Self-Supervised GANs via Auxiliary Rotation Loss

Ting Chen
University of California, Los Angeles
tingchen@cs.ucla.edu

Xiaohua Zhai
Google Brain
xzhai@google.com

Marvin Ritter
Google Brain
marvinritter@google.com

Mario Lucic
Google Brain
lucic@google.com

Neil Houlsby
Google Brain
neilhoulsby@google.com

Abstract

Conditional GANs are at the forefront of natural image synthesis. The main drawback of such models is the necessity for labeled data. In this work we exploit two popular unsupervised learning techniques, adversarial training and self-supervision, and take a step towards bridging the gap between conditional and unconditional GANs. In particular, we allow the networks to collaborate on the task of representation learning, while being adversarial with respect to the classic GAN game. The role of self-supervision is to encourage the discriminator to learn meaningful feature representations which are not forgotten during training. We test empirically both the quality of the learned image representations, and the quality of the synthesized images. Under the same conditions, the self-supervised GAN attains a similar performance to state-of-the-art conditional counterparts. Finally, we show that this approach to fully unsupervised learning can be scaled to attain an FID of 23.4 on unconditional IMAGENET generation.1

1. Introduction

Generative Adversarial Networks (GANs) are a class of unsupervised generative models [1]. GANs involve training a generator and discriminator model in an adversarial game, such that the generator learns to produce samples from a desired data distribution. Training GANs is challenging because it involves searching for a Nash equilibrium of a non-convex game in a high-dimensional parameter space. In practice, GANs are typically trained using alternating stochastic gradient descent which is often unstable and lacks theoretical guarantees [2]. Consequently, training may exhibit instability, divergence, cyclic behavior, or mode collapse [3]. As a result, many techniques to stabilize GAN training have been proposed [4, 5, 6, 7, 8, 9, 10]. A major contributor to training instability is the fact that the generator and discriminator learn in a non-stationary environment. In particular, the discriminator is a classifier for which the distribution of one class (the fake samples) shifts as the generator changes during training. In non-stationary online environments, neural networks forget previous tasks [11, 12, 13]. If the discriminator forgets previous classification boundaries, training may become unstable or cyclic. This issue is usually addressed either by reusing old samples or by applying continual learning techniques [14, 15, 16, 17, 18, 19]. These issues become more prominent in the context of complex data sets. A key technique in these settings is conditioning [9, 20, 21, 22] whereby both the generator and discriminator have access to labeled data. Arguably, augmenting the discriminator with supervised information encourages it to learn more stable representations which opposes catastrophic forgetting. Furthermore, learning the conditional model for each class is easier than learning the joint distribution. The main drawback in this setting is the necessity for labeled data. Even when labeled data is available, it is usually sparse and covers only a limited amount of high level abstractions.

Motivated by the aforementioned challenges, our goal is to show that one can recover the benefits of conditioning, without requiring labeled data. To ensure that the representations learned by the discriminator are more stable and useful, we add an auxiliary, self-supervised loss to the discriminator. This leads to more stable training because the dependence of the discriminator’s representations on the quality of the generator’s output is reduced. We introduce a novel model – the self-supervised GAN – in which the generator and discriminator collaborate on the task of representation learning, and compete on the generative task.

Our contributions We present an unsupervised generative model that combines adversarial training with self-supervised learning. Our model recovers the benefits of conditional GANs, but requires no labeled data. In particu-
lar, under the same training conditions, the self-supervised GAN closes the gap in natural image synthesis between unconditional and conditional models. Within this setting the quality of discriminator’s representations is greatly increased which might be of separate interest in the context of transfer learning. A large-scale implementation of the model leads to promising results on unconditional IMAGENET generation, a task considered daunting by the community. We believe that this work is an important step in the direction of high quality, fully unsupervised, natural image synthesis.

2. A Key Issue: Discriminator Forgetting

The original value function for GAN training is [1]:

\[
V(G, D) = \mathbb{E}_{x \sim P_{\text{data}}(x)} \left[ \log P_D(S = 1 \mid x) \right] + \mathbb{E}_{z \sim P_{G}(z)} \left[ \log(1 - P_D(S = 0 \mid z)) \right]
\]

where \( P_{\text{data}} \) is the true data distribution, and \( P_G \) is the distribution induced by transforming a simple distribution \( z \sim P(z) \) using the deterministic mapping given by the generator, \( x = G(z) \), and \( P_D \) is the discriminator’s Bernoulli distribution over the labels (true or fake). In the original minimax setting the generator maximizes Equation 1 with respect to it’s parameters, while the discriminator minimizes it. Training is typically performed via alternating stochastic gradient descent. Therefore, at iteration \( t \) during training, the discriminator classifies samples as coming from \( P_{\text{data}} \) or \( P_{G}^{(t)} \). As the parameters of \( G \) change, the distribution \( P_{G}^{(t)} \) changes, which implies a non-stationary online learning problem for the discriminator.

This challenge has received a great deal of attention and explicit temporal dependencies have been proposed to improve training in this setting [2, 15, 17, 19]. Furthermore, in online learning of non-convex functions, neural networks have been shown to forget previous tasks [11, 12, 13]. In the context of GANs, learning varying levels of detail, structure, and texture, can be considered different tasks. For example, if the generator first learns the global structure, the discriminator will naturally try to build a representation which allows it to efficiently penalize the generator based only on the differences in global structure, or the lack of local structure. As such, one source of instability in training is that the discriminator is not incentivised to maintain a useful data representation as long as the current representation is useful to discriminate between the classes.

Further evidence can be gathered by considering the generator and discriminator at convergence. Indeed, Goodfellow et al. [1] show that the optimal discriminator estimates the likelihood ratio between the generated and real data distributions. Therefore, given a perfect generator, where \( P_G = P_{\text{data}} \), the optimal discriminator simply outputs 0.5, which is a constant and doesn’t depend on the input. Hence, this discriminator would have no requirement to retain meaningful representations. Furthermore, if regularization is applied, the discriminator might ignore all but the minor features which distinguish real and fake data.

We demonstrate the impact of discriminator forgetting in two settings. (1) A simple scenario shown in Figure 3(a), and, (2) during the training of a GAN shown in Figure 2. In the first case a classifier is trained sequentially on 1-vs.-all
classification tasks on each of the ten classes in CIFAR10. It is trained for 1k iterations on each task before switching to the next. At 10k iterations the training cycle repeats from the first task. Figure 2(a) shows substantial forgetting, despite the tasks being similar. Each time the task switches, the classifier accuracy drops substantially. After 10k iterations, the cycle of tasks repeats, and the accuracy is the same as the first cycle. No useful information is carried across tasks. This demonstrates that the model does not retain generalizable representations in this non-stationary environment. In the second setting shown in Figure 2 we observe a similar effect during GAN training. Every 100k iterations, the discriminator representations are evaluated on ImageNet classification; the full protocol is described in Section 4.4. During training, classification of the unconditional GAN increases, then decreases, indicating that information about the classes is acquired and later forgotten. This forgetting correlates with training instability. Adding self-supervision, as detailed in the following section, prevents this forgetting of the classes in the discriminator representations.

3. The Self-Supervised GAN

Motivated by the main challenge of discriminator forgetting, we aim to imbue the discriminator with a mechanism which allows learning useful representations, independently of the quality of the current generator. To this end, we exploit recent advancements in self-supervised approaches for representation learning. The main idea behind self-supervision is to train a model on a pretext task like predicting rotation angle or relative location of an image patch, and then extracting representations from the resulting networks [23, 24, 25]. We propose to add a self-supervised task to our discriminator.

In particular, we apply the state-of-the-art self-supervision method based on image rotation [26]. In this method, the images are rotated, and the angle of rotation becomes the artificial label (cf. Figure 1). The self-supervised task is then to predict the angle of rotation of an image. The effects of this additional loss on the image classification task is evident in Figure 3(b): When coupled with the self-supervised loss, the network learns representations that transfer across tasks and the performance continually improves. On the second cycle through the tasks, from 10k iterations onward, performance is improved. Intuitively, this loss encourages the classifier to learn useful image representations to detect the rotation angles, which transfers to the image classification task.

We augment the discriminator with a rotation-based loss which results in the following loss functions:

\[
L_G = -V(G, D) - \alpha \mathbb{E}_{x \sim P_{G}} \mathbb{E}_{r \sim R} \log Q_D(R = r \mid x'), \\
L_D = V(G, D) - \beta \mathbb{E}_{x \sim P_{G}} \mathbb{E}_{r \sim R} \log Q_D(R = r \mid x'),
\]

where \(V(G, D)\) is the value function from Equation 1, \(r \in R\) is a rotation selected from a set of possible rotations. In this work we use \(R = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}\) as in Gidaris et al. [26]. Image \(x\) rotated by \(r\) degrees is denoted as \(x'\), and \(Q(R \mid x')\) is the discriminator’s predictive distribution over the angles of rotation of the sample.
Collaborative Adversarial Training  In our model, the generator and discriminator are adversarial with respect to the true vs. fake prediction loss, $V(G, D)$, however, they are collaborative with respect to the rotation task. First, consider the value function of the generator which biases the generation towards images, that when rotated, the discriminator can detect their rotation angle. Note that the generator is not conditional but only generates “upright” images which are subsequently rotated and fed to the discriminator. On the other hand, the discriminator is trained to detect rotation angles based only on the true data. In other words, the parameters of the discriminator get updated only based on the rotation loss on the true data. This prevents the undesirable collaborative solution whereby the generator generates images whose subsequent rotation is easy to detect. As a result, the generator is encouraged to generate images that are rotation-detectable because they share features with real images that are used for rotation classification.

In practice, we use a single discriminator network with two heads to compute $P_D$ and $Q_D$. Figure 1 depicts the training pipeline. We rotate the real and generated images in four major rotations. The goal of the discriminator on non-rotated images is to predict whether the input is true or fake. On rotated real images, its goal is to detect the rotation angle. The goal of the generator is to generate images matching the observed data, whose representation in the feature space of the discriminator allows detecting rotations. With $\alpha > 0$ convergence to the true data distribution $P_G = P_{data}$ is not guaranteed. However, annealing $\alpha$ towards zero during training will restore the guarantees.

4. Experiments

We demonstrate empirically that (1) self-supervision improves the representation quality with respect to baseline GAN models, and that (2) it leads to improved unconditional generation for complex datasets, matching the performance of conditional GANs, under equal training conditions.

4.1. Experimental Settings

Datasets  We focus primarily on IMAGENET, the largest and most diverse image dataset commonly used to evaluate GANs. Until now, most GANs trained on IMAGENET are conditional. IMAGENET contains 1.3M training images and 50k test images. We resize the images to $128 \times 128 \times 3$ as done in Miyato and Koyama [21] and Zhang et al. [9]. We provide additional comparison on three smaller datasets, namely CIFAR10, CELEBA-HQ, LSUN-BEDROOM, for which unconditional GANs can be successfully trained. The LSUN-BEDROOM dataset [27] contains 3M images. We partition these randomly into a test set containing approximately 30k images and a train set containing the rest. CELEBA-HQ contains 30k images [10]. We use the $128 \times 128 \times 3$ version

<table>
<thead>
<tr>
<th>DATASET</th>
<th>METHOD</th>
<th>FID</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10</td>
<td>Uncond-GAN</td>
<td>19.73</td>
</tr>
<tr>
<td></td>
<td>Cond-GAN</td>
<td>15.60</td>
</tr>
<tr>
<td></td>
<td>SS-GAN</td>
<td>17.11</td>
</tr>
<tr>
<td></td>
<td>SS-GAN (sBN)</td>
<td>15.65</td>
</tr>
<tr>
<td>IMAGENET</td>
<td>Uncond-GAN</td>
<td>56.67</td>
</tr>
<tr>
<td></td>
<td>Cond-GAN</td>
<td>42.07</td>
</tr>
<tr>
<td></td>
<td>SS-GAN</td>
<td>47.56</td>
</tr>
<tr>
<td></td>
<td>SS-GAN (sBN)</td>
<td>43.87</td>
</tr>
<tr>
<td>LSUN-BEDROOM</td>
<td>Uncond-GAN</td>
<td>16.02</td>
</tr>
<tr>
<td></td>
<td>SS-GAN</td>
<td>13.66</td>
</tr>
<tr>
<td></td>
<td>SS-GAN (sBN)</td>
<td>13.30</td>
</tr>
<tr>
<td>CELEBA-HQ</td>
<td>Uncond-GAN</td>
<td>23.77</td>
</tr>
<tr>
<td></td>
<td>SS-GAN</td>
<td>26.11</td>
</tr>
<tr>
<td></td>
<td>SS-GAN (sBN)</td>
<td>24.36</td>
</tr>
</tbody>
</table>

Table 1: Best FID attained across three random seeds. In this setting the proposed approach recovers most of the benefits of conditioning.

Models  We compare the self-supervised GAN (SS-GAN) to two well-performing baseline models, namely (1) the unconditional GAN with spectral normalization proposed in Miyato et al. [6], denoted Uncond-GAN, and (2) the conditional GAN using the label-conditioning strategy and the Projection Conditional GAN (Cond-GAN) [21]. We chose the latter as it was shown to outperform the AC-GAN [20], and is adopted by the best performing conditional GANs [9, 3, 22].

We use ResNet architectures for the generator and discriminator as in Miyato et al. [6]. For the conditional generator in Cond-GAN, we apply label-conditional batch normalization. In contrast, SS-GAN does not use conditional batch normalization. However, to have a similar effect on the generator, we consider a variant of SS-GAN where we apply the self-modulated batch normalization which does not require labels [7] and denote it SS-GAN (sBN). We note that labels are available only for CIFAR10 and IMAGENET, so Cond-GAN is only applied on those data sets.

We use a batch size of 64 and to implement the rotation-loss we rotate 16 images in the batch in all four considered directions. We do not add any new images into the batch to compute the rotation loss. For the true vs. fake task we use the hinge loss from Miyato et al. [6]. We set $\beta = 1$ or the the self-supervised loss. For $\alpha$ we performed a small sweep.

We train with self-modulation it matches the performance on the conditional GAN. In terms of mean performance (Figure 4), the proposed approach matches the conditional GAN, and in terms of the best models selected across random seeds (Table 1), the performance gap is within 5%. On CIFAR10 and LSUN-BEDROOM we observe a substantial improvement over the unconditional GAN and matching the performance of the conditional GAN. Self-supervision appears not to significantly improve the results on CELEBA-HQ. We posit that this is due to low-diversity in CELEBA-HQ, and also for which rotation task is less informative.

<table>
<thead>
<tr>
<th>TYPE</th>
<th>λ</th>
<th>β_1</th>
<th>β_2</th>
<th>D ITERS</th>
<th>UNCOND-GAN</th>
<th>SS-GAN</th>
<th>UNCOND-GAN</th>
<th>SS-GAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRADIENT PENALTY</td>
<td>1</td>
<td>0.0</td>
<td>0.999</td>
<td>1</td>
<td>121.05 ± 31.44</td>
<td>25.8 ± 0.71</td>
<td>183.36 ± 77.21</td>
<td>80.67 ± 0.43</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0</td>
<td>0.999</td>
<td>1</td>
<td>28.11 ± 0.66</td>
<td>26.98 ± 0.54</td>
<td>85.13 ± 2.88</td>
<td>83.08 ± 0.38</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.0</td>
<td>0.999</td>
<td>1</td>
<td>188.52 ± 64.54</td>
<td>28.48 ± 0.68</td>
<td>227.04 ± 31.45</td>
<td>85.38 ± 2.7</td>
</tr>
<tr>
<td>SPECTRAL NORM</td>
<td>0</td>
<td>0.0</td>
<td>0.999</td>
<td>1</td>
<td>117.67 ± 17.46</td>
<td>25.22 ± 0.38</td>
<td>242.71 ± 13.62</td>
<td>144.35 ± 91.4</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.999</td>
<td>1</td>
<td>87.86 ± 3.44</td>
<td>19.65 ± 0.9</td>
<td>129.96 ± 6.6</td>
<td>86.09 ± 7.66</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>20.24 ± 0.62</td>
<td>17.88 ± 0.64</td>
<td>80.05 ± 1.33</td>
<td>70.64 ± 0.31</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>20.24 ± 0.62</td>
<td>17.88 ± 0.64</td>
<td>80.05 ± 1.33</td>
<td>70.64 ± 0.31</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: FID for unconditional GANs under different hyperparameter settings. Mean and standard deviations are computed across three random seeds. Adding the self-supervision loss reduces the sensitivity of GAN training to hyperparameters.

Robustness across hyperparameters GANs are fragile; changes to the hyperparameter settings have a substantial impact to their performance [30, 32]. Therefore, we evaluate different hyperparameter settings to test the stability of SS-GAN. We consider two classes of hyperparameters: First, those controlling the Lipschitz constant of the discriminator, a central quantity analyzed in the GAN literature [6, 33]. We evaluate two state-of-the-art techniques: gradient penalty [5], and spectral normalization [6]. The gradient penalty introduces a regularization strength parameter, λ. We test two values λ ∈ {1, 10}. Second, we vary the hyperparameters of the Adam optimizer. We test two popular settings (β_1, β_2): (0.5, 0.999) and (0, 0.9). Previous studies find that multiple discriminator steps per generator step help training [1, 2], so we try both 1 and 2 discriminator steps per generator step.

Table 2 compares the mean FID scores of the unconditional models across penalties and optimization hyperparameters. We observe that the proposed approach yields consistent performance improvements. We observe that in settings where the unconditional GAN collapses (yielding FIDs larger than 100) the self-supervised model does not exhibit such a collapse.

4.3. Large Scale Self-Supervised GAN

We scale up training the SS-GAN to attain the best possible FID for unconditional IMAGENET generation. To do this, we increase the model’s capacity to match the model.
Figure 4: FID learning curves on CIFAR10 and IMAGENET. The curves show the mean performance across three random seeds. The unconditional GAN (Uncond-GAN) attains significantly poorer performance than the conditional GAN (Cond-GAN). The unconditional GAN is unstable on IMAGENET and the runs often diverge after 500k training iterations. The addition of self-supervision (SS-GAN) stabilizes Uncond-GAN and boosts performance. Finally, when we add the additional self-modulated Batch Norm (sBN) \cite{7} to SS-GAN, which mimics generator conditioning in the unconditional setting, this unconditional model attains the same mean performance as the conditional GAN.

We train the model on 128 cores of Google TPU v3 Pod for 500k steps using batch size of 2048. For comparison, we also train the same model without the auxiliary self-supervised loss (Uncond-GAN). We report the FID at 50k to be comparable other literature reporting results on IMAGENET. We repeat each run three times with different random seeds.

For SS-GAN we obtain the FID of 23.6 \(\pm 0.1\) and 71.6 \(\pm 66.3\) for Uncond-GAN. Self-supervision stabilizes training; the mean and variance across random seeds is greatly reduced because, unlike for the regular unconditional GAN, SS-GAN never collapsed. We observe improvement in the best model across random seeds, and the best SS-GAN attains an FID of 23.4. To our knowledge, this is the best results attained training unconditionally on IMAGENET.

4.4. Representation Quality

We test empirically whether self-supervision encourages the discriminator to learn meaningful representations. For this, we compare the quality of the representations extracted from the intermediate layers of the discriminator’s ResNet architecture. We apply a common evaluation method for representation learning, proposed in Zhang et al. \cite{25}. In particular, we train a logistic regression classifier on the feature maps from each ResNet block to perform the 1000-way classification task on IMAGENET or 10-way on CIFAR10 and report top-1 classification accuracy.

We report results using the Cond-GAN, Uncond-GAN, and SS-GAN models. We also ablate the adversarial loss from our SS-GAN which results in a purely rotation-based self-supervised model (Rot-only) which uses the same architecture and hyperparameters as the SS-GAN discriminator. We report the mean accuracy and standard deviation

\footnote{The details can be found at \url{https://github.com/google/compare_gan}.}
across three independent models with different random seeds. Training details for the logistic classifier are in the appendix.

**Results** Table 4 shows the quality of representation at after 1M training steps on IMAGENET. Figure 6 shows the learning curves for representation quality of the final ResNet block on IMAGENET. The curves for the other blocks are provided in appendix. Note that “training steps” refers to the training iterations of the original GAN, and not to the linear classifier which is always trained to convergence. Overall, the SS-GAN yields the best representations across all blocks and training iterations. We observe similar results on CIFAR10 provided in Table 3.

In detail, the IMAGENET ResNet contains six blocks. For Uncond-GAN and Rot-only, Block 3 performs best, for Cond-GAN and SS-GAN, the final Block 5 performs best. The representation quality for Uncond-GAN drops at 500k steps, which is consistent with the FID drop in Figure 4. Overall, the SS-GAN and Cond-GAN representations are better than Uncond-GAN, which correlates with their improved sample quality. Surprisingly, the the SS-GAN overtakes Cond-GAN after training for 300k steps. One possibility is that the Cond-GAN is overfitting the training data. We inspect the representation performance of Cond-GAN on the training set and indeed see a very large generalization gap, which indicates overfitting.

When we ablate the GAN loss, leaving just the rotation loss, the representation quality substantially decreases. It seems that the adversarial and rotation losses complement each other both in terms of FID and representation quality. We emphasize that our discriminator architecture is optimized for image generation, not representation quality. Rot-only, therefore, is an ablation method, and is not a state-of-the-art self-supervised learning algorithm. We discuss these next.

Table 5 compares the representation quality of SS-GAN to state-of-the-art published self-supervised learning algorithms. Despite the architecture and hyperparameters being optimized for image quality, the SS-GAN model achieves competitive results on IMAGENET. Among those methods, only BiGAN [34] also uses a GAN to learn representations; but SS-GAN performs substantially (0.073 accuracy points) better. BiGAN learns the representation with an additional encoder network, while SS-GAN is arguably simpler because it extracts the representation directly from the discriminator. The best performing method is the recent DeepClustering algorithm [35]. This method is just 0.027 accuracy points ahead of SS-GAN and requires expensive offline clustering after every training epoch.

In summary, the representation quality evaluation highlights the correlation between representation quality and image quality. It also confirms that the SS-GAN does learn relatively powerful image representations.

![Figure 6: IMAGENET Top 1 accuracy (mean across three seeds) to predict labels from discriminator representations. X-axis gives the number of GAN training iterations.](image)

**Table 3:** Top-1 accuracy on CIFAR10. Mean score across three training runs of the original model. All standard deviations are smaller than 0.01 and are reported in the appendix.

<table>
<thead>
<tr>
<th>Method</th>
<th>Uncond</th>
<th>Cond</th>
<th>Rot-only</th>
<th>SS-GAN (sBN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block0</td>
<td>0.719</td>
<td>0.719</td>
<td>0.710</td>
<td><strong>0.721</strong></td>
</tr>
<tr>
<td>Block1</td>
<td>0.762</td>
<td>0.759</td>
<td>0.749</td>
<td><strong>0.774</strong></td>
</tr>
<tr>
<td>Block2</td>
<td>0.778</td>
<td>0.776</td>
<td>0.762</td>
<td><strong>0.796</strong></td>
</tr>
<tr>
<td>Block3</td>
<td>0.776</td>
<td>0.780</td>
<td>0.752</td>
<td><strong>0.799</strong></td>
</tr>
<tr>
<td>Best</td>
<td>0.778</td>
<td>0.780</td>
<td>0.762</td>
<td><strong>0.799</strong></td>
</tr>
</tbody>
</table>

**Table 4:** Top-1 accuracy on IMAGENET. Mean score across three training runs of the original model. All standard deviations are smaller than 0.01, except for Uncond-GAN whose results exhibit high variance due to training instability. All standard deviations are reported in the appendix.

<table>
<thead>
<tr>
<th>Method</th>
<th>Uncond</th>
<th>Cond</th>
<th>Rot-only</th>
<th>SS-GAN (sBN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block0</td>
<td>0.074</td>
<td>0.156</td>
<td>0.147</td>
<td><strong>0.158</strong></td>
</tr>
<tr>
<td>Block1</td>
<td>0.063</td>
<td>0.187</td>
<td>0.134</td>
<td><strong>0.222</strong></td>
</tr>
<tr>
<td>Block2</td>
<td>0.073</td>
<td>0.217</td>
<td>0.158</td>
<td><strong>0.250</strong></td>
</tr>
<tr>
<td>Block3</td>
<td>0.083</td>
<td>0.272</td>
<td>0.202</td>
<td><strong>0.327</strong></td>
</tr>
<tr>
<td>Block4</td>
<td>0.077</td>
<td>0.253</td>
<td>0.196</td>
<td><strong>0.358</strong></td>
</tr>
<tr>
<td>Block5</td>
<td>0.074</td>
<td>0.337</td>
<td>0.195</td>
<td><strong>0.383</strong></td>
</tr>
<tr>
<td>Best</td>
<td>0.083</td>
<td>0.337</td>
<td>0.202</td>
<td><strong>0.383</strong></td>
</tr>
</tbody>
</table>

**5. Related Work**

**GAN forgetting** Catastrophic forgetting was previously considered as a major cause for GAN training instability. The main remedy suggested in the literature is to introduce...
This architecture improves performance over AC-GAN. The table 5: Comparison with other self-supervised representation learning methods by top-1 accuracy on ImageNet. For SS-GAN, the mean performance is presented.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Context [24]</td>
<td>0.317</td>
</tr>
<tr>
<td>BiGAN [34, 36]</td>
<td>0.310</td>
</tr>
<tr>
<td>Colorization [25]</td>
<td>0.326</td>
</tr>
<tr>
<td>RotNet [26]</td>
<td>0.387</td>
</tr>
<tr>
<td>DeepClustering [35]</td>
<td>0.410</td>
</tr>
<tr>
<td>SS-GAN (sBN)</td>
<td>0.383</td>
</tr>
</tbody>
</table>

Table 5: Comparison with other self-supervised representation learning methods by top-1 accuracy on ImageNet. For SS-GAN, the mean performance is presented.

Self-supervised learning

Conditional GANs

Conditional GANs are currently the best approach for generative modeling of complex data sets, such as ImageNet. The AC-GAN was the first model to introduce an auxiliary classification loss for the discriminator [20]. The main difference between AC-GAN and the proposed approach is that self-supervised GAN requires no labels. Furthermore, the AC-GAN generator generates images conditioned on the class, whereas our generator is unconditional and the images are subsequently rotated to produce the artificial label. Finally, the self-supervision loss for the discriminator is applied only over real images, whereas the AC-GAN uses both real and fake.

More recently, the P-cGAN model proposed by Miyato and Koyama [21] includes one real/fake head per class [21]. This architecture improves performance over AC-GAN. The best performing GANs trained on GPUs [9] and TPUs [22] use P-cGAN style conditioning in the discriminator. We note that conditional GANs also use labels in the generator, either by concatenating with the latent vector, or via FiLM modulation [38].

Self-supervised learning

Self-supervised learning is a family of methods that learn the high level semantic representation by solving a surrogate task. It has been widely used in the video domain [39, 40], the robotics domain [41, 42] and the image domain [24, 35]. We focused on the image domain in this paper. Gidaris et al. [26] proposed to rotate the image and predict the rotation angle. This conceptually simple task yields useful representations for downstream image classification tasks. Apart from trying to predict the rotation, one can also make edits to the given image and ask the network to predict the edited part. For example, the network can be trained to solve the context prediction problem, like the relative location of disjoint patches [24, 43] or the patch permutation of a jigsaw puzzle [44]. Other surrogate tasks include image inpainting [45], predicting the color channels from a grayscale image [25], and predicting the unsupervised clustering classes [35]. Recently, Kolesnikov et al. [46] conducted a study on self-supervised learning with modern neural architectures.

6. Conclusions and Future Work

Motivated by the desire to counter discriminator forgetting, we propose a deep generative model that combines adversarial and self-supervised learning. The resulting novel model, namely self-supervised GAN when combined with the recently introduced self-modulation, can match equivalent conditional GANs on the task of image synthesis, without having access to labeled data. We then show that this model can be scaled to attain an FID of 23.4 on unconditional ImageNet generation which is an extremely challenging task.

This line of work opens several avenues for future research. First, it would be interesting to use a state-of-the-art self-supervised architecture for the discriminator, and optimize for best possible representations. Second, the self-supervised GAN could be used in a semi-supervised setting where a small number of labels could be used to fine-tune the model. Finally, one may exploit several recently introduced techniques, such as self-attention, orthogonal normalization and regularization, and sampling truncation [9, 22], to yield even better performance in unconditional image synthesis.

We hope that this approach, combining collaborative self-supervision with adversarial training, can pave the way towards high quality, fully unsupervised, generative modeling of complex data.

Acknowledgements

We would also like to thank Marcin Michalski, Karol Kurach and Anton Raichuk for their help with infrastructure, and major contributions to the Compare GAN library. We appreciate useful discussions with Ilya Tolstikhin, Olivier Bachem, Alexander Kolesnikov, Josip Djolonga, and Tian-sheng Yao. Finally, we are grateful for the support of other members of the Google Brain team, Zürich.
References


