This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

On Aliased Resizing and Surprising Subtleties in GAN Evaluation

Gaurav Parmar[1] Richard Zhang[2] Jun-Yan Zhu[1]
[1]Carnegie Mellon University [2]Adobe Research

Abstract

Metrics for evaluating generative models aim to measure the discrepancy between real and generated images. The often-used Fréchet Inception Distance (FID) metric, for example, extracts "high-level" features using a deep network from the two sets. However, we find that the differences in "low-level" preprocessing, specifically image resizing and compression, can induce large variations and have unforeseen consequences. For instance, when resizing an image, e.g., with a bilinear or bicubic kernel, signal processing principles mandate adjusting prefilter width depending on the downsampling factor, to antialias to the appropriate bandwidth. However, commonly-used implementations use a fixed-width prefilter, resulting in aliasing artifacts. Such aliasing leads to corruptions in the feature extraction downstream. Next, lossy compression, such as JPEG, is commonly used to reduce the file size of an image. Although designed to minimally degrade the perceptual quality of an image, the operation also produces variations downstream. Furthermore, we show that if compression is used on real training images, FID can actually improve if the generated images are also subsequently compressed. This paper shows that choices in low-level image processing have been an underappreciated aspect of generative modeling. We identify and characterize variations in generative modeling development pipelines, provide recommendations based on signal processing principles, and release a reference implementation to facilitate future comparisons.

1. Introduction

With the proliferation of generative modeling techniques, such as Generative Adversarial Networks (GANs) [24], accurately discerning which methods are performing better has become a critical aspect of the field. For visual data, metrics such as Inception Score (IS) [59], Kernel Inception Distance (KID) [4], and the ubiquitously-used Fréchet Inception Distance (FID) [26] have become standard practice for developing and adopting models. Under the hood, these methods evaluate the discrepancy between generated and natural images, in a deep feature space, to capture relevant features of the two distributions. After all, at its core, generative modeling involves learning and mimicking *high-order, complex* statistics of visual data.

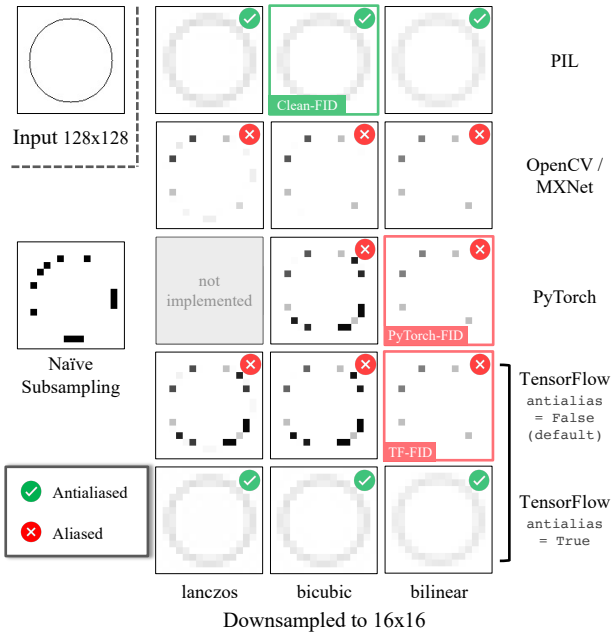

Figure 1. **Downsampling a circle.** We resize an input image (left) by a factor of 8, using different image processing libraries. The Lanczos, bicubic, and bilinear implementations by PIL (top row) adjust the antialiasing filter width by the downsampling factor (marked as ✓). Other implementations (including those used for PyTorch-FID and TensorFlow-FID) use fixed filter widths, introducing aliasing artifacts (marked as ✗) and resemble naive nearest subsampling. Aliasing artifacts induce inconsistencies in the calculation of downstream metrics such as Fréchet Inception Distance [26], KID [4], IS [59], and PPL [33]. Note that `antialias` flag is available in TensorFlow 2, but is set to `False` (default value) for the FID calculation.

However, we find that *low-level, seemingly innocuous operations*, can induce surprisingly large discrepancies in high-level statistics. For example, consider Figure 1. Given the same input image, different image processing libraries produce drastically different results. Specifically, the implementations using OpenCV, TensorFlow and PyTorch libraries with default flags, contain severe aliasing artifacts. Similarly, the simple act of saving images as JPEG with the

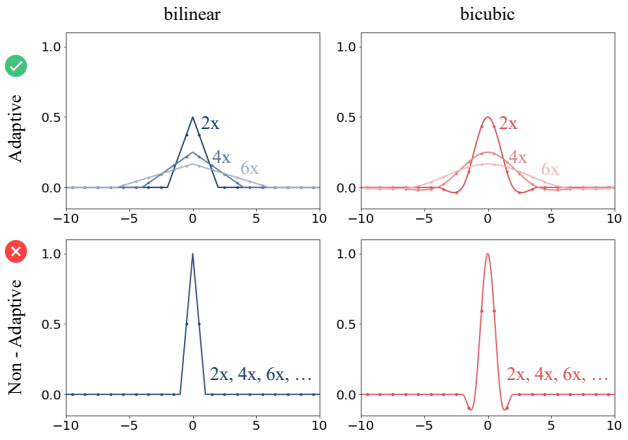

Figure 2. **Interpolation Filters.** We show adaptive filters by PIL (top) and non-adaptive filter from PyTorch (bottom). The FID implementations in PyTorch and TensorFlow use a *fixed-width* bilinear interpolation, independent of resizing ratio. In contrast, the proposed Clean-FID uses an implementation that follows standard signal processing principles and adaptively stretches the filter to prevent aliasing. The horizontal axes represent the spatial coordinates and the vertical axes represents the kernel intensity.

default parameters, either when building the training dataset or collection of generated images, adds quantization and low-level statistical differences to the underlying data. The differences induced cause meaningful variations when used for evaluation protocols. As the Fréchet Inception Distance (FID) metric [26] is the most ubiquitous [6,26,33,36,56], it is the focus of our experiments. We offer a standard benchmark, *clean-fid* (`github.com/GaParmar/clean-fid`), and concrete suggestions on resizing and quantization procedures to enable clean comparisons in future evaluation protocols.

First, we investigate the implications of image resizing. When downsampling, signal processing techniques recommend "prefiltering" the input, to prevent high-frequency elements from aliasing into the output. When the downsampling factor is larger, the prefilter kernel should be correspondingly stretched. However, as shown in Figure 2, the resizing function used by the FID implementations in TensorFlow and PyTorch *do not* prefilter the image, resulting in aliasing artifacts shown in Figure 1. Resizing can occur in two locations – during data preprocessing (training with lower resolution) or at evaluation time (resizing to 299 resolution to compute the FID metric). In both cases, inconsistent resizing functions induce variations downstream. If used for data preprocessing, the training data distribution itself is changed. When used for the evaluation metric, small variations in resizing can cause changes in subsequent feature extraction. We quantify the effects of these inconsistencies and offer standard recommendations. Specifically, we propose to use a stronger bicubic filter [35]; more importantly, we propose to adjust prefiltering width based on the resizing factors, as guided by signal processing principles.

Secondly, we investigate the implication of image compression. While the JPEG protocol is a lossy compression scheme, designed to preserve perceptual similarity to the original [67], it can perturb an image enough to corrupt downstream feature extraction. This affects performance drastically and can create mismatches when comparing methods. Perhaps more surprisingly, when training images are saved with JPEG compression, modern GANs are unable to fully mimic the induced artifacts, and large FID improvements can actually be artificially achieved by tweaking the JPEG compression ratios when storing the generated images. We quantify the surprising effects of this compression operation, and again offer a concrete, standardized protocol to avoid inconsistencies and hindrances to proper evaluation.

In conclusion, we characterize the surprising importance of low-level image processing steps, resizing and quantization, when training and evaluating generative models, such as GANs. We focus our experiments on the widely adopted FID metric, and show additional results on the KID metric [4] as well as IS [59] and Perceptual Path Length (PPL) metrics [33] (in the supplement). Importantly, *any* metric, present or future, that derives statistics from images undergoing these processing steps, will be affected by these factors.

2. Related Work

Deep generative models. A wide range of image and video synthesis applications [41, 50, 61, 77] have been enabled, as a result of tremendous progress in deep generative models such as GANs [7, 24, 30, 33, 54], VAEs [37, 51, 56], autoregressive models [48], flow-based models [14, 36], and energy-based models [17, 46, 58]. It is often relatively easier to evaluate individual model's performance on downstream tasks, as they have a clear target for a given input. However, evaluating unconditional generative models remains an open problem. It is still an important goal, as most generative models are not tailored to any downstream task.

Evaluating generative models. The community has introduced many evaluation protocols. One idea is to conduct user studies on cloud-sourcing platforms for either assessing the samples' image quality [13, 59, 76] or identifying duplicate images [1]. Due to the subtle differences in user study protocols (e.g., UI design, fees, date/time), it is not easy to replicate results across different papers. Large-scale user studies can also be expensive, prohibiting its usage when evaluating hundreds of model variants and checkpoints during the development stage. Several methods propose evaluating generative models from a self-supervised feature learning perspective, by repurposing the learned discriminators [54] or accompanying encoders [15] for a downstream classification task. However, the representation power of the discriminator or encoder does not directly reflect the generators' sample quality and diversity. In addition, not every generative model is trained with a discriminator or encoder.

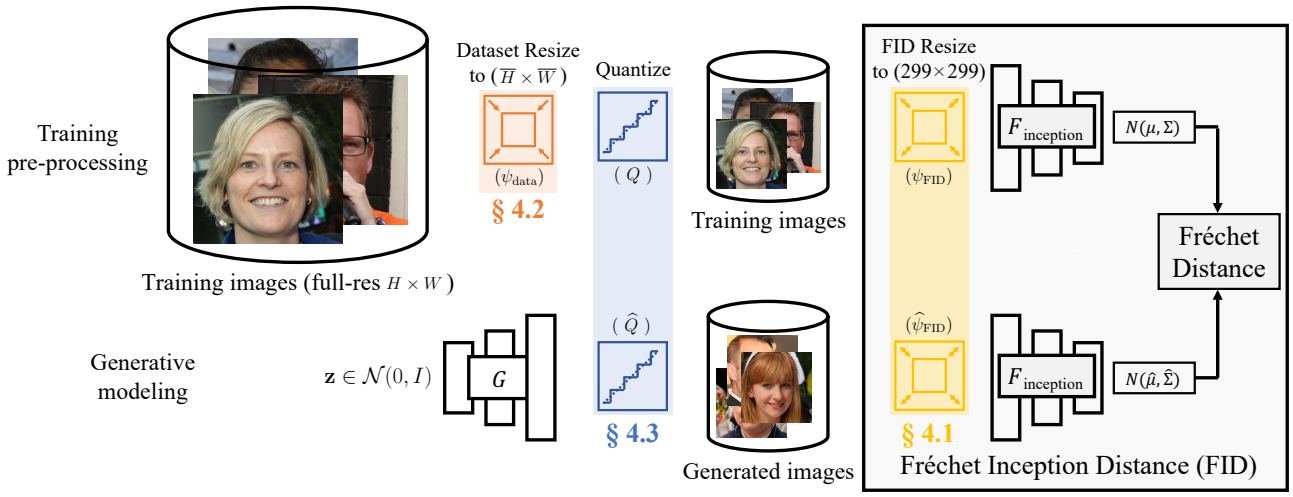

Figure 3. **Overview of the steps involved in FID.** Generative modeling and evaluation involve subtleties in image pre-processing. **Top**: First, the dataset may be downsampled before training (e.g., 1024→256 for FFHQ), requiring a resize (ψ_{data}) and possibly compression (Q). **Bottom**: Generated images may be saved as an unsigned 8-bit integer, resulting in a quantization and possible further compression ($\widehat{Q}$). FID aims to measure how well a generative model $G(z)$ mimics the training distribution. The calculation resizes real and generated images to 299 resolution (ψ_{FID} and $\widehat{\psi}_{\text{FID}}$, respectively), extracts deep features using the Inception network [63], fits Gaussians, and takes the Fréchet distance between two distributions. We study the effects of resizing the training images ψ_{data} in Section 4.3, resizing to 299×299 ψ_{FID} and $\widehat{\psi}_{\text{FID}}$ in Section 4.1 and the quantizations/compressions $\widehat{Q}$ and Q in Section 4.2.

To overcome the previous issues, an area of focus is developing automatic metrics that directly assess the samples of generative models. Various metrics been proposed, criticized, and modified. Commonly-used ones include log-likelihood [24, 37], density estimate with Parzen window [24], Inception Score [59], Perceptual Path Length [33], Fréchet Inception Distance (FID) [26], Classification Accuracy Score and its early variants [55, 59], Classifier Two-sample Tests [40, 43], precision and recall [38, 57], Kernel Inception Distance (KID) [4], among others. Each metric has associated pros and cons [5, 66] and none are perfect.

Among them, Fréchet Inception Distance (FID) has become the most widely-used metrics, as it can model intra-class diversity better than Inception Score. FID is also easy and fast to compute without training additional classifiers [55], and has been shown to be consistent with human perception [26]. As a result, it has been used in recent GANs papers [7, 33, 71] as well as large-scale evaluation study [44], despite facing criticism about the fact that FID is a biased estimator and sensitive to the number of samples used in the evaluation [4, 10]. Our goal here is *not* to study which one is a better metric. Instead, we focus our study on the popular FID metric. Note that the resizing and quantization we study apply to any evaluation metric that contains such operations.

Antialiasing and robustness. The study of resampling signals is central in signal processing [49], image processing [23], and computer graphics [19]. In particular, when downsampling a signal, one must consider the Nyquist sampling criterion [47] and antialias to prevent high-frequency information from aliasing into the output. Without proper antialiasing, in the worst case, an adversary can embed a completely different image in the original, resulting in a "scaling attack" [53, 69]. In convolutional network design, antialiasing has taken form in average pooling [39] and Gaussian filtering [45]. While it was replaced by operations such as max-pooling, based on empirical performance [60], recent works have demonstrated that antialiasing can be compatible and improve performance in convolutional networks [73, 79], transformers [52], NeRFs [3], and GANs [32]. Despite these advances, generative methods continue to be detectable [8, 68], and discriminative networks continue to be sensitive to small perturbations, such as shifts [2, 18] and JPEG compression [25]. Achieving robustness to such perturbations remains an open problem [65], and the preprocessing steps, such as image resizing, used before feature extraction remain consequential. We study the effect of such steps in a generative modeling pipeline and propose a standardization following signal processing principles, in order to facilitate easy and fair comparisons.

3. Preliminaries

In this section, we discuss several low-level image processing steps using different popular libraries, and show that these can have a large effect on the FID score being computed. Figure 3 details the step-by-step process for both dataset preparation and model evaluations.

3.1. Generative Modeling and Evaluation Pipeline

The Fréchet Inception Distance (FID) score aims to measure the gap between two data distributions [26], such as

between a training set and samples from a generator.

Dataset pre-processing. We denote the original real image distribution as $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$, where $x \in \mathbb{Z}^{H \times W \times 3}$. Note that images are saved as 8-bit integers, represented by $\mathbb{Z}$. Training and developing large-scale GANs at the original resolution [7, 33] is often prohibitively expensive, sometimes requiring training hundreds of models during development. As such, developing on lower-resolution versions of the original dataset is a common practice [42, 72, 75], such as 1024→256 on FFHQ or 256 → 128 on ImageNet.

As shown in the top branch of Figure 3, to prepare a lower-resolution training set, one must downsample the training set, denoted by ψ_{data}. Note that downsampling requires an antialiasing step according to standard textbooks [19, 49, 64] that converts integers into a floating point number, $\mathbb{Z} \rightarrow \mathbb{R}$. A quantization step is added afterwards to cast back to $\mathbb{Z}$. This data preparation step introduces a new data distribution of low-res real images: $\overline{\mathbf{x}} \sim p_{\text{data}}(\overline{\mathbf{x}})$, where $\overline{\mathbf{x}} \in \mathbb{Z}^{\overline{H} \times \overline{W} \times 3}$.

Evaluating a generator with FID. A generator G that learns to map a latent code $\mathbf{z} \in \mathcal{N}(0, I)$ to output images $G(\mathbf{z}) \in \mathbb{R}^{\overline{H} \times \overline{W} \times 3}$ is trained on the lower resolution dataset. A common evaluation method is passing both real and generated images through a feature extractor $\mathcal{F}$, fitting a Gaussian distribution, and measuring the Fréchet distance between the two distributions. Deep network activations are used as the statistics of interest, as they have been shown to correspond well with human perceptual judgments [74] and are often used as training objectives [16, 22, 29]. The feature extractor $\mathcal{F}$ used for this task is an InceptionV3 model [63]. Because this model is trained on $299 \times 299 \times 3$ ImageNet image crops [12], the training and generated images are resized denoted by functions ψ_{FID} and $\widehat{\psi}_{\text{FID}}$, respectively, before being processed. As these images may be saved in development pipelines, different image compressions may be applied. These operations are represented by Q for reference images $\mathbf{x}$ and by $\widehat{Q}$ for synthesized images $G(\mathbf{z})$.

$$\mathbf{f} = \mathcal{F}(\psi_{\text{FID}}(Q(\psi_{\text{data}}(\mathbf{x})))), \tag{1}$$

$$\hat{\mathbf{f}} = \mathcal{F}(\widehat{\psi}_{\text{FID}}(\widehat{Q}(G(\mathbf{z})))). \tag{2}$$

After the images are appropriately resized, and the features are extracted, the mean (μ, $\hat{\mu}$) and covariance matrix (Σ, $\widehat{\Sigma}$) of the corresponding set of features $\mathbf{f}$ and $\hat{\mathbf{f}}$ are used to compute the Fréchet distance shown in the equation below.

$$\text{FID} = ||\mu - \hat{\mu}||_2^2 + \text{Tr}(\Sigma + \widehat{\Sigma} - 2(\Sigma\widehat{\Sigma})^{1/2}), \tag{3}$$

The Tr operation calculates the trace of the matrix.The different choices for the resizing ($\psi_{\text{data}}, \psi_{\text{FID}}, \widehat{\psi}_{\text{FID}}$) and quantization ($Q, \widehat{Q}$) add potential sources of inconsistencies.

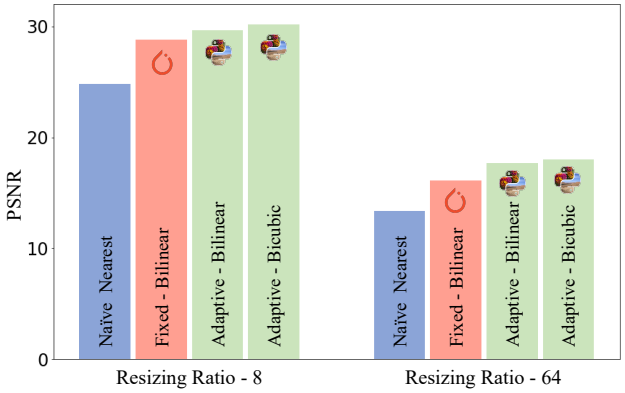

Figure 4. **Reconstruction after downsampling and upsampling.** To illustrate the differences between resizing functions, we downsample images with the different functions and upsample with PIL-Lanczos, and compute similarity to the original with PSNR. The implementation that adjusts prefilter size (PIL) reconstructs the original more accurately than the implementations that used a fixed filter size (PyTorch). This is especially apparent for larger resizing ratios (64×), where performance is closer to naive nearest.

3.2. Image Resizing

Depending on the dataset and training size, the resizing operations (ψ_{FID}, $\widehat{\psi}_{\text{FID}}$) in Figure 3 can either be downsampling or upsampling. Downsampling is the primary focus of this investigation, as it involves *throwing away information*. Methods for downsampling is a common study in the fields of signal and image processing [23, 49].

Antialiasing by prefiltering. The most naive approach is to simply subsample (taking every N^{th} element if performing downsampling by an integer factor N), sometimes referred to as *nearest*. This corresponds to filtering the input image with Kronecker delta function, as only a single value is drawn. Such an approach leads to aliasing, as high-frequency elements of the input alias to the output.

A central principle in signal processing, graphics, and vision [20, 21, 23, 49, 64] is to blur or "prefilter" before subsampling, as a means of removing high-frequency information (thus preventing its misrepresentation downstream). For linear filters, this corresponds to a "depth-wise convolution", using deep learning parlance [27, 62]. We explain two important ways in which prefiltering implementations can vary.

Filter size adaptation to downsampling factor. First, according to signal processing principles, the size of the filter *should* be adjusted, in accordance with the downsampling factor. Widening the low-pass filter in the spatial domain corresponds to reducing its bandwidth and filtering more aggressively in frequency space. As a larger downsampling factor means a lower bandwidth can be represented on the output signal, *widening the filter accordingly is necessary to prevent aliasing*. However, in many common implementations, this is not implemented (or is not used by default); instead, a filter of *fixed, non-adaptive* size is used.

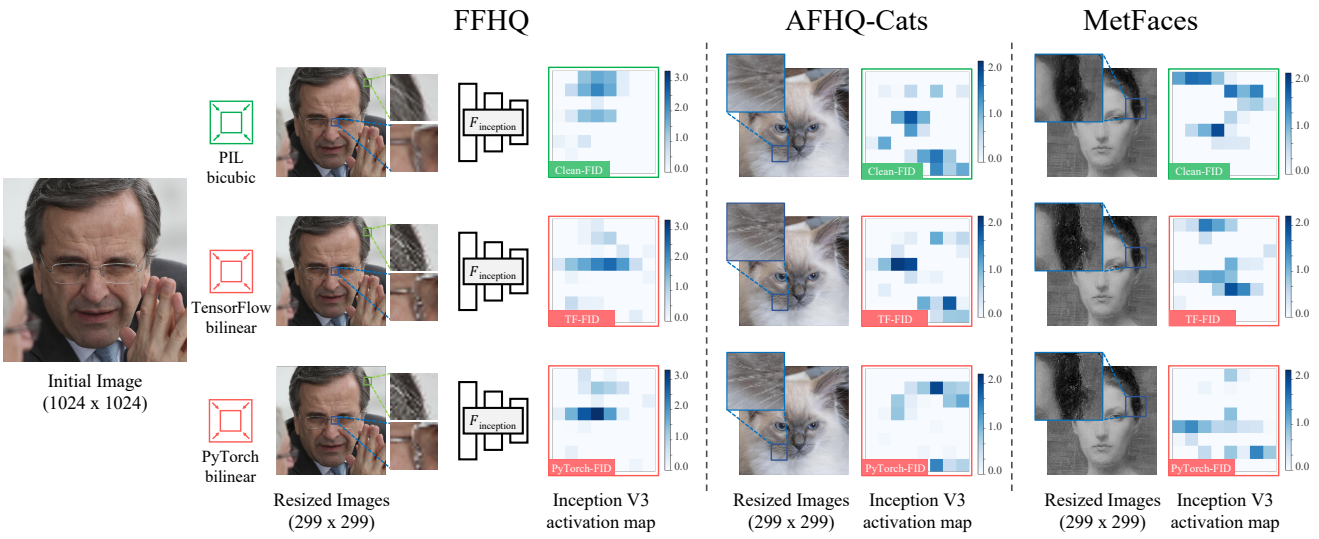

Figure 5. **Differences in Inception features induced by inconsistent resizing.** We resize full resolution 1024×1024 FFHQ [33] image (left) to 299×299 using PIL-bicubic (top), TensorFlow-bilinear (used by TF-FID) (middle), and PyTorch-bilinear (used by PyTorch-FID) (bottom). The resizing functions using current FID implementations (middle and bottom rows) introduce artifacts; for example, the hair and glasses appear noisier and aliased, as compared to the top row. We observe similar behavior on other commonly-used datasets - AFHQ-Cats (512×512) and MetFaces (1024×1024). Furthermore, these resizing implementations are inconsistent with each other, inducing different activation maps when passed through the Inception-V3 network [63]. We propose to resolve this inconsistency and also reduce the aliasing, by standardizing bicubic downsampling as the preprocessing function for a "Clean-FID" (using filtering that adjusts to the downsampling factor, adhering to signal processing principles).

Choice of filters. Secondly, there is a choice of different convolutional filters. The idealized low-pass filter is a sinc, requiring infinite support. As such, approximate filters with different subtle tradeoffs in runtime and behavior are used instead. The *box*, also known as *area* filter, corresponds to a rectangular filter, computing the average of values within a neighborhood. The *bilinear* filter is a triangular filter, *bicubic* [35] is a stronger cubic function, and the *lanczos* filter is an enveloped sinc. All perform a weighted average and have stronger antialiasing, closer to the idealized sinc.

Practical implications of implementation variations. We investigate inconsistencies that arise, when these two factors are varied, and show a toy example in Figure 1. While the choice of filter is largely constant across libraries (lanczos, bicubic, bilinear are shown in each column), *the choice of whether the filter adapts to the downsampling factor is not*. While the PIL library adapts the filter (top row), other libraries do not by default, leading to aliased results. In particular, FID implementations of TensorFlow-FID and PyTorch-FID, use bilinear downsampling implementations that exhibit aliasing, and thus are the focus of our study.

An implication of aliasing is a suboptimal representation of the original image. In Figure 4, we show the result of downsampling and upsampling 300 FFHQ images, and comparing it to the original with PSNR. The methods with fixed filters achieve worse reconstruction than a method that adapts the filter. This effect is accentuated by larger downsampling factors, where high-frequency aliasing dominates when using non-adaptive filters. Figure 5 shows how the Inception features are affected by aliased resizing functions.

Recommendation. Above, we have established that the implementations of FID are inconsistent and aliased. Ideally, the community can (a) use a consistent pipeline to facilitate fair comparisons across papers, and (b) follow signal processing principles and antialias, in order to best represent the underlying data being characterized. We propose to use an adaptive filter (and thus produce consistently antialiased results). Second, we propose to use a bicubic, instead of bilinear filter, which offers stronger reconstruction. While such an implementation is currently found in PIL, future equivalent and efficient implementations would be of use.

3.3. Quantization and Image Compression

8-bit Quantization. While images are represented by 8-bit integers $\mathbb{Z}$, operations such as resizing and data augmentation, as well as the raw generator output will provide floating point numbers $\mathbb{R}$. Post-processing the results introduces more subtleties and affects standard metrics such as FID. Most simply, an image can be quantized by clipping the output between $[0, 255]$ and rounding to produce integers. This is a lossy step and only done when images need to be saved. Additionally, we observe that performing this step has a minor effect on the FID score (< 0.01).

Image compression. Saving the image as a raw matrix of values is data-intensive. However, an image contains redundant information that can be exploited. For example, the

PNG	JPEG 100	JPEG 90	JPEG 75
FID: 0	**FID:** 0.23	**FID:** 6.08	**FID:** 20.96
PSNR: ∞	**PSNR:** 49.3	**PSNR:** 41.7	**PSNR:** 38.9

Figure 6. **Effects of JPEG compression on an image.** We show a sample FFHQ [33] image, saved with lossless PNG and different JPEG compression ratios. The FID scores under the images are calculated between FFHQ images saved using the corresponding JPEG format and the PNG format. PSNR is computed with 1000 images. While the images are perceptually similar, this induces changes in the Inception-V3 activations, resulting in large FID.

PNG format compresses an image losslessly. To further save storage, images are commonly saved using the JPEG codec. While JPEG is a lossy compression technique, it aims to make changes that the human visual system is less sensitive to, namely reducing information in higher frequencies and chroma (color) components [67]. JPEG converts an image into a YCbCr space, subsamples the chroma components, divides images into 8×8 blocks, computes the Discrete Cosine Transform (DCT), and performs quantization. The quantization step facilitates a trade-off between the fidelity of the original image and the amount of the storage saved. In the PIL implementation [11], this is done using a "quality" option (0-100), which linearly scales the quantization tables (which controls which frequencies are quantized to what granularity). Note that setting the quality flag to 100 is *not* a lossless operation. Even when the quantization tables are not scaled, the DCT coefficients are quantized to integer values and the chroma components are subsampled.

Image compression changes deep network activations. In Figure 6, we show a real FFHQ [33] image at a resolution of 256, saved with lossless PNG and lossy JPEG (quality set to 100, 90, and 75). Despite being perceptually indistinguishable (with high PSNR values of ≥ 39), the FID scores increases. The PIL default of 75 results in a high score (21), for example. Note that this FID score is far higher than the score from a powerful generative model, StyleGAN2 [34] (around 3). Also, variations across recent methods are typically within 1 FID on FFHQ. We investigate the implications of using JPEG compression in the experiments below.

4. Experiments

As outlined in Section 3 and depicted in Figure 3, variations in FID arise from three distinct steps: resizing in the

Resize function	PIL–bicubic(Real Images) vs. Resize(Real Images) FID ↓	KID $\times10^3$ ↓	PSNR [db] ↑	Resize(StyleGAN2) FID ↓	KID $\times10^3$ ↓
PIL–bicubic (✓)	0	0	∞	2.98	0.51
PIL–bilinear (✓)	0.64	0.61	45.7	4.03	1.52
TensorFlow–bilinear (✗)	4.34	4.32	37.66	7.45	5.12
PyTorch–bilinear (✗)	4.36	4.31	37.66	7.45	5.15
Naive nearest (✗)	7.43	7.54	35.16	10.67	8.47

Table 1. **Deviations induced by varying resizing implementations.** We measure the discrepancy between real images downsampled with PIL–bicubic ($1024 \rightarrow 299$) vs. other functions ($\widehat{\psi}_{\text{FID}}$) on the left. If all downsampling functions were equivalent, the metrics (FID & KID) should be 0 and PSNR ∞. PIL–bilinear and bicubic adjust antialiasing to the downsampling factor (✓) and produce relatively low neural metric scores and high PSNRs. Functions using fixed width filters (✗) produce higher discrepancies. Naive nearest does not antialias at all. A similar trend holds on synthetic StyleGAN2 [34] images.

FID evaluation step (ψ_{FID}, $\widehat{\psi}_{\text{FID}}$), resizing in the data preprocessing step (ψ_{data}), and quantizing of images (Q, $\widehat{Q}$). We introduce sources of variation at these steps and investigate their impacts in Sections 4.1, 4.2, and 4.3 respectively.

4.1. Variation due to FID Resizing

Here we investigate the effects of different resizing methods (ψ_{FID}, $\widehat{\psi}_{\text{FID}}$) used in the FID calculation step.

Variation induced by resizing functions on real images. We start with two sets of full-resolution 1024×1024 face images - from the FFHQ dataset, and from a pre-trained StyleGAN2 generator. Each of the sets of images is resized from $1024\rightarrow299$ using different methods. In Table 1 (left), we compare the set of real images resized with the antialiased resizing operation (PIL bicubic) to the *same set of real images*, resized using other aliased functions that use a fixed width prefiltering kernel. As we compare the same set of images, we anticipate all FID and KID scores to be close to 0 and the PSNR values to be very high. However, as shown in Figures 1, 2, and 5, only a subset of the commonly used resizing operators adjust the filter width and antialias the images. These differences in resizing operations cause drastic changes in the Inception-V3 [63] activation maps.

Filters that antialias are more consistent, even with different filter types – PIL-bilinear has FID 0.64 when compared to PIL-bicubic. On the other hand, implementations that ignore the downsampling factor (PyTorch and TensorFlow) show much larger deviation (FID 4.3), with scores nearing naive nearest (FID 7.4), that does not filter at all. This indicates that whether the filter adapts to the downsampling factor can change the modeled data distribution by non-trivial amounts.

Variation induced by resizing functions on generated images. After studying the effects on real images, we evaluate how different resizing function $\widehat{\psi}_{\text{FID}}$ choices affect the metric

Resize function	Resize(Dataset Images) vs. Resize(StyleGAN2)			
	FFHQ FID ↓	MetFaces FID ↓	AFHQ-Cats FID ↓	AFHQ-Dogs FID ↓
PIL–bicubic (✓)	2.98	65.32	5.13	20.16
PIL–bilinear (✓)	2.99	64.31	5.01	19.60
TensorFlow–bilinear (✗)	2.75	57.45	4.93	19.45
PyTorch–bilinear (✗)	2.75	57.46	4.94	19.46
Naive nearest (✗)	2.68	55.09	4.80	18.25

Table 2. **Resizing functions affect FID scores.** Here, both resizing functions on real and synthetic images (ψ_{FID}, $\widehat{\psi}_{\text{FID}}$) are the same. If all resizing functions were consistent, all rows would be equal. Interestingly, the downsampling methods that alias result in lower scores; the lowest score is achieved by naive nearest subsampling. Methods that adjust the prefilter size to downsampling factor (implemented by PIL) better preserve information of the original images. This indicates that antialiasing enables subsequent FID to more sensitive to differences in the distributions.

when used in a full generative modeling pipeline. Here, we evaluate a pretrained StyleGAN2 generator [34] trained on FFHQ (1024), MetFaces (1024), and AFHQ (512) dataset images, and compute the FID with 50,000 images. In Table 1 (right), we consider the asymmetric case, where features for the real images and generated images use different resizing functions. This case arises when features for real images are pre-computed and shared by one group of authors, while generated features may be calculated on the fly with a different library. Here, we observe that using the same resizing function as the reference dataset (PIL-bicubic) achieves the lowest performance. Using a different resize function, such as PIL-bilinear increases the score to 4. Using an aliased function increases the score drastically to 7, close to naive subsampling (> 10).

Next, in Table 2, we show a comparison when the same resizing function is used for the real dataset images and the StyleGAN2 generated images. Interestingly, we observe that the *aliased* resizing functions result in lower FID scores across multiple commonly used datasets - FFHQ (1024), MetFaces [31] (1024), and AFHQ [9] (512). This indicates that using the antialiased function as preprocessing makes the downstream FID calculation more sensitive at measuring the discrepancies between distributions.

4.2. Variation due to Dataset Resizing

Previously, we considered scenarios where the dataset was not downsampled. However, as discussed in Section 1 and illustrated in Figure 3, dataset downsampling is needed when training a model on a low-resolution version of the original dataset [31, 72, 75] (e.g., 256 for FFHQ). Before, the target distribution was fixed, and differences were purely introduced during post-hoc metric evaluation. Now, the situation is much more intricate. *Different resizing choices will result in different training distributions entirely*. In Table 3, we train three different StyleGAN2 [34] (config-e) models, following the official PyTorch implementation* for 25k iterations. We resize FFHQ [33] to 256 using Naive Nearest, PIL–bicubic, PyTorch–bilinear, and TensorFlow–bilinear. We use the same PIL–bicubic function (ψ_{FID}, $\widehat{\psi}_{\text{FID}}$) for FID evaluation; note that here, it is upsampling ($256 \rightarrow 299$). Qualitatively, using an aliased downsampling function produces a training distribution with visual artifacts for the generative model to mimic, likely different than the natural visual data we wish to model. Quantitatively, interestingly, we observe that that the aliased pre-processing results in *lower* FID values. As the antialiased function better preserves signal in the original images, we hypothesize that retaining more information from the original input actually produces a more difficult distribution to model.

*https://github.com/NVlabs/stylegan2-ada

Dataset preprocessing	FID ↓ on FFHQ PIL-bicubic
Naive Nearest (✗)	4.82 ± 0.09
PyTorch–bilinear (✗)	5.13 ± 0.20
TensorFlow–bilinear (✗)	5.08 ± 0.16
PIL–bicubic (✓)	6.21 ± 0.23

Table 3. **Dataset resizing.** We downsample the FFHQ dataset using different resize functions $\psi_{\mathbf{data}}$ from 1024 to 256. We train StyleGAN2 [34] (Config-E) models, using the identical training procedure and report FID of the result. The score is computed across three different training runs for each of the setting. The scores show large variation, indicating the resizing function can greatly affect the training distribution. Using a preprocessing function that antialiases (marked by ✓) preserves more information from the original images and interestingly results in a higher score.

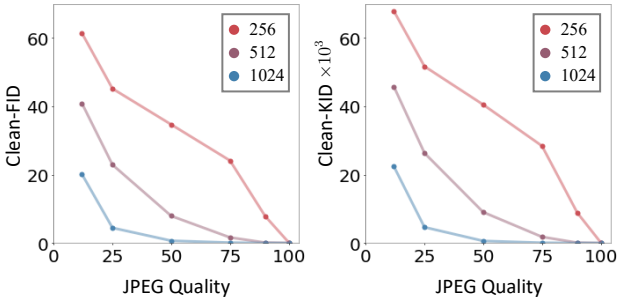

Figure 7. **Effects of JPEG compression.** The FFHQ dataset images are resized from 1024 to different resolutions (512 and 256) using PIL-bicubic and compressed using the JPEG format, with different compression ratios. Subsequently, we plot the FID (left) and KID (right) between the compressed images and uncompressed images, at the same resolution, as a function of JPEG compression. The effect of JPEG compression is more severe for smaller images.

4.3. Variation due to Quantization/Compression

JPEG during evaluation. In Figure 7, we test the effect of quantization applied to real FFHQ images at different resolutions on FID (left) and KID (right). For each resolution, the real dataset images are correspondingly downsampled using PIL–bicubic, and the scores are computed between the resized uncompressed PNG images and the resized JPEG-

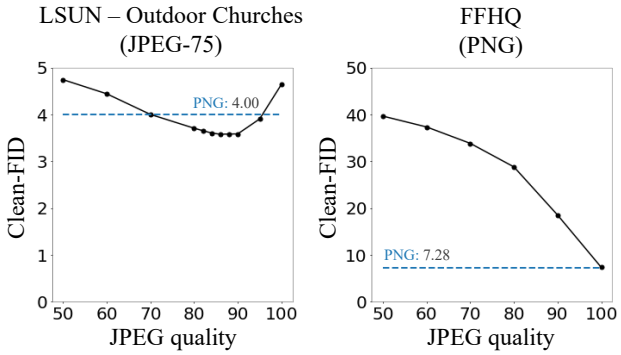

Figure 8. **Effects of image quantization/compression.** We plot FID as a function of JPEG compression, applied to StyleGAN2 images [34], trained on LSUN Churches [70] (left) and FFHQ [33] (right) at a resolution of 256×256. The blue dashed line shows FID when the generated images are quantized to 8-bit unsigned integers (PNG). Interestingly, when training with JPEG-75 dataset images (left), applying lossy compression artifically improves the FID score by a large margin (4.00→3.48).

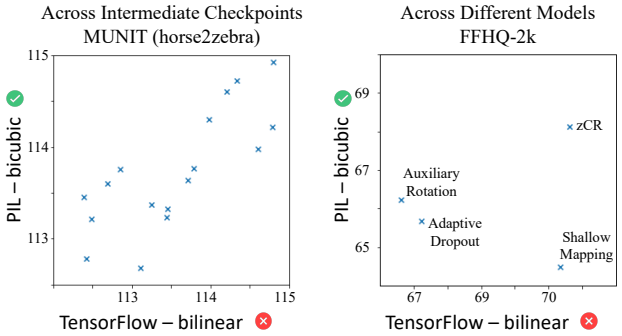

Figure 9. **FID inconsistencies when evaluating models and checkpoints.** We compare the FID scores induced by different resizing functions. (Left) We show different intermediate checkpoints while training a MUNIT model [28] on the horse2zebra dataset [78]. (Right) We compare methods trained on FFHQ-2k. The non-monotonic relationship demonstrates the sensitivity of the FID metric to the resizing function. As a consequence, different checkpoints or methods may be selected, depending on if an aliased or an anti-aliased resizing function is chosen.

compressed images. Figure 7 shows that the effect of the JPEG compression on both metrics. The effect is more pronounced for lower resolutions, where the artifacts remain after the subsequent resampling step.

JPEG on training images. The comparisons above use the FFHQ dataset, which was collected as uncompressed PNG files. Any additional compression only monotonically increases the FID score (Figure 8 right). This is expected, as information is being removed from the generator. However, this does not apply to other datasets which were collected as JPEG images. To study this effect, we train a StyleGAN2 model [34] on the LSUN outdoor Church dataset [70], which was collected as JPEG-75. In Figure 8 (left), we plot the FID of the trained generator as a function of JPEG compression and observe that the FID score for the StyleGAN2 model surprisingly *improves when slight JPEG compression is added*. This indicates that interestingly, though the model is able to capture complex variations in the dataset, it is unable to fully model the low-level statistics induced by JPEG compression. The best FID score (3.48) is obtained when the generated images are compressed with JPEG quality 87 (not the full 75), indicating the model is able to replicate some of the artifacts, but not all. The FID score for the generated images stores as PNG files is 4.00. Furthermore, this indicates that the metric is sensitive to low-level statistics, and a large gain in the metric could be achieved simply through manual post-processing. Following these observations, we recommend that researchers curate and store training images as PNG formats for the future image synthesis datasets.

4.4. Consequences in model selection

In this section, we show that using aliased resizing can result in different conclusions, both when comparing across different methods and when choosing a "best" model checkpoint. Concretely, in Figure 9 (left) we evaluate the different intermediate checkpoints when training an image-to-image translation model [28] on the horse2zebra dataset. In Figure 9 (right) we evaluate the StyleGAN2 [34] models with different data augmentation trained to generate 256×256 FFHQ images [33] in a few shot setting (2000 training images). In both cases, the choice of the resizing function leads to a different best model getting selected.

5. Recommendations

We have shown surprisingly sensitivities to seemingly inconsequential implementation details when evaluating generative models. The resize operation and the image quantization/compression are especially impactful. Based on our observations, we discuss some best practices when training and evaluating generative models. We recommend using implementations that adapt the filter size to the downsampling factor, following signal processing principles, at each resizing step (ψ_{data}, ψ_{FID}, and $\widehat{\psi}_{\text{FID}}$). There are many details one needs to keep track of when computing FID. Any inconsistency leads to results that are no longer comparable to other methods. To facilitate an easy comparison, avoid inconsistent comparisons, and encourage the usage of critical operations that are correctly implemented, we provide an easy-to-use library (`github.com/GaParmar/clean-fid`), and pre-computed Inception statistics for common datasets.

Acknowledgments. We thank Jaakko Lehtinen and Assaf Shocher for bringing attention to this issue and for helpful discussion. We thank Sheng-Yu Wang, Nupur Kumari, Kangle Deng, and Andrew Liu for useful discussions. We thank William S. Peebles, Shengyu Zhao, Taesung Park, Kris Kitani, and Stanislav Panev for proofreading our manuscript. We are grateful for the support of Adobe, Naver Corporation, and Sony Corporation.

References

[1] Sanjeev Arora and Yi Zhang. Do gans actually learn the distribution? an empirical study. In *International Conference on Learning Representations (ICLR)*, 2018.

[2] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20:1–25, 2019.

[3] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. *ICCV*, 2021.

[4] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *ICLR*, 2018.

[5] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

[8] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *European Conference on Computer Vision*, pages 103–120. Springer, 2020.

[9] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[10] Min Jin Chong and David Forsyth. Effectively unbiased fid and inception score and where to find them. In *CVPR*, 2020.

[11] Alex Clark. Pillow (pil fork) documentation, 2015.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in Neural Information Processing Systems*, 2015.

[14] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations (ICLR)*, 2017.

[15] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, 2019.

[16] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, 2016.

[17] Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. In *Advances in Neural Information Processing Systems*, 2019.

[18] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019.

[19] James D Foley, Foley Dan Van, Andries Van Dam, Steven K Feiner, John F Hughes, and J Hughes. *Computer graphics: principles and practice*, volume 12110. Addison-Wesley Professional, 1996.

[20] James D Foley, Andries Van Dam, Steven K Feiner, John F Hughes, and Richard L Phillips. *Introduction to computer graphics*, volume 55. Addison-Wesley Reading, 1994.

[21] David A Forsyth and Jean Ponce. *Computer vision: a modern approach*. Pearson,, 2012.

[22] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[23] Rafael C Gonzalez, Richard E Woods, et al. Digital image processing, 2002.

[24] 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*, 2014.

[25] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.

[26] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

[27] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[28] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *European Conference on Computer Vision (ECCV)*, 2018.

[29] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.

[30] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018.

[31] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *NIPS*, 33, 2020.

[32] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. In *Proc. NeurIPS*, 2021.

[33] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[34] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving

the image quality of stylegan. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[35] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981.

[36] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

[37] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

[38] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, 2019.

[39] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[40] Erich L Lehmann and Joseph P Romano. *Testing statistical hypotheses*. Springer Science & Business Media, 2006.

[41] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[42] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[43] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *ICLR*, 2017.

[44] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study. In *Advances in Neural Information Processing Systems*, 2018.

[45] Julien Mairal, Piotr Koniusz, Zaid Harchaoui, and Cordelia Schmid. Convolutional kernel networks. *Advances in neural information processing systems*, 27:2627–2635, 2014.

[46] Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[47] Harry Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[48] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. In *Advances in Neural Information Processing Systems*, 2016.

[49] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Pearson, 2nd edition, 1999.

[50] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[51] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[52] Shengju Qian, Hao Shao, Yi Zhu, Mu Li, and Jiaya Jia. Blending anti-aliasing into vision transformer. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[53] Erwin Quiring, David Klein, Daniel Arp, Martin Johns, and Konrad Rieck. Adversarial preprocessing: Understanding and preventing image-scaling attacks in machine learning. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 1363–1380, 2020.

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

[55] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems*, 2019.

[56] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NIPS*, 2019.

[57] Mehdi SM Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, 2018.

[58] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009.

[59] 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*, 2016.

[60] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer, 2010.

[61] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russ Webb. Learning from simulated and unsupervised images through adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[62] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for texture classification. *arXiv preprint arXiv:1403.1687*, 2014.

[63] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[64] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[65] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[66] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In *ICLR*, 2016.

[67] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[68] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. Cnn-generated images are surprisingly easy to spot... for now. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[69] Qixue Xiao, Yufei Chen, Chao Shen, Yu Chen, and Kang Li. Seeing is not believing: Camouflage attacks on image scaling algorithms. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 443–460, 2019.

[70] Fisher Yu, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[71] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[72] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020.

[73] Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning (ICML)*, 2019.

[74] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[75] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[76] Sharon Zhou, Mitchell L Gordon, Ranjay Krishna, Austin Narcomey, Li Fei-Fei, and Michael S Bernstein. Hype: A benchmark for human eye perceptual evaluation of generative models. In *Advances in Neural Information Processing Systems*, 2019.

[77] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*, 2016.

[78] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[79] Xueyan Zou, Fanyi Xiao, Zhiding Yu, and Yong Jae Lee. Delving deeper into anti-aliasing in convnets. In *BMVC*, 2020.