arXiv:1711.10337v2 [stat.ML] 30 Nov 2017

Are GANs Created Equal? A Large-Scale Study

Mario Lucic* Karol Kurach* Marcin Michalski Sylvain Gelly Olivier Bousquet
Google Brain

Abstract

Generative adversarial networks (GAN) are a powerful subclass of generative models. Despite a very rich research activity leading to numerous interesting GAN algorithms, it is still very hard to assess which algorithm(s) perform better than others. We conduct a neutral, multi-faceted large-scale empirical study on state-of-the art models and evaluation measures. We find that most models can reach similar scores with enough hyperparameter optimization and random restarts. This suggests that improvements can arise from a higher computational budget and tuning more than fundamental algorithmic changes. To overcome some limitations of the current metrics, we also propose several data sets on which precision and recall can be computed. Our experimental results suggest that future GAN research should be based on more systematic and objective evaluation procedures. Finally, we did not find evidence that any of the tested algorithms consistently outperforms the original one.

1. Introduction

Generative adversarial networks (GAN) are a powerful subclass of generative models and were successfully applied to image generation and editing, semi-supervised learning, and domain adaptation [20, 24]. In the GAN framework the model learns a deterministic transformation G of a simple distribution p_z, with the goal of matching the data distribution p_d. This learning problem may be viewed as a two-player game between the *generator*, which learns how to generate samples which resemble real data, and a *discriminator*, which learns how to discriminate between real and *fake* data. Both players aim to minimize their own cost and the solution to the game is the Nash equilibrium where neither player can improve their cost unilaterally [8].

Various flavors of GANs have been recently proposed, both purely unsupervised [8, 1, 9, 4] as well as conditional [18, 19]. While these models achieve compelling results in specific domains, there is still no clear consensus on which GAN algorithm(s) perform objectively better than the others. That's partially due to the lack of robust and consistent metric, as well as limited comparisons which put all algorithms on equal footage, including the computational budget to search over all hyperparameters. Why is it important? Firstly, to help the practitioner choose a better algorithm from a very large set. Secondly, to make progress towards better algorithms and their understanding, it is useful to clearly assess which modifications are critical, and which ones are only good on paper, but don't make a significant difference in practice.

The main issue with evaluation stems from the fact that one cannot explicitly compute the probability $p_g(x)$. As a result, classic measures, such as log-likelihood on the test set, cannot be evaluated. As a remedy, many researchers focus on qualitative comparison, such as comparing the visual quality of samples. Unfortunately, such approach is subjective and possibly misleading [7].

As a remedy, two evaluation metrics were proposed to quantitatively assess the performance of GANs. Both assume access to a pre-trained classifier. *Inception Score (IS)* [21] is based on the fact that a good model should generate samples for which, when evaluated by the classifier, the class distribution has low entropy. At the same time, it should produce diverse samples covering all classes. In contrast, *Fréchet Inception Distance* is computed by considering the difference in embedding of true and fake data [10]. Assuming that the coding layer follows a multivariate Gaussian distribution, the distance between the distributions is reduced to the Fréchet distance between the corresponding Gaussians.

Our main contributions:

1. We provide a fair and comprehensive comparison of the state-of-the-art GANs, and empirically demonstrate that nearly all of them can reach similar values of FID, given a high enough computational budget.
2. We provide strong empirical evidence[1] that to compare GANs it is necessary to report a summary of distribution of results, rather than the best result achieved, due to the randomness of the optimization process and

*Indicates equal authorship. Correspondence to Mario Lucic (lucic@google.com) and Karol Kurach (kkurach@google.com).

[1] As a note on the scale of the setup, the computational budget to reproduce those experiments is approximately 60K P100 GPU hours.

model instability.

3. We assess the robustness of FID to mode dropping, use of a different encoding network, and provide estimates of the best FID achievable on classic data sets.
4. We introduce a series of tasks of increasing difficulty for which undisputed measures, such as precision and recall, can be approximately computed.
5. We will soon open-source our experimental setup and model implementations.

2. Background and Related Work

There are several ongoing challenges in the study of GANs, including their convergence properties [2, 17], and optimization stability [21, 1]. Arguably, the most critical challenge is their *quantitative* evaluation.

The classic approach towards evaluating generative models is based on model likelihood which is often intractable. While the log-likelihood can be approximated for distributions on low-dimensional vectors, in the context of complex high-dimensional data the task becomes extremely challenging. Wu et al. [23] suggest an annealed importance sampling algorithm to estimate the hold-out log-likelihood. The key drawback of the proposed approach is the assumption of the Gaussian observation model which carries over all issues of kernel density estimation in high-dimensional spaces. Theis et al. [22] provide an analysis of common failure modes and demonstrate that it is possible to achieve high likelihood, but low visual quality, and vice-versa. Furthermore, they argue against using Parzen window density estimates as the likelihood estimate is often incorrect. In addition, ranking models based on these estimates is discouraged [3]. For a discussion on other drawbacks of likelihood-based training and evaluation consult Huszár [11].

Inception Score (IS). Proposed by [21], IS offers a way to quantitatively evaluate the quality of generated samples. The score was motivated by the following considerations: (i) The conditional label distribution of samples containing meaningful objects should have low entropy, and (ii) The variability of the samples should be high, or equivalently, the marginal $\int_z p(y|x = G(z))dz$ should have high entropy. Finally, these desiderata are combined into one score,

$$\texttt{IS}(G) = \exp(\mathbb{E}_{x\sim G}[d_{KL}(p(y \mid x), p(y)])$$

The classifier is Inception Net trained on Image Net which is publicly available. The authors found that this score is well-correlated with scores from human annotators [21]. Drawbacks include insensitivity to the prior distribution over labels and not being a proper *distance*.

Fréchet Inception Distance (FID). Proposed by [10], FID provides an alternative approach. To quantify the quality of generated samples, they are first embedded into a feature space given by (a specific layer) of Inception Net. Then, viewing the embedding layer as a continuous multivariate Gaussian, the mean and covariance is estimated for both the generated data and the real data. The Fréchet distance between these two Gaussians is then used to quantify the quality of the samples, i.e.

$$\texttt{FID}(x, g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}),$$

where (μ_x, Σ_x), and (μ_g, Σ_g) are the mean and covariance of the sample embeddings from the data distribution and model distribution, respectfully. The authors show that the score is consistent with human judgment and more robust to noise than IS [10]. Furthermore, the authors present compelling results showing negative correlation between the FID and visual quality of generated samples. Unlike IS, FID can detect intra-class mode dropping, i.e. a model that generates only one image per class can score a perfect IS, but will have a bad FID. We provide a thorough empirical analysis of FID in Section 5.

A significant drawback of both measures is the inability to detect overfitting. A "memory GAN" which stores all training samples would score perfectly.

A very recent study comparing several GANs using IS has been presented by Fedus et al. [6]. The authors focus on IS and consider a smaller subset of GANs. In contrast, our focus is on providing a *fair assessment* of the current state-of-the-art GANs using FID, as well as precision and recall, and also verifying the robustness of these models in a large-scale empirical evaluation.

3. Flavors of Generative Adversarial Networks

In this work we focus on *unconditional* generative adversarial networks. In this setting, only unlabeled data is available for learning. The optimization problems arising from existing approaches differ by (i) the constraint on the discriminators output and corresponding loss, and the presence and application of gradient norm penalty.

In the original GAN formulation [8] two loss functions were proposed. In the *minimax* GAN the discriminator outputs a probability and the loss function is the negative log-likelihood of a binary classification task (MM GAN in Table 1). Here the generator learns to generate samples that have a low probability of being fake. To improve the gradient signal, the authors also propose the *non-saturating* loss (NS GAN in Table 1), where the generator instead aims to maximize the probability of generated samples being real.

In Wasserstein GAN [1] the discriminator is allowed to output a real number and the objective function is equivalent to the MM GAN loss without the sigmoid (WGAN in Table 1). The authors prove that, under an optimal (Lipschitz smooth) discriminator, minimizing the value function with respect to the generator minimizes the Wasserstein distance between model and data distributions. Weights of the

Table 1: Generator and discriminator loss functions. The main difference whether the discriminator outputs a probability (MM GAN, NS GAN, DRAGAN) or its output is unbounded (WGAN, WGAN GP, LS GAN, BEGAN), whether the gradient penalty is present (WGAN GP, DRAGAN) and where is it evaluated. We chose those models based on their popularity.

GAN	DISCRIMINATOR LOSS	GENERATOR LOSS
MM GAN	$\mathcal{L}_D^{GAN} = -\mathbb{E}_{x\sim p_d}[\log(D(x))] + \mathbb{E}_{\hat{x}\sim p_g}[\log(1-D(\hat{x}))]$	$\mathcal{L}_G^{GAN} = -\mathcal{L}_D^{GAN}$
NS GAN	$\mathcal{L}_D^{NSGAN} = \mathcal{L}_D^{GAN}$	$\mathcal{L}_G^{NSGAN} = \mathbb{E}_{\hat{x}\sim p_g}[\log(D(\hat{x}))]$
WGAN	$\mathcal{L}_D^{WGAN} = -\mathbb{E}_{x\sim p_d}[D(x)] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$	$\mathcal{L}_G^{WGAN} - = \mathcal{L}_D^{WGAN}$
WGAN GP	$\mathcal{L}_D^{WGAN} = \mathcal{L}_D^{WGAN} + \lambda\mathbb{E}_{\hat{x}\sim p_g}[(\|\|\nabla D(\alpha x + (1-\alpha\hat{x})\|\|_2 - 1)^2]$	$\mathcal{L}_G^{WGAN} = -\mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})]$
LS GAN	$\mathcal{L}_D^{LSGAN} = -\mathbb{E}_{x\sim p_d}[(D(x)-1)^2] + \mathbb{E}_{\hat{x}\sim p_g}[D(\hat{x})^2]$	$\mathcal{L}_G^{LSGAN} = -\mathbb{E}_{\hat{x}\sim p_g}[(D(\hat{x}-1)^2]$
DRAGAN	$\mathcal{L}_D^{DRAGAN} = \mathcal{L}_D^{GAN} + \lambda\mathbb{E}_{\hat{x}\sim p_d+\mathcal{N}(0,c)}[(\|\|\nabla D(\hat{x})\|\|_2 - 1)^2]$	$\mathcal{L}_G^{DRAGAN} = -\mathcal{L}_D^{NS\ GAN}$
BEGAN	$\mathcal{L}_D^{BEGAN} = \mathbb{E}_{x\sim p_d}[\|\|x - \mathrm{AE}(x)\|\|_1] - k_t\mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x} - \mathrm{AE}(\hat{x})\|\|_1]$	$\mathcal{L}_G^{BEGAN} = \mathbb{E}_{\hat{x}\sim p_g}[\|\|\hat{x} - \mathrm{AE}(\hat{x})\|\|_1]$

discriminator are clipped to a small absolute value to enforce smoothness. To improve on the stability of the training, Gulrajani et al. [9] instead add a soft constraint on the norm of the gradient which encourages the discriminator to be 1-Lipschitz. The gradient norm is evaluated on points obtained by linear interpolation between data points and generated samples where the optimal discriminator should have unit gradient norm [9].

Gradient norm penalty can also be added to both MM GAN and NS GAN and evaluated around the data manifold (DRAGAN [14] in Table 1 based on NS GAN). This encourages the discriminator to be piecewise linear around the data manifold. Note that the gradient norm can also be evaluated between fake and real points, similarly to WGAN GP, and added to either MM GAN or NS GAN [6].

Mao et al. [16] propose a least-squares loss for the discriminator and show that minimizing the corresponding objective (LS GAN in Table 1) implicitly minimizes the Pearson χ^2 divergence. The idea is to provide smooth loss which saturates slower than the sigmoid cross-entropy loss of the original MM GAN.

Finally, Berthelot et al. [4] propose to use an autoencoder as a discriminator and optimize a lower bound of the Wasserstein distance between auto-encoder *loss distributions* on real and fake data. They introduce an additional hyperparameter γ to control the equilibrium between the generator and discriminator.

4. Challenges of a Fair Comparison

There are several interesting dimensions to this problem, and there is no single *right way* to compare these models (i.e. the loss function used in each GAN). Unfortunately, due to the combinatorial explosion in the number of choices and their ordering, not all relevant options can be explored. While there is *no definite answer* on how to best compare two models, in this work we have made several pragmatic choices which were motivated by two practical concerns: providing a neutral and fair comparison, and a hard limit on the computational budget.

Which metric to use? Comparing models implies access to some metric. As discussed in Section 2, classic measures, such as model likelihood cannot be applied. We will argue for and study two sets of evaluation metrics in Section 5: FID, which can be computed on all data sets, and precision, recall, and F_1, which we can compute for the proposed tasks.

How to compare models? Even when the metric is fixed, a given algorithm can achieve very different scores, when varying the architecture, hyperparameters, random initialization (i.e. random seed for initial network weights), or the data set. Sensible targets include best score across all dimensions (e.g. to claim the best performance on a fixed data set), average or median score (rewarding models which are good in expectation), or even the worst score (rewarding models with worst-case robustness). These choices can even be combined — for example, one might train the model multiple times using the best hyperparameters, and average the score over random initializations).

For each of these dimensions, we took several pragmatic choices to reduce the number of possible configurations, while still exploring the most relevant options.

1. **Architecture**: We use the *same* architecture for all models. The architecture is rich enough to achieve good performance.
2. **Hyperparameters**: For both training hyperparameters (e.g. the learning rate), as well as model specific ones

(e.g. gradient penalty multiplier), there are two valid approaches: (i) perform the hyperparameter optimization for each data set, or (ii) perform the hyperparameter optimization on one data set and *infer* a good range of hyperparameters to use on other data sets. We explore both avenues in Section 6.

3. **Random seed**: Even with everything else being fixed, varying the random seed may have a non-trivial influence on the results. We study this particular effect and report the corresponding confidence intervals.
4. **Data set**: We chose four popular data sets from GAN literature and report results separately for each data set.
5. **Computational budget**: Depending on the budget to optimize the parameters, different algorithms can achieve the best results. We explore how the results vary depending on the budget.

5. Metrics

In this work we focus on two sets of metrics. We first analyze the recently proposed FID in terms of robustness (of the metric itself), and conclude that it has desirable properties and can be used in practice. Nevertheless, this metric, as well as Inception Score, is incapable of detecting overfitting: a *memory GAN* which simply stores all training samples would score perfectly under both measures. Based on these shortcomings, we propose an approximation to precision and recall for GANs and how that it can be used to quantify the degree of overfitting. We stress that the proposed method should be viewed as complementary to IS or FID, rather than a replacement.

5.1. Fréchet Inception Distance

FID was shown to be robust to noise [10]. Here we quantify the bias and variance of FID, its sensitivity to the encoding network and sensitivity to mode dropping. To this end, we partition the data set into two groups, i.e. $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$. Then, we define the data distribution p_d as the empirical distribution on a random subsample of $\mathcal{X}_1$ and the model distribution p_g to be the empirical distribution on a random subsample from $\mathcal{X}_2$. Intuitively, for a random partition, this "model distribution" should follow the data distribution.

Bias and variance. We evaluate the bias and variance of FID on four classic data sets used in the GAN literature. We start by using the default train vs. test partition and compute the FID between the test set (limited to $N = 10000$ samples for CelebA) and the sample of size N from the train set. The sampling from the train set is performed $M = 50$ times. The optimistic estimate of FID are reported in Table 2. We observe that FID has rather high bias, but small variance, From this perspective, estimating the full covariance matrix might be unnecessary and counter-productive, and a constrained version might suffice.

Table 2: Bias and variance of FID. If the data distribution matches the model distribution, FID should evaluate to zero. However, we observe some bias and low variance on samples of size 10000.

DATA SET	AVG. FID	DEV. FID
CELEBA	2.27	0.02
CIFAR10	5.19	0.02
FASHION-MNIST	2.60	0.03
MNIST	1.25	0.02

To test the sensitivity to this initial choice of train vs. test partitioning, we consider 50 random partitions (keeping the relative sizes fixed, i.e. $6:1$ for MNIST) and compute the FID with $M = 1$ sample. We observe results similar to Table 2 which is to be expected if the train and test data sets are drawn from the same distribution.

Detecting mode dropping with FID. To simulate missing modes, we fix a partition of data set $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2$ and we subsample $\mathcal{X}_2$ and keep only samples from the first k classes, increasing k from 1 to 10. For each k, we consider 50 random subsamples from $\mathcal{X}_2$. Figure 1 shows that FID is heavily influenced by the missing modes.

Figure 1: As the sample captures more classes the FID with respect to the reference data set decreases. We observe that FID drastically increases under mode dropping.

Sensitivity to encoding network. Suppose we compute FID using a different network and encoding layer. Would the ranking of models change? To test this we apply VGG trained on ImageNet and consider the layer FC7 of dimension 4096. Figure 2 shows the resulting distribution. We observe high Spearman's rank correlation $\rho = 0.9$ which encourages the use of the default coding layer suggested by the authors. Of course, a natural comparison would be to apply VGG trained on some other data set, which we leave for future work.

5.2. Precision, Recall and F_1 Score

Precision, recall and F_1 score are proven and widely adopted techniques for quantitatively evaluating the quality

Figure 2: The difference between FID score computed on InceptionNet vs FID computed using VGG for the CELEBA data set (for interesting range: FID < 200). We observe high rank correlation (Spearman's $\rho = 0.9$) which encourages the use of the default coding layer suggested by the authors.

of discriminative models. Precision measures the fraction of relevant retrieved instances among the retrieved instances, while recall measures the fraction of the retrieved instances among relevant instances. F_1 score is the harmonic average of precision and recall.

Notice that IS only captures precision: It will not penalize the model for not producing all modes of the data distribution — it will only penalize the model for not producing all *classes*. On the other hand, FID captures both precision and recall. Indeed, a model which fails to recover different modes of the data distribution will suffer in terms of FID.

We propose a simple and effective data set for evaluating (and comparing) generative models. Our main motivation is that the currently used data sets are either too simple (e.g. simple mixtures of Gaussians, or MNIST) or too complex (e.g. ImageNet). We argue that it is critical to be able to increase the complexity of the task in a relatively smooth and controlled fashion. To this end, we present a set of tasks for which we can *approximate* the precision and recall of each model. As a result, we can compare different models based on established metrics.

Manifold of convex polygons. The main idea is to construct a data manifold such that the distances from samples to the manifold can be computed efficiently. As a result, the problem of evaluating the quality of the generative model is effectively transformed into a problem of computing the distance to the manifold. This enables an intuitive approach for defining the quality of the model. Namely, if the samples from the model distribution p_g are (on average) close to the manifold, its *precision* is high. Similarly, high *recall* implies that the generator can recover (i.e. generate something close to) any sample from the manifold.

For general data sets, this reduction is impractical as one has to compute the distance to the manifold which we are trying to learn. However, if we *construct a manifold* such that this distance is efficiently computable, the precision and recall can be efficiently evaluated.

(a) High precision, high recall (b) High precision, low recall

(c) Low precision, high recall (d) Low precision, low recall

Figure 3: Samples from models with (a) high recall and precision, (b) high precision, but low recall (lacking in diversity), (c) low precision, but high recall (can decently reproduce triangles, but fails to capture convexity), and (d) low precision and low recall.

To this end, we propose a set of toy data sets for which such computation can be performed efficiently: The manifold of convex polygons. As the simplest example, let us focus on gray-scale triangles represented as one channel images as in Figure 3. These triangles belong to a low-dimensional manifold $\mathcal{C}_3$ embedded in $\mathbb{R}^{d\times d}$. Intuitively, the coordinate system of this manifold represents the axes of variation (e.g. rotation, translation, minimum angle size, etc.). A good generative model should be able to capture these factors of variation and recover the training samples. Furthermore, it should recover any sample from this manifold from which we can efficiently sample which is illustrated in Figure 3.

Computing the distance to the manifold. Let us consider the simplest case: single-channel gray scale images represented as vectors $x \in \mathbb{R}^{d^2}$. The distance of a sample $\hat{x} \in \mathbb{R}^{d^2}$ to the manifold is defined as the squared Euclidean distance to the closest sample from the manifold $\mathcal{C}_3$, i.e.

$$\min_{x\in\mathcal{C}_3} \ell(x,\hat{x}) = \sum_{i=1}^{d^2} ||x_i - \hat{x}_i||_2^2.$$

Figure 4: How does the minimum FID behave as a function of the budget? The plot shows the distribution of the minimum FID achievable for a fixed budget along with one standard deviation interval. For each budget, we estimate the mean and variance using 5000 bootstrap resamples out of 100 runs. We observe that, given a relatively low budget (say less than 15 hyperparameter settings), all models achieve a similar minimum FID. Furthermore, for a fixed FID, "bad" models can outperform "good" models given enough computational budget. We argue that the computational budget to search over hyperparameters is an important aspect of the comparison between algorithms.

This is a non-convex optimization problem. We find an approximate solution by gradient descent on the vertices of the triangle (more generally, a convex polygon), ensuring that each iterate is a valid triangle (more generally, a convex polygon). To reduce the false-negative rate we repeat the algorithm several times from random initial solutions.

To compute the latent representation of a sample $\hat{x} \in \mathbb{R}^{d \times d}$ we *invert* the generator, i.e. we solve

$$z^\star = \arg\min_{z \in \mathbb{R}^{d_z}} ||\hat{x} - G(z)||_2^2,$$

using gradient descent on z while keeping G fixed [15].

6. Large-scale Experimental Evaluation

We consider two budget-constrained experimental setups whereby in the (i) **wide one-shot setup** one may select 100 samples of hyper-parameters per model, and where the range for each hyperparameter is *wide*, and (ii) the **narrow two-shots setup** where one is allowed to select 50 samples from more narrow ranges which were manually selected by first performing the wide hyperparameter search over a specific data set. For the exact ranges and hyperparameter search details we refer the reader to the Appendix A. In the second set of experiments we evaluate the models based on the "novel" metric: F_1 score on the proposed data set. Finally, we included the Variational Autoencoder [13] in the experiments as a popular alternative.

6.1. Experimental Setup

To ensure a fair comparison, we made the following choices: (i) we use the generator and discriminator architecture from INFO GAN [5] as the resulting function space is rich enough and all considered GANs were not originally designed for this architecture. Furthermore, it is similar to a proven architecture used in DCGAN [20]. The exception is BEGAN where an autoencoder is used as the discriminator. We maintain similar expressive power to INFO GAN by using identical convolutional layers the encoder and approximately matching the total number of parameters.

For all experiments we fix the latent code size to 64 and the prior distribution over the latent space to be uniform on $[-1, 1]^{64}$. We choose Adam [12] as the optimization algorithm as it was the most popular choice in the GAN literature. We apply the same learning rate for both generator and discriminator. We set the batch size to 64 and perform optimization for 20 epochs on MNIST and FASHION MNIST, 40 on CELEBA and 100 on CIFAR[2].

Finally, we allow for recent suggestions, such as batch normalization in the discriminator, and imbalanced update frequencies of generator and discriminator. We explore these possibilities, together with learning rate, parameter β_1 for ADAM, and hyperparameters of each model. We report the hyperparameter ranges and other details in Appendix A.

6.2. A Large Hyperparameter Search

We perform hyperparameter optimization and, for each run, look for the *best* FID across the training run (simulating early stopping). To choose the *best* model, every 5 epochs we compute the FID between the 10k samples generated by the model and the 10k samples from the test set. We have performed this computationally expensive search for each data set. We present the sensitivity of models to the hyperparameters in Figure 5 and the best FID achieved by each model in Table 3.

Critically, we consider the *mean* FID as the computational budget increases which is shown in Figure 4. There are three important observations. Firstly, there is no algorithm which clearly dominates others. Secondly, for an interesting range of FIDs, a "bad" model trained on a large budget can out perform a "good" model trained on a small

[2]Those four data sets are a popular choice for generative modeling. They are of simple to medium complexity, making it possible to run many experiments as well as getting decent results.

Figure 5: A *wide range* hyperparameter search (100 hyperparameter samples per model). We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than others.

Table 3: Best FID obtained in a large-scale hyperparameter search for each data set. The scores were computed in two phases: first, we run a large-scale search on a wide range of hyperparameters, and select the best model. Then, we re-run the training of the selected model 50 times with different initialization seeds, to estimate the stability of the training and report the mean FID and standard deviation, excluding outliers. The asterisk (*) on some combinations of models and data sets indicates the presence of significant outlier runs, usually severe mode collapses or training failures (** indicates up to 20% failures). We observe that the performance of each model heavily depends on the data set and no model strictly dominates the others. We note that VAE is heavily penalized due to the blurriness of the generated images. Note that these results are **not "state-of-the-art"**: (i) larger architectures could improve all models, (ii) authors often report the best FID which opens the door for random seed optimization.

	MNIST	FASHION	CIFAR	CELEBA
MM GAN	10.0 ± 0.9	25.6 ± 0.9	70.6 ± 2.4	68.0 ± 2.7
NS GAN	6.7 ± 0.3	26.6 ± 1.5	58.6 ± 2.1	58.0 ± 2.7
LSGAN	$8.5 \pm 0.8^{**}$	31.2 ± 4.0	67.1 ± 2.9	$53.6 \pm 4.2^{*}$
WGAN	6.8 ± 0.4	18.0 ± 1.1	55.9 ± 2.8	42.9 ± 1.8
WGAN GP	$8.9 \pm 0.8^{*}$	20.6 ± 1.3	52.9 ± 1.3	26.8 ± 1.2
DRAGAN	7.7 ± 0.3	26.0 ± 1.2	68.5 ± 1.6	41.4 ± 3.3
BEGAN	12.3 ± 0.9	33.2 ± 1.3	$71.4 \pm 1.1^{*}$	38.1 ± 1.1
VAE	40.1 ± 0.7	100.9 ± 3.0	$168.5 \pm 11.5^{*}$	93.2 ± 4.3

budget. Finally, when the budget is limited, any statistically significant comparison of the models is unattainable.

6.3. Impact of Limited Computational Budget

In some cases, the computational budget available to a practitioner is too small to perform such a large-scale hyperparameter search. Instead, one can tune the range of hyperparameters on one data set and interpolate the good hyperparameter ranges for other data sets. We now consider this setting in which we allow only 50 samples from a set of narrow ranges, which were selected based on the wide hyperparameter search on the FASHION-MNIST data set. We report the narrow hyperparameter ranges in Appendix A.

Figure 15 shows the variance of FID per model, where the hyperparameters were selected from narrow ranges. From the practical point of view, there are significant differences between the models: in some cases the hyperparameter ranges *transfer* from one data set to the others (e.g. NS GAN), while others are more sensitive to this choice (e.g. WGAN). We note that better scores can be obtained by a wider hyperparameter search. These results supports the conclusion that discussing the *best* score obtained by a model on a data set is not a meaningful way to discern between these models. One should instead discuss the distribution of the obtained scores.

6.4. Robustness to Random Initialization

For a fixed model, hyperparameters, training algorithm, and the order that the data is presented to the model, one would expect similar model performance. To test this hypothesis we re-train the best models from the limited hyperparameter range considered for the previous section, while changing the initial weights of the generator and discriminator networks (i.e. by varying a random seed). Table 3 and Figure 16 show the results for each data set. Most models are relatively robust to random initialization, except LSGAN, even though for all of them the variance is significant and should be taken into account when comparing models.

6.5. Precision, recall, and F_1

We perform a search over the wide range of hyperparameters and compute precision and recall by considering $n = 1024$ samples. In particular, we compute the precision of the model by computing the fraction of generated samples with distance below a threshold $\delta = 0.75$. We then consider n samples from the test set and invert each sample x to compute $z^\star = G^{-1}(x)$ and compute the squared Euclidean distance between x and $G(z^\star)$. We define the recall as the fraction of samples with squared Euclidean distance below δ. Figure 6 shows the results where we select the best F_1 score for a fixed model and hyperparameters and vary the budget. We observe that even for this seemingly simple task, many models struggle to achieve a high F_1 score.

Figure 6: How does F_1 score vary with computational budget? The plot shows the distribution of the maximum F_1 score achievable for a fixed budget with a 95% confidence interval. For each budget, we estimate the mean and confidence interval (of the mean) using 5000 bootstrap resamples out of 100 runs. When optimizing for F_1 score, both NS GAN and WGAN enjoy high precision and recall. The underwhelming performance of BEGAN and VAE on this particular data set merits further investigation.

Analogous plots where we instead maximize precision or recall for various thresholds are presented in Appendix E.

7. Conclusion & Open Problems

In this paper we have started a discussion on how to neutrally and fairly compare GANs. We focus on two sets of evaluation metrics: (i) The Fréchet Inception Distance, and (ii) precision, recall and F_1. We provide empirical evidence that FID is a reasonable metric due to its robustness with respect to mode dropping and encoding network choices.

Comparison based on FID. Our main insight is that to compare models it is meaningless to report the *minimum* FID achieved. Instead, distributions of the FID for a fixed computational budget should be compared. Indeed, empirical evidence presented herein imply that algorithmic differences in state-of-the-art GANs become less relevant, as the computational budget increases. Furthermore, given a limited budget (say a month of compute-time), a "good" algorithm might be outperformed by a "bad" algorithm.

Comparison based on precision, recall and F_1 score. Our simple triangle data set allows us to compute well understood precision and recall metrics, and consequently the F_1 score. We observe that even for this seemingly simple task, many models struggle to achieve a high F_1 score. When optimizing for F_1 score both NS GAN and WGAN enjoy both high precision and recall. Other models, such as DRAGAN and WGAN GP fail to reach high recall values. Finally, we observe that it is possible to achieve high precision and high recall on this task (cf. Appendix E).

Comparison with respect to original GAN. While many algorithms have claimed superiority over the original GAN model [8], we found no empirical evidence which supports such claims, across all data sets. In fact, the NS GAN performs on par with most other models and achieves the best overall FID on MNIST. Furthermore, it outperforms other models in terms of the F_1 score on TRIANGLES.

Open problems. It remains to be examined whether FID is stable under a more radical change of the encoding, e.g using a network trained on a different task. Also, FID cannot detect overfitting to the training data set, and an algorithm that just remembers all the training examples would perform very well. Finally, FID can probably be "fooled" by artifacts that are not detected by the embedding network.

The triangles data set can be made progressively more complex by: (i) introducing multiple convex polygons at once, (ii) providing color or texture inside the polygon, and (iii) gradually increasing the resolution. While the performance of existing models might be improved given a bigger computational budget and larger model capacity, we argue that algorithmic improvements should drive better performance. Having such a series of tasks of increasing complexity should greatly benefit the research community.

As discussed in Section 4, many dimensions have to be taken into account when comparing different models, and this work only explores a subset of the options. We cannot exclude the possibility that that some models significantly outperform others under currently unexplored conditions.

Finally, this work strongly suggest that future GAN research should be more experimentally systematic and models should be compared on a neutral ground.

Acknowledgements

We would like to acknowledge Tomas Angles for advocating convex polygons as a benchmark data set. We would like to thank Ian Goodfellow, Michaela Rosca, Ishaan Gulrajani, David Berthelot, and Xiaohua Zhai for useful discussions and remarks.

References

[1] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017. 1, 2

[2] Sanjeev Arora, Rong Ge, Yingyu Liang, Tengyu Ma, and Yi Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning (ICML)*, 2017. 2

[3] Philip Bachman and Doina Precup. Variational generative stochastic networks with collaborative shaping. In *International Conference on Machine Learning (ICML)*, 2015. 2

[4] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017. 1, 3

[5] Xi Chen, 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 (NIPS)*, 2016. 6

[6] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. *arXiv preprint arXiv:1710.08446*, 2017. 2, 3

[7] Holly E Gerhard, Felix A Wichmann, and Matthias Bethge. How sensitive is the human visual system to the local statistics of natural images? *PLoS computational biology*, 9(1), 2013. 1

[8] 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 (NIPS)*, 2014. 1, 2, 8

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017. 1, 3

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a Nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017. 1, 2, 4

[11] Ferenc Huszár. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *arXiv preprint arXiv:1511.05101*, 2015. 2

[12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 6

[14] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*, 2017. 3

[15] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 6

[16] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint ArXiv:1611.04076*, 2016. 3

[17] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of GANs. *arXiv preprint arXiv:1705.10461*, 2017. 2

[18] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[19] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Machine Learning (ICML)*, 2017. 1

[20] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 1, 6

[21] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2

[22] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015. 2

[23] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016. 2

[24] 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 arXiv:1612.03242*, 2016. 1

A. Wide and narrow hyperparameter ranges

The *wide* and *narrow* ranges of hyper-parameters are presented in Table 4 and Table 5 respectively. In both tables, U(a, b) means that the variable was sample uniformly from the range $[a, b]$. The L(a, b) means that that the variable was sampled on a log-scale, that is $x\ L(a,b) \iff x\ 10^{U(log(a),log(b))}$. The parameters used in the search:

- β_1: the parameter of the Adam optimization algorithm.
- Learning rate: generator/discriminator learning rate.
- λ: Multiplier of the gradient penalty for DRAGAN and WGAN GP. Learning rate for k_t in BEGAN.
- Disc iters: Number of discriminator updates per one generator update.
- batchnorm: If True, the batch normalization will be used in the discriminator.
- γ: Parameter of BEGAN.
- clipping: Parameter of WGAN, weights will be clipped to this value.

Table 4: Wide ranges of hyper-parameters used for the large-scale search. "U" denotes uniform sampling, "L" sampling on a log-scale.

	MM GAN	NS GAN	LSGAN	WGAN	WGAN GP	DRAGAN	BEGAN	VAE
Adam's β_1	U(0, 1)							
learning rate	L($10^{-5}, 10^{-2}$)							
λ	N/A	N/A	N/A	N/A	L($10^{-1}, 10^{2}$)	L($10^{-1}, 10^{2}$)	L($10^{-4}, 10^{-2}$)	N/A
disc iter	Either 1 or 5, sampled with the same probablity							
batchnorm	True or False, sampled with the same probability							
γ	N/A	N/A	N/A	N/A	N/A	N/A	U(0, 1)	N/A
clipping	N/A	N/A	N/A	L($10^{-3}, 10^{0}$)	N/A	N/A	N/A	N/A

Table 5: Narrow ranges of hyper-parameters used in the search with 50 samples per model. The ranges were optimized by looking at the wide search results for fashion-mnist data set. "U" denotes uniform sampling, "L" sampling on a log-scale.

	MM GAN	NS GAN	LSGAN	WGAN	WGAN GP	DRAGAN	BEGAN	VAE
Adam's β_1	Always 0.5							
learning rate	L($10^{-4}, 10^{-3}$)							
λ	N/A	N/A	N/A	N/A	L($10^{-1}, 10^{1}$)	L($10^{-1}, 10^{1}$)	L($10^{-4}, 10^{-2}$)	N/A
disc iter	Always 1							
batchnorm	True/False	True/False	True/False	True/False	False	False	True/False	True/False
γ	N/A	N/A	N/A	N/A	N/A	N/A	U(0.6, 0.9)	N/A
clipping	N/A	N/A	N/A	L($10^{-2}, 10^{0}$)	N/A	N/A	N/A	N/A

B. Which parameters really matter?

Figure 7, Figure 8, Figure 9 and Figure 10 present scatter plots for data sets FASHION MNIST, MNIST, CIFAR, CELEBA respectively. For each model and hyper-parameter we estimate its impact on the final FID. Figure 7 was used to select narrow ranges of hyper-parameters.

FASHION-MNIST (wide range)

Figure 7: Wide range scatter plots for FASHION MNIST. For each algorithm (column) and each parameter (row), the corresponding scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. each parameter. Those results have been used to choose the *narrow range* in Table 5. For example, Adam's β_1 does not seem to significantly impact any algorithm, so for the narrow range, we fix its value to always be 0.5. Likewise, we fix the number of discriminator iterations to be always 1. For other parameters, the selected range is smaller (e.g. learning rate), or can differ for each algorithm (e.g. batch norm).

MNIST (wide range)

Figure 8: Wide range scatter plots for MNIST. For each algorithm (column) and each parameter (row), the corresponding scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. each parameter.

CIFAR10 (wide range)

Figure 9: Wide range scatter plots for CIFAR10. For each algorithm (column) and each parameter (row), the corresponding scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. each parameter.

CELEBA (wide range)

Figure 10: Wide range scatter plots for CELEBA. For each algorithm (column) and each parameter (row), the corresponding scatter plot shows the FID in function of the parameter. This illustrates the sensitivity of each algorithm w.r.t. each parameter.

C. Fréchet Inception Distance and Image Quality

It is interesting to see how the FID translates to the image quality. In Figure 11, Figure 12, Figure 13 and Figure 14, we present, for every model, the distribution of FIDs and the corresponding samples.

Figure 11: MNIST: Distribution of FIDs and corresponding samples for each model when sampling parameters from *wide* ranges.

Figure 12: FASHION-MNIST: Distribution of FIDs and corresponding samples for each model when sampling parameters from *wide* ranges.

Figure 13: CIFAR10: Distribution of FIDs and corresponding samples for each model when sampling parameters from *wide* ranges.

Figure 14: CELEBA: Distribution of FIDs and corresponding samples for each model when sampling parameters from *wide* ranges.

D. Hyper-parameter Search over Narrow Ranges

In Figure 5 we presented the sensitivity of GANs to hyperparameters, assuming the samples are taken from the wide ranges (see Table 4). For completeness, in Figure 15 we present a similar comparison for the narrow ranges of hyperparameters (presented in Table 5).

Figure 15: A *narrow range* search of hyperparameters which were selected based on the wide hyperparameter search on the FASHION-MNIST data set. For each model we allow 50 hyperparameter samples. From the practical point of view, there are significant differences between the models: in some cases the hyperparameter ranges *transfer* from one data set to the others (e.g. NS GAN), while others are more sensitive to this choice (e.g. WGAN). We note that better scores can be obtained by a wider hyperparameter search.

Figure 16: For each model we search for best hyperparameters on the *wide range*. Then, we retrain each model using the best parameters 50 times with random initialization of the weights, keeping everything else fixed. We observe a slight variance in the final FID. Hence, when an FID is reported it is paramount that one compares the entire distribution, instead of the *best* seed for the best run. The figure corresponds to Table 3.

E. Precision, Recall and F_1 as a Function of the Budget

Figure 17: Optimizing for F_1, threshold $\delta = 1.0$.

Figure 18: Optimizing for F_1, threshold $\delta = 0.5$.

Figure 19: Optimizing for precision, threshold $\delta = 1.0$.

Figure 20: Optimizing for precision, threshold $\delta = 0.5$.

Figure 21: Optimizing for recall, threshold $\delta = 1.0$.

Figure 22: Optimizing for recall, threshold $\delta = 0.5$.