leibler divergence from a GAN-generated image set to a real image set. Since our measure utilizes a GAN's whole distribution, our measure penalizes outputs lacking in diversity, and it contrasts with evaluating GANs based upon a few cherry-picked examples. We demonstrate the measure's efficacy on the MNIST, SVHN, and CIFAR-10 datasets.

1 INTRODUCTION

Generative adversarial networks (GANs) are a class of generative models that use neural networks for realistic data generation Goodfellow et al. (2014). While there have been numerous works on the subject, a remaining problem is a grounded way to compare GANs. Currently, comparisons are largely qualitative and based upon cherry-picked samples from a GAN. Cherry-picked examples emphasize small sections of the distribution learned by a GAN, while a more meaningful evaluation would utilize entire GAN distributions. We are interested in approximating a natural image distribution not approximating a handful of points from said distribution.

There are few prior quantitative measures to evaluate GANs. For example, Im et al. (2016) proposes a relative measure between two GANs that essentially compares the ability of the discriminator to discriminate between real and fake. Another metric is the Inception score (Salimans et al., 2016) which is a "rough guideline." This useful guideline is unfortunately ineffective should the GAN learn to constantly generate a single compelling example, and this measure is limited to images like CIFAR-10. Other measures such as total variation and calculating the number of missing modes can be useful features but assuredly do not capture most generative modeling desiderata. The work by Lopez-Paz & Oquab (2017) provides a parametric approach to compare distributions through a discriminator. Comparing GAN performance can be done in a semi-supervised setting where GANs generate a large corpus of data from a small number of labeled data (Salimans et al., 2016).

Another technique to evaluate GANs is Parzen window estimation. This non-parametric method can be used to compute the log-likelihood of GAN samples. However as Theis et al. (2015) show Parzen window estimates can be a poor approximation when the data is high-dimensional. We show that this is indeed the case for a complex dataset like CIFAR-10. Our measure addresses the concerns facing kernel density estimators by embedding the samples into a lower dimensional space and by using KL-divergence instead of computing log-likelihoods.

This work provides a measure that can quantitatively evaluate the quality of GAN models. To do this, we measure the similarity of a GAN's generated distribution to the distribution of a held out test set. By using the entire distribution, our measure penalizes against GANs which lack diverse outputs. Since it compares entire distributions, it contrasts with the current trend of summarizing a GAN's quality via unrepresentative cherry-picked samples like in Figure 2.

Our contributions are as follows:

*Equal contribution. Author ordering determined by coin flip.

- The measure approximates the divergence between generated and real distributions. The measure compares entire distributions rather than comparing on a per-image basis.
- The measure tracks image quality.
- The measure penalizes GANs that miss modes.

2 MEASURES

We now establish notation to define to define our measure. The GAN distribution is approximated through m images sampled from the GAN, giving us the images $G = \{g_i\}_{i=1}^m$, $g_i \in \mathbb{R}^d$. The set of n real images is denoted $R = \{r_i\}_{i=1}^n$. We desire to find the divergence from the GAN distribution to the real image distribution, but finding the KL divergence from our *empirical* GAN distribution G to R would likely be infinite because not all images in R are in G. To meet this challenge, we make a continuous approximation of the distributions G and R. We will use the empirical distribution G to make a Gaussian mixture model with each image g_i defining a component of this Gaussian mixture model. Likewise we use R to create a Gaussian mixture model for real images to approximate the real image distribution. We approximate the GAN distribution by sampling images and letting each image become a component of a Gaussian mixture model.

The representation of images with a Gaussian mixture is described below. First, we compute the singular value decomposition U_G, Σ_G, V_G of G and replace each image g_i with the image's principal component coefficients g_i'. Then each image is replaced with the Gaussian centered around g_i', $\mathcal{N}(g_i', \Sigma_G)$. Each Gaussian in the Gaussian mixture model has equal weight. We also truncate the coefficient vector g_i' and the diagonal(**is it necessarily diag**) matrix Σ_G so as to preserve the most important reconstruction information while reducing dimensionality. Thus the GAN distribution is represented as $\mathcal{D}_{\text{GAN}} = \frac{1}{m}\sum_{i=1}^m \mathcal{N}(g_i', \Sigma_G)$. We do the same process to create a Gaussian mixture model for the set R.

We would now like to capture the statistical distance between the GAN Gaussian mixture model and real Gaussian mixture model, so we compute $\text{KL}[\mathcal{D}_{\text{Real}} \| \mathcal{D}_{\text{GAN}}]$. To accomplish this we will use a fast approximation. First, note that the KL divergence between any two d-dimensional Gaussians has the closed form value

$$\text{KL}[\mathcal{N}(r_i', \Sigma_R) \| \mathcal{N}(g_j', \Sigma_G)] = \frac{1}{2}\left[\log\frac{|\Sigma_G|}{|\Sigma_R|} + \text{Tr}(\Sigma_G^{-1}\Sigma_R) - d + (r_i' - g_j')^\intercal \Sigma_G^{-1}(r_i' - g_j')\right].$$

Unfortunately, this does not give us the KL divergence between two Gaussian mixture models. Goldberger et al. (2003) approximates the KL divergence between Gaussian mixture models by computing the KL divergence from a Gaussian in a mixture model and its closest Gaussian from the other mixture. That is, they let

$$\pi(i) = \operatorname*{argmin}_j \text{KL}[\mathcal{N}(r_i', \Sigma_R) \| \mathcal{N}(g_j', \Sigma_g)],$$

and approximate the KL divergence between Gaussian mixtures with

$$\frac{1}{n}\sum_{i=1}^n \text{KL}[\mathcal{N}(r_i', \Sigma_R) \| \mathcal{N}(g_{\pi(i)}', \Sigma_G)].$$

This KL divergence approximation is how we compute our GAN quality measure.

3 EXPERIMENTS

On MNIST, SVHN, and CIFAR-10, we visualize our measure as GANs train.

For MNIST, we use DCGAN (Radford et al., 2016) and DCGAN without batch normalization (Ioffe & Szegedy, 2015). Removing batch normalization makes the samples of far lower quality. When replacing the images with their principal component coefficients in the Gaussian mixture model, we keep the first 150 coefficients to preserve most of the data variance. In Figure 1, we observe that the DCGAN without batch normalization produces low-quality and redundant outputs. In contrast, DCGAN with batch normalization improves sample quality and diversity. Our measure correctly reflects this in Figure 3.

For SVHN, we use a DCGAN and the higher-quality Improved GAN (Salimans et al., 2016) for our comparisons. When replacing the images with their principal component coefficients in the Gaussian mixture model, we keep the first 50 coefficients to preserve most of the data variance. Figure 4 shows samples from DCGAN and the Improved GAN. Figure 5 shows that the divergence is less for the higher quality images from the Improved GAN distribution, as it should be.

For CIFAR-10, we use the Improved GAN for high-quality examples and we remove batch normalization to create lower quality examples. When replacing the images with their principal component coefficients in the Gaussian mixture model, we keep the first 500 coefficients to preserve most of the data variance. Figure 6 shows samples from the Improved GAN without and with batch normalization. Figure 7 shows smaller divergence for the higher quality images made by the unhindered Improved GAN.

As a comparison to our measure, Parzen windows allows us to compute the log-likelihoods for the examples generated by the Improved GAN and Improved GAN without batch normalization. Figure 8 shows how Parzen window estimation does not provide a desirable measure for comparing the two GAN distributions, and at the same time Figure 9 exhibits how our measure is unlike compressing examples with their primary PCA coefficients and then using Parzen windows. This shows how our method better captures the quality differences between the two distributions.

4 Conclusion

We considered a measure which aimed to approximate the divergence from a GAN's distribution to a real distribution. The measure penalized low-quality samples and mode-missing GANs. Ultimately, we hope that this nonparametric measure can be used as an informative guideline in future research.

Acknowledgments

We would also like to thank NVIDIA Corporation for donating several TITAN X GPUs used in this research.

References

Jacob Goldberger, Shiri Gordon, and Hayit Greenspan. Approximating the kullback leibler divergence between gaussian mixture models. In *International Conference on Computer Vision (ICCV)*, 2003.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Daniel Jiwoong Im, Roland Memisevic, Chris Dongjoo Kim, and Hui Jiang. Generative adverarial metric [sic]. In *International Conference on Learning Representations (ICLR) Workshop*, 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Learning Representations (ICLR)*, 2015.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations (ICLR)*, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Neural Information Processing Systems (NIPS)*, 2016.

Lucas Theis, Aaron van den Oord, and Matthis Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR)*, 2015.

APPENDIX: FIGURES

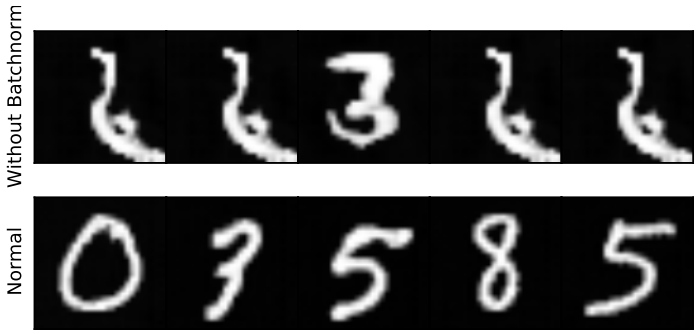

Figure 1: MNIST samples from DCGAN without batch normalization and DCGAN.

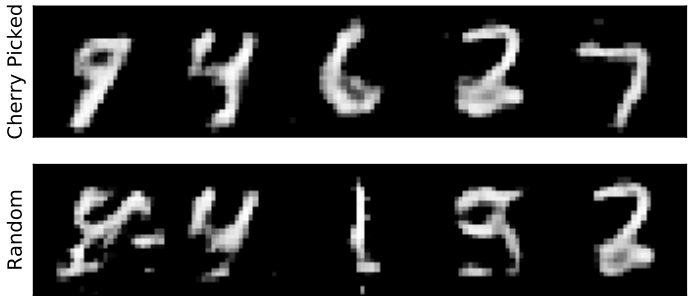

Figure 2: MNIST learned by a DCGAN without Batch Normalization show that cherry-picked examples accrued from a GAN can be deceptive.

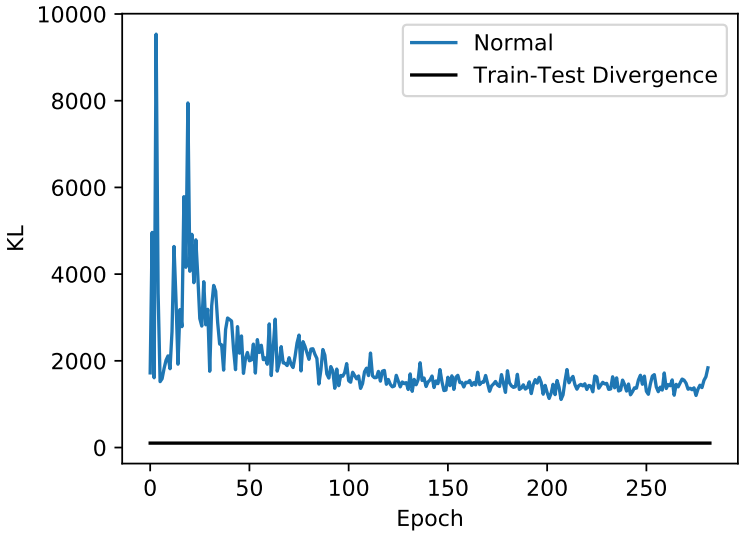

Figure 3: MNIST results. The Gaussian divergence of the DCGAN without batch normalization was several orders of magnitude larger than the normal DCGAN (over one million) and thus is not depicted. The black line indicates the divergence approximation between train and test images.

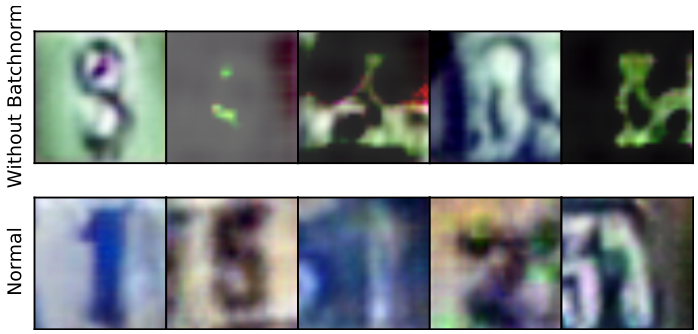

Figure 4: SVHN samples from DCGAN and Improved GAN. Images are randomly sampled during the end of training.

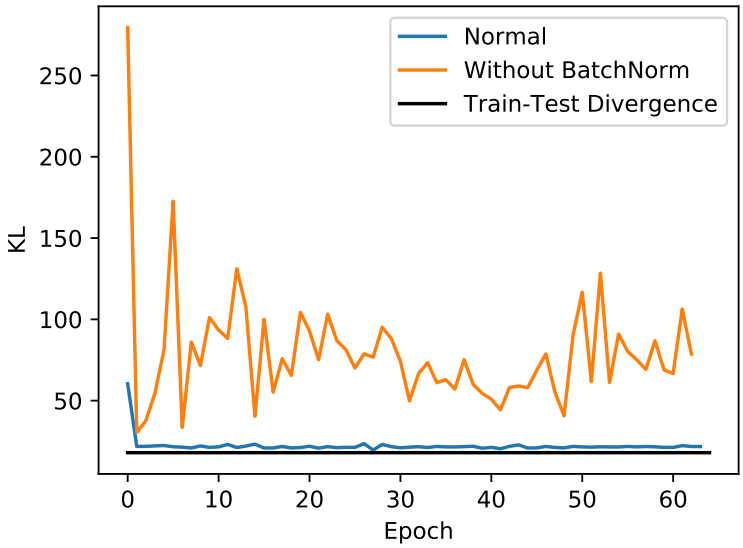

Figure 5: SVHN results. The black line indicates the divergence approximation between train and test images.

Figure 6: CIFAR samples from Improved GAN without batch normalization and the normal Improved GAN. Images are randomly sampled during the end of training.

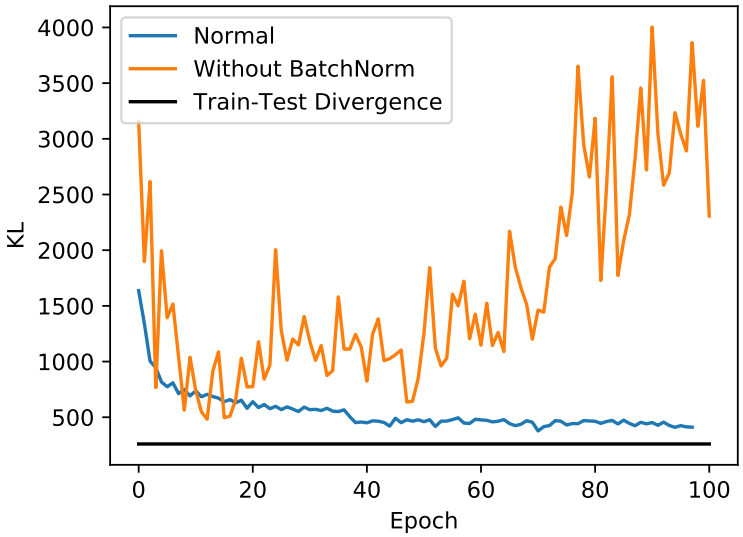

Figure 7: CIFAR-10 results. The black line indicates the divergence approximation between train and test images.

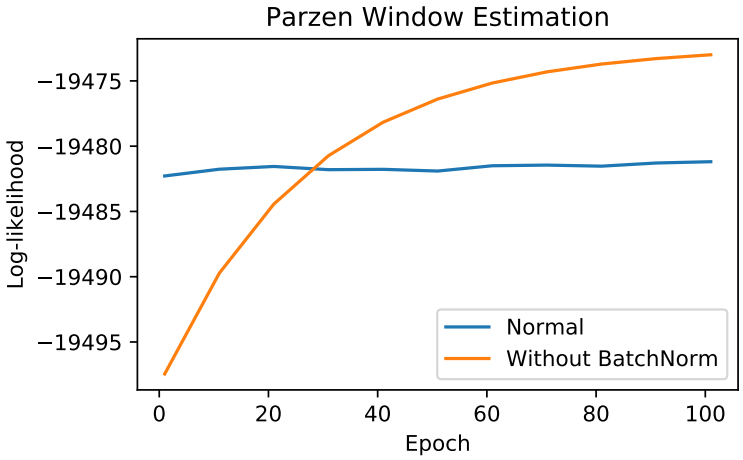

Figure 8: Parzen window estimation on CIFAR-10. The samples from the model without Batch Normalization never achieve good image quality, yet they improve beyond the normal model with respect to log-likelihood.

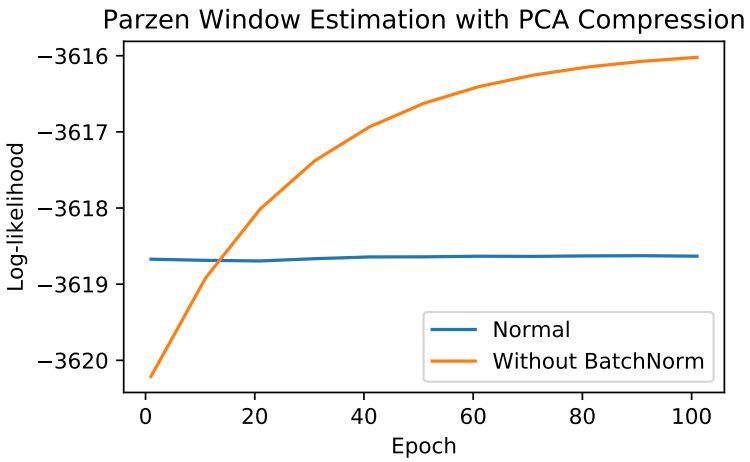

Figure 9: Parzen window estimation on CIFAR-10 examples truncated at the first 500 principal components coefficients. Consequently, a lower dimensional embedding is not sufficient for Parzen window estimates to track image quality.