Evaluating Generative Adversarial Networks on Explicitly Parameterized Distributions

Shayne O'Brien, Matt Groh, Abhimanyu Dubey
{shayneob, groh, dubeya}@mit.edu
Massachusetts Institute of Technology

Abstract

The true distribution parameterizations of commonly used image datasets are inaccessible. Rather than designing metrics for feature spaces with unknown characteristics, we propose to measure GAN performance by evaluating on explicitly parameterized, synthetic data distributions. As a case study, we examine the performance of 16 GAN variants on six multivariate distributions of varying dimensionalities and training set sizes. In this learning environment, we observe that: GANs exhibit similar performance trends across dimensionalities; learning depends on the underlying distribution and its complexity; the number of training samples can have a large impact on performance; evaluation and relative comparisons are metric-dependent; diverse sets of hyperparameters can produce a "best" result; and some GANs are more robust to hyperparameter changes than others. These observations both corroborate findings of previous GAN evaluation studies and make novel contributions regarding the relationship between size, complexity, and GAN performance.

1 Introduction

Generative adversarial network (GAN) optimization stability and convergence properties remain poorly understood despite the introduction of hundreds of GAN variants since their conception [**? ?**]. While GAN learning and performance behavior has been studied [**? ? ?**], most existing work examining this relationship focuses on image datasets for which the underlying distribution parameterization is inaccessible [**? ? ? ? ?**]. This is problematic since claims of behavior that are made by modeling an unknown target distribution require a strong assumption for generalizability.

The goal of generative modeling is to approximate a distribution p_d by learning a parameterized distribution p_g, where both p_d and p_g are defined over samples. If we do not have full access to p_d, generalizability requires us to assume that the modeled dataset is a reasonable proxy for the family of distributions from which it was sampled. Without this assumption that is often only implicitly made, using images to understand GAN behavior limits conclusions to the data context being modeled.

We seek to address a gap in the literature by investigating GAN variant performance on datasets for which we have full access to the distribution parameterization. This allows us to study empirical performance on data where we can make claims of model behavior that generalize to the full distribution, as opposed to on image datasets for which this is not necessarily true. To this end, we examine the performance of 16 GAN variants on six explicitly parameterized multivariate distributions of four different dimensionalities and three different training set sizes.

Across 20 grid search trials, we observe that: (1) GANs exhibit similar performance trends across dimensionalities, (2) learning depends on the underlying distribution and its complexity, (3) the

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

number of training samples can have a large impact on performance, (4) evaluation and relative comparisons are metric-dependent, (5) diverse sets of hyperparameters can produce a "best" result, and (6) some GANs are more robust to hyperparameter changes than others. These findings corroborate those of previous GAN evaluation studies as well as contribute novel insights regarding the relationship between size, complexity, and GAN performance.[1]

2 Related Work

One notable work in this area by **?**] compares seven GAN variants in terms of modeling ability and optimization stability. The authors find that as computational budget increases, all tested models reach similar Frechét Inception Distance on the MNIST, Fashion-MNIST, CIFAR10, and CelebA datasets; and F1, precision, and recall on a synthetic dataset of convex polygons. They also discuss the difficulties of comparing GANs due to multiple valid ways to analyze performance.

?] measure Inception Score and classification accuracy and report that the five GAN variants they train do not succeed at capturing distributional properties of the training set on the CelebA and LSUN datasets. The authors observe that the GAN distributions exhibit significantly less diversity at test time compared to the evaluation dataset, suggesting p_g is far from p_d.

In another study, **?**] evaluate GAN variant performance based on the original GAN criterion, least squares, maximum mean discrepancy, and improved Wasserstein distance. They show that for the three GAN variants they consider, test-time metrics do not favor networks that use the same training-time criterion on the MNIST, CIFAR10, LSUN, and Fashion-MNIST image datasets. The authors also examine performance as a function of sample size and show that some GANs exhibit faster performance increases than others as the number of training samples increases.

Lastly, **?**] provide a thorough discussion of the strengths and weaknesses of 26 quantitative and qualitative measures used for evaluating GANs trained on image datasets. They conclude that there is no single, best GAN evaluation measure. The authors suggest benchmarking models under identical architectures and computational budgets, and using more than a single metric to make comparisons.

3 Experimental Setup

In GANs, we define a prior probability distribution on input noise variables $p_z(\mathbf{z})$ and represent a mapping to the target data space $p_d(\mathbf{x})$ as $G(\mathbf{z}, \theta_G)$, where G is a fully differentiable neural network called the *generator* and θ_G are its parameters. We train G by simultaneously learning a fully differentiable network D, called the *discriminator* or *critic* and defined by $D(\mathbf{x}, \theta_D)$, that helps G during training. Whereas G is trained to mimic p_d, the learning objective, output, and precise task of D vary depending upon the GAN variant.

Models

As a case study, we examine the same seven GAN variants evaluated by **?**] and nine additional GAN variants that have been popularly discussed since their study was published. The primary difference between considered variants is whether the discriminator output can be interpreted as a probability (MMGAN, NSGAN [**?**], RaGAN [**?**], DRAGAN [**?**], FisherGAN [**?**], InfoGAN [**?**], ForwGAN, RevGAN, HellingerGAN, PearsonGAN, JSGAN [**?**]) or is unbounded (WGAN [**?**], WGANGP [**?**], LSGAN [**?**], BEGAN [**?**]). We summarize these models in Table 1.

In our implementations, both D and G consist of two feedforward network layers each; the full architecture has four layers total. We apply a ReLU activation function to the output of each layer and sample the noise prior $\mathbf{z}$ from $\mathcal{N}(0, \frac{h}{4}I)$, where h is the hidden dimension size. All models have the same number of trainable parameters except InfoGAN and BEGAN due to their use of latent variables as inputs to D and formation of D as an autoencoder, respectively. **?**] argue that this difference is negligible for InfoGAN and we do not observe that it gives BEGAN any tangible advantage over other models. Trainable parameter counts can be found in Table 8.

[1] All code is publicly available at `https://github.com/shayneobrien/explicit-gan-eval`.

GAN Variant Loss Functions	
$\mathcal{L}^{\text{MMGAN}} = \mathbb{E}[\log(D(\mathbf{x}))] + \mathbb{E}[\log(1 - D(G(\mathbf{z})))]$	$\mathcal{L}^{\text{RaGAN}} = \mathbb{E}[\log(D(\mathbf{x}) - D(G(\mathbf{z})))] + \mathbb{E}[\log(1 - (D(G(\mathbf{z})) - D(\mathbf{x})))]$
$\mathcal{L}^{\text{NSGAN}} = \mathbb{E}[\log(D(\mathbf{x}))] - \mathbb{E}[\log(D(G(\mathbf{z})))]$	$\mathcal{L}^{\text{LSGAN}} = -\mathbb{E}[(D(\mathbf{x}) - 1)^2] + \mathbb{E}[D(G(\mathbf{z}))^2]$
$\mathcal{L}^{\text{WGAN}} = -\mathbb{E}[D(\mathbf{x})] + \mathbb{E}[D(G(\mathbf{z}))]$	$\mathcal{L}^{\text{BEGAN}} = \mathbb{E}[\|\mathbf{x} - D_{\text{AE}}(\mathbf{x})\|_1] - k_t\mathbb{E}[\|G(\mathbf{z}) - D_{\text{AE}}(G(\mathbf{z}))\|_1]$
$\mathcal{L}^{\text{WGANGP}} = \mathcal{L}^{\text{WGAN}} + \lambda\mathbb{E}\left[(\|\nabla_{\mathbf{z}} D(G(\mathbf{z}))\|_2 - 1)^2\right]$	$\mathcal{L}^{\text{DRAGAN}} = \mathcal{L}^{\text{MMGAN}} + \lambda\mathbb{E}[(\|\nabla_{\mathbf{x}} D(\mathbf{x} + \delta))\|_2 - 1)^2]]$
$\mathcal{L}^{\text{FisherGAN}} = \mathcal{L}^{\text{WGAN}} + \lambda(1 - \hat{\Omega}(D, G)) - \frac{\rho}{2}(\hat{\Omega}(D, G) - 1)$	$\mathcal{L}^{\text{InfoGAN}} = \mathcal{L}^{\text{MMGAN}}$ - $\lambda(\mathbb{E}[\log(Q(\mathbf{c}'\|\mathbf{x}))])$
$\mathcal{L}^{\text{PearsonGAN}} = \mathbb{E}[D(\mathbf{x})] + \mathbb{E}[\frac{1}{4}D(G(\mathbf{z}))^2 + D(G(\mathbf{z}))]$	$\mathcal{L}^{\text{TVGAN}} = -\frac{1}{2}\mathbb{E}[\tanh(D(\mathbf{x}))] + \frac{1}{2}\mathbb{E}[\tanh(D(G(\mathbf{z})))]$
$\mathcal{L}^{\text{ForwGAN}} = \mathbb{E}[D(\mathbf{x})] + \mathbb{E}[\exp(D(G(\mathbf{z}))) - 1]$	$\mathcal{L}^{\text{RevGAN}} = \mathbb{E}[-\exp(D(\mathbf{x}))] + \mathbb{E}[-1 - (D(G(\mathbf{z})))]$
$\mathcal{L}^{\text{HellingerGAN}} = \mathbb{E}[1 - \exp(-D(\mathbf{x}))] + \mathbb{E}[\frac{1-\exp(D(G(\mathbf{z})))}{\exp(D(G(\mathbf{z})))}]$	$\mathcal{L}^{\text{JSGAN}} = \mathbb{E}[2 - (1 + \exp(-D(\mathbf{x})))] - \mathbb{E}[2 - \exp(D(G(\mathbf{z})))]$

Table 1: The loss function for each GAN variant with slight abuse of parameterization notation on the expectations, G, and D. Note that G is parameterized by θ_G, D is parameterized by θ_D, $x \sim p_d$, $z \sim p_g$, $\delta \sim \mathcal{N}(0, cI)$, $\hat{\Omega}(D, G) = \frac{1}{2}\mathbb{E}[D(\mathbf{x})]^2 - \frac{1}{2}\mathbb{E}[D(G(\mathbf{z}))]^2$, D_{AE} indicates that D is an autoencoder, $\mathbf{c} = [\mathbf{c_1}, \mathbf{c_2}]$ are structured latent variables where $\mathbf{c}'$ is sampled from the approximated distribution $p_{\mathbf{c}}(\mathbf{c}|\mathbf{x})$, $\nabla_{(\cdot)}$ is the gradient of the loss with respect to $(\cdot)$, and k_t, λ, and ρ are introduced hyperparameters.

Data

We train each of these variants by randomly sampling 1,000, 10,000, and 100,000 data points from the following six explicitly parameterized multivariate distributions: Gaussian with mean μ and symmetric, full rank covariance $\mathbf{\Sigma}$ both from $[0, 1]$; exponential with inverse mean shape λ from $[0, 1]$; beta with shape parameters a and β both from $[0, 1]$; gamma with shape k from $[0, 10]$ and scale θ from $[0, 2]$; Gumbel with location μ and scale β both from $[0, 1]$; and Laplace with location μ and scale β both from $[0, 1]$. For each of these distributions and numbers of samples, we generate datasets of 16, 32, 64, and 128 dimensions. We note that by the Universal Approximation Theorem, our proposed network architecture should be able to model each of these distributions without exception.

Hyperparameters

For all models and data distributions, we conduct 20 grid search trials with random network initializations for learning rates $\gamma \in [2\text{e}^{-1}, 2\text{e}^{-2}, 2\text{e}^{-3}]$, hidden dimension sizes $h \in [32, 64, 128, 256, 512]$, and batch size $b = 1024$.[2] For models with introduced hyperparameters, we use those given in the original the paper. We use the Adam optimizer with default settings [**?**] and train for 25 epochs.

Measuring Divergence

We evaluate the difference between p_d and p_g using Kullback-Leibler divergence (KL), Jensen-Shannon divergence (JS), and Wasserstein Distance (WD). KL and JS focus on the alignment of the modes of the distributions and WD emphasizes how much p_g must be modified to reach p_d. Whereas JS and WD are symmetric, KL is not. For any of these measures, a value of 0 can be interpreted as indicating the two distributions being compared are identical [**? ? ?**]. We report results as the divergence between a generated batch and a test batch of size $b = 1024$ at the end of every epoch.

Estimating p_g

Although we have access to the true data distribution p_d, we must estimate the probability distribution of p_g. Since the data dimensionality is low, we construct a dimension-wise histogram for each data point.[3] In doing so, we assume that each dimension is independent from the others. This assumption is valid in the case of all experiments involving non-Gaussian data, which follows from the multivariate model being a product of the marginal distributions. To select the optimal bin width B_w in the histogram, we follow the Freedman-Diaconis rule: $B_w = \frac{2 \cdot IQR(\tilde{\mathbf{x}})}{\sqrt[3]{M}}$, where IQR is the

[2]We also ran full experiments for $b \in [128, 256, 512]$, but limit our analyses to b = 1024 as results across different batch sizes are not comparable due to greater noise in the data generation process at lower values of b.

[3]Kernel density estimation was found to give similar outputs while being more computationally expensive.

inter-quartile range of the M samples $\tilde{\mathbf{x}} = \{x_1, ..., x_M\}$ from the distribution being approximated. This initialization minimizes the difference between the areas under the empirical and theoretical probability distributions [**?**].

4 Results

In our analyses, we take the same approach as **?**] and **?**]: we let the "best" hyperparameter setting be the one that achieved the lowest minimum performance on average across all trials for each distribution, metric, and number of training samples, respectively. We include results, visualizations, and evidence to support all conclusions in the appendices. For the best hyperparameter settings in our learning environment, we find that:

1. **GANs exhibit similar learning trends across dimensionalities**: For many of the models, performance under the best hyperparameter setting consistently follows a trend across dimensionalities for all three tested metrics. At the same time, performance generally worsens with increased dimensionality. See Figures 1, 2, 3.
2. **Learning depends on the underlying distribution and its complexity:** Models which do well on some distributions perform poorly on the same distribution with higher dimensionality, or on other distributions of the same dimensionality. It is not immediately apparent that these differences are due to model design. **?**] make a similar finding in the case of image datasets with varying complexities. See Figures 1, 2, 3.
3. **Number of training samples can have a large impact on performance:** Some GAN variants are able to achieve the same performance learning from 1,000 samples as 10,000 or 100,000 samples, while others show large performance jumps with increased amounts of data. At the same time, almost all GAN variants begin to worsen in performance within five epochs for 1,000 training samples. The number of training samples seems to be critical to some models' performances, which was also noted by **?**]. See Figures 4, 5, and 6.
4. **Evaluation and comparison are metric-dependent:** Relative ranking of GAN variants according to performance varies depending on the evaluation metric used to rank them. No single GAN performed best across all metrics for any dataset or dimensionality. We concur with previous studies that GANs generally perform the same, although there are variants that perform worse than others on some distributions [**? ? ? ? ?**]. We warn against ranking models as relative differences can be marginal. See Tables 2, 3, 4, and 5.
5. **Diverse sets of hyperparameters can produce a "best" result:** Many, diverse hyperparameter settings yielded superior performances to the best average minimum performance, but these models did not achieve those minima with tight confidence bounds. Furthermore, we see that even on the best performing hyperparameter settings, our tested models preferred widely different hidden dimensionalities and learning rates; some variants with less parameters outperformed others that had more. We agree with previous work that it is important to present results that are able to be consistently reproduced [**? ? ? ? ?**]. See Table 6.
6. **Some GANs are more robust to hyperparameter changes than others:** With respect to the distribution, dimensionality, and training set size being approximated, some models yielded average minimum performances for more hyperparameters than others that fell within the confidence interval of the best average minimum performance under consideration. This is an indication that some GANs can perform well under a greater range of hyperparameter settings than others. See Table 7.

5 Future Work

In future work, we plan to analyze cases where GAN variants underperform relative to others and relate the characteristics of the distribution being modeled to the assumptions made in designing the variant, e.g. by empirically considering whether a normally distributed prior hurts performance on non-normal distributions. We would also like to use longer training times and more complex models to evaluate additional synthetic datasets such as multivariate mixture models, colored circles, and autoencoded image datasets.

Acknowledgements

We would like to thank Deb Roy, Iyad Rahwan, Manolis Zampetakis, and the Media Lab Consortium for their support of this research. Aleksander Mądry and Costis Daskalakis provided us with inspiration to pursue this project. Shayne O'Brien is partially funded through a National Science Foundation Graduate Research Fellowship.