High-Fidelity Image Generation With Fewer Labels

Mario Lucic [*1] **Michael Tschannen** [*2] **Marvin Ritter** [*1] **Xiaohua Zhai** [1] **Olivier Bachem** [1] **Sylvain Gelly** [1]

Abstract

Deep generative models are becoming a cornerstone of modern machine learning. Recent work on conditional generative adversarial networks has shown that learning complex, high-dimensional distributions over natural images is within reach. While the latest models are able to generate high-fidelity, diverse natural images at high resolution, they rely on a vast quantity of labeled data. In this work we demonstrate how one can benefit from recent work on self- and semi-supervised learning to outperform the state of the art on both unsupervised ImageNet synthesis, as well as in the conditional setting. In particular, the proposed approach is able to match the sample quality (as measured by FID) of the current state-of-the-art conditional model BigGAN on ImageNet *using only 10% of the labels* and outperform it using 20% of the labels.

1. Introduction

Deep generative models have received a great deal of attention due to their power to learn complex high-dimensional distributions, such as distributions over natural images (van den Oord et al., 2016b; Dinh et al., 2017; Brock et al., 2019), videos (Kalchbrenner et al., 2017), and audio (van den Oord et al., 2016a). Recent progress was driven by scalable training of large-scale models (Brock et al., 2019; Menick & Kalchbrenner, 2019), architectural modifications (Zhang et al., 2019; Chen et al., 2019a; Karras et al., 2019), and normalization techniques (Miyato et al., 2018).

Currently, high-fidelity natural image generation hinges upon having access to vast quantities of labeled data. The labels induce rich side information into the training process which effectively decomposes the extremely challenging image generation task into semantically meaningful sub-tasks.

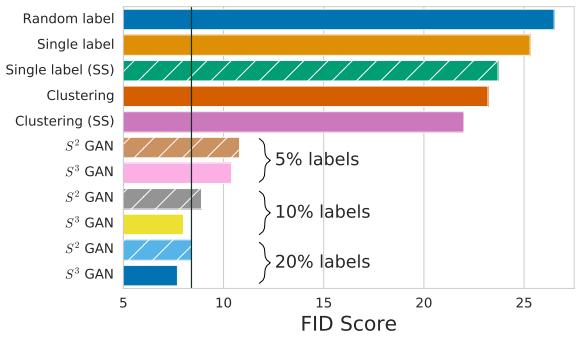


Figure 1. Median FID of the baselines and the proposed method. The vertical line indicates the baseline (BIGGAN) which uses all the labeled data. The proposed method (S^3GAN) is able to match the state-of-the-art while using only 10% of the labeled data and outperform it with 20%.

However, this dependence on vast quantities of labeled data is at odds with the fact that most data is unlabeled, and labeling itself is often costly and error-prone. Despite the recent progress on unsupervised image generation, the gap between conditional and unsupervised models in terms of sample quality is significant.

In this work, we take a significant step towards closing the gap between conditional and unsupervised generation of high-fidelity images using generative adversarial networks (GANs). We leverage two simple yet powerful concepts:

(i) Self-supervised learning: A semantic feature extractor for the training data can be learned via self-supervision, and the resulting feature representation can then be employed to guide the GAN training process.

(ii) Semi-supervised learning: Labels for the entire training set can be inferred from a small subset of labeled training images and the inferred labels can be used as conditional information for GAN training.

Our contributions In this work, we

1. propose and study various approaches to reduce or fully omit ground-truth label information for natural image generation tasks,
2. achieve a new state of the art (SOTA) in unsupervised generation on IMAGENET, match the SOTA on 128×128 IMAGENET using only 10% of the labels, and set a new SOTA (measured by FID) using 20% of the labels, and
3. open-source all the code used for the experiments at `github.com/google/compare_gan`.

[*]Equal contribution [1]Google Research, Brain Team [2]ETH Zurich. Correspondence to: Mario Lucic <lucic@google.com>, Michael Tschannen <mi.tschannen@gmail.com>, Marvin Ritter <marvinritter@google.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

2. Background and related work

High-fidelity GANs on IMAGENET Besides BIGGAN (Brock et al., 2019) only a few prior methods have managed to scale GANs to ImageNet, most of them relying on class-conditional generation using labels. One of the earliest attempts are GANs with auxiliary classifier (ACGANs) (Odena et al., 2017) which feed one-hot encoded label information with the latent code to the generator and equip the discriminator with an auxiliary head predicting the image class in addition to whether the input is real or fake. More recent approaches rely on a label projection layer in the discriminator, essentially resulting in per-class real/fake classification (Miyato & Koyama, 2018), and self-attention in the generator (Zhang et al., 2019). Both methods use modulated batch normalization (De Vries et al., 2017) to provide label information to the generator. On the unsupervised side, Chen et al. (2019b) showed that auxiliary rotation loss added to the discriminator has a stabilizing effect on the training. Finally, appropriate gradient regularization enables scaling MMD-GANs to ImageNet without using labels (Arbel et al., 2018).

Semi-supervised GANs Several recent works leveraged GANs for semi-supervised learning of classifiers. Both Salimans et al. (2016) and Odena (2016) train a discriminator that classifies its input into $K + 1$ classes: K image classes for real images, and one class for generated images. Similarly, Springenberg (2016) extends the standard GAN objective to K classes. This approach was also considered by Li et al. (2017) where separate discriminator and classifier models are applied. Other approaches incorporate inference models to predict missing labels (Deng et al., 2017) or harness joint distribution (of labels and data) matching for semi-supervised learning (Gan et al., 2017). Up to our knowledge, improvements in sample quality through partial label information are only reported in Li et al. (2017); Deng et al. (2017); Sricharan et al. (2017), all of which consider only low-resolution data sets from a restricted domain.

Self-supervised learning Self-supervised learning methods employ a label-free auxiliary task to learn a semantic feature representation of the data. This approach was successfully applied to different data modalities, such as images (Doersch et al., 2015; Caron et al., 2018), video (Agrawal et al., 2015; Lee et al., 2017), and robotics (Jang et al., 2018; Pinto & Gupta, 2016). The current state-of-the-art method on IMAGENET is due to Gidaris et al. (2018) who proposed predicting the rotation angle of rotated training images as an auxiliary task. This simple self-supervision approach yields representations which are useful for downstream image classification tasks. Other forms of self-supervision include predicting relative locations of disjoint image patches of a given image (Doersch et al., 2015; Mundhenk et al., 2018) or estimating the permutation of randomly swapped image patches on a regular grid (Noroozi & Favaro, 2016). A study on self-supervised learning with modern neural architectures is provided in Kolesnikov et al. (2019).

Figure 2. Top row: 128×128 samples from our implementation of the fully supervised current SOTA model BIGGAN. Bottom row: Samples form the proposed S^{3}GAN which matches BIGGAN in terms of FID and IS using only 10% of the ground-truth labels.

3. Reducing the appetite for labeled data

In a nutshell, instead of providing hand-annotated ground truth labels for real images to the discriminator, we will provide inferred ones. To obtain these labels we will make use of recent advancements in self- and semi-supervised learning. We propose and study several different methods with different degrees of computational and conceptual complexity. We emphasize that *our work focuses on using few labels to improve the quality of the generative model*, rather than training a powerful classifier from a few labels as extensively studied in prior work on semi-supervised GANs.

Before introducing these methods in detail, we discuss how label information is used in state-of-the-art GANs. The following exposition assumes familiarity with the basics of the GAN framework (Goodfellow et al., 2014).

Incorporating the labels To provide the label information to the discriminator we employ a linear projection layer as proposed by Miyato & Koyama (2018). To make the exposition self-contained, we will briefly recall the main ideas. In a "vanilla" (unconditional) GAN, the discriminator D learns to predict whether the image x at its input is real or generated by the generator G. We decompose the discriminator into a learned discriminator representation, $\tilde{D}$, which is fed into a linear classifier, $c_{\text{r/f}}$, i.e., the discriminator is given by $c_{\text{r/f}}(\tilde{D}(x))$. In the *projection discriminator*, one learns an embedding for each class of the same dimension as the representation $\tilde{D}(x)$. Then, for a given image, label input x, y the decision on whether the sample is real or generated is based on two components: (a) on whether the representation $\tilde{D}(x)$ itself is consistent with the real data, and (b) on whether the representation $\tilde{D}(x)$ is consistent with the real data *from class* y. More formally, the discrim-

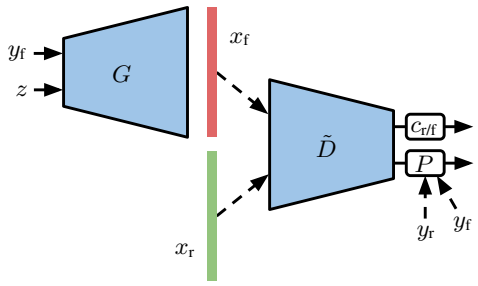


Figure 3. Conditional GAN with projection discriminator. The discriminator tries to predict from the representation $\tilde{D}$ whether a real image x_r (with label y_r) or a generated image x_f (with label y_f) is at its input, by combining an unconditional classifier $c_{r/f}$ and a class-conditional classifier implemented through the projection layer P. This form of conditioning is used in BIGGAN. Outward-pointing arrows feed into losses.

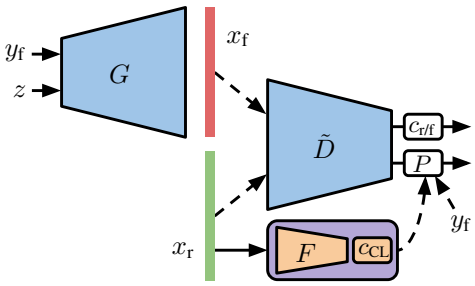


Figure 4. CLUSTERING: Unsupervised approach based on clustering the representations obtained by solving a self-supervised task. F corresponds to the feature extractor learned via self-supervision and c_{CL} is the cluster assignment function. After learning F and c_{CL} on the real training images in the pre-training step, we proceed with conditional GAN training by inferring the labels as $\hat{y}_{CL} = c_{CL}(F(x))$.

inator takes the form $D(x, y) = c_{r/f}(\tilde{D}(x)) + P(\tilde{D}(x), y)$, where $P(\tilde{x}, y) = \tilde{x}^\top W y$ is a linear projection layer with learned weight matrix W applied to a feature vector $\tilde{x}$ and the one-hot encoded label y as an input. As for the generator, the label information y is incorporated through class-conditional BatchNorm (Dumoulin et al., 2017; De Vries et al., 2017). The conditional GAN with projection discriminator is illustrated in Figure 3.

We proceed with describing the proposed pre-trained and co-training approaches to infer labels for GAN training in Sections 3.1 and 3.2, respectively.

3.1. Pre-trained approaches

Unsupervised clustering-based method We first learn a representation of the real training data using a state-of-the-art self-supervised approach (Gidaris et al., 2018; Kolesnikov et al., 2019), perform clustering on this representation, and use the cluster assignments as a replacement for labels. Following Gidaris et al. (2018) we learn the feature extractor F (typically a convolutional neural network) by minimizing the following *self-supervision loss*

$$\mathcal{L}_R = -\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}} \mathbb{E}_{x\sim p_{data}(x)}[\log p(c_R(F(x^r)) = r)], \quad (1)$$

where $\mathcal{R}$ is the set of the 4 rotation degrees $\{0°, 90°, 180°, 270°\}$, x^r is the image x rotated by r, and c_R is a linear classifier predicting the rotation degree r. After learning the feature extractor F, we apply mini batch k-Means clustering (Sculley, 2010) on the representations of the training images. Finally, given the cluster assignment function $\hat{y}_{CL} = c_{CL}(F(x))$ we train the GAN using the hinge loss, alternatively minimizing the discriminator loss $\mathcal{L}_D$ and generator loss $\mathcal{L}_G$, namely

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{x\sim p_{data}(x)}[\min(0, -1 + D(x, c_{CL}(F(x))))] \\ &\quad - \mathbb{E}_{(z,y)\sim\hat{p}(z,y)}[\min(0, -1 - D(G(z, y), y))] \\ \mathcal{L}_G &= -\mathbb{E}_{(z,y)\sim\hat{p}(z,y)}[D(G(z, y), y)],\end{aligned}$$

where $\hat{p}(z, y) = p(z)\hat{p}(y)$ is the prior distribution with $p(z) = \mathcal{N}(0, I)$ and $\hat{p}(y)$ the empirical distribution of the cluster labels $c_{CL}(F(x))$ over the training set. We call this approach CLUSTERING and illustrate it in Figure 4.

Semi-supervised method While semi-supervised learning is an active area of research and a large variety of algorithms has been proposed, we follow Zhai et al. (2019) and simply extend the self-supervised approach described in the previous paragraph with a semi-supervised loss. This ensures that the two approaches are comparable in terms of model capacity and computational cost. Assuming we are provided with labels for a subset of the training data, we attempt to learn a good feature representation via self-supervision and simultaneously train a good linear classifier on the so-obtained representation (using the provided labels).[1] More formally, we minimize the loss

$$\begin{aligned}\mathcal{L}_{S^2L} = -\frac{1}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\Big\{&\mathbb{E}_{x\sim p_{data}(x)}[\log p(c_R(F(x^r)) = r)] \\ &+ \gamma\mathbb{E}_{(x,y)\sim p_{data}(x,y)}[\log p(c_{S^2L}(F(x^r)) = y)]\Big\},\end{aligned} \quad (2)$$

where c_R and c_{S^2L} are linear classifiers predicting the rotation angle r and the label y, respectively, and $\gamma > 0$

[1] Note that an even simpler approach would be to first learn the representation via self-supervision and *subsequently* the linear classifier, but we observed that learning the representation and classifier simultaneously leads to better results.

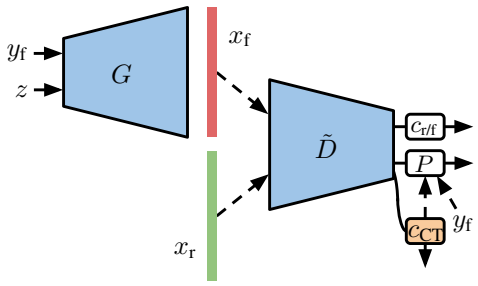


Figure 5. S^{2}GAN-CO: During GAN training we learn an auxiliary classifier c_{CT} on the discriminator representation $\tilde{D}$, based on the labeled real examples, to predict labels for the unlabeled ones. This avoids training a feature extractor F and classifier $c_{\text{S}^2\text{L}}$ prior to GAN training as in S^{2}GAN.

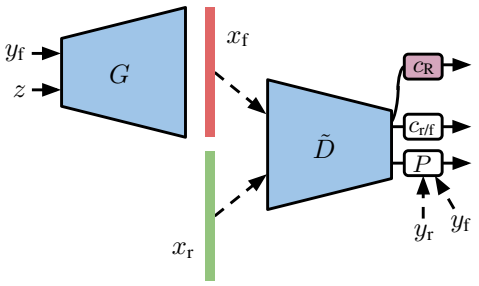


Figure 6. Self-supervision by rotation-prediction during GAN training. Additionally to predicting whether the images at its input are real or generated, the discriminator is trained to predict rotations of both rotated real and fake images via an auxiliary linear classifier c_{R}. This approach was successfully applied by Chen et al. (2019b) to stabilize GAN training. Here we combine it with our pre-trained and co-training approaches, replacing the ground truth labels y_r with predicted ones.

balances the loss terms. The first term in (2) corresponds to the self-supervision loss from (1) and the second term to a (semi-supervised) cross-entropy loss. During training, the latter expectation is replaced by the empirical average over the subset of labeled training examples, whereas the former is set to the empirical average over the entire training set (this convention is followed throughout the paper). After we obtain F and $c_{\text{S}^2\text{L}}$ we proceed with GAN training where we label the real images as $\hat{y}_{\text{S}^2\text{L}} = c_{\text{S}^2\text{L}}(F(x))$. In particular, we alternatively minimize the same generator and discriminator losses as for CLUSTERING except that we use $c_{\text{S}^2\text{L}}$ and F obtained by minimizing (2):

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{x\sim p_{\text{data}}(x)}[\min(0, -1 + D(x, c_{\text{S}^2\text{L}}(F(x))))] \\ &- \mathbb{E}_{(z,y)\sim p(z,y)}[\min(0, -1 - D(G(z,y), y))] \\ \mathcal{L}_G = &-\mathbb{E}_{(z,y)\sim p(z,y)}[D(G(z,y), y)],\end{aligned}$$

where $p(z, y) = p(z)p(y)$ with $p(z) = \mathcal{N}(0, I)$ and $p(y)$ uniform categorical. We use the abbreviation S^{2}GAN for this method.

3.2. Co-training approach

The main drawback of the transfer-based methods is that one needs to train a feature extractor F via self-supervision and learn an inference mechanism for the labels (linear classifier or clustering). In what follows we detail co-training approaches that avoid this two-step procedure and learn to infer label information during GAN training.

Unsupervised method We consider two approaches. In the first one, we completely remove the labels by simply labeling all real and generated examples with the same label[2] and removing the projection layer from the discriminator, i.e., we set $D(x) = c_{\text{r/f}}(\tilde{D}(x))$. We use the abbreviation SINGLE LABEL for this method. For the second approach we assign random labels to (unlabeled) real images. While the labels for the real images do not provide any useful signal to the discriminator, the sampled labels could potentially help the generator by providing additional randomness with different statistics than z, as well as additional trainable parameters due to the embedding matrices in class-conditional BatchNorm. Furthermore, the labels for the fake data could facilitate the discrimination as they provide side information about the fake images to the discriminator. We term this method RANDOM LABEL.

[2]Note that this is not necessarily equivalent to replacing class-conditional BatchNorm with standard (unconditional) BatchNorm as the variant of conditional BatchNorm used in this paper also uses chunks of the latent code as input; besides the label information.

Semi-supervised method When labels are available for a subset of the real data, we train an auxiliary linear classifier c_{CT} directly on the feature representation $\tilde{D}$ of the discriminator, *during GAN training*, and use it to predict labels for the unlabeled real images. In this case the discriminator loss takes the form

$$\begin{aligned}\mathcal{L}_D = &-\mathbb{E}_{(x,y)\sim p_{\text{data}}(x,y)}[\min(0, -1 + D(x,y))] \\ &- \lambda\mathbb{E}_{(x,y)\sim p_{\text{data}}(x,y)}[\log p(c_{\text{CT}}(\tilde{D}(x)) = y)] \\ &- \mathbb{E}_{x\sim p_{\text{data}}(x)}[\min(0, -1 + D(x, c_{\text{CT}}(\tilde{D}(x))))] \\ &- \mathbb{E}_{(z,y)\sim p(z,y)}[\min(0, -1 - D(G(z,y), y))], \quad (3)\end{aligned}$$

where the first term corresponds to standard conditional training on (k%) labeled real images, the second term is the cross-entropy loss (with weight $\lambda > 0$) for the auxiliary classifier c_{CT} on the labeled real images, the third term is an unsupervised discriminator loss where the labels for the unlabeled real images are predicted by c_{CT}, and the last term is the standard conditional discriminator loss on the generated data. We use the abbreviation S^{2}GAN-CO for this method. See Figure 5 for an illustration.

3.3. Self-supervision during GAN training

So far we leveraged self-supervision to either craft good feature representations, or to learn a semi-supervised model (cf. Section 3.1). However, given that the discriminator itself is just a classifier, one may benefit from augmenting this classifier with an auxiliary task—namely self-supervision through rotation prediction. This approach was already explored in Chen et al. (2019b), where it was observed to stabilize GAN training. Here we want to assess its impact when combined with the methods introduced in Sections 3.1 and 3.2. To this end, similarly to the training of F in (1) and (2), we train an additional linear classifier c_{R} on the discriminator feature representation $\tilde{D}$ to predict rotations $r \in \mathcal{R}$ of the rotated real images x^r and rotated fake images $G(z, y)^r$. The corresponding loss terms added to the discriminator and generator losses are

$$-\frac{\beta}{|\mathcal{R}|}\sum_{r\in\mathcal{R}}\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log p(c_{\text{R}}(\tilde{D}(x^r)=r)] \quad (4)$$

and

$$-\frac{\alpha}{|\mathcal{R}|}\mathbb{E}_{(z,y)\sim p(z,y)}[\log p(c_{\text{R}}(\tilde{D}(G(z,y)^r)=r)], \quad (5)$$

respectively, where $\alpha, \beta > 0$ are weights to balance the loss terms. This approach is illustrated in Figure 6.

4. Experimental setup

Architecture and hyperparameters GANs are notoriously unstable to train and their performance strongly depends on the capacity of the neural architecture, optimization hyperparameters, and appropriate regularization (Lucic et al., 2018; Kurach et al., 2019). We implemented the conditional BIGGAN architecture (Brock et al., 2019) which achieves state-of-the-art results on ImageNet.[3] We use exactly the same optimization hyper-parameters as Brock et al. (2019). Specifically, we employ the Adam optimizer with the learning rates $5 \cdot 10^{-5}$ for the generator and $2 \cdot 10^{-4}$ for the discriminator ($\beta_1 = 0$, $\beta_2 = 0.999$). We train for 250k generator steps with 2 discriminator steps before each generator step. The batch size was fixed to 2048, and we use a latent code z with 120 dimensions. We employ spectral normalization in both generator and discriminator. In contrast to BIGGAN, we do not apply orthogonal regularization as this was observed to only marginally improve sample quality (cf. Table 1 in Brock et al. (2019)) and we do not use the truncation trick.

[3]We dissected the model checkpoints released by Brock et al. (2019) to obtain exact counts of trainable parameters and their dimensions, and match them to *byte* level (cf. Tables 11 and 10 in Appendix B). We want to emphasize that at this point this methodology is *bleeding-edge* and successful state-of-the-art methods require careful architecture-level tuning. To foster reproducibility we meticulously detail this architecture at tensor-level detail in Appendix B and open-source our code at https://github.com/google/compare_gan.

Table 1. A short summary of the analyzed methods. The detailed descriptions of pre-training and co-trained approaches can be found in Sections 3.1 and 3.2, respectively. Self-supervision during GAN training is described in Section 3.3.

METHOD	DESCRIPTION
BIGGAN	Conditional (Brock et al., 2019)
SINGLE LABEL	Co-training: Single label
RANDOM LABEL	Co-training: Random labels
CLUSTERING	Pre-trained: Clustering
BIGGAN-$k\%$	BIGGAN using only $k\%$ labeled data
S^2GAN-CO	Co-training: Semi-supervised
S^2GAN	Pre-trained: Semi-supervised
S^3GAN	S^2GAN with self-supervision
S^3GAN-CO	S^2GAN-CO with self-supervision

Datasets We focus primarily on IMAGENET, the largest and most diverse image data set commonly used to evaluate GANs. IMAGENET contains 1.3M training images and 50k test images, each corresponding to one of 1k object classes. We resize the images to $128 \times 128 \times 3$ as done in Miyato & Koyama (2018) and Zhang et al. (2019). Partially labeled data sets for the semi-supervised approaches are obtained by randomly selecting $k\%$ of the samples from each class.

Evaluation metrics We use the Fréchet Inception Distance (FID) (Heusel et al., 2017) and Inception Score (Salimans et al., 2016) to evaluate the quality of the generated samples. To compute the FID, the real data and generated samples are first embedded in a specific layer of a pre-trained Inception network. Then, a multivariate Gaussian is fit to the data and the distance computed as $\text{FID}(x, g) = ||\mu_x - \mu_g||_2^2 + \text{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}})$, where μ and Σ denote the empirical mean and covariance, and subscripts x and g denote the real and generated data respectively. FID was shown to be sensitive to both the addition of spurious modes and to mode dropping (Sajjadi et al., 2018; Lucic et al., 2018). Inception Score posits that conditional label distribution of samples containing meaningful objects should have low entropy, and the variability of the samples should be high leading to the following formulation: $\text{IS} = \exp(\mathbb{E}_{x\sim Q}[d_{KL}(p(y \mid x), p(y))])$. Although it has some flaws (Barratt & Sharma, 2018), we report it to enable comparison with existing methods. Following Brock et al. (2019), the FID is computed using the 50k IMAGENET testing images and 50k randomly sampled fake images, and the IS is computed from 50k randomly sampled fake images. All metrics are computed for 5 different randomly sampled sets of fake images and are then averaged.

Methods We conduct an extensive comparison of methods detailed in Table 1, namely: Unmodified BIGGAN, the unsupervised methods SINGLE LABEL, RANDOM LABEL, CLUSTERING, and the semi-supervised methods S^{2}GAN and S^{2}GAN-CO. In all S^{2}GAN-CO experiments we use soft labels, i.e., the soft-max output of c_{CT} instead of one-hot encoded hard estimates, as we observed in preliminary experiments that this stabilizes training. For S^{2}GAN we use hard labels by default, but investigate the effect of soft labels in separate experiments. For all semi-supervised methods we have access only to $k\%$ of the ground truth labels where $k \in \{5, 10, 20\}$. As an additional baseline, we retain $k\%$ labeled real images and discard all unlabeled real images, then using the remaining labeled images to train BIGGAN (the resulting model is designated by BIGGAN-$k\%$). Finally, we explore the effect of self-supervision during GAN training on the unsupervised and semi-supervised methods.

We train every model three times with a different random seed and report the median FID and the median IS. With the exception of the SINGLE LABEL and BIGGAN-$k\%$, the standard deviation of the mean across three runs is very low. We therefore defer tables with the mean FID and IS values and standard deviations to Appendix D. All models are trained on 128 cores of a Google TPU v3 Pod with BatchNorm statistics synchronized across cores.

Unsupervised approaches For CLUSTERING we simply used the best available self-supervised rotation model from Kolesnikov et al. (2019). The number of clusters for CLUSTERING is selected from the set $\{50, 100, 200, 500, 1000\}$. The other unsupervised approaches do not have hyper-parameters.

Pre-trained and co-training approaches We employ the wide ResNet-50 v2 architecture with widening factor 16 (Zagoruyko & Komodakis, 2016) for the feature extractor F in the pre-trained approaches described in Section 3.1.

We optimize the loss in (2) using SGD for 65 epochs. The batch size is set to 2048, composed of B unlabeled examples and $2048 - B$ labeled examples. Following the recommendations from Goyal et al. (2017) for training with large batch size, we (i) set the learning rate to $0.1\frac{B}{256}$, and (ii) use linear learning rate warm-up during the initial 5 epochs. The learning rate is decayed twice with a factor of 10 at epoch 45 and epoch 55. The parameter γ in (2) is set to 0.5 and the number of unlabeled examples per batch B is 1536. The parameters γ and B are tuned on 0.1% labeled examples held out from the training set, the search space is $\{0.1, 0.5, 1.0\} \times \{1024, 1536, 1792\}$. The accuracy of the so-obtained classifier $c_{\text{S}^2\text{L}}(F(x))$ on the IMAGENET validation set is reported in Table 3. The parameter λ in the loss used for S^{2}GAN-CO in (3) is selected form the set $\{0.1, 0.2, 0.4\}$.

Self-supervision during GAN training For all approaches we use the recommend parameter $\alpha = 0.2$ from (Chen et al., 2019b) in (5) and do a small sweep for β in (4). For the values tried ($\{0.25, 0.5, 1.0, 2\}$) we do not see a large effect and use $\beta = 0.5$ for S^{3}GAN. For S^{3}GAN-CO we did not repeat the sweep, and instead used $\beta = 1.0$.

5. Results and discussion

Recall that the main goal of this work is to match (or outperform) the fully supervised BIGGAN in an unsupervised fashion, or with a small subset of labeled data. In the following, we discuss the advantages and drawbacks of the analyzed approaches with respect to this goal.

As a baseline, our reimplementation of BIGGAN obtains an FID of 8.4 and IS of 75.0, and hence reproduces the result reported by Brock et al. (2019) in terms of FID. We observed some differences in training dynamics, which we discuss in detail in Section 5.4.

5.1. Unsupervised approaches

The results for unsupervised approaches are summarized in Figure 7 and Table 2. The fully unsupervised RANDOM LABEL and SINGLE LABEL models both achieve a similar FID of ~ 25 and IS of ~ 20. This is a quite considerable gap compared to BIGGAN and indicates that additional supervision is necessary. We note that one of the three SINGLE LABEL models collapsed whereas all three RANDOM LABEL models trained stably for 250k generator iterations.

Pre-training a semantic representation using self-supervision and clustering the training data on this representation as done by CLUSTERING reduces the FID by about 10% and increases IS by about 10%. These results were obtained for 50 clusters, all other options led to worse results. While this performance is still considerably worse than that of BIGGAN this result is the current SOTA in unsupervised image generation (Chen et al. (2019b) report an FID of 33 for unsupervised generation).

Example images from the clustering are shown in Figures 14,

Table 2. Median FID and IS for the unsupervised approaches (see Table 14 in the appendix for mean and standard deviation).

	FID	IS
RANDOM LABEL	26.5	20.2
SINGLE LABEL	25.3	20.4
SINGLE LABEL (SS)	23.7	22.2
CLUSTERING	23.2	22.7
CLUSTERING (SS)	22.0	23.5

15, and 16 in the supplementary material. The clustering is clearly meaningful and groups similar objects within the same cluster. Furthermore, the objects generated by CLUSTERING conditionally on a given cluster index reflect the distribution of the training data belonging to the corresponding cluster. On the other hand, we can clearly observe multiple classes being present in the same cluster. This is to be expected when under-clustering to 50 clusters. Interestingly, clustering to many more clusters (say 500) yields results similar to SINGLE LABEL.

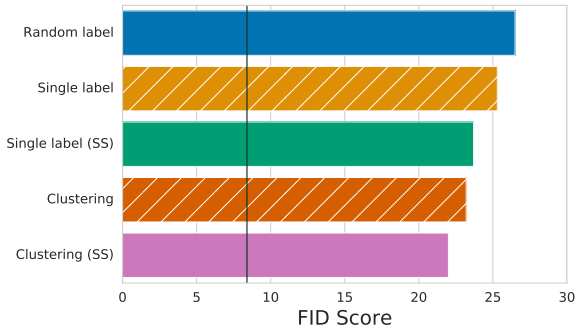


Figure 7. Median FID obtained by our unsupervised approaches. The vertical line indicates the median FID of our BIGGAN implementation which uses labels for all training images. While the gap between unsupervised and fully supervised approaches remains significant, using a pre-trained self-supervised representation (CLUSTERING) improves the sample quality compared to SINGLE LABEL and RANDOM LABEL, leading to a new SOTA in unsupervised generation on IMAGENET.

5.2. Semi-supervised approaches

Pre-trained The S^{2}GAN model where we use the classifier pre-trained with both a self-supervised and semi-supervised loss (cf. Section 3.1) matches the BIGGAN baseline when 20% of the labels are used and incurs a minor increase in FID when 10% and 5% are used (cf. Table 3). We stress that this is despite the fact that the classifier used to infer the labels has a top-1 accuracy of only 50%, 63%, and 71% for 5%, 10%, and 20% labeled data, respectively (cf. Table 3), compared to 100% of the original labels. The results are shown in Table 4 and Figure 8, and random samples as well as interpolations can be found in Figures 9–17 in the supplementary material.

Co-trained The results for our co-trained model S^{2}GAN-CO which trains a linear classifier in semi-supervised fashion on top of the discriminator representation during GAN training (cf. Section 3.2) are shown in Table 4. It can be seen that S^{2}GAN-CO outperforms all fully unsupervised approaches for all considered label percentages. While the gap between S^{2}GAN-CO with 5% labels and CLUSTERING in terms of FID is small, S^{2}GAN-CO has a considerably larger IS. When using 20% labeled training examples S^{2}GAN-CO obtains an FID of 13.9 and an IS of 49.2, which is remarkably close to BIGGAN and S^{2}GAN given the simplicity of the S^{2}GAN-CO approach. As the the percentage of labels decreases, the gap between S^{2}GAN and S^{2}GAN-CO increases.

Interestingly, S^{2}GAN-CO does not seem to train less stably than S^{2}GAN approaches even though it is forced to learn the classifier during GAN training. This is particularly remarkable as the BIGGAN-k% approaches, where we only retain the labeled data for training and discard all unlabeled data, *are very unstable and collapse after 60k to 120k iterations*, for all three random seeds and for both 10% and 20% labeled data.

5.3. Self-supervision during GAN training

So far we have seen that the pre-trained semi-supervised approach, namely S^{2}GAN, is able to achieve state-of-the-art performance for 20% labeled data. Here we investigate whether self-supervision during GAN training as described in Section 3.3 can lead to further improvements. Table 4 and Figure 8 show the experimental results for S^{3}GAN, namely S^{2}GAN coupled with self-supervision in the discriminator.

Self-supervision leads to a reduction in FID and increase in IS across all considered settings. In particular *we can match the state-of-the-art* BIGGAN *with only* 10% *of the labels and outperform it using* 20% *labels, both in terms of FID and IS.*

For S^{3}GAN the improvements due to self-supervision during GAN training in FID are considerable, around 10% in most of the cases. Tuning the parameter β of the discriminator self-supervision loss in (4) did not dramatically increase the benefits of self-supervision during GAN training, at least for the range of values considered. As shown in Tables 2 and 4, self-supervision during GAN training (with default parameters α, β) also leads to improvements by 5 to 10% for both S^{2}GAN-CO and SINGLE LABEL. In summary, self-

Table 3. Top-1 and top-5 error rate (%) on the IMAGENET validation set of $c_{\text{S}^2\text{L}}(F(x))$ using both self- and semi-supervised losses as described in Section 3.1. While the models are clearly not state-of-the-art compared to the fully supervised IMAGENET classification task, the quality of labels is sufficient to match and in some cases improve the state-of-the-art GAN natural image synthesis.

	LABELS		
METRIC	5%	10%	20%
TOP-1 ERROR	50.08	36.74	29.21
TOP-5 ERROR	26.94	16.04	10.33

Table 4. Pre-trained vs co-training approaches, and the effect of self-supervision during GAN training (see Table 12 in the appendix for mean and standard deviation). While co-training approaches outperform fully unsupervised approaches, they are clearly outperformed by the pre-trained approaches. Self-supervision during GAN training helps in all cases.

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	10.8	8.9	8.4	57.6	73.4	77.4
S^2GAN-CO	21.8	17.7	13.9	30.0	37.2	49.2
S^3GAN	10.4	8.0	7.7	59.6	78.7	83.1
S^3GAN-CO	20.2	16.6	12.7	31.0	38.5	53.1

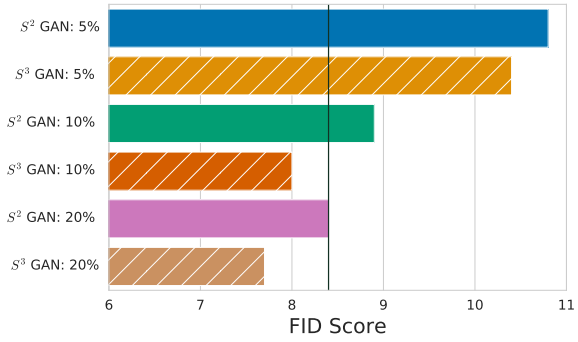


Figure 8. The vertical line indicates the median FID of our BIGGAN implementation which uses all labeled data. The proposed S^3GAN approach is able to match the performance of the state-of-the-art BIGGAN model using 10% of the ground-truth labels and outperforms it using 20%.

supervision during GAN training with default parameters leads to a stable improvement across all approaches.

5.4. Other insights

Effect of soft labels A design choice available to practitioners is whether to use hard labels (i.e., the argmax over the logits), or soft labels (softmax over the logits) for S^2GAN and S^3GAN (recall that we use soft labels by default for S^2GAN-CO and S^3GAN-CO). Our initial expectation was that soft labels should help when very little labeled data is available, as soft labels carry more information which can potentially be exploited by the projection discriminator. Surprisingly, the results presented in Table 5 show clearly that the opposite is true. Our current hypothesis is that this is due to the way labels are incorporated in the projection discriminator, but we do not have empirical evidence yet.

Optimization dynamics Brock et al. (2019) report the FID and IS of the model *just before the collapse*, which can be seen as a form of early stopping. In contrast, we manage to stably train the proposed models for 250k generator iterations. In particular, we also observe stable training for our "vanilla" BIGGAN implementation. The evolution of the FID and IS as a function of the training steps is shown in Figure 21 in the appendix. At this point we can only speculate about the origin of this difference. We finally note that by tuning the learning rate we obtained slightly different (but still stable) training dynamics in terms of IS, achieving FID 6.9 and IS 98 for S^3GAN with 20% labels.

Table 5. Training with hard (predicted) labels leads to better models than training with soft (predicted) labels (see Table 13 in the appendix for mean and standard deviation).

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	10.8	8.9	8.4	57.6	73.4	77.4
+SOFT	15.4	12.9	10.4	40.3	49.8	62.1

Higher resolution and going below 5% labels Training these models at higher resolution becomes computationally harder and it necessitates tuning the learning rate. We trained several S^3GAN models at 256×256 resolution and show the resulting samples in Figures 12–13 and interpolations in Figures 19–20. We also conducted S^3GAN experiments in which only 2.5% of the labels are used and observed FID of 13.6 and IS of 46.3. This indicates that given a small number of samples one can significantly outperform the unsupervised approaches (c.f. Figure 7).

6. Conclusion and future Work

In this work we investigated several avenues to reduce the appetite for labeled data in state-of-the-art GANs. We showed that recent advances in self and semi-supervised learning can be used to achieve a new state of the art, both for unsupervised and supervised natural image synthesis.

We believe that this is a great first step towards the ultimate goal of few-shot high-fidelity image synthesis. There are several important directions for future work: (i) investigating the applicability of these techniques for even larger and more diverse data sets, and (ii) investigating the impact of other self- and semi-supervised approaches on the model quality. (iii) investigating the impact of self-supervision in other deep generative models. Finally, we would like to emphasize that further progress might be hindered by the engineering challenges related to training large-scale generative adversarial networks. To help alleviate this issue and to foster reproducibility, we have open-sourced all the code used for the experiments.

Acknowledgments

We would like to thank Ting Chen and Neil Houlsby for fruitful discussions on self-supervision and its application to GANs. We would like to thank Lucas Beyer, Alexander Kolesnikov, and Avital Oliver for helpful discussions on self-supervised semi-supervised learning. We would like to thank Karol Kurach and Marcin Michalski their major contributions the Compare GAN library. We would also like to thank the BigGAN team (Andy Brock, Jeff Donahue, and Karen Simonyan) for their insights into training GANs on TPUs. Finally, we are grateful for the support of members of the Google Brain team in Zurich. This work was partially done while Michael Tschannen was at Google Research.

References

Agrawal, P., Carreira, J., and Malik, J. Learning to see by moving. In *International Conference on Computer Vision*, 2015.

Arbel, M., Sutherland, D., Bińkowski, M., and Gretton, A. On gradient regularizers for MMD GANs. In *Advances in Neural Information Processing Systems*, 2018.

Barratt, S. and Sharma, R. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

Brock, A., Donahue, J., and Simonyan, K. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision*, 2018.

Chen, T., Lucic, M., Houlsby, N., and Gelly, S. On self modulation for generative adversarial networks. In *International Conference on Learning Representations*, 2019a.

Chen, T., Zhai, X., Ritter, M., Lucic, M., and Houlsby, N. Self-supervised GANs via auxiliary rotation loss. In *Computer Vision and Pattern Recognition*, 2019b.

De Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 2017.

Deng, Z., Zhang, H., Liang, X., Yang, L., Xu, S., Zhu, J., and Xing, E. P. Structured generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2017.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision*, 2015.

Dumoulin, V., Shlens, J., and Kudlur, M. A learned representation for artistic style. In *International Conference on Learning Representations*, 2017.

Gan, Z., Chen, L., Wang, W., Pu, Y., Zhang, Y., Liu, H., Li, C., and Carin, L. Triangle generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2017.

Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Klambauer, G., and Hochreiter, S. GANs trained by a two time-scale update rule converge to a Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Jang, E., Devin, C., Vanhoucke, V., and Levine, S. Grasp2Vec: Learning object representations from self-supervised grasping. In *Conference on Robot Learning*, 2018.

Kalchbrenner, N., van den Oord, A., Simonyan, K., Danihelka, I., Vinyals, O., Graves, A., and Kavukcuoglu, K. Video pixel networks. In *International Conference on Machine Learning*, 2017.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Computer Vision and Pattern Recognition*, 2019.

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting self-supervised visual representation learning. In *Computer Vision and Pattern Recognition*, 2019.

Kurach, K., Lucic, M., Zhai, X., Michalski, M., and Gelly, S. The GAN Landscape: Losses, architectures, regularization, and normalization. In *International Conference on Machine Learning*, 2019.

Lee, H.-Y., Huang, J.-B., Singh, M., and Yang, M.-H. Unsupervised representation learning by sorting sequences. In *International Conference on Computer Vision*, 2017.

Li, C., Xu, T., Zhu, J., and Zhang, B. Triple Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2017.

Lucic, M., Kurach, K., Michalski, M., Gelly, S., and Bousquet, O. Are GANs created equal? A large-scale study. In *Advances in Neural Information Processing Systems*, 2018.

Menick, J. and Kalchbrenner, N. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *International Conference on Learning Representations*, 2019.

Miyato, T. and Koyama, M. cgans with projection discriminator. In *International Conference on Learning Representations*, 2018.

Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations*, 2018.

Mundhenk, T. N., Ho, D., and Chen, B. Y. Improvements to context based self-supervised learning. In *Computer Vision and Pattern Recognition*, 2018.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, 2016.

Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In *International Conference on Machine Learning*, 2017.

Pinto, L. and Gupta, A. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *IEEE International Conference on Robotics and Automation*, 2016.

Sajjadi, M. S., Bachem, O., Lucic, M., Bousquet, O., and Gelly, S. Assessing generative models via precision and recall. In *Advances in Neural Information Processing Systems*, 2018.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, 2016.

Sculley, D. Web-scale k-means clustering. In *International Conference on World Wide Web*, 2010.

Springenberg, J. T. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *International Conference on Learning Representations*, 2016.

Sricharan, K., Bala, R., Shreve, M., Ding, H., Saketh, K., and Sun, J. Semi-supervised conditional GANs. *arXiv preprint arXiv:1708.05789*, 2017.

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016a.

van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 2016b.

Zagoruyko, S. and Komodakis, N. Wide residual networks. *British Machine Vision Conference*, 2016.

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. S^4L: Self-Supervised Semi-Supervised Learning. *arXiv preprint arXiv:1905.03670*, 2019.

Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, 2019.

A. Additional samples and interpolations

Figure 9. Samples obtained from S^{3}GAN (20% labels, 128×128) when interpolating in the latent space (left to right).

Figure 10. Samples obtained from S^{3}GAN (20% labels, 128×128) when interpolating in the latent space (left to right).

Figure 11. Samples obtained from S^{3}GAN (20% labels, 128×128) when interpolating in the latent space (left to right).

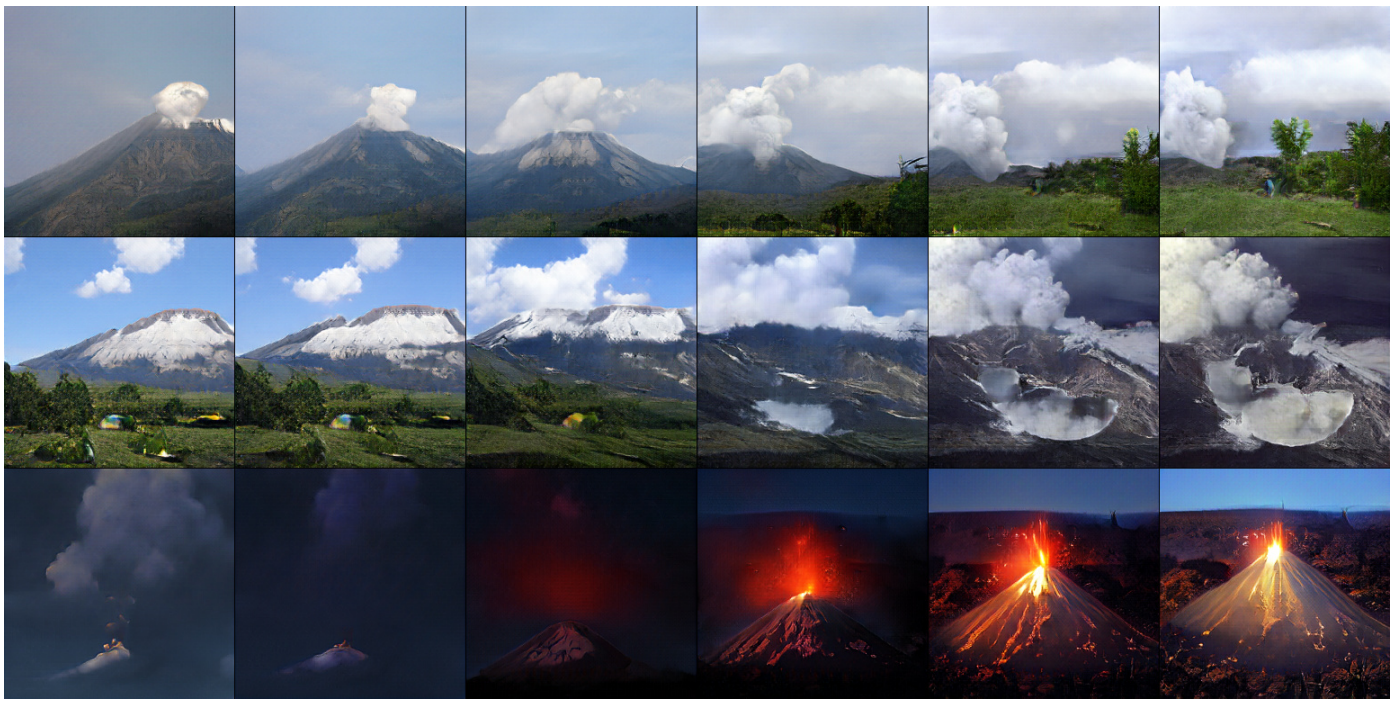


Figure 12. Samples obtained from S^{3}GAN (10% labels, 256×256) when interpolating in the latent space (left to right).

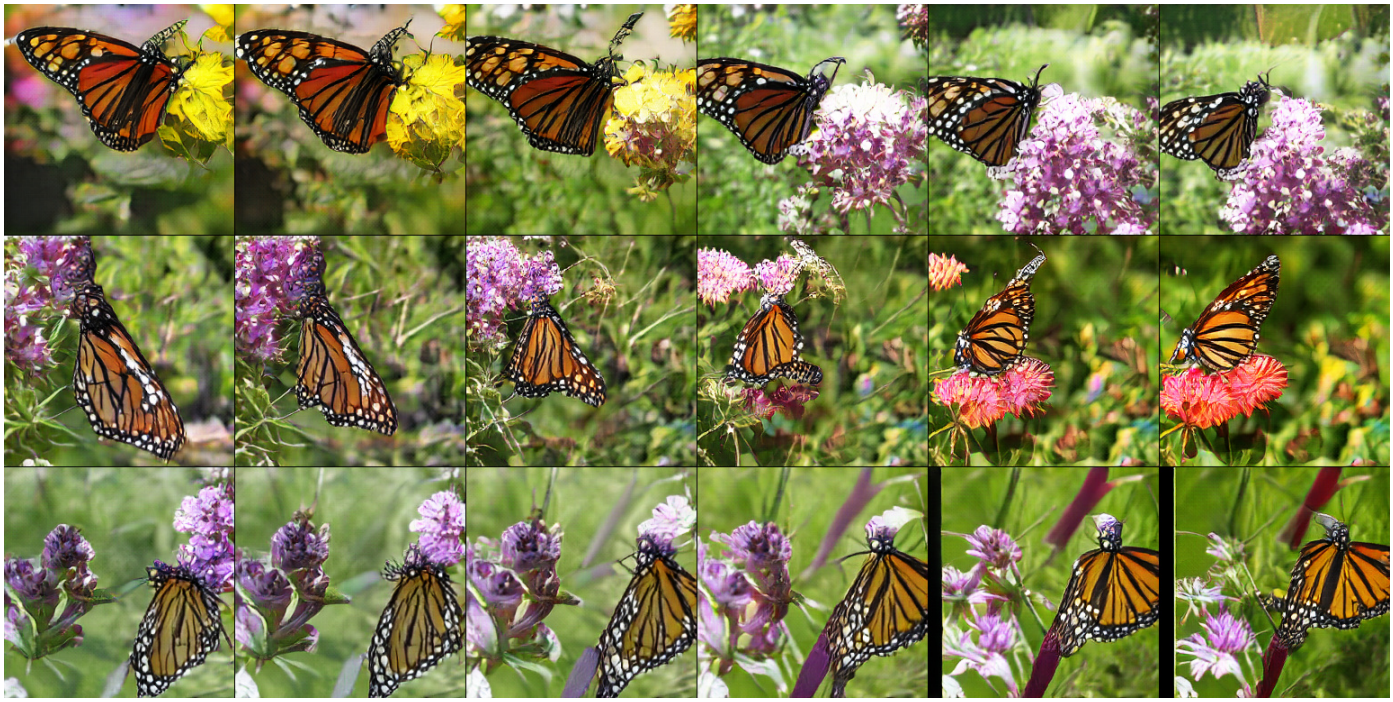


Figure 13. Samples obtained from S^3GAN (10% labels, 256×256) when interpolating in the latent space (left to right).

Real images.

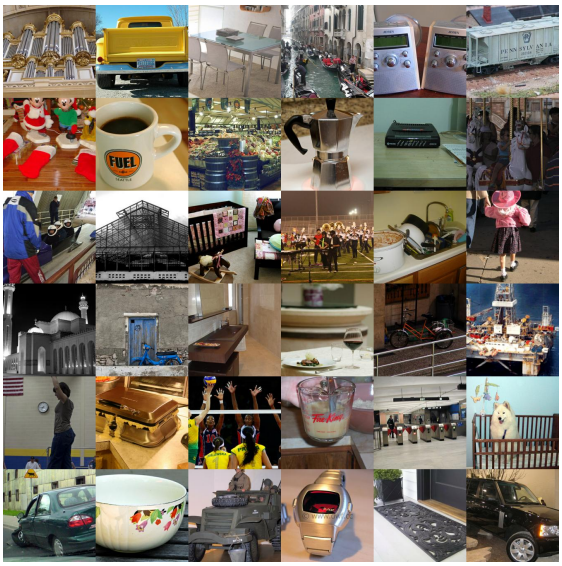
Generated images.

Figure 14. Real and generated images (128×128) for one of the 50 clusters produced by CLUSTERING. Both real and generated images show mostly underwater scenes.

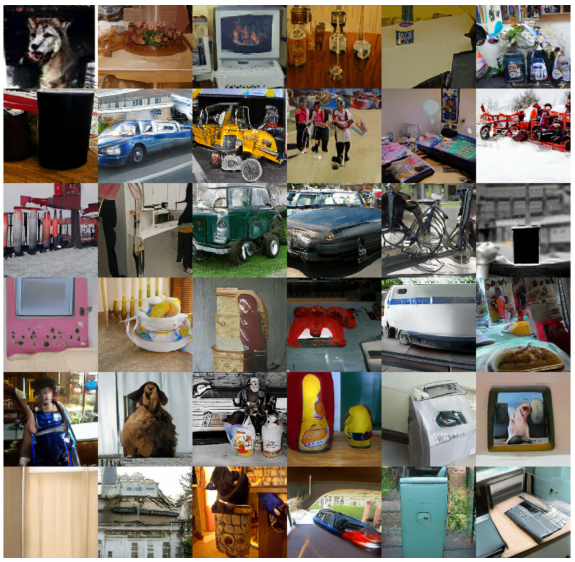
Real images.

Generated images.

Figure 15. Real and generated images (128 × 128) for one of the 50 clusters produced by CLUSTERING. Both real and generated images show mostly outdoor scenes featuring different animals.



Real images.



Generated images.

Figure 16. Real and generated images (128 × 128) for one of the 50 clusters produced by CLUSTERING. In contrast to the examples shown in Figures 14 and 15 the clusters show diverse indoor and outdoor scenes.

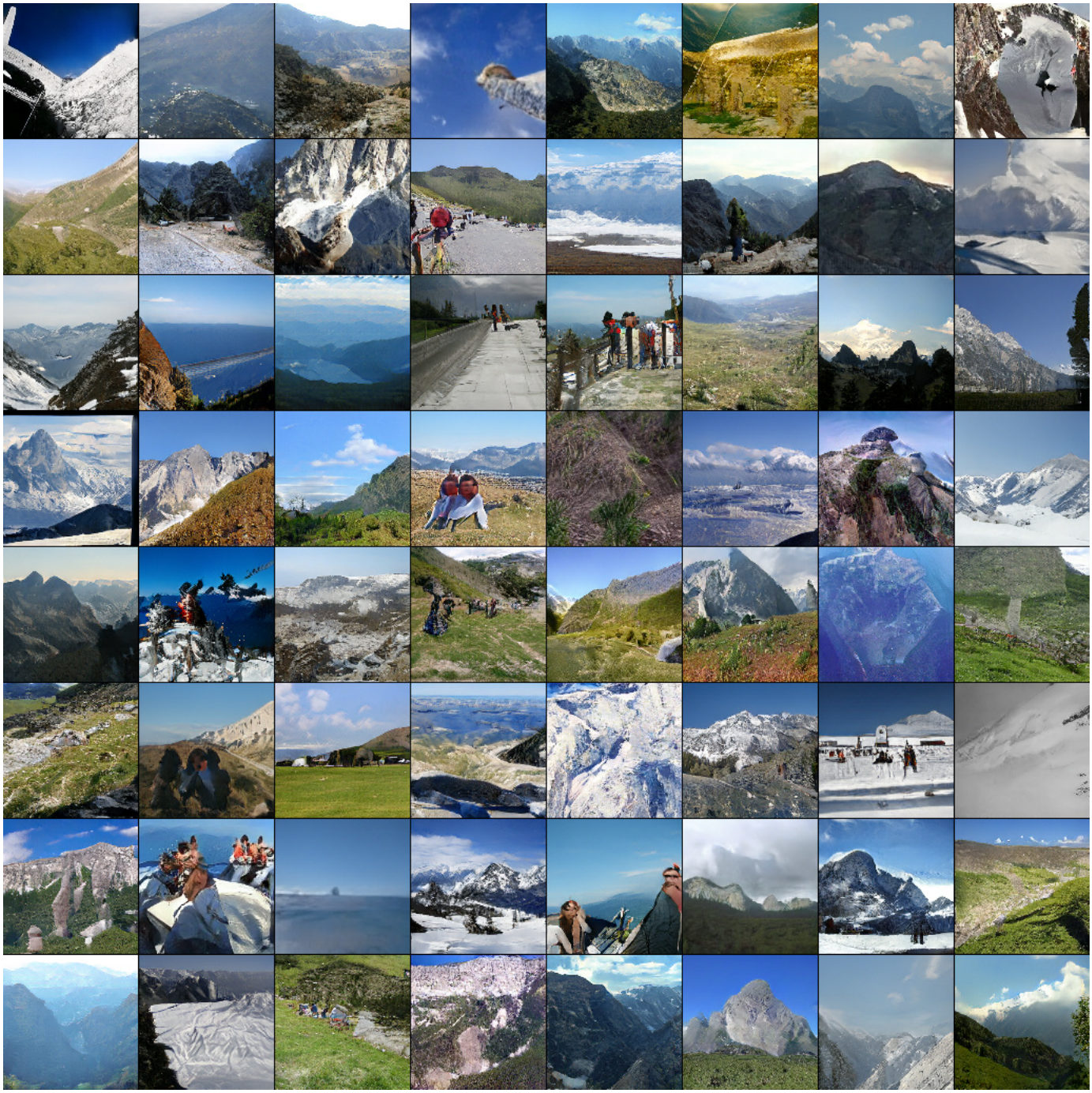


Figure 17. Samples generated by S^3GAN (20% labels, 128×128) for a single class. The model captures the great diversity within the class. Human faces and more dynamic scenes present challenges.

Figure 18. Generated samples by S^{3}GAN (20% labels, 128 $\times$ 128) for different classes. The model correctly learns the different classes and we do not observe class leakage.

Figure 19. Generated samples by S^{3}GAN (10% labels, 256×256) for a single class. The model captures the diversity within the class.

Figure 20. Generated samples by S^{3}GAN (10% labels, 256×256) for a single class. The model captures the diversity within the class.

B. Architectural details

The ResNet architecture implemented following Brock et al. (2019) is described in Tables 6 and 8. We use the abbreviations RS for resample, BN for batch normalization, and cBN for conditional BN (Dumoulin et al., 2017; De Vries et al., 2017). In the resample column, we indicate downscale(D)/upscale(U)/none(-) setting and in the spectral norm column shows whether spectral normalization is applied to all weights in the layer. In Table 8, y stands for the labels and $\boldsymbol{h}$ is the output from the layer before (i.e., the pre-logit layer). Tables 7 and 9 show ResBlock details. The addition layer merges the shortcut path and the convolution path by adding them. h and w are the input height and width of the ResBlock, c_i and c_o are the input channels and output channels for a ResBlock. For the last ResBlock in the discriminator without downsampling, we simply drop the shortcut layer from ResBlock. We list all the trainable variables and their shape in Tables 10 and 11.

Table 6. ResNet generator architecture. "ch" represents the channel width multiplier and is set to 96.

LAYER	RS	SN	OUTPUT
$\boldsymbol{z} \sim \mathcal{N}(0, 1)$	-	-	120
Dense	-	-	$4 \times 4 \times 16 \cdot ch$
ResBlock	U	SN	$8 \times 8 \times 16 \cdot ch$
ResBlock	U	SN	$16 \times 16 \times 8 \cdot ch$
ResBlock	U	SN	$32 \times 32 \times 4 \cdot ch$
ResBlock	U	SN	$64 \times 64 \times 2 \cdot ch$
Non-local block	-	-	$64 \times 64 \times 2 \cdot ch$
ResBlock	U	SN	$128 \times 128 \times 1 \cdot ch$
BN, ReLU	-	-	$128 \times 128 \times 3$
Conv $[3, 3, 1]$	-	-	$128 \times 128 \times 3$
Tanh	-	-	$128 \times 128 \times 3$

Table 7. ResBlock generator with upsample.

LAYER	KERNEL	RS	OUTPUT
Shortcut	$[1, 1, 1]$	U	$2h \times 2w \times c_o$
cBN, ReLU	-	-	$h \times w \times c_i$
Conv	$[3, 3, 1]$	U	$2h \times 2w \times c_o$
cBN, ReLU	-	-	$2h \times 2w \times c_o$
Conv	$[3, 3, 1]$	-	$2h \times 2w \times c_o$
Addition	-	-	$2h \times 2w \times c_o$

Table 8. ResNet discriminator architecture. "ch" represents the channel width multiplier and is set to 96. Spectral normalization is applied to all layers.

LAYER	RS	OUTPUT
Input image	-	$128 \times 128 \times 3$
ResBlock	D	$64 \times 64 \times 1 \cdot ch$
Non-local block	-	$64 \times 64 \times 1 \cdot ch$
ResBlock	D	$32 \times 32 \times 2 \cdot ch$
ResBlock	D	$16 \times 16 \times 4 \cdot ch$
ResBlock	D	$8 \times 8 \times 8 \cdot ch$
ResBlock	D	$4 \times 4 \times 16 \cdot ch$
ResBlock (without shortcut)	-	$4 \times 4 \times 16 \cdot ch$
ReLU	-	$4 \times 4 \times 16 \cdot ch$
Global sum pooling	-	$1 \times 1 \times 16 \cdot ch$
Sum(embed(y)$\cdot \boldsymbol{h}$)+(dense$\rightarrow 1$)	-	1

Table 9. ResBlock discriminator with downsample.

LAYER	KERNEL	RS	OUTPUT
Shortcut	$[1, 1, 1]$	D	$h/2 \times w/2 \times c_o$
ReLU	-	-	$h \times w \times c_i$
Conv	$[3, 3, 1]$	-	$h \times w \times c_o$
ReLU	-	-	$h \times w \times c_o$
Conv	$[3, 3, 1]$	D	$h/2 \times w/2 \times c_o$
Addition	-	-	$h/2 \times w/2 \times c_o$

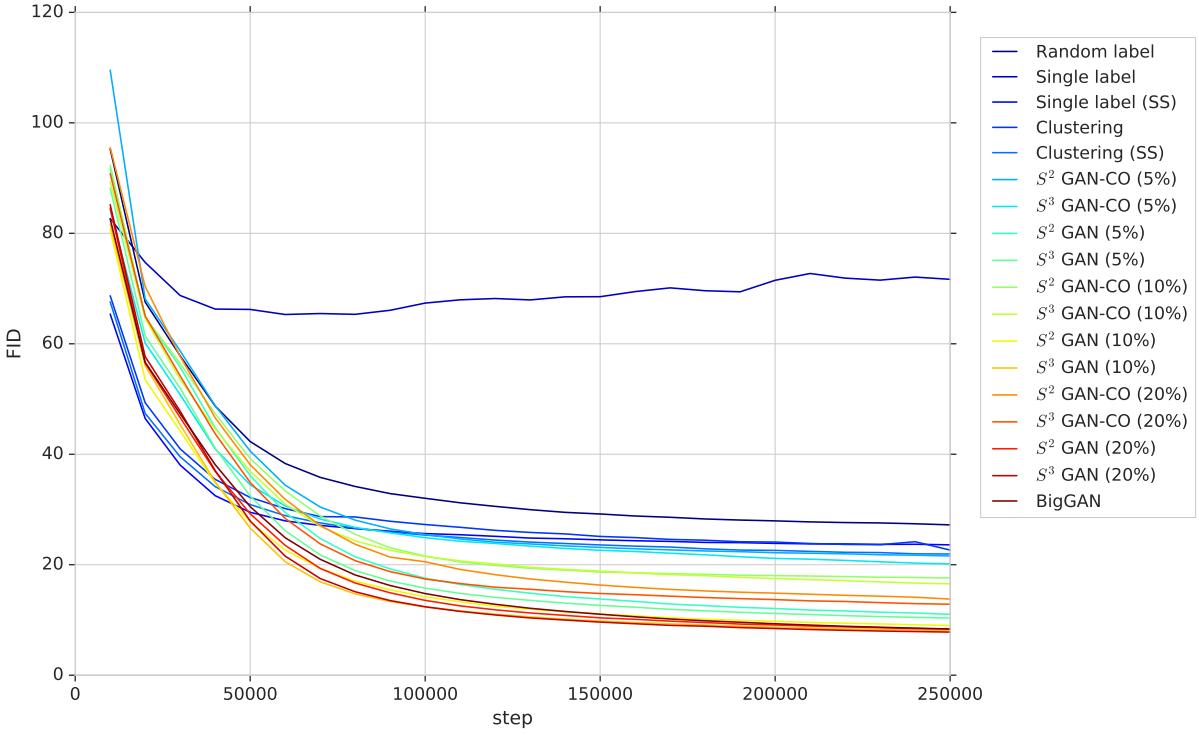
NAME	SHAPE	SIZE
generator/embed_y/kernel:0	(1000, 128)	128,000
generator/fc_noise/kernel:0	(20, 24576)	491,520
generator/fc_noise/bias:0	(24576,)	24,576
generator/B1/bn1/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B1/bn1/condition/beta/kernel:0	(148, 1536)	227,328
generator/B1/up_conv1/kernel:0	(3, 3, 1536, 1536)	21,233,664
generator/B1/up_conv1/bias:0	(1536,)	1,536
generator/B1/bn2/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B1/bn2/condition/beta/kernel:0	(148, 1536)	227,328
generator/B1/same_conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
generator/B1/same_conv2/bias:0	(1536,)	1,536
generator/B1/up_conv_shortcut/kernel:0	(1, 1, 1536, 1536)	2,359,296
generator/B1/up_conv_shortcut/bias:0	(1536,)	1,536
generator/B2/bn1/condition/gamma/kernel:0	(148, 1536)	227,328
generator/B2/bn1/condition/beta/kernel:0	(148, 1536)	227,328
generator/B2/up_conv1/kernel:0	(3, 3, 1536, 768)	10,616,832
generator/B2/up_conv1/bias:0	(768,)	768
generator/B2/bn2/condition/gamma/kernel:0	(148, 768)	113,664
generator/B2/bn2/condition/beta/kernel:0	(148, 768)	113,664
generator/B2/same_conv2/kernel:0	(3, 3, 768, 768)	5,308,416
generator/B2/same_conv2/bias:0	(768,)	768
generator/B2/up_conv_shortcut/kernel:0	(1, 1, 1536, 768)	1,179,648
generator/B2/up_conv_shortcut/bias:0	(768,)	768
generator/B3/bn1/condition/gamma/kernel:0	(148, 768)	113,664
generator/B3/bn1/condition/beta/kernel:0	(148, 768)	113,664
generator/B3/up_conv1/kernel:0	(3, 3, 768, 384)	2,654,208
generator/B3/up_conv1/bias:0	(384,)	384
generator/B3/bn2/condition/gamma/kernel:0	(148, 384)	56,832
generator/B3/bn2/condition/beta/kernel:0	(148, 384)	56,832
generator/B3/same_conv2/kernel:0	(3, 3, 384, 384)	1,327,104
generator/B3/same_conv2/bias:0	(384,)	384
generator/B3/up_conv_shortcut/kernel:0	(1, 1, 768, 384)	294,912
generator/B3/up_conv_shortcut/bias:0	(384,)	384
generator/B4/bn1/condition/gamma/kernel:0	(148, 384)	56,832
generator/B4/bn1/condition/beta/kernel:0	(148, 384)	56,832
generator/B4/up_conv1/kernel:0	(3, 3, 384, 192)	663,552
generator/B4/up_conv1/bias:0	(192,)	192
generator/B4/bn2/condition/gamma/kernel:0	(148, 192)	28,416
generator/B4/bn2/condition/beta/kernel:0	(148, 192)	28,416
generator/B4/same_conv2/kernel:0	(3, 3, 192, 192)	331,776
generator/B4/same_conv2/bias:0	(192,)	192
generator/B4/up_conv_shortcut/kernel:0	(1, 1, 384, 192)	73,728
generator/B4/up_conv_shortcut/bias:0	(192,)	192
generator/non_local_block/conv2d_theta/kernel:0	(1, 1, 192, 24)	4,608
generator/non_local_block/conv2d_phi/kernel:0	(1, 1, 192, 24)	4,608
generator/non_local_block/conv2d_g/kernel:0	(1, 1, 192, 96)	18,432
generator/non_local_block/sigma:0	()	1
generator/non_local_block/conv2d_attn_g/kernel:0	(1, 1, 96, 192)	18,432
generator/B5/bn1/condition/gamma/kernel:0	(148, 192)	28,416
generator/B5/bn1/condition/beta/kernel:0	(148, 192)	28,416
generator/B5/up_conv1/kernel:0	(3, 3, 192, 96)	165,888
generator/B5/up_conv1/bias:0	(96,)	96
generator/B5/bn2/condition/gamma/kernel:0	(148, 96)	14,208
generator/B5/bn2/condition/beta/kernel:0	(148, 96)	14,208
generator/B5/same_conv2/kernel:0	(3, 3, 96, 96)	82,944
generator/B5/same_conv2/bias:0	(96,)	96
generator/B5/up_conv_shortcut/kernel:0	(1, 1, 192, 96)	18,432
generator/B5/up_conv_shortcut/bias:0	(96,)	96
generator/final_norm/gamma:0	(96,)	96
generator/final_norm/beta:0	(96,)	96
generator/final_conv/kernel:0	(3, 3, 96, 3)	2,592
generator/final_conv/bias:0	(3,)	3

Table 10. Tensor-level description of the generator containing a total of 70,433,988 parameters.

NAME	SHAPE	SIZE
discriminator/B1/same_conv1/kernel:0	(3, 3, 3, 96)	2,592
discriminator/B1/same_conv1/bias:0	(96,)	96
discriminator/B1/down_conv2/kernel:0	(3, 3, 96, 96)	82,944
discriminator/B1/down_conv2/bias:0	(96,)	96
discriminator/B1/down_conv_shortcut/kernel:0	(1, 1, 3, 96)	288
discriminator/B1/down_conv_shortcut/bias:0	(96,)	96
discriminator/non_local_block/conv2d_theta/kernel:0	(1, 1, 96, 12)	1,152
discriminator/non_local_block/conv2d_phi/kernel:0	(1, 1, 96, 12)	1,152
discriminator/non_local_block/conv2d_g/kernel:0	(1, 1, 96, 48)	4,608
discriminator/non_local_block/sigma:0	()	1
discriminator/non_local_block/conv2d_attn_g/kernel:0	(1, 1, 48, 96)	4,608
discriminator/B2/same_conv1/kernel:0	(3, 3, 96, 192)	165,888
discriminator/B2/same_conv1/bias:0	(192,)	192
discriminator/B2/down_conv2/kernel:0	(3, 3, 192, 192)	331,776
discriminator/B2/down_conv2/bias:0	(192,)	192
discriminator/B2/down_conv_shortcut/kernel:0	(1, 1, 96, 192)	18,432
discriminator/B2/down_conv_shortcut/bias:0	(192,)	192
discriminator/B3/same_conv1/kernel:0	(3, 3, 192, 384)	663,552
discriminator/B3/same_conv1/bias:0	(384,)	384
discriminator/B3/down_conv2/kernel:0	(3, 3, 384, 384)	1,327,104
discriminator/B3/down_conv2/bias:0	(384,)	384
discriminator/B3/down_conv_shortcut/kernel:0	(1, 1, 192, 384)	73,728
discriminator/B3/down_conv_shortcut/bias:0	(384,)	384
discriminator/B4/same_conv1/kernel:0	(3, 3, 384, 768)	2,654,208
discriminator/B4/same_conv1/bias:0	(768,)	768
discriminator/B4/down_conv2/kernel:0	(3, 3, 768, 768)	5,308,416
discriminator/B4/down_conv2/bias:0	(768,)	768
discriminator/B4/down_conv_shortcut/kernel:0	(1, 1, 384, 768)	294,912
discriminator/B4/down_conv_shortcut/bias:0	(768,)	768
discriminator/B5/same_conv1/kernel:0	(3, 3, 768, 1536)	10,616,832
discriminator/B5/same_conv1/bias:0	(1536,)	1,536
discriminator/B5/down_conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B5/down_conv2/bias:0	(1536,)	1,536
discriminator/B5/down_conv_shortcut/kernel:0	(1, 1, 768, 1536)	1,179,648
discriminator/B5/down_conv_shortcut/bias:0	(1536,)	1,536
discriminator/B6/same_conv1/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B6/same_conv1/bias:0	(1536,)	1,536
discriminator/B6/same_conv2/kernel:0	(3, 3, 1536, 1536)	21,233,664
discriminator/B6/same_conv2/bias:0	(1536,)	1,536
discriminator/final_fc/kernel:0	(1536, 1)	1,536
discriminator/final_fc/bias:0	(1,)	1
discriminator_projection/kernel:0	(1000, 1536)	1,536,000

Table 11. Tensor-level description of the discriminator containing a total of 87,982,370 parameters.

C. FID and IS training curves



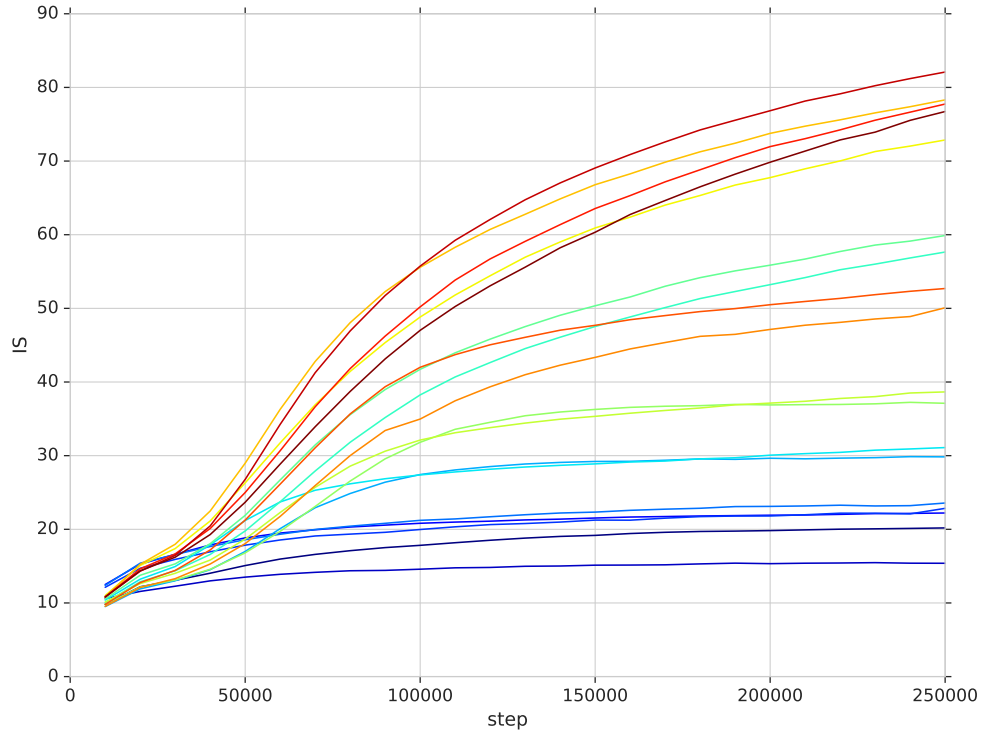


Figure 21. Mean FID and IS (3 runs) on ImageNet (128×128) for the models considered in this paper, as a function of the number of generator steps. All models train stably, except SINGLE LABEL (where one run collapsed).

D. FID and IS: Mean and standard deviations

Table 12. Pre-trained vs co-training approaches, and the effect of self-supervision during GAN training. While co-training approaches outperform fully unsupervised approaches, they are clearly outperformed by the pre-trained approaches. Self-supervision during GAN training helps in all cases.

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	11.0±0.31	9.0±0.30	8.4±0.02	57.6±0.86	72.9±1.41	77.7±1.24
S^2GAN-CO	21.6±0.64	17.6±0.27	13.8±0.48	29.8±0.21	37.1±0.54	50.1±1.45
S^3GAN	10.3±0.16	8.1±0.14	7.8±0.20	59.9±0.74	78.3±1.08	82.1±1.89
S^3GAN-CO	20.2±0.14	16.5±0.12	12.8±0.51	31.1±0.18	38.7±0.36	52.7±1.08

Table 13. Training with hard (predicted) labels leads to better models than training with soft (predicted) labels.

	FID			IS		
	5%	10%	20%	5%	10%	20%
S^2GAN	11.0±0.31	9.0±0.30	8.4±0.02	57.6±0.86	72.9±1.41	77.7±1.24
S^2GAN SOFT	15.6±0.58	13.3±1.71	11.3±1.42	40.1±0.97	49.3±4.67	58.5±5.84

Table 14. Mean FID and IS for the unsupervised approaches.

	FID	IS
CLUSTERING	22.7±0.80	22.8±0.42
CLUSTERING(SS)	21.9±0.08	23.6±0.19
RANDOM LABEL	27.2±1.46	20.2±0.33
SINGLE LABEL	71.7±66.32	15.4±7.57
SINGLE LABEL(SS)	23.6±0.14	22.2±0.10