Self-supervised GANs with Label Augmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recently, transformation-based self-supervised learning has been applied to generative adversarial networks (GANs) to mitigate the catastrophic forgetting problem of discriminator by learning stable representations. However, the separate self-supervised tasks in existing self-supervised GANs cause an inconsistent goal with generative modeling due to the learning of the generator from their generator distribution-agnostic classifiers. To address this issue, we propose a novel self-supervised GANs framework with label augmentation, i.e., augmenting the GAN labels (real or fake) with the self-supervised pseudo-labels. In particular, the discriminator and the self-supervised classifier are unified to learn a single task that predicts the augmented label such that the discriminator/classifier is aware of the generator distribution, while the generator tries to confuse the discriminator/classifier by optimizing the discrepancy between the transformed real and generated distributions. Theoretically, we prove that the generator, at the equilibrium point, converges to replicate the data distribution. Empirically, we demonstrate that the proposed method significantly outperforms competitive baselines on both generative modeling and representation learning across benchmark datasets.

1 Introduction

Generative adversarial networks (GANs) [11] have developed rapidly to synthesize realistic images in recent years [3, 18, 19]. Conventional GANs consist of a generator and a discriminator, which have undergone adversarial training in a minimax game. The discriminator attempts to distinguish the real data from the fake ones synthesized by the generator. The generator tries to confuse the discriminator to mimic generating the real data. The continuous evolution of the generator results in a non-stationary learning environment for the discriminator. In such a non-stationary environment, the single discriminative signal (real or fake) provides limited and unstable information to the discriminator [28, 6, 57, 45]. Therefore, the discriminator may encounter the catastrophic forgetting problem [20, 5, 7], leading to the training instability of GANs and mode collapse [41, 49, 25, 27] of the generator in learning high-dimensional, complex natural images.

To alleviate the forgetting problem of the discriminator, transformation-based self-supervised tasks such as rotation prediction [10] have been applied to GANs [5]. Self-supervised tasks are designed to predict the constructed pseudo-labels of data, establishing a stationary learning environment. The discriminator enhanced by the stationary self-supervision is able to learn stable representations to resist the forgetting issue, and thus continuously provides informative gradients to obtain a well-performed generator. After obtaining empirical success, self-supervised GAN was analyzed and improved by a new self-supervised task [45]. However, these self-supervised tasks cause an inconsistent goal with generative modeling that try to approximate the data distribution. Besides, the hyper-parameter that maintain the balance between the primary GAN task and the additional self-supervised task bring extra workload to tuning in different datasets. To sum up, we still lack a proper approach to improve the training stability of GANs via self-supervised learning.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Particularly, we found that the undesired goal of existing self-supervised GANs lies in that their self-supervised classifiers are agnostic to the generator distribution because these classifiers try to recognize only the pseudo-label of the transformed real data but not the transformed generated data. Therefore, the classifiers cannot provide the discrepancy between the real and fake data to optimize the generator to approximate the real data distribution. To address this issue, in this paper, we allow the classifier to recognize the pseudo-label of the transformed generated data as well as the transformed real data. We achieve this goal by augmenting the GAN labels (real or fake) with the self-supervised labels to construct new augmented labels and unify the self-supervised classifier with the original discriminator into a new discriminator with classification capability which can distinguish between real and fake data from multiple self-supervised perspectives. Specifically, the discriminator attempts to predict the augmented label of data, establishing a single multi-class classification task in the proposed framework. Therefore, the framework does not require any hyper-parameter to trade-off between the self-supervised objective and the objective that discriminates between the real and fake data unlike existing self-supervised GANs that have two separate classification tasks. The generator tries to confuse the discriminator by generating samples that are predicted as real and not fake with the performed transformation. The proposed framework seamlessly combines two unsupervised learning paradigms, generative adversarial networks and self-supervised learning, to achieve both stable generative modeling and strong representation learning by the generator and the discriminator, respectively, by taking advantage of each other. In the case of generative modeling, the unified adversarial training paradigm allows the generator to unbiasedly learn the data distribution. In the case of representation learning, the joint learning paradigm enables the discriminator to simultaneously capture both semantic information brought by the real/fake-distinguished signals as well as the self-supervised signals to learn comprehensive representations. The major contributions of this paper are summarized as follows:

- We propose novel self-supervised GANs by unifying the GAN task with the transformation-based self-supervised task to obtain the well-performed discriminator and generator.
- We theoretically prove that the optimal generator, at the equilibrium point, approximates the data distribution under the reverse Kullback–Leibler (KL) divergence measure, which is unbiased in learning the data distribution, unlike existing self-supervised GANs.
- We empirically demonstrate that the proposed method significantly outperforms competitive baselines on both generative modeling and representation learning.

2 Preliminaries

In this section, we introduce the concepts and analyze the drawbacks of the standard minimax GAN and existing transformation-based self-supervised GANs.

2.1 Generative Adversarial Networks

Learning the underlying data distribution $\mathcal{P}_d$, which is hard to specify but easy to sample from, lies at the heart of generative modeling. In deep latent variable generative models, the generator $G : \mathcal{Z} \to \mathcal{X}$ maps a latent code $z \in \mathcal{Z}$ endowed with a tractable prior $\mathcal{P}_z$ to a data sample $x \in \mathcal{X}$, and thus implies a generator distribution $\mathcal{P}_g = G \circ \mathcal{P}_z$. To approximate the data distribution through the generator, i.e., $\mathcal{P}_g = \mathcal{P}_d$, one can minimize the Jensen–Shannon (JS) divergence between the data distribution and the generator distribution, i.e., $\min_G \mathrm{JS}(\mathcal{P}_d \| \mathcal{P}_g)$. To estimate this divergence, generative adversarial networks (GANs) [11] leverage a discriminator $D : \mathcal{X} \to \{0, 1\}$ that distinguishes the real data from the fake ones which are generated by the generator. Formally, the objective function for the discriminator and generator of the minimax GAN is defined as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim \mathcal{P}_d}[\log D(1|x)] + \mathbb{E}_{x \sim \mathcal{P}_g}[\log D(0|x)], \tag{1}$$

with $D(1|x) + D(0|x) = 1$ and $D(\cdot|x) \geq 0$. Although GANs have such theoretical basis, the non-stationary learning environment caused by the evolution of the generator results in the discriminator forgetting problem [5], which means that the discriminator forgets the previously learned experiences by overfitting the current data and may thereby hurt the performance of GANs.

2.2 Self-supervised GAN

To mitigate the forgetting problem of the discriminator, self-supervised GANs (SSGAN) [5] introduced an auxiliary self-supervised rotation prediction task to a classifier $C : \mathcal{Y} \to \{1, 2, \cdots, K\}$ with shared layers of the discriminator, and also forced the generator to collaborate with the classifier on the rotation detection task. The objective functions of SSGAN are given by:

$$\begin{aligned} &\max_{D,C} V(G, D) + \lambda_d \cdot \mathbb{E}_{x\sim\mathcal{P}_d}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(k|T_k(x))] \\ &\min_{G} V(G, D) - \lambda_g \cdot \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(k|T_k(x))], \end{aligned} \tag{2}$$

where λ_d and λ_g are two hyper-parameters that balance the GAN task and the self-supervised task, $\mathcal{T} = \{T_k\}_{k=1}^K$ is the set of data transformations such as rotation ($\mathcal{T} = \{0°, 90°, 180°, 270°\}$). The transformation is uniformly distributed, i.e., $p(T_k) = \frac{1}{K}, \forall T_k \in \mathcal{T}$.

Theorem 1 ([45]). *Given the optimal classifier C^* in SSGAN, at the equilibrium point, maximizing the self-supervised task of the generator is equivalent to:*

$$\max_{G} \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{y\sim\mathcal{P}_g^{T_k}} \log\left(\frac{p_d^{T_k}(y)}{\sum_{k=1}^{K} p_d^{T_k}(y)}\right)\right], \tag{3}$$

where $p_g^{T_k}(y) = \int \delta(y - T_k(x))p_g(x)dx$, $p_d^{T_k}(y) = \int \delta(y - T_k(x))p_d(x)dx$ represent the density of transformed generated data or real data, respectively, under transformation by T_k.

Proof. Please refer to Appendix A.1. □

Theorem 1 reveals that the self-supervised task enforces the generator to produce rotation-detectable images other than the real images [45], causing an inconsistent goal with generative modeling that minimizes the divergence between the data distribution and the generator distribution. Therefore, SSGAN requires the original GAN task to learn the real data distribution, which forms a multi-task learning framework that needs hyper-parameters to trade off the two separate tasks.

2.3 SSGAN with Multi-class Minimax Game

To improve the self-supervised task in SSGAN, SSGAN-MS [45] proposed a multi-class minimax self-supervised task by adding the fake class $K + 1$ to the classifier $C : \mathcal{Y} \to \{1, 2, \cdots, K + 1\}$, whereby the generator and the discriminator also compete on the self-supervised task in addition to the original GAN task. The objective functions of SSGAN-MS are given by:

$$\begin{aligned} &\max_{D,C} V(G, D) + \lambda_d \cdot (\mathbb{E}_{x\sim\mathcal{P}_d}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(k|T_k(x))] + \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(K+1|T_k(x))]) \\ &\min_{G} V(G, D) - \lambda_g \cdot (\mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(k|T_k(x))] - \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log C(K+1|T_k(x))]). \end{aligned} \tag{4}$$

Theorem 2. *Given the optimal classifier C^* in SSGAN-MS, at the equilibrium point, maximizing the self-supervised task of the generator is equivalent to:*

$$\max_{G} -KL(\mathcal{P}_g^T \| \mathcal{P}_d^T) + \frac{1}{K}\sum_{k=1}^{K}\left[\mathbb{E}_{y\sim\mathcal{P}_g^{T_k}} \log\left(\frac{p_d^{T_k}(y)}{\sum_{k=1}^{K} p_d^{T_k}(y)}\right)\right], \tag{5}$$

where $\mathcal{P}_g^T$, $\mathcal{P}_d^T$ represent the mixture distribution of transformed generated data or real data with density of $p_g^T(y) = \sum_{k=1}^{K} p(T_k)p_g^{T_k}(y)$ or $p_d^T(y) = \sum_{k=1}^{K} p(T_k)p_d^{T_k}(y)$, respectively.

Proof. Please refer to Appendix A.2. □

Theorem 2 states that, however, the improved self-supervised task for the generator of SSGAN-MS under the optimal classifier retains the inconsistent goal in SSGAN that enforces the generator to produce rotation-detectable images. In addition, minimizing $\mathrm{KL}(\mathcal{P}_g^T \| \mathcal{P}_d^T)$ cannot guarantee to recover the data distribution by the generator (i.e., $\mathcal{P}_g = \mathcal{P}_d$) under the used data transformation setting. Due to the existence of these issues, SSGAN-MS also needs adversarial training between the generator and the discriminator to learn the real data distribution. We attribute the learning paradigm of SSGAN-MS and SSGAN to the multi-task learning framework as shown in Figure 1.

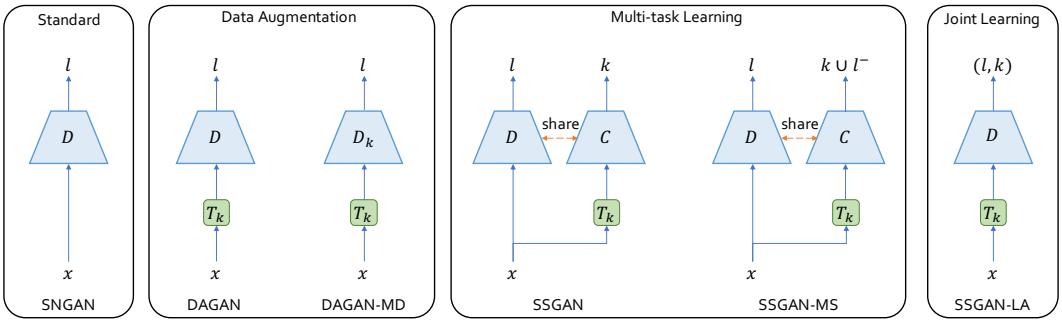


Figure 1: Schematics of the discriminators and classifiers of competitive baselines and the proposed SSGAN-LA. We systematically divide these methods into four categories: Standard (GAN), Data Augmentation (DAGAN and DAGAN-MD), Multi-task Learning (SSGAN and SSGAN-MS), and Joint Learning (SSGAN-LA). We denote l as the GAN label (l^- is the fake label) and k as the self-supervised label.

3 Proposed Methods

The classifiers in existing self-supervised GANs are trained to unequally recognize the corresponding angles for only the rotated real images but not the rotated fake images, resulting in that their classifiers are agnostic to the density of the current generator $p_g(x)$ (see Equation 16 and 20 in Appendix A). In other words, their classifiers do not know how far away the current generator distribution $\mathcal{P}_g$ is from the data distribution $\mathcal{P}_d$. Therefore, their self-supervised classifiers cannot provide any guidance to match the generator distribution with the data distribution. Consequently, existing self-supervised GANs suffer from the problem of involving undesired goals for the generator when learning from the generator-agnostic self-supervised classifiers.

To solve this problem, we propose to allow the classifier to recognize the corresponding angles for both rotated real and fake images to enable it to be aware of the densities of the data distribution and the generator distribution. Specifically, we augment the original GAN labels (real or fake) with the self-supervised labels into new augmented labels, and unify the classifier with the discriminator into a new discriminator with self-supervised classification ability to predict the augmented joint label, forming a single multi-class classification task for the discriminator, instead of the single binary classification in standard GANs and two separate classification tasks in existing self-supervised GANs (see Figure 1 for visualization). Formally, the objective function of the discriminator in the proposed self-supervised GANs with label augmentation (SSGAN-LA) is defined as follows:

$$\max_{D} \mathbb{E}_{x\sim\mathcal{P}_d}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D(k,1|T_k(x))] + \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D(k,0|T_k(x))], \tag{6}$$

where $D(k,i|T_k(x))$ outputs the softmax probability of a transformed data $T_k(x)$ that classified as the augmented joint label of real ($i=1$) or fake ($i=0$) and the k-th transformation.

Proposition 1. *For any fixed generator, given a data $y \in \mathcal{Y}$ that drawn from mixture distribution of transformed data, the optimal discriminator of SSGAN-LA outputs:*

$$D(k,1|y) = \frac{p_d^{T_k}(y)}{\sum_{k=1}^{K} p_d^{T_k}(y) + p_g^{T_k}(y)}, D(k,0|y) = \frac{p_g^{T_k}(y)}{\sum_{k=1}^{K} p_d^{T_k}(y) + p_g^{T_k}(y)}. \tag{7}$$

Proof. Please refer to Appendix A.3. □

Proposition 1 exhibits that the optimal discriminator of SSGAN-LA is able to access the densities of both real and generated data distribution under every transformation ($p_d^{T_k}(x)$ and $p_g^{T_k}(x)$), thus can provide the discrepancy between the data distribution and the generator distribution to the generator.

Based on this analysis, we now propose to optimize the generator of the proposed SSGAN-LA by optimizing such discrepancy using the objective function formulated as follows:

$$\max_{G} \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D(k,1|T_k(x))] - \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D(k,0|T_k(x))]. \tag{8}$$

The objective function of the generator contains both the saturating and non-saturating losses of the discriminator, establishing an adversarial game against the discriminator that can unbiasedly learn the data distribution as we will prove below.

Theorem 3. *The objective function of the generator, given the optimal discriminator, boils down to:*

$$\min_G \frac{1}{K} \sum_{k=1}^{K} KL(\mathcal{P}_g^{T_k} \| \mathcal{P}_d^{T_k}). \tag{9}$$

The global minimum is achieved if and only if $\mathcal{P}_g = \mathcal{P}_d$ when every transformation T_k is affine.

Proof. Please refer to Appendix A.4. □

Theorem 3 reveals that the generator of SSGAN-LA is encouraged to approximate the data distribution under the reverse KL divergence without any inconsistent goal with generative modeling when performing affine transformation such as rotation, unlike existing self-supervised GANs.

Compared with standard GANs, the transformation prediction especially on real samples of SSGAN-LA establishes stationary self-supervised signals to prevent the discriminator from the forgetting problem. The multiple discriminative signals from different transformations are able to provide information-rich gradients to obtain a well-performed generator. Compared with existing self-supervised GANs, in addition to the unbiased learning objective of the generator, the discriminator of SSGAN-LA also has better representation learning ability. The discriminator utilizes not only the real images but also the generated images as augmented data to achieve robust rotation-prediction ability, and distinguishes between real and fake images from multiple perspectives via different transformations to obtain strong discrimination ability. These two abilities enable the discriminator to learn more meaningful semantic representations than existing self-supervised GANs and the standard GANs. In summary, SSGAN-LA incorporates stable self-supervised learning into the original GAN objective to improve the training stability on the premise of unbiased learning of the data distribution.

4 The Importance of Self-supervision

Transformation-based self-supervised learning actually brings two potential benefits, i.e., augmented data and the corresponding self-supervision. To verify the importance of the latter, which is used in SSGAN-LA, we introduce two baselines without explicit feedback from the self-supervision.

4.1 Data Augmentation GANs

We first consider data augmentation GANs (DAGAN) by ignoring the self-supervised signals as:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim P_d} \mathbb{E}_{T_k \sim \mathcal{T}} [\log D(1|T_k(x))] + \mathbb{E}_{x \sim P_g} \mathbb{E}_{T_k \sim \mathcal{T}} [\log D(0|T_k(x))]. \tag{10}$$

This approach is mathematically equivalent to IDA [47] and DiffAugment [53]. We consider this formulation as a simplified version of SSGAN-LA as it train with only the GAN labels. Note that we do not aim to compare with state-of-the-art data augmentation GANs in this work. Next, we discuss the indispensability of transformation-based self-supervision.

Proposition 2. *At the equilibrium point of DAGAN, its global optimal generator implies $\mathcal{P}_g^T = \mathcal{P}_d^T$.*

Proof. Please refer to Appendix A.5. □

Proposition 2 suggests that the objective function of DAGAN enables the mixture distribution of transformed generated data $\mathcal{P}_g^T$ to match with the mixture distribution of transformed real data $\mathcal{P}_d^T$. However, this result cannot guarantee to learn the data distribution by the optimal generator when performing a specific type of data transformation such as the used rotation that form a group with respect to the composition operator.

Theorem 4. *If the set of transformations $\mathcal{T}$ forms a group with respect to the composition operator $\circ$, i.e., $(\mathcal{T}, \circ)$ is a group, even though at the equilibrium point of DAGAN, the probability that the generator replicates the data distribution is $\mathbb{P}(\mathcal{P}_g = \mathcal{P}_d | \mathcal{P}_g^T = \mathcal{P}_d^T) = 0$.*

Proof. Please refer to Appendix A.6. □

Theorem 4 states that the objective function of rotation-based DAGAN is an insufficient condition to recover the data distribution as $(\mathcal{T} = \{0^\circ, 90^\circ, 180^\circ, 270^\circ\}, \circ)$ forms a cycle group. Intuitively, if the self-supervision is not used to identify the transformation when they form a group, then the transformed data will be indistinguishable from the original data so that the transformed data may be leaked to the generator. In more detail, the generator will ideally converge to an arbitrary mixture distribution of rotated real images (see the Proof of Theorem 4 in Appendix A.6). Note that all theoretical results in this paper are not limited to rotation but many other types of data transformation, e.g., RBG permutation and patch shuffling. Nonetheless, we follow the practice of SSGAN to use rotation as the data transformation for fairness. To verify the importance of self-supervision while eliminating the effect of the convergence issue of DAGAN, we need another data augmentation baseline that has the same convergence point as the proposed SSGAN-LA.

4.2 DAGAN with Multiple Discriminators

DAG [47] presented a data augmentation GANs method with multiple discriminators $\{D_k\}_{k=1}^K$ (we refer it as DAGAN-MD throughout this paper), which are indexed by the self-supervised signals and share all layers but the last one with each other, to solve the convergence problem of DAGAN. The objective function of DAGAN-MD is formulated as the following:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x\sim\mathcal{P}_d}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D_k(1|T_k(x))] + \mathbb{E}_{x\sim\mathcal{P}_g}\mathbb{E}_{T_k\sim\mathcal{T}}[\log D_k(0|T_k(x))]. \quad (11)$$

We consider DAGAN-MD as a competitive baseline of SSGAN-LA since DAGAN-MD takes the self-supervised signals as input while SSGAN-LA takes them as target (see Figure 1 for illustration).

Theorem 5. *The objective function of the generator, given the optimal discriminators, boils down to:*

$$\min_G \frac{1}{K}\sum_{k=1}^{K} JS(\mathcal{P}_d^{T_k} \| \mathcal{P}_g^{T_k}) \quad (12)$$

The global minimum is achieved if and only if $\mathcal{P}_g = \mathcal{P}_d$ when every transformation T_k is affine.

Proof. Please refer to Appendix A.7. □

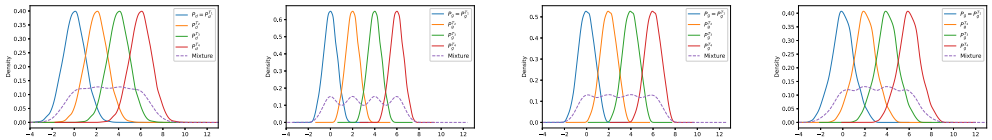
Theorem 5 reveals that the generator of DAGAN-MD can approximate the real data distribution at optima like SSGAN-LA. However, the lack of use of self-supervised signals as supervision will disadvantage the performance of DAGAN-MD in two ways. On one hand, the discriminator cannot ensure to learn stable representations to resist the catastrophic forgetting problem. On the other hand, the discriminator cannot learn high-level representations contained in the specific semantics of different transformations. Particularly, these shared discriminators presumably store the learned features of different transformations into the layer they specified (the last layer) while enforcing invariance to the layers they shared (the residual blocks), which could hurt the performance on representation learning as transformation may modify the semantics of images. As we emphasized above, the more useful information learned by the discriminator, the better guidance provided to the generator and the better optimized result reached by the generator.

5 Experiments

In this section, we first experiment on a one-dimensional Gaussian distribution to demonstrate the unbiased distribution learning ability of the proposed method compared to existing self-supervised methods. We then evaluate the generative modeling and representation learning performance of the proposed method compared with competitive baselines on several real-world benchmark datasets.

5.1 Synthetic Data

We experiment on a simulated dataset to verify that SSGAN-LA can accurately match the data distribution while SSGAN and SSGAN-MS cannot achieve this goal. The real data distribution $\mathcal{P}_d$ is a one-dimensional Gaussian distribution with mean $\mu = 0$ and standard deviation $\sigma = 1$.



(a) Data (b) SSGAN (c) SSGAN-MS (d) SSGAN-LA

Figure 2: Distribution learning results on synthetic data.

Table 1: FID and IS results on CIFAR-10, STL-10, and Tiny-ImageNet, respectively.

Dataset	Metric	BigGAN	SSGAN	SSGAN-MS	DAGAN-MD	SSGAN-LA
CIFAR-10	FID (↓)	10.83	7.52	7.08	7.80	**5.87**
	IS (↑)	8.42 ± 0.13	8.29 ± 0.08	8.45 ± 0.09	8.31 ± 0.07	$\mathbf{8.57 \pm 0.12}$
STL-10	FID (↓)	20.15	16.84	16.46	15.01	**14.58**
	IS (↑)	10.25 ± 0.10	10.36 ± 0.13	10.40 ± 0.09	10.55 ± 0.13	$\mathbf{10.61 \pm 0.14}$
Tiny-ImageNet	FID (↓)	31.01	30.09	25.76	26.85	**24.41**
	IS (↑)	9.80 ± 0.18	10.21 ± 0.15	10.57 ± 0.13	10.28 ± 0.12	$\mathbf{11.14 \pm 0.17}$

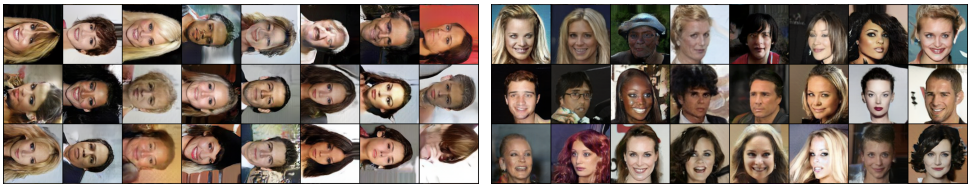
The set of transformations is $\mathcal{T} = \{T_k\}_{k=1}^{K=4}$ with k-th transformation $T_k(x) := x + 2(k-1)$ (T_1 is the identity transformation). Indeed, we assume that the distributions of the real data through different transformations have non-negligible overlap. The generator and discriminator networks are implemented by multi-layer perceptrons with hidden size of 10 and non-linearity of Tanh. Figure 2 shows the density function of the target data and the experimented methods, which are calculated by kernel density estimation [34] through sampling 10k data. SSGAN learns a biased distribution as its self-supervised task forces the generator to produce classifiable data. SSGAN-MS slightly mitigates this issue, but still cannot accurately approximate the data distribution. In contrast, SSGAN-LA can accurately recover the data distribution and achieves the best maximum mean discrepancy (MMD) [12] results (see Appendix C.1). Experimental results on synthetic data accord with theoretical results of SSGAN, SSGAN-MS, and SSGAN-LA.

5.2 Real-world Data

In this section, we experiment with three benchmark datasets: CIFAR-10/100 consists of 50k training images with resolution of 32×32. STL-10 contains 100k training images with resolution of 48×48 as done in [29]. Tiny ImageNet includes 100k training images with resolution of 64×64.

5.2.1 Image Generation

We implement all methods based on unconditional BigGAN [3] without the access of human-annotated labels. We train all methods for 100 epochs with batch size of 100 on all datasets. The optimizer is Adam with betas $(\beta_1, \beta_2) = (0.0, 0.999)$ for both the generator and discriminator. The learning rate for the generator is 2×10^{-4} on CIFAR-10 and STL-10, and 1×10^{-4} on Tiny-ImageNet, while the learning rate for the discriminator/classifier is 2×10^{-4} on CIFAR-10 and STL-10, and 4×10^{-4} on Tiny-ImageNet. All baselines use the hinge loss [24, 44] as the implementation of GAN



(a) DAGAN on CelebA (b) SSGAN-LA on CelebA

Figure 3: Generated samples of DAGAN and SSGAN-LA on CelebA with rotation transformation.

Table 2: Top-1 accuracy (↑) on CIFAR-10, CIFAR-100, and Tiny-ImageNet based on representations across different residual blocks in discriminators.

Dataset	Blocks	BigGAN	SSGAN	SSGAN-MS	DAGAN-MD	SSGAN-LA
CIFAR-10	Block1	**0.6409**	0.6357	0.6397	0.6325	0.6372
	Block2	0.7288	0.7221	0.7281	0.7201	**0.7407**
	Block3	0.7651	0.7690	0.7698	0.7462	**0.7816**
	Block4	0.7686	0.7840	0.7865	0.7675	**0.8027**
CIFAR-100	Block1	0.3226	0.3199	0.3219	**0.3240**	0.3182
	Block2	0.4557	0.4518	0.4614	0.4434	**0.4676**
	Block3	0.4938	0.4933	0.5019	0.4777	**0.5114**
	Block4	0.5036	0.5131	0.5206	0.4923	**0.5426**
Tiny-ImageNet	Block1	0.1421	**0.1376**	0.1322	0.1347	0.1323
	Block2	0.2205	**0.2239**	0.2136	0.2204	0.2133
	Block3	0.2505	0.2772	0.2802	0.2622	**0.2820**
	Block4	0.3245	0.3464	0.3436	0.3404	**0.3490**
	Block5	0.1413	0.3128	0.3412	0.1919	**0.3505**

objective $V(G, D)$ and the proposed SSGAN-LA adopt the multi-hinge loss as defined in Equation 31. As for the detailed network architecture, including the channels of layers and normalization, we follow the BigGAN framework except the output layer of discriminators as different methods solve different tasks. Other hyper-parameters in baselines are the same as the authors' suggestion in their papers unless otherwise specified. To obtain an optimal discriminator of SSGAN-LA as much as possible and because that the discriminator faces a more challenging classification task, we set the discriminator updating steps per generator step as $n_{\text{dis}} = 4$. And we follow the practices in [10, 21, 47] to perform all transformations on each sample for SSGAN-LA and DAGAN-MD. For performance of SSGAN-LA using the same data transformation setting as SSGAN, please refer to Appendix C.2.

For evaluating the overall performance of methods on image generation, we adopt two widely-used evaluation metrics: Fréchet Inception Distance (FID) [14] and Inception Score (IS) [38]. We follow the practice of BigGAN [3] to randomly sample 50K images to calculate IS and FID on all datasets.

Quantitative results. We report the quantitative results of image generation on CIFAR-10, STL-10, and Tiny-ImageNet in Table 1. With auxiliary self-supervised tasks, both SSGAN and SSGAN-MS significantly outperform BigGAN. We believe that the reason is the self-supervised tasks allow their discriminators to learn better representations and improve the anti-forgetting ability, which are also inherited by our method. Compared with BigGAN, DAGAN-MD has made substantial improvements by taking benefits from augmented data. However, it is still inferior to the proposed SSGAN-LA, indicating that only data augmentation but lack of self-supervised signals bring limited improvement. The proposed SSGAN-LA outperforms all competitive baselines, verifying the unbiased distribution learning ability compared to SSGAN and SSGAN-MS and the benefits from the self-supervised signals compared to DAGAN-MD.

Qualitative results. Figure 3 shows images randomly generated by DAGAN and SSGAN-LA on the CelebA dataset with resolution of 64×64. Apparently, the generator of DAGAN converges to approximate the rotated images, which validates Theorem 4 that DAGAN cannot guarantee to recover the original data distribution under the rotation-based transformation. We found this issue of DAGAN in experiments on all datasets, therefore we omit the quantitative comparison with DAGAN. The proposed SSGAN-LA avoids this rotated data leaking problem attributed to explicitly predicting self-supervised pseudo signals.

5.2.2 Image Classification

In order to check whether the discriminator learns meaningful feature representations, we train a 10-way logistic regression classifier on CIFAR-10, 100-way on CIFAR-100, and 200-way on Tiny-ImageNet, respectively, using the representation extracted from each residual block in the discriminator. Specifically, we train the classifier with a batch size of 128 for 50 epochs. The optimizer is Adam with a learning rate of 0.05 and decayed by 10 at both epoch 30 and epoch 40 as done in SSGAN [5]. We do not take any data augmentation to train the linear model.

We report the Top 1 accuracy results in Table 2. Compared to BigGAN, SSGAN and SSGAN-MS perform, to some extent, improvement attributed to adding the extra self-supervised task that can learn more meaningful information about data. There is no significant difference between SSGAN and SSGAN-MS except the deepest block on Tiny-ImageNet. Overall, DAGAN-MD performs substantially worse than the self-supervised methods, and even worse than BigGAN in some cases. These results validate that the multiple discriminators of DAGAN-MD tend to strengthen the expressive ability of the specific output layer but weaken the shared residual blocks. In other words, different output heads learn the specific features of different transformations, and the shared blocks are limited to the invariant features, which may lose useful information. Compared to all competitive baselines, the proposed SSGAN-LA achieves the best Top 1 accuracy across most discriminator blocks, especially the deep blocks. These results verify that the discriminator of SSGAN-LA learns meaningful high-level representations. And such a powerful representation ability of the discriminator, in turn, promotes the improvement of the quality of generated samples.

6 Related Work

6.1 Generative Adversarial Networks

Since generative adversarial networks (GANs) [11] appearance, many variants focus on improving the performance of generative modeling from perspectives of network architectures [37, 51, 4, 3], divergences [33, 2, 13, 23, 42, 16], optimization procedure [28, 40, 15], normalization [29, 39, 54], and regularization [36, 50, 43].

However, most variants still focus on a single signal for training GAN. As for training GANs via multiple signals, [31] proposed dual discriminators to tackle the mode collapse issue. [9] explored an ensemble of multiple discriminators to train the generator. [30] provided multiple signals to the generator via discriminator random projection. [1] used hypervolume maximization algorithm to solve the multi-objective training of GANs with multiple discriminators. [48] extended to output realness probability distribution in the principle of distinguishing samples from multiple angles. [17] utilized gradients at multi-scale to train the generator.

6.2 Self-supervised Learning in GANs

Self-supervised learning is a family of unsupervised representation learning methods by solving pretext tasks. Various self-supervised pretext tasks on image representation learning have been proposed, e.g., predicting context [8], solving jigsaw puzzle [32], and recognizing rotation [10]. To combine the self-supervision with supervised learning, [21] proposed a label augmentation approach that removes the unnecessary invariant property of the classifier.

Recently, self-supervision has been applied to GANs to improve its training stability. [5] introduced an auxiliary rotation detection task onto the discriminator to alleviate the forgetting issue. [45, 46] analyzed the drawback of the self-supervised task in [5] and proposed a multi-class minimax self-supervised task. [26] combined self- and semi-supervised learning to outperform supervised GANs. [35] extended the self-supervision into the latent space by detecting the latent transformation. [22] explored contrastive learning and mutual information maximization to simultaneously tackle the catastrophic forgetting and mode collapse issues. [52] presented a transformation-based consistency regularization, which has improved in [55, 56].

7 Conclusions

In this paper, we propose a novel framework for training unsupervised generative adversarial networks (GANs) with transformation-based self-supervision. Specifically, we unify the original GAN task with the self-supervised task into a single multi-class classification task for the discriminator by augmenting the real/fake labels with the self-supervised labels, and force it to predict the augmented labels. Conversely, the generator tries to fool the discriminator by generating data that are predicted as real and not fake with the performed transformation. We provide both theoretical analysis and empirical evidence to support the superiority of our proposed GAN framework, compared to existing transformation-based self-supervised GANs and data augmentation GANs. In the future, we will explore the application of the proposed framework to semi- and fully-supervised GANs.

References

[1] I. Albuquerque, J. Monteiro, T. Doan, B. Considine, T. Falk, and I. Mitliagkas. Multi-objective training of generative adversarial networks with multiple discriminators. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[2] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

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

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

[5] T. Chen, X. Zhai, M. Ritter, M. Lucic, and N. Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems*, 2016.

[7] Y. Cong, M. Zhao, J. Li, S. Wang, and L. Carin. Gan memory with no forgetting. In *Advances in Neural Information Processing Systems*, 2020.

[8] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673*, 2016.

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

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*. 2014.

[12] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 2012.

[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems 30*. 2017.

[14] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30*. 2017.

[15] L. Hou, Z. Yuan, L. Huang, H. Shen, X. Cheng, and C. Wang. Slimmable generative adversarial networks. *arXiv preprint arXiv:2012.05660*, 2020.

[16] A. Jolicoeur-Martineau. On relativistic f-divergences. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[17] A. Karnewar and O. Wang. Msg-gan: Multi-scale gradients for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[19] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila. Analyzing and improving the image quality of stylegan. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 2017.

[21] H. Lee, S. J. Hwang, and J. Shin. Self-supervised label augmentation via input transformations. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[22] K. S. Lee, N.-T. Tran, and N.-M. Cheung. Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning. *arXiv preprint arXiv:2007.04589*, 2020.

[23] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Poczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, 2017.

[24] J. H. Lim and J. C. Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017.

[25] Z. Lin, A. Khetan, G. Fanti, and S. Oh. Pacgan: The power of two samples in generative adversarial networks. In *Advances in Neural Information Processing Systems*, 2018.

[26] M. Lučić, M. Tschannen, M. Ritter, X. Zhai, O. Bachem, and S. Gelly. High-fidelity image generation with fewer labels. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[27] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[28] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

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

[30] B. Neyshabur, S. Bhojanapalli, and A. Chakrabarti. Stabilizing gan training with multiple random projections. *arXiv preprint arXiv:1705.07831*, 2017.

[31] T. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2017.

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

[33] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems 29*. 2016.

[34] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 1962.

[35] P. Patel, N. Kumari, M. Singh, and B. Krishnamurthy. Lt-gan: Self-supervised gan with latent transformation detection. *arXiv preprint arXiv:2010.09893*, 2020.

[36] H. Petzka, A. Fischer, and D. Lukovnikov. On the regularization of wasserstein GANs. In *International Conference on Learning Representations*, 2018.

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[38] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems 29*. 2016.

[39] A. Sanyal, P. H. Torr, and P. K. Dokania. Stable rank normalization for improved generalization in neural networks and gans. In *International Conference on Learning Representations*, 2020.

[40] F. Schaefer, H. Zheng, and A. Anandkumar. Implicit competitive regularization in GANs. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[41] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, 2017.

[42] C. Tao, L. Chen, R. Henao, J. Feng, and L. C. Duke. Chi-square generative adversarial network. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[43] D. Terjék. Adversarial lipschitz regularization. In *International Conference on Learning Representations*, 2020.

[44] D. Tran, R. Ranganath, and D. M. Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 2017.

[45] N.-T. Tran, V.-H. Tran, B.-N. Nguyen, L. Yang, and N.-M. M. Cheung. Self-supervised gan: Analysis and improvement with multi-class minimax game. In *Advances in Neural Information Processing Systems*, 2019.

[46] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, and N.-M. Cheung. An improved self-supervised gan via adversarial training. *arXiv preprint arXiv:1905.05469*, 2019.

[47] N.-T. Tran, V.-H. Tran, N.-B. Nguyen, T.-K. Nguyen, and N.-M. Cheung. Towards good practices for data augmentation in gan training. *arXiv preprint arXiv:2006.05338*, 2020.

[48] Y. Xiangli, Y. Deng, B. Dai, C. C. Loy, and D. Lin. Real or not real, that is the question. In *International Conference on Learning Representations*, 2020.

[49] C. Xiao, P. Zhong, and C. Zheng. Bourgan: Generative networks with metric embeddings. In *Advances in Neural Information Processing Systems*, 2018.

[50] S. Yamaguchi and M. Koyama. Distributional concavity regularization for GANs. In *International Conference on Learning Representations*, 2019.

[51] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[52] H. Zhang, Z. Zhang, A. Odena, and H. Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2020.

[53] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 2020.

[54] Y. Zhao, C. Li, P. Yu, J. Gao, and C. Chen. Feature quantization improves GAN training. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[55] Z. Zhao, S. Singh, H. Lee, Z. Zhang, A. Odena, and H. Zhang. Improved consistency regularization for gans. *arXiv preprint arXiv:2002.04724*, 2020.

[56] Z. Zhao, Z. Zhang, T. Chen, S. Singh, and H. Zhang. Image augmentations for gan training. *arXiv preprint arXiv:2006.02595*, 2020.

[57] Z. Zhou, J. Liang, Y. Song, L. Yu, H. Wang, W. Zhang, Y. Yu, and Z. Zhang. Lipschitz generative adversarial nets. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Theorem 3.
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] All theoretical results are based on the universal approximation theorem of neural networks.
   (b) Did you include complete proofs of all theoretical results? [Yes] See Appendix A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We follow the training details of the open-source BigGAN repository.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We report them when calculating IS in Table 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] NVIDIA Tesla K80 GPU.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We use publicly available data.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]