Improving the Leaking of Augmentations in Data-Efficient GANs via Adaptive Negative Data Augmentation

Zhaoyu Zhang[1], Yang Hua[1], Guanxiong Sun[1,2], Hui Wang[1], Seán McLoone[1]
[1]Queen's University Belfast [2]Huawei UKRD
{zzhang55,Y.Hua,gsun02,h.wang,s.mcloone}@qub.ac.uk

Abstract

Data augmentation (DA) has shown its effectiveness in training Data-Efficient GANs (DE-GANs). However, applying DA in DE-GANs results in transforming the distributions of generated data and real data to augmented distributions of generated data and real data. This augmentation process could produce some out-of-distribution samples, known as the leaking of augmentations problem, which is highly undesirable in DE-GANs training. Although some methods propose "leaking-free" DAs for DE-GANs, we theoretically and practically argue that the leaking of augmentations problem still exists in these methods. To alleviate the leaking of augmentations in DE-GANs, in this paper, we propose a simple yet effective method called adaptive negative data augmentation (ANDA) for DE-GANs, with a negligible computational cost increase. Specifically, ANDA adaptively augments the augmented distribution of generated data using the augmented distribution of negative real data, where the negative real data is produced by applying negative data augmentation (NDA) on the real data. In this case, potential leaking samples can be presented as "fake" instances to the discriminator adaptively, which avoids the generator (G) learning such samples, thus resulting in better performance. Extensive experiments on several datasets with different DE-GANs demonstrate that ANDA can effectively alleviate the leaking of augmentations problem during training and achieve better performance. Codes are available at https://github.com/zzhang05/ANDA

1. Introduction

Generative Adversarial Networks (GANs) [8] have achieved great success [14, 15, 17, 18, 23, 39, 43] in the past few years when working with large amounts of data. However, gathering and cleaning such enormous datasets is expensive, time-consuming, and often impossible. Therefore, Data-Efficient GANs (DE-GANs [19]) are receiving significant attention [4, 16, 19, 36, 44].

Data augmentation (DA) has recently shown its importance in training DE-GANs. Many studies [4, 16, 44] apply DA to both real and fake data for the discriminator (D) and generator (G) in DE-GANs to improve the DE-GANs training. However, training DE-GANs with DA leads to transforming the distributions of generated data and real data to augmented distributions of generated data and real data [31]. This augmentation process could produce out-of-distribution samples [42, 45], known as the leaking of augmentations problem in DE-GANs [16]. Although some approaches [16, 35] carefully design the DA in DE-GANs and state that their DA is leaking-free, we argue that the leaking of augmentations still exists in these methods. Specifically, we provide the theoretical analysis that applying DA in DE-GANs with non-saturating loss can yield the learning of augmented distributions, which can unavoidably produce some out-of-distribution samples, thus harming the training of DE-GANs. Based on the conclusion in ADA [16], i.e., "a noise augmentation leads to noisy results, even if there is none in the dataset", the leaking of augmentations problem can be better visualized by applying noise augmentation in DE-GANs. Therefore, we select DE-GANs with noise augmentation, i.e., Diffusion-GAN, to further demonstrate the leaking of augmentations problem in DE-GANs. As shown in Figure 1 (a), Diffusion-GAN [35] produces noise-based images.

To alleviate the leaking of augmentations problem in DE-GANs, we propose a simple yet effective method called adaptive negative data augmentation (ANDA) for DE-GANs. In contrast to previous DAs [4, 13, 16, 44], ANDA augments the augmented generated data distribution with the augmented negative real data distribution adaptively, where the negative real data is produced by applying negative data augmentation (NDA) [30] on the real data. Such augmented negative real samples are adaptively presented to the discriminator as "fake" instances to avoid G learning potential leaking samples, hence resulting in better performance, as shown in Figure 1 (b).

To sum up, the main contributions of this paper are as follows:

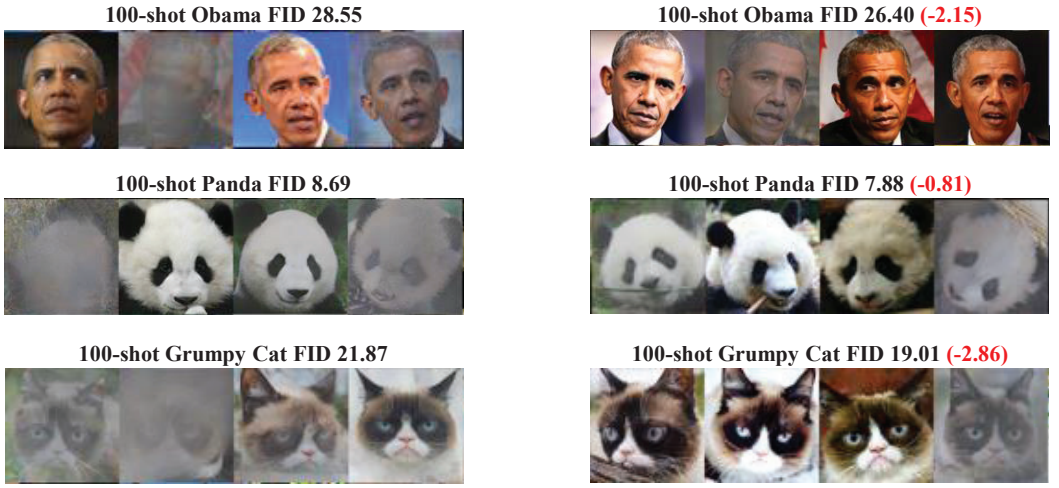

Figure 1. An illustration of the leaking of augmentations problem with Diffusion-GAN (StyleGAN2 [18] backbone) and Diffusion-GAN (StyleGAN2 backbone) + ANDA on the 100-shot Obama, Panda and Grumpy Cat datasets without cherry-picking the results. (a) Images generated by Diffusion-GAN (StyleGAN2 backbone). Diffusion-GAN (StyleGAN2 backbone) produces noise-based images caused by the leaking of augmentations problem. (b) Images generated by Diffusion-GAN (StyleGAN2 backbone) + ANDA. Adding ANDA to Diffusion-GAN (StyleGAN2 backbone) can effectively improve the leaking of augmentations problem, thus generating less noisy images with better quality (measured by the Fréchet Inception Distance, i.e., FID [11] scores). *Best viewed in color.*

1. We propose a novel adaptive negative data augmentation (ANDA) for DE-GANs. This method makes D regard potential leaking samples as "fake" instances during training to avoid G learning these leaking samples adaptively, hence resulting in better performance.

2. We analyze the leaking of augmentations problem existing in DE-GANs with non-saturating loss in both theory and practice. Based on this, we theoretically connect ANDA with optimizing the non-saturating loss in DE-GANs, proving its convergence and rationality.

3. Extensive experiments on different DE-GANs [16, 35, 44] with six commonly used datasets demonstrate that the proposed ANDA can effectively mitigate the leaking of augmentations problem and achieve better performance with negligible computational cost.

2. Related Work

2.1. Generative Adversarial Networks (GANs)

Generative adversarial networks (GANs) [8] is a form of generative models [27, 32] in which a game is played between two players: A generator (G) and a discriminator (D). Specifically, G aims to produce realistic-looking samples with some given noise z to deceive D, while D aims to distinguish whether the input sample is from the generator's output or real data. The objective function of GANs can be formulated as follows:

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

The parameters of G and D are updated iteratively with gradient descent methods. Theoretically, GANs have been shown to optimize the Jensen-Shannon (JS) divergence between the generator's distribution (P_G) and real data distribution (P_R). GANs are known to suffer from training instability, yielding poor quality and diversity of generated images. To stabilize GANs training and improve the quality and diversity of generated images, various approaches have been proposed, focusing on more sophisticated network architectures [3, 22, 23, 28, 38, 40], more stable objective functions [2, 9, 10, 21, 29], and better training strategies [6, 15, 19, 20, 41] to achieve photorealistic results.

2.2. Data-Efficient GANs (DE-GANs)

Recently, data augmentation (DA) has played an important role in improving the performance of training Data Efficient GANs (DE-GANs). Many studies [4, 16, 31, 44] apply DA to both real and fake samples for D and G to guide the discriminator to avoid overfitting, thus enhancing the training of DE-GANs. The most popular methods are Diff-Augment [44], ADA [16], and Diffusion-GAN [35]. Diff-Augment applies the DAs to both real and fake images for D and G without manipulating the target distribution.

ADA is similar to Diff-Augment, while it further devises an adaptive approach that controls the strength of data augmentations. Diffusion-GAN applies the forward diffusion process [12] as DA to both real and fake images adaptively. The objective function of DA in StyleGAN2 [18] can be formulated as follows:

$$\begin{aligned} V_D(G,D) &= \mathbb{E}_{T(x)\sim P_R^T}[\log D(T(x))] \\ &\quad + \mathbb{E}_{T(x)\sim P_G^T}[\log(1-D(T(x)))], \\ V_G(G,D) &= -\mathbb{E}_{T(x)\sim P_G^T}[\log(D(T(x)))], \end{aligned} \tag{2}$$

where T is a certain augmentation method. Following the existing studies [16, 42, 45], DA applied to both real and fake samples for D and G can cause the leaking of augmentations problem during training. In this work, we extend the study of the leaking of augmentations problem in DE-GANs.

2.3. Negative Data Augmentation (NDA)

Recently, negative data augmentation (NDA) [30] has been proposed to produce out-of-distribution samples to benefit the GANs training. NDA-GANs guide the discriminator to regard the out-of-distribution samples as "fake" instances to improve GANs training. A recent study, OMAS-GAN [7], shows that the performance of NDA varies on different datasets and backbones of GANs.

3. Methodology

3.1. Leaking of Augmentations Problem

Recently, some approaches [31, 35] have focused on applying DA in DE-GANs. They concluded that DA in DE-GANs is leaking-free as long as two conditions are met simultaneously, i.e., the use of invertible DA and the application of saturating loss in DE-GANs. However, the commonly used DE-GANs backbone, i.e., StyleGAN2 [18], applies non-saturating loss, which can still cause the leaking of augmentations problem in DE-GANs. To better understand this, we first provide a theoretical analysis of the leaking of augmentations problem for StyleGAN2. We conclude that DA applied to both real and fake samples for D and G in DE-GANs with non-saturating loss results in optimization of the **KL-2JS** divergence between augmented generated data distribution P_G^T and augmented real data distribution P_R^T, shown as follows.

Based on the theory developed for the original GAN [8], we can obtain the optimal discriminator $D^*(T(x))$ for Eq.(2) as

$$D^*(T(x)) = \frac{P_R^T(T(x))}{P_R^T(T(x)) + P_G^T(T(x))}. \tag{3}$$

Then, given the optimal D^*, based on the Theorem 2.5 as in [1], training generator with these augmented samples $T(x)$ in Eq.(2) can be formulated as

$$V_G(G,D^*) = \mathbf{KL}(P_G^T \,||\, P_R^T) - 2\mathbf{JS}(P_G^T \,||\, P_R^T), \tag{4}$$

where **KL** is the Kullback-Leibler divergence and **JS** is the Jensen-Shannon divergence. Eq.(4) demonstrates that the DA applied to both G and D in DE-GANs leads to the optimization of the **KL-2JS** divergence between augmented distributions P_G^T and P_R^T. Compared with the DE-GANs without DA, applying DA in DE-GANs transforms the distribution of the original generated data (P_G) and real data (P_R) to augmented distributions P_G^T and P_R^T [31]. This augmentation process can unavoidably produce some out-of-distribution samples for both real data and generated data. Then, these out-of-distribution samples are applied to update the parameters of both G and D, therefore, harming the training of DE-GANs.

Next, we analyze the leaking of augmentations problem of three widely-used DA methods in DE-GANs with non-saturating loss.

Diff-Augment [44] in DE-GANs. Diff-Augment does not consider the leaking of augmentations problem when designing the augmentation method. Therefore, as shown in Eq.(4), applying Diff-Augment in DE-GANs results in optimization of the **KL-2JS** divergence between augmented distributions P_G^T and P_R^T, which demonstrates that the leaking of augmentations problem exists when Diff-Augment is applied to DE-GANs.

ADA [16] in DE-GANs. ADA introduces adaptive discriminator augmentation to alleviate the leaking of augmentations problem and states that ADA is leaking-free. However, we argue that the leaking of augmentations still exists when ADA is applied in DE-GANs. In ADA, the augmentation is controlled by a probability p (defined in Section 3 in the ADA paper). Specifically, augmentation is applied with p or skipped with $1-p$, where p is controlled by the degree of overfitting of D. In other words, what ADA does is to reduce the augmentation degree during the training of DE-GANs. Particularly, augmentation in ADA is only applied when D suffers from overfitting. As a result, ADA can relieve the leaking of augmentations but the augmentation operations still exist during training. As shown in Eq.(4), these existing augmentation operations in ADA can yield optimization of the augmented distributions, which still causes the leaking of augmentations problem.

Diffusion [35] in DE-GANs. Diffusion-GAN applies the forward Diffusion process to augment both real and fake images in DE-GANs. Because Diffusion as the augmentation is invertible, Diffusion in DE-GANs is leaking-free when optimizing the saturating loss, as shown in Theorem 2 in Diffusion-GAN [35]. However, based on our analysis above, the commonly-used DE-GANs utilize StyleGAN2 as backbones which applies non-saturating loss, leading to optimization of Eq.(4) in DE-GANs. Therefore, the leaking

Method	FID (FFHQ-100)	FID (FFHQ-140K)
StyleGAN2 [18]	179.21	3.71
StyleGAN2 + ADA [16]	82.17	3.81
StyleGAN2 + Diff-Augment [44]	61.91	4.84
Diffusion-GAN (StyleGAN2 backbone) [35]	91.11	4.99

Table 1. Experiment results on FFHQ-100 and FFHQ-140K datasets (256×256). The FIDs (lower is better) are averaged over three runs; all standard deviations are less than 1%, relatively.

of augmentations still exists in Diffusion-GAN.

We also demonstrate that the leaking of augmentations problem exists in DE-GANs in practice. The experiments are conducted on the FFHQ dataset [18] using two different settings: a limited data setting (FFHQ-100) and a full data setting (FFHQ-140K). For the limited data setting, DE-GANs suffer from the heavy overfitting of D problem [13, 16, 44]. Although applying DA in DE-GANs can cause the leaking of augmentations problem, it can significantly address the overfitting of D problem. Since the overfitting of D is far more significant than the leaking of augmentations problem under the limited data setting, DA in DE-GANs can achieve great improvement compared with the baseline, with the result that the leaking of augmentations issue is often ignored. In contrast, for the full data setting, DE-GANs no longer suffer from the overfitting of D problem. In this case, the leaking of augmentations caused by applying DA in DE-GANs can decrease performance compared with the baseline. As shown in Table 1, compared with baseline StyleGAN2, all commonly-used DAs in DE-GANs achieve great improvement for the FFHQ-100 setting but decrease the performance for the FFHQ-140K setting, which shows that the leaking of augmentations problem exists in DE-GANs in practice.

3.2. Adaptive Negative Data Augmentation (ANDA)

We first directly apply negative data augmentation (NDA) [30] in DE-GANs to address the leaking of augmentations problem, and the results are shown in Table 2. By adding NDA on different DE-GANs, the FID only achieves limited improvement or even deteriorates. This is because the performance of NDA varies on different datasets and backbones of GANs [7]. In this case, directly applying NDA in DE-GANs could produce part of the in-distribution samples on the 100-shot-Obama dataset with the StyleGAN2 backbone. These in-distribution samples presented as the "fake" instance to D could lead to less real data being shown to D as the "real" instance during training. Consequently, directly applying NDA in DE-GANs causes more heavy overfitting of D in the data-efficient domain, yielding undesirable results.

To better utilize NDA and address the leaking of augmentations problem for DE-GANs, motivated by ADA [16] and APA [13], we propose a simple yet effective method called adaptive negative data augmentation (ANDA) for DE-GANs, as shown in Figure 2. ANDA adaptively augments the augmented generated data distribution with the augmented negative real data distribution during training, where the negative real data is produced by applying NDA on the real data. Specifically, the augmented negative real samples, i.e., potential leaking samples, will be presented to D as fake samples adaptively. We apply a hyperparameter λ to balance the NDA real samples and generated samples in the proposed ANDA and perform ANDA with the probability p, where $p \in [0, 1)$. The probability p should be intuitively adjusted according to the overfitting degree of D adaptively without any manual adjustment, irrespective of data scales and characteristics. In order to achieve this, following the ADA [16] and APA [13], we utilize an overfitting heuristic η which aims to quantify the overfitting degree of D as follows

$$\eta = \mathbb{E}(\text{sign}(D_{real})), D_{real} = \text{logit}(D(T(x))), \tag{5}$$

where sign() indicates the sign function that returns $+1$ for a non-negative input; -1, otherwise. Then, we follow the same step as in ADA [16] and APA [13] for using η to adjust p. The more serious overfitting of D, the less NDA-produced data should be present as fake in our proposed ANDA. Therefore, we design a novel adaptive strategy for the proposed ANDA. Specifically, the NDA will be applied with the probability $(1-p)$ or be skipped with the probability p. In this way, the strength of NDA can be adaptively controlled based on the degree of overfitting. This process can effectively avoid G learning these potential leaking samples, finally, preventing G from producing such samples and achieving a better result.

3.3. Theoretical Analysis

Let P_R^T be the distribution of augmented real samples, P_G^T be the distribution of augmented generated samples and $\hat{P}_R^T$ be the distribution of augmented NDA real samples. Let λ be the hyperparameter, which aims to balance the negative real samples and generated samples. For a given sample x, $D(x)$ represents the estimated probability of x being classified as real or fake. To evaluate the soundness of ANDA, we follow the theoretical analysis in APA [13] to

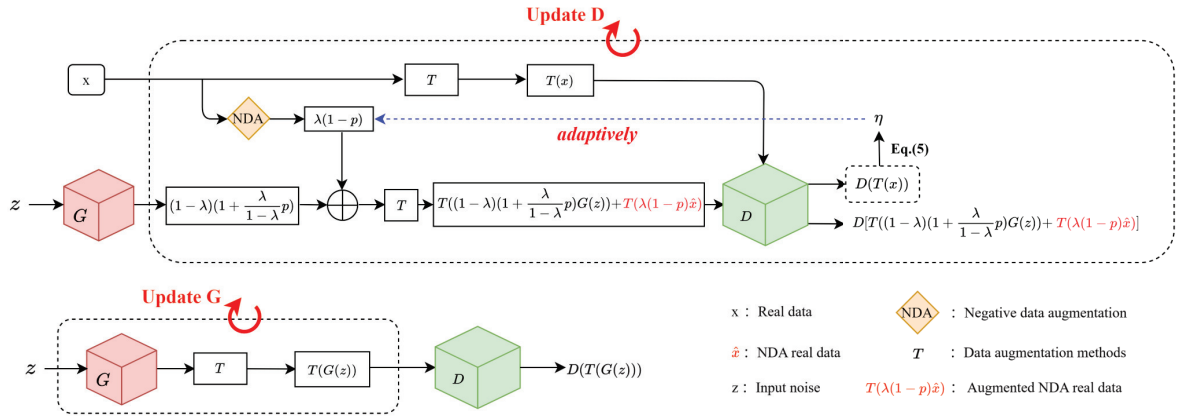

Figure 2. The overview of adaptive negative data augmentation (ANDA) for updating D (above) and G (below) in DE-GANs. For updating D, we augment the augmented distribution of generated data using the augmented distribution of negative real data adaptively. The negative real data is produced by applying NDA on the real data. Specifically, such augmented negative real data is adaptively presented to the discriminator as "fake" instances to avoid the leaking of augmentations during training. We introduce a hyperparameter λ to balance the negative real data and generated data, and apply an overfitting heuristic η to control the adaptive process. For updating G, according to [16, 44], the augmented generated samples are applied to update the parameters of G.

Method	FID (100-shot Obama)
StyleGAN2 + ADA	45.69
StyleGAN2 + ADA + NDA	44.68 (-1.01)
StyleGAN2 + Diff-Augment	46.87
StyleGAN2 + Diff-Augment + NDA	45.47 (-1.40)
Diffusion-GAN (StyleGAN2 backbone)	28.55
Diffusion-GAN (StyleGAN2 backbone) + NDA	29.66 (+1.11)

Table 2. FID score (lower is better) on directly applying the NDA to DE-GANs on the 100-shot Obama dataset [44]. By adding NDA, the FID only achieves limited improvement (red color) or even deteriorates (green color) on different DE-GANs. The FIDs are averaged over three runs; all standard deviations are less than 1%, relatively.

investigate ANDA within a non-parametric framework. By analyzing its convergence within the domain of probability density functions, a model is portrayed with limitless capacity. In an ideal scenario, the estimated probability distribution of augmented generated samples P_G^T should perfectly model the distribution of augmented real samples P_R^T without any bias when provided with sufficient capability and training time.

Given the adaptive adjustment of the probability p, we introduce an α representing the anticipated strength, approximating the impact of dynamic distribution adjustment over the whole training process. Considering that $p \in [0, 1)$, we have $0 \leq \alpha < p_{max} < 1$, where p_{max} is the maximum value of probability p during training. Consequently, the objective function $V(G, D)$ under saturating loss with ANDA can be formulated as:

$$\begin{aligned}\min_G \max_D V(G,D) = &\mathbb{E}_{T(x)\sim P_R^T}[\log D(T(x))] \\ &+ \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(1 - D(T(x)))] \\ &+ \mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(1 - D(T(x)))].\end{aligned} \tag{6}$$

The loss function of StyleGAN2 is the non-saturating loss. Therefore, the objective function $V_D(G, D)$ and $V_G(G, D)$ for the min-max game of ANDA on StyleGAN2 can be formulated as:

$$\begin{aligned}V_D(G,D) = &\mathbb{E}_{T(x)\sim P_R^T}[\log D(T(x))] \\ &+ \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(1 - D(T(x)))] \\ &+ \mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(1 - D(T(x)))], \\ V_G(G,D) = &-\mathbb{E}_{T(x)\sim P_G^T}[\log D(T(x))].\end{aligned} \tag{7}$$

To analyze the convergence of Eq.(7), following the proof of GANs [8], first, we develop a lemma for the objective function as follows.

Lemma 1. Given two types of objective function $\mathbb{E}_{x\sim\beta P+\gamma Q}[f(x)]$ and $\beta\mathbb{E}_{x\sim P}[f(x)] + \gamma\mathbb{E}_{x\sim Q}[f(x)]$, we have that

$$\mathbb{E}_{x\sim\beta P+\gamma Q}[f(x)] = \beta\mathbb{E}_{x\sim P}[f(x)] + \gamma\mathbb{E}_{x\sim Q}[f(x)], \quad (8)$$

Proof. See supplementary materials.

where β and γ are any scalable parameters; P and Q represent one kind of distribution, respectively. Based on Lemma 1, we consider the optimal discriminator for any given generator.

Proposition 1. *If the generator G is fixed, the optimal discriminator $D^*(T(x))$ for ANDA is:*

$$\begin{aligned} D^*(T(x)) =& P_R^T(T(x))/[P_R^T(T(x)) + \lambda(1-\alpha)\hat{P}_R^T(T(x)) \\ &+ (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))]. \end{aligned} \quad (9)$$

Proof. See supplementary materials.

Given the optimal discriminator $D^*(T(x))$, for the loss function of StyleGAN2, the goal of generator G is to minimize the $V_G(G, D^*)$ in Eq.(7). To analyze the convergence of $V_G(G, D^*)$ in Eq.(7), we should first provide the theoretical analysis for Eq.(6). We replace the $D(T(x))$ as $D^*(T(x))$ and apply Lemma 1 in Eq.(6), then we have that

$$\begin{aligned} C(G) =& \mathbb{E}_{T(x)\sim P_R^T}[\log D^*(T(x))] \\ &+ \lambda(1-\alpha)\mathbb{E}_{T(x)\sim \hat{P}_R^T}[\log(1-D^*(T(x)))] \\ &+ (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)\mathbb{E}_{T(x)\sim P_G^T}[\log(1-D^*(T(x)))]. \end{aligned} \quad (10)$$

Then, let us consider the optimization of $C(G)$ in Eq.(10) trained with the proposed ANDA.

Proposition 2. *Given the optimal discriminator $D^*(T(x))$, the minimization of $C(G)$ in Eq.(10) can be regarded as:*

$$\begin{aligned} C(G) =& 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T) \\ &- 2\log 2. \end{aligned} \quad (11)$$

Proof. See supplementary materials.

Then, for $V_G(G, D)$ in Eq.(7), we replace $D(T(x))$ as $D^*(T(x))$. Based on the proposition 2, we investigate the item $\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T)$, and the minimization of $V_G(G, D^*)$ with the proposed ANDA is shown in Theorem 1.

Theorem 1. *Given the optimal discriminator $D^*(T(x))$, the minimization of $V_G(G, D^*)$ in Eq.(7) can be regarded as:*

$$\begin{aligned} V_G(G, D^*) =& \frac{1}{(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)} \times \\ &[\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T) \\ &- 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T)]. \end{aligned} \quad (12)$$

Proof. See supplementary materials.

According to f-GAN [26], both the terms **KL** and **JS** in Eq.(12) are f-divergences in GANs. Based on the proofs of Theorem 1 developed in NDA-GAN [30], we investigate both **KL** divergence items $\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T)$ and $\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_R^T \,||\, P_R^T)$, as well as both **JS** divergence items $2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T)$ and $2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_R^T)$, respectively. Then, we can conclude that both $\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T)$ and $2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T)$ items in Eq.(12) leads to the optimization of the **KL-2JS** divergence between P_G^T and $P_R^T \setminus \left\{\hat{P}_R^T\right\}$. This goal is similar to the original DE-GANs shown in Eq.(4). At the same time, it avoids the optimization between P_G^T and potential leaking samples distribution $\hat{P}_R^T$, which demonstrates that ANDA in DE-GANs does not influence the convergence of G and can alleviate the leaking of augmentations problem. The detailed proof of the theory and the training algorithm for ANDA is shown in supplementary materials.

4. Experiment

We demonstrate the superiority of ANDA with several state-of-the-art DE-GANs on widely used datasets, i.e., 100-shot Obama, 100-shot Panda, 100-shot Grumpy Cat, AnimalFace Dog, AnimalFace Cat in [44], and FFHQ [18] datasets. All of these datasets are commonly used without limitations. We conduct all the experiments on a workstation with four NVIDIA V100 GPUs. **More details of the experiments can be found in the supplementary materials.**

4.1. Datasets Preparation and Implementation Details

We follow ADA [16] and Diff-Augment [44] to prepare the dataset. For FFHQ, according to ADA, the images are resized to 256×256. FID is measured using 50K generated samples; the full training set (70K) is used as the reference distribution. Furthermore, we set batchsize as 64 for the experiments. For several low-shot datasets, the resolution of images is 256×256. FID is measured using 5k generated samples; the training set is the reference distribution. For NDA, we select the best NDA strategy Jigsaw developed in NDA-GAN [30] for our proposed ANDA. The hyperparameter λ is set as 0.2 for all experiments. For the implementation of Diffusion-GAN [35], according to [35], Diff-Augment [44] is applied alongside Diffusion as the noise augmentation to enhance the training of GANs in the data-efficient domain.

Method	MA	Pre-training?	100-shot			Animal-Face	
			Obama	Grumpy Cat	Panda	Cat	Dog
Scale/shift [25]	No	Yes	50.72	34.20	21.38	54.83	83.04
MineGAN [33]	No	Yes	50.63	34.54	14.84	54.45	93.03
TransferGAN [34]	No	Yes	48.73	34.06	23.20	52.61	82.38
TransferGAN + DA [44]	Yes	Yes	39.85	29.77	17.12	49.10	65.57
FreezeD [24]	No	Yes	41.87	31.22	17.95	47.70	70.46
StyleGAN2 [18]	No	No	80.20	48.90	34.27	71.71	131.90
StyleGAN2* [18]	Yes	No	65.57	39.92	22.08	51.66	77.96
StyleGAN2 + Diff-Augment [44]	Yes	No	46.87	27.08	12.06	42.44	58.85
+ ANDA	Yes	No	**38.61**	**24.31**	**10.63**	**37.38**	**49.66**
StyleGAN2 + ADA [16]	Yes	No	45.69	26.62	12.90	40.77	56.83
+ ANDA	Yes	No	**39.66**	**25.11**	**11.72**	**38.15**	**54.45**
Diffusion-GAN (StyleGAN2 backbone) [35]	Yes	No	28.55	21.87	8.69	33.18	68.15
+ ANDA	Yes	No	**26.40**	**19.01**	**7.88**	**29.26**	**65.74**
InsGen [37]	Yes	No	32.42	22.01	9.85	33.01	44.93
+ ANDA	Yes	No	**23.55**	**18.01**	**8.00**	**23.87**	**39.20**

Table 3. FID score (lower is better) on several low-shot datasets (256×256). We follow the setting as in [44]. MA means Massive Augmentation, which has the same meaning as in Genco [5]. For a fair comparison, the FIDs are averaged over three runs; all standard deviations are less than 1%, relatively. The results of StyleGAN2* and Diffusion-GAN (StyleGAN2 backbone) are run by ourselves based on their official open source codes.

4.2. Results on Low-shot Datasets

The results on 256×256 low-shot datasets are shown in Table 3. We add ANDA on three types of DE-GANs, i.e., StyleGAN + Diff-Augment, StyleGAN2 + ADA and Diffusion-GAN (StyleGAN2 backbone). The results of the Diffusion-GAN (StyleGAN2 backbone) are run by ourselves based on the official open source codes[1]. StyleGAN2 + Diff-Augment does not consider avoiding the leaking of augmentations problem in the design of the augmentation method. Therefore, adding ANDA can achieve great improvement for StyleGAN2 + Diff-Augment. For StyleGAN2 + ADA and Diffusion-GAN (StyleGAN2 backbone), although they consider avoiding the leaking of augmentations problem in designing their augmentation methods, adding ANDA can still achieve further improvement on StyleGAN2 + ADA and Diffusion-GAN (StyleGAN2 backbone). More generated images are shown in the supplementary materials.

To further demonstrate the generalization ability of ANDA on DE-GANs, we also apply ANDA to the more advanced DE-GANs, i.e., InsGen [37], and the results are shown in Table 3. By adding ANDA, InsGen obtains lower FIDs compared with the baseline, which shows that ANDA can further increase the performance on more advanced DE-GANs. The images generated by InsGen + ANDA can be found in the supplementary materials.

[1] https://github.com/Zhendong-Wang/Diffusion-GAN

Method	Seconds per 1K images
Diffusion-GAN (StyleGAN2 backbone)	24.82
+ANDA	**25.17**

Table 4. The training time on the 100-shot Obama dataset. The results are calculated by averaging over ten runs on an NVIDIA V100 GPU with batch size 64. All standard deviations are less than 1%, relatively.

4.3. Results on FFHQ Dataset

For the experiments on the FFHQ dataset, we apply two different experimental settings, i.e., a limited data setting (FFHQ-100) and a full data setting (FFHQ-140K), to better illustrate that ANDA can alleviate the leaking of augmentations problem in DE-GANs. The results are shown in Table 5. Adding ANDA in DE-GANs can improve the performance in both settings. Particularly, ANDA can increase the performance of DE-GANs on the FFHQ-140K setting, which indicates that ANDA can mitigate the leaking of augmentations problem in DE-GANs.

4.4. Computational Cost

The training time on the 100-shot-Obama dataset (256×256) with Diffusion-GAN (StyleGAN2 backbone) is shown in Table 4. The computational increase by adding ANDA in DE-GANs is negligible.

Method	FID (FFHQ-100)	FID (FFHQ-140K)
StyleGAN2 [18]	179.21	3.71
StyleGAN2 + ADA [16]	82.17	3.81
+ ANDA	**71.42**	**3.69**
StyleGAN2 + Diff-Augment [44]	61.91	4.84
+ ANDA	**53.74**	**4.27**
Diffusion-GAN (StyleGAN2 backbone) [35]	91.11	4.99
+ ANDA	**61.66**	**4.85**

Table 5. FID score (lower is better) on 256×256 FFHQ dataset. We perform experiments on 100 and 140K training samples on the FFHQ dataset. Massive Augmentation (MA) is applied in all of the methods. For a fair comparison, FID is measured using 50K generated samples. The FIDs are averaged over three runs; all standard deviations are less than 1%, relatively.

λ	0.1	0.2	0.4	0.8
FID (100-shot Obama)	39.78	**39.66**	40.67	41.28

Table 6. Experiment results by selecting different hyperparameters λ in ANDA on the StyleGAN2 + ADA + ANDA method. Here, we report the FID (lower is better) on the 100-shot Obama dataset. The FIDs are averaged over three runs; all standard deviations are less than 1%, relatively.

NDA methods in ANDA	Jigsaw	Stitching	Mixup	Cutmix
FID (100-shot Obama)	**39.66**	40.74	40.61	40.77

Table 7. Experiment results by selecting different NDA methods in ANDA upon the StyleGAN2 + ADA + ANDA method. Here, we report the FID (lower is better) on the 100-shot Obama dataset. The hyperparameter λ is set as 0.2 for all the experiments. The FIDs are averaged over three runs; all standard deviations are less than 1%, relatively.

4.5. Ablation Study

Ablation study on applying different values of the hyperparameter λ (defined in Figure 2). We conduct an ablation study by selecting the different values of the hyperparameter λ in ANDA with the StyleGAN2 + ADA backbone on the 100-shot Obama dataset, and the results are shown in Table 6. Setting $\lambda = 0.2$ achieves the best performance.

Ablation study on applying different NDA methods in proposed ANDA. We conduct an ablation study by selecting different NDA methods, i.e., Jigsaw, Stitching, Mixup, and Cutmix as in NDA-GAN [30], for the proposed ANDA upon the StyleGAN2 + ADA backbone. The results of the 100-shot Obama dataset are shown in Table 7. Jigsaw achieves better performance compared with other NDA methods in the proposed ANDA.

Ablation study on the effectiveness of ANDA alleviating the leaking of augmentations problem in DE-GANs. To further demonstrate the proposed ANDA can alleviate the leaking of augmentations problem, rather than other problems, in DE-GANs. We conduct an ablation study by applying the ANDA to the StyleGAN2, in which no DA method is applied in this case. The results are shown in Table 8. It is clear that directly applying ANDA to StyleGAN2 can slightly decrease the performance compared with the baseline, which shows that ANDA can not directly alleviate the overfitting of D problem. On the contrary, the improvements caused by ANDA in Tables 3 and 5 demonstrate that ANDA can alleviate the leaking of augmentations problem in DE-GANs.

Method	FID (100-shot Obama)
StyleGAN2 [18]	65.57
+ANDA	**65.83**

Table 8. FID score (lower is better) on applying ANDA to StyleGAN2 method. Massive Augmentation (MA) is applied in all of the methods. The FIDs are averaged over three runs; all standard deviations are less than 1%, relatively.

5. Conclusion

In this paper, we propose a simple yet effective method called adaptive negative data augmentation (ANDA) for DE-GANs, which can effectively alleviate the leaking of augmentations problem with a negligible computational cost increase. Experiments on several low-shot datasets with different DE-GANs demonstrate that ANDA can effectively address the leaking of augmentations problem and achieve better performance. The discussion of Boarder Impact can be found in the supplementary materials.

Acknowledgement

We are grateful for use of the computing resources from the Northern Ireland High Performance Computing (NI-HPC) service funded by EPSRC (EP/T022175).

References

[1] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 3

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017. 2

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[4] Tianlong Chen, Yu Cheng, Zhe Gan, Jingjing Liu, and Zhangyang Wang. Data-efficient gan training beyond (just) augmentations: A lottery ticket perspective. In *NeurIPS*, 2021. 1, 2

[5] Kaiwen Cui, Jiaxing Huang, Zhipeng Luo, Gongjie Zhang, Fangneng Zhan, and Shijian Lu. Genco: Generative co-training for generative adversarial networks with limited data. In *AAAI*, 2022. 7

[6] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, 2015. 2

[7] Nikolaos Dionelis. Omasgan: Out-of-distribution minimum anomaly score gan for sample generation on the boundary. *arXiv preprint arXiv:2110.15273*, 2021. 3, 4

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014. 1, 2, 3, 5

[9] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, 2017. 2

[10] Tianyu Guo, Chang Xu, Jiajun Huang, Yunhe Wang, Boxin Shi, Chao Xu, and Dacheng Tao. On positive-unlabeled classification in gan. In *CVPR*, 2020. 2

[11] 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 *NeurIPS*, 2017. 2

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[13] Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. Deceive d: Adaptive pseudo augmentation for gan training with limited data. In *NeurIPS*, 2021. 1, 4

[14] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan. In *NeurIPS*, 2021. 1

[15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 1, 2

[16] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[17] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1

[18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *CVPR*, 2020. 1, 2, 3, 4, 6, 7, 8

[19] Ziqiang Li, Chaoyue Wang, Heliang Zheng, Jing Zhang, and Bin Li. Fakeclr: Exploring contrastive learning for solving latent discontinuity in data-efficient gans. In *ECCV*, 2022. 1, 2

[20] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *CVPR*, 2020. 2

[21] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017. 2

[22] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 2

[23] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *ICLR*, 2018. 1, 2

[24] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning gans. *arXiv preprint arXiv:2002.10964*, 2020. 7

[25] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *ICCV*, 2019. 7

[26] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *NeurIPS*, 2016. 6

[27] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 2

[28] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016. 2

[29] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 2

[30] Abhishek Sinha, Kumar Ayush, Jiaming Song, Burak Uzkent, Hongxia Jin, and Stefano Ermon. Negative data augmentation. In *ICLR*, 2021. 1, 3, 4, 6, 8

[31] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. On data augmentation for gan training. *IEEE Transactions on Image Processing*, 30:1882–1897, 2021. 1, 2, 3

[32] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016. 2

[33] Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. Minegan: effective knowledge transfer from gans to target domains with few images. In *CVPR*, 2020. 7

[34] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *ECCV*, 2018. 7

[35] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *ICLR*, 2023. 1, 2, 3, 4, 6, 7, 8

[36] Ceyuan Yang, Yujun Shen, Yinghao Xu, Deli Zhao, Bo Dai, and Bolei Zhou. Improving gans with a dynamic discriminator. In *NeurIPS*, 2022. 1

[37] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. In *NeurIPS*, 2021. 7

[38] Mengping Yang, Zhe Wang, Ziqiu Chi, and Yanbing Zhang. Fregan: Exploiting frequency components for training gans under limited data. In *NeurIPS*, 2022. 2

[39] Ning Yu, Guilin Liu, Aysegul Dundar, Andrew Tao, Bryan Catanzaro, Larry S Davis, and Mario Fritz. Dual contrastive loss and attention for gans. In *ICCV*, 2021. 1

[40] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, 2019. 2

[41] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 2

[42] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *ICLR*, 2020. 1, 3

[43] Zhaoyu Zhang, Mengyan Li, and Jun Yu. On the convergence and mode collapse of gan. In *SIGGRAPH Asia 2018 Technical Briefs*, 2018. 1

[44] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training. In *NeurIPS*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[45] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans. In *AAAI*, 2021. 1, 3