TRAINING GANS WITH STRONGER AUGMENTATIONS VIA CONTRASTIVE DISCRIMINATOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent works in Generative Adversarial Networks (GANs) are actively revisiting various data augmentation techniques as an effective way to prevent discriminator overfitting. It is still unclear, however, that which augmentations could actually improve GANs, and in particular, how to apply a wider range of augmentations in training. In this paper, we propose a novel way to address these questions by incorporating a recent *contrastive representation learning* scheme into the discriminator, coined *ContraD*. This "fusion" enables discriminators to work with much stronger augmentations without catastrophic forgetting, which can significantly stabilize GAN training. Even better, we observe that the contrastive learning itself *also* benefits from GAN training, *i.e.*, keeping discriminative features between real and fake samples, suggesting a strong coherence between the two worlds: a good contrastive representation is also good for GAN discriminators, and *vice versa*. Our experimental results show that GAN with ContraD consistently improves FID scores compared to other recent techniques using data augmentations, still maintaining highly discriminative features in the discriminator in terms of the linear evaluation. Finally, as a byproduct, we show that our GANs trained in an unsupervised manner (without labels) can induce many conditional generative models via a simple latent sampling, leveraging the learned features of ContraD.

1 INTRODUCTION

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have become one of the most prominent approaches for generative modeling with a wide range of applications (Ho & Ermon, 2016; Zhu et al., 2017; Karras et al., 2019; Rott Shaham et al., 2019). In general, a GAN is defined by a minimax game between two neural networks: a *generator* network that maps a random vector into the data domain, and a *discriminator* network that classifies whether a given sample is real (from the training dataset) or fake (from the generator). Provided that both generator and discriminator attain their optima at each minimax objective alternatively, it is theoretically guaranteed that the generator implicitly converges to model the data generating distribution (Goodfellow et al., 2014).

Due to the non-convex/stationary nature of the minimax game, however, training GANs in practice is often very unstable with an extreme sensitivity to many hyperparameters (Salimans et al., 2016; Lucic et al., 2018; Kurach et al., 2019). Stabilizing the GAN dynamics has been extensively studied in the literature (Arjovsky et al., 2017; Gulrajani et al., 2017; Miyato et al., 2018; Wei et al., 2018; Jolicoeur-Martineau, 2019; Chen et al., 2019; Schonfeld et al., 2020), and the idea of incorporating *data augmentation* techniques has recently gained a particular attention on this line of research: more specifically, Zhang et al. (2020) have shown that *consistency regularization* between discriminator outputs of clean and augmented samples could greatly stabilize GAN training, and Zhao et al. (2020c) further improved this idea. The question of which augmentations are good for GANs has been investigated very recently in several works (Zhao et al., 2020d; Tran et al., 2020; Karras et al., 2020a; Zhao et al., 2020a), while they unanimously conclude only a limited range of augmentations (*e.g.*, flipping and spatial translation) were actually helpful for the current form of training GANs.

Meanwhile, not only for GANs, data augmentation has also been played a key role in the literature of self-supervised representation learning (Doersch et al., 2015; Gidaris et al., 2018; Wu et al., 2018), especially with the recent advances in *contrastive learning* (Bachman et al., 2019; van den Oord et al., 2018; Chen et al., 2020a;b; Grill et al., 2020): *e.g.*, Chen et al. (2020a) have shown

that the performance gap between supervised- and unsupervised learning can be significantly closed with large-scale contrastive learning over strong data augmentations. In this case, contrastive learning aims to extract the mutual information shared across augmentations, so good augmentations for contrastive learning should keep information relevant to downstream tasks (*e.g.*, classification), while discarding nuisances for generalization. Finding such augmentations is still challenging, yet in some sense, it is more tangible than the case of GANs, as there are some known ways to formulate the goal rigourously, *e.g.*, InfoMax (Linsker, 1988) or InfoMin principles (Tian et al., 2020).

Contribution. In this paper, we propose *Contrastive Discriminator* (ContraD), a new way of training discriminators of GAN that incorporates the principle of contrastive learning. Specifically, instead of directly optimizing the discriminator network for the GAN loss, ContraD uses the network mainly to extract a *contrastive representation* from a given set of data augmentations and (real or generated) samples. The actual discriminator that minimizes the GAN loss is defined independently upon the contrastive representation, which turns out that a simple 2-layer network is sufficient to work as a complete GAN. By design, ContraD can be naturally trained with augmentations used in the literature of contrastive learning, *e.g.*, those proposed by SimCLR (Chen et al., 2020a), which are in fact much stronger than typical practices in the context of GAN training (Zhang et al., 2020; Zhao et al., 2020c;a; Karras et al., 2020a). Our key observation here is that, the task of contrastive learning to discriminate each of independent real samples and that of GAN discriminator to discriminate fake samples from the reals benefit each other when jointly trained with a shared representation.

Self-supervised learning, including contrastive learning, have been recently applied in GAN as an auxiliary task upon the GAN loss (Chen et al., 2019; Tran et al., 2019; Lee et al., 2020b; Zhao et al., 2020d), mainly in attempt to alleviate catastopic forgetting in discriminators (Chen et al., 2019). For conditional GANs, Kang & Park (2020) have proposed a contrastive form of loss to efficiently incorporate a given conditional information into discriminators. Our work can be differentiated to these prior works in a sense that, to the best of our knowledge, it is the first method that successfully leverage contrastive learning alone to incorporate a wide range of data augmentations in GAN training. Indeed, for example, Zhao et al. (2020d) recently reported that simply regularizing auxiliary SimCLR loss (Chen et al., 2020a) improves GAN training, but could not outperform existing methods based on simple data augmentations, *e.g.*, bCR (Zhao et al., 2020c).

2 Background

Generative adversarial networks. We consider a problem of learning a generative model p_g from a given dataset $\{\mathbf{x}_i\}_{i=1}^N$, where $\mathbf{x}_i \sim p_{\text{data}}$ and $\mathbf{x}_i \in \mathcal{X}$. To this end, *generative adversarial network* (GAN) (Goodfellow et al., 2014) considers two neural networks: (a) a *generator* network $G : \mathcal{Z} \to \mathcal{X}$ that maps a latent variable $\mathbf{z} \sim p(\mathbf{z})$ into $\mathcal{X}$, where $p(\mathbf{z})$ is a specific prior distribution, and (b) a *discriminator* network $D : \mathcal{X} \to [0, 1]$ that discriminates samples from p_{data} and those from the implicit distribution p_g derived from $G(\mathbf{z})$. The primitive form of training G and D is the following:

$$\min_G \max_D V(G, D) := \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{1}$$

For a fixed G, the inner maximization objective (1) with respect to D leads to the following *optimal discriminator* D^*_G, and consequently the outer minimization objective with respect to G becomes to minimize the *Jensen-Shannon divergence* between p_{data} and p_g: $D^*_G := \max_D V(G, D) = \frac{p_{\text{data}}}{p_{\text{data}}+p_g}$. Although this formulation (1) theoretically guarantees $p^*_g = p_{\text{data}}$ as the global optimum, the *non-saturating loss* (Goodfellow et al., 2014) is more favored in practice for better optimization stability:

$$\begin{aligned} &\min_D L(D) := -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[\log(D(\mathbf{x}))] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))], \text{ and} \\ &\min_G L(G) := -\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(D(G(\mathbf{z})))]. \end{aligned} \tag{2}$$

Here, the only difference compared to (1) is at the minimization objective for G, where now G is optimized to let D to classify $G(\mathbf{z})$ as 1, *i.e.*, the "real".

Contrastive representation learning. Consider two random variables $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$, which are often referred as *views*. Generally speaking, *contrastive learning* aims to extract a useful representation of $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$ from learning a function that identifies whether a given sample is from $p(\mathbf{v}^{(1)})p(\mathbf{v}^{(2)}|\mathbf{v}^{(1)})$ or $p(\mathbf{v}^{(1)})p(\mathbf{v}^{(2)})$, *i.e.*, whether two views are dependent or not. More specifically, the function estimates the *mutual information* $I(\mathbf{v}^{(1)}; \mathbf{v}^{(2)})$ between the two views. To this

end, van den Oord et al. (2018) proposed to minimize *InfoNCE* loss, which turns out to maximize a lower bound of $I(\mathbf{v}^{(1)};\mathbf{v}^{(2)})$. Formally, assuming that $\mathbf{v}_i^{(2)} \sim p(\mathbf{v}^{(2)}|\mathbf{v}_i^{(1)})$ while $\mathbf{v}_j^{(2)} \sim p(\mathbf{v}^{(2)})$ for $j = 1, \cdots, K$, the InfoNCE loss is defined by:

$$L_{\texttt{NCE}}(\mathbf{v}^{(1)},\mathbf{v}^{(2)};s) := -\mathbb{E}_{\mathbf{v}_i^{(1)}}\left[\log \frac{\exp(s(\mathbf{v}_i^{(1)},\mathbf{v}_i^{(2)}))}{\sum_{j=1}^{K}\exp(s(\mathbf{v}_i^{(1)},\mathbf{v}_j^{(2)}))}\right], \tag{3}$$

where $s(\cdot,\cdot)$ is the score function that models the log-density ratio of $p(\mathbf{v}^{(2)}|\mathbf{v}^{(1)})$ to $p(\mathbf{v}^{(2)})$, possibly including some parametrized encoders for $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$.

Many of recent unsupervised representation learning methods are based on this general framework of contrastive learning (Wu et al., 2018; Bachman et al., 2019; Hénaff et al., 2019; He et al., 2019; Chen et al., 2020a). In this paper, we primarily focus on one of these methods called *SimCLR* (Chen et al., 2020a), that adopts a wide range of independent data augmentations to define the views: more specifically, for a given set of data samples $\mathbf{x} = [\mathbf{x}_i]_{i=1}^N$, SimCLR applys two independent augmentations, namely $\mathbf{t}_1$ and $\mathbf{t}_2$, to the given data to obtain $\mathbf{v}^{(1)}$ and $\mathbf{v}^{(2)}$, *i.e.*, $(\mathbf{v}^{(1)},\mathbf{v}^{(2)}) := (\mathbf{t}_1(\mathbf{x}),\mathbf{t}_2(\mathbf{x}))$. The actual loss of SimCLR is slightly different to InfoNCE, mainly due to practical considerations for sample efficiency:

$$L_{\texttt{SimCLR}}(\mathbf{v}^{(1)},\mathbf{v}^{(2)}) := \frac{1}{2}\left(L_{\texttt{NCE}}(\mathbf{v}^{(1)},[\mathbf{v}^{(2)},\mathbf{v}_{-i}^{(1)}]_i;s_{\texttt{SimCLR}}) + L_{\texttt{NCE}}(\mathbf{v}^{(2)},[\mathbf{v}^{(1)},\mathbf{v}_{-i}^{(2)}]_i;s_{\texttt{SimCLR}})\right), \tag{4}$$

where $\mathbf{v}_{-i} := \mathbf{v} \setminus \{\mathbf{v}_i\}$. For $s_{\texttt{SimCLR}}$, SimCLR specifies to use (a) an *encoder* network $f : \mathcal{X} \to \mathbb{R}^{d_e}$, (b) a small neural network called *projection head* $h : \mathbb{R}^{d_e} \to \mathbb{R}^{d_p}$, and (c) the *normalized temperature-scaled cross entropy* (NT-Xent). Putting altogether, $s_{\texttt{SimCLR}}$ is defined by:

$$s_{\texttt{SimCLR}}(\mathbf{v}^{(1)},\mathbf{v}^{(2)};f,h) := \frac{h(f(\mathbf{v}^{(1)}))\cdot h(f(\mathbf{v}^{(2)}))}{\tau\cdot||h(f(\mathbf{v}^{(1)}))||_2||h(f(\mathbf{v}^{(2)}))||_2}, \tag{5}$$

where τ is a temperature hyperparameter. Once the training is done with respect to $L_{\texttt{SimCLR}}$, the projection head h is discarded and f is served as the learned representation for downstream tasks.

3 ContraD: Contrastive Discriminator for GANs

We aim to develop a training scheme for GANs that is capable to extract more useful information under stronger data augmentation beyond the existing yet limited practices, *e.g.*, random translations up to few pixels (Zhang et al., 2020; Zhao et al., 2020c;a). For examples, one may consider to apply augmentations introduced in SimCLR (Chen et al., 2020a) for training GANs, *i.e.*, a stochastic composition of random resizing, cropping, horizontal flipping, color distortion, and Gaussian blurring. Indeed, existing approaches that handles data augmentation in GANs are not guranteed to work best in this harsh case (as also observed empirically in Table 5), and some of them are even expected to harm the performance: *e.g.*, applying consistency regularization (Zhang et al., 2020; Zhao et al., 2020c), which regularizes a discriminator to be *invariant* to these augmentations, could overly restrict the expressivity of the discriminator to learn meaningful features.

Our proposed architecture of *Contrastive Discriminator* (ContraD) is designed to alleviate such potential risks from using stronger data augmentations by incorporating the contrastive learning scheme of SimCLR (Chen et al., 2020a), which arguably works with those augmentations, and its design practices into the discriminator. More concretely, the main objective of ContraD is not to minimize the discriminator loss of GAN, but to learn a contrastive representation that is *compatible* to GAN. This means that the objective does not break the contrastive learning, while the representation still contains sufficient information to discriminate real and fake samples, so that a small neural network discriminator is enough to perform its task upon the representation. We describe how to learn such a representation of ContraD in Section 3.1, and how to incorporate this in the actual GAN training in Section 3.2. Finally, Section 3.3 introduces a natural application of ContraD for deriving conditional generative models from an unconditionally trained GAN. Figure 1 illustrates ContraD.

3.1 Contrastive representation for GANs

Recall the standard training of GANs, *i.e.*, we train $G : \mathcal{Z} \to \mathcal{X}$ and $D : \mathcal{X} \to [0,1]$ via optimizing (2) from a given dataset $\{\mathbf{x}_i\}_{i=1}^N$, and let $\mathcal{T}$ be a family of possible transforms for data augmentation,

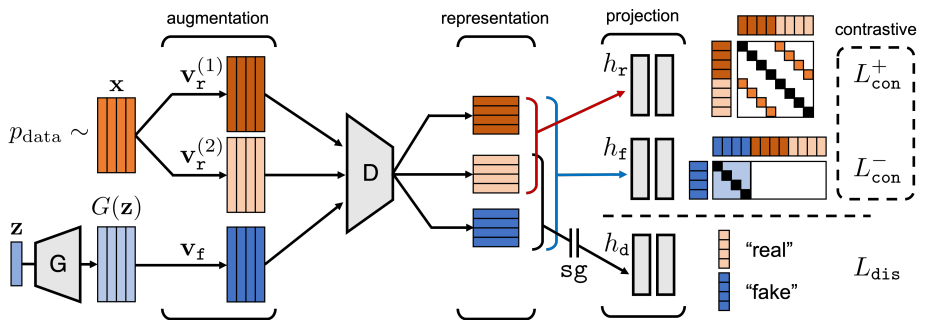

Figure 1: An overview on our proposed method: *Contrastive Discriminator* (ContraD). Overall, the representation of ContraD is not learned from the discriminator loss ($L_{\mathtt{dis}}$), but from two contrastive losses $L_{\mathtt{con}}^{+}$ and $L_{\mathtt{con}}^{-}$, each is for real and fake samples, respectively.

which we assume in this paper to be those used in SimCLR (Chen et al., 2020a). In order to define the training objective of ContraD, we start by re-defining the scalar-valued D to have vector-valued outputs, *i.e.*, $D : \mathcal{X} \to \mathbb{R}^{d_e}$, to represent an encoder network of contrastive learning. Overall, the encoder network D of ContraD is trained by minimizing two different *contrastive* losses: (a) the SimCLR loss (4) on the real samples, and (b) the supervised contrastive loss (Khosla et al., 2020) on the fake samples, which will be explained one-by-one in what follows.

Loss for real samples. By default, ContraD is built upon the SimCLR training: in fact, the training becomes equivalent to SimCLR if the loss (b) is missing, *i.e.*, without considering the fake samples. Here, we attempt to simply follow the contrastive training scheme for real samples $\mathbf{x}$, to keep open the possibility to improve the method by adopting other self-supervised learning methods with data augmentations (Chen et al., 2020b; Grill et al., 2020). More concretely, we first compute two independent views $\mathbf{v}_{\mathtt{r}}^{(1)}, \mathbf{v}_{\mathtt{r}}^{(2)} := \mathbf{t}_1(x), \mathbf{t}_2(x)$ for real samples using $\mathbf{t}_1, \mathbf{t}_2 \sim \mathcal{T}$, and minimize:

$$L_{\mathtt{con}}^{+}(D, h_{\mathtt{r}}) := L_{\mathtt{SimCLR}}(\mathbf{v}_{\mathtt{r}}^{(1)}, \mathbf{v}_{\mathtt{r}}^{(2)}; D, h_{\mathtt{r}}), \tag{6}$$

where $h_{\mathtt{r}} : \mathbb{R}^{d_e} \to \mathbb{R}^{d_p}$ is a projection head for this loss.

Loss for fake samples. Although the representation learned via $L_{\mathtt{con}}^{+}$ (6) may be enough, *e.g.*, for discriminating two independent real samples, it does not give any guarantee that this representation would still work for discriminating real and fake samples, which is needed as a GAN discriminator. We compensate this by considering an auxiliary loss that regularizes the encoder to keep necessary information to discriminate real and fake samples. Here, several design choices can be possible: nevertheless, we observe that one should choose the loss deliberately, otherwise the contrastive representation from $L_{\mathtt{con}}^{+}$ could be significantly affected. For example, we found using the original GAN loss (2) for this could completely negate the effectiveness of ContraD.

In this respect, we propose to use the *supervised contrastive loss* (Khosla et al., 2020) over fake samples to this end, an extended version of contrastive loss to support supervised learning by allowing more than one view to be positive, so that views of the same label can be attracted to each other in the embedding space. In our case, we assume all the views from fake samples have the same label against those from real samples. Formally, for each $\mathbf{v}_i^{(1)}$, let $V_{i+}^{(2)}$ be a subset of $\mathbf{v}^{(2)}$ that represent the positive views for $\mathbf{v}_i^{(1)}$. Then, the supervised contrastive loss is defined by:

$$L_{\mathtt{SupCon}}(\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, V_{+}^{(2)}) := -\mathbb{E}_{\mathbf{v}_i^{(1)}} \left[\frac{1}{|V_{i+}^{(2)}|} \log \frac{\sum_{\mathbf{v}_{i+}^{(2)} \in V_{i+}^{(2)}} \exp(s_{\mathtt{SimCLR}}(\mathbf{v}_i^{(1)}, \mathbf{v}_{i+}^{(2)}))}{\sum_j \exp(s_{\mathtt{SimCLR}}(\mathbf{v}_i^{(1)}, \mathbf{v}_j^{(2)}))} \right]. \tag{7}$$

Using the notation, we define the ContraD loss for fake samples as follows:

$$L_{\mathtt{con}}^{-}(D, h_{\mathtt{f}}) := L_{\mathtt{SupCon}}(\mathbf{v}_{\mathtt{f}}, [\mathbf{v}_{\mathtt{f},-i}, \mathbf{v}_{\mathtt{r}}^{(1)}, \mathbf{v}_{\mathtt{r}}^{(2)}]_i, [\mathbf{v}_{\mathtt{f},-i}]_i; D, h_{\mathtt{f}}), \tag{8}$$

where again $\mathbf{v}_{\mathtt{f}} := \mathbf{t}_3(\mathbf{x})$ is a random view of fake samples, and $\mathbf{v}_{-i} := \mathbf{v} \setminus \{\mathbf{v}_i\}$. Remark that we use an independent projection header $h_{\mathtt{f}} : \mathbb{R}^{d_e} \to \mathbb{R}^{d_p}$ instead of $h_{\mathtt{r}}$ in (6) for this loss.

To sum up, ContraD learns its contrastive representation by minimizing the following loss:

$$L_{\mathtt{con}}(D, h_{\mathtt{r}}, h_{\mathtt{f}}) := L_{\mathtt{con}}^{+}(D, h_{\mathtt{r}}) + \lambda_{\mathtt{con}} L_{\mathtt{con}}^{-}(D, h_{\mathtt{f}}), \tag{9}$$

where $\lambda_{\mathtt{con}} > 0$ is a hyperparameter. Nevertheless, we simply use $\lambda_{\mathtt{con}} = 1$ in our experiments.

3.2 TRAINING GANS WITH CONTRAD

The contrastive training loss defined in (9) is jointly trained with a standard GAN loss (2) for training the generator network G. To obtain a (scalar) discriminator score needed to define the GAN loss, we simply use an additional *discriminator head* $h_{\mathtt{d}} : \mathbb{R}^{d_e} \to \mathbb{R}$ upon the contrastive representation of D. Here, a key difference of ContraD training from other GANs is that ContraD only optimizes the parameters of $h_{\mathtt{d}}$ for minimizing the GAN loss: in practical situations of using stochastic gradient descent for the training, this can be implemented by *stopping gradient* before feeding inputs to $h_{\mathtt{d}}$. Therefore, the discriminator loss of ContraD is defined as follows:

$$L_{\mathtt{dis}}(h_{\mathtt{d}}) := -\mathbb{E}_{\mathbf{r}_{\mathtt{r}}}[\log(\sigma(h_{\mathtt{d}}(\mathbf{r}_{\mathtt{r}})))] - \mathbb{E}_{\mathbf{r}_{\mathtt{f}}}[\log(1-\sigma(h_{\mathtt{d}}(\mathbf{r}_{\mathtt{f}})))], \tag{10}$$

where $\mathbf{r}_{\mathtt{r}} = \mathtt{sg}(D(\mathbf{v}_{\mathtt{r}}))$ and $\mathbf{r}_{\mathtt{f}} = \mathtt{sg}(D(\mathbf{v}_{\mathtt{f}}))$. Here, $\sigma(\cdot)$ denotes the sigmoid function, $\mathbf{v}_{\mathtt{r}}$ and $\mathbf{v}_{\mathtt{f}}$ are random views of real and fake samples augmented via $\mathcal{T}$, respectively, and $\mathtt{sg}(\cdot)$ is the stop-gradient operation. Combined with the contrastive training loss (9), the joint training loss L_D for a single discriminator update is the following:

$$L_D := L_{\mathtt{con}} + \lambda_{\mathtt{dis}} L_{\mathtt{dis}} = L^{+}_{\mathtt{con}} + \lambda_{\mathtt{con}} L^{-}_{\mathtt{con}} + \lambda_{\mathtt{dis}} L_{\mathtt{dis}}, \tag{11}$$

where $\lambda_{\mathtt{dis}} > 0$ is a hyperparameter. Again, as like $\lambda_{\mathtt{con}}$, we simply use $\lambda_{\mathtt{dis}} = 1$ in our experiments.

Finally, the loss for the generator G can be defined simply like other standard GANs using the discriminator score, except that we also augment the generated samples $G(\mathbf{z})$ to $\mathbf{v}_{\mathtt{f}}$, in a similar way to DiffAug (Zhao et al., 2020a) or ADA (Karras et al., 2020a): namely, in case of the non-saturating loss (Goodfellow et al., 2014), we have

$$L_G := -\mathbb{E}_{\mathbf{v}_{\mathtt{f}}}[\log(\sigma(h_{\mathtt{d}}(D(\mathbf{v}_{\mathtt{f}}))))]. \tag{12}$$

Algorithm 1 in Appendix A describe a concrete training procedure of GANs with ContraD.

3.3 SELF-CONDITIONAL SAMPLING WITH CONTRAD

Apart from the improved training of GANs, the learned contrastive representation of ContraD could offer some additional benefits in practical scenarios compared to the standard GANs. In this section, we present an example of how one could further utilize this additional information to derive a conditional generative model from an unconditionally-trained GAN with ContraD. More specifically, here we consider a variant of *discriminator-driven latent sampling* (DDLS) (Che et al., 2020) to incorporate a given representation vector from the ContraD encoder as a conditional information. Originally, DDLS attempts to improve the sample quality of a pre-trained GAN via a Langevin sampling (Welling & Teh, 2011) on the latent space of the following form:

$$\mathbf{z}_{t+1} := \mathbf{z}_t - \frac{\varepsilon}{2}\nabla_{\mathbf{z}_t} E(\mathbf{z}_t) + \sqrt{\varepsilon}\mathbf{n}_t, \mathbf{n}_t \sim \mathcal{N}(0,1), \tag{13}$$

where $E(\mathbf{z}) := -\log p(\mathbf{z}) - d(G(\mathbf{z}))$ and $d(\cdot)$ denotes the logit of D. In this manner, for our ContraD, we propose to use the following "conditional" version of energy function $E(\mathbf{z}, v)$ for an arbitrary vector $v \in \mathbb{R}^{d_e}$ in the ContraD representation space, called *conditional DDLS* (cDDLS):

$$E(\mathbf{z}, v) := -\log p(\mathbf{z}) - h_{\mathtt{d}}(D(G(\mathbf{z}))) - \lambda(v \cdot D(G(\mathbf{z}))). \tag{14}$$

In our experiments, for instance but not limited to, we show that using the learned weights obtained from linear evaluation as conditional vectors successfully recovers class-conditional generations from an unconditional ContraD model.

4 EXPERIMENTS

We verify the effectiveness of our ContraD in three different aspects: (a) its performance on image generation compared to other techniques for data augmentations in GANs, (b) the quality of representations learned from ContraD in terms of linear evaluation, and (c) its ability of self-conditional sampling via cDDLS. Overall, we constantly observe integrating contrastive learning into GANs and *vice versa* have positive effects to each other: (a) for image generation, ContraD enables a simple SNDCGAN to outperform a state-of-the-art architecture of StyleGAN2 on CIFAR-10/100; (b) in case of linear evaluation, on the other hand, we observe the representation from ContraD could outperform those learned from SimCLR. Finally, we perform an ablation study to further understand each of the components we propose. We provide the detailed specification on the experimental setups, *e.g.*, architectures, training configurations and hyperparameters in Appendix E.

Table 1: Comparison of the best FID score and IS on unconditional image generation of CIFAR-10 and CIFAR-100. Values in the rows marked by * are from those reported in its reference.

Architecture	Method	Augment.	CIFAR-10		CIFAR-100	
			FID ↓	IS ↑	FID ↓	IS ↑
G: SNDCGAN D: SNDCGAN	-	-	26.6	7.38	28.5	7.25
	CR (Zhang et al., 2020)	HFlip, Trans	19.5	7.87	22.2	7.91
	bCR (Zhao et al., 2020c)	HFlip, Trans	14.0	8.35	19.2	8.46
	DiffAug (Zhao et al., 2020a)	Trans, CutOut	22.9	7.64	27.0	7.47
	ContraD (ours)	**SimCLR**	**10.9**	**8.78**	**15.2**	**9.09**
G: SNDCGAN D: SNResNet-18	-	-	41.3	6.33	52.3	5.24
	CR (Zhang et al., 2020)	HFlip, Trans	32.1	7.08	36.5	6.55
	bCR (Zhao et al., 2020c)	HFlip, Trans	22.8	7.29	28.2	7.30
	DiffAug (Zhao et al., 2020a)	Trans, CutOut	59.5	5.62	58.7	5.39
	ContraD (ours)	**SimCLR**	**9.86**	**9.09**	**15.0**	**9.56**
G: StyleGAN2 D: StyleGAN2	-	-	11.1	9.18	16.5	9.51
	DiffAug* (Zhao et al., 2020a)	Trans, CutOut	9.89	9.40	15.2	**10.0**
	ContraD (ours)	**SimCLR**	**9.80**	**9.47**	**14.1**	**10.0**

Table 2: Comparison of the best FID score and IS on unconditional image generation of CelebA-HQ-128 with SNDCGAN. We report the mean and standard deviation of best scores across 3 trials.

CelebA-HQ	W/O	Hinge	CR	bCR	ContraD (ours)
FID ↓	$24.8_{\pm 1.69}$	$26.4_{\pm 1.32}$	$20.9_{\pm 0.37}$	$19.5_{\pm 0.08}$	$\mathbf{17.1}_{\pm 1.38}$
IS ↑	$2.06_{\pm 0.06}$	$2.05_{\pm 0.06}$	$2.11_{\pm 0.09}$	$2.21_{\pm 0.10}$	$\mathbf{2.44}_{\pm 0.06}$

We consider a variety of datasets including CIFAR-10/100 (Krizhevsky, 2009), CelebA-HQ-128 (Lee et al., 2020a), and ImageNet (Russakovsky et al., 2015) in our experiments, with three well-known GAN architectures: SNDCGAN (Miyato et al., 2018), StyleGAN2 (Karras et al., 2020b) and BigGAN (Brock et al., 2019).[1] We measure *Fréchet Inception distance* (FID) (Heusel et al., 2017) and *Inception score* (IS) (Salimans et al., 2016) as quantitative metrics to evaluate generation quality.[2] We compute FIDs between the *test set* and generated samples of the same size. We follow the best hyperparameter practices explored by Kurach et al. (2019) for SNDCGAN models on CIFAR and CelebA-HQ-128 datasets. For StyleGAN2, on the other hand, we follow the training details of Zhao et al. (2020a) in their CIFAR experiments. All the training details are shared across methods per experiment, but the choice of batch size for training ContraD: we generally observe that ContraD greatly benefits from using larger batch (Appendix D), similarly to SimCLR (Chen et al., 2020a), but in contrast to the common practice of GAN training. Therefore, in our experiments, we consider to use $2\times$ to $8\times$ larger batch sizes for training ContraD, where the details are specified in Appendix E.

4.1 Results

Image generation on CIFAR-10/100. CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) consist of 60K images of size 32×32 in 10 and 100 classes, respectively, 50K for training and 10K for testing. We start by comparing ContraD on CIFAR-10/100 with three recent methods that applies data augmentation in GANs: (a) Consistency Regularization (CR) (Zhang et al., 2020), (b) the "balanced" Consistency Regularization (bCR) (Zhao et al., 2020c), and (c) Differentiable Augmentation (DiffAug) (Zhao et al., 2020a) as baselines.[3] We follow their best pratices on which data augmentations to use: specifically, we use a combination of horizontal flipping (HFlip) and random translation up to 4 pixels (Trans) for CR and bCR, and Trans + CutOut (DeVries & Taylor, 2017) for DiffAug for CIFAR datasets. We mainly consider SNDCGAN and StyleGAN2 for training, but in addition, we further consider a scenario that the discriminator architecture of SNDCGAN is scaled up into *SNResNet-18*, which is identical to ResNet-18 (He et al., 2016) without batch normalization (Ioffe & Szegedy, 2015) but with spectral normalization (Miyato et al., 2018), in order to see that how each training method behaves when the capacity of generator and discriminator are significantly different.

[1] Results on BigGAN models for class-conditional generation of ImageNet can be found in Appendix B.

[2] We use the official InceptionV3 model in TensorFlow to compute both FID and IS in our experiments.

[3] Here, we notice that ADA (Karras et al., 2020a), a concurrent work to DiffAug, also offers a highly similar method to DiffAug, but with an additional heuristic of tuning augmentations on-the-fly with a validation dataset. We consider this addition is orthogonal to ours, and focus on comparing DiffAug for a clearer comparison.

Table 3: Comparison of classification accuracy under linear evaluation protocol on CIFAR-10 and CIFAR-100. We report the mean and standard deviation across 3 runs of the evaluation.

Dataset	Training	SNDCGAN	SNResNet-18	StyleGAN2
CIFAR-10	SimCLR	72.9$\pm$0.02	80.3$\pm$0.05	86.2$\pm$0.06
	ContraD (ours)	**77.5**$\pm$0.20	**85.7**$\pm$0.10	**88.6**$\pm$0.06
CIFAR-100	SimCLR	30.8$\pm$0.11	41.2$\pm$0.06	61.1$\pm$0.06
	ContraD (ours)	**37.4**$\pm$0.06	**51.1**$\pm$0.18	**68.1**$\pm$0.07

Table 1 shows the results: overall, we observe our ContraD uniformly improves the baseline GAN training by a large margin, significantly outperforming other baselines as well. Indeed, our results on SNDCGAN already achieve better FID than those from the baseline training of StyleGAN2. Moreover, when the discriminator of SNDCGAN is increased to SNResNet-18, ContraD could further improve its results, without even modifying any of the hyperparameters. It is particularly remarkable that all the other baselines could not improve their results in this setup: A further ablation study presented in Section 4.2 reveals that *stopping gradients* before the discriminator head in our design is an important factor to benefit from SNResNet-18. Finally, our ContraD can also be applied to a larger, sophisticated StyleGAN2, further improving both FID and IS compared to DiffAug.

Image generation on CelebA-HQ-128. We also perform experiments on CelebA-HQ-128 (Lee et al., 2020a), which consist of 30K human faces of size 128×128, to see that our design of ContraD still applies well for high-resolution datasets. We split 3K images out of CelebA-HQ-128 for testing, and use the remaining for training. We use SNDCGAN for this experiment, and compare FID and IS to (a) the default non-saturating loss (W/O) (Goodfellow et al., 2014), (b) the hinge loss (Hinge) (Lim & Ye, 2017; Tran et al., 2017), (c) CR and (d) bCR. Similarly to the CIFAR experiments, we use the "HFlip+Trans" augmentation for both CR and bCR, but for CelebA-HQ-128 we allow them to translate up to 16 pixels, following (Zhang et al., 2020). Again, as shown in Table 2, we confirm that ContraD could still improve training of GANs compared to other methods, even with this high-resolution, yet with limited samples and a straightforward choice of architecture.

Linear evaluation. Recall that the training objective of ContraD we propose in (11) can be simply reduced into the SimCLR loss (Chen et al., 2020a) without considering the fake samples, *i.e.*, by letting $\lambda_{\texttt{con}} = 0$. A natural question from here is whether those fake samples could help to improve the representation of SimCLR: we verify this by comparing the *linear evaluation* performance (Kolesnikov et al., 2019; Chen et al., 2020a; Grill et al., 2020) of the learned representation of ContraD to those from SimCLR: specifically, linear evaluation measures the test accuracy that a linear classifier on the top of given (frozen) representation could maximally achieve for a given (downstream) labeled data. Following this protocol, we compare the linear evaluation of a ContraD model with its ablation that $\lambda_{\texttt{con}}$ is set to 0 during its training. We consider two datasets and three discriminator architectures for this experiments: namely, we train SNDCGAN, SNResNet-18, and StyleGAN2 models on CIFAR-10 and CIFAR-100. In this experiment, we use batch size 256 for training all the SNDCGAN and SNResNet-18 models, while in case of StyleGAN2 we use batch size 64 instead. Table 3 summarizes the results, and somewhat interestingly, it shows that ContraD significantly improves the SimCLR counterpart in terms of linear evaluation for all the models tested: in other words, we observe that keeping discriminative features between real and fake samples in their representation, which is what ContraD does in addition to SimCLR, can be beneficial to improve the contrastive learning itself. Again, our observation supports a great potential of ContraD not only in the context of GANs, but also in the contrastive learning.

Conditional DDLS with ContraD. We further evaluate the performance of conditional DDLS (cDDLS) (14) proposed for ContraD, with a task of deriving class-conditional generative models for a given (unconditionally-trained) GAN model. Here, we measure the performance of this conditional generation by computing the *class-wise* FIDs to the test set, *i.e.*, FIDs computed for each subsets of the same class in the test set. We test cDDLS on SNDCGAN and SNResNet-18 ContraD models trained on CIFAR-10. To perform a conditional generation for a specific class, we simply take a learned linear weight obtained from the linear evaluation protocol for that class. The results summarized in Table 4 show that applying cDDLS upon unconditional generation significantly improves class-wise FIDs for all the classes in CIFAR-10. Some of the actual samples conditionally generated from an SNResNet-18 ContraD model can be found in Figure 5 of Appendix C.

Table 4: Comparison of FID (lower is better) from the class-wise subsets of CIFAR-10 test set (1K images per class). "Random" indicates unconditional generation, and "Train-1K" indicates a 1K random subsamples from the CIFAR-10 train set of the same class.

D Arch.	Sampling	Plane	Car	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck	Mean
SNDCGAN	Random	110.6	125.9	103.8	104.4	107.0	121.0	134.1	120.1	128.8	130.0	118.6
	+ cDDLS	**65.3**	**72.1**	**63.9**	**69.4**	**57.5**	**69.0**	**107.0**	**61.5**	**58.2**	**53.6**	**67.8**
SNResNet-18	Random	117.6	136.3	100.5	99.6	98.6	115.0	128.2	111.7	140.9	140.3	118.9
	+ cDDLS	**67.1**	**65.8**	**59.8**	**63.8**	**59.7**	**61.9**	**68.8**	**59.9**	**58.9**	**51.6**	**61.7**
-	Train-1K	38.9	28.4	41.4	49.7	36.4	40.2	41.7	33.6	30.7	24.6	36.6

Table 5: Comparison of the best FID score and IS of bCR (Zhao et al., 2020c) and DiffAug (Zhao et al., 2020a) on CIFAR-10 with SNDCGAN under the SimCLR augmentation.

Method	Augment.	FID ↓	IS ↑
bCR	HFlip, Trans	14.0	8.35
+ Aug.	SimCLR	20.6	7.44
DiffAug	Trans, CutOut	22.9	7.64
+ Aug.	SimCLR	21.9	7.42
ContraD	**SimCLR**	**10.9**	**8.78**

Table 6: Comparison of the best FID score and IS on CIFAR-10 for ablations of our proposed components. All the models are trained with batch size 256. For the ablation of "MLP $h_{\mathtt{d}}$", we replace $h_{\mathtt{d}}$ with a linear model. "$\mathtt{sg}(\cdot)$" indicates the use of stop-gradient operation.

				D: SNDCGAN		D: SNResNet-18	
MLP $h_{\mathtt{d}}$	$\mathtt{sg}(\cdot)$	$L^{+}_{\mathtt{con}}$	$L^{-}_{\mathtt{con}}$	FID ↓	IS ↑	FID ↓	IS ↑
✓	✓	✓	✓	**11.1**	**8.62**	**10.6**	**8.99**
✗	✓	✓	✓	185	3.43	274	2.09
✓	✗	✓	✓	11.6	8.61	28.0	7.52
✓	✓	✗	✓	11.9	8.45	182	2.02
✓	✓	✓	✗	210	1.93	232	2.56

4.2 Ablation Study

We conduct an ablation study for a more detailed analysis on the proposed method. For this section, unless otherwise specified, we perform experiments on CIFAR-10 with SNDCGAN architecture.

Stronger augmentations for other methods. One of the key characteristics of ContraD compared to the prior works is the use of stronger data augmentations for training GANs, *e.g.*, we use the augmentation pipeline of SimCLR (Chen et al., 2020a) in our experiments. In Table 5, we examine the performance bCR (Zhao et al., 2020c) and DiffAug (Zhao et al., 2020a) when this stronger augmentation is applied, and shows that the SimCLR augmentation could not meaningfully improve FID or IS for both methods. Unlike these methods, our ContraD effectively handles the SimCLR augmentations without overly regularizing the discriminator, possibly leveraging many components of contrastive learning: *e.g.*, the normalized loss, or the use of separate projection heads.

MLP discriminator head. In our experiments, we use a 2-layer network for the discriminator head $h_{\mathtt{d}}$. As shown in Table 6, this choice can be minimal: replacing $h_{\mathtt{d}}$ with a linear model severely breaks the training. Conversely, although not presented here, we have also observed that MLPs of more than two layers neither give significant gains on the performance in our experimental setup.

Stopping gradients. We use the stop-gradient operation $\mathtt{sg}(\cdot)$ before the discriminator head $h_{\mathtt{d}}$ in attempt to decouple the effect of GAN loss to the ContraD representation. Table 6 also reports the ablation of this component: overall, not using $\mathtt{sg}(\cdot)$ does not completely break the training, but it does degrade the performance especially when a deeper discriminator is used, *e.g.*, SNResNet-18.

Contrastive losses. Recall that the representation of ContraD is trained by two losses, namely $L^{+}_{\mathtt{con}}$ and $L^{-}_{\mathtt{con}}$ (Section 3.1). From an ablation study on each loss, as presented in Table 6, we observe that both of the losses are indispensable for our training to work in a larger discriminator architecture.

5 Conclusion

In this paper, we explore a novel combination of GAN and contrastive learning, two important, yet seemingly different paradigms in unsupervised representation learning. They both have been independently observed the crucial importance of maintaining consistency to data augmentations in their representations (Zhang et al., 2020; Tian et al., 2020), and here we further observe that these two representations greatly complement each other when combined upon the shared principle of *view-invariant* representation. We believe our method could open up a new direction of research both in GAN and contrastive learning.

REFERENCES

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/arjovsky17a.html.

Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems 32*, pp. 15535–15545. Curran Associates, Inc., 2019.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*, 2020.

Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12154–12163, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020a.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Terrance DeVries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout, 2017.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1422–1430, 2015.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=S1v4N2l0-.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27*, pp. 2672–2680. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein GANs. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5767–5777. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems 30*, pp. 6626–6637. Curran Associates, Inc., 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 4565–4573. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6391-generative-adversarial-imitation-learning.pdf.

Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding, 2019.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pp. 448–456, 2015.

Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1erHoR5t7.

Minguk Kang and Jaesik Park. Contrastive generative adversarial networks, 2020.

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

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data, 2020a.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8110–8119, 2020b.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1920–1929, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Department of Computer Science, University of Toronto, 2009.

Karol Kurach, Mario Lučić, Xiaohua Zhai, Marcin Michalski, and Sylvain Gelly. A large-scale study on regularization and normalization in GANs. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3581–3590, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/kurach19a.html.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. MaskGAN: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020a.

Kwot Sin Lee, Ngoc-Trung Tran, and Ngai-Man Cheung. Infomax-GAN: Improved adversarial image generation via information maximization and contrastive learning, 2020b.

Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.

R. Linsker. Self-organization in a perceptual network. *Computer*, 21(3):105–117, 1988.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are GANs created equal? a large-scale study. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 700–709. Curran Associates, Inc., 2018. URL http://papers.nips.cc/paper/7350-are-gans-created-equal-a-large-scale-study.pdf.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? *arXiv preprint arXiv:1801.04406*, 2018.

Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ByS1VpgRZ.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=B1QRgziT-.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pp. 8026–8037, 2019.

Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. SinGAN: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved techniques for training GANs. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 29*, pp. 2234–2242. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

Edgar Schonfeld, Bernt Schiele, and Anna Khoreva. A U-Net based discriminator for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8207–8216, 2020.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

Dustin Tran, Rajesh Ranganath, and David M. Blei. Hierarchical implicit models and likelihood-free variational inference, 2017.

Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, and Ngai-Man Cheung. An improved self-supervised GAN via adversarial training, 2019.

Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for GAN training, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

Xiang Wei, Zixia Liu, Liqiang Wang, and Boqing Gong. Improving the improved training of wasserstein GANs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJx9GQb0-.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lxKlSKPH.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. *arXiv preprint arXiv:2006.10738*, 2020a.

Yang Zhao, Chunyuan Li, Ping Yu, Jianfeng Gao, and Changyou Chen. Feature quantization improves GAN training. In *Proceedings of the 37th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, 2020b.

Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans, 2020c.

Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training, 2020d.

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

A TRAINING PROCEDURE OF CONTRAD

Algorithm 1 GANs with Contrastive Discriminator (ContraD)

Require: generator G, discriminator D, Adam hyperparameters α, β_1, β_2, number of D updates per G update N_D, family of data augmentations $\mathcal{T}$, $\lambda > 0$.

for # training iterations **do**
  for $t = 1$ **to** N_D **do**
    Sample $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$ and $\mathbf{z} \sim p(\mathbf{z})$
    Sample $\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_3 \sim \mathcal{T}$
    $\mathbf{v}_\mathtt{r}^{(1)}, \mathbf{v}_\mathtt{r}^{(2)}, \mathbf{v}_\mathtt{f} \leftarrow \mathbf{t}_1(\mathbf{x}), \mathbf{t}_2(\mathbf{x}), \mathbf{t}_3(G(\mathbf{z}))$
    $L_\mathtt{con}^+, L_\mathtt{con}^- \leftarrow L_\mathtt{SimCLR}(\mathbf{v}_\mathtt{r}^{(1)}, \mathbf{v}_\mathtt{r}^{(2)}; D, h_\mathtt{r}), L_\mathtt{SupCon}(\mathbf{v}_\mathtt{f}, [\mathbf{v}_{\mathtt{f},-i}, \mathbf{v}_\mathtt{r}^{(1)}, \mathbf{v}_\mathtt{r}^{(2)}]_i, [\mathbf{v}_{\mathtt{f},-i}]_i; D, h_\mathtt{f})$
    $\mathbf{r}_\mathtt{r}, \mathbf{r}_\mathtt{f} \leftarrow \mathtt{sg}(D(\mathbf{v}_\mathtt{r}^{(2)})), \mathtt{sg}(D(\mathbf{v}_\mathtt{f}))$
    $L_\mathtt{dis} \leftarrow -\mathbb{E}_{\mathbf{r}_\mathtt{r}}[\log(\sigma(h_\mathtt{d}(\mathbf{r}_\mathtt{r})))] - \mathbb{E}_{\mathbf{r}_\mathtt{f}}[\log(1 - \sigma(h_\mathtt{d}(\mathbf{r}_\mathtt{f})))]$
    $L_D \leftarrow L_\mathtt{con}^+ + \lambda_\mathtt{con} L_\mathtt{con}^- + \lambda_\mathtt{dis} L_\mathtt{dis}$
    $D, h_\mathtt{r}, h_\mathtt{f}, h_\mathtt{d} \leftarrow \mathrm{Adam}([D, h_\mathtt{r}, h_\mathtt{f}, h_\mathtt{d}], L_D; \alpha, \beta_1, \beta_2)$
  end for
  $L_G \leftarrow -\mathbb{E}_{\mathbf{v}_\mathtt{f}}[\log(\sigma(h_\mathtt{d}(D(\mathbf{v}_\mathtt{f}))))]$
  $G \leftarrow \mathrm{Adam}(G, L_G; \alpha, \beta_1, \beta_2)$
end for

B RESULTS ON IMAGENET DATASET

Table 7: Comparison of the best FID score and IS on conditional generation of ImageNet (64×64) with BigGAN architecture. Values in the rows marked by * are from those reported in its reference.

ImageNet	FID ↓	IS ↑
BigGAN* (Brock et al., 2019)	12.75	21.84±0.34
FQ-BigGAN* (Zhao et al., 2020b)	12.62	21.99±0.32
BigGAN-ContraD (ours)	**12.12**	**23.52**±0.23

We also apply ContraD to the state-of-the-art BigGAN (Brock et al., 2019) architecture on ImageNet (Russakovsky et al., 2015), to examine the effectiveness of our method on this large-scale, class-conditional GAN model. For this experiment, we exactly follow the training setups done in FQ-BigGAN (Zhao et al., 2020b), which is based on the official PyToch (Paszke et al., 2019) implementation of BigGAN.[4] Following Zhao et al. (2020b), we down-scale the ImageNet dataset into 64×64, and reduce the default channel width of 96 into 64. We train the model for 120K generator steps. In case of ImageNet, we compute FIDs between the training set and 50K of generated samples for a fair comparison to the baseline scores reported by Zhao et al. (2020b). In Table 7, we show that ContraD further improves the baseline BigGAN training and FQ-BigGAN (Zhao et al., 2020b) both in FID and IS under the same training setups.

[4] https://github.com/ajbrock/BigGAN-PyTorch

C QUALITATIVE RESULTS

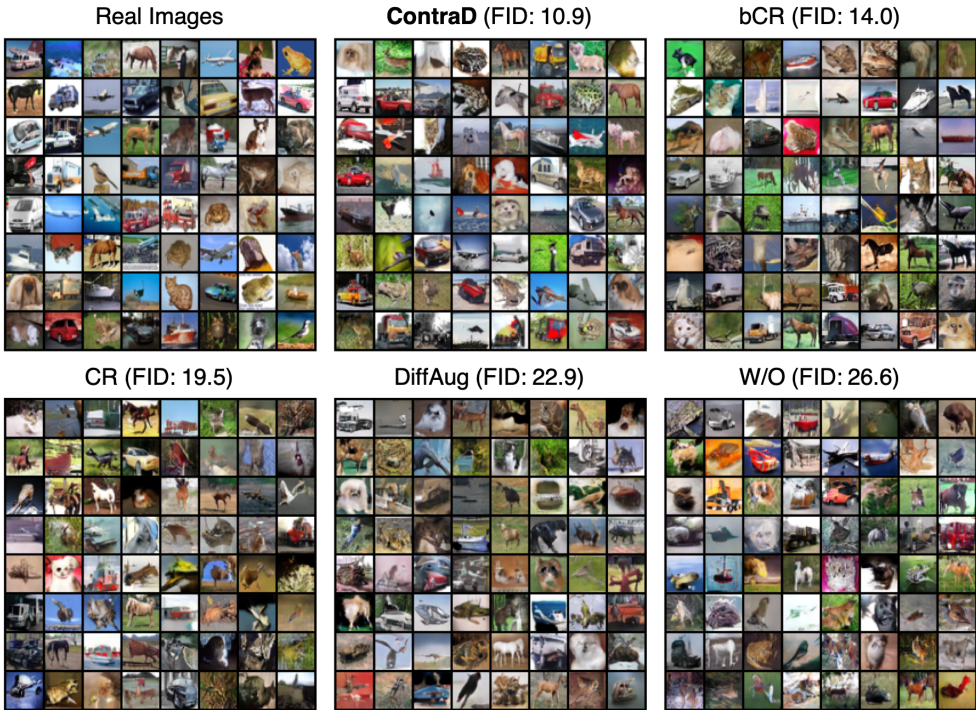

Figure 2: Qualitative comparison of unconditionally generated samples from GANs with different training methods. All the models are trained on CIFAR-10 with an SNDCGAN architecture.

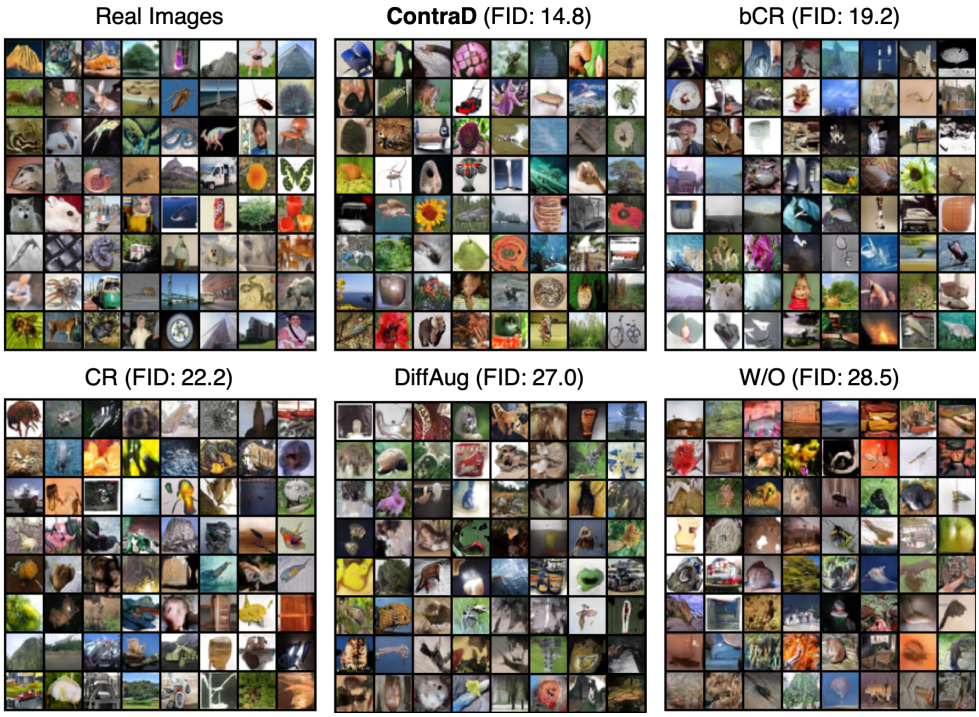

Figure 3: Qualitative comparison of unconditionally generated samples from GANs with different training methods. All the models are trained on CIFAR-100 with an SNDCGAN architecture.

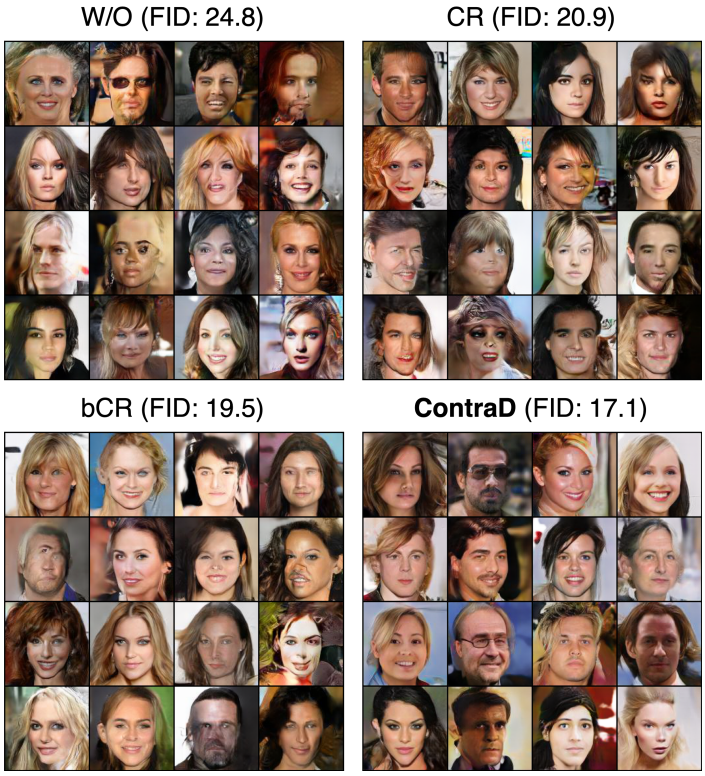

Figure 4: Qualitative comparison of unconditionally generated samples from GANs with different training methods. All the models are trained on CelebA-HQ-128 with an SNDCGAN architecture.

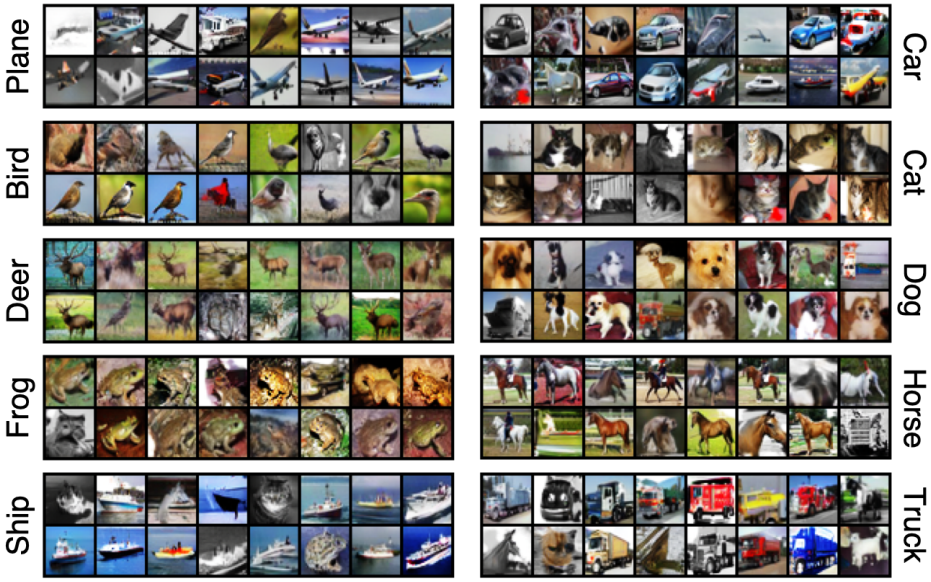

Figure 5: Visualization of conditionally generated samples via conditional DDLS (Section 3.3) with ContraD. We use an SNDCGAN generator trained with an SNResNet-18 ContraD for the generation.

D EFFECT OF USING LARGER BATCH SIZES

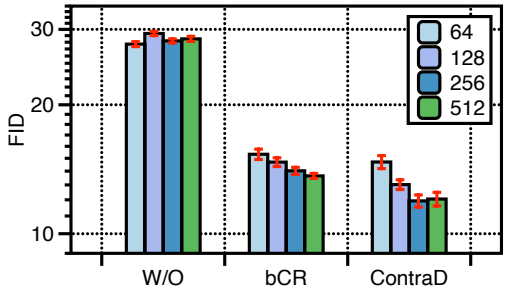

Figure 6: Comparion of the FID distribution of the top 25% of trained models (Kurach et al., 2019) on CIFAR-10 (SNDCGAN) for different batch sizes.

Contrary to existing practices for training GANs, we observe in our experiments that ContraD typically needs a larger batch size in training to perform best. We further confirm this effect of batch sizes in Figure 6: overall, we observe the stanard training ("W/O") is not significantly affected by a particular batch size, while ContraD offers much better FIDs on larger batch sizes. We also observe bCR (Zhao et al., 2020c) slightly benefits from larger batches, but not as much as ContraD does. This observation is consistent with Chen et al. (2020a) that contrastive learning benefits from using larger batch sizes, and it also supports our key hypothesis on the close relationship between contrastive and discriminator representations.

E EXPERIMENTAL DETAILS

E.1 ARCHITECTURE

Overall, we consider four architectures in our experiments: SNDCGAN (Miyato et al., 2018), StyleGAN-2 (Karras et al., 2020b), and BigGAN (Brock et al., 2019) for GAN architectures, and additionally SNResNet-18 for a discriminator architecture only. For all the architectures considered, we have modified the last linear layer into a *multi-layer perceptron* (MLP) having the same input and output size, for a fair comparison to ContraD that requires such an MLP for the discriminator header $h_{\mathtt{d}}$. When a GAN model is trained via ContraD, we additionally introduce two 2-layer MLP projection heads of the output size 128 upon the penutimate representation in the discriminator to stand $h_{\mathtt{r}}$ and $h_{\mathtt{f}}$. All the models are implemented in PyTorch (Paszke et al., 2019) framework.

SNDCGAN. We adopt a modified version of SNDCGAN architecture by Kurach et al. (2019) following the experimental setups of other baselines (Zhang et al., 2020; Zhao et al., 2020c). The official implementation can be found in `compare_gan` codebase,[5] and we have re-implemented this TensorFlow implementation into PyTorch framework. The detailed structures of the generator and discriminator of SNDCGAN are summarized in Table 8 and 9, respectively.

SNResNet-18. We modify the ResNet-18 (He et al., 2016) architecuture, where a PyTorch implementation is available from `torchvision`[6] library, to not include batch normalization (Ioffe & Szegedy, 2015) layers inside the network. Instead, we apply spectral normalization (Miyato et al., 2018) for all the convolutional and linear layers. We made such a modification to adapt this model as a stable GAN discriminator, as batch normalization layers often harm the GAN dynamics in discriminator in practice.

StyleGAN2. We follow the StyleGAN2 architecture used in the DiffAug baseline (Zhao et al., 2020a). Specifically, the architecture generally follows the official StyleGAN2 implementation, but the number of channels at 32×32 resolution is reduced to 128 and doubled at each coarser level with a maximum of 512 channels, in order to optimize the model for CIFAR datasets, as suggested by Zhao et al. (2020a).

[5] https://www.github.com/google/compare_gan
[6] https://github.com/pytorch/vision/blob/master/torchvision

Table 8: SNDCGAN generator

Layer	Kernel	Output
Latent $\mathbf{z}$	-	128
Linear, BN, ReLU	-	$h/8 \times w/8 \times 512$
DeConv, BN, ReLU	$[4, 4, 2]$	$h/4 \times w/4 \times 256$
DeConv, BN, ReLU	$[4, 4, 2]$	$h/2 \times w/2 \times 128$
DeConv, BN, ReLU	$[4, 4, 2]$	$h \times w \times 64$
DeConv, Tanh	$[3, 3, 1]$	$h \times w \times 3$

Table 9: SNDCGAN discriminator

Layer	Kernel	Output
Conv, LeakyReLU	$[3, 3, 1]$	$h \times w \times 64$
Conv, LeakyReLU	$[4, 4, 2]$	$h/2 \times w/2 \times 128$
Conv, LeakyReLU	$[3, 3, 1]$	$h/2 \times w/2 \times 128$
Conv, LeakyReLU	$[4, 4, 2]$	$h/4 \times w/4 \times 256$
Conv, LeakyReLU	$[3, 3, 1]$	$h/4 \times w/4 \times 256$
Conv, LeakyReLU	$[4, 4, 2]$	$h/8 \times w/8 \times 512$
Conv, LeakyReLU	$[3, 3, 1]$	$h/8 \times w/8 \times 512$

BigGAN. We use the official PyToch implementation of BigGAN[7] in our experiments. In order to adapt ContraD into the class-conditional BigGAN architecture, we apply the stop-gradient operation not only on the penultimate representation before computing $h_{\mathtt{d}}$, but also before applying the class-conditional projection (Miyato & Koyama, 2018).

E.2 Training details

SNDCGAN. Overall, we follow the best hyperparameter practices explored by Kurach et al. (2019) for SNDCGAN models on CIFAR and CelebA-HQ-128 datasets: We use Adam (Kingma & Ba, 2014) with $(\alpha, \beta_1, \beta_2) = (0.0002, 0.5, 0.999)$ for optimization with batch size of 64, while in case of ContraD on CIFAR datasets the batch size of 512 is used instead. For CelebA-HQ-128, we keep the default batch size of 64 even for ContraD. All models but the "Hinge" baselines in Table 2 are trained with the non-saturating loss (Goodfellow et al., 2014). We stop training after 200K generator updates for CIFAR-10/100 and 100K for CelebA-HQ-128. When CR or bCR is used, we set their regularization strengths $\lambda = 10$ for both real and fake images.

StyleGAN2. We follow the training details of DiffAug (Zhao et al., 2020a) in their CIFAR experiments: we use Adam with $(\alpha, \beta_1, \beta_2) = (0.002, 0.0, 0.99)$ for optimization with batch size of 32, but 64 for ContraD models. We use non-saturating loss for training, and use R_1 regularization (Mescheder et al., 2018) with $\gamma = 0.1$. We do not use, however, the path length regularization and the lazy regularization (Karras et al., 2020b) in training. We take exponential moving average on the generator weights with half-life of 10^6. We stop training after 800K generator updates. When CR or bCR is used, we set their regularization strengths $\lambda = 10$ for both real and fake images.

Linear evaluation. For a single linear evaluation run, we train a linear classifier upon a given (frozen) representation via stochastic gradient descent for 100 epochs. The initial learning rate of 0.1, and it is decayed by 0.1 at 60, 75, and 90-th epoch. For linear evaluation, we augment training data with random crop, resizing and horizontal flipping.

Conditional DDLS. We run Langevin sampling of step size $\varepsilon = 0.1$ for 1,000 iterations. We follow all the practical considerations proposed by Che et al. (2020) to improve the sample quality of DDLS: specifically, (a) we separately set the standard deviation of the Gaussian noise (13) to 0.1, and (b) to alleviate mode dropping issues, instead of running DDLS on G itself, we run the sampling with $G^*(\mathbf{z}, \mathbf{z}') := G(\mathbf{z}) + \varepsilon \mathbf{z}'$, where $\mathbf{z}'$ is also jointly updated via Langevin dynamics.

Hyperparameters in ContraD. Unless otherwise specified, we use $\lambda_{\mathtt{con}} = \lambda_{\mathtt{dis}} = 1$, and $\tau = 0.1$ in our experiments, where τ is the temperature hyperparameter defined in (5). We use hidden layer of size 512 for $h_{\mathtt{r}}$, $h_{\mathtt{f}}$, $h_{\mathtt{d}}$ in SNDCGAN and StyleGAN2, and 1024 in SNResNet-18 and BigGAN. For ContraD models, we use linear warm-up strategy on learning rate up to 3K steps of training for both generator and discriminator, following practices in contrastive learning (Chen et al., 2020a).

SimCLR augmentations. We use the augmentation pipeline proposed in SimCLR (Chen et al., 2020a) for training ContraD. Specifically, for CIFAR datasets, we sequentially transform a given image by: (a) random crop and resizing, (b) horizontal flipping, (c) color jittering with probability of $p = 0.8$, and (d) graying the image with probability of $p = 0.2$. We additionally apply (f) Gaussian blurring with probability of $p = 0.5$ for higher resolution images such as CelebA-HQ-128 and ImageNet. We ensure that every transformation in this pipelines are differentiably implemented, so that the augmentation still can be used for training the generator of GAN.

[7] https://github.com/ajbrock/BigGAN-PyTorch