Self-Supervised Generative Adversarial Networks

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 labelled data. In this work we exploit two popular unsupervised learning techniques, adversarial training and self-supervision, to close 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 33 on unconditional IMAGENET *generation.*

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 labelled data. In particular, *under the same training conditions*, the self-supervised GAN closes the gap in natural image synthesis between un-

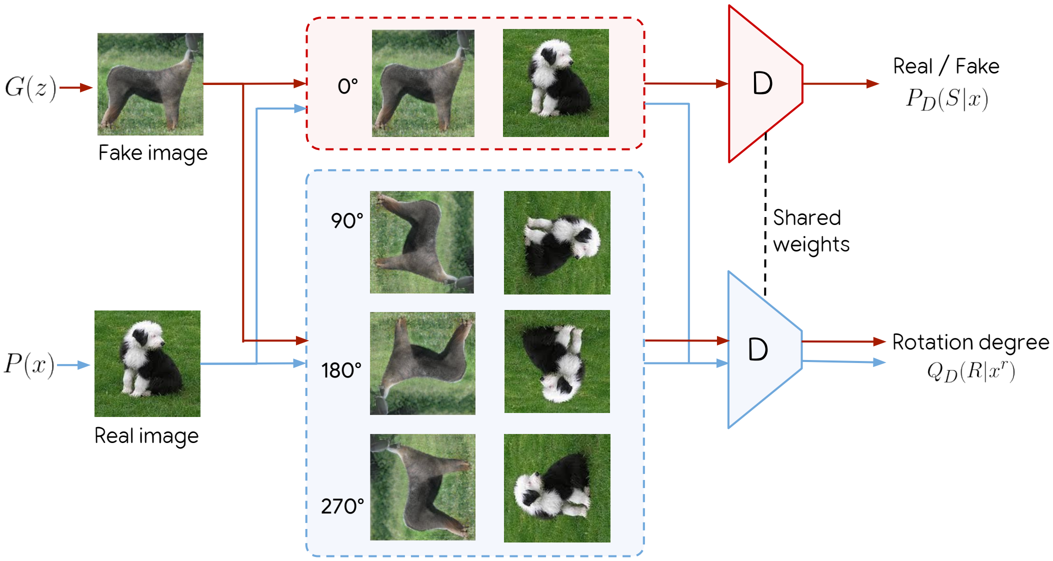


Figure 1: Discriminator with rotation-based self-supervision. The discriminator, D, performs two tasks: true vs. fake binary classification, and rotation degree classification. Both the fake and real images are rotated by 0, 90, 180, and 270 degrees. The colored arrows indicate that only the upright images are considered for true vs. fake classification loss task. For the rotation loss, all images are classified by the discriminator according to their rotation degree.

conditional 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]:

$$\begin{aligned} V(G,D) =&\mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}(\boldsymbol{x})}[\log P_D(S=1\mid \boldsymbol{x})] \\ &+\mathbb{E}_{\boldsymbol{x}\sim P_G(\boldsymbol{x})}[\log(1-P_D(S=0\mid \boldsymbol{x}))] \end{aligned} \tag{1}$$

where P_{data} is the true data distribution, and P_G is the distribution induced by transforming a simple distribution $\boldsymbol{z}\sim P(\boldsymbol{z})$ using the deterministic mapping given by the generator, $\boldsymbol{x}=G(\boldsymbol{z})$, 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_{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

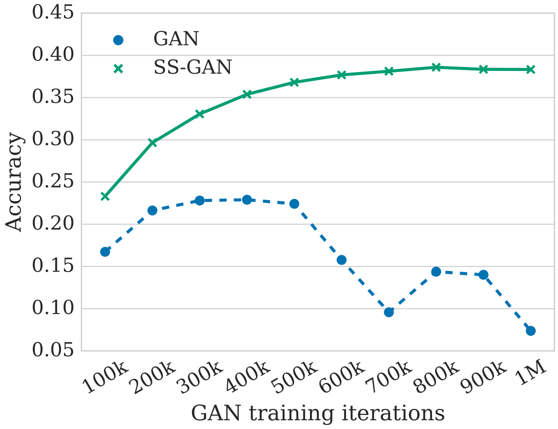
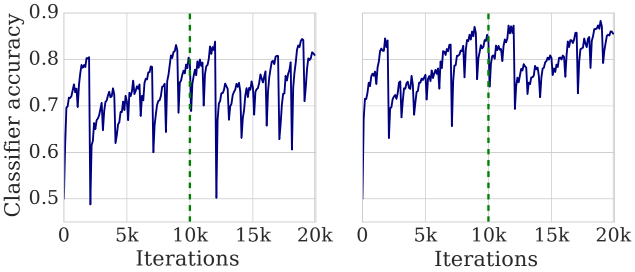


Figure 2: Performance of a linear classification model, trained on IMAGENET on representations extracted from the final layer of the discriminator. Uncond-GAN denotes an unconditional GAN. SS-GAN denotes the same model when self-supervision is added. For the Uncond-GAN, the representation gathers information about the class of the image and the accuracy increases. However, after 500k iterations, the representations lose information about the classes and performance decreases. SS-GAN alleviates this problem. More details are presented in Section 4.



(a) Regular training.

(b) With self-supervision.

Figure 3: Image classification accuracy when the underlying class distribution shifts every 1k iterations. The vertical dashed line indicates the end of an entire cycle through the tasks, and return to the original classification task at $t = 0$. *Left*: vanilla classifier. *Right*: classifier with an additional self-supervised loss. This example demonstrates that a classifier may fail to learn generalizable representations in a non-stationary environment, but self-supervision helps mitigate this problem.

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 3(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:

$$L_G = -V(G, D) - \alpha \mathbb{E}_{\boldsymbol{x} \sim P_G} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r)\right],$$
$$L_D = V(G, D) - \beta \mathbb{E}_{\boldsymbol{x} \sim P_{\text{data}}} \mathbb{E}_{r \sim \mathcal{R}} \left[\log Q_D(R = r \mid \boldsymbol{x}^r)\right],$$

where $V(G, D)$ is the value function from Equation 1, $r \in \mathcal{R}$ is a rotation selected from a set of possible rotations. In this work we use $\mathcal{R} = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}$ as in Gidaris et al. [26]. Image $\boldsymbol{x}$ rotated by r degrees is denoted as $\boldsymbol{x}^r$, and $Q(R \mid \boldsymbol{x}^r)$ 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 $P_G = P_{\text{data}}$ is not guaranteed. However, annealing α 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 obtained by running the code provided by the authors.[1] We use 3k examples as the test set and the remaining examples as the training set. CIFAR10 contains 70k images ($32 \times 32 \times 3$), partitioned into 60k training instances and 10k test instances.

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

[1] https://github.com/tkarras/progressive_growing_of_gans.

DATASET	METHOD	FID
CIFAR10	Uncond-GAN	19.73
	Cond-GAN	**15.60**
	SS-GAN	17.11
	SS-GAN (sBN)	15.65
IMAGENET	Uncond-GAN	56.67
	Cond-GAN	**42.07**
	SS-GAN	47.56
	SS-GAN (sBN)	43.87
LSUN-BEDROOM	Uncond-GAN	16.02
	SS-GAN	13.66
	SS-GAN (sBN)	**13.30**
CELEBA-HQ	Uncond-GAN	**23.77**
	SS-GAN	26.11
	SS-GAN (sBN)	24.36

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

					CIFAR10		IMAGENET	
TYPE	λ	β_1	β_2	D ITERS	UNCOND-GAN	SS-GAN	UNCOND-GAN	SS-GAN
GRADIENT PENALTY	1	0.0	0.900	1	121.05 ± 31.44	**25.8 ± 0.71**	183.36 ± 77.21	**80.67 ± 0.43**
				2	28.11 ± 0.66	**26.98 ± 0.54**	85.13 ± 2.88	**83.08 ± 0.38**
		0.5	0.999	1	78.54 ± 6.23	**25.89 ± 0.33**	104.73 ± 2.71	**91.63 ± 2.78**
	10	0.0	0.900	1	188.52 ± 64.54	**28.48 ± 0.68**	227.04 ± 31.45	**85.38 ± 2.7**
				2	29.11 ± 0.85	**27.74 ± 0.73**	227.74 ± 16.82	**80.82 ± 0.64**
		0.5	0.999	1	117.67 ± 17.46	**25.22 ± 0.38**	242.71 ± 13.62	**144.35 ± 91.4**
SPECTRAL NORM	0	0.0	0.900	1	87.86 ± 3.44	**19.65 ± 0.9**	129.96 ± 6.6	**86.09 ± 7.66**
				2	20.24 ± 0.62	**17.88 ± 0.64**	80.05 ± 1.33	**70.64 ± 0.31**
		0.5	0.999	1	86.87 ± 8.03	**18.23 ± 0.56**	201.94 ± 27.28	**99.97 ± 2.75**

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.

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 α we performed a small sweep $\alpha \in \{0.2, 0.5, 1\}$, and select $\alpha = 0.2$ for all datasets (see the appendix for details). For all other hyperparameters, we use the values in Miyato et al. [6] and Miyato and Koyama [21]. We train CIFAR10, LSUN-BEDROOM and CELEBA-HQ for 100k steps on a single P100 GPU. For IMAGENET we train for 1M steps. For all datasets we use the Adam optimizer with learning rate 0.0002.

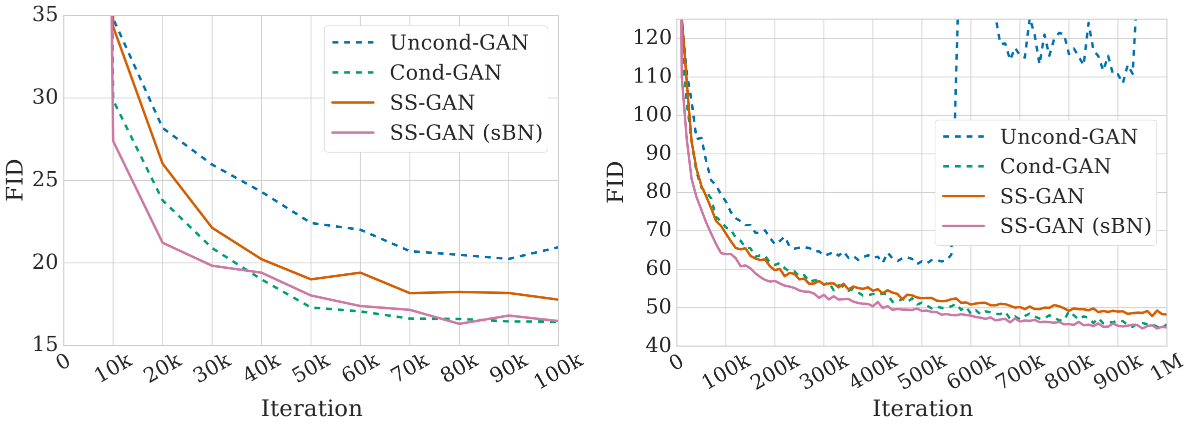
4.2. Comparison of Sample Quality

Metrics To evaluate generated samples from different methods quantitatively, we use the Frechet Inception Distance (FID) [28]. In FID, the true 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 true and generated data respectively. FID is shown to be sensitive to both the addition of spurious modes and to mode dropping [29, 30]. An alternative approximate measure of sample quality is Inceptions Score (IS) Salimans et al. [2]. Since it has some flaws Barratt and Sharma [31], we use FID as the main metric in this work.

Results Figure 4 shows FID training curves on CIFAR10 and IMAGENET. Table 1 shows the FID of the best run across three random seeds for each dataset and model combination. The unconditional GAN is unstable on IMAGENET and the training often diverges. The conditional counterpart outperforms it substantially. The proposed method, namely SS-GAN, is stable on IMAGENET, and performs substantially better than the unconditional GAN. When equipped 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.

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 $\lambda \in \{1, 10\}$. Second, we vary the hyperparmeters 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.



(a) CIFAR10

(b) IMAGENET

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) [7] to SS-GAN, which mimics generator conditioning in the unconditional setting, this unconditional model attains the same mean performance as the conditional GAN.

4.3. Scaling up the Self-Supervised GAN

We scale up training of the SS-GAN to attain the best possible FID for unconditional IMAGENET generation. We train the proposed model on 128 cores of Google TPU v3 Pod for 350k steps using batch size of 1024 and attain an FID score of 33.5. The details of the architectures are provided in the appendix.

Figure 5 shows a random sample of images generated by the model. Training unconditionally on IMAGENET is much harder than conditional training, therefore, there is not much prior art. Unconditional IMAGENET generation results using autoregressive models have been presented, however, FID score are not reported [34]. Using many additional tricks and techniques, a very recent approach based on conditional generation has attained an FID of 9.6 with large scale training. Although our unconditional model is still far behind, our results in Section 4.2 indicate that under the same conditions, SS-GAN can match conditional settings. We are hopeful that this results, which dramatically improves upon previous unconditional models, paves the way to attaining comparable scores with unconditional models in the future. 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

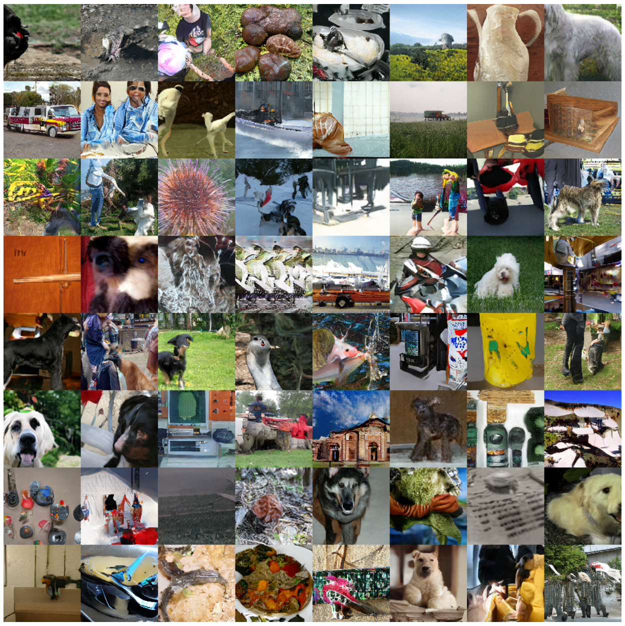


Figure 5: A random sample (non-cherrypicked) of unconditionally generated images from the self-supervised model. Although our unconditional model can clearly be improved, our results in Section 4.2 indicate that under the same conditions the proposed approach can match the conditional counterpart. To our knowledge, this is the best results attained training unconditionally on IMAGENET.

architecture. We apply a common evaluation method for representation learning, proposed in Zhang et al. [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 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 9 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.

	Uncond.	Cond.	Rot-only	SS-GAN (sBN)
Block0	0.719	0.719	0.710	**0.721**
Block1	0.762	0.759	0.749	**0.774**
Block2	0.778	0.776	0.762	**0.796**
Block3	0.776	0.780	0.752	**0.799**
Best	0.778	0.780	0.762	**0.799**

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.

Method	Uncond.	Cond.	Rot-only	SS-GAN (sBN)
Block0	0.074	0.156	0.147	**0.158**
Block1	0.063	0.187	0.134	**0.222**
Block2	0.073	0.217	0.158	**0.250**
Block3	0.083	0.272	0.202	**0.327**
Block4	0.077	0.253	0.196	**0.358**
Block5	0.074	0.337	0.195	**0.383**
Best	0.083	0.337	0.202	**0.383**

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.

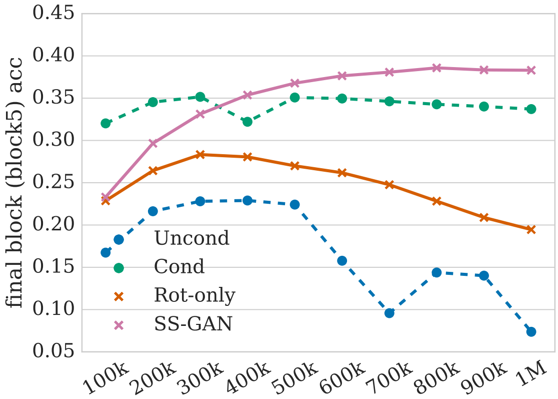


Figure 6: IMAGENET Top 1 accuracy (mean across three seeds) to predict labels from discriminator representations. X-axis gives the number of GAN training iterations.

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 [35] 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 [36]. 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 represenatations.

Method	Accuracy
Context [24]	0.317
BiGAN [35]	0.310
Colorization [25]	0.326
RotNet [26]	0.387
DeepClustering [36]	0.410
SS-GAN (sBN)	0.383

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

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 temporal memory into the training algorithm in various ways. For example, Grnarova et al. [19] induce discriminator memory by replaying previously generated images. An alternative is to instead reuse previous models: Salimans et al. [2] introduce checkpoint averaging, where a running average of the parameters of each player is kept, and Grnarova et al. [19] maintain a queue of models that are used at each training iteration. Kim et al. [18] add memory to retain information about previous samples. Other papers frame GAN training as a continual learning task. Thanh-Tung et al. [14] study catastrophic forgetting in the discriminator and mode collapse, relating these to training instability. Anonymous [15] counter discriminator forgetting by leveraging techniques from continual learning directly (Elastic Weight Sharing [11] and Intelligent Synapses [37]).

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, 36]. 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 form 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 [36].

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 33 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 modelling of complex data.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

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

[3] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018.

[4] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *International Conference on Computer Vision (ICCV)*, 2016.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2018.

[7] Ting Chen, Mario Lucic, Neil Houlsby, and Sylvain Gelly. On self modulation for generative adversarial networks. *arXiv preprint arXiv:1810.01365*, 2018.

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

[9] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

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

[11] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

[12] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24. Elsevier, 1989.

[13] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 1999.

[14] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. On catastrophic forgetting and mode collapse in generative adversarial networks. *ICML Workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018.

[15] Anonymous. Generative adversarial network training is a continual learning problem. In *Submitted to International Conference on Learning Representations (ICLR)*, 2019. URL `https://openreview.net/forum?id=SJzuHiA9tQ`. under review.

[16] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Continual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, 2017.

[17] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] Youngjin Kim, Minjung Kim, and Gunhee Kim. Memorization precedes generation: Learning unsupervised gans with memory networks. *International Conference on Learning Representations (ICLR)*, 2018.

[19] Paulina Grnarova, Kfir Y Levy, Aurelien Lucchi, Thomas Hofmann, and Andreas Krause. An online learning approach to generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018.

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

[21] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. *International Conference on Learning Representations (ICLR)*, 2018.

[22] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[23] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[24] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015.

[25] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 2016.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations (ICLR)*, 2018.

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

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

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

[30] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs Created Equal? A Large-scale Study. In *To appear in Advances in Neural Information Processing Systems (NIPS)*, 2018.

[31] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[32] Karol Kurach, Mario Lucic, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. The GAN Landscape: Losses, architectures, regularization, and normalization. *arXiv preprint arXiv:1807.04720*, 2018.

[33] Zhiming Zhou, Yuxuan Song, Lantao Yu, and Yong Yu. Understanding the effectiveness of lipschitz constraint in training of gans via gradient analysis. *arXiv preprint arXiv:1807.00751*, 2018.

[34] Anonymous. Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In *Submitted to International Conference on Learning Representations (ICLR)*, 2019. under review.

[35] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *International Conference on Learning Representations (ICLR)*, 2017.

[36] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. *European Conference on Computer Vision (ECCV)*, 2018.

[37] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, 2017.

[38] Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[39] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *International Conference on Computer Vision (ICCV)*, 2015.

[40] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *International Conference on Computer Vision (ICCV)*, 2017.

[41] Eric Jang, Coline Devin, Vincent Vanhoucke, and Sergey Levine. Grasp2vec: Learning object representations from self-supervised grasping. In *Conference on Robot Learning (CoRL*, 2018.

[42] Lerrel Pinto and Abhinav Gupta. Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, 2016.

[43] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[44] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision (ECCV)*, 2016.

[45] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[46] TFGAN: A lightweight library for generative adversarial networks, 2017. URL https://ai.googleblog.com/2017/12/tfgan-lightweight-library-for.html.

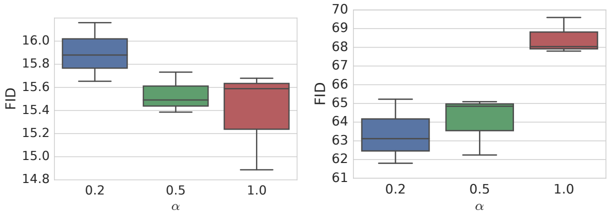
[47] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

A. FID Metric Details

We compute the FID score using the protocol as decribed in [28]. The image embeddings are extracted from an Inception V1 network provided by the TF library [46], We use the layer "pool_3". We fit the multivariate Gaussians used to compute the metric to real samples from the test sets and fake samples. We use 3000 samples for CELEBA-HQ and 10000 for the other datasets.

B. SS-GAN Hyper-parameters

We compare different choices of α, while fixing $\beta = 1$ for simplicity. A reasonable value of α helps aqa the generator to train using the self-supervision task, however, an inappropriate value of α could bias the convergence point of the generator. Table 7 shows the effectiveness of α. In the values compared, the optimal α is 1 for CIFAR10, and 0.2 for IMAGENET. In our main experiments, we used $\alpha = 0.2$ for all datasets.



(a) CIFAR10 (b) IMAGENET

Figure 7: Performance under different α values.

C. Details for Scaling Up SS-GAN

We scale up SS-GAN to train on 128 cores of a Google TPU v3 Pod with a batch size of 1024. For the rotation-loss we use 512 rotated images from each batch. We use Orthogonal Initialization [?] instead of $N(0, 0.02)$ for initializing the weights in both generator and discriminator. We also find $(0, 0.999)$ for (β_1, β_2) of the Adam optimizer to work slightly better. We did not change the learning rate.

We use the variant of SS-GAN that incorporates self-modulated batch normalization. For the batch normalization we use a decay rate of 0.9 and epsilon set to $1e^{-5}$. The batch statistics are computed across all cores.

The architecture is based on Miyato et al. [6] but incorporates the changes by Chen et al. [7]. We change the channel width multiplier ch from 64 to 96. Details are are shown in Tables 6 and 7.

Layer	Details	Output size
Latent noise	$\boldsymbol{z} \sim \text{Unif}(-1, 1)$	128
Fully conn	Linear	$4 \cdot 4 \cdot 16 \cdot ch$
ResNet block	sBN,ReLU,Upsample	$8 \times 8 \times 16 \cdot ch$
	Conv3x3,sBN,ReLU	$8 \times 8 \times 16 \cdot ch$
	Conv3x3	$8 \times 8 \times 16 \cdot ch$
ResNet block	sBN,ReLU,Upsample	$16 \times 16 \times 16 \cdot ch$
	Conv3x3,sBN,ReLU	$16 \times 16 \times 16 \cdot ch$
	Conv3x3	$16 \times 16 \times 8 \cdot ch$
ResNet block	sBN,ReLU,Upsample	$32 \times 32 \times 8 \cdot ch$
	Conv3x3,sBN,ReLU	$32 \times 32 \times 8 \cdot ch$
	Conv3x3	$32 \times 32 \times 4 \cdot ch$
ResNet block	sBN,ReLU,Upsample	$64 \times 64 \times 4 \cdot ch$
	Conv3x3,sBN,ReLU	$64 \times 64 \times 4 \cdot ch$
	Conv3x3	$64 \times 64 \times 2 \cdot ch$
ResNet block	sBN,ReLU,Upsample	$128 \times 128 \times 2 \cdot ch$
	Conv3x3,sBN,ReLU	$128 \times 128 \times 2 \cdot ch$
	Conv3x3	$128 \times 128 \times 1 \cdot ch$
Conv	Conv3x3, Sigmoid	$128 \times 128 \times 3$

Table 6: ResNet Generator of the upscaled SS-GAN. Each ResNet block has a skip-connection that uses up-sampling of its input and a 1x1 convolution.

Layer	Details	Output size
Input image		$128 \times 128 \times 3$
ResNet block	Conv3x3	$128 \times 128 \times 1 \cdot ch$
	ReLU,Conv3x3	$128 \times 128 \times 1 \cdot ch$
	Downsample	$64 \times 64 \times 1 \cdot ch$
ResNet block	ReLU,Conv3x3	$64 \times 64 \times 1 \cdot ch$
	ReLU,Conv3x3	$64 \times 64 \times 2 \cdot ch$
	Downsample	$32 \times 32 \times 2 \cdot ch$
ResNet block	ReLU,Conv3x3	$32 \times 32 \times 2 \cdot ch$
	ReLU,Conv3x3	$32 \times 32 \times 4 \cdot ch$
	Downsample	$16 \times 16 \times 4 \cdot ch$
ResNet block	ReLU,Conv3x3	$16 \times 16 \times 4 \cdot ch$
	ReLU,Conv3x3	$16 \times 16 \times 8 \cdot ch$
	Downsample	$8 \times 8 \times 8 \cdot ch$
ResNet block	ReLU,Conv3x3	$8 \times 8 \times 8 \cdot ch$
	ReLU,Conv3x3	$8 \times 8 \times 16 \cdot ch$
	Downsample	$4 \times 4 \times 16 \cdot ch$
ResNet block	ReLU,Conv3x3	$4 \times 4 \times 16 \cdot ch$
	ReLU,Conv3x3	$4 \times 4 \times 16 \cdot ch$
Fully connected	ReLU,Pooling	$16 \cdot ch$
	Linear	1

Table 7: ResNet Discriminator of the upscaled SS-GAN. Each ResNet block has a skip-connection that applies a 1x1 convolution with possible down-sampling according to spatial dimension.

D. Representation Quality

D.1. Implementation Details

We train the linear evaluation models with batch size 128 and learning rate of $0.1 \times \frac{\text{batch\_size}}{256}$ following the linear scaling rule [47], for 50 epochs. The learning rate is decayed by a factor of 10 after epoch 30 and epoch 40. For data augmentation we resize the smaller dimension of the image to 146 and preserve the aspect ratio. After that we crop the image to 128×128. We apply a random crop for training and a central crop for testing. The model is trained on a single NVIDIA Tesla P100 GPU.

D.2. Additional Results

Table 8 shows the top-1 accuracy with on CIFAR10 with standard deviations. The results are stable on CIFAR10 as all the standard deviation is within 0.01. Table 9 shows the top-1 accuracy with on IMAGENET with standard deviations. Uncond-GAN representation quality shows large variance as we observe that the unconditional GAN collapses in some cases.

Figure 8 shows the representation quality on all 4 blocks on the CIFAR10 dataset. SS-GAN consistently outperforms other models on all 4 blocks. Figure 9 shows the representation quality on all 6 blocks on the IMAGENET dataset. We observe that all methods perform similarly before 500k steps on block0, which contains low level features. While going from block0 to block6, the conditional GAN and SS-GAN achieve much better representation results. The conditional GAN benefits from the supervised lables in layers closer to the classification head. However, the unconditional GAN attains worse result at the last layer and the rotation only model gets decreasing quality with more training steps. When combining the self-supervised loss and the adversarial loss, SS-GAN representation quality becomes stable and outperforms the other models.

Figure 10 and Figure 11 show the correlation between top-1 accuracy and FID score. We report the FID and top-1 accuracy from training steps 10k to 100k on CIFAR10, and 100k to 1M on IMAGENET. We evaluate 10×3 models in total, where 10 is the number of training steps at which we evaluate and 3 is the number of random seeds for each run. The collapsed models with FID score larger than 100 are removed from the plot. Overall, the representation quality and the FID score is correlated for all methods on the CIFAR10 dataset. On IMAGENET, only SS-GAN gets better representation quality with better sample quality on block4 and block5.

Method	Uncond-GAN	Cond-GAN	Rot-only	SS-GAN (sBN)
Block0	0.719 ± 0.002	0.719 ± 0.003	0.710 ± 0.002	**0.721 ± 0.002**
Block1	0.762 ± 0.001	0.759 ± 0.003	0.749 ± 0.003	**0.774 ± 0.003**
Block2	0.778 ± 0.001	0.776 ± 0.005	0.762 ± 0.003	**0.796 ± 0.005**
Block3	0.776 ± 0.005	0.780 ± 0.006	0.752 ± 0.006	**0.799 ± 0.003**
Best	0.778 ± 0.001	0.780 ± 0.006	0.762 ± 0.003	**0.799 ± 0.003**

Table 8: Top-1 accuracy on CIFAR10 with standard variations.

Method	Uncond-GAN	Cond-GAN	Rot-only	SS-GAN (sBN)
Block0	0.074 ± 0.074	0.156 ± 0.002	0.147 ± 0.001	**0.158 ± 0.001**
Block1	0.063 ± 0.103	0.187 ± 0.010	0.134 ± 0.003	**0.222 ± 0.001**
Block2	0.073 ± 0.124	0.217 ± 0.007	0.158 ± 0.003	**0.250 ± 0.001**
Block3	0.083 ± 0.142	0.272 ± 0.014	0.202 ± 0.005	**0.327 ± 0.001**
Block4	0.077 ± 0.132	0.253 ± 0.040	0.196 ± 0.001	**0.358 ± 0.005**
Block5	0.074 ± 0.126	0.337 ± 0.010	0.195 ± 0.029	**0.383 ± 0.007**
Best	0.083 ± 0.142	0.337 ± 0.010	0.202 ± 0.005	**0.383 ± 0.007**

Table 9: Top-1 accuracy on IMAGENET with standard variations.

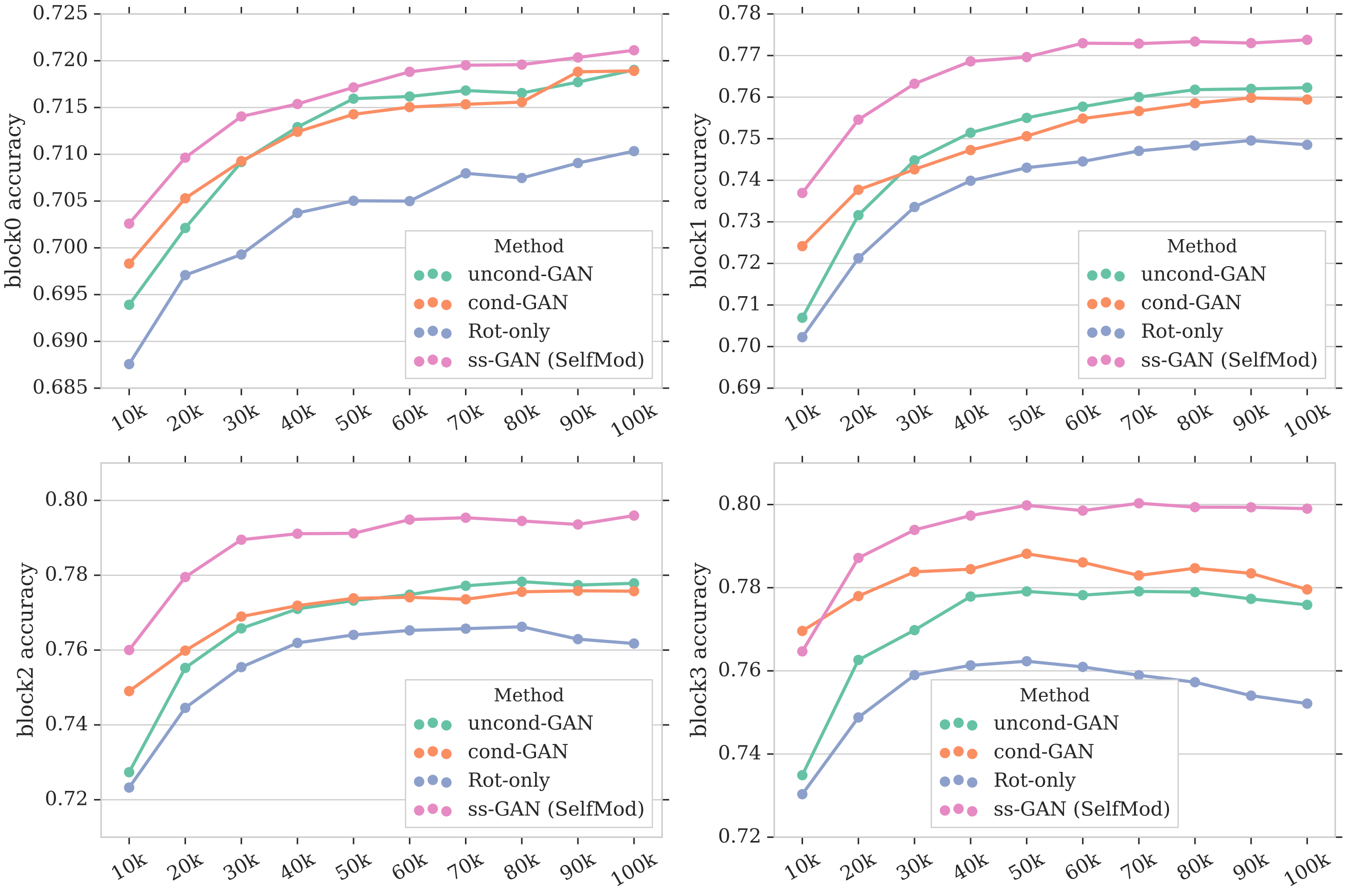


Figure 8: Top 1 accuracy on CIFAR10 with training steps from 10k to 100k.

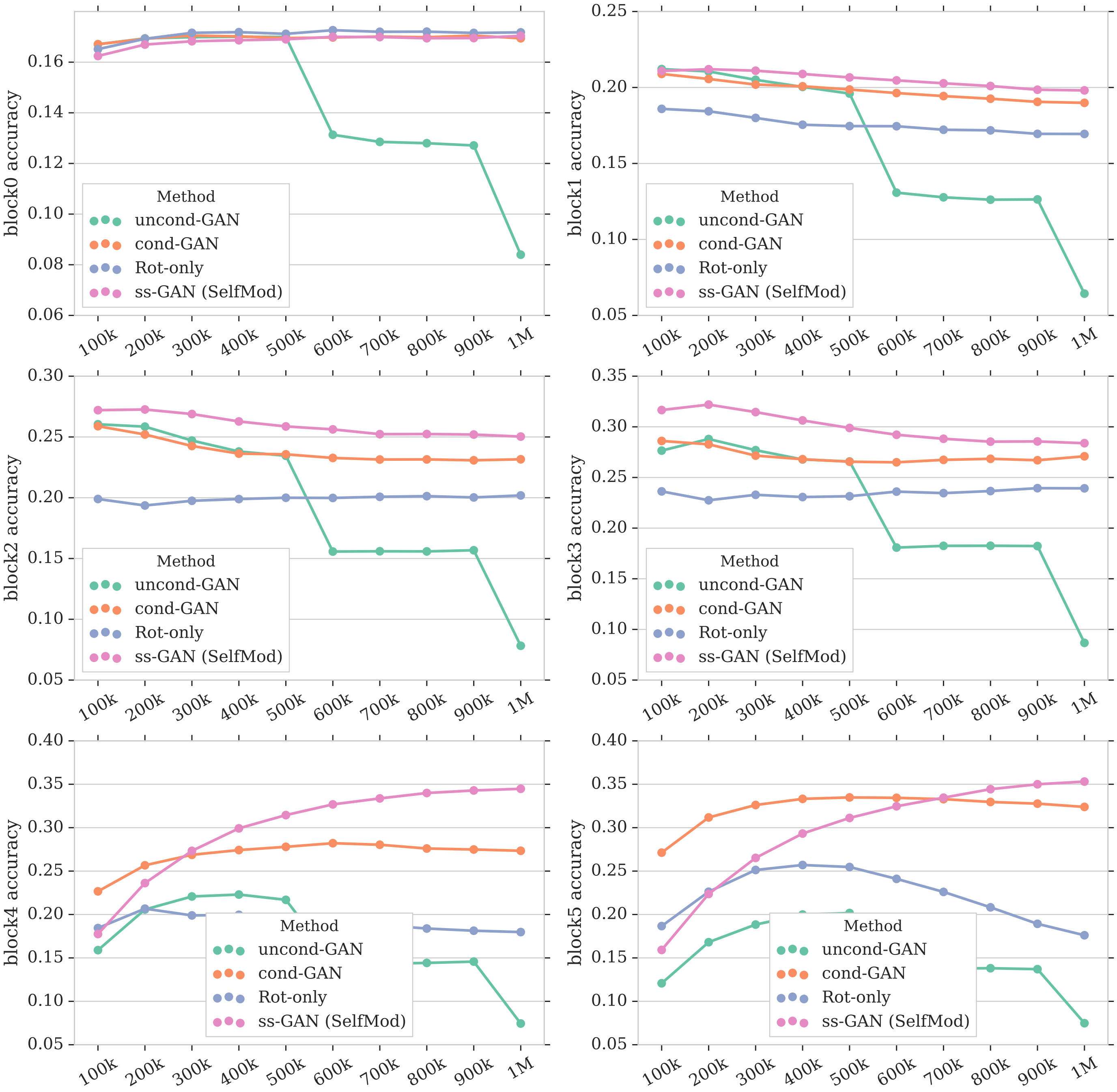


Figure 9: Top 1 accuracy on IMAGENET validation set with training steps from 10k to 1M.

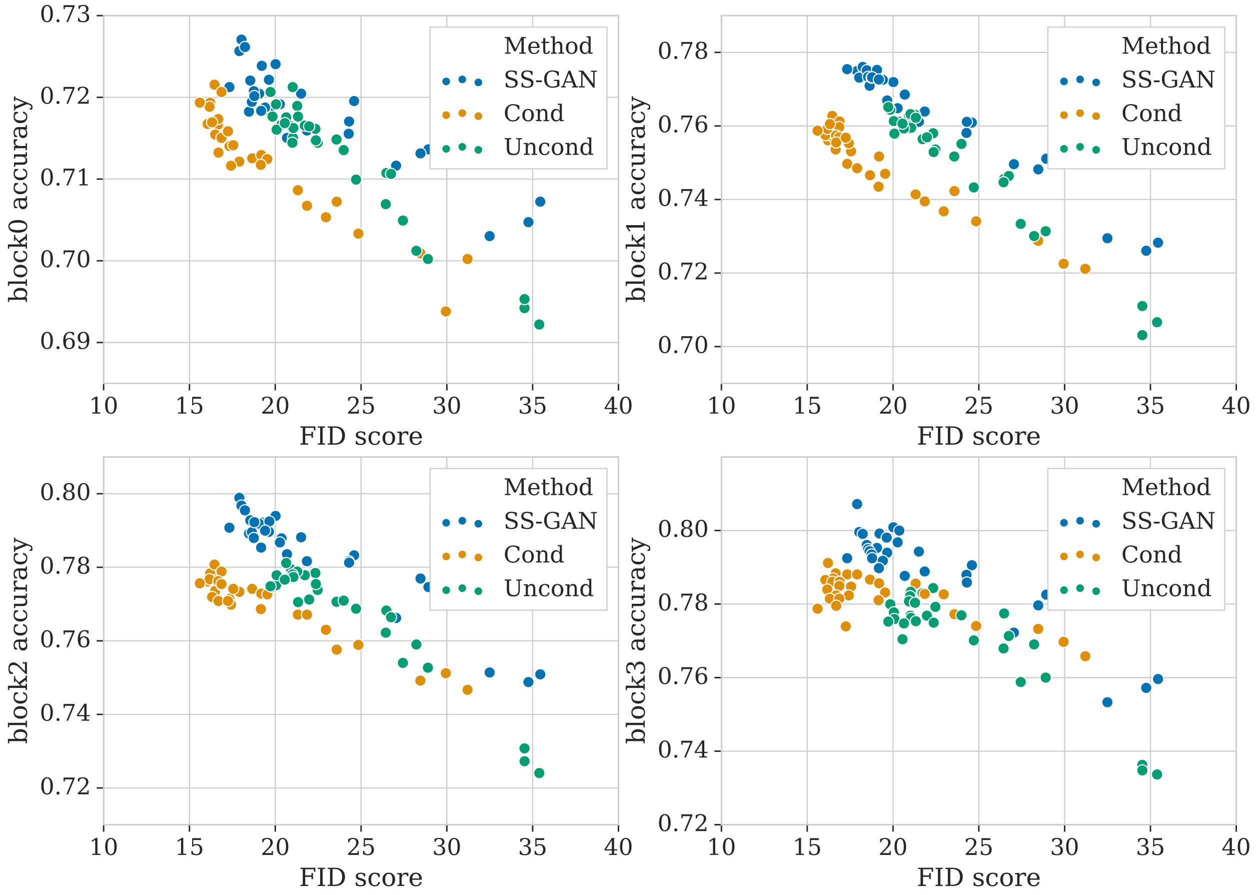


Figure 10: Correlation between top-1 accuracy and FID score for different numbers of GAN training steps from 10k to 100k on CIFAR10. Overall, the representation quality and the FID score is correlated for all methods. The representation quality varies up to 4% with the same FID score.

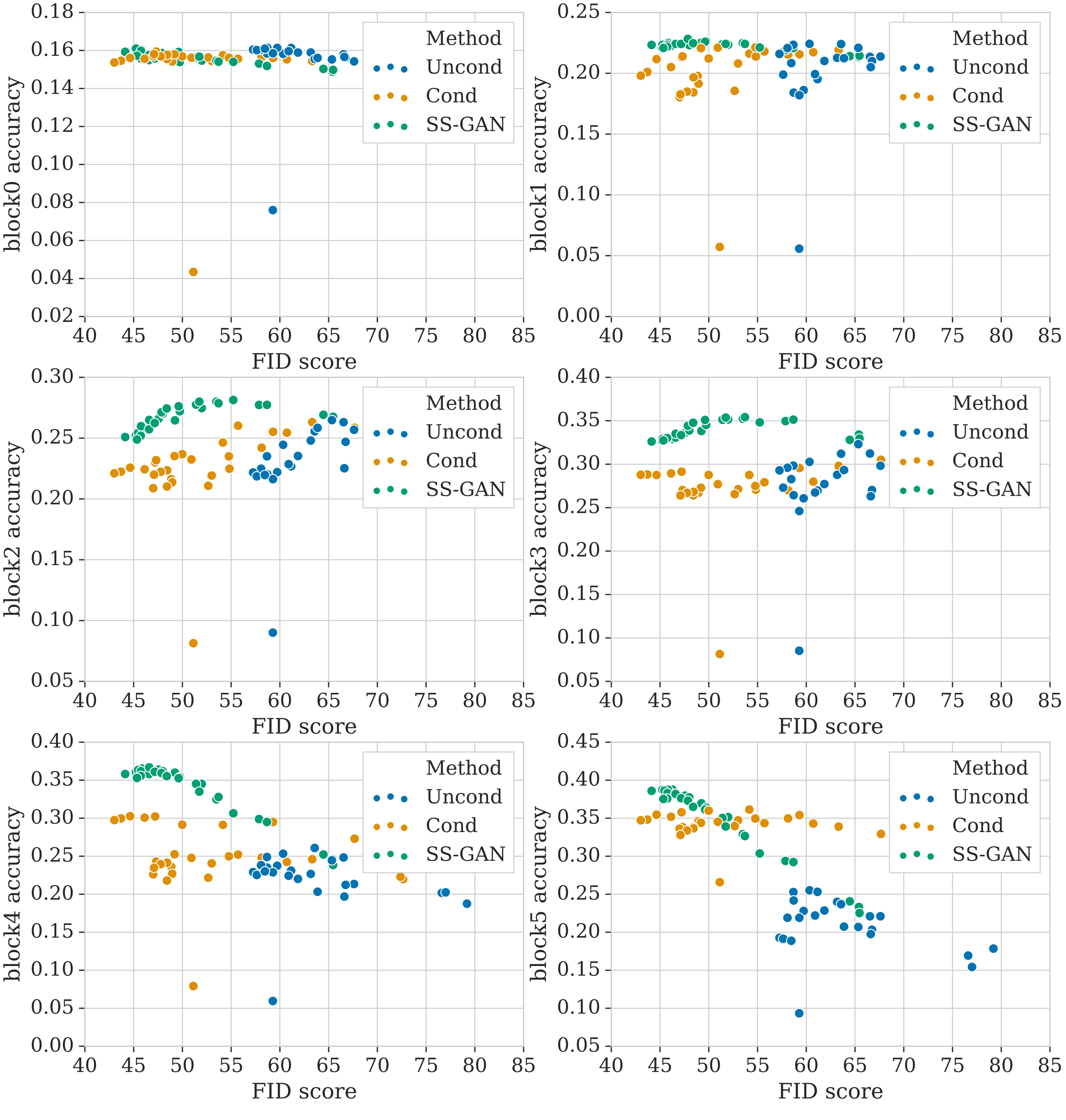


Figure 11: Correlation between top-1 accuracy and FID score for different numbers of GAN training steps from 100k to 1M on IMAGENET. Representation quality and FID score are not correlated on any of block0 to block4. This indicates that low level features are being extracted, which perform similarly on the IMAGENET dataset. Starting from block4, SS-GAN attains better representation as the FID score improves.