Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training

Anonymous Author(s)
Affiliation
Address
email

Abstract

Conditional Generative Adversarial Networks (cGAN) can generate realistic images by incorporating class information into GAN. One of the most popular cGANs is an auxiliary classifier GAN with softmax cross-entropy loss (ACGAN). However, it is widely known that training ACGAN is challenging as the number of classes in the dataset increases. Also, ACGAN tends to generate easily classifiable samples without considering diversity. In this paper, we introduce two cures for ACGAN. First, we identify that gradient exploding in the classifier can cause an undesirable early training collapse, and projecting input vectors onto a unit hypersphere can resolve the problem. Second, we propose Data-to-Data Cross-Entropy loss (D2D-CE) to exploit fruitful relational information in the class-labeled dataset. On this foundation, we propose the Rebooted Auxiliary Classifier Generative Adversarial Network (ReACGAN). The experimental results show that ReACGAN achieves state-of-the-art generation performances on CIFAR10, CUB200, Tiny-Imagenet, and Imagenet datasets. In addition, we show that ReACGAN can benefit from consistency regularization and differentiable augmentations for better performance. All weights, logs, and source codes will be released to the public.

1 Introduction

Generative Adversarial Networks (GAN) [1] are known for the forefront approach for generating high-fidelity images of diverse categories [2, 3, 4, 5, 6, 7, 8, 9]. Behind the sensational generation ability of GANs, there has been tremendous effort to develop adversarial objectives free from vanishing gradient problem [10, 11, 12], regularizations for stabilizing adversarial training [11, 13, 14, 3, 15, 16, 17], and conditioning techniques to support the adversarial training using category information of dataset [18, 19, 20, 21, 22, 23, 24, 25]. Subsequently, the conditioning techniques have become the *de facto* standard for high-quality image generation. The models with the conditioning methods are called Conditional Generative Adversarial Networks (cGAN), and cGANs can be divided into two groups depending on the discriminator's conditioning way: classifier-based GANs [18, 20, 22, 23, 24, 25] and projection-based GANs [3, 4, 19].

The classifier-based GANs facilitate an auxiliary classifier to generate class-specific images by penalizing the generator if the synthesized outputs are not consistent with the conditioned labels. ACGAN [18] has been one of the widely used classifier-based GANs for their simple design and satisfactory generation performance. While ACGAN can exploit class information by pushing and pulling classifier's weights (proxy) against image embeddings [22, 24], it is well known that ACGAN training is prone to collapse at the early stage of training as the number of classes increases [19, 22, 24]. Also, the generator of ACGAN tends to generate easily classifiable images at the cost of reduced diversity [18, 19]. Projection-based GANs, on the other hand, have shown cutting-edge generation results on datasets with a large number of categories. SNGAN [3], SAGAN [26], and BigGAN [4]

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

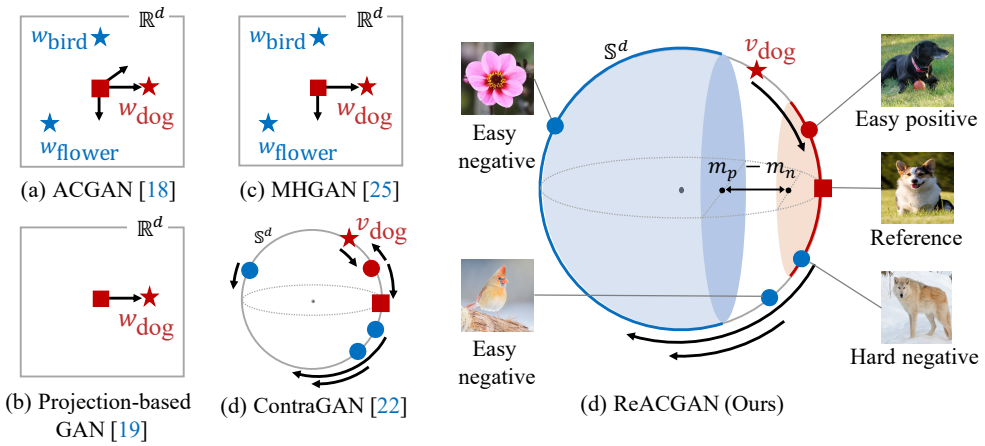


Figure 1: Schematics that depict how cGANs perform class conditioning. The red color means positive samples/proxy, and the blue color indicates negative samples. Arrows represent push-pull forces based on the reference sample. The length of an arrow indicates the magnitude of force. While Projection-based GAN only considers a pairwise relationship between an image and the target proxy, ReACGAN can explicitly consider data-to-data and data-to-class relationships and implicitly exploit class-to-class relationships for high-quality well-conditioned image generation.

are representative GANs in this category and can generate realistic images on CIFAR10 [27] and Imagenet [28] datasets. However, projection-based GANs only consider a pairwise relationship between an image and its label proxy (data-to-class relationships). As the result, the projection-based GANs can miss an additional opportunity to consider relation information between data instances (data-to-data relationships) and between instance categories (class-to-class relationships) as discovered by [22, 24].

In this paper, we analyze why ACGAN training becomes unstable as the number of classes increases and then propose remedies for (1) the instability and (2) the relatively poor generation performance of ACGAN compared to the projection-based models. First, we begin by analytically calculating gradients of the softmax cross-entropy loss used in ACGAN. By examining the exact values of analytic gradients, we discover that the unboundedness of input feature vectors and poor classification performance in the early training phase can cause an undesirable gradient exploding problem. Second, with alleviating the instability, we propose the Rebooted Auxiliary Classifier Generative Adversarial Networks (ReACGAN) using the Data-to-data Cross-Entropy loss (D2D-CE). ReACGAN projects image embeddings and proxies onto a unit hypersphere and computes similarities for data-to-data and data-to-class consideration. Additionally, we introduce two threshold values for inter-class and intra-class separability. In this way, ReACGAN overcomes the training instability and can exploit fruitful supervisory signals by explicitly considering data-to-class and data-to-data relationships, and also by implicitly looking at class-to-class relationships in the same mini-batch.

To validate our approach, we conduct image generation experiments using four datasets: CIFAR10 [27], CUB200 [29], Tiny-Imagenet [30], and Imagenet [28] using various evaluation metrics: Inception Score (IS) [31], Frechet Inception Distance (FID) [32], and Precision and Recall [33]. Through extensive experiments, we demonstrate that ReACGAN *beats both classifier- and projection-based cGANs and achieves state-of-the-art results* by 2.5%, 15.8%, 5.1%, and 20.2% in terms of FID on the four datasets, respectively. In addition, we confirm that ReACGAN is well harmonized with consistency regularizations [16] and differentiable augmentations [9]. In the ablation study, we verify the effectiveness of each part of ReACGAN.

2 Background: Generative Adversarial Networks

Generative Adversarial Network (GAN) [1] is an implicit generative model that aims to generate a sample indistinguishable from the real. GAN consists of two networks: a *Generator* $G : \mathcal{Z} \longrightarrow \mathcal{X}$ that tries to map a latent variable $\mathbf{z} \sim p(\mathbf{z})$ into the real data space $\mathcal{X}$ and a *Discriminator* $D : \mathcal{X} \longrightarrow$

$[0, 1]$ that strives to discriminate whether a given sample $\mathbf{x}$ is from the real data distribution $p_{\text{real}}(\mathbf{x})$ or the implicit distribution $p_{\text{gen}}(G(\mathbf{z}))$ derived from the generator $G(\mathbf{z})$. The objective of a vanilla GAN [1] can be expressed as follows:

$$\min_{G} \max_{D} \; \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]. \tag{1}$$

While GANs have shown impressive outputs in the image generation task [2, 34, 11, 13], training GANs often ends up encountering a mode-collapse problem [35, 11, 12]. As one of the prescriptions for stabilizing and reinforcing GANs, training GANs with categorical information, named Conditional Generative Adversarial Networks (cGAN), is suggested [36, 18, 19]. Depending on the presence of explicit classification losses, cGAN can be divided into two groups: Classifier-based GANs [18, 20, 25, 22, 23, 24] and Projection-based GANs [19, 3, 4]. ACGAN is one of the widely used classifier-based GANs with softmax cross-entropy loss. ACGAN utilizes an auxiliary classifier and is trained to perform adversarial training and image classification jointly. Although ACGAN has shown satisfactory generation results, training ACGAN becomes unstable as the number of classes in the training dataset increases [19, 22, 24]. Besides, ACGAN tends to generate easily classifiable images at the cost of limited diversity. To alleviate those problems, Zhou *et al.* [37] propose performing adversarial training on the classifier. Gong *et al.* [20] deploy one more classifier to train a missing entropy objective in ACGAN. However, ACGAN training still suffers from the early training collapse and reduced diversity problems when training ACGAN models on datasets with the large number of class categories, such as Tiny-Imagenet [30], and Imagenet [28].

In this situation, Miyato *et al.* [19] propose a projection discriminator for cGAN and show significant improvement in generating Imagenet dataset with a single pair of the generator and discriminator. Motivated by the promising generation results, many projection-based models [3, 26, 4, 16, 6, 17] are proposed and become the standard for conditional generation. In this paper, we revisit ACGAN and try to unveil the source of instability. Coping with the instability, we propose the Rebooted Auxiliary Classifier GANs for high-quality and diverse image generation with stable training.

3 Rebooting Auxiliary Classifier GANs

3.1 Feature Normalization

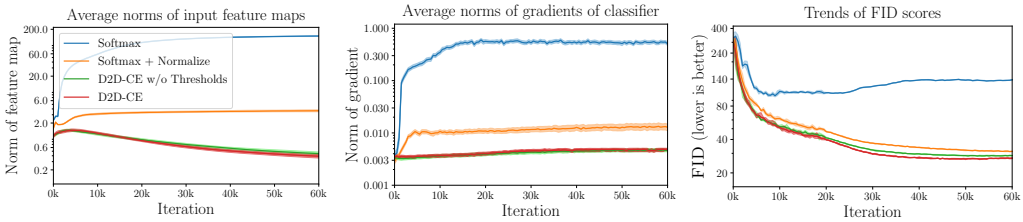
To uncover a nuisance that causes the instability of ACGAN, we start by analytically calculating gradients of weight vectors in the softmax classifier. Let the part of the discriminator before the fully connected layer be a *Feature extractor* $F : \mathcal{X} \longrightarrow \mathcal{F} \in \mathbb{R}^d$ and let *Classifier* $C : \mathcal{F} \longrightarrow \mathbb{R}^c$ be a single fully connected layer parameterized by $\boldsymbol{W} = [\boldsymbol{w}_1, ..., \boldsymbol{w}_c] \in \mathbb{R}^{d\times c}$, where c denotes the number of classess. We can sample training images $\boldsymbol{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$ and integer labels $\boldsymbol{y} = \{y_1, ..., y_N\}$ from the joint distribution $p(\mathbf{x}, \mathbf{y})$. Using above notations, we can express the empirical cross-entropy loss used in ACGAN [18] as follows:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N} \log\Big(\frac{\exp\left(F(\boldsymbol{x}_i)^\top \boldsymbol{w}_{y_i}\right)}{\sum_{j=1}^{c}\exp\left(F(\boldsymbol{x}_i)^\top \boldsymbol{w}_{y_j}\right)}\Big). \tag{3}$$

Based on this, we can calculate the derivative of the cross-entropy loss, w.r.t $\boldsymbol{w}_{k\in\{1,...,c\}}$ as follows:

$$\frac{\partial \mathcal{L}_{\text{CE}}}{\partial \boldsymbol{w}_k} = -\frac{1}{N}\sum_{i=1}^{N}\Big\{F(\boldsymbol{x}_i)\Big(\mathbf{1}_{y_i=k} - p_{i,k}\Big)\Big\}, \tag{4}$$

where $\mathbf{1}_{y_i=k}$ is an indicator function that will output 1 when $y_i = k$ is satisfied, and $p_{i,k}$ is a class probability that represents the probability of i-th sample belongs to class k, mathematically $\frac{\exp\left(F(\boldsymbol{x}_i)^\top \boldsymbol{w}_k\right)}{\sum_{j=1}^{c}\exp\left(F(\boldsymbol{x}_i)^\top \boldsymbol{w}_j\right)}$. The above equation implies that the norm of gradient of the softmax cross-entropy loss is closely related to the norms and directions of each input feature map $F(\boldsymbol{x}_i)$ and the class probability $p_{i,k}$. In the early training stage, the classifier is prone to make incorrect predictions, resulting in low probability outputs. This phenomenon gets severe as the number of categories of the dataset increases. As a result, the norm of the gradient $|\frac{\partial \mathcal{L}_{\text{CE}}}{\partial \boldsymbol{w}_k}|$ begins to explode as the inner-product value of $F(\boldsymbol{x}_i)^\top \boldsymbol{w}_k$ arises, and the vector $F(\boldsymbol{x}_i)$ stretches out to $\boldsymbol{w}_k$ direction. This may break the balance between adversarial and classification training and can lead to early training collapse. In



(a) Feature norm (b) Gradient norm (c) FID score

Figure 2: Merits of integrating feature normalization and data-to-data relationship consideration. (a) average norms of input feature maps, (b) average norms of gradients of classification losses, and (c) trends of FID scores. Compared to ACGAN [18], the proposed ReACGAN does not experience early training collapse caused by excessively large feature and gradient norms. In addition, ReACGAN can converge to better equilibrium by considering data-to-data relationships with easy positive and negative suppression using hard thresholds.

Fig. 2a, we experimentally demonstrate that the average norm of input feature maps increases as the training progresses. As can be seen in Fig. 2b, the average norm of the gradient is on the rise sharply, and ACGAN training goes to the collapse (Fig. 2c).

As one of the remedies for the gradient exploding problem, we find that simply *normalizing the feature embeddings onto a unit hypersphere* $\frac{F(\boldsymbol{x}_i)}{||F(\boldsymbol{x}_i)||}$ effectively resolves ACGAN's early training collapse (see Fig. 2). The motivation is that normalizing features onto the hypersphere will restrict the norm of gradient under 1.0 explicitly. Thus the discriminator does not experience the gradient exploding problem. From the next section, we will normalize both feature embeddings and weight vectors (proxies) together because applying normalization on both features and weight vectors does not degrade image generation performance, and it is convenient for future development. We denote the normalized feature $\frac{F(\boldsymbol{x}_i)}{||F(\boldsymbol{x}_i)||}$ as $\boldsymbol{f}_i$ and the normalized weight vector $\frac{\boldsymbol{w}_{y_i}}{||\boldsymbol{w}_{y_i}||}$ as $\boldsymbol{v}_{y_i}$.

3.2 Data-to-Data Cross-Entropy loss (D2D-CE)

We expand the feature normalized softmax cross-entropy loss described in Sec. 3.1 to Data-to-Data Cross-Entropy (D2D-CE) loss in this section. The motivations can be summarized in two points: (1) replace data-to-class similarities in the denominator of Eq. (3) with data-to-data similarities and (2) introduce strict thresholds to the modified softmax cross-entropy loss. We expect that point (1) will encourage the feature extractor to consider data-to-data as well as data-to-class relationships. Also, point (2) will guarantee inter-class separability and intra-class variations in the feature space while preventing ineffective gradient updates induced by easy negative and positive samples. To develop the feature normalized cross-entropy loss into D2D-CE, we firstly replace the similarities between a sample embedding and all label proxies except for the positive one in the denominator, $\sum_{j\in\{1,\dots,c\}\setminus\{y_i\}} \exp\left(\boldsymbol{f}_i^\top \boldsymbol{v}_j\right)$ with similarities between negative samples in the same mini-batch. The modified cross-entropy loss can be expressed as follows:

$$\mathcal{L}_{\text{CE}}^{'} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp\left(\boldsymbol{f}_i^\top \boldsymbol{v}_{y_i}/\tau\right)}{\exp\left(\boldsymbol{f}_i^\top \boldsymbol{v}_{y_i}/\tau\right) + \sum_{j\in\mathcal{N}(i)}\exp\left(\boldsymbol{f}_i^\top \boldsymbol{f}_j/\tau\right)}\right), \tag{5}$$

where τ is a temperature hyperparameter, and $\mathcal{N}(i)$ is the set of indices that indicate negative samples whose labels are different form the reference embedding f_i in the mini-batch. This process is described in Fig 3. The similarity matrix calculated by two mini-batches of sample embeddings is used to calculate the similarities between negative samples in the denominator with False Negative Mask. As a result, Eq. (5) encourages the discriminator to compare differences between data instances, so it will give a rich supervisory signal for conditional image generation. Finally, we introduce threshold values to $\mathcal{L}_{\text{CE}}^{'}$ and name it *Data-to-Data Cross-Entropy loss* (D2D-CE). The proposed D2D-CE can be expressed as follows:

$$\mathcal{L}_{\text{D2D-CE}} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{\exp\left([\boldsymbol{f}_i^\top \boldsymbol{v}_{y_i} - m_p]_-/\tau\right)}{\exp\left([\boldsymbol{f}_i^\top \boldsymbol{v}_{y_i} - m_p]_-/\tau\right) + \sum_{j\in\mathcal{N}(i)}\exp\left([\boldsymbol{f}_i^\top \boldsymbol{f}_j - m_n]_+/\tau\right)}\right), \tag{6}$$

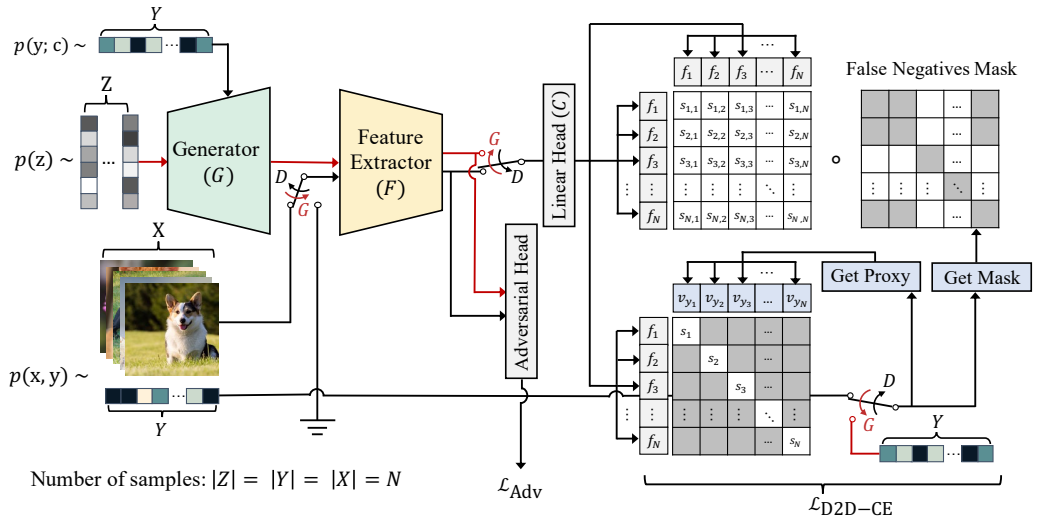


Figure 3: Overview of ReACGAN. ReACGAN performs adversarial training using the loss denoted as $\mathcal{L}_{\text{Adv}}$. At the same time, ReACGAN tries to minimize Data-to-Data Cross-Entropy loss (D2D-CE) on the linear head (C) to exploit relational information in the labeled dataset. $\circ$ means the element-wise product. Note that *False Negatives Mask* operates on the similarity matrix between two batches of image embeddings to compute negative terms in the denominator of Eq. (5, 6).

where m_p is a threshold for positive samples, m_n is a threshold for negative samples. $[\cdot]_-$ and $[\cdot]_+$ denote min($\cdot$,0) and max($\cdot$,0) functions, respectively.

3.3 Useful Properties of Data-to-Data Cross-Entropy loss

In this subsection, we explain four useful properties of D2D-CE. Let s_i be a similarity between the reference sample $\boldsymbol{f}_i$ and the corresponding normalized proxy $\boldsymbol{v}_{y_i}$, $s_{i,j}$ be a similarity between $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$, and a, b are arbitrary indices of negative samples, *i.e.*, $a, b \in \mathcal{N}(i)$. Then the properties of D2D-CE loss can be summarized as follows:

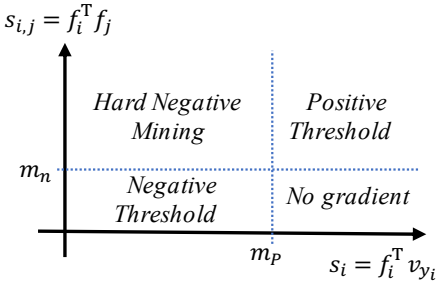


Figure 4: Graph showing the regions where hard negative mining, positive threshold, negative threshold, and no gradient update occur.

Property 1. *Hard Negative Mining. If the value of $\boldsymbol{s}_{i,a}$ is greater than $\boldsymbol{s}_{i,b}$, the derivative of $\mathcal{L}_{\text{D2D-CE}}$ w.r.t $\boldsymbol{s}_{i,a}$ is greater than or equal to the derivative w.r.t $\boldsymbol{s}_{i,b}$; that is $\frac{\partial \mathcal{L}_{\text{D2D-CE}}}{\partial \boldsymbol{s}_{i,a}} \geq \frac{\partial \mathcal{L}_{\text{D2D-CE}}}{\partial \boldsymbol{s}_{i,b}} \geq 0$.*

Property 2. *Positive threshold. If $\boldsymbol{s}_i - m_p > 0$, the derivative of $\mathcal{L}_{\text{D2D-CE}}$ w.r.t $\boldsymbol{s}_i$ is 0.*

Property 3. *Negative threshold. If $\boldsymbol{s}_{i,j} - m_n < 0$, the derivative of $\mathcal{L}_{\text{D2D-CE}}$ w.r.t $\boldsymbol{s}_{i,j}$ is 0.*

Property 4. *$\mathcal{L}_{\text{D2D-CE}}$ has the global minima of $\log(1 + |\mathcal{N}(i)|)$ when there does not exist an arbitrary i and j such that $\boldsymbol{s}_i - m_p < 0$ or $\boldsymbol{s}_{i,j} - m_n > 0$.*

We put proofs of the above properties in the supplement. Property 1 shows that D2D-CE can implicitly conduct hard negative mining and benefit from comparing samples each other. Also, Property 2 and 3 implies that samples will not affect gradient updates if the sample are trained sufficiently. As a result, the classifier can concentrate on pushing and pulling hard negative and positive examples without being dominated by easy negative and positive samples.

3.4 Rebooted Auxiliary Classifier Generative Adversarial Networks (ReACGAN)

With the proposed D2D-CE objective, we propose the *Rebooted Auxiliary Classifier Generative Adversarial Networks* (ReACGAN). As ACGAN does, ReACGAN jointly optimizes adversarial and

classification objectives. Specifically, the discriminator of ReACGAN tries to discriminate whether a given image is sampled from the real distribution or not. At the same time, the discriminator tries to maximize similarities between real positive samples while minimizing similarities between real negative samples using D2D-CE loss. After predetermined numbers of the discriminator updates end, the generator strives to deceive the discriminator by generating realistic images that will give a low D2D-CE objective value. By alternatively training the discriminator and generator until convergence, ReACGAN can generate high-quality images of diverse categories without early training collapse. We attach the algorithm table for ReACGAN training in the supplement.

Differences between ReACGAN and ContraGAN. The authors of ContraGAN [22] propose a conditional contrastive loss (2C loss) to cover data-to-data relationships when training cGANs. The main differences between 2C loss and D2D-CE are summed up into three points: (1) while 2C loss is derived from NT-Xent loss [38], which is popularly used in self-supervised learning, our D2D-CE is developed to resolve the early training collapsing problem and poor generation performance of ACGAN, (2) 2C loss holds false negative samples in the denominator, and it causes unexpected forces to repulse the positive samples out of the reference image, and (3) 2C loss contains similarities between all positive samples in the numerator, and it will give rise to large gradients on easy positive samples. As a result, GANs with 2C loss tend to synthesize images of unintended class as reported in the original author's implementation [39]. However, D2D-CE loss does not contain the false negatives in the denominator and only considers the similarity between data and its proxy in the numerator. Therefore, ReACGAN does not conduct unnecessary easy positive mining and is free from the unwanted repelling forces. More detailed explanations of this are attached in the supplement.

Consistency Regularization and D2D-CE loss. Zhang *et al.* [16] propose consistency regularization to let the discriminator output consistent predictions if given two images are visually close to each other. They make a visually similar image pair by augmenting the reference image with pre-defined augmentations. After that, they force the discriminator to minimize the L2 distance between logits of the reference and augmented images. While ReACGAN locates an image embedding nearby to its corresponding proxy and far apart from multiple image embeddings according to class labels, consistency regularization only pulls the original and augmented images towards each other. Since Consistency Regularization and D2D-CE can be applied together, we will show that ReACGAN benefits from consistency regularization in the experiments section (Table 1, 2).

4 Experiments

4.1 Datasets and Evaluation Metrics

To verify the effectiveness of ReACGAN, we conduct conditional image generation experiments using four datasets: CIFAR10 [27], Tiny-Imagenet [30], CUB200 [29], and Imagenet [28] and four evaluation metrics: Inception Score (IS) [31], Frechet Inception Distance (FID) [32], and $F_{0.125}$ (Precision) and F_8 (Recall) [33]. The details of training datasets are in the supplement.

Inception Score (IS) [31] **and Frechet Inception Distance (FID)** [32] are commonly used metrics for generative model evaluation. We utilize IS and FID together because some studies [4, 6, 24] have shown that IS has a tendency to indicate the fidelity of images better while FID tends to capture the diversity of samples.

Precision ($F_{0.125}$) and Recall (F_8) [33] are metrics for estimating precision and recall of the approximated distribution $p(G(\mathbf{z}))$ against the true data distribution $p(\mathbf{x})$. Instead of evaluating generative models using one-dimensional scores, such as IS and FID, Sajjadi *et al.* [33] suggest using two-dimensional score $F_{0.125}$ and F_8 that can quantify how precisely the generated images are and how well the generated images cover the reference distribution.

4.2 Experimental Details

The implementation of our proposed method basically follows PyTorch-StudioGAN library [39, 22][1] that supports various experimental setups from ACGAN [18] to ContraGAN [22] with diverse scales of datasets [27, 30, 28] and architectures [2, 13, 4]. For a fair comparison, we use the same backbones over all baselines, except otherwise noted, for both discriminator and generator. For stable

[1]PyTorch-StudioGAN is an open-source library under the MIT License (MIT).

Table 1: Comparisons with classifier-based GANs [18, 22] and projection-based GANs [3, 4, 26] on CIFAR10 [27], Tiny-Imagenet [30], and CUB200 [29] datasets using IS [31], FID [32], $F_{0.125}$ and F_8 [33] metrics. For baselines, both the numbers in the original paper (denoted as * in method) and from our experiments using StudioGAN library [22] are reported.

Method	CIFAR10 [27]				Tiny-Imagenet [30]				CUB200 [29]			
	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑
SNGAN* [3]	8.22	21.7	-	-	-	-	-	-	-	-	-	-
BigGAN* [4]	9.22	14.73	-	-	-	-	-	-	-	-	-	-
ContraGAN* [22]	-	10.60	-	-	-	29.49	-	-	-	-	-	-
ACGAN [3]	9.84	8.45	0.993	**0.992**	6.00	96.04	0.656	0.368	**6.09**	60.73	0.726	0.891
SNGAN [3]	8.67	13.33	0.985	0.976	8.71	51.15	0.900	0.702	5.41	47.75	0.754	0.912
SAGAN [26]	8.66	14.31	0.983	0.973	8.74	49.90	0.872	0.712	5.48	54.29	0.728	0.882
BigGAN [4]	9.81	8.08	0.993	**0.992**	12.78	32.03	0.948	0.868	4.98	18.30	0.924	**0.967**
ContraGAN [22]	9.70	8.22	0.993	0.991	13.46	28.55	**0.974**	0.881	5.34	21.16	0.935	0.942
ReACGAN	**9.89**	**7.88**	**0.994**	**0.992**	**14.06**	**27.10**	0.970	**0.894**	4.91	**15.40**	**0.970**	0.954
CR-BigGAN* [16]	-	11.48	-	-	-	-	-	-	-	-	-	-
CR-BigGAN [16]	9.97	**7.18**	0.995	0.993	15.94	19.96	0.972	**0.950**	**5.14**	11.97	0.978	**0.981**
CR-ContraGAN [16, 22]	9.59	8.55	0.992	0.972	15.81	**19.21**	0.983	0.941	4.90	11.08	0.984	0.967
CR-ReACGAN [16]	**10.11**	7.20	**0.996**	**0.994**	**16.56**	19.69	**0.984**	0.940	4.87	**10.72**	**0.985**	0.971
DiffAug-BigGAN* [9]	9.17	8.49	-	-	-	-	-	-	-	-	-	-
DiffAug-BigGAN [9]	9.94	7.17	0.995	0.992	18.08	15.70	0.980	**0.972**	**5.53**	12.15	0.967	0.981
DiffAug-ContraGAN [9, 22]	9.95	7.27	0.995	0.992	18.20	15.40	0.986	0.963	5.39	11.02	0.978	0.970
DiffAug-ReACGAN [9]	**10.22**	**6.79**	**0.996**	**0.993**	**20.60**	**14.25**	**0.988**	**0.972**	5.22	**9.27**	**0.985**	**0.983**
Real Data	11.54				34.11				5.49			

training, we apply Spectral Normalization (SN) [3] to the generator and discriminator except for experiments using SNGAN (in this case, we apply SN to the discriminator only). We also use the same conditioning method for generator with conditional Batch Normalization (cBN) [40, 41, 19]. This allows us to investigate solely the conditioning methods and adversarial losses of discriminator, which are the main interests to validate our proposed method. If not specified, we use Hinge loss [42] as a default for the adversarial loss.

Before conducting main experiments, we perform grid search for hyperparameter selection with the candidates of temperature $\tau \in \{0.125, 0.25, 0.5, 0.75, 1.0\}$ and negative threshold $m_n \in \{0.5, 0.75, 0.9, 0.95, 0.98, 1.0\}$. For convenience, we set positive threshold m_p as $1 - m_n$. Through extensive experiments with 3 runs per each setting, we choose τ with $\{0.5, 0.75, 0.25, 0.5\}$ and m_n with $\{0.98, 1.0, 0.95, 0.98\}$ for CIFAR10, Tiny-Imagenet, CUB200 and Imagenet, respectively. Low temperature seems to work well on fine-grained image generation task, but generally, ReACGAN is robust to the choice of hyperparameters. The results of hyperparameter search and other hyperparameter setups are provided in the supplement.

We evaluate all methods through the same protocols of [9, 16, 17], which use the same amounts of generated images from the reference splits specialized for each dataset.[2] Besides that, we run all the experiments three times with different random seeds and report the averaged best performances for reliable evaluation with the lone exception of Imagenet experiments. We run the Imagenet experiments using the batch size of 256 due to the limited computation resources. Please refer to supplementary materials for other experimental details. From the evaluation results section, the numbers in bold-faced denote the best performance, and in underline indicate that the values are in one standard deviation from the best.

4.3 Evaluation Results

Comparison with other GANs. We compare ReACGAN with previous state-of-the-art GANs in Table 1 and 2. Note that implemented GANs in PyTorch-StudioGAN library give outstanding generation results on CIFAR10 dataset and even surpass the current reported state-of-the-art [9] with a relatively simple framework, e.g., BigGAN [4]. Thus, we provide both the numbers in the original paper (denoted as * in method) and from StudioGAN library [22] for a fair comparison. Moreover, we

[2] We use the validation split as the default reference set, but we use the test split of CIFAR10 and the training split of CUB200 due to the absence of the validation split of such datasets.

Table 2: Experiments using Imagenet [28] dataset. (Left): Comparisons with previous cGANs approaches. (Right): Learning curves of BigGAN [4], ContraGAN [22], and ReACGAN (ours) using FID [32] and IS [31] scores. B.S. means batch size for training.

	Method	Imagenet [28]			
		IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑
B.S. = 256	ACGAN* [18, 20]	7.26	184.41	-	-
	SNGAN* [3]	36.80	27.62	-	-
	SAGAN* [26]	52.52	18.28	-	-
	BigGAN* [4, 20]	38.05	22.77	-	-
	TAC-GAN* [20]	28.86	23.75	-	-
	ContraGAN‡,* [22]	31.10	19.69	0.951	0.927
	ACGAN [18]	62.99	26.35	0.935	0.963
	SNGAN [3]	32.25	26.79	0.938	0.913
	SAGAN [26]	29.85	34.73	0.849	0.914
	BigGAN [4]	28.63	24.68	0.941	0.921
	ContraGAN [22]	25.25	25.16	0.947	0.855
	ReACGAN	**63.16**	**14.59**	**0.974**	**0.974**
	DiffAug-BigGAN [9]	36.97	18.57	0.956	0.941
	DiffAug-ReACGAN [9]	**65.88**	**12.25**	**0.979**	**0.977**
	Real Data	173.33			

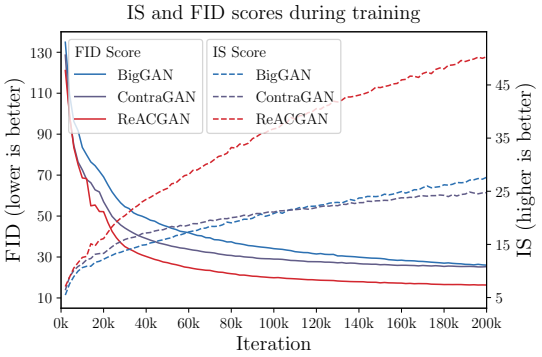


Table 3: Experiments on the effectiveness of D2D-CE loss compared to other conditioning losses.

Conditioning Method	CIFAR10 [27]				Tiny-Imagenet [30]				CUB200 [29]			
	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑
BigGAN w/o Condition [4] (Abbreviated to Big)	9.46	12.21	0.987	0.982	7.38	76.15	0.804	0.576	5.16	35.17	0.852	0.936
Big + AC [18]	9.84	8.45	0.993	**0.992**	6.00	96.04	0.656	0.368	**6.09**	60.73	0.726	0.891
Big + PD [19]	9.88	8.01	**0.994**	0.990	12.78	32.03	0.948	0.868	4.98	18.30	0.924	**0.967**
Big + MH [25]	**10.05**	7.94	**0.994**	0.990	4.37	140.74	0.282	0.156	5.18	245.69	0.625	0.832
Big + 2C [22]	9.70	8.21	0.993	0.991	13.46	28.50	**0.974**	0.881	5.34	21.16	0.935	0.942
Big + D2D-CE	9.89	**7.88**	**0.994**	**0.992**	**14.06**	**27.10**	0.970	**0.894**	4.91	**15.40**	**0.970**	0.954

conduct experiments with popular augmentation-based methods, consistency regularization (CR) [16] and differentiable augmentations (DiffAug) [9].

Compared with other strong baselines, ReACGAN remains the best one on all datasets and almost all metrics, surpassing the previous art by 2.5%, 5.1%, 15.8%, and 20.2% in FIDs, respectively. In addition, ReACGANs are well harmonized with augmentation-based regularizations (the bottom two and the last sections in Table 1 and 2), bringing incremental improvements on all metrics. For Imagenet experiments, our ReACGAN reaches higher IS and lower FID with fewer iterations than other competitive approaches. Lastly, ReACGAN and its augmented version with DiffAug [9] set new state-of-the-art records on Imagenet based on all metrics. Note that ACGAN showes high IS values (6.09 on CUB200 and 62.99 on Imagenet), but relatively poor FID scores (60.73 on CUB200 and 26.35 on Imagenet), which means ACGAN generates easily-classifiable samples without considering diversity.

Comparison with other conditional GANs. We investigate how the generation qualities vary with different conditioning losses while keeping the other configurations fixed. We compare our D2D-CE loss with cross-entropy loss of ACGAN (AC) [18], the loss of projection discriminator (PD) [19], multi-hinge loss (MH) [25], and conditional contrastive loss (2C) [22]. The results are shown in Table 3. AC- and MH-based models present decent performances on CIFAR10, but undergo early collapse on Tiny-Imagenet and CUB200 training. Replacing them with PD and 2C losses produces satisfactory results across all datasets, where PD loss makes the best F_8 (recall) on CUB200 experiment while giving third $F_{0.125}$ (precision) value. The noticeable point is that the proposed D2D-CE shows considerable degree of robustness showing the highest FID, F_8, and $F_{0.125}$ values in most cases. This means that ReACGANs generate high-fidelity images and are relatively free from the precision and recall trade-off [33] than the others.

4.4 Ablation Study

We study how the proposed components of ReACGAN affect ACGAN training. By adding or ablating each part of ReACGAN, as shown in Table 4, we identify four major observations. (1) Feature and

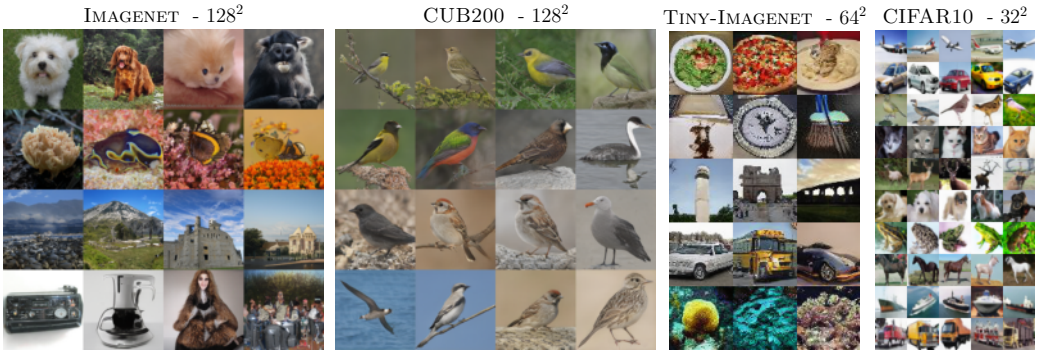


Figure 5: Curated images generated by the proposed ReACGAN.

Table 4: Ablation study on normalizing feature maps, data-to-data consideration, and easy negative and positive thresholding. The last row of the table shows the performances of ACGAN with the normalization and easy negative and positive thresholding without data-to-data consideration.

Ablation	Tiny-Imagenet [30]				CUB200 [29]				Imagenet [43]			
	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑	IS ↑	FID ↓	$F_{0.125}$ ↑	F_8 ↑
ACGAN [18]	6.00	96.04	0.656	0.368	**6.09**	60.73	0.726	0.891	62.99	26.35	0.935	0.963
+ Normalization	13.46	30.33	0.955	0.889	4.78	25.54	0.883	0.952	18.16	36.40	0.879	0.787
+ Data-to-Data (Eq. 5)	12.96	28.71	0.967	0.863	5.08	25.12	0.894	0.946	-	-	-	-
+ Thresholding (Eq. 6)	**14.06**	**27.10**	**0.970**	**0.894**	4.91	**15.40**	**0.970**	**0.954**	**63.16**	**14.59**	**0.974**	**0.974**
- Data-to-data	12.96	30.79	0.960	0.857	5.39	30.36	0.863	0.947	-	-	-	-

weight normalizations greatly stabilize ACGAN training and improve generation performances on Tiny-Imagenet and CUB200 datasets. (2) D2D-CE loss can enhance the generation performance by considering data-to-data relationships and by performing easy positive and negative thresholding (3rd-4th rows). (3) Thresholding technique does not work well on the feature normalized cross-entropy loss Eq. (5) (5th row). (4) While ACGAN shows high Inception Score on Imagenet experiment, it shows relatively low FID, $F_{0.125}$, and F_8 values compared to the model trained with normalization, data-to-data consideration, and the easy sample thresholding (ReACGAN). This is consistent with the qualitative results on ImageNet, where the images from ACGAN are easily classifiable with mode-shrinking, but images from ReACGAN of high quality and diverse (See Fig. 5). We attribute this improvement to the discriminator that successfully leverages rich data-to-data and data-to-class relationships with easy positive and negative thresholding.

5 Conclusion

In this paper, we have analyzed why training ACGAN becomes unstable as the number of classes of the dataset increases. By deriving the analytic form of gradients of the classifier and numerically computing the gradient values, we have discovered that the training instability comes from a gradient exploding problem caused by the unboundedness of input feature vectors and poor classification ability in the early training phase. To alleviate the instability and reinforce ACGAN models, we have proposed a novel Data-to-Data cross-entropy loss (D2D-CE) and Rebooted Auxiliary Classifier Generative Adversarial Networks (ReACGAN). Although ReACGAN requires additional hyperparameters of temperature and threshold values, we have found that they do not significantly affect generation performance. The experimental results demonstrate the superiority of ReACGAN compared with the existing classifier and projection-based GANs on four benchmark datasets. We would like to conduct a large-scale conditional image generation experiment on Imagenet with a larger batch size as future work. Also, we want to explore applying more diverse augmentations used in SimCLR [38], adaptive discriminator augmentations [8], and contrastive discriminator [44] to identify that ReACGAN can benefit from more complex data augmentations.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv preprint arXiv 1511.06434*, 2016.

[3] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral Normalization for Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4401–4410, 2019.

[6] Yan Wu, Jeff Donahue, David Balduzzi, Karen Simonyan, and Timothy P. Lillicrap. LOGAN: Latent Optimisation for Generative Adversarial Networks. *arXiv preprint arXiv 1912.00953*, 2019.

[7] 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 International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8110–8119, 2020.

[8] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

[10] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhixiang Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017.

[11] Martín Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv 1701.07875*, 2017.

[12] Martín Arjovsky and Léon Bottou. Towards Principled Methods for Training Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[13] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved Training of Wasserstein GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5767–5777, 2017.

[14] Naveen Kodali, James Hays, Jacob D. Abernethy, and Zsolt Kira. On Convergence and Stability of GANs. *arXiv preprint arXiv 1705.07215*, 2018.

[15] Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, and Zhihua Zhang. Lipschitz generative adversarial nets. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7584–7593, 2019.

[16] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency Regularization for Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[17] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved Consistency Regularization for GANs. *arXiv preprint arXiv 2002.04724*, 2020.

[18] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis with Auxiliary Classifier GANs. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2642–2651, 2017.

[19] Takeru Miyato and Masanori Koyama. cGANs with Projection Discriminator. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[20] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin Auxilary Classifiers GAN. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[21] Aliaksandr Siarohin, Enver Sangineto, and Nicu Sebe. Whitening and Coloring Batch Transform for GANs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[22] Minguk Kang and Jaesik Park. ContraGAN: Contrastive Learning for Conditional Image Generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[23] Woohyeon Shim and Minsu Cho. CircleGAN: Generative Adversarial Learning across Spherical Circles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[24] Peng Zhou, Lingxi Xie, Bingbing Ni, and Qi Tian. Omni-GAN: On the Secrets of cGANs and Beyond. *arXiv preprint arXiv:2011.13074*, 2020.

[25] Ilya Kavalerov, Wojciech Czaja, and Rama Chellappa. A multi-class hinge loss for conditional gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[26] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7354–7363, 2019.

[27] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. PhD thesis, University of Toronto, 2012.

[28] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[29] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical report, California Institute of Technology, 2010.

[30] Johnson et al. Tiny ImageNet Visual Recognition Challenge. https://tiny-imagenet.herokuapp.com.

[31] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2234–2242, 2016.

[32] 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 (NeurIPS)*, pages 6626–6637, 2017.

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

[34] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 271–279, 2016.

[35] Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles A Sutton. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[36] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv preprint arXiv 1411.1784*, 2014.

[37] Zhiming Zhou, Han Cai, Shu Rong, Yuxuan Song, Kan Ren, Weinan Zhang, Jun Wang, and Yong Yu. Activation Maximization Generative Adversarial Nets. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[38] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv preprint arXiv 2002.05709*, 2020.

[39] Minguk Kang and Jaesik Park. Pytorch-studiogan. https://github.com/POSTECH-CVLab/PyTorch-StudioGAN, 2020.

[40] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A Learned Representation For Artistic Style. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[41] Harm de Vries, Florian Strub, Jeremie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6594–6604, 2017.

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

[43] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[44] Jongheon Jeong and Jinwoo Shin. Training GANs with Stronger Augmentations via Contrastive Discriminator. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The main contributions of our paper can be summarized into two points: (1) analyzing why ACGAN training becomes unstable as the number of classes in the training dataset increases, (2) developing a new cGAN model: Rebooted Auxiliary Classifier Generative Adversarial Networks (ReACGAN). Both claims are well reflected in the abstract and introduction.
   (b) Did you describe the limitations of your work? [Yes] The proposed ReACGAN can generate high-quality images using fruitful data-to-data and data-to-class relationships in the dataset. However, it requires additional hyperparameters tunning for temperature and threshold values. We explain the weaknesses of our work in Sec. 5
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Generative Adversarial Networks can be used to synthesize indistinguishable images from the real. This might cause fake news, sexual harnesses, and hacking machine vision applications. Although the potential negative social impacts are not discussed in our main paper, we will make a section to deal with this in the supplement.
   (d) Have you read the ethics review and ensured that your paper conforms to them? [Yes] We carefully read the ethics review guidelines and comply with them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We state all the assumptions that arise from the theoretical results, i.e., Property 1-4 in Sec. 3.3.
   (b) Did you include complete proofs of all theoretical results? [Yes] We provide the complete proofs in the supplement.
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] We will include the code for reproducing our main results in the supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We state the details for training setups in Sec. 4.2, which include the evaluation procedure, hyperparameter selection, and controlled environment for a fair comparison. In addition to this, we will add how we preprocess the training dataset in the supplement.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] For reliable evaluation, we run all the experiments three times with different random seeds. We will provide standard deviations of Table 1 and 2 in the supplement except for Imagenet experiments where we perform just a single run due to the limited resources.
   (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] We include this information in the supplement.
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] We cite the papers and the GitHub repository that we used for our experiments in Sec 4.2.

(b) Did you mention the license of the assets? [No] We utilize four datasets to validate our method: CIFAR10, Tiny-Imagenet, CUB200, and Imagenet. These are widely used public datasets, and we cite them in the paper. However, we could not find any license information for CIFAR10, Tiny-Imagenet, and CUB200. The Imagenet dataset can be used for the research proposes only. In addition, we mention the license of PyTorch-StudioGAN in the main paper which is used for our experiments.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We will include a zip file for the implementation of our model in the supplement.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The CIFAR10, Tiny-Imagenet, CUB200, Imagenet datasets are properly curated, and they are popular datasets for research purposes. To our best knowledge, there is no inappropriate content in the datasets.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The benchmark datasets we used are widely adopted by others and do not contain any harmful or troublesome contents.

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] Our research has no relevance to the crowdsourcing or human subjects.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] Our research has no relevance to the crowdsourcing or human subjects.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] Our research has no relevance to the crowdsourcing or human subjects.