
ComGAN: Toward GANs Exploiting Multiple Samples

Haeone Lee

Independent Researcher
poirot946@gmail.com

Abstract

In this paper, we propose *ComGAN*(ComparativeGAN) which allows the generator in GANs to refer to the semantics of comparative samples(e.g. real data) by comparison. ComGAN generalizes relativistic GANs by using arbitrary architecture and mostly outperforms relativistic GANs in simple input-concatenation architecture. To train the discriminator in ComGAN, we also propose *equality regularization*, which fits the discriminator to a neutral label for equally real or fake samples. Equality regularization highly boosts the performance of ComGAN including WGAN while being exceptionally simple compared to existing regularizations. Finally, we generalize comparative samples fixed to real data in relativistic GANs toward fake data and show that such objectives are sound in both theory and practice. Our experiments demonstrate superior performances of ComGAN and equality regularization, achieving the best FIDs in 7 out of 8 cases of different losses and data against ordinary GANs and relativistic GANs.

1 Introduction

Generative Adversarial Networks(GANs)[6] train the generator G to generate fake data that follows the distributions of real data by fooling the discriminator D which distinguishes fake data from real data. GANs have shown promising performance in generative modeling, being able to generate photo-realistic images[4, 11, 12]. However, they suffer from training instability stemming from the non-stationary adversarial game[1, 4, 35], which might lead to mode collapse where the generated samples have very few modes or complete failure, e.g., generating only random noises.

To improve stability, several works explored applying different loss functions from vanilla GAN[24] such as least square loss[25](LSGAN), hinge loss [21](HingeGAN), and wasserstein loss[2](WGAN). Especially, WGAN[2] minimizes the wasserstein distance induced from integral probability metrics(IPMs)[31] which is mathematically weaker than other divergences, not causing diverging or vanishing gradients in theory. Along with WGAN, various GANs adopting losses from IPMs(IPM-GANs)[3, 29, 30] were also claimed to improve stability upon vanilla GAN. Apart from changing the loss function, other works regularize the discriminator to improve GANs training. The common approaches are constraining lipschitz norm[7, 28], regularizing the input gradient norm[14, 26], or posing invariance over data augmentation[42, 45] for better conditioning of the discriminator. Still, applying regularization might be computationally demanding in the case of computing input gradient[7, 14, 26]. Also, Lucic et al. [24] showed that GANs using different losses do not consistently outperform vanilla GAN.

Instead, some previous works explored utilizing multiple samples in the discriminator, which can be paired with any GAN losses. Jolicoeur-Martineau [9] proposed relativistic GANs where the discriminator logit is calculated by both real and fake samples after subtracting one from the other, i.e. $C(x) - C(y)$ where $x, y \sim p_{d,g}$. The author claims that relativistic GANs improve training stability by (i) reflecting prior knowledge that half of samples in mini-batch are fake and (ii) following training

procedure in real divergence minimization and (iii) resembling IPM-GANs which are known to be more stable than other variants. Similarly, Lin et al. [22] proposed PacGAN where the discriminator is trained by multiple real and fake samples, matching divergence between joint distributions of n real and fake samples. PacGAN inhibits mode-collapse due to the property of n -packed joint distributions[22]. Similar batch-wise discrimination was also attempted in [35] to alleviate mode-collapse.

In this paper, we propose ComparativeGAN(ComGAN) which generalizes the relativistic GANs toward arbitrary architecture. Departing from logit-level comparison in relativistic GANs, ComGAN allows the generator to refer to the semantics of the comparative sample(e.g. real data) when updating fake samples, mostly outperforming relativistic GANs under simple input concatenation. While Yu et al. [40] also proposed to employ real data as a reference in the discriminator, their training relied on the attention mechanism, different from ComGAN which naturally includes reference samples in loss functions. In addition, we show that relativistic GAN overlooks training terms for equally real or fake data and propose *equality regularization* to include such terms. Equality regularization highly boosts the performance of ComGAN, including WGAN and vanilla relativistic GAN. In contrast to gradient penalty regularizations[7, 14, 26] and spectral normalization[28], equality regularization is much more simple and easy to implement. Finally, we show that PacGAN and ComGAN are tied for matching either *packed* or *swapped* joint distributions and derive a regularization for PacGAN which corresponds to equality regularization in ComGAN. In our experiments, the combination of ComGAN and equality regularization outperformed relativistic GANs and ordinary GANs in 7 out of 8 cases under various losses and data. To summarize, our contributions are as follows:

- We propose ComGAN which enables semantic comparison from relativistic GANs and demonstrate improved performance.
- We generalize comparative samples which were fixed to real data in relativistic GANs towards fake data and show such objective is sound in both theory and practice.
- We propose equality regularization which improves the performance of ComGAN including WGAN while being exceptionally simple compared to existing regularizations.
- We derive a regularization objective for PacGAN corresponding to equality regularization in ComGAN, which can train a WGAN-like objective without other regularization.

The last of this paper is organized as follows: we review related works in Section 2 and define notations of GANs in Section 3. In Section 4, we introduce ComGAN and equality regularization and analyze them in statistical divergences and optimal discriminators. We summarize our experimental results in Section 5 and conclude the paper in Section 6.

2 Related Works

Generative adversarial networks GANs[6] are powerful generative models that have been adopted in various tasks such as synthesizing photo-realistic images[4, 11, 12], super-resolution[39, 43, 44], domain translation[5, 23]. However, GANs training is known to be unstable, resulting in mode collapse[1, 4, 35]. To resolve this, various mechanisms have been proposed such as using different loss functions[2, 21, 25] and applying regularizations[4, 12, 26, 28, 42].

Regularizing discriminators To improve and stabilize GANs training, numerous works have focused on regularizing the discriminator. [7] proposed gradient penalty which fits the gradient norm of the discriminator to 1.0 to satisfy lipschitz constraints. Similarly, [26] penalizes the gradient-norm of real data in the discriminator and [14] fits the gradient norm around real data to 1.0, which was shown to be effective to mitigate mode-collapse. [28] proposed spectral normalization which divides the discriminator weights by their spectral norms for capacity regularization. Instead of relying on lipschitz constraint, consistency regularization[42, 45] imposes the discriminator output to be consistent under different data augmentation, reducing computational overhead. Our equality regularization in ComGAN is much more simple and easy to implement compared to gradient penalties and spectral normalizations. Also, it is free from choosing proper data augmentation suitable for training data compared to consistency regularization.

Employing multiple samples in discriminators Instead of employing regularizations, other works improve GANs by modifying discriminator structures to exploit multiple samples. [9] proposed

relativistic GANs which calculate the discriminator logit by using both real and fake samples after subtracting one from the other. Relativistic GANs were claimed to improve training stability[9] and have been adopted in various tasks such as super-resolution[39, 43] and deblurring[16]. Similarly, [22] proposed PacGAN which trains a discriminator by multiple real samples and fake samples to surpass mode-collapse using the property of joint distributions. [35] proposed mini-batch discrimination which is similar to PacGAN except for using more specialized architecture. Our work generalizes over relativistic GANs and permits the use of arbitrary architecture, enabling semantic comparison to provide better learning signals. We also show the connections between relativistic GANs and PacGAN in statistical divergences and optimal discriminators.

Employing reference samples Since our ComGAN enables semantic comparison between real data and fake data using arbitrary architecture, the generator is able to refer to real data when updating fake samples. Similarly, [40] proposed GANs that employ real data as a reference in a discriminator using an attention mechanism in order to better guide the generator. However, their reference samples are unrelated to the loss function and complicated attention mechanism thus had to be used to combine reference samples in the training. Reference-based super-resolution(RefSR)[37, 44, 46] also uses reference samples to synthesize an HR image from an LR image, given the information from a reference image(e.g. similar scenes). In RefSR, such information is directly synthesized to the generator in various ways, e.g., style transfer[44], warping[46], deformable convolution[37] while ComGAN indirectly leverages reference samples by comparison in the discriminator. Finally, in text generation, [47] proposed *self-improvement* which trains the generator using reinforcement learning by rewarding the fake sentences that are better than previously generated ones and the other way around for worse sentences. Self-improvement was claimed to solve reward sparsity issues and prevent mode collapse[47]. In fact, self-improvement is equivalent to employing fake data as a reference in ComGAN and we expect that ComGAN using fake data would have similar advantages.

3 Background

3.1 Generative adversarial networks

GANs training involves an adversarial game between two models, the discriminator D_ψ and the generator G_θ , which are neural networks parameterized by ψ and θ . We express the training objectives of D and G as follows(we omitted the expression of ψ and θ for simplicity):

$$L_D(\min) = E_{x \sim p_d}[f_1(C(x))] + E_{x \sim p_g}[f_2(C(x))] \quad (1)$$

$$L_G(\min) = E_{x \sim p_d}[g_1(C(x))] + E_{x \sim p_g}[g_2(C(x))] \quad (2)$$

where $C(x)$ denotes a discriminator logit function which corresponds to a pre-activation value of the discriminator¹, i.e. $D(x) = \mathcal{A}(C(x))$ for output activation $\mathcal{A}(\cdot)$, and f_1, f_2, g_1, g_2 are scalar to scalar functions depending on the loss function applied, and p_d, p_g, p_z are real data distribution, fake data distribution, and generator prior(e.g. gaussian) respectively. We assume that both L_D and L_G are minimized unless otherwise specified by $L_D(\max)$ and $L_G(\max)$ which are $-L_D(\min)$ and $-L_G(\min)$. By setting $f_1(x) = \log(1 + e^{-x})$ and $f_2(x) = \log(1 + e^x)$, one can recover vanilla discriminator loss used in SGAN(standard-GAN)[6], expressed by

$$L_D = E_{x \sim p_d}[-\log D(x)] + E_{x \sim p_g}[-\log(1 - D(x))] \quad (\text{SGAN})$$

where $D(x) = \sigma(C(x))$ and $\sigma(\cdot)$ denotes sigmoid function. In such a case, $D(x)$ can be seen as a probability that x is real. For the generator objective, setting $g_1 = -f_1, g_2 = -f_1$ and $g_1 = f_2, g_1 = f_2$ respectively correspond to saturating loss and non-saturating loss[6, 9]. Under the assumption of the optimal discriminator, GANs can also be interpreted as a divergence minimization between real data and fake data, e.g., Jensen-Shannon divergence(JSD) in SGAN[6], f -divergence[33] or IPMs(Integral Probability Metrics)[2, 30, 36]. Normally, IPM-GANs are known to have benefits in stability due to the desirable properties of their metrics(e.g. weak divergence)[2, 30].

3.2 Relativistic GANs

Relativistic GANs modify the discriminator structure in GANs to improve stability[9]. Relativistic GANs are divided into RGAN and RaGAN by using two samples or multiple samples, whose

¹We also use the term "discriminator" to refer to $C(x)$

discriminator objectives are given by

$$L_D = E_{x \sim p_d, z \sim p_z} [f_1(C(x) - C(G(z))) + f_2(C(G(z)) - C(x))] \quad (\text{RGAN})$$

$$L_D = E_{x \sim p_d} [f_1(C(x) - \overline{C(x_f)})] + E_{z \sim p_z} [f_2(C(G(z)) - \overline{C(x_r)})] \quad (\text{RaGAN})$$

where $\overline{C(x_f)} = E_{z \sim p_z} [C(G(z))]$ and $\overline{C(x_r)} = E_{x \sim p_d} [C(x)]$. The generator losses are defined using the same g_1 and g_2 in ordinary GANs. Therefore, RGAN and RaGAN are equivalent to substituting the logit value in ordinary GANs with the value after subtracting the logit of opponent data. Note that WGAN[2] can be seen as a special case of RGAN and RaGAN, which corresponds to $f_1 = -I$ and $f_2 = I$ without consideration of capacity constraints(e.g. weight clipping[2], gradient penalty[7], etc.).

4 Method

4.1 ComGAN

To introduce our method of ComGAN, we first note that the discriminator objective of RGAN can be re-expressed by

$$L_D = E_{x, y \sim p_{d, g}} [f_1(C(x) - C(y))] + E_{x, y \sim p_{g, d}} [f_2(C(x) - C(y))]$$

where $p_{d, g}$ is the joint distribution of two independent real and fake data, and $p_{g, d}$ is similarly defined. A similar objective also holds for the generator. Instead of sticking to the subtraction form of $C(x) - C(y)$, we generalize a discriminator structure by arbitrary $C(x, y)$, resulting in the following:

$$L_D = E_{x, y \sim p_{d, g}} [f_1(C(x, y))] + E_{x, y \sim p_{g, d}} [f_2(C(x, y))] \quad (3)$$

$$L_G = E_{x, y \sim p_{d, g}} [g_1(C(x, y))] + E_{x, y \sim p_{g, d}} [g_2(C(x, y))] \quad (4)$$

where $C(x, y)$ is the discriminator logit function. Note that RGAN is a special case of (3) and (4), corresponding to letting $C(x, y) = \phi(x) - \phi(y)$ where $\phi : \mathbb{R}^{n_d} \rightarrow \mathbb{R}$ and n_d is the dimension of input data. In the case of using cross-entropy loss to train the discriminator, (3) is equal to

$$L_D = E_{x, y \sim p_{d, g}} [-\log D(x, y)] + E_{x, y \sim p_{g, d}} [-\log(1 - D(x, y))] \quad (5)$$

where $D(x, y) = \sigma(C(x, y))$ and $\sigma(\cdot)$ denotes sigmoid function. Intuitively, training the discriminator to minimize (5) tasks $D(x, y)$ to compare the ‘reality’ of x and y in the range of 0-1 and $D(x, y)$ can be understood as a probability that x is more real than y (i.e. $D(x, y) \approx 1$ indicates x is more real than y and vice-versa). Motivated by this, we name GANs using equations (3) and (4) *ComGAN*(ComparativeGAN). ComGAN allows the use of arbitrary discriminator structure for $C(x, y)$ and includes RGAN as its special case. Such property is useful since advanced neural network architecture(e.g. attention) can be utilized to compare x and y (see Section 4.2). Meanwhile, in *SComGAN*(standard-ComGAN) which employs the objective of (5), the optimal discriminator D^* is given by

$$D^*(x, y) = \frac{p_{d, g}(x, y)}{p_{d, g}(x, y) + p_{g, d}(x, y)}$$

Intuitively, we see that $D^*(x, y)$ imposes the value close to 1 if sample (x, y) has a high likelihood of $p_{d, g}$ and a low likelihood of $p_{g, d}$ and the other way around for $p_{g, d}$. The following proposition gives a further intuition of the behavior of ComGAN.

Proposition 1. *The optimal discriminator of SComGAN is equal to $\sigma(C^*(x) - C^*(y))$ where C^* is the optimal discriminator logit function of SGAN.*

Proof. Note that the optimal discriminator of SGAN is expressed by

$$D^*(x) = \frac{p_d(x)}{p_d(x) + p_g(x)} = \sigma \left(\log \frac{p_d(x)}{p_g(x)} \right). \therefore C^*(x) = \log \frac{p_d(x)}{p_g(x)}$$

Meanwhile,

$$D^*(x, y) = \sigma \left(\log \frac{p_{d, g}(x, y)}{p_{g, d}(x, y)} \right) = \sigma \left(\log \frac{p_d(x)}{p_g(x)} - \log \frac{p_d(y)}{p_g(y)} \right) \quad \square$$

Thus, the optimal discriminator of SComGAN equals subtracting the logits of the optimal SGAN discriminator with respect to two input samples, before applying the sigmoid function. In the case of RGAN where $C(x, y) = \phi(x) - \phi(y)$, we conclude $\phi^*(x) = C^*(x) + k$ (constant). Meanwhile, in the case of LSGAN using label 1/-1, we observe a conflict between the optimal discriminator given by $D^*(x, y) = \frac{p_{d,g}(x,y) - p_{g,d}(x,y)}{p_{d,g}(x,y) + p_{g,d}(x,y)}$ [25] and an RGAN discriminator $D(x, y) = \phi(x) - \phi(y)$ ($\mathcal{A} = I$) since the former cannot be expressed by the latter.

In divergence minimization, ComGAN matches the divergence between $p_{d,g}$ and $p_{g,d}$ as shown in Lemma 1.

Lemma 1. *Suppose L_G in (2) is equal to $D(p_d \parallel p_g)$ under the optimal discriminator $C^*(x)$. L_G in (4) defined by the same f_1, f_2, g_1, g_2 is equal to $D(p_{d,g} \parallel p_{g,d})$ under $C^*(x, y)$.*

Proof. (3) and (4) are equal to substituting p_d and p_g with $p_{d,g}$ and $p_{g,d}$ in (1) and (2) and $p_{d,g}$ and $p_{g,d}$ are probability distributions. \square

Therefore, SComGAN and WComGAN minimize $JSD(p_{d,g} \parallel p_{g,d})$ and $W^1(p_{d,g} \parallel p_{g,d})$ respectively [2, 6].

4.2 Comparative sample

In ComGAN, fake samples are trained to be considered more real than real samples in the comparison by the discriminator, aiming to increase the value of $C(G(z), y)$ where $y \sim p_d$. We refer to y as *comparative sample*, which is a reference employed to assist the training of $G(z)$. We suppose that the ‘good’ discriminator should compare two samples in high-level semantics to avoid small details of comparative sample from affecting the update of $G(z)$. In such aspects, the subtraction structure of the discriminator in RGAN conducts comparison in a discriminator logit level, the highest semantics that can be captured by the discriminator (i.e. whether to be real or fake). Although this is desirable due to the aforementioned property, RGAN renders it impossible for the generator to directly utilize information from comparative samples such as textures or frequent patterns. In fact, our experiments show that simply using x and $G(z)$ as network input by concatenation outperforms RaGAN in SComGAN and HingeComGAN. Taking $G(z)$ and y as the network input might also help to alleviate the mode-collapse issue since the limited modes in $G(z)$ would easily be captured from the comparison to real data. Similarly, Yu et al. [40] proposed reference attention where the attention map is computed upon real samples and is applied to the main layers of the discriminator as a reference. Such training objective can be expressed by $E_{x,y \sim p_{d,d}}[-\log D(x, y)] + E_{x,y \sim p_{g,d}}[-\log(1 - D(x, y))]$ in the case of SGAN with attention mechanism employed in $D(x, y)$. In this case, the network might learn to ignore a reference y since the input only differs by x and the optimal discriminator does not change from vanilla SGAN. On the other hand, applying such attention-based architectures in ComGAN is promising since ComGAN naturally leverages comparative samples in the objective. In addition, ComGAN can be extended toward multiple comparative samples (i.e. comparing a real sample to the batch of fake samples and vice-versa) in order to broadly capture data semantics by using the discriminator $C(x, x_1, \dots, x_n)$ where x is the main input and $x_1 \dots x_n$ are comparative samples. In case that $C(x, x_1, \dots, x_n) = \phi(x) - \frac{1}{n} \sum_{i=1}^n \phi(x_i)$, we recover RaGAN-like objective where mean logits are estimated using n samples yet other architectures such as simply concatenating $x, x_1 \dots, x_n$ for the network input would be possible.

Meanwhile, it should be noted that comparative can be generalized beyond real data as long as it can guide $G(z)$ to be improved upon itself. In previous work, Zhou et al. [47] proposed *self-improvement*, a reinforcement learning-based text generation algorithm that rewards the generator for generating sentences that are better than its previously generated samples, distinguished by the discriminator. Self-improvement was claimed to address the reward scarcity issue since improving upon fake samples is easier than improving upon real samples [47]. Employing fake data as comparative samples in ComGAN might enjoy the same advantages, e.g., preventing gradient-saturating issues, although the ability to refer to the semantics of real data will be lost. In the same sense, we consider using $G(z)$ itself as a comparative sample. From now on, we refer to ComGAN adopting fake data and the same sample (i.e. $G(z)$) for comparative samples as ComFakeGAN and ComSameGAN and express the generator objectives as follows (note that the discriminator objective remains unchanged):

$$L_G = E_{x,z \sim p_{g,z}}[g_1(C(\bar{x}, G(z)))] + E_{z,x \sim p_{z,g}}[g_2(C(G(z), \bar{x}))] \quad (\text{ComFakeGAN})$$

$$L_G = E_{z \sim p_z} [g_1(C(\overline{G(z)}, G(z)))] + E_{z \sim p_z} [g_2(C(G(z), \overline{G(z)}))] \quad (\text{ComSameGAN})$$

where \overline{x} indicates that the training is not performed with respect to x . In the case of non-saturating loss, i.e. $g_1 = f_2$ and $g_2 = f_1$, the objective of ComFakeGAN is equivalent to replacing the real data part of the joint distributions in equation (3) with fake data. Since the discriminator is trained to minimize (3) with respect to $p_{d,g}$ and $p_{g,d}$, minimizing such an objective with respect to fake data used in place of real data is sound. Similarly, ComSameGAN can be seen as improving $G(z)$ upon itself. In statistical divergences, those objectives can be analyzed as Theorem 1.

Theorem 1. *Under the optimal discriminator, the following holds for derivatives of generator objectives with respect to the generator parameter θ in saturating SComFakeGAN, non-saturating SComFakeGAN, saturating SComSameGAN, and non-saturating SComSameGAN:*

$$\begin{aligned} \nabla_{\theta} L_{SComFakeGAN}^{sat} + \nabla_{\theta} L_{SComFakeGAN}^{nonsat} &= 2\nabla_{\theta} KL(p_g \parallel p_d) \\ \nabla_{\theta} L_{SComSameGAN}^{sat} &= \nabla_{\theta} L_{SComSameGAN}^{nonsat} = \nabla_{\theta} KL(p_g \parallel p_d) \end{aligned}$$

Proof. See Appendix A.2

Therefore, our objectives are theoretically sound being related to minimizing KL divergence. In our experiments, ComFakeGAN using input concatenation surpassed RaGAN in SGAN and HingeGAN and showed similar performance to RaGAN in LSGAN, performing on par with ComGAN, while ComSameGAN failed in most cases. We speculate that the training signals given by $G(z)$ might be insufficient to train the same sample since the useful information not included in $G(z)$ will not be referred to.

4.3 Equality regularization

While the discriminator of SComGAN outputs a probability that x is more real than y , equation (5) does not take the case that x is equally real as y into account since the training objective is confined to $x, y \sim p_{d,g}$ or $x, y \sim p_{g,d}$. To elicit more accurate training, we propose to fit $D(x, y) \approx 0.5$ in case of $x, y \sim p_{d,d}$ and $x, y \sim p_{g,g}$ as a form of regularization, expressed by

$$\begin{aligned} L_D &= E_{x,y \sim p_{d,g}} [-\log D(x, y)] + E_{x,y \sim p_{g,d}} [-\log(1 - D(x, y))] \\ &+ \lambda_{reg} E_{x,y \sim p_{d,d}} [CE(0.5 \parallel D(x, y))] + \lambda_{reg} E_{x,y \sim p_{g,g}} [CE(0.5 \parallel D(x, y))] \end{aligned} \quad (6)$$

where $CE(p \parallel q)$ denotes cross-entropy between bernoulli distributions with a probability of p and q and λ_{reg} controls the strength of the regularization (normally we set $\lambda_{reg} = 1$). Note that the generator loss remains unchanged. For similar work, we account Cutmix[41] which improves classifier training by mixing patches from different images with their labels properly interpolated. Likewise, we perturb input (real, fake) and (fake, real) to (real, real) and (fake, fake) and impose a label of 0.5. For losses other than cross-entropy, we train $C(x, y) \approx 0$ since it corresponds to a neutral label or decision boundary in most GANs², resulting in

$$\begin{aligned} L_D &= E_{x,y \sim p_{d,g}} [f_1(C(x, y))] + E_{x,y \sim p_{g,d}} [f_2(C(x, y))] \\ &+ \lambda_{reg} E_{x,y \sim p_{d,d}} [\|C(x, y)\|^2] + \lambda_{reg} E_{x,y \sim p_{g,g}} [\|C(x, y)\|^2] \end{aligned} \quad (7)$$

where $\|\cdot\|$ denotes l2 norm. We refer to newly introduced terms in (6) and (7) as equality regularization and express ComGAN trained by (6) or (7) as ComGAN-eq.

Proposition 2. *Under $\lambda_{reg} = 1$, $D^*(x, y)$ of SComGAN-eq in (6) equals $\frac{1}{2} + \frac{1}{2}(D^*(x) - D^*(y))$ where $D^*(x)$ is the optimal discriminator of SGAN. $D^*(x, y)$ of LSCoMGAN-eq in (7) equals $\frac{1}{2}(D^*(x) - D^*(y))$ where $D^*(x)$ is the optimal discriminator of LSGAN.*

Proof. See Appendix A.3

Compared to Proposition 1, we see that SComGAN-eq corresponds to subtracting the output of SGAN discriminator (i.e. after the sigmoid) while vanilla SComGAN subtracts the logits (i.e. prior to the sigmoid). Interestingly, as LSCoMGAN-eq subtracts the output of LSGAN discriminator, it accords with LSRGAN architecture using $D(x, y) = \phi(x) - \phi(y)$. In our experiments, applying equality regularization significantly boosted performance in both RGAN and ComGAN using

²This holds in all the GAN loss we use in experiments, e.g., SGAN, HingeGAN, LSGAN with -1/1 coding, WGAN.

input-concatenation architectures. Especially, SRGAN-eq achieved the lowest Fréchet Inception Distance(FID)[8] among SGAN variants in CIFAR10[15] and LSCoMGAN-eq achieved the lowest FID in LSGAN variants of Tiny-ImageNet[18]. Meanwhile, in the case of using multiple comparative samples(i.e. a batch of real samples x_r and a batch of fake samples x_f), we propose to regularize $C(x, x_r) \approx 0$ where $x \sim p_d$ and $C(x, x_f) \approx 0$ where $x \sim p_g$, asking $x \sim p_d$ to be aligned with x_r and similarly for $x \sim p_g$. In RaGAN, which currently is the only algorithm that employs multiple comparative samples, this results in the following:

$$L_D = E_{x \sim p_d} [f_1(\phi(x) - E_{y \sim p_g}[\phi(y)])] + E_{x \sim p_g} [f_2(\phi(x) - E_{y \sim p_d}[\phi(x)])] \\ + \lambda_{reg} E_{x \sim p_d} [\|\phi(x) - E_{y \sim p_d}[\phi(x)]\|^2] + \lambda_{reg} E_{x \sim p_g} [\|\phi(x) - E_{y \sim p_g}[\phi(y)]\|^2]. \quad (8)$$

In our experiments, RaGAN-eq defined by (8) outperformed RaGAN by large margins demonstrating the effect of equality regularization. As WGAN can be seen as a special case of RaGAN, WGAN-GP-eq which uses $f_1 = -I$ and $f_2 = I$ in (8) also outperformed WGAN-GP[7], achieving the lowest FID in Tiny-ImageNet experiments. Furthermore, we show that objectives (7) and (8) can be trained under simple $f_1 = -I$ and $f_2 = I$, which is similar to WGAN yet without regularizing discriminator capacity. We refer to such objectives as WCoMGAN-eq and WGAN-eq respectively, although the mathematical connection to Wasserstein distance is not proven. WCoMGAN-eq outperformed both WCoMGAN-GP and WCoMGAN-GP-eq and WGAN-eq obtained the lowest FID of 24.86 in our CIFAR10 experiments. Finally, we show that LSRGAN and LSRaGAN indirectly include equality and rf regularization in their objectives. To see this, note that the discriminator in RGAN is trained to satisfy $\phi(x) - \phi(y) \approx 1$ for $x, y \sim p_{d,g}$, which is equivalent to letting $\phi(x) \approx c_1$ and $\phi(x_f) \approx c_2$ for $c_1 - c_2 = 1$ and $x, y \sim p_{d,g}$. In such a case, $\phi(x) - E_{x \sim p_d}[\phi(x)] \approx 0$ already holds for $x \sim p_d$. A similar result can be derived for LSRaGAN. This is also supported by our experiments where LSRaGAN shows notably better performance than other RaGANs and adding equality regularization to LSRaGAN rarely affects the performance. Still, the performance of LSRaGAN was lower than WGAN-eq.

Meanwhile, one might find (8) to be similar to LeCam regularization[38] which was introduced to assist GANs training under limited data by matching the discriminator logit of current data to the moving average value of the opposite data. However, (8) matches the discriminator logits to the mean logit of the same data(not a moving average), and the mean logit $E_{x \sim p_d}[\phi(x)]$ and $E_{z \sim p_z}[\phi(G(z))]$ are also trainable(not a constant target). In the case of fixing $E_{x \sim p_d}[\phi(x)]$ and $E_{z \sim p_z}[\phi(G(z))]$ as constants denoted by α_r and α_f and assuming $\alpha_r = -\alpha_f$ in the discriminator, the generator objective of WGAN-eq becomes $(\frac{1}{2\lambda} + \alpha_r)\Delta(p_d \parallel p_g)$ in optimality where $\Delta(P \parallel Q)$ denotes LeCam divergence[19, 38](see Appendix B).

4.4 Regularizing PacGAN

Similar to CoMGAN, PacGAN[22] uses multiple samples in the discriminator. We hereby show that PacGAN is comparable to CoMGAN with respect to logit operation and derive the corresponding regularization. Firstly, the objective of PacGAN is defined by

$$L_D = E_{x_1 \dots x_n \sim p_d^n} [f_1(C(x_1 \dots x_n))] + E_{z_1 \dots z_n \sim p_g^n} [f_2(C(x_1 \dots x_n))] \quad (\text{PacGAN})$$

where p_d^n and p_g^n denote the joint distribution of n independent real and fake samples[22]. Therefore, PacGAN matches the divergence between p_d^n and p_g^n while CoMGAN matches $p_{d,g}$ and $p_{g,d}$. Lin et al. [22] showed that the higher n is, the more the generator is penalized against mode-collapse due to the property of joint distributions. For SPacGAN, the optimal discriminator is expressed by the sum of ordinary discriminators as Proposition 3.

Proposition 3. *The optimal discriminator of SPacGAN, $D^*(x_1 \dots x_n)$, is equal to $\sigma(\sum_{i=1}^n C^*(x_i))$ where C^* is the optimal discriminator logit function of SGAN.*

Proof. See Appendix A.4

For $n = 2$, this is comparable to Proposition 1 in CoMGAN with respect to adding two logits or subtracting one logit from the other. In case we let $C(x_1 \dots x_n) = \sum_{i=1}^n \phi(x_i)$, deviating from the original implementation where the discriminator uses the concatenation of $x_1 \dots x_n$ as input[22], we obtain $\phi^*(x) = C^*(x)$. Such architecture is invariant to input permutation, satisfying $C(x, y) = C(y, x)$ when $n = 2$, compared to the property of $C(x, y) = -C(y, x)$ in RGAN.

Motivated by equality regularization in ComGAN, we now study the objectives that pose $C(x, y) \approx 0$ when $x, y \sim p_{d,g}$ and $x, y \sim p_{g,d}$ in PacGAN-2, which are expressed as follows for SPacGAN and the generalized version:

$$L_D = E_{x,y \sim p_{d,d}}[-\log D(x, y)] + E_{x,y \sim p_{g,g}}[-\log(1 - D(x, y))] \\ + \lambda_{reg} E_{x,y \sim p_{d,g}}[CE(0.5 \parallel D(x, y))] + \lambda_{reg} E_{x,y \sim p_{g,d}}[CE(0.5 \parallel D(x, y))] \quad (9)$$

$$L_D = E_{x,y \sim p_{d,d}}[f_1(C(x, y))] + E_{x,y \sim p_{g,g}}[f_2(C(x, y))] \\ + \lambda_{reg} E_{x,y \sim p_{d,g}}[\|C(x, y)\|^2] + \lambda_{reg} E_{x,y \sim p_{g,d}}[\|C(x, y)\|^2] \quad (10)$$

Since SPacGAN imposes a label of 1 for (real, real) and 0 for (fake, fake), it is reasonable to use the label of 0.5 for (real, fake) and (fake, real) which are ‘half-real’. From now on, we refer to added terms in (9) and (10) as rf regularization and PacGAN using rf regularization as PacGAN-rf.

Proposition 4. *Under $\lambda_{reg} = 1$, $D^*(x, y)$ of SPacGAN-rf in (9) equals $\frac{1}{2}(D^*(x) + D^*(y))$ where $D^*(x)$ is the optimal discriminator of SGAN. $D^*(x, y)$ of LSPacGAN-rf in (10) equals $\frac{1}{2}(D^*(x) + D^*(y))$ where $D^*(x)$ is the optimal discriminator of LSGAN.*

Proof. See Appendix A.5

Proposition 4 of PacGAN and Proposition 2 of ComGAN are closely related and the differences are whether to sum the value of SGAN discriminators or subtract one from the other. In case of letting $C(x, y) = \phi(x) + \phi(y)$, we propose to fit $C(x, x_f) \approx 0$ for $x \sim p_d$ and x_f which represents a batch of fake samples, resulting in

$$L_D = E_{x,y \sim p_{d,d}}[f_1(\phi(x) + \phi(y))] + E_{x,y \sim p_{g,g}}[f_2(\phi(x) + \phi(y))] \\ + \lambda_{reg} E_{x \sim p_d}[\|\phi(x) + E_{y \sim p_g}[\phi(y)]\|^2] + \lambda_{reg} E_{x \sim p_g}[\|\phi(x) + E_{y \sim p_d}[\phi(y)]\|^2]. \quad (11)$$

Especially, given $f_1 = -I$, $f_2 = I$ in (11), we obtain an objective similar to WGAN and refer to it as WGAN-rf. In our experiments, WGAN-rf showed competitive performance to SGAN and LSGAN in CIFAR10[15] yet its performance was generally below WGAN-eq.

In figure 1, we visualize equality regularization and rf regularization in relativistic GAN and PacGAN using a geometric fashion inspired by [21]. Especially, we let $\phi(x) = \langle w, \Phi_\psi(x) \rangle$ where Φ_ψ is discriminator feature extractor parameterized by ψ and w is a normal vector of the separating hyperplane(i.e. a decision boundary)³. In RaGAN-eq, by training $\langle w, \Phi(x) - E_{x \sim p_d}[\Phi(x)] \rangle \approx 0$ where $x \sim p_d$, it is equal to asking $\Phi(x) - E_{x \sim p_d}[\Phi(x)]$ to be orthogonal to the normal vector or parallel to separating hyperplane, and similarly for fake data. We particularly used mean-matching GAN for visualization, whose objective is given by

$$L_D(\max) = \max_{\|w\|_2 \leq 1, \|\psi\|_\beta \leq c} \langle w, E_{x \sim p_d}[\Phi_\psi(x)] - E_{x \sim p_g}[\Phi_\psi(x)] \rangle \quad (12)$$

where c is a finite constant. Mean-matching GAN can be seen as a special case of RGAN, RaGAN, and PacGAN using $f_1 = -I$ and $f_2 = I$ where the optimal normal vector w^* is given by $cE_{x \sim p_d}[\Phi(x)] - cE_{x \sim p_g}[\Phi(x)]$ [21, 30]. As shown in Figure 1(a), equality regularization poses feature vectors to form a parallel shape to the decision boundary in both real and fake data, which can roughly be interpreted as pushing each feature vector toward a line that is parallel to the decision boundary(blue dotted line) passing through the mean feature vector. Likewise, we observe that rf regularization in (11) asks $\Phi(x) + E_{x \sim p_g}[\Phi(x)]$ to be orthogonal to the normal vector in real data and similarly for fake data. As shown in Figure 1(b) using mean-matching GAN, it is equal to encouraging $\Phi(x) + E_{x \sim p_g}[\Phi(x)]$ (i.e. real features translated by the mean fake features) to lie on the decision boundary.

³If bias b in the final layer is excluded, this covers most discriminators implemented by neural networks, although RGAN and RaGAN do not depend on b which is subtracted out.

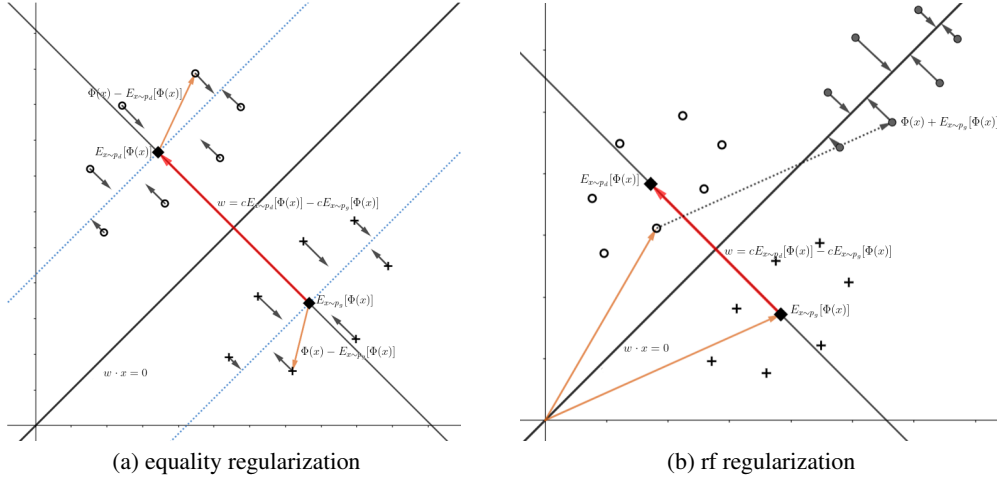


Figure 1: A visualization of the proposed regularizations. **(all)** Empty circles(o), cross marks(x), and filled rectangles(■) represent feature vectors of real data, fake data, and their mean feature vectors respectively and the black solid line represents the decision boundary with respect to the normal vector expressed by the red arrow. **(a)** equality regularization: real and fake data points are regularized to the direction of the blue dotted lines, generating a parallel shape to the decision boundary. **(b)** rf regularization: real data points transformed by the addition of the mean feature of fake data are regularized toward the decision boundary.

5 Experiments

We experimented with our methods in CIFAR10 dataset[15] composed of 60000 32×32 images(50k for training and 10k for test) and Tiny-Imagenet dataset[18] composed of 100000 64×64 images(90k for training and 10k for validation) using the loss functions of non-saturating SGAN[6], LSGAN[25], HingeGAN[21], and WGAN[2]. We implemented our algorithms upon StudioGAN framework[10] which provides the implementations of different GANs and various metrics for evaluation in order to assess GANs under a consistent environment with minimal changes. Our implementation is publicly available at here⁴. Following the basic configuration of StudioGAN, we used the deep convolutional architecture replicated from [34] for training in CIFAR10 and used the ResNet architecture replicated from [7] for training in Tiny-ImageNet except for WGAN which only uses ResNet architecture. For input-concatenation architecture in ComGAN, we only modified the number of input channels and other parts of the networks were kept the same. To see further details on network architecture, we ask the readers to refer to [10] and our implementation. For hyperparameters, we copied the default values of CIFAR10 training in StudioGAN and used the same values for Tiny-ImageNet experiments. Especially, we used the batch size of 64 and learning rate of 0.0002 for both discriminators and generators and set the discriminator to be updated twice per one generator update(i.e. $n_d = 2$) in SGAN and LSGAN while we set $n_d = 5$ in HingeGAN and WGAN. We employed Adam optimizer[13] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$ except for WGAN in Tiny-ImageNet where we use $\beta_1 = 0.0$ for stability. We trained SGAN, LSGAN, HingeGAN, and WGAN for 100k, 100k, 50k, and 50k steps in CIFAR10 and for 60k, 60k, 20k, and 20k steps in Tiny-ImageNet. We ran each algorithm 4 times and aggregated average performance and deviation. We also used mixed precision training[27] in CIFAR10 experiments apart from WGAN. For evaluations, we calculated Fréchet Inception Distance(FID)[8] which is the feature-wise distance between real data and fake data, Inception-score[35], Precision/Recall[17], and Density/Coverage[32] using test data in CIFAR10 and validation data in Tiny-ImageNet. Employing such various metrics for evaluation remarkably contributes to fair comparison since the performance of the model could vary along metrics[10].

⁴<https://github.com/rl-max/PyTorch-StudioGAN>

5.1 ComGAN experiments

We first display the performance of RaGAN with a comparison to ordinary GAN in Table 1. Note that WGAN can be seen as the special case of RaGAN.

Table 1: RaGAN results. We report the best FID and Inception-score during training averaged by 4 runs and Precision/Recall and Density/Coverage at the best FID with their standard deviations.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(CIFAR10)						
SGAN	50.65(±4.65)	6.73(±0.34)	0.60(±0.03)	0.29(±0.03)	0.46(±0.04)	0.37(±0.04)
SRaGAN	58.99(±5.76)	6.28(±0.35)	0.57(±0.07)	0.24(±0.03)	0.42(±0.12)	0.27(±0.05)
LSGAN	47.95(±17.71)	6.71(±0.57)	0.62(±0.01)	0.38(±0.15)	0.48(±0.04)	0.39(±0.11)
LSRaGAN	34.15(±1.36)	7.55(±0.15)	0.63(±0.02)	0.51(±0.01)	0.54(±0.04)	0.50(±0.02)
HingeGAN	37.33(±1.83)	7.18(±0.26)	0.67(±0.02)	0.33(±0.01)	0.64(±0.05)	0.49(±0.02)
HingeRaGAN	46.31(±2.23)	7.23(±0.19)	0.61(±0.04)	0.33(±0.01)	0.51(±0.09)	0.38(±0.04)
WGAN-GP	33.20(±5.91)	7.07(±0.40)	0.63(±0.01)	0.57(±0.05)	0.54(±0.03)	0.49(±0.07)
(Tiny-ImageNet)						
SGAN	76.94(±3.45)	6.99(±0.24)	0.49(±0.02)	0.11(±0.03)	0.33(±0.04)	0.22(±0.01)
SRaGAN	87.13(±3.17)	6.70(±0.25)	0.42(±0.05)	0.07(±0.01)	0.22(±0.05)	0.17(±0.01)
LSGAN	77.91(±2.67)	6.73(±0.19)	0.46(±0.02)	0.13(±0.02)	0.28(±0.03)	0.21(±0.02)
LSRaGAN	73.70(±4.29)	6.99(±0.29)	0.52(±0.03)	0.18(±0.01)	0.36(±0.04)	0.23(±0.02)
HingeGAN	78.46(±2.02)	7.05(±0.30)	0.48(±0.06)	0.12(±0.03)	0.31(±0.06)	0.21(±0.01)
HingeRaGAN	96.38(±5.87)	6.05(±0.26)	0.52(±0.08)	0.04(±0.02)	0.35(±0.12)	0.17(±0.03)
WGAN-GP	62.96(±1.78)	7.56(±0.21)	0.47(±0.01)	0.29(±0.02)	0.29(±0.01)	0.25(±0.01)

As shown in Table 1, applying RaGAN in SGAN and LSGAN actually deteriorates the performance except for LSRaGAN which indirectly includes equality regularization. Also, WGAN-GP performs reasonably well achieving the best FIDs and Recalls in both CIFAR10 and Tiny-ImageNet datasets. In Figure 3 in Appendix D, we further show that performances of ordinary GAN and RaGAN are reversed at the later part of training as RaGAN diverges more slowly. This quite corresponds to the stability argument in the original paper[9]. In fact, experiments in [9] were conducted with stability focused, reporting FIDs of different algorithms at the specific training step(100k) and FIDs averaged over different time steps(i.e. performance variations during training).

We also show the results of ComGAN using input-concatenation architecture in Table 2.

Table 2: ComGAN using input-concatenation architecture results. Metrics were averaged by 4 runs. Values in brackets denote performance gain over RaGAN(see Table 8 and 9 for standard deviations).

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(CIFAR10)						
SComGAN	46.56(-12.43)	6.82(+0.54)	0.60(+0.02)	0.32(+0.08)	0.45(+0.03)	0.38(+0.11)
LSCoGAN	76.03(+41.87)	5.14(-2.41)	0.65(+0.02)	0.22(-0.3)	0.51(-0.03)	0.25(-0.24)
HingeComGAN	37.28(-9.03)	7.46(+0.22)	0.65(+0.05)	0.34(+0.01)	0.62(+0.11)	0.49(+0.10)
WComGAN-GP	79.19(+45.99)	3.96(-3.10)	0.69(+0.06)	0.09(-0.47)	0.60(+0.06)	0.19(-0.30)
(Tiny-ImageNet)						
SComGAN	75.20(-11.93)	7.57(+0.87)	0.43(+0.01)	0.13(+0.07)	0.25(+0.03)	0.20(+0.04)
LSCoGAN	82.58(+8.88)	6.93(-0.06)	0.45(-0.07)	0.11(-0.07)	0.28(-0.08)	0.19(-0.03)
HingeComGAN	85.45(-10.93)	6.63(+0.58)	0.43(-0.09)	0.09(+0.05)	0.26(-0.10)	0.18(+0.01)
WComGAN-GP	154.41(+91.45)	3.49(-4.07)	0.32(-0.15)	0.01(-0.29)	0.12(-0.17)	0.06(-0.19)

In Table 2, we observe that using input concatenation architecture in ComGAN surpasses RaGAN by significant margins in SGAN and HingeGAN while it deteriorates the performance in LSGAN and WGAN(against WGAN-GP). Note that a such comparison might be unfavorable to ComGAN since RaGAN uses multiple comparative samples and LSRaGAN benefits from the implicit equality regularization effect. In fact, we show in Section 5.2 that LSCoGAN with equality regularization

outperforms LSRaGAN in Tiny-ImageNet and shows competitive performance in CIFAR10. Such results prove the potential of applying semantic comparison in the discriminator.

Finally, we present the results of using fake data as the comparative sample in Table 3.

Table 3: ComFakeGAN using input concatenation architecture results. We denote ComFakeGAN by FakeGAN.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(CIFAR10)						
SFakeGAN	45.55(±4.53)	6.57(±0.33)	0.63(±0.03)	0.33(±0.03)	0.51(±0.05)	0.39(±0.05)
LSFakeGAN	35.45(±1.98)	7.09(±0.15)	0.64(±0.01)	0.48(±0.02)	0.54(±0.01)	0.49(±0.04)
HingeFakeGAN	36.24(±1.72)	7.42(±0.07)	0.63(±0.02)	0.36(±0.03)	0.56(±0.05)	0.48(±0.01)
WFakeGAN-GP	75.55(±5.44)	4.20(±0.19)	0.67(±0.02)	0.12(±0.03)	0.55(±0.05)	0.21(±0.02)
(Tiny-ImageNet)						
SFakeGAN	80.43(±5.57)	6.97(±0.40)	0.44(±0.05)	0.13(±0.02)	0.27(±0.04)	0.20(±0.02)
LSFakeGAN	78.63(±5.58)	6.86(±0.53)	0.49(±0.06)	0.10(±0.02)	0.33(±0.07)	0.21(±0.04)
HingeFakeGAN	82.89(±4.74)	6.70(±0.49)	0.47(±0.03)	0.09(±0.02)	0.29(±0.04)	0.20(±0.01)
WFakeGAN-GP	169.11(±23.83)	3.34(±0.22)	0.25(±0.11)	0.00(±0.00)	0.10(±0.05)	0.05(±0.02)

By comparison with Table 2, we see that ComFakeGAN performs on par with ComGAN in all cases except for LSComFakeGAN in CIFAR10 which achieves significantly higher performance than LSComGAN. Meanwhile, in the case of ComSameGAN, we encountered training failures in most cases(see Table 8 in Appendix). We hypothesize that the training signal available for the generator is limited in ComSameGAN since it is unable to utilize diverse information from different comparative samples.

5.2 Applying regularization

We first display the results of applying equality regularization in SRGAN(standard-RGAN) and ComGAN using input concatenation architecture under $\lambda_{reg} = 1.0$.

Table 4: ComGAN-eq results. We apply equality regularization of (6) in SComGAN-eq and equality regularization of (7) in other algorithms. Input concatenation architecture was used except for RGAN and λ_{reg} was set to 1.0. Values in brackets denote performance gain over ComGAN.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(CIFAR10)						
SComGAN-eq	45.71(-0.85)	6.61(-0.21)	0.64(+0.04)	0.42(+0.10)	0.53(+0.08)	0.38(+0.00)
SRGAN-eq	32.73(-24.55)	7.34(+1.02)	0.63(+0.03)	0.54(+0.29)	0.53(+0.05)	0.48(+0.18)
LSComGAN-eq	37.62(-38.40)	6.87(+1.73)	0.65(+0.00)	0.46(+0.25)	0.56(+0.05)	0.46(+0.20)
HingeComGAN-eq	28.85(-8.43)	7.57(+0.12)	0.63(-0.02)	0.55(+0.21)	0.54(-0.07)	0.53(+0.04)
WComGAN-GP-eq ⁵	56.71(-22.48)	6.00(+2.04)	0.64(-0.05)	0.36(+0.27)	0.54(-0.05)	0.37(+0.18)
WComGAN-eq	34.92(-44.27)	7.14(+3.18)	0.61(-0.08)	0.48(+0.39)	0.50(-0.09)	0.47(+0.28)
(Tiny-ImageNet)						
SComGAN-eq	77.74(+2.54)	7.05(-0.52)	0.49(+0.06)	0.15(+0.01)	0.33(+0.07)	0.21(+0.01)
SRGAN-eq	72.24(-29.08)	6.78(+0.40)	0.54(+0.16)	0.18(+0.14)	0.39(+0.18)	0.23(+0.09)
LSComGAN-eq	70.34(-12.24)	7.47(+0.54)	0.51(+0.06)	0.20(+0.09)	0.33(+0.05)	0.24(+0.05)
HingeComGAN-eq	78.70(-6.74)	6.54(-0.08)	0.47(+0.04)	0.12(+0.04)	0.28(+0.02)	0.20(+0.02)
WComGAN-GP-eq	106.64(-47.77)	5.04(+1.55)	0.33(+0.01)	0.05(+0.04)	0.16(+0.04)	0.11(+0.05)
WComGAN-eq	105.58(-48.83)	5.52(+2.03)	0.35(+0.03)	0.03(+0.02)	0.18(+0.06)	0.12(+0.06)

Table 4 shows that ComGAN-eq outperforms ComGAN by a large margin in most cases demonstrating the effectiveness of equality regularization. We further show in Table 8 and 9 that ComGAN-eq surpasses RaGAN in all experiments except for LSRaGAN in CIFAR10 and WGAN-GP. Especially,

⁵terminated at 32k step

WComGAN-eq outperforms WComGAN-GP-eq in CIFAR10, showing competitive performance with WGAN-GP, and performs on par with WComGAN-GP-eq in Tiny-ImageNet proving that equality regularization can replace gradient penalty.

In the case of applying equality regularization in RaGAN, the results are given by Table 5.

Table 5: RaGAN-eq results. We apply equality regularization of (8) under $\lambda_{reg} = 1.0$ except for WGAN-GP-eq in CIFAR10 where we use $\lambda_{reg} = 0.1$. Values in brackets denote performance gain over RaGAN.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(CIFAR10)						
SRaGAN-eq	32.76(-26.23)	7.52(+1.24)	0.62(+0.04)	0.56(+0.32)	0.52(+0.10)	0.48(+0.21)
LSRaGAN-eq	31.16(-3.00)	7.64(+0.09)	0.62(-0.01)	0.58(+0.07)	0.55(+0.01)	0.52(+0.03)
HingeRaGAN-eq	25.06(-21.25)	7.93(+0.70)	0.66(+0.05)	0.57(+0.24)	0.64(+0.13)	0.61(+0.22)
WGAN-GP-eq	26.96(-6.24)	7.76(+0.69)	0.65(+0.02)	0.60(+0.04)	0.59(+0.05)	0.57(+0.08)
WGAN-eq	24.86(-8.35)	7.90(+0.83)	0.65(+0.02)	0.58(+0.02)	0.62(+0.09)	0.62(+0.13)
(Tiny-ImageNet)						
SRaGAN-eq	69.06(-18.07)	7.17(+0.48)	0.56(+0.14)	0.20(+0.13)	0.38(+0.16)	0.25(+0.08)
LSRaGAN-eq	74.29(+0.59)	6.87(-0.12)	0.52(+0.00)	0.16(-0.01)	0.33(-0.03)	0.22(+0.00)
HingeRaGAN-eq	82.80(-13.58)	6.10(+0.05)	0.50(-0.02)	0.13(+0.10)	0.30(-0.05)	0.20(+0.03)
WGAN-GP-eq ⁶	57.18(-5.78)	8.23(+0.67)	0.49(+0.02)	0.38(+0.09)	0.32(+0.03)	0.29(+0.04)
WGAN-eq	85.72(+22.77)	6.04(-1.52)	0.51(+0.04)	0.11(-0.19)	0.31(+0.02)	0.19(-0.06)

Table 5 demonstrates that applying equality regularization to RaGAN significantly improves the performance apart from LSRaGAN. This corresponds to the fact that LSRaGAN indirectly includes equality regularization in the objective. RaGAN-eq also outperforms ordinary GAN by a significant margin except for HingeGAN in Tiny-ImageNet(see Table 8 and 9). Especially, the lowest FIDs in CIFAR10 and Tiny-ImageNet are respectively achieved by WGAN-eq and WGAN-GP-eq. WGAN-eq also outperforms WGAN-GP-eq in CIFAR10. We thus confirm that equality regularization renders it possible for ComGAN and RaGAN to be trained under the simplest loss of $f_1 = -I$ and $f_2 = I$. For the value of λ_{reg} , we observed that simply using 1.0 was sufficient in most experiments except for WGAN-GP-eq in CIFAR10 where $\lambda_{reg} = 1.0$ led the discriminator to collapse to constant output due to strong regularization, which was addressed by setting $\lambda_{reg} = 0.1$. Finally, in Figure 2, we show the training curves of RaGAN and RaGAN-eq in CIFAR10, where we observe the stable performance of equality regularization, outperforming RaGAN throughout the whole training apart from LSRaGAN.

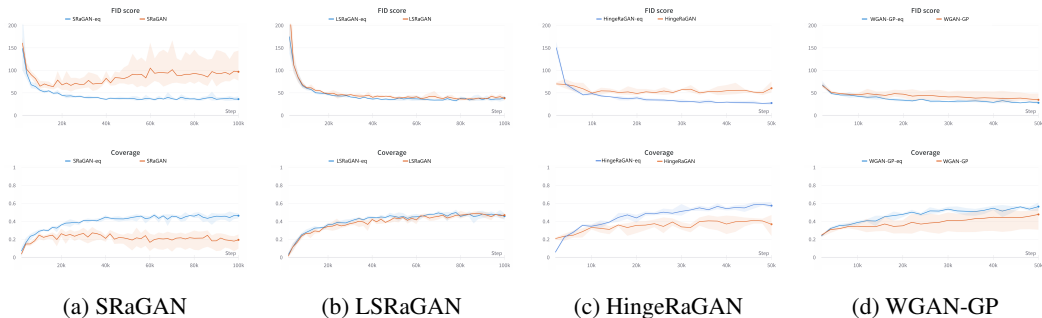


Figure 2: FID and Coverage curves of RaGAN(orange) and RaGAN-eq(blue) in CIFAR10 dataset.

We also experimented on WGAN-rf which applies rf regularization in PacGAN under $C(x, y) = \phi(x) + \phi(y)$ and $f_1 = -I$ and $f_2 = I$ as shown in Table 6.

⁶terminated at 18k step

Table 6: WGAN-rf results. We apply equality regularization of (11) under $\lambda_{reg} = 1.0$. (C) and (T) denote experiments in CIFAR10 and Tiny-ImageNet.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
WGAN-rf(C)	46.46(±2.25)	6.54(±0.20)	0.61(±0.02)	0.29(±0.01)	0.48(±0.03)	0.35(±0.03)
WGAN-rf(T)	106.27(±4.45)	5.65(±0.20)	0.41(±0.10)	0.03(±0.01)	0.24(±0.10)	0.13(±0.02)

From Table 6, we conclude that WGAN-rf is solely trainable without other regularization such as gradient penalty, attaining FID close to vanilla SGAN and LSGAN in CIFAR10 experiments yet the performance is still worse than WGAN-eq. We surmise that more experiments would be required to fairly compare rf-regularization and equality regularization.

Meanwhile, since restricting the capacity of discriminators has resulted in better performance in most GANs[4, 20, 28, 42], one might think that equality regularization performs well because of the same reason. To test such an argument, we apply equality regularization in ordinary GANs and present results in Table 7.

Table 7: GAN-eq results. We apply equality regularization of (8) under $\lambda_{reg} = 1.0$. The results of CIFAR10 and Tiny-ImageNet are respectively shown in the upper rows and the lower rows. Values in brackets denote performance gain over ordinary GANs.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
SGAN-eq	34.74(-15.92)	7.31(+0.58)	0.63(+0.03)	0.50(+0.21)	0.53(+0.07)	0.48(+0.10)
HingeGAN-eq	54.09(+16.76)	5.73(-1.45)	0.60(-0.07)	0.34(+0.01)	0.47(-0.17)	0.30(-0.20)
SGAN-eq	80.58(+3.64)	6.43(-0.56)	0.49(+0.00)	0.14(+0.02)	0.29(-0.04)	0.20(-0.01)
HingeGAN-eq	95.55(+17.09)	5.95(-1.10)	0.43(-0.05)	0.05(-0.07)	0.26(-0.05)	0.15(-0.06)

Except for SGAN-eq in CIFAR10, we see that applying equality regularization in ordinary GANs actually deteriorates the performance, proving that the role of equality regularization is different from existing regularization. Such results also imply that equality regularization can promisingly be used with existing regularization(e.g. spectral normalization, r1 regularization).

Summary of experiments In our experiments, RaGAN did not perform better than ordinary GANs except for LSRaGAN which indirectly possesses equality regularization. Also, applying input concatenation architecture in ComGAN surpassed RaGAN by significant margins in SGAN and HingeGAN albeit it showed worse performance in LSGAN and WGAN. For comparative samples, ComFakeGAN performed on par with ComGAN while ComSameGAN failed. With equality regularization applied, the performance of ComGAN using input-concatenation significantly improved, outperforming RaGAN in most cases. Applying equality regularization in RaGAN also improved the performances demonstrated by WGAN-eq and WGAN-GP-eq which respectively achieve the lowest FID in CIFAR10 and Tiny-ImageNet. For rf regularization, we confirmed that WGAN-rf can be trained without difficulty. Finally, applying equality regularization in ordinary GANs degraded the performances proving that the role of equality regularization is different from existing regularizations. We summarized our training results in Table 8 and 9 and observed RGAN-eq and RaGAN-eq variants show superb performance achieving the lowest FID among algorithms in CIFAR10 experiments and Tiny-ImageNet experiments of SGAN and WGAN.

6 Conclusion and Future Work

In this work, we generalized relativistic GANs towards arbitrary architecture to enable semantic comparison. We showed training in equivalent samples is critical for performance improvements in relativistic GANs. The proposed methods, ComGAN and equality regularization outperformed relativistic GANs and ordinary GANs in various losses and evaluation metrics. Theoretically, we showed that ComGAN matches the divergences between swapped joint distributions between real

and fake data, including RGAN. We also expressed optimal discriminators of proposed methods and showed connections to PacGAN. In future works, we plan to employ more advanced architecture(e.g. attentions) in ComGAN to fully exploit semantic comparisons. We also intend to analyze upon employing multiple comparative samples in ComGAN and explore rf regularization in PacGAN under various settings.

References

- [1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.
- [2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.
- [3] Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.
- [4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.
- [5] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.
- [6] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2*, pages 2672–2680, 2014.
- [7] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein GANs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 5769–5779, 2017.
- [8] 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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6629–6640, 2017.
- [9] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. *arXiv preprint arXiv:1807.00734*, 2018.
- [10] Minguk Kang, Joonghyuk Shin, and Jaesik Park. Studiogan: A taxonomy and benchmark of GANs for image synthesis. *arXiv preprint arXiv:2206.09479*, 2022.
- [11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
- [12] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.
- [13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [14] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215*, 2017.
- [15] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.
- [16] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8878–8887, 2019.

- [17] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3927–3936, 2019.
- [18] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015.
- [19] Lucien Le Cam. *Asymptotic methods in statistical decision theory*. Springer Science & Business Media, 2012.
- [20] Minhyeok Lee and Junhee Seok. Regularization methods for generative adversarial networks: An overview of recent studies. *arXiv preprint arXiv:2005.09165*, 2020.
- [21] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.
- [22] Zinan Lin, Ashish Khetan, Giulia Fanti, and Sewoong Oh. PacGAN: the power of two samples in generative adversarial networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 1505–1514, 2018.
- [23] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 700–708, 2017.
- [24] Mario Lucic, Karol Kurach, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. Are GANs created equal? a large-scale study. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 698–707, 2018.
- [25] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.
- [26] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for GANs do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018.
- [27] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017.
- [28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.
- [29] Youssef Mroueh and Tom Sercu. Fisher GAN. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 2510–2520, 2017.
- [30] Youssef Mroueh, Tom Sercu, and Vaibhava Goel. Mcgan: Mean and covariance feature matching GAN. In *International Conference on Machine Learning*, pages 2527–2535. PMLR, 2017.
- [31] Alfred Müller. Integral probability metrics and their generating classes of functions. *Advances in applied probability*, 29(2):429–443, 1997.
- [32] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. Reliable fidelity and diversity metrics for generative models. In *International Conference on Machine Learning*, pages 7176–7185. PMLR, 2020.
- [33] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-GAN: training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 271–279, 2016.
- [34] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2234–2242, 2016.

- [36] Tim Salimans, Han Zhang, Alec Radford, and Dimitris Metaxas. Improving GANs using optimal transport. *arXiv preprint arXiv:1803.05573*, 2018.
- [37] Gyumin Shim, Jinsun Park, and In So Kweon. Robust reference-based super-resolution with similarity-aware deformable convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8425–8434, 2020.
- [38] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7921–7931, 2021.
- [39] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.
- [40] Ning Yu, Guilin Liu, Aysegul Dundar, Andrew Tao, Bryan Catanzaro, Larry S Davis, and Mario Fritz. Dual contrastive loss and attention for GANs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6731–6742, 2021.
- [41] Sangdoon Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.
- [42] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. *arXiv preprint arXiv:1910.12027*, 2019.
- [43] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3217–3226, 2020.
- [44] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7982–7991, 2019.
- [45] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for GANs. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 11033–11041, 2021.
- [46] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super-resolution network using cross-scale warping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 88–104, 2018.
- [47] Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. Self-adversarial learning with comparative discrimination for text generation. *arXiv preprint arXiv:2001.11691*, 2020.

A Proofs

A.1 Proof of Theorem 1

Let us express the optimal discriminator of SComGAN by $D(x, y) = \sigma(C^*(x) - C^*(y))$ where $C^*(x) = \log \frac{p_d(x)}{p_g(x)}$. Observe that

$$E_{z \sim p_z}[-C^*(G_\theta(z))] = KL(p_g \parallel p_d). \quad (13)$$

Given $D^*(x, y) = \sigma(C^*(x) - C^*(y))$ where $C^*(x) = \log \frac{p_d(x)}{p_g(x)}$,

(i)

$$\nabla_\theta L_{SComFakeGAN}^{sat} = 2E_{x \sim p_g, z \sim p_z}[\nabla_\theta \log \sigma(C^*(x) - C^*(G_\theta(z)))] \quad (14)$$

$$= -2E_{x \sim p_g, z \sim p_z}[\sigma(C^*(G_\theta(z)) - C^*(x))\nabla_\theta C^*(G_\theta(z))] \quad (15)$$

$$= -2E_{x \sim p_g, z \sim p_z}[\nabla_\theta C^*(G_\theta(z)) - \sigma(C^*(x) - C^*(G_\theta(z)))\nabla_\theta C^*(G_\theta(z))] \quad (16)$$

and

$$\nabla_\theta L_{SComFakeGAN}^{nonsat} = -2E_{x \sim p_g, z \sim p_z}[\nabla_\theta \log \sigma(C^*(G_\theta(z)) - C^*(x))] \quad (17)$$

$$= -2E_{x \sim p_g, z \sim p_z}[\sigma(C^*(x) - C^*(G_\theta(z)))\nabla_\theta C^*(G_\theta(z))]. \quad (18)$$

$$\therefore \nabla_\theta L_{SComFakeGAN}^{sat} + \nabla_\theta L_{SComFakeGAN}^{nonsat} = -2E_{z \sim p_z}[\nabla_\theta C^*(G_\theta(z))] \quad (19)$$

$$= 2\nabla_\theta KL(p_g \parallel p_d) \quad \therefore (13) \quad \square$$

(ii)

$$\nabla_\theta L_{SComSameGAN}^{sat} = 2E_{z \sim p_z}[\nabla_\theta \log \sigma(C^*(\overline{G_\theta(z)}) - C^*(G_\theta(z)))] \quad (20)$$

$$= -2E_{z \sim p_z}[\sigma(C^*(G_\theta(z)) - C^*(\overline{G_\theta(z)}))\nabla_\theta C^*(G_\theta(z))] \quad (21)$$

$$= \nabla_\theta KL(p_g \parallel p_d) \quad \square$$

$$\nabla_\theta L_{SComSameGAN}^{nonsat} = -2E_{z \sim p_z}[\nabla_\theta \log \sigma(C^*(G_\theta(z)) - C^*(\overline{G_\theta(z)}))] \quad (22)$$

$$= -2E_{z \sim p_z}[\sigma(C^*(G_\theta(z)) - C^*(\overline{G_\theta(z)}))\nabla_\theta C^*(G_\theta(z))] \quad (23)$$

$$= \nabla_\theta KL(p_g \parallel p_d) \quad \square$$

where \bar{x} indicates x is constant not having gradients.

A.2 Proof of Proposition 2

Equation (6) is equal to

$$-\iint \{p_{d,g}(x, y) + 0.5p_{d,d}(x, y) + 0.5p_{g,g}(x, y)\} \log D(x, y) + \quad (24)$$

$$\{p_{g,d}(x, y) + 0.5p_{d,d}(x, y) + 0.5p_{g,g}(x, y)\} \log(1 - D(x, y)) dx dy.$$

By minimizing the above equation, we obtain the following optimal discriminator:

$$D^*(x, y) = \frac{p_{d,g}(x, y) + 0.5p_{d,d}(x, y) + 0.5p_{g,g}(x, y)}{p_{d,d}(x, y) + p_{d,g}(x, y) + p_{g,d}(x, y) + p_{g,g}(x, y)} \quad (25)$$

$$= \frac{p_{d,g}(x, y) + 0.5p_{d,d}(x, y) + 0.5p_{g,g}(x, y)}{(p_d(x) + p_g(x))(p_d(y) + p_g(y))}. \quad (26)$$

By expressing $D^*(x, y)$ with respect to $D^*(x) = \frac{p_d(x)}{p_d(x) + p_g(x)}$, it is equal to

$$= D^*(x)(1 - D^*(y)) + 0.5D^*(x)D^*(y) + 0.5(1 - D^*(x))(1 - D^*(y)) \quad (27)$$

$$= 0.5 + 0.5(D^*(x) - D^*(y)). \quad \square$$

Meanwhile, the objective of LSComGAN-eq in (7) is

$$\iint p_{d,g}(x, y)(D(x, y) - 1)^2 + p_{g,d}(x, y)(D(x, y) + 1)^2 + \quad (28)$$

$$p_{d,d}(x, y)D(x, y)^2 + p_{g,g}D(x, y)^2 dx dy.$$

where $D^*(x, y)$ is given by

$$D^*(x, y) = \frac{p_{d,g}(x, y) - p_{g,d}(x, y)}{(p_d(x) + p_g(x))(p_d(y) + p_g(y))}. \quad (29)$$

By expressing $D^*(x, y)$ with respect to LSGAN optimal discriminator $D^*(x) = \frac{p_d(x) - p_g(x)}{p_d(x) + p_g(x)}$, it is equal to

$$= 0.5(D^*(x) - D^*(y)). \quad \square$$

A.3 Proof of Proposition 3

In SPacGAN-n, the optimal discriminator is given by

$$D^*(x_1 \dots x_n) = \frac{p_d^n(x_1 \dots x_n)}{p_d^n(x_1 \dots x_n) + p_g^n(x_1 \dots x_n)} \quad (30)$$

$$= \sigma \left(\log \frac{p_d^n(x_1 \dots x_n)}{p_g^n(x_1 \dots x_n)} \right) = \sigma \left(\sum_{i=1}^n \log \frac{p_d(x_i)}{p_g(x_i)} \right). \quad (31)$$

Since $C^*(x) = \log \frac{p_d(x)}{p_g(x)}$, we conclude the proof. \square

A.4 Proof of Proposition 4

By minimizing the equation (9) expressed by

$$- \iint \{p_{d,d}(x, y) + 0.5p_{d,g}(x, y) + 0.5p_{g,d}(x, y)\} \log D(x, y) + \{p_{g,g}(x, y) + 0.5p_{d,g}(x, y) + 0.5p_{g,d}(x, y)\} \log(1 - D(x, y)) dx dy, \quad (32)$$

we obtain the optimal discriminator as

$$D^*(x, y) = \frac{p_{d,d}(x, y) + 0.5p_{d,g}(x, y) + 0.5p_{g,d}(x, y)}{p_{d,d}(x, y) + p_{d,g}(x, y) + p_{g,d}(x, y) + p_{g,g}(x, y)} \quad (33)$$

$$= \frac{p_{d,d}(x, y) + 0.5p_{d,g}(x, y) + 0.5p_{g,d}(x, y)}{(p_d(x) + p_g(x))(p_d(y) + p_g(y))}. \quad (34)$$

By expressing $D^*(x, y)$ with respect to $D^*(x) = \frac{p_d(x)}{p_d(x) + p_g(x)}$, it is equal to

$$= D^*(x)D^*(y) + 0.5D^*(x)(1 - D^*(y)) + 0.5(1 - D^*(x))D^*(y) \quad (35)$$

$$= 0.5(D^*(x) + D^*(y)). \quad \square$$

Meanwhile, the objective of LSPacGAN-rf in (10) is

$$\iint p_{d,d}(x, y)(D(x, y) - 1)^2 + p_{g,g}(x, y)(D(x, y) + 1)^2 + p_{d,g}(x, y)D(x, y)^2 + p_{g,d}(x, y)D(x, y)^2 dx dy, \quad (36)$$

where $D^*(x, y)$ is given by

$$D^*(x, y) = \frac{p_{d,d}(x, y) - p_{g,g}(x, y)}{(p_d(x) + p_g(x))(p_d(y) + p_g(y))}. \quad (37)$$

By expressing $D^*(x, y)$ with respect to LSGAN optimal discriminator $D^*(x) = \frac{p_d(x) - p_g(x)}{p_d(x) + p_g(x)}$, it is equal to

$$= 0.5(D^*(x) + D^*(y)). \quad \square$$

B Connection to LeCam divergence

Let us define the discriminator objective as follows:

$$L_D = E_{x \sim p_g}[\phi(x)] - E_{x \sim p_d}[\phi(x)] + \lambda E_{x \sim p_d}[\|\phi(x) - E_{x \sim p_d}[\overline{\phi(x)}]\|^2] \\ + \lambda E_{x \sim p_g}[\|\phi(x) - E_{z \sim p_z}[\overline{\phi(G(z))}]\|^2]. \quad (38)$$

(38) is similar to WGAN-eq except that it does not train the discriminator with respect to $E_{x \sim p_d}[\phi(x)]$ and $E_{z \sim p_z}[\phi(G(z))]$. From now on, we let $\alpha_r = E_{x \sim p_d}[\overline{\phi(x)}]$ and $\alpha_f = E_{z \sim p_z}[\overline{\phi(G(z))}]$.

Proposition 5. *In case that $\alpha_r = -\alpha_f$, the generator objective defined by $L_G = E_{x \sim p_d}[\phi^*(x)] - E_{x \sim p_g}[\phi^*(x)]$ is equal to $(\frac{1}{2\lambda} + \alpha_r)\Delta(p_d \parallel p_g)$ under the optimal discriminator $\phi^*(x)$ of (38), where $\Delta(P \parallel Q)$ is LeCam divergence[19] given by*

$$\int \frac{(P(x) - Q(x))^2}{P(x) + Q(x)} dx.$$

Proof.

Note that L_D in (38) can be re-expressed by

$$L_D = E_{x \sim p_d}[\lambda \|\phi(x) - \alpha_r\|^2 - \phi(x)] + E_{x \sim p_g}[\lambda \|\phi(x) + \alpha_r\|^2 + \phi(x)] \quad (39)$$

$$= E_{x \sim p_d}[\lambda \|\phi(x) - \alpha_r\|^2 - \phi(x) + \alpha_r + \frac{1}{4\lambda}] + \quad (40)$$

$$E_{x \sim p_g}[\lambda \|\phi(x) + \alpha_r\|^2 + \phi(x) + \alpha_r + \frac{1}{4\lambda}] + C$$

$$= \lambda \int p_d(x) (\phi(x) - \alpha_r - \frac{1}{2\lambda})^2 + p_g(x) (\phi(x) + \alpha_r + \frac{1}{2\lambda})^2 dx + C \quad (41)$$

where $C = 2\alpha_r - \frac{1}{2\lambda}$. By maximizing the above equation, we obtain the optimal ϕ^* as

$$\phi^*(x) = \frac{(\frac{1}{2\lambda} + \alpha_r)(p_d(x) - p_g(x))}{p_d(x) + p_g(x)} \quad \therefore \frac{dL_D}{d\phi} = 0. \quad (42)$$

By substituting ϕ^* in the generator objective,

$$L_G = \int p_d(x) \phi^*(x) - p_g(x) \phi^*(x) dx \quad (43)$$

$$= (\frac{1}{2\lambda} + \alpha_r) \int \frac{(p_d(x) - p_g(x))^2}{p_d(x) + p_g(x)} dx \quad (44)$$

$$= (\frac{1}{2\lambda} + \alpha_r) \Delta(p_d \parallel p_g)$$

□

C Training losses

C.1 SGAN

$$\begin{aligned} L_D^{sat} &= E_{x \sim p_d}[-\log D(x)] + E_{x \sim p_z}[-\log(1 - D(G(z)))] \\ L_G^{sat} &= E_{x \sim p_d}[\log D(x)] + E_{x \sim p_z}[\log(1 - D(G(z)))] \quad (\text{Not used}) \\ L_G^{nonsat} &= E_{x \sim p_d}[-\log(1 - D(x))] + E_{x \sim p_z}[-\log D(G(z))] \end{aligned}$$

where $f_1 = \log(1 + e^{-x})$, $f_2 = \log(1 + e^x)$, and $\mathcal{A} = \sigma$.

C.2 LSGAN

$$\begin{aligned} L_D &= E_{x \sim p_d}[(D(x) - 1)^2] + E_{x \sim p_z}[(D(G(z)) + 1)^2] \\ L_G &= E_{x \sim p_d}[(D(x) + 1)^2] + E_{x \sim p_z}[(D(G(z)) - 1)^2] \end{aligned}$$

where $f_1 = (x - 1)^2$, $f_2 = (x + 1)^2$, $g_1 = f_2$, $g_2 = f_1$, and $\mathcal{A} = I$.

C.3 HingeGAN

$$\begin{aligned} L_D &= E_{x \sim p_d}[\max(0, 1 - D(x))] + E_{x \sim p_z}[\max(0, 1 + D(G(z)))] \\ L_G &= E_{x \sim p_d}[D(x)] - E_{x \sim p_z}[D(G(z))] \end{aligned}$$

where $f_1 = \max(0, 1 - x)$, $f_2 = \max(0, 1 + x)$, $g_1 = I$, $g_2 = -I$, and $\mathcal{A} = I$.

C.4 WGAN

$$\begin{aligned} L_D &= E_{x \sim p_d}[-D(x)] + E_{x \sim p_z}[D(G(z))] \\ L_G &= E_{x \sim p_d}[D(x)] - E_{x \sim p_z}[D(G(z))] \end{aligned}$$

where $f_1 = -I$, $f_2 = I$, $g_1 = I$, $g_2 = -I$, and $\mathcal{A} = I$.

RGAN, RaGAN, ComGAN, and PacGAN are defined as described in previous sections.

D Supplementary plots

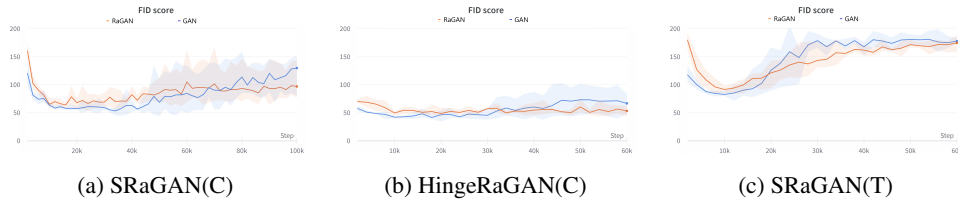


Figure 3: FID curves of RaGAN(orange) and ordinary GAN(blue). (C) and (T) denotes CIFAR10 and Tiny-ImageNet(We did not include experiments of HingeRaGAN in Tiny-ImageNet since it was trained for much fewer iterations(20k)).

E Performance tables

Table 8: Comprehensive training results in CIFAR10 dataset[15]. We report the best FID and Inception Score during training averaged by 4 runs and Precision/Recall and Density/Coverage metrics at the best FID with their standard deviations. ComGAN denotes ComGAN using input concatenation architecture and FakeGAN and SameGAN denote ComFakeGAN and ComSameGAN using input concatenation architecture respectively.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(training steps: 100k)						
SGAN	50.65(±4.65)	6.73(±0.34)	0.60(±0.03)	0.29(±0.03)	0.46(±0.04)	0.37(±0.04)
SComGAN	46.56(±2.31)	6.82(±0.19)	0.60(±0.03)	0.32(±0.03)	0.45(±0.04)	0.38(±0.01)
SComGAN-eq	45.71(±7.52)	6.61(±0.36)	0.64 (±0.02)	0.42(±0.07)	0.53 (±0.04)	0.38(±0.05)
SFakeGAN	45.55(±4.53)	6.57(±0.33)	0.63(±0.03)	0.33(±0.03)	0.51(±0.05)	0.39(±0.05)
SFakeGAN-eq	40.18(±1.80)	6.62(±0.11)	0.64 (±0.01)	0.46(±0.02)	0.53 (±0.03)	0.42(±0.02)
SSameGAN	246.58(±61.19)	2.42(±0.50)	0.47(±0.21)	0.00(±0.00)	0.28(±0.22)	0.02(±0.02)
SRGAN	57.28(±3.95)	6.32(±0.11)	0.59(±0.01)	0.26(±0.04)	0.48(±0.03)	0.30(±0.03)
SRGAN-eq	32.73 (±1.53)	7.34(±0.07)	0.63(±0.02)	0.54(±0.02)	0.53 (±0.04)	0.48 (±0.03)
SRaGAN	58.99(±5.76)	6.28(±0.35)	0.57(±0.07)	0.24(±0.03)	0.42(±0.12)	0.27(±0.05)
SRaGAN-eq	32.76(±0.24)	7.52 (±0.14)	0.62(±0.01)	0.56 (±0.03)	0.52(±0.04)	0.48 (±0.02)
(training steps: 100k)						
LSGAN	47.95(±17.71)	6.71(±0.57)	0.62(±0.01)	0.38(±0.15)	0.48(±0.04)	0.39(±0.11)
LSComGAN	76.03(±29.71)	5.14(±1.27)	0.65 (±0.05)	0.22(±0.19)	0.51(±0.09)	0.25(±0.13)
LSComGAN-eq	37.62(±1.82)	6.87(±0.14)	0.65 (±0.01)	0.46(±0.03)	0.56 (±0.01)	0.46(±0.02)
LSFakeGAN	35.45(±1.98)	7.09(±0.15)	0.64(±0.01)	0.48(±0.02)	0.54(±0.01)	0.49(±0.04)
LSFakeGAN-eq	37.15(±1.80)	6.88(±0.16)	0.64(±0.01)	0.50(±0.01)	0.55(±0.02)	0.45(±0.02)
LSSameGAN	201.97(±13.76)	2.92(±0.39)	0.57(±0.20)	0.00(±0.00)	0.39(±0.27)	0.03(±0.01)
LSRaGAN	34.15(±1.36)	7.55(±0.15)	0.63(±0.02)	0.51(±0.01)	0.54(±0.04)	0.50(±0.02)
LSRaGAN-eq	31.16 (±0.76)	7.64 (±0.13)	0.62(±0.01)	0.58 (±0.02)	0.55(±0.02)	0.52 (±0.01)
(training steps: 50k)						
HingeGAN	37.33(±1.83)	7.18(±0.26)	0.67 (±0.02)	0.33(±0.01)	0.64 (±0.05)	0.49(±0.02)
HingeComGAN	37.28(±2.29)	7.46(±0.15)	0.65(±0.03)	0.34(±0.03)	0.62(±0.07)	0.49(±0.02)
HingeComGAN-eq	28.85(±1.66)	7.57(±0.12)	0.63(±0.02)	0.55(±0.01)	0.54(±0.04)	0.53(±0.03)
HingeFakeGAN	36.24(±1.72)	7.42(±0.07)	0.63(±0.02)	0.36(±0.03)	0.56(±0.05)	0.48(±0.01)
HingeFakeGAN-eq	28.69(±1.00)	7.55(±0.15)	0.65(±0.00)	0.55(±0.01)	0.58(±0.01)	0.55(±0.01)
HingeSameGAN	96.97(±28.97)	4.62(±1.06)	0.63(±0.05)	0.07(±0.10)	0.47(±0.08)	0.17(±0.10)
HingeRaGAN	46.31(±2.23)	7.23(±0.19)	0.61(±0.04)	0.33(±0.01)	0.51(±0.09)	0.38(±0.04)
HingeRaGAN-eq	25.06 (±0.71)	7.93 (±0.1)	0.66(±0.02)	0.57 (±0.01)	0.64 (±0.03)	0.61 (±0.01)
(training steps: 50k)						
WGAN-GP	33.20(±5.91)	7.07(±0.40)	0.63(±0.01)	0.57(±0.05)	0.54(±0.03)	0.49(±0.07)
WGAN-GP-eq	26.96(±0.76)	7.76(±0.12)	0.65(±0.01)	0.60 (±0.01)	0.59(±0.01)	0.57(±0.02)
WGAN-eq	24.86 (±0.69)	7.90 (±0.12)	0.65(±0.01)	0.58(±0.01)	0.62 (±0.02)	0.62 (±0.01)
WGAN-rf	46.46(±2.25)	6.54(±0.20)	0.61(±0.02)	0.29(±0.01)	0.48(±0.03)	0.35(±0.03)
WComGAN-GP	79.19(±5.49)	3.96(±0.22)	0.69 (±0.04)	0.09(±0.02)	0.60(±0.07)	0.19(±0.02)
WComGAN-GP-eq ⁷	56.71(±37.28)	6.00(±1.66)	0.64(±0.06)	0.36(±0.21)	0.54(±0.08)	0.37(±0.15)
WComGAN-eq	34.92(±1.01)	7.14(±0.10)	0.61(±0.01)	0.48(±0.01)	0.50(±0.03)	0.47(±0.00)
WFakeGAN-GP	75.55(±5.44)	4.20(±0.19)	0.67(±0.02)	0.12(±0.03)	0.55(±0.05)	0.21(±0.02)
WSameGAN-GP ⁸	338.58(±71.10)	1.57(±0.46)	0.56(±0.21)	0.00(±0.00)	0.15(±0.06)	0.00(±0.01)

⁷terminated at 32k step

⁸intentionally terminated at 12k step due to poor performance.

Table 9: Comprehensive training results in Tiny-ImageNet dataset[18]. We report the best FID and Inception Score during training averaged by 4 runs and Precision/Recall and Density/Coverage metrics at the best FID with their standard deviations. ComGAN denotes ComGAN using input concatenation architecture and FakeGAN and SameGAN denote ComFakeGAN and ComSameGAN using input concatenation architecture respectively.

Algorithms	FID ↓	IS ↑	Precision ↑	Recall ↑	Density ↑	Coverage ↑
(training steps: 60k)						
SGAN	76.94(±3.45)	6.99(±0.24)	0.49(±0.02)	0.11(±0.03)	0.33(±0.04)	0.22(±0.01)
SComGAN	75.20(±2.07)	7.57 (±0.38)	0.43(±0.04)	0.13(±0.01)	0.25(±0.05)	0.20(±0.02)
SComGAN-eq	77.74(±5.49)	7.05(±0.27)	0.49(±0.06)	0.15(±0.04)	0.33(±0.08)	0.21(±0.04)
SFakeGAN	80.43(±5.57)	6.97(±0.40)	0.44(±0.05)	0.13(±0.02)	0.27(±0.04)	0.20(±0.02)
SSameGAN	230.53(±6.10)	3.20(±0.26)	0.39(±0.29)	0.00(±0.00)	0.20(±0.23)	0.01(±0.00)
SRGAN	101.32(±2.52)	6.38(±0.23)	0.39(±0.02)	0.03(±0.01)	0.21(±0.03)	0.14(±0.01)
SRGAN-eq	72.24(±0.73)	6.78(±0.11)	0.54(±0.01)	0.18(±0.01)	0.39 (±0.03)	0.23(±0.00)
SRaGAN	87.13(±3.17)	6.70(±0.25)	0.42(±0.05)	0.07(±0.01)	0.22(±0.05)	0.17(±0.01)
SRaGAN-eq	69.06 (±3.63)	7.17(±0.32)	0.56 (±0.02)	0.20 (±0.04)	0.38(±0.03)	0.25 (±0.02)
(training steps: 60k)						
LSGAN	77.91(±2.67)	6.73(±0.19)	0.46(±0.02)	0.13(±0.02)	0.28(±0.03)	0.21(±0.02)
LSComGAN	82.58(±6.03)	6.93(±0.60)	0.45(±0.04)	0.11(±0.02)	0.28(±0.03)	0.19(±0.02)
LSComGAN-eq	70.34 (±4.87)	7.47 (±0.34)	0.51(±0.02)	0.20 (±0.05)	0.33(±0.03)	0.24 (±0.02)
LSFakeGAN	78.63(±5.58)	6.86(±0.53)	0.49(±0.06)	0.10(±0.02)	0.33(±0.07)	0.21(±0.04)
LSRaGAN	73.70(±4.29)	6.99(±0.29)	0.52 (±0.03)	0.18(±0.01)	0.36 (±0.04)	0.23(±0.02)
LSRaGAN-eq	74.29(±3.40)	6.87(±0.27)	0.52 (±0.03)	0.16(±0.03)	0.33(±0.03)	0.22(±0.01)
(training steps: 20k)						
HingeGAN	78.46 (±2.02)	7.05 (±0.30)	0.48(±0.06)	0.12(±0.03)	0.31(±0.06)	0.21 (±0.01)
HingeComGAN	85.45(±8.72)	6.63(±0.42)	0.43(±0.03)	0.09(±0.03)	0.26(±0.05)	0.18(±0.02)
HingeComGAN-eq	78.70(±2.09)	6.54(±0.21)	0.47(±0.02)	0.12(±0.02)	0.28(±0.03)	0.20(±0.01)
HingeFakeGAN	82.89(±4.74)	6.70(±0.49)	0.47(±0.03)	0.09(±0.02)	0.29(±0.04)	0.20(±0.01)
HingeRaGAN	96.38(±5.87)	6.05(±0.26)	0.52 (±0.08)	0.04(±0.02)	0.35 (±0.12)	0.17(±0.03)
HingeRaGAN-eq	82.80(±2.84)	6.10(±0.13)	0.50(±0.05)	0.13 (±0.01)	0.30(±0.04)	0.20(±0.01)
(training steps: 20k)						
WGAN-GP	62.96(±1.78)	7.56(±0.21)	0.47(±0.01)	0.29(±0.02)	0.29(±0.01)	0.25(±0.01)
WGAN-GP-eq ⁹	57.18 (±1.93)	8.23 (±0.10)	0.49(±0.01)	0.38 (±0.02)	0.32 (±0.01)	0.29 (±0.01)
WGAN-eq	85.72(±1.18)	6.04(±0.06)	0.51 (±0.01)	0.11(±0.02)	0.31(±0.01)	0.19(±0.00)
WGAN-rf	106.27(±4.45)	5.65(±0.20)	0.41(±0.10)	0.03(±0.01)	0.24(±0.10)	0.13(±0.02)
WComGAN-GP	154.41(±28.23)	3.49(±0.55)	0.32(±0.04)	0.01(±0.01)	0.12(±0.03)	0.06(±0.02)
WComGAN-GP-eq	106.64(±6.38)	5.04(±0.35)	0.33(±0.07)	0.05(±0.02)	0.16(±0.05)	0.11(±0.02)
WComGAN-eq	105.58(±4.02)	5.52(±0.31)	0.35(±0.05)	0.03(±0.02)	0.18(±0.05)	0.12(±0.01)
WFakeGAN-GP	169.11(±23.83)	3.34(±0.22)	0.25(±0.11)	0.00(±0.00)	0.10(±0.05)	0.05(±0.02)

⁹terminated at 18k step