Conditional GANs with Auxiliary Discriminative Classifier

Liang Hou [1 2]   Qi Cao [1]   Huawei Shen [1 2]   Siyuan Pan [3]   Xiaoshuang Li [3]   Xueqi Cheng [4 2]

Abstract

Conditional generative models aim to learn the underlying joint distribution of data and labels to achieve conditional data generation. Among them, the auxiliary classifier generative adversarial network (AC-GAN) has been widely used, but suffers from the problem of low intra-class diversity of the generated samples. The fundamental reason pointed out in this paper is that the classifier of AC-GAN is generator-agnostic, which therefore cannot provide informative guidance for the generator to approach the joint distribution, resulting in a minimization of the conditional entropy that decreases the intra-class diversity. Motivated by this understanding, we propose a novel conditional GAN with an auxiliary discriminative classifier (ADC-GAN) to resolve the above problem. Specifically, the proposed auxiliary discriminative classifier becomes generator-aware by recognizing the class-labels of the real data and the generated data discriminatively. Our theoretical analysis reveals that the generator can faithfully learn the joint distribution even without the original discriminator, making the proposed ADC-GAN robust to the value of the coefficient hyperparameter and the selection of the GAN loss, and stable during training. Extensive experimental results on synthetic and real-world datasets demonstrate the superiority of ADC-GAN in conditional generative modeling compared to state-of-the-art classifier-based and projection-based conditional GANs.

---
[1]Data Intelligence System Research Center, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China [2]University of Chinese Academy of Sciences, Beijing, China [3]Shanghai Jiao Tong University, Shanghai, China [4]CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. Correspondence to: Huawei Shen <shenhuawei@ict.ac.cn>.

1. Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) have achieved substantial progress in learning high-dimensional, complex data distribution such as images (Brock et al., 2019; Karras et al., 2019; 2020b;a; Karras et al.). Standard GANs consist of a generator network, which transfers latent codes sampled from tractable distributions such as Gaussian in the latent space to data points in the data space, and a discriminator network, which attempts to distinguish real data and generated data. The generator is trained in an adversarial game against the discriminator so that it can learn the data distribution at the Nash equilibrium. Remarkably, training GANs unconditionally is difficult to achieve equilibrium, making the generator prone to mode collapse (Salimans et al., 2016; Lin et al., 2018; Chen et al., 2019). In addition, practitioners are interested in being able to control in advance the content of the generated samples (Yan et al., 2015; Tan et al., 2020) in practical applications. A promising solution to these issues is conditioning the generator, leading to conditional GANs.

Conditional GANs (cGANs) (Mirza & Osindero, 2014) is a family of variants of GANs that leverages the side information from annotated labels of samples to implement and train a conditional generator for conditional image generation from class-labels (Odena et al., 2017; Miyato & Koyama, 2018; Brock et al., 2019). To implement the conditional generator, the common technique nowadays injects the conditional information via conditional batch normalization (de Vries et al., 2017; Hou et al., 2021b). To train the conditional generator, a lot of effort put into effectively injecting the conditional information into the discriminator or auxiliary classifier that guides the conditional generator (Odena, 2016; Miyato & Koyama, 2018; Zhou et al., 2018; Kavalerov et al., 2021; Kang & Park, 2020; Zhou et al., 2020). Among them, the auxiliary classifier generative adversarial network (AC-GAN) (Odena et al., 2017) has been widely used due to its simplicity and extensibility. Specifically, AC-GAN utilizes an auxiliary classifier that first attempts to recognize the labels of data and then teaches the generator to produce label-consistent (classifiable) data. However, it has been reported that AC-GAN suffers from the low intra-class diversity problem in the generated samples, especially on datasets with a large number of classes (Odena et al., 2017; Shu et al., 2017; Gong et al., 2019).

In this study, we point out that the fundamental reason for the low intra-class diversity problem of AC-GAN is that the classifier is agnostic to the generated data distribution and thus cannot provide informative guidance for the generator to learn the target distribution. Motivated by this understanding, we propose a novel conditional GAN with an auxiliary discriminative classifier, namely ADC-GAN, to resolve the above problem by enabling the classifier to be aware of the generated data distribution as well as the real data distribution. To this end, the discriminative classifier is trained to distinguish between the real and generated data while recognizing their class-labels. The discriminative capability allows the classifier to provide the discrepancy between the real and generated data distributions like the discriminator, and the classification capability enables it to capture the dependencies between data and labels. We show in theory that the generator of our proposed ADC-GAN can learn the joint data and label distribution under the optimal discriminative classifier even without the discriminator, making the method robust to the value of the coefficient hyperparameter and the selection of the GAN loss and stable during training. We also highlight the superiority of ADC-GAN compared to the two most related works (TAC-GAN (Gong et al., 2019) and PD-GAN (Miyato & Koyama, 2018)) by analyzing their potential issues and limitations. Results on synthetic data clearly show that the proposed ADC-GAN successfully resolves the problem of AC-GAN by faithfully recovering the joint distribution of real data and labels. Extensive experiments based on two popular codebases demonstrate the effectiveness of the proposed ADC-GAN compared with state-of-the-art cGANs in conditional generative modeling.

2. Preliminaries and Analysis

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) (Goodfellow et al., 2014) consist of two types of neural networks: the generator $G : \mathcal{Z} \to \mathcal{X}$ that maps a latent code $z \in \mathcal{Z}$ endowed with an easily sampled distribution P_Z to a data point $x \in \mathcal{X}$, and the discriminator $D : \mathcal{X} \to [0, 1]$ that distinguishes between real data that sampled from the real data distribution P_X and fake data that sampled from the generated data distribution $Q_X = G_\sharp P_Z$ induced by the generator. The goal of the generator is to confuse the discriminator by producing data that are as real as possible. Formally, the objective functions for the discriminator and generator are defined as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim P_X}[\log D(x)] \\ + \mathbb{E}_{x \sim Q_X}[\log(1 - D(x))]. \quad (1)$$

Theoretically, learning the generator under the optimal discriminator can be regarded as minimizing the Jensen-Shannon (JS) divergence between the real data distribution and the generated data distribution, i.e., $\min_G \mathrm{JS}(P_X \| Q_X)$.

This would enable the generator to restore the real data distribution at its optimum. However, the training of GANs on complex natural images is typically unstable (Che et al., 2016), especially in the absence of supervision such as conditional information. In addtition, the content of the images generated by GANs cannot be specified in advance.

2.2. Base Method: AC-GAN

Learning GANs with conditional information can not only improve the training stability but also achieve conditional generation. As one of the most representative conditional GANs, AC-GAN (Odena et al., 2017) utilizes an auxiliary classifier $C : \mathcal{X} \to \mathcal{Y}$ to learn the dependencies between data and labels endowed with a label prior P_Y and then encourages the conditional generator $G : \mathcal{Z} \times \mathcal{Y} \to \mathcal{X}$ to generate as much classifiable data as possible. The objective functions for the discriminator, the auxiliary classifier, and the generator of AC-GAN[1] are defined as follows:

$$\max_{D,C} V(G, D) + \lambda \cdot \left(\mathbb{E}_{x,y \sim P_{X,Y}}[\log C(y|x)] \right), \quad (2)$$

$$\min_G V(G, D) - \lambda \cdot \left(\mathbb{E}_{x,y \sim Q_{X,Y}}[\log C(y|x)] \right), \quad (3)$$

where $\lambda > 0$ is a coefficient hyperparameter, $P_{X,Y}$ indicates the joint distribution of real data and labels, and $Q_{X,Y} = G_\sharp(P_Z \times P_Y)$ denotes the joint distribution of the generated data and labels induced by the conditional generator.

Proposition 2.1. *For fixed generator, the optimal classifier of AC-GAN has the form of $C^*(y|x) = \frac{p(x,y)}{p(x)}$.*

Theorem 2.2. *Given the optimal classifier, at the equilibrium point, optimizing the classification task for the generator of AC-GAN is equivalent to:*

$$\min_G \mathrm{KL}(Q_{X,Y} \| P_{X,Y}) - \mathrm{KL}(Q_X \| P_X) + H_Q(Y|X), \quad (4)$$

where $H_Q(Y|X) = -\int \sum_y q(x,y) \log q(y|x) \mathrm{d}x$ is the conditional entropy of the generated samples.

The proofs of all theorems are referred to Appendix A. Our Theorem 2.2 exposes two shortcomings of AC-GAN. Firstly, maximization of the KL divergence between the marginal generator and data distributions ($\max_G \mathrm{KL}(Q_X \| P_X)$) contradicts the goal of conditional generative modeling that matches $Q_{X,Y}$ with $P_{X,Y}$. Although this issue can be mitigated to some extent by the adversarial game between the discriminator and generator that minimizes the JS divergence between the two marginal distributions ($\min_G \mathrm{JS}(Q_X \| P_X)$), we find that it still has a negative impact on training stability and generation performance. Secondly, minimization of the entropy of labels conditioned

---

[1]We follow the common practice in the literature to adopt the stable version instead of the original one. We also provide an analysis of the original AC-GAN in Appendix B.
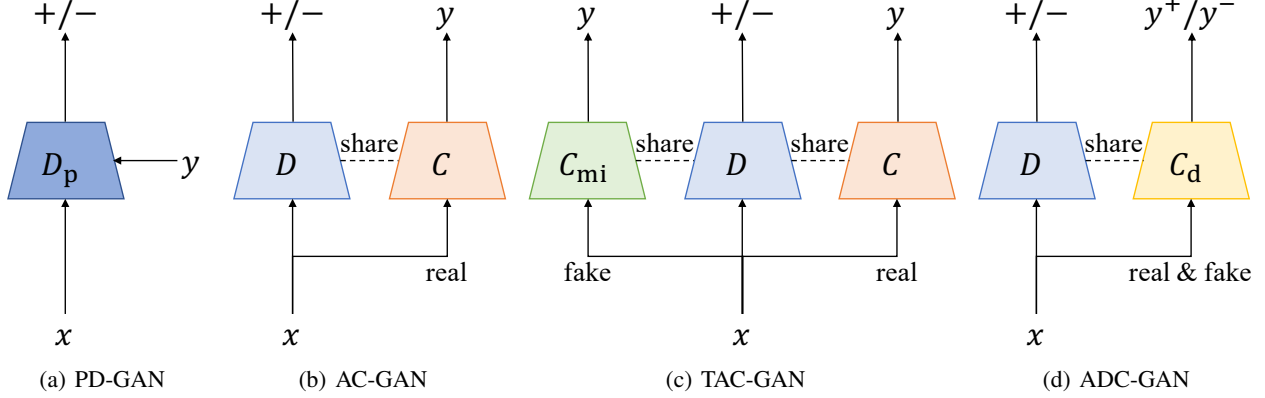
Figure 1: Illustration of discriminators/classifiers of existing cGANs (PD-GAN (Miyato & Koyama, 2018), AC-GAN (Odena et al., 2017), and TAC-GAN (Gong et al., 2019)) and ADC-GAN. The symbol $+/-$ indicates the GAN labels (real or fake) and y is the class-label of data x. ADC-GAN is different from PD-GAN by explicitly predicting the label and is different from AC-GAN and TAC-GAN in that the classifier C_d also distinguishes real from generated, like the discriminator.

on data of the generated distribution ($\min_G H_Q(Y|X)$) will result in the label of the generated data being deterministic. In other words, it forces the generated data for each class away from the classification hyperplane, explaining the low intra-class diversity of the generated samples in AC-GAN, especially when the distributions of different classes have non-negligible overlap, which occurs naturally as the fact that neither state-of-the-art classifiers nor human beings can achieve 100% classification accuracy on real-world datasets (Russakovsky et al., 2015). The original AC-GAN, whose classifier is trained from both real and generated samples, suffers from the same issue (cf. Appendix B).

3. Proposed Method: ADC-GAN

The goal of conditional generative modeling is to faithfully learn the joint distribution of real data and labels regardless of the shape of the joint distribution (whether there is overlap between the distributions of different classes). We first note that the reason why AC-GAN fails to learn the target joint distribution (Theorem 2.2) originates from that the optimal classifier $C^*(y|x) = \frac{p(x,y)}{p(x)}$ (Proposition 2.1) is agnostic to the density of the generated (marginal or joint) distribution ($q(x)$ or $q(x,y)$). As a result, the classifier cannot provide the discrepancy between the target distribution and the generated distribution, resulting in a biased learning objective of the generator. Recall that the optimal discriminator $D^*(x) = \frac{p(x)}{p(x)+q(x)}$ is aware of the real data distribution as well as the generated data distribution (Goodfellow et al., 2014), and can therefore provide the discrepancy between the real and generated data distributions $\frac{p(x)}{q(x)} = \frac{D^*(x)}{1-D^*(x)}$ for faithful generative modeling of the generator. Intuitively, the distribution-aware ability on both real and generated data is caused by the fact that the discriminator distinguishes

between the real and generated data with different labels (real or fake). Motivated by this understanding, we propose to make the classifier capable of classifying the the real and generated data with different class-labels, establishing a discriminative classifier $C_\mathrm{d} : \mathcal{X} \to \mathcal{Y}^+ \cup \mathcal{Y}^-$ ($\mathcal{Y}^+$ for real data and $\mathcal{Y}^-$ for generated data) that recognizes the label of the real and generated samples discriminatively. The generator is encouraged to produce classifiable real data rather than classifiable fake data. Mathematically, the objective functions for the discriminator, the discriminative classifier, and the generator of ADC-GAN are defined as:

$$\max_{D,C_\mathrm{d}} V(G,D) + \lambda \cdot (\mathbb{E}_{x,y\sim P_{X,Y}}[\log C_\mathrm{d}(y^+|x)]$$
$$+ \mathbb{E}_{x,y\sim Q_{X,Y}}[\log C_\mathrm{d}(y^-|x)]), \quad (5)$$
$$\min_G V(G,D) - \lambda \cdot (\mathbb{E}_{x,y\sim Q_{X,Y}}[\log C_\mathrm{d}(y^+|x)]$$
$$- \mathbb{E}_{x,y\sim Q_{X,Y}}[\log C_\mathrm{d}(y^-|x)]), \quad (6)$$

where $C_\mathrm{d}(y^+|x) = \frac{\exp(\varphi^+(y)\cdot\phi(x))}{\sum_{\bar{y}}\exp(\varphi^+(\bar{y})\cdot\phi(x))+\sum_{\bar{y}}\exp(\varphi^-(\bar{y})\cdot\phi(x))}$ (resp. $C_\mathrm{d}(y^-|x) = \frac{\exp(\varphi^-(y)\cdot\phi(x))}{\sum_{\bar{y}}\exp(\varphi^+(\bar{y})\cdot\phi(x))+\sum_{\bar{y}}\exp(\varphi^-(\bar{y})\cdot\phi(x))}$) indicates the probability that a data x is classified as the label y and real (resp. fake) simultaneously by the discriminative classifier. Here, $\phi : \mathcal{X} \to \mathbb{R}^d$ is a feature extractor that is shared with the original discriminator in our implementation ($D = \sigma \circ \psi \circ \phi$ with a linear mapping $\psi : \mathbb{R}^d \to \mathbb{R}$ and a sigmoid function $\sigma : \mathbb{R} \to [0,1]$), and $\varphi^+ : \mathcal{Y} \to \mathbb{R}^d$ and $\varphi^- : \mathcal{Y} \to \mathbb{R}^d$ capture learnable embeddings of labels responsible to the real and generated data, respectively.

At the first glance, the objective function with the discriminative classifier for the generator seems to be redundant as maximization of $\log C_\mathrm{d}(y^+|x)$ implicitly contains the goal of minimization of $\log C_\mathrm{d}(y^-|x)$. However, we show below that the second term is indispensable for accurately learning

Table 1: Theoretical learning objective for the generator of competing methods under the optimal discriminator and classifier.

METHOD	THEORETICAL LEARNING OBJECTIVE FOR THE GENERATOR
AC-GAN (ODENA ET AL., 2017)	$\min_G \text{JS}(P_X \| Q_X) + \lambda(\text{KL}(Q_{X,Y} \| P_{X,Y}) - \text{KL}(Q_X \| P_X) + H_Q(Y|X))$
TAC-GAN (GONG ET AL., 2019)	$\min_G \text{JS}(P_X \| Q_X) + \lambda(\text{KL}(Q_{X,Y} \| P_{X,Y}) - \text{KL}(Q_X \| P_X))$
ADC-GAN (OURS)	$\min_G \text{JS}(P_X \| Q_X) + \lambda(\text{KL}(Q_{X,Y} \| P_{X,Y}))$
PD-GAN (MIYATO & KOYAMA, 2018)	$\min_G \text{JS}(Q_{X,Y} \| P_{X,Y})$

the real joint data-label distribution. Arguably, maximization of $\log C_\text{d}(y^+|x)$ forces the generator to produce only few label-consistent data, facilitating the fidelity but losing the diversity of the generated samples. On the other hand, minimization of $\log C_\text{d}(y^-|x)$ encourages the generator to not synthesis the typically label-consistent data, increasing the diversity but may degrade the fidelity of the generated samples. In general, the two objectives together assist the generator in achieving its goal as we proved below.

Proposition 3.1. *For fixed generator, the optimal discriminative classifier of ADC-GAN has the form of the following:*

$$C_\text{d}^*(y^+|x) = \frac{p(x,y)}{p(x) + q(x)}, C_\text{d}^*(y^-|x) = \frac{q(x,y)}{p(x) + q(x)}.$$

Proposition 3.1 shows that the optimal discriminative classifier is aware of the densities of the real and generated joint distributions, therefore it is able to provide the discrepancy $\frac{p(x,y)}{q(x,y)} = \frac{C_\text{d}^*(y^+|x)}{C_\text{d}^*(y^-|x)}$ to optimize the generator.

Theorem 3.2. *Given the optimal discriminative classifier, at the equilibrium point, optimizing the classification task for the generator of ADC-GAN is equivalent to:*

$$\min_G \text{KL}(Q_{X,Y} \| P_{X,Y}). \tag{7}$$

Theorem 3.2 confirms that the discriminative classifier itself can guarantee the generator to restore the real joint distribution at the optimum. In practice, we retain the discriminator to train the generator for better training stability and convergence. The overall learning objective for the generator under the optimal discriminator and discriminative classifier is to minimize the JS divergence between the marginal data distributions and the reversed KL divergence bewteen the joint data-label distributions $(\min_G \text{JS}(P_X \| Q_X) + \lambda \cdot \text{KL}(Q_{X,Y} \| P_{X,Y}))$. Since the optimal solution set for generative modeling contains the optimal solution set for conditional generative modeling $(\arg\min_G \text{JS}(P_X \| Q_X) \supseteq \arg\min_G \text{KL}(Q_{X,Y} \| P_{X,Y}))$, the guidance to the generator provided by discriminator and discriminative classifier are harmonious, which makes ADC-GAN robust to the value of the hyperparameter λ and the selection of the GAN loss $V(G, D)$.

4. Analysis on Competing Methods

In this section, we analyze the drawbacks of the two competing methods, TAC-GAN (Gong et al., 2019) and PD-GAN (Miyato & Koyama, 2018), to show the superiority of ADC-GAN. We also analyze AM-GAN (Zhou et al., 2018) in Appendix C. Before diving into the details, we show diagrams of the discriminator and classifier of these methods in Figure 1 and summarize the theoretical learning objective for the generator under the optimal discriminator and classifier of these methods in Table 1 for an overview.

4.1. Competing Method: TAC-GAN

TAC-GAN (Gong et al., 2019) addresses the low intra-class diversity problem of AC-GAN by eliminating the conditional entropy of the generated data distribution $H_Q(Y|X)$ by learning the generator with another classifier $C_\text{mi} : \mathcal{X} \to \mathcal{Y}$, which is trained with the generated samples. The objective functions for the discriminator, the twin classifiers, and the generator of TAC-GAN are defined as follows:

$$\max_{D,C,C_\text{mi}} V(G, D) + \lambda \cdot (\mathbb{E}_{x,y \sim P_{X,Y}}[\log C(y|x)]$$
$$+ \mathbb{E}_{x,y \sim Q_{X,Y}}[\log C_\text{mi}(y|x)]), \tag{8}$$
$$\min_G V(G, D) - \lambda \cdot (\mathbb{E}_{x,y \sim Q_{X,Y}}[\log C(y|x)]$$
$$- \mathbb{E}_{x,y \sim Q_{X,Y}}[\log C_\text{mi}(y|x)]). \tag{9}$$

Theorem 4.1. *Given the twin optimal classifiers, at the equilibrium point, optimizing the classification tasks for the generator of TAC-GAN is equivalent to:*

$$\min_G \text{KL}(Q_{X,Y} \| P_{X,Y}) - \text{KL}(Q_X \| P_X). \tag{10}$$

Our Theorem 4.1 reveals that the learning objective of the generator of TAC-GAN, under the twin optimal classifiers, can be regarded as optimizing contradictory divergences, i.e., minimization between joint distributions but maximization between marginal distributions. Although theoretically the JS divergence or others (Nowozin et al., 2016; Arjovsky et al., 2017) introduced through the adversarial training between the discriminator and generator may remedy this issue, it is difficult to obtain the optimal discriminator and classifier in the practical optimization to ensure the elimination of the contradiction. We argue that the training

instability of TAC-GAN reported in the literature (Kocaoglu et al., 2018; Han et al., 2020) and found in our experiments (cf. Figures 3(a) and 5) can be explained by this analysis.

4.2. Competing Method: PD-GAN

PD-GAN (Miyato & Koyama, 2018) injects the conditional information into the projection discriminator $D_p : \mathcal{X} \times \mathcal{Y} \rightarrow [0,1]$ via the inner-product between the embedding of the label and the representation of the data to calculate the joint discriminative score of the data-label pair. In such a way, PD-GAN inherits the property of convergence point similar to the standard GAN such that it can avoid the low intra-class diversity problem of AC-GAN ideally. Specifically, the objective functions for the projection discriminator and the generator of PD-GAN are defined as follows:

$$
\min_G \max_{D_p} V(G, D_p) = \mathbb{E}_{x,y \sim P_{X,Y}}[\log D_p(x,y)]
$$
$$
+ \mathbb{E}_{x,y \sim Q_{X,Y}}[\log(1 - D_p(x,y))]. \quad (11)
$$

Based on this formulation, the optimal projection discriminator has the following form:

$$
D_p^*(x,y) = \frac{1}{1 + \exp(-d^*(x,y))} = \frac{p(x,y)}{p(x,y) + q(x,y)}
$$
$$
\Rightarrow d^*(x,y) = \log \frac{p(x,y)}{q(x,y)} = \log \frac{p(x)}{q(x)} + \log \frac{p(y|x)}{q(y|x)}, \quad (12)
$$

where $p(y|x) = \frac{\exp(\varphi^+(y) \cdot \phi(x))}{\sum_{\bar{y}} \exp(\varphi^+(\bar{y}) \cdot \phi(x))}$ and $q(y|x) = \frac{\exp(\varphi^-(y) \cdot \phi(x))}{\sum_{\bar{y}} \exp(\varphi^-(\bar{y}) \cdot \phi(x))}$. And PD-GAN accordingly defines:

$$
r(x) := \log \frac{p(x)}{q(x)} := \psi(\phi(x)),
$$

$$
r(y|x) := \log \frac{p(y|x)}{q(y|x)} := \underbrace{\overbrace{(\varphi^+(y) - \varphi^-(y))}^{\varphi(y)} \cdot \phi(x)}_{\hat{r}(y|x)} - \quad (13)
$$

$$
\underbrace{\log \sum_{\bar{y} \in \mathcal{Y}} \exp\left(\varphi^+(\bar{y}) \cdot \phi(x)\right) + \log \sum_{\bar{y} \in \mathcal{Y}} \exp\left(\varphi^-(\bar{y}) \cdot \phi(x)\right)}_{ⓐ}.
$$

However, PD-GAN actually ignores the partition term ⓐ[2] in Equation 13 and heuristically constructs the logit of the projection discriminator in the form of:

$$
d(x,y) = r(x) + \hat{r}(y|x) = \psi(\phi(x)) + \varphi(y) \cdot \phi(x). \quad (14)
$$

Discarding the partition term would make PD-GAN no longer belong to probability models that are able to model

---

[2]PD-GAN discards ⓐ in implementing the projection discriminator based on the hypothesis that ⓐ can be merged into $r(x)$. However, $r(x)$ does not model any label information, which should be involved by ⓐ. Therefore, it is unreasonable to do this.

the conditional probabilities $p(y|x)$ and $q(y|x)$, resulting in losing the complete dependencies between data and labels. Particularly, for mismatched data-label pair (x,y) with probabilities of $p(x,y) = 0$ and $q(x,y) = 0$, the projection discriminator $D_p^*(x,y) = \frac{p(x,y)}{p(x,y)+q(x,y)} = \frac{0}{0}$ is undefined and thus unreliable. Our ADC-GAN can penalize the mismatched data-label pair because $C_d^*(y^+|x) = \frac{p(x,y)}{p(x)+q(x)} = \frac{0}{>0} = 0$ ($p(x) + q(x) > 0$ for valid data x). Moreover, the optimal projection discriminator constructed according to the minimax GAN lacks theoretical guarantees on other GAN loss functions. The proposed ADC-GAN can be flexibly applied to any version of the GAN loss as we do not require a specific form of the discriminator.

5. Experiments

5.1. Synthetic Data

We first conduct experiments on a one-dimensional synthetic mixture of Gaussians, following the practices of (Gong et al., 2019), to qualitatively show the fidelity of distribution learning capability of ADC-GAN. As shown in Figure 2(a), the real data distribution consists of three classes with non-negligible overlaps. Figures 2(b) to 2(d) show the learned distributions, which are estimated by kernel density estimation (KDE) (Parzen, 1962) on the generated data of AC-GAN, TAC-GAN, and ADC-GAN without the original GAN loss $V(G, D)$, respectively. Figures 2(e) to 2(h) show the KDE results of PD-GAN, AC-GAN, TAC-GAN, and ADC-GAN trained with the non-saturating GAN loss (Goodfellow et al., 2014), respectively. AC-GAN tends to generate classifiable data so that it decreases the intra-class diversity. Without the GAN loss $V(G, D)$, AC-GAN outputs nearly deterministic data for each class. TAC-GAN without the GAN loss also cannot accurately capture the real data distribution, verifying the contradiction in Theorem 4.1. Impressively, the proposed ADC-GAN faithfully restores the real data distribution even without the GAN loss, validating Theorem 3.2 that the discriminative classfier alone can guide the generator to learn the real data distribution.

5.2. Experiments based on BigGAN-PyTorch

In this section, we conduct experiments on three common real-world datasets: CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), and Tiny-ImageNet (Le & Yang, 2015) based on the BigGAN-PyTorch repository[3] with our extensions[4]. The optimizer is Adam with learning rate of 2×10^{-4} on CIFAR-10/100 and 1×10^{-4} for the generator and 4×10^{-4} for the discriminator on Tiny-ImageNet. We train all methods for 1000 and 500 epochs with batch size of 50 and 100

---

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

(a) Real Data (b) AC-GAN w/o $V(G, D)$ (c) TAC-GAN w/o $V(G, D)$ (d) ADC-GAN w/o $V(G, D)$

(e) PD-GAN (f) AC-GAN w/ $V(G, D)$ (g) TAC-GAN w/ $V(G, D)$ (h) ADC-GAN w/ $V(G, D)$
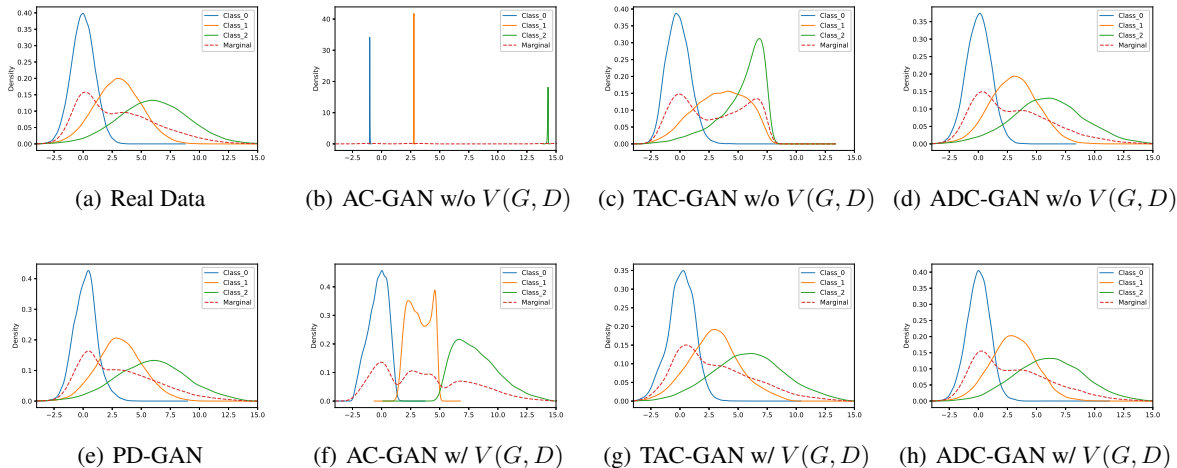
Figure 2: Qualitative comparison of distribution modeling results on the one-dimensional synthetic data.

Table 2: FID and Intra-FID and Accuracy (%) comparisons on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively.

DATASETS	METRICS	PD-GAN	AC-GAN	AM-GAN	TAC-GAN	ADC-GAN
	FID ($\downarrow$)	6.23	6.50	6.81	5.83	**5.66**
CIFAR-10	INTRA-FID ($\downarrow$)	48.90	57.67	69.31	56.67	**40.45**
	ACCURACY ($\uparrow$)	66.22	84.69	83.63	88.27	**89.51**
	FID ($\downarrow$)	8.70	11.24	10.42	10.38	**8.12**
CIFAR-100	INTRA-FID ($\downarrow$)	51.15	83.06	78.11	79.59	**49.24**
	ACCURACY ($\uparrow$)	37.89	55.26	55.77	60.03	**64.24**
	FID ($\downarrow$)	26.10	25.02	21.34	21.12	**19.02**
TINY-IMAGENET	INTRA-FID ($\downarrow$)	66.23	99.04	90.56	95.48	**63.05**
	ACCURACY ($\uparrow$)	27.79	44.59	44.67	44.44	**48.89**

on CIFAR-10/100 and Tiny-ImageNet, respectively. The discriminator/classifier are updated 4 and 2 times per generator update step on CIFAR-10/100 and Tiny-ImageNet, respectively. We follow the practice of (Miyato & Koyama, 2018; Gong et al., 2019) to adopt the hinge loss (Lim & Ye, 2017; Tran et al., 2017) as the implementation of $V(G, D)$. The coefficient hyperparameters of AC-GAN and AM-GAN (Zhou et al., 2018) (cf. Appendix C for analysis) are set as $\lambda = 0.2$ as it performs the best. As for TAC-GAN and ADC-GAN, the coefficient hyperparameters are set as $\lambda = 1.0$ on CIFAR-10/100 and $\lambda = 0.5$ on Tiny-ImageNet.

Image Generation. We use the Fréchet Inception Distance (FID) (Heusel et al., 2017) and Intra-FID (Miyato & Koyama, 2018) metrics to measure the overall and intra-class qualities of the generated images, respectively. Table 2 shows that ADC-GAN obtains the best FID and Intra-FID scores on all three datasets, indicating consistent superiority over previous cGANs in conditional image generation.

Training Stability. We also note that ADC-GAN yields the best training stability according to the FID training curves (cf. Figures 3(a) and 5). Even without the discriminator, the training stability ADC-GAN (w/o D) still exceeds that of most competing methods. AC-GAN diverges during training on all three datasets. TAC-GAN also diverges on CIFAR-100 and Tiny-ImageNet and achieves a relatively stable FID training curve only on the simplest dataset, CIFAR-10. We hence report the results of all methods using the best checkpoint. These unstable FID training curves implicitly verify the drawback of existing classifier-based cGANs that optimize contradictory divergences.

Different Coefficients. To explicitly show the above issues, we set the objective function of classifier-based cGANs as $(1 - \lambda')V(G, D) + \lambda'V_{\mathrm{C}}(G, C)$, where $V_{\mathrm{C}}(G, C)$ is the task between the generator and classifier. As shown in Figures 3(b) and 6, ADC-GAN consistently gains superior FID scores across different coefficient hyperparameters even for
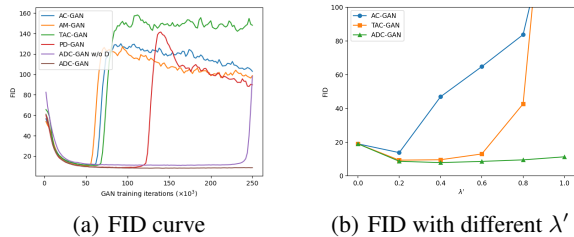
(a) FID curve

(b) FID with different λ'

Figure 3: (a) FID curves during GAN training on CIFAR-100. (b) FID scores of classifier-based cGANs with different λ' on CIFAR-100. The objective function in this experiment is $(1 - \lambda')V(G, D) + \lambda'V_\mathrm{C}(G, C)$, where $V_\mathrm{C}(G, C)$ is the task between the generator and classifier.
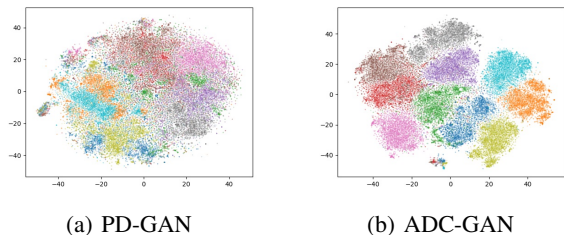

(a) PD-GAN

(b) ADC-GAN

Figure 4: T-SNE visualization of CIFAR-10 validation data based on learned representations extracted from the penultimate layer in the discriminator/classifier $\phi(x)$. Different colors indicate different classes.

$\lambda' = 1.0$ (i.e., without the discriminator), showing strong robustness with respect to λ', while AC-GAN and TAC-GAN perform substantially worse when λ' becomes larger.

Data-to-Class Relations. To investigate whether the model captures appropriate data-to-class relations, we conduct image classification experiments based on the learned representations of the discriminator/classifier $\phi(x)$. Specifically, we first train a logistic regression classifier using the scikit-learn library with the training data and compute the classification accuracy of the validation data. As reported in Table 2, ADC-GAN significantly outperforms competing methods on all datasets in terms of the Accuracy metrics. The reason is that the discriminative classifier needs to recognize the labels of data while simultaneously distinguishing between real and fake data, which facilitates the robustness of the classifier in modeling data-to-class relations. Notice that PD-GAN obtains the worst results. By comparing the CIFAR-10 T-SNE (Van der Maaten & Hinton, 2008) visualization results of PD-GAN and ADC-GAN in Figure 4, it is clear that PD-GAN does not have the ability to learn proper data-to-class relations as ADC-GAN does, reflecting the problem caused by the loss of partition terms in PD-GAN.

Table 3: FID and IS comparisons on ImageNet (128×128). B.S. means the batch size and Iters. means the training iterations. Results of BigGAN and ReACGAN are copied from the ReACGAN paper (Kang et al., 2021).

B.S.	Iters.	Methods	IS ($\uparrow$)	FID ($\downarrow$)
256	500K	BigGAN	43.97	16.36
		ReACGAN	**68.27**	13.98
		ADC-GAN	66.96	**11.65**
2048	200K	BigGAN	99.71	**7.89**
		ReACGAN	92.74	8.23
		ADC-GAN	97.47	9.46
	500K	ADC-GAN	**108**.10	8.02

5.3. Experiments based on PyTorch-StudioGAN

In this section, we compare ADC-GAN with state-of-the-art cGANs using the PyTorch-StudioGAN repository[5], of which evaluation protocols are different from that of the BigGAN-PyTorch repository that we used in Table 2. Nonetheless, our comparison is fair because the methods in each experiment follows the same evaluation protocol.

Image Generation on ImageNet. We first conduct experiments on ImageNet (128×128) following the experimental settings of ReACGAN (Kang et al., 2021). Table 3 reports the Inception Score (IS) (Salimans et al., 2016) and FID results. Our ADC-GAN is comparable with the state-of-the-art cGANs, BigGAN and ReACGAN (Kang et al., 2021), in the batch size of 256 and 2048, showing effectiveness on large-scale high-resolution image datasets. Notice that, however, we only ran our ADC-GAN once with $\lambda = 1$ in each of the two batch size settings, and did not make other attempts due to our limited computational resources. We argue that the results of ADC-GAN can be improved by choosing an appropriate coefficient hyperparameter λ.

Different GAN Losses. We also investigate the robustness of ADC-GAN with respect to the GAN loss function $V(G, D)$ by adopting different versions. Table 4 report the qualitative results on CIFAR-100 (cf. Table 5 in Appendix D for complete results). Impressively, the proposed ADC-GAN achieves the best iFID (intra-FID), recall (Kynkäänniemi et al., 2019), and coverage (Naeem et al., 2020) scores across the non-saturation (Goodfellow et al., 2014), WGAN-GP (Gulrajani et al., 2017), and hinge (Lim & Ye, 2017) versions of the GAN loss. The best iFID scores indicate the best conditional generative modeling performance, and the best recall and coverage results reflect the best (intra-class) diversity of the generated samples.

---

[5] https://github.com/POSTECH-CVLab/PyTorch-StudioGAN

Table 4: IS, FID, iFID, Precision, Recall, Density, and Coverage comparisons with state-of-the-art methods under different GAN loss functions on CIFAR-100, respectively. The best results are bold and the second best are underlined.

GAN Loss	Methods	IS ↑	FID ↓	iFID ↓	Precision ↑	Recall ↑	Density ↑	Coverage ↑
Non-saturation	PD-GAN	11.48	11.59	105.38	0.7337	0.6804	0.8646	0.8513
	AC-GAN	7.98	49.46	207.56	0.7322	0.0793	0.6225	0.4112
	TAC-GAN	11.34	14.47	131.90	0.7429	0.6077	0.8324	0.7887
	ADC-GAN	11.88	11.07	104.21	0.7379	0.6972	0.8521	0.8609
	ContraGAN	11.15	13.54	146.86	0.7390	0.6155	0.8481	0.7729
	ReACGAN	11.79	13.72	125.21	0.7541	0.5861	0.8695	0.8005
W-GP	PD-GAN	5.66	69.48	–	0.5976	0.1603	0.4310	0.2649
	AC-GAN	10.97	19.30	148.40	0.6880	0.5444	0.6770	0.7242
	TAC-GAN	11.04	15.56	121.23	0.7023	0.6474	0.7048	0.7535
	ADC-GAN	11.01	14.02	101.14	0.7058	0.6804	0.7549	0.7956
	ContraGAN	6.72	49.77	147.22	0.6498	0.2834	0.5827	0.3549
	ReACGAN	6.67	47.74	150.7	0.6188	0.3104	0.4806	0.3396
Hinge	PD-GAN	11.76	10.96	108.08	0.7436	0.6812	0.8790	0.8609
	AC-GAN	11.66	21.65	168.87	0.7577	0.3649	0.8297	0.7225
	TAC-GAN	12.07	12.56	134.75	0.7572	0.6020	0.8957	0.8400
	ADC-GAN	11.82	10.73	103.78	0.7387	0.7023	0.8721	0.8707
	ContraGAN	10.08	13.22	128.50	0.7372	0.6251	0.8356	0.7790
	ReACGAN	11.80	12.52	140.47	0.7510	0.5982	0.9300	0.8327

6. Related Work

Efforts on developing cGANs (Mirza & Osindero, 2014) can be divided into two steps. The first is to study how to implement a conditional generator. Methods in this category are concatenation (Mirza & Osindero, 2014), conditional batch normalization (de Vries et al., 2017), and conditional convolution layers (Sagong et al., 2019). The second is to study how to train the conditional generator to produce label-dependent samples, which can be further divided into two categories, classifier-based and projection-based cGANs.

Classifier-based cGANs. AC-GAN (Odena et al., 2017) leveraged an auxiliary classifier to identify consistency between data and labels. MH-GAN (Kavalerov et al., 2021) improved AC-GAN by replacing the cross-entropy loss of the classifier with the multi-hinge loss. AM-GAN (Zhou et al., 2018) replaced the discriminator with a $K + 1$-way classifier with an additional "fake" label. Omni-GAN (Zhou et al., 2020) combined the discriminator with the classifier to construct a $K + 2$-dimensional multi-label classifier. TAC-GAN (Gong et al., 2019) corrected the biased learning objective of AC-GAN by introducing another classifier, which is the multi-class version of Anti-Labeler of CausalGAN (Kocaoglu et al., 2018). UAC-GAN (Han et al., 2020) improved the training stability of TAC-GAN with MINE (Belghazi et al., 2018). ECGAN (Chen et al., 2021) provides a unified view of cGANs with and without classifiers. Orthogonally to our work, ContraGAN (Kang & Park, 2020) and ReACGAN (Kang et al., 2021) modeled data-to-data relations as well as data-to-class relations using the conditional contrastive loss and the data-to-data cross-entropy loss, respectively. However, they did not solve

the low intra-class diversity problem of AC-GAN as they inherited the generator-agnostic classifier.

Projection-based cGANs. PD-GAN (Miyato & Koyama, 2018) injected the class information into the discriminator via label projection and achieved the state-of-the-art generation quality of natural images (Brock et al., 2019; Wu et al., 2019; Zhang et al., 2020; Zhao et al., 2021). P2GAN (Han et al., 2021) further improved PD-GAN by compensating the missed partition term in the objective function.

Discriminative classifiers. Watanabe & Favaro (2021) exploited the discriminative classifier for training GANs with any level of labeling but different from us with the objective function for the generator, which enables ADC-GAN to faithfully learn the target distribution. SSGAN-LA (Hou et al., 2021a) presented the similar idea but different loss functions with ADC-GAN (multi-hinge v.s. cross-entropy) to tackle the degraded learning objective of self-supervised GANs, while ADC-GAN is for conditional GANs. Moreover, our analysis of the degradation objective is more accurate and informative than that of SSGAN-LA.

7. Conclusion

In this paper, we present a novel conditional generative adversarial network with an auxiliary discriminative classifier (ADC-GAN) to achieve faithful conditional generative modeling. We also discuss the differences between ADC-GAN with competing cGANs and analyze their potential issues and limitations. Extensive experimental results validate the theoretical superiority of ADC-GAN compared with state-of-the-art classifier-based and projection-based cGANs.

Acknowledgements

References

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

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

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

Che, T., Li, Y., Jacob, A. P., Bengio, Y., and Li, W. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

Chen, S.-A., Li, C.-L., and Lin, H.-T. A unified view of cGANs with and without classifiers. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

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

de Vries, H., Strub, F., Mary, J., Larochelle, H., Pietquin, O., and Courville, A. C. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, 2017.

Gong, M., Xu, Y., Li, C., Zhang, K., and Batmanghelich, K. Twin auxiliary classifiers gan. In *Advances in Neural Information Processing Systems*, 2019.

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

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

Han, L., Stathopoulos, A., Xue, T., and Metaxas, D. Unbiased auxiliary classifier gans with mine. *arXiv preprint arXiv:2006.07567*, 2020.

Han, L., Min, M. R., Stathopoulos, A., Tian, Y., Gao, R., Kadav, A., and Metaxas, D. N. Dual projection generative adversarial networks for conditional image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021.

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

Hou, L., Shen, H., Cao, Q., and Cheng, X. Self-supervised GANs with label augmentation. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021a.

Hou, L., Yuan, Z., Huang, L., Shen, H., Cheng, X., and Wang, C. Slimmable generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7746–7753, 2021b.

Kang, M. and Park, J. Contragan: Contrastive learning for conditional image generation. In *Advances in Neural Information Processing Systems*, 2020.

Kang, M., Shim, W. J., Cho, M., and Park, J. Rebooting ACGAN: Auxiliary classifier GANs with stable training. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. Alias-free generative adversarial networks. In *Advances in Neural Information Processing Systems*.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training generative adversarial networks with limited data. 2020a.

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

Kavalerov, I., Czaja, W., and Chellappa, R. A multi-class hinge loss for conditional gans. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1290–1299, January 2021.

Kocaoglu, M., Snyder, C., Dimakis, A. G., and Vishwanath, S. CausalGAN: Learning causal implicit generative models with adversarial training. In *International Conference on Learning Representations*, 2018.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. In *Advances in Neural Information Processing Systems*, 2019.

Le, Y. and Yang, X. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

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

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

Mirza, M. and Osindero, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Miyato, T. and Koyama, M. cGANs with projection discriminator. In *International Conference on Learning Representations*, 2018.

Naeem, M. F., Oh, S. J., Uh, Y., Choi, Y., and Yoo, J. Reliable fidelity and diversity metrics for generative models. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

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

Odena, A. Semi-supervised learning with generative adversarial networks. *arXiv preprint arXiv:1606.01583*, 2016.

Odena, A., Olah, C., and Shlens, J. Conditional image synthesis with auxiliary classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Parzen, E. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3): 1065–1076, 1962.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252, 2015.

Sagong, M.-C., Shin, Y.-G., Yeo, Y.-J., Park, S., and Ko, S.-J. cgans with conditional convolution layer. *arXiv preprint arXiv:1906.00709*, 2019.

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

Shu, R., Bui, H., and Ermon, S. Ac-gan learns a biased distribution. In *NIPS Workshop on Bayesian Deep Learning*, volume 8, 2017.

Tan, Z., Chai, M., Chen, D., Liao, J., Chu, Q., Yuan, L., Tulyakov, S., and Yu, N. Michigan: Multi-input-conditioned hair image generation for portrait editing. *arXiv preprint arXiv:2010.16417*, 2020.

Tran, D., Ranganath, R., and Blei, D. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, 2017.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

Watanabe, T. and Favaro, P. A unified generative adversarial network training via self-labeling and self-attention. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Wu, Y., Donahue, J., Balduzzi, D., Simonyan, K., and Lillicrap, T. Logan: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2image: Conditional image generation from visual attributes. *arXiv preprint arXiv:1512.00570*, 2015.

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

Zhao, Z., Singh, S., Lee, H., Zhang, Z., Odena, A., and Zhang, H. Improved consistency regularization for gans. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11033–11041, 2021.

Zhou, P., Xie, L., Ni, B., Geng, C., and Tian, Q. Omnigan: On the secrets of cgans and beyond. *arXiv preprint arXiv:2011.13074*, 2020.

Zhou, Z., Cai, H., Rong, S., Song, Y., Ren, K., Zhang, W., Wang, J., and Yu, Y. Activation maximization generative adversarial nets. In *International Conference on Learning Representations*, 2018.

A. Proofs

A.1. Proof of Proposition 2.1

Proposition 2.1. *For fixed generator, the optimal classifier of AC-GAN has the form of $C^*(y|x) = \frac{p(x,y)}{p(x)}$.*

Proof.

$$\max_C \mathbb{E}_{x,y \sim P_{X,Y}}[\log C(y|x)] = \mathbb{E}_{x \sim P_X}\mathbb{E}_{y \sim P_{Y|X}}[\log C(y|x)] \tag{15}$$

$$\Rightarrow \min_C \mathbb{E}_{x \sim P_X}\mathbb{E}_{y \sim P_{Y|X}}[-\log C(y|x)] = \mathbb{E}_{x \sim P_X}[H(p(y|x)) + \mathrm{KL}(p(y|x)\|C(y|x))] \tag{16}$$

$$\Rightarrow C^*(y|x) = \arg\min_C \mathrm{KL}(p(y|x)\|C(y|x)) = p(y|x) = \frac{p(x,y)}{p(x)} \tag{17}$$

$\square$

A.2. Proof of Theorem 2.2

Theorem 2.2. *Given the optimal classifier, at the equilibrium point, optimizing the classification task for the generator of AC-GAN is equivalent to:*

$$\min_G \mathrm{KL}(Q_{X,Y}\|P_{X,Y}) - \mathrm{KL}(Q_X\|P_X) + H_Q(Y|X), \tag{4}$$

where $H_Q(Y|X) = -\int \sum_y q(x,y) \log q(y|x)\mathrm{d}x$ is the conditional entropy of the generated samples.

Proof.

$$\max_G \mathbb{E}_{x,y \sim Q_{X,Y}}[\log C^*(y|x)] = \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{p(x,y)}{p(x)}\right] = \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{p(x,y)}{q(x,y)}\frac{q(x)}{p(x)}\frac{q(x,y)}{q(x)}\right] \tag{18}$$

$$\Rightarrow \min_G \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{p(x,y)}\right] - \mathbb{E}_{x \sim Q_X}\left[\log \frac{q(x)}{p(x)}\right] - \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{q(x)}\right] \tag{19}$$

$$\Rightarrow \min_G \mathrm{KL}(Q_{X,Y}\|P_{X,Y}) - \mathrm{KL}(Q_X\|P_X) + H_Q(Y|X) \tag{20}$$

$\square$

A.3. Proof of Proposition 3.1

Proposition 3.1. *For fixed generator, the optimal discriminative classifier of ADC-GAN has the form of the following:*

$$C_{\mathrm{d}}^*(y^+|x) = \frac{p(x,y)}{p(x) + q(x)}, C_{\mathrm{d}}^*(y^-|x) = \frac{q(x,y)}{p(x) + q(x)}.$$

Proof.

$$\max_{C_{\mathrm{d}}} \mathbb{E}_{x,y \sim P_{X,Y}}[\log C_{\mathrm{d}}(y^+|x)] + \mathbb{E}_{x,y \sim Q_{X,Y}}[\log C_{\mathrm{d}}(y^-|x)] \Rightarrow \max_{C_{\mathrm{d}}} \mathbb{E}_{x,y \sim P_{X,Y}^m}[\log C_{\mathrm{d}}(y|x)], \tag{21}$$

with $p^m(x,y^+) = \frac{1}{2}p(x,y)$, $p^m(x,y^-) = \frac{1}{2}q(x,y)$, and $p^m(x) = \sum_y p^m(x,y) = \frac{1}{2}p(x) + \frac{1}{2}q(x)$.

$$\Rightarrow \max_{C_{\mathrm{d}}} \mathbb{E}_{x \sim P_X^m}\mathbb{E}_{y \sim P_{Y|X}^m}[\log C_{\mathrm{d}}(y|x)] \Rightarrow \min_{C_{\mathrm{d}}} \mathbb{E}_{x \sim P_X^m}\mathbb{E}_{y \sim P_{Y|X}^m}[-\log C_{\mathrm{d}}(y|x)] \tag{22}$$

$$\Rightarrow \min_{C_{\mathrm{d}}} \mathbb{E}_{x \sim P_X^m}[H(p^m(y|x)) + \mathrm{KL}(p^m(y|x)\|C_{\mathrm{d}}(y|x))] \tag{23}$$

$$\Rightarrow C_{\mathrm{d}}^*(y|x) = \arg\min_{C_{\mathrm{d}}} \mathrm{KL}(p^m(y|x)\|C_{\mathrm{d}}(y|x)) = p^m(y|x) = \frac{p^m(x,y)}{p^m(x)} \tag{24}$$

Therefore, the optimal discriminative classifier of ADC-GAN has the form of $C_{\mathrm{d}}^*(y^+|x) = \frac{p^m(x,y^+)}{p^m(x)} = \frac{p(x,y)}{p(x)+q(x)}$ and $C_{\mathrm{d}}^*(y^-|x) = \frac{p^m(x,y^-)}{p^m(x)} = \frac{q(x,y)}{p(x)+q(x)}$ that conclude the proof.

$\square$

A.4. Proof of Theorem 3.2

Theorem 3.2. *Given the optimal discriminative classifier, at the equilibrium point, optimizing the classification task for the generator of ADC-GAN is equivalent to:*

$$\min_{G} \mathrm{KL}(Q_{X,Y} \| P_{X,Y}). \tag{7}$$

Proof.

$$\max_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log C_{\mathrm{d}}^{*}(y^{+}|x)\right] - \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log C_{\mathrm{d}}^{*}(y^{-}|x)\right] \tag{25}$$

$$\Rightarrow \max_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{p(x,y)}{p(x) + q(x)}\right] - \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{q(x,y)}{p(x) + q(x)}\right] \tag{26}$$

$$\Rightarrow \min_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{q(x,y)}{p(x,y)}\right] \Rightarrow \min_{G} \mathrm{KL}(Q_{X,Y} \| P_{X,Y}) \tag{27}$$

$\square$

A.5. Proof of Theorem 4.1

Proposition A.1. *For fixed generator, the twin optimal classifiers of TAC-GAN have the following forms:*

$$C^{*}(y|x) = \frac{p(x,y)}{p(x)}, C_{\mathrm{mi}}^{*}(y|x) = \frac{q(x,y)}{q(x)}. \tag{28}$$

Proof. The proof is similar to that of Proposition 2.1 in Appendix A.1 by considering C and C_{mi} as two independent classifiers with respect to distribution P and Q, respectively. $\square$

Theorem 4.1. *Given the twin optimal classifiers, at the equilibrium point, optimizing the classification tasks for the generator of TAC-GAN is equivalent to:*

$$\min_{G} \mathrm{KL}(Q_{X,Y} \| P_{X,Y}) - \mathrm{KL}(Q_{X} \| P_{X}). \tag{10}$$

Proof.

$$\max_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} [\log C^{*}(y|x)] - \mathbb{E}_{x,y \sim Q_{X,Y}} [\log C_{\mathrm{mi}}^{*}(y|x)] \tag{29}$$

$$\Rightarrow \max_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{p(x,y)}{p(x)}\right] - \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{q(x,y)}{q(x)}\right] \tag{30}$$

$$\Rightarrow \max_{G} \mathbb{E}_{x,y \sim Q_{X,Y}} \left[\log \frac{p(x,y)}{q(x,y)}\right] - \mathbb{E}_{x \sim Q_{X}} \left[\log \frac{p(x)}{q(x)}\right] \tag{31}$$

$$\Rightarrow \min_{G} \mathrm{KL}(Q_{X,Y} \| P_{X,Y}) - \mathrm{KL}(Q_{X} \| P_{X}) \tag{32}$$

$\square$

B. Analysis on the Original AC-GAN

In this section, we show that original AC-GAN whose auxiliary classifier is trained with both real and generated samples still suffers from the same issue as we proved in Theorem 2.2. Formally, the full objective function of the original AC-GAN is formulated as the following:

$$\max_{D,C} V(G, D) + \lambda \cdot \left(\mathbb{E}_{x,y \sim P_{X,Y}} [\log C(y|x)] + \mathbb{E}_{x,y \sim Q_{X,Y}} [\log C(y|x)]\right), \tag{33}$$

$$\min_{G} V(G, D) - \lambda \cdot \left(\mathbb{E}_{x,y \sim Q_{X,Y}} [\log C(y|x)]\right). \tag{34}$$

The objective function for training the classifier can be rewritten as:

$$\max_{C} \mathbb{E}_{x,y \sim P_{X,Y}} [\log C(y|x)] + \mathbb{E}_{x,y \sim Q_{X,Y}} [\log C(y|x)] \Rightarrow \max_{C} \mathbb{E}_{x,y \sim P_{X,Y}^{m}} [\log C(y|x)], \tag{35}$$

with $p^m(x,y) = \frac{1}{2}(p(x,y) + q(x,y))$ and $p^m(x) = \sum_y p^m(x,y) = \frac{1}{2}(p(x) + q(x))$. And we can obtain the optimal classifier according to the following:

$$\max_C \mathbb{E}_{x,y \sim P^m_{X,Y}}[\log C(y|x)] \Rightarrow \min_C \mathbb{E}_{x \sim P^m_X, y \sim P^m_{Y|X}}[-\log C(y|x)] \tag{36}$$

$$\Rightarrow \min_C \mathbb{E}_{x \sim P^m_X}[H(p^m(y|x)) + \mathrm{KL}(p^m(y|x)\|C(y|x))] \tag{37}$$

$$\Rightarrow C^*(y|x) = p^m(y|x) = \frac{p(x,y) + q(x,y)}{p(x) + q(x)}. \tag{38}$$

Suppose that the conditional generator learns the joint distribution of real data and labels, i.e., $q(x,y) = p(x,y)$ and $q(x) = p(x)$, the optimal classifier $C^*(y|x) = \frac{p(x,y)+q(x,y)}{p(x)+q(x)} = \frac{p(x,y)}{p(x)}$ also provide the objective stated in Theorem 2.2 for the generator, which contains the conditional entropy of the generated samples $H_Q(Y|X)$ that reduces the intra-class diversity of the generated samples. In other words, the original classifier does not allow the generator to remain on the desired distribution because it still provides momentum to update the generator, resulting in a biased learning objective for the generator in the original version of AC-GAN. The essential reason is that the classifier of the original AC-GAN is incapable of distinguishing the real data from the generated data. Therefore, the classifier of the original AC-GAN cannot provide the difference between the real and generated joint distributions to optimize the generator.

C. Analysis on AM-GAN

AM-GAN (Zhou et al., 2018) optimizes the following objectives with an label-extended discriminator $D_+ : \mathcal{X} \to \mathcal{Y} \cup \{0\}$:

$$\max_{D_+} \mathbb{E}_{x,y \sim P_{X,Y}}[\log D_+(y|x)] + \mathbb{E}_{x,y \sim Q_{X,Y}}[\log D_+(0|x)], \tag{39}$$

$$\min_G \mathbb{E}_{x,y \sim Q_{X,Y}}[\log D_+(y|x)]. \tag{40}$$

The objective function for training the discriminator D_+ can be rewritten as:

$$\max_{D_+} \mathbb{E}_{x,y \sim P_{X,Y}}[\log D_+(y|x)] + \mathbb{E}_{x,y \sim Q_{X,Y}}[\log D_+(0|x)] \Rightarrow \max_{D_+} \mathbb{E}_{x,y \sim P^m_{X,Y}}[\log D_+(y|x)], \tag{41}$$

where $p^m(x,y) = \frac{1}{2}p(x,y), \forall y \in \mathcal{Y}, p^m(x,0) = \frac{1}{2}q(x)$, and $p^m(x) = \sum_y p^m(x,y) = \frac{1}{2}(p(x) + q(x))$. Then we have:

$$\max_{D_+} \mathbb{E}_{x,y \sim P^m_{X,Y}}[\log D_+(y|x)] \Rightarrow \min_{D_+} \mathbb{E}_{x \sim P^m_X, y \sim P^m_{Y|X}}[-\log D_+(y|x)] \tag{42}$$

$$\Rightarrow \min_{D_+} \mathbb{E}_{x \sim P^m_X}[H(p^m(y|x)) + \mathrm{KL}(p^m(y|x)\|D^+(y|x))] \Rightarrow D^*_+(y|x) = p^m(y|x) = \frac{p(x,y)}{p(x) + q(x)}, \forall y \in \mathcal{Y}. \tag{43}$$

Under the optimal discriminator D^*_+, the generator of AM-GAN can be regarded as optimizing the following:

$$\max_G \mathbb{E}_{x,y \sim Q_{X,Y}}[\log D^*_+(y|x)] \Rightarrow \max_G \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{p(x,y)}{p(x) + q(x)}\right] \tag{44}$$

$$\Rightarrow \min_G \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{p(x,y)} \frac{p(x) + q(x)}{q(x,y)}\right] = \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{p(x,y)} + \log \frac{p(x) + q(x)}{2} - \log q(x,y) + \log 2\right] \tag{45}$$

$$\geq \min_G \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{p(x,y)} + \frac{1}{2}\log p(x) + \frac{1}{2}\log q(x) - \log q(x,y) + \log 2\right] \tag{46}$$

$$\Rightarrow \min_G \mathbb{E}_{x,y \sim Q_{X,Y}}\left[\log \frac{q(x,y)}{p(x,y)} - \frac{1}{2}\log \frac{q(x)}{p(x)} - \log \frac{q(x,y)}{q(x)} + \log 2\right] \tag{47}$$

$$\Rightarrow \min_G \mathrm{KL}(Q_{X,Y}\|P_{X,Y}) - \frac{1}{2}\mathrm{KL}(Q_X\|P_X) + H_Q(Y|X) + \log 2. \tag{48}$$

In summary, AM-GAN with the original discriminator remained (compared in our experiments) can be considered to be minimizing an upper bound of $\mathrm{JS}(Q_X\|P_X) + \mathrm{KL}(Q_{X,Y}\|P_{X,Y}) - \frac{1}{2}\mathrm{KL}(Q_X\|P_X) + H_Q(Y|X) + \log 2$.

D. More Results

(a) FID curves on CIFAR-10

(b) FID curves on Tiny-ImageNet
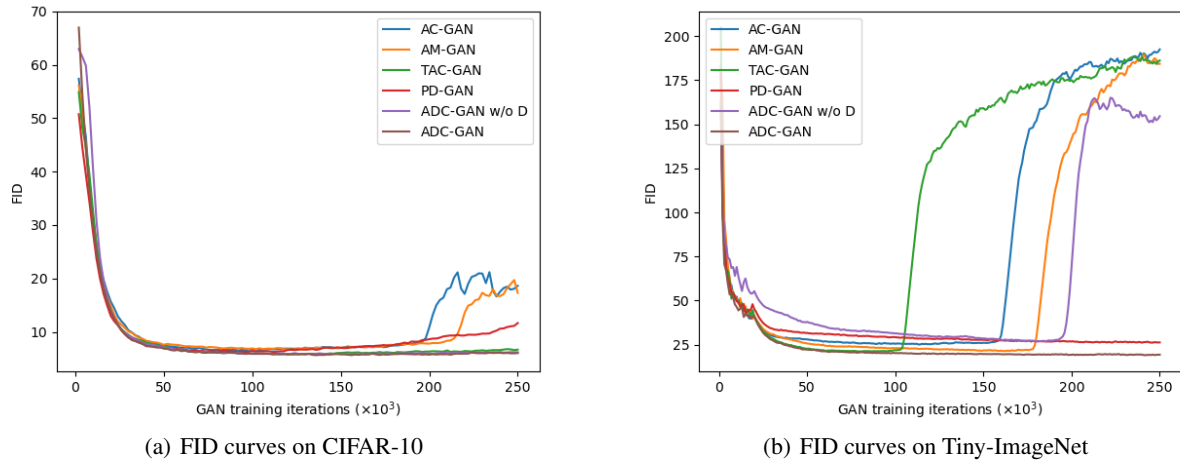
Figure 5: FID curves during GAN training on CIFAR-10 and Tiny-ImageNet, respectively.

(a) FID with different λ' on CIFAR-10

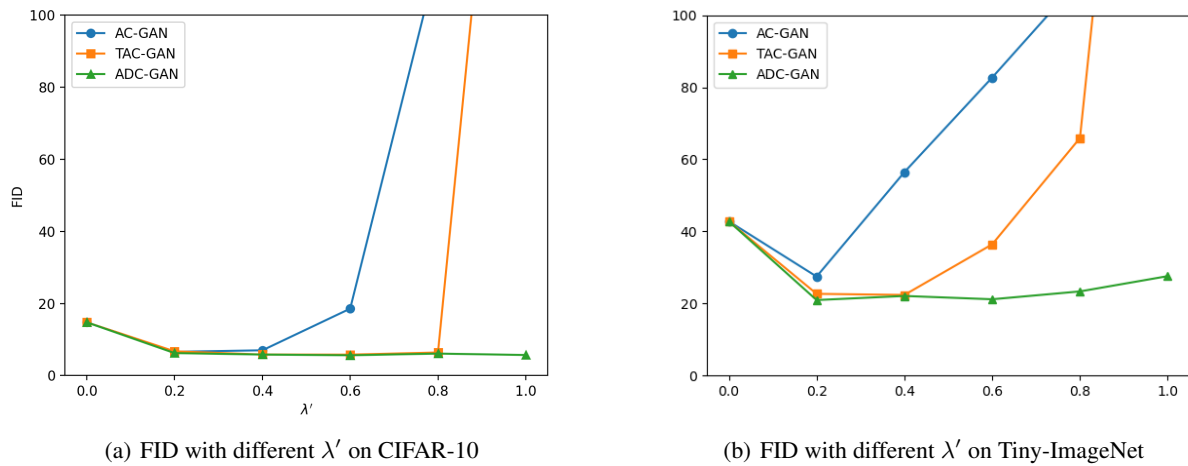(b) FID with different λ' on Tiny-ImageNet

Figure 6: FID comparisons of classifier-based cGANs with different coefficient hyperparameters λ' on CIFAR-10 and Tiny-ImageNet, respectively. The objective function in this experiment is $(1 - \lambda')V(G, D) + \lambda'V_{\mathrm{C}}(G, C)$, where $V_{\mathrm{C}}(G, C)$ is the task between the generator and classifier.

Table 5: IS, FID, iFID, Precision, Recall, Density, and Coverage comparisons of competing methods under different GAN loss functions on CIFAR-10 and CIFAR-100, respectively. The best results are bold and the second best are underlined.

CIFAR-10	METHODS	IS ↑	FID ↓	iFID ↓	PRECISION ↑	RECALL ↑	DENSITY ↑	COVERAGE ↑
NON-SATURATION	PD-GAN	9.68	8.93	81.30	0.7581	<u>0.6718</u>	**1.0622**	<u>0.9208</u>
	AC-GAN	<u>9.74</u>	9.21	87.76	0.7592	0.6484	1.0491	0.9147
	TAC-GAN	9.61	9.31	<u>81.04</u>	0.7349	0.6717	0.9575	0.8990
	ADC-GAN	**9.87**	8.47	**77.69**	0.7497	**0.6912**	0.9968	0.9202
	ContraGAN	9.60	8.87	120.45	<u>0.7598</u>	0.6595	1.0025	0.9061
	ReACGAN	9.69	<u>8.51</u>	113.23	**0.7648**	0.6594	<u>1.0532</u>	**0.9242**
LEAST SQUARE	PD-GAN	**9.99**	<u>8.72</u>	<u>80.11</u>	0.7525	<u>0.6771</u>	<u>1.0395</u>	0.9182
	AC-GAN	5.01	81.93	176.24	0.7389	0.0037	0.7484	0.2129
	TAC-GAN	9.41	10.67	80.92	0.7386	0.6520	0.9159	0.8657
	ADC-GAN	9.89	**8.61**	**75.86**	0.7405	**0.6919**	0.9944	**0.9223**
	ContraGAN	9.10	12.93	135.75	<u>0.7661</u>	0.5761	1.0236	0.8262
	ReACGAN	<u>9.80</u>	9.52	125.83	**0.7772**	0.5988	**1.1008**	0.9138
W-GP	PD-GAN	5.27	75.24	104.15	0.5569	0.2132	0.3678	0.2141
	AC-GAN	8.88	14.77	88.02	**0.7015**	0.6477	0.7421	0.7798
	TAC-GAN	<u>8.93</u>	<u>13.26</u>	<u>76.93</u>	0.6847	<u>0.6705</u>	<u>0.7454</u>	<u>0.8127</u>
	ADC-GAN	**9.49**	**11.25**	**74.98**	<u>0.6996</u>	**0.7019**	**0.8182**	**0.8517**
	ContraGAN	6.38	51.43	137.17	0.5640	0.3995	0.4040	0.2931
	ReACGAN	6.60	44.62	117.25	0.5813	0.4333	0.4559	0.3287
HINGE	PD-GAN	9.79	<u>8.45</u>	79.40	0.7464	<u>0.6853</u>	1.0083	0.9158
	AC-GAN	**9.96**	8.97	88.40	**0.7681**	0.6523	<u>1.0250</u>	<u>0.9168</u>
	TAC-GAN	9.78	8.80	81.30	0.7446	0.6749	1.0026	0.9103
	ADC-GAN	9.63	**8.42**	75.50	0.7447	**0.6882**	0.9854	**0.9193**
	ContraGAN	9.63	8.89	85.39	0.7582	0.6538	**1.0411**	0.9098
	ReACGAN	<u>9.83</u>	8.84	<u>78.07</u>	<u>0.7623</u>	0.6675	1.0003	0.9158
CIFAR-100	METHODS	IS ↑	FID ↓	iFID ↓	PRECISION ↑	RECALL ↑	DENSITY ↑	COVERAGE ↑
NON-SATURATION	PD-GAN	11.48	<u>11.59</u>	<u>105.38</u>	0.7337	<u>0.6804</u>	<u>0.8646</u>	<u>0.8513</u>
	AC-GAN	7.98	49.46	207.56	0.7322	0.0793	0.6225	0.4112
	TAC-GAN	11.34	14.47	131.90	<u>0.7429</u>	0.6077	0.8324	0.7887
	ADC-GAN	**11.88**	**11.07**	104.21	0.7379	**0.6972**	0.8521	**0.8609**
	ContraGAN	11.15	13.54	146.86	0.7390	0.6155	0.8481	0.7729
	ReACGAN	<u>11.79</u>	13.72	125.21	**0.7541**	0.5861	**0.8695**	0.8005
LEAST SQUARE	PD-GAN	11.32	<u>12.19</u>	**101.92**	0.7263	<u>0.6903</u>	0.8318	<u>0.8471</u>
	AC-GAN	4.93	87.70	252.85	0.7087	0.0007	0.5836	0.2220
	TAC-GAN	7.27	49.08	162.58	0.7427	0.2114	0.7210	0.4438
	ADC-GAN	11.56	**11.85**	<u>103.06</u>	0.7334	**0.6949**	0.8145	**0.8526**
	ContraGAN	<u>12.59</u>	15.62	122.71	**0.7866**	0.4642	<u>1.0109</u>	0.7863
	ReACGAN	**12.90**	15.09	164.93	<u>0.7827</u>	0.4672	**1.0454**	0.8282
W-GP	PD-GAN	5.66	69.48	—	0.5976	0.1603	0.4310	0.2649
	AC-GAN	10.97	19.30	148.40	0.6880	0.5444	0.6770	0.7242
	TAC-GAN	**11.04**	<u>15.56</u>	<u>121.23</u>	<u>0.7023</u>	<u>0.6474</u>	<u>0.7048</u>	<u>0.7535</u>
	ADC-GAN	<u>11.01</u>	**14.02**	101.14	**0.7058**	**0.6804**	**0.7549**	**0.7956**
	ContraGAN	6.72	49.77	147.22	0.6498	0.2834	0.5827	0.3549
	ReACGAN	6.67	47.74	150.7	0.6188	0.3104	0.4806	0.3396
HINGE	PD-GAN	11.76	<u>10.96</u>	<u>108.08</u>	0.7436	<u>0.6812</u>	0.8790	<u>0.8609</u>
	AC-GAN	11.66	21.65	168.87	**0.7577**	0.3649	0.8297	0.7225
	TAC-GAN	**12.07**	12.56	134.75	<u>0.7572</u>	0.6020	<u>0.8957</u>	0.8400
	ADC-GAN	<u>11.82</u>	**10.73**	**103.78**	0.7387	**0.7023**	0.8721	**0.8707**
	ContraGAN	10.08	13.22	128.50	0.7372	0.6251	0.8356	0.7790
	ReACGAN	11.80	12.52	140.47	0.7510	0.5982	**0.9300**	0.8327