Improving the Training of Data-Efficient GANs via Quality Aware Dynamic Discriminator Rejection Sampling

Zhaoyu Zhang[1], Yang Hua[1], Guanxiong Sun[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-Efficient Generative Adversarial Nets (DE-GANs) have become more and more popular in recent years. Existing methods apply data augmentation, noise injection and pre-trained models to maximumly increase the number of training samples thus improving the training of DE-GANs. However, none of these methods considers the sample quality during training, which can also significantly influence the training of DE-GANs. Focusing on sample quality during training, in this paper, we are the first to incorporate discriminator rejection sampling (DRS) into the training process and introduce a novel method, called quality aware dynamic discriminator rejection sampling (QADDRS). Specifically, QADDRS consists of two steps: (1) the sample quality aware step, which aims to obtain the sorted critic scores, i.e., the ordered discriminator outputs, on real/fake samples in the current training stage; (2) the dynamic rejection step that obtains dynamic rejection number N, where N is controlled by the overfitting degree of discriminator (D) during training. When updating the parameters of D, the N high critic score real samples and the N low critic score fake samples in the minibatch are rejected dynamically based on the overfitting degree of D. As a result, QADDRS can avoid D becoming overly confident in distinguishing both real and fake samples, thereby alleviating the overfitting of D issue during training. Extensive experiments on several datasets demonstrate that integrating QADDRS into different DE-GANs can achieve better performance and deliver state-of-the-art results. Codes are available at https://github.com/zzhang05/QADDRS.

1. Introduction

Over the past decade, Generative Adversarial Nets (GANs) [12] have demonstrated impressive performance in synthesizing images and videos [20, 21, 23, 59, 65]. However, this success heavily depends on access to large-scale datasets, which can be costly, time-consuming, and even impractical to collect and clean in certain domains. As a result, data-efficient GANs (DE-GANs) [28, 38, 39] have garnered significant attention.

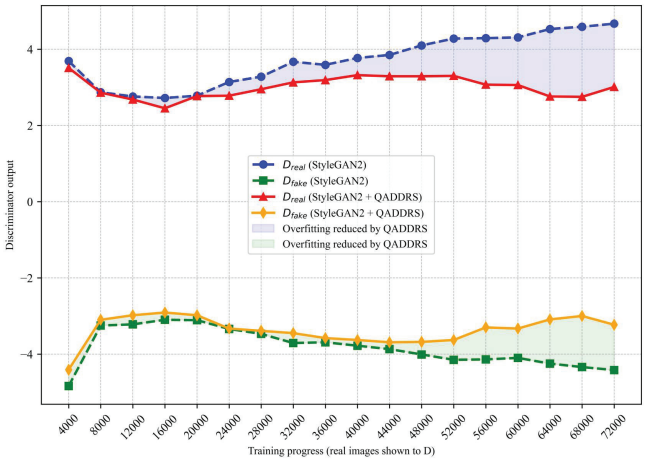

Figure 1. Comparison of discriminator outputs on both real and fake samples with StyleGAN2 and StyleGAN2 + QADDRS using the 100-shot Obama dataset. Applying QADDRS to StyleGAN2 can effectively prevent D from being overly confident in distinguishing both real and fake samples, thereby alleviating the overfitting of D issue.

Recent studies have applied techniques such as data augmentation [8, 9, 19, 22, 66], noise injection [55, 63] and pre-trained models [25, 44] to improve the DE-GANs training. All of these methods aim to maximumly increase the number of training samples under data-efficient scenarios. However, none of these methods considers the sample quality during training, which can also significantly influence the DE-GANs training. For instance, as pointed out in several existing approaches [22, 55, 64], the highly undesirable quality samples produced by data augmentation or noise injection can be detrimental to DE-GANs training.

Focusing on sample quality during training, in this paper, we are the first to introduce discriminator rejection sampling (DRS) [5] into the training process and propose a novel method, termed *quality aware dynamic discrimina-*

tor rejection sampling (QADDRS). Specifically, QADDRS comprises two steps: (1) the sample quality aware step that obtains the ranked critic scores, i.e., the sorted discriminator (D) outputs, on real/fake samples to measure the quality of the real/fake samples in the current training stage and (2) the dynamic rejection step which is determined by the dynamic rejection number N, where N adapts based on the overfitting degree of D. During updating the parameters of D process, the N real samples with the high critic scores and the N fake samples with the low critic score in the minibatch are dynamically rejected according to the overfitting degree of D. As illustrated in Figure 1, QADDRS can effectively prevent D from becoming overly confident in distinguishing real and fake samples, thereby alleviating the overfitting of D issue during training.

To sum up, the key contributions are shown as follows: (1) We introduce DRS into the GANs training process and propose a novel method named quality aware dynamic discriminator rejection sampling (QADDRS) to improve DE-GANs training. As far as we are aware, QADDRS is the first approach that considers sample quality during DE-GANs training. (2) We present the theoretical analysis of QADDRS in GANs training, proving its robustness and rationality. Furthermore, we observe that applying QADDRS to DE-GANs can avoid D becoming overly confident in distinguishing both real and fake samples, thereby alleviating the overfitting of D issue during training. (3) Experiments on low-shot [66] and FFHQ [23] datasets with different methods demonstrate that QADDRS can improve performance across various DE-GANs and deliver state-of-the-art performance over existing GANs and diffusion models.

2. Related Work

2.1. Generative Adversarial Nets (GANs)

In the past ten years, generative adversarial nets (GANs) [12] have been among the most popular generative models [41, 50]. GANs consist of two networks: a generator (G) and a discriminator (D), which engage in a min-max game. The G seeks to generate realistic-looking synthetic samples from noise input, while the D attempts to distinguish whether the input sample originates from the G or from real samples. GANs are known to suffer from training instability, often resulting in low-quality and limited diversity in generated data [62]. To overcome this challenge, different approaches have been presented, including more advanced architectures [6, 34, 35, 42, 58, 61], more stable objective functions [4, 14, 15, 33, 43], and improved training strategies [10, 20, 28, 31, 32, 57, 60] to generate more realistic and diverse samples.

2.2. Data-Efficient GANs

Recently, training Data-Efficient GANs (DE-GANs) has gained increasing significance. Studies [8, 9, 19, 22, 66] have shown that DE-GAN training often suffers from discriminator (D) overfitting. To address this issue, different approaches have been introduced, including novel data augmentation techniques [8, 9, 19, 22, 66], improved regularization methods for D [11, 30, 48], noise injection [55, 63], and pre-trained models [25, 44].

2.3. Rejection Sampling in Generative Models

In recent years, rejection sampling has played a significant role in increasing the generated image quality as a post-training technique. Rejection sampling is commonly used in several kinds of generative models, such as GANs [5], diffusion models [37] and IMLE [51]. Among these methods, the most well-known method in GANs is discriminator rejection sampling (DRS). DRS has inspired the development of numerous techniques in GANs, such as MH-GAN [49], DDLS [7], DOT [46], DGflow [2] and LatentRS [18]. Although these studies have achieved great success in rejecting lower-quality generated samples, they are all post-training methods and cannot directly benefit the training of DE-GANs. In this study, we are the first to incorporate the DRS in the training stage to improve the DE-GANs training by considering sample quality.

3. Methodology

3.1. Quality Aware Dynamic Discriminator Rejection Sampling (QADDRS)

As discussed in §2.3, existing discriminator rejection sampling (DRS) methods are rejecting low-quality generated samples in the post-training process to improve performance. However, it is infeasible to directly apply DRS to the training process, as it can cause the number of real and generated samples for the input of D to become unbalanced during training.

In this paper, we propose a novel approach called quality aware dynamic discriminator rejection sampling (QADDRS) to improve DE-GANs training by considering sample quality. The overview of applying QADDRS to update both G and D is shown in Figure 2. Specifically, when updating G, all of the fake samples are used to update the parameters of G, following the prior approaches [22, 66]. In contrast, when updating D, the N high critic score real samples and the N low critic score fake samples in the minibatch are dynamically rejected, and the rest of the real and fake samples are used to update the parameters of D. Consequently, QADDRS can prevent D from becoming overly confident in distinguishing both real and fake samples, thereby alleviating the overfitting of D issue during training. The selection of high critic score real samples and low critic score

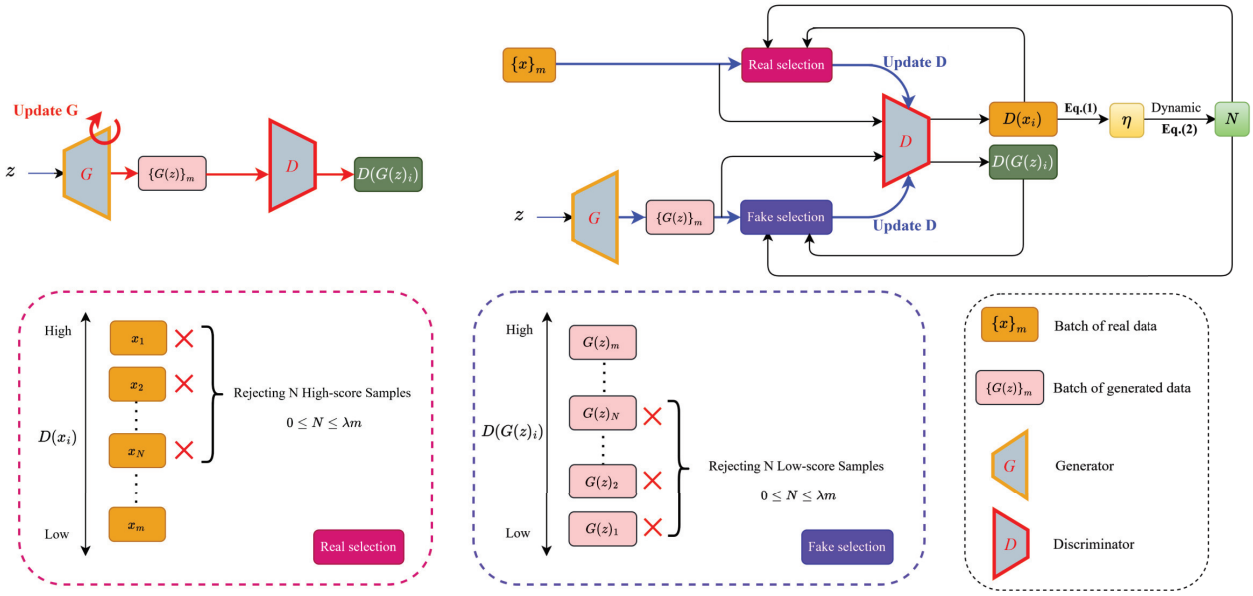

Figure 2. An overview of quality aware dynamic discriminator rejection sampling (QADDRS) for updating G (left) and updating D (right) in DE-GANs training. When updating G, all of the generated samples are used to update the parameters of G. When updating D, the N high critic score real samples and N low critic score fake samples in the minibatch are dynamically rejected, and only the rest of the real and fake samples are used to update the parameters of D. The selection of high critic score real samples and low critic score fake samples is guided by the discriminator outputs on both real and fake data during training. *Best viewed in color.*

fake samples is guided by the sample quality-aware step, while the rejection of N real and fake samples is controlled by the dynamic rejection step.

For the sample quality aware step, the quality of real/fake samples can be represented by different critic scores on real/fake samples. Based on this, we sort real/fake samples via critic scores, which are based on the discriminator outputs on real/fake data in the current training stage.

For the dynamic rejection step, the dynamic rejection number N is introduced to control sample rejection. It is obvious that rejecting real samples in QADDRS is highly undesirable, especially under data-efficient scenarios, which could cause a further reduction of training samples. To address this, we introduce a dynamic mechanism for the QADDRS by setting the rejection number N to be adjusted dynamically according to the degree of overfitting without manual tuning. To achieve this, following ANDA [64], APA [19] and ADA [22], we first utilize an overfitting heuristic η which can quantify the D's overfitting degree as follows:

$$\eta = \mathbb{E}(\text{sign}(D_{real})), D_{real} = \text{logit}(D(x)), \qquad (1)$$

where $\text{sign}(\cdot)$ denotes the sign function, which returns $+1$ for a non-negative input; -1, otherwise. Next, we follow ADA, APA and ANDA by using η to obtain the overfitting probability p. The larger value of p means the greater overfitting degree. Next, we set the initial rejection number N as 0, and adjust its value based on the overfitting probability p as follows:

$$N = \begin{cases} N+1 & (p) \\ N-1 & (1-p) \end{cases} \qquad (2)$$

where N is adjusted to $N+1$ under the probability p and is adjusted to $N-1$ under the probability $1-p$. We introduce

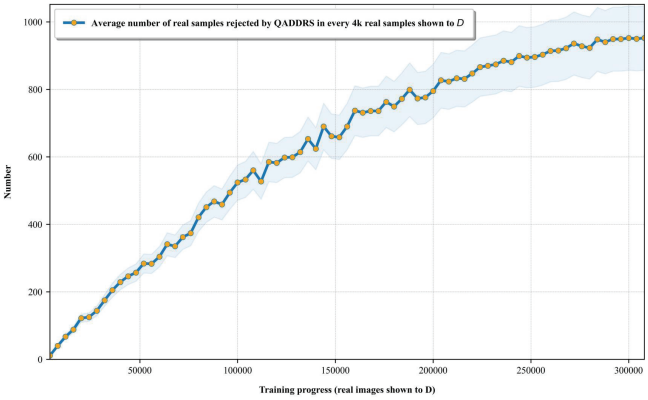

Figure 3. The curve of the average number of real samples rejected by QADDRS ($\lambda = 0.25$) when updating D for every 4K fake samples shown to D during training. The experiments are conducted on the 100-shot Obama dataset using StyleGAN2.

a hyperparameters λ and set the range of N as $N_{min} = 0$ and $N_{max} = \lambda \times$ batchsize. To further demonstrate how the number N evolves during the training process, the value of N in every 4K real images shown to D during training can be found in Figure 3.

In summary, employing QADDRS in DE-GANs can be formulated as:

$$\begin{aligned} V_D(G,D) &= \mathbb{E}_{x\sim P_R}[\tilde{g}_1(D(x_{m-N}))] + \mathbb{E}_{x\sim P_G}[\tilde{g}_2(D(x_{m-N}))], \\ V_G(G,D) &= -\mathbb{E}_{x\sim P_G}[\tilde{f}((D(x_m)))], \end{aligned} \tag{3}$$

where x_m is a batch of samples, m denotes the total samples number in a batch, and $\tilde{g}_1$, $\tilde{g}_2$ and $\tilde{f}$ are scalar-to-scalar functions. For the StyleGAN2 [23] with non-saturating loss, $\tilde{g}_1(x) = \log(x)$, $\tilde{g}_2(x) = \log(1-x)$ and $\tilde{f}(x) = \log(x)$. For the FastGAN [30] with hinge loss, $\tilde{g}_1(x) = \min(0, -1+x)$, $\tilde{g}_2(x) = \min(0, -1-x)$ and $\tilde{f}(x) = x$.

3.2. Theoretical Analysis

3.2.1 QADDRS in GANs training

To further demonstrate how QADDRS can influence GANs training in theory, we first revisit the hinge loss in GANs shown below:

$$\begin{aligned} V_D(D,G) &= \mathbb{E}_{x\sim P_R}[\min(0, -1 + D(x))] \\ &\quad + \mathbb{E}_{x\sim P_G}[\min(0, -1 - D(x))], \\ V_G(D,G) &= -\mathbb{E}_{x\sim P_G}[D(x)]. \end{aligned} \tag{4}$$

For Eq.(4), $D(x) > 1$ on the real data or $D(G(z)) < -1$ on the fake data can lead to the loss value becoming 0, resulting in no gradient for updating D. Therefore, the key essence of hinge loss in GANs training is to reject the very high critic score real samples (score $>$ 1) and very low critic score fake samples (score $<$ -1). Following this, our proposed QADDRS further rejects the real and fake samples based on the overfitting degree of D to enhance the DE-GANs training. We then consider how QADDRS influences the distributions of real data (P_R) and fake data (P_G) in GANs training as follows.

According to Figure 2, QADDRS can be regarded as involving two actions, i.e., rejecting low critic score fake samples and rejecting high critic score real samples. Because rejecting real/fake samples in QADDRS is performed dynamically, the key operations of QADDRS on the distributions are re-weighting both real and fake samples for their data distributions, respectively. Consequently, in rejecting the low critic score fake samples action, applying QADDRS can transform the original fake data distribution (P_G) to the re-weighted fake data distribution ($w_1 P_G$). Similarly, in rejecting the high critic score real samples action, utilizing QADDRS can transform the original real data distribution (P_R) to the re-weighted real data distribution ($w_2 P_R$).

3.2.2 Convergence analysis of QADDRS in StyleGAN2 and FastGAN

Based on the analysis in §3.2.1, we then analyze the QADDRS on both StyleGAN2 with non-saturating loss and FastGAN with hinge loss as follows.

1. For the StyleGAN2 with non-saturating loss, QADDRS can effectively introduce the advantages of hinge loss into the non-saturating loss via dynamically rejecting the N high critic score real samples and the N low critic score fake samples. Following §3.2.1 and [3], the optimal discriminator of QADDRS with non-saturating loss is defined as:

$$D^*(x) = \frac{w_2(x)P_R(x)}{w_2(x)P_R(x) + w_1(x)P_G(x)}. \tag{5}$$

With D^* as the optimal discriminator, according to Theorem 2.5 in [3], training the G with non-saturating loss can be expressed as:

$$V_G(G, D^*) = \mathbf{KL}(w_1 P_G \,||\, w_2 P_R) - 2\mathbf{JS}(w_1 P_G \,||\, w_2 P_R), \tag{6}$$

where **KL** and **JS** denote the Kullback-Leibler divergence and the Jensen-Shannon divergence, respectively. Eq.(6) shows that applying QADDRS to the non-saturating loss can lead to the minimization of **KL-2JS** divergence between the $w_1 P_G$ and $w_2 P_R$. Then, according to [3], this minimization process depends on the supports of $w_1 P_G$ and $w_2 P_R$. Previous studies [6, 13, 21, 45] have established that re-weighting operations applied to distributions do not alter their support. Therefore, the minimization of Eq.(6) corresponds to the minimization of **KL-2JS** divergence between the P_G and P_R, which demonstrates that applying QADDRS to non-saturating loss does not influence convergence.
2. For the FastGAN with hinge loss, QADDRS can further take advantage by dynamically rejecting the N high critic score real samples and the N low critic score fake samples if all $D(x)$ and $D(G(z)$ are in $[-1, 1]$. According to the §3.2.1 and [29, 35, 47], optimizing the hinge loss with QADDRS is equivalent to minimizing the reverse **KL** divergence term $\mathbf{KL}(w_1 P_G \,||\, w_2 P_R)$. With the fact that the supports of $w_1 P_G$ and $w_2 P_R$ are the same as the supports of P_G and P_R, applying QADDRS to hinge loss does not influence the convergence.

3.2.3 Analysis of QADDRS in DE-GANs

According to the theory presented in [64], data augmentation (DA) is commonly used in DE-GANs, which transforms the real and fake data distributions as P_R^T and P_G^T, respectively. As a result, applying QADDRS can result in the optimization between the $w_1 P_G^T$ and $w_2 P_R^T$ for both StyleGAN2 with non-saturating loss and FastGAN with hinge

Method	Backbone	MA	Pre-training	100-shot			Animal-Face	
				Obama	Grumpy Cat	Panda	Cat	Dog
EDM [24]	Diffusion	Yes	No	51.30	36.90	23.70	48.60	100.10
EDM + DA [24]	Diffusion	Yes	No	37.10	29.94	10.81	36.88	57.14
Patch Diffusion [54]	Diffusion	Yes	No	41.47	30.89	13.25	43.71	72.17
Scale/shift [40]	GAN	No	Yes	50.72	34.20	21.38	54.83	83.04
MineGAN [53]	GAN	No	Yes	50.63	34.54	14.84	54.45	93.03
TransferGAN [52]	GAN	No	Yes	48.73	34.06	23.20	52.61	82.38
TransferGAN + DA [66]	GAN	Yes	Yes	39.85	29.77	17.12	49.10	65.57
FreezeD [36]	GAN	No	Yes	41.87	31.22	17.95	47.70	70.46
StyleGAN2 [23]	GAN	No	No	80.20	48.90	34.27	71.71	131.90
FakeCLR [28]	GAN	Yes	No	26.95	19.56	8.42	26.34	42.02
FastGAN [30]	GAN	Yes	No	35.80	25.75	9.70	33.85	52.46
FreGAN [58]	GAN	Yes	No	33.39	24.93	8.97	31.05	47.85
AugSelf-StyleGAN2 [17]	GAN	Yes	No	26.00	19.81	8.36	30.53	48.19
CHAIN [39]	GAN	Yes	No	20.94	17.57	6.93	19.58	30.88
NICE [38]	GAN	Yes	No	20.08	15.63	8.18	22.70	28.65
Projected GAN [44]	GAN	Yes	Yes	11.21	15.80	3.98	18.01	17.88
StyleGAN2* [23]	GAN	Yes	No	65.57	39.92	25.08	51.66	87.96
+ QADDRS	GAN	Yes	No	**53.54**	**35.77**	**20.59**	**47.87**	**73.29**
StyleGAN2 + Diff-Augment [66]	GAN	Yes	No	46.87	27.08	12.06	42.44	58.85
+ QADDRS	GAN	Yes	No	**41.27**	**26.53**	**10.83**	**38.15**	**51.83**
StyleGAN2 + ADA [22]	GAN	Yes	No	45.69	26.62	12.90	40.77	56.83
+ QADDRS	GAN	Yes	No	**40.83**	**26.27**	**11.99**	**36.25**	**53,01**
InsGen + ADA [56]	GAN	Yes	No	32.42	22.01	9.85	33.01	44.93
+ QADDRS	GAN	Yes	No	**25.41**	**19.01**	**7.99**	**23.69**	**35.01**
Diffusion-Projected GAN (FastGAN) [55]	GAN	Yes	Yes	10.54	15.13	3.39	17.86	17.22
+ QADDRS	GAN	Yes	Yes	**9.93**	**14.72**	**3.08**	**16.94**	**16.57**

Table 1. FID [16] scores (lower is better) of different methods on 256×256 low-shot datasets. MA donates Massive Augmentation [9], i.e., xflipping. For a fair comparison, FID is computed using 5K generated samples, with the entire dataset serving as the reference distribution. The results are averaged over three evaluations, with relative standard deviations below 1%.

loss. As the supports of $w_1 P_G^T$ and $w_2 P_R^T$ are the same as the supports of P_R^T and P_G^T, utilizing QADDRS in the DE-GANs does not influence convergence. Furthermore, DE-GANs are known to suffer from the leaking of augmentations issue [22, 64]. In this case, QADDRS can also reweight the out-of-distribution samples for both real and fake data to alleviate the leaking issue.

4. Experiments

4.1. Datasets and Implementation Details

We select FFHQ [23] and low-shot [66] datasets for the experiments. The FFHQ dataset includes 70K high-resolution face images. In line with FakeCLR [28], we preprocess and resize the FFHQ dataset to 256×256 and select different subsets consisting of 100, 1K, 2K, and 5K images for all experiments. Low-shot datasets (256×256) consist of five sets (i.e., Obama, Grumpy Cat, Panda, Animal-Face Cat, and Animal-Face Dog) with 100, 100, 100, 160, and 389 training samples, respectively. The commonly used Fréchet Inception Distance (FID) [16], CLIP FID (FID_{CLIP}) [27], and Precision and Recall [26] are applied as the evaluation metrics. Following prior methods [22, 66], the complete dataset serves as the reference distribution for all metric evaluations.

The implementation details are shown as follows. For the experiments on low-shot datasets with StyleGAN2, StyleGAN2 + Diff-Augment and StyleGAN2 + ADA, we set the batchsize as 4 and the overall training duration as 2000K images. For the experiments on low-shot datasets with InsGen + ADA, we set the batchsize as 64 and the overall training duration as 35000K images. For the experiments on low-shot datasets with Diffusion-Projected GAN, we set the batchsize as 64 and the overall training duration as 100000K images. For the experiments on FFHQ datasets, we set the batchsize as 64 and the overall training duration as 25000K images. The hyperparameter λ is set as 0.25 in QADDRS across all experiments.

4.2. Results on Low-shot Datasets

The FID results of different methods on the 256×256 low-shot datasets are shown in Table 1. We show the effec-

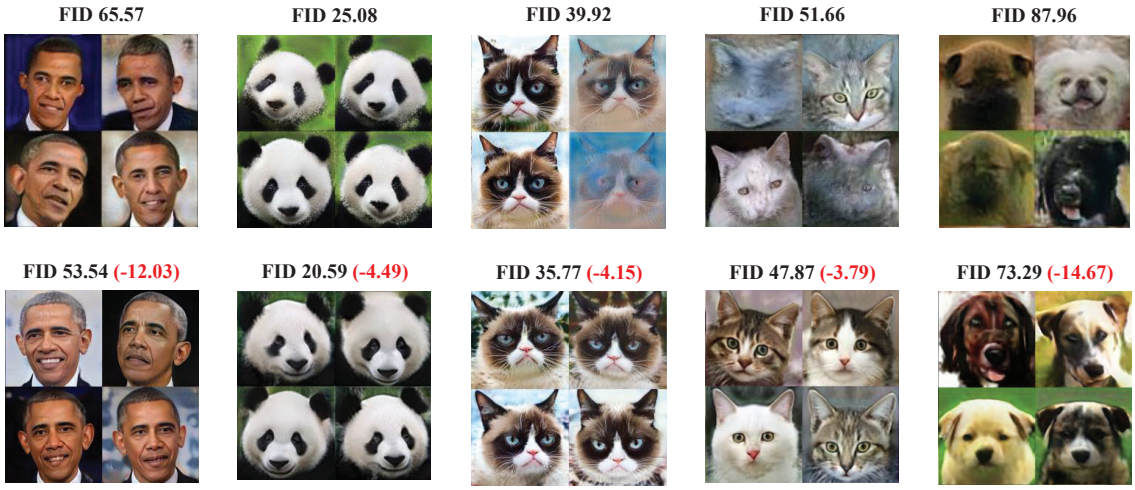

Figure 4. The compared generated results between StyleGAN2 and StyleGAN2 + QADDRS on low-shot datasets. **Top**: Images generated by StyleGAN2. **Bottom**: Images generated by StyleGAN2 + QADDRS. The reduction of FID in red color shows the improvement of StyleGAN2 + QADDRS over the baseline StyleGAN2.

Method	**Backbone**	**MA**	**Pre-training**	FFHQ 100	FFHQ 1K	FFHQ 2K	FFHQ 5K
EDM [24]	Diffusion	Yes	No	79.10	-	-	-
EDM + DA [24]	Diffusion	Yes	No	50.73	30.75	27.17	24.51
Patch Diffusion [54]	Diffusion	Yes	No	44.45	28.03	25.32	22.63
StyleGAN2 [23]	GAN	No	No	179.00	100.16	54.30	49.68
StyleGAN2 + Diff-Augment [66]	GAN	Yes	No	61.91	25.66	24.32	10.45
StyleGAN2 + ADA [22]	GAN	Yes	No	85.8	21.29	15.39	10.96
APA [19]	GAN	Yes	No	65.00	18.89	16.90	8.38
AugSelf-StyleGAN2+ [17]	GAN	Yes	No	-	20.39	-	9.15
FakeCLR [28]	GAN	Yes	No	42.56	15.92	9.90	7.25
Projected GAN [44]	GAN	Yes	Yes	26.25	11.12	8.25	6.85
InsGen [56]	GAN	Yes	No	45.75	18.21	11.47	7.83
+ QADDRS	GAN	Yes	No	**43.67**	**17.16**	**10.73**	**7.51**
Diffusion-Projected GAN (FastGAN) [55]	GAN	Yes	Yes	25.47	8.76	7.99	6.59
+ QADDRS	GAN	Yes	Yes	**24.46**	**8.51**	**7.88**	**6.48**
Diffusion-Projected GAN (FastGAN Lite) [55]	GAN	Yes	Yes	20.65	7.02	5.47	4.74
+ QADDRS	GAN	Yes	Yes	**19.89**	**6.57**	**5.18**	**4.69**

Table 2. FID scores (lower is better) of different methods on the 256×256 FFHQ dataset. In line with FakeCLR [28], the experiments are performed on 100, 1K, 2K and 5K subsets of the FFHQ dataset. MA donates Massive Augmentation [9], i.e., xflipping. The results are averaged over five evaluations, with relative standard deviations below 1%.

tiveness of the QADDRS with both StyleGAN2 and FastGAN using five different commonly-used DE-GANs, i.e., StyleGAN2, StyleGAN2 + Diff-Augment, StyleGAN2 + ADA, InsGen + ADA and Diffusion-Projected GAN. As shown in Table 1, the proposed QADDRS can increase the performance of these five different DE-GANs, demonstrating that QADDRS can be regarded as a general plug-in training strategy for DE-GANs. As shown in Table 7, Diffusion-Projected GAN (FastGAN) achieves better performance than Diffusion-Projected GAN (FastGAN Lite), i.e., applying the lite version generator in FastGAN, on low-shot datasets. Therefore, we report the results of the Diffusion-Projected GAN (FastGAN) in Table 1. Although Diffusion-Projected GAN (FastGAN) has already utilized all widely-used approaches, i.e., DA, regularizations, noise injection and pre-trained models, to boost performance on the low-shot datasets, adding QADDRS can further improve its performance. In addition, following [1], we also report

Method	Obama		Grumpy Cat		Panda		AF-Cat		AF-Dog	
	P	**R**	**P**	**R**	**P**	**R**	**P**	**R**	**P**	**R**
Diffusion-Projected GAN (FastGAN)	0.999	0.240	0.996	0.149	0.999	0.410	0.997	0.182	0.998	0.203
+ QADDRS	**0.999**	**0.242**	**0.998**	**0.151**	**0.999**	**0.418**	**0.998**	**0.186**	**0.999**	**0.203**

Table 3. The comparison of Precision (**P**) and Recall (**R**) (higher is better) with Diffusion-Projected GAN (FastGAN) [55] and Diffusion-Projected GAN (FastGAN) + QADDRS on the 256×256 low-shot datasets. The results are averaged over three evaluations, with relative standard deviations below 1%.

Method	FFHQ-100		FFHQ-1K		FFHQ-2K		FFHQ-5K	
	P	**R**	**P**	**R**	**P**	**R**	**P**	**R**
Diffusion-Projected GAN (FastGAN Lite)	0.737	0.018	0.734	0.269	0.736	0.321	0.730	0.377
+ QADDRS	**0.739**	**0.021**	**0.734**	**0.276**	**0.739**	**0.323**	**0.733**	**0.379**

Table 4. The comparison of Precision (**P**) and Recall (**R**) (higher is better) with Diffusion-Projected GAN (FastGAN Lite) [55] and Diffusion-Projected GAN (FastGAN Lite) + QADDRS on the 256×256 FFHQ dataset. The results are averaged over five evaluations, with relative standard deviations below 1%.

Method	100			AF	
	O	GC	P	C	D
Diffusion Projected GAN (FastGAN)	2.02	4.78	2.59	8.36	7.21
+ QADDRS	**1.98**	**4.71**	**2.53**	**8.20**	**7.09**

Table 5. Compared FID_{CLIP} scores between Diffusion Projected GAN (FastGAN) and Diffusion Projected GAN (FastGAN) + QADDRS on low-shot datasets. We use the FID calculation model to calculate the FID_{CLIP}. Here, AF denotes AnimalFace, O denotes Obama, GC denotes Grumpy Cat, P denotes Panda, C denotes Cat and D denotes Dog.

Method	FFHQ			
	100	1K	2K	5K
Diffusion Projected GAN (FL)	23.91	11.45	9.61	8.33
+ QADDRS	**23.32**	**11.14**	**9.42**	**8.19**

Table 6. Compared FID_{CLIP} scores between Diffusion Projected GAN (FastGAN Lite) and Diffusion Projected GAN (FastGAN Lite) + QADDRS on FFHQ datasets. We use the FID calculation model to calculate the FID_{CLIP}. FL denotes the FastGAN Lite.

the results of several SOTA diffusion models in Table 1. It is clear that GAN-based methods outperform diffusion-based methods on low-shot datasets. To further show the effectiveness of QADDRS, the compared Precision and Recall, and FID_{CLIP} of adding QADDRS to Diffusion-Projected GAN (FastGAN) are shown in Tables 3 and 5, respectively. The compared generated images on StyleGAN2 and StyleGAN2 + QADDRS are shown in Figure 4.

Method	FID
Diffusion Projected GAN (FastGAN)	**17.22**
Diffusion Projected GAN (FastGAN Lite)	20.91

Table 7. Compared FID scores of Diffusion Projected GAN (FastGAN) and Diffusion Projected GAN (FastGAN Lite) on 256×256 AnimalFace-dog dataset. The results are averaged over three evaluations, with relative standard deviations below 1%.

4.3. Results on FFHQ Dataset

According to the results of five different DE-GANs on low-shot datasets, InsGen + ADA and Diffusion-Projected GAN achieve SOTA performance on StyleGAN2 and FastGAN, respectively. Therefore, we conduct experiments on the FFHQ dataset by selecting these two types of DE-GANs. Following FakeCLR [28], we conduct experiments on the FFHQ 100, FFHQ 1K, FFHQ 2K and FFHQ 5K subsets. As shown in Table 2, QADDRS can improve performance with these two DE-GANs on the FFHQ dataset. More importantly, we report the results of both the Diffusion-Projected GAN (FastGAN) and the Diffusion-Projected GAN (FastGAN Lite) in Table 2. It is clear that Diffusion-Projected GAN (FastGAN Lite) achieves better performance than Diffusion-Projected GAN (FastGAN) on the FFHQ dataset. Additionally, the comparison of Precision and Recall, as well as FID_{CLIP} with Diffusion-Projected GAN (FastGAN Lite) and Diffusion-Projected GAN (FastGAN Lite) + QADDRS are shown in Tables 4 and 6, respectively. Diffusion-Projected GAN (FastGAN Lite) + QADDRS outperforms Diffusion-Projected GAN (FastGAN Lite) by adding QADDRS, which demonstrates the

λ	0.00	0.25	0.50	0.75
FID	46.87	**41.27**	42.40	43.32

Table 8. FID scores on the 256×256 100-shot Obama dataset with StyleGAN2 + Diff-Augment. Massive Augmentation [9], i.e., xflipping, is applied to all methods. The results are averaged over three evaluations, with relative standard deviations below 1%.

λ	0.00	0.25	0.50	0.75
FID	58.85	**51.83**	53.31	54.45

Table 9. FID scores on the 256×256 AnimalFace-Dog dataset with StyleGAN2 + Diff-Augment. Massive Augmentation [9], i.e., xflipping, is applied to all methods. The results are averaged over three evaluations, with relative standard deviations below 1%.

Method	Obama	AF-Dog
StyleGAN2 + Diff-Augment	46.87	58.85
+ Fixed rejecting number N	44.12	54.85
+ QADDRS	**41.27**	**51.83**

Table 10. FID scores on the 256×256 100-shot Obama and AnimalFace-Dog datasets. Massive Augmentation [9], i.e., xflipping, is applied to all methods. The results are averaged over three evaluations, with relative standard deviations below 1%.

effectiveness of the proposed QADDRS. To further show the superiority of QADDRS, more generated images on the FFHQ dataset can be found in supplementary materials.

4.4. Ablation Studies

Effectiveness of the hyperparameter λ in proposed QADDRS. To demonstrate the effectiveness of the hyperparameter λ in the proposed QADDRS, we conduct experiments by selecting different values of λ on the 100-shot Obama and AnimalFace-Dog datasets using StyleGAN2 + Diff-Augment. The results, shown in Tables 8 and 9, indicate that setting λ to 0.25 achieves better performance.

Effectiveness of the dynamic rejection number N in the proposed QADDRS. To show that the dynamic rejection number N in QADDRS is effective, we conduct an ablation study by comparing QADDRS with a fixed rejection number $N = 0.25 \times$ batchsize on the 100-shot Obama dataset with StyleGAN2 + Diff-Augment. The results are shown in Table 10. It is obvious that the dynamically determined rejection number N can achieve better performance compared with the fixed rejection number.

4.5. Comparisons of DRS and QADDRS

We conduct the experiments by comparing QADDRS and the existing DRS method [5] using StyleGAN2 + Diff-Augment on the 100-shot Obama and AnimalFace-dog datasets. The results are shown in Table 11. Based on ADA [22], the overfitting of D during training finally results in the DE-GANs generating low-quality images under limited data settings. Although the existing DRS method can reject lower-quality generated samples during post-training, they can not benefit the training of DE-GANs. Thus, existing DRS methods can not prevent generating low-quality images in average, leaving limited gains under limited data settings. In contrast, QADDRS can alleviate the overfitting issue during training, thereby effectively avoiding producing low-quality images in average and achieving significant improvements. As a result, QADDRS outperforms the existing DRS method for DE-GANs.

Method	Obama	AnimalFace-Dog
StyleGAN2 + Diff-Augment	46.87	58.85
+ DRS [5]	45.22	56.17
+ QADDRS	**41.27**	**51.83**

Table 11. FID scores on the 256×256 100-shot Obama and Animal-Face Dog datasets. Massive Augmentation [9], i.e., xflipping, is applied to all methods. The results are averaged over three evaluations, with relative standard deviations below 1%.

Method	**Seconds / 1K images**
Diffusion-Projected GAN (FastGAN)	17
+ QADDRS	17.25

Table 12. The training time on the 256×256 100-shot Obama dataset with or without QADDRS using Diffusion-Projected GAN (FastGAN). The results are averaged over ten evaluations on two NVIDIA RTX A5000 GPUs with a batch size of 64. All standard deviations are below 1% relative to the mean.

4.6. Computational Cost

The results of the training time on the 100-shot Obama dataset with or without QADDRS using Diffusion-Projected GAN (FastGAN) have been shown in Table 12. The proposed QADDRS results in a negligible increase in computational cost.

5. Conclusion

In this paper, we propose a novel method called *quality aware dynamic discriminator rejection sampling (QADDRS)*, to improve DE-GANs training by considering sample quality. QADDRS can effectively avoid D becoming overly confident in distinguishing both real and fake samples, thereby alleviating the overfitting of D issue. Extensive experiments on multiple datasets show that applying QADDRS to various DE-GANs can achieve better performance and deliver state-of-the-art results.

6. Acknowledgement

This work is partially supported by the High Performance Computing center of Queen's University Belfast and the Kelvin-2 supercomputer (EPSRC grant EP/T022175/1). First author is supported by a scholarship of the Queen's University Belfast and the Department for the Economy, Northern Ireland.

References

[1] Mehran Aghabozorgi, Shichong Peng, and Ke Li. Adaptive imle for few-shot pretraining-free generative modelling. In *ICML*, 2023. 6

[2] Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. *arXiv preprint arXiv:2012.00780*, 2020. 2

[3] Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017. 4

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

[5] Samaneh Azadi, Catherine Olsson, Trevor Darrell, Ian Goodfellow, and Augustus Odena. Discriminator rejection sampling. In *ICLR*, 2019. 1, 2, 8

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019. 2, 4

[7] Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. In *NeurIPS*, 2020. 2

[8] 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

[9] 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. 1, 2, 5, 6, 8

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

[11] Tiantian Fang, Ruoyu Sun, and Alex Schwing. Diggan: Discriminator gradient gap regularization for GAN training with limited data. In *NeurIPS*, 2022. 2

[12] 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

[13] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012. 4

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

[15] 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

[16] 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. 5

[17] Liang Hou, Qi Cao, Yige Yuan, Songtao Zhao, Chongyang Ma, Siyuan Pan, Pengfei Wan, Zhongyuan Wang, Huawei Shen, and Xueqi Cheng. Augmentation-aware self-supervision for data-efficient gan training. In *NeurIPS*, 2023. 5, 6

[18] Thibaut Issenhuth, Ugo Tanielian, David Picard, and Jeremie Mary. Latent reweighting, an almost free improvement for gans. In *WACV*, 2022. 2

[19] 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, 2, 3, 6

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

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 1, 4

[22] 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, 5, 6, 8

[23] 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, 4, 5, 6

[24] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 5, 6

[25] Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Ensembling off-the-shelf models for GAN training. In *CVPR*, 2022. 1, 2

[26] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019. 5

[27] Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. The role of imagenet classes in fr\'echet inception distance. In *ICLR*, 2023. 5

[28] 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, 5, 6, 7

[29] Jae Hyun Lim and Jong Chul Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017. 4

[30] Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. Towards faster and stabilized GAN training for high-fidelity few-shot image synthesis. In *ICLR*, 2021. 2, 4, 5

[31] Haozhe Liu, Wentian Zhang, Bing Li, Haoqian Wu, Nanjun He, Yawen Huang, Yuexiang Li, Bernard Ghanem, and Yefeng Zheng. Adaptivemix: Improving GAN training via feature space shrinkage. In *CVPR*, 2023. 2

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

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

[34] Takeru Miyato and Masanori Koyama. cGANs with projection discriminator. In *ICLR*, 2018. 2

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 2, 4

[36] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze the discriminator: a simple baseline for fine-tuning GANs. *arXiv preprint arXiv:2002.10964*, 2020. 5

[37] Byeonghu Na, Yeongmin Kim, Minsang Park, Donghyeok Shin, Wanmo Kang, and Il-Chul Moon. Diffusion rejection sampling. In *ICML*, 2024. 2

[38] Yao Ni and Piotr Koniusz. Nice: Noise-modulated consistency regularization for data-efficient gans. In *NeurIPS*, 2023. 1, 5

[39] Yao Ni and Piotr Koniusz. Chain: Enhancing generalization in data-efficient gans via lipschitz continuity constrained normalization. In *CVPR*, 2024. 1, 5

[40] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *ICCV*, 2019. 5

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

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

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

[44] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected GANs converge faster. In *NeurIPS*, 2021. 1, 2, 5, 6

[45] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8 (5), 2007. 4

[46] Akinori Tanaka. Discriminator optimal transport. In *NeurIPS*, 2019. 2

[47] Dustin Tran, Rajesh Ranganath, and David M Blei. Deep and hierarchical implicit models. *arXiv preprint arXiv:1702.08896*, 7(3):13, 2017. 4

[48] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *CVPR*, 2021. 2

[49] Ryan Turner, Jane Hung, Eric Frank, Yunus Saatchi, and Jason Yosinski. Metropolis-hastings generative adversarial networks. In *ICML*, 2019. 2

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

[51] Chirag Vashist, Shichong Peng, and Ke Li. Rejection sampling imle: Designing priors for better few-shot image synthesis. In *ECCV*, 2024. 2

[52] 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. 5

[53] 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. 5

[54] Zhendong Wang, Yifan Jiang, Huangjie Zheng, Peihao Wang, Pengcheng He, Zhangyang Wang, Weizhu Chen, and Mingyuan Zhou. Patch diffusion: Faster and more data-efficient training of diffusion models. In *NeurIPS*, 2023. 5, 6

[55] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. In *ICLR*, 2023. 1, 2, 5, 6, 7

[56] Ceyuan Yang, Yujun Shen, Yinghao Xu, and Bolei Zhou. Data-efficient instance generation from instance discrimination. In *NeurIPS*, 2021. 5, 6

[57] Ceyuan Yang, Yujun Shen, Yinghao Xu, Deli Zhao, Bo Dai, and Bolei Zhou. Improving GANs with a dynamic discriminator. In *NeurIPS*, 2022. 2

[58] Mengping Yang, Zhe Wang, Ziqiu Chi, and Yanbing Zhang. Fregan: Exploiting frequency components for training gans under limited data. In *NeurIPS*, 2022. 2, 5

[59] Mengping Yang, Zhe Wang, Wenyi Feng, Qian Zhang, and Ting Xiao. Improving few-shot image generation by structural discrimination and textural modulation. In *ACM MM*, 2023. 1

[60] 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

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

[62] Zhaoyu Zhang, Mengyan Li, and Jun Yu. On the convergence and mode collapse of GAN. In *SIGGRAPH Asia 2018 Technical Briefs*, 2018. 2

[63] Zhaoyu Zhang, Yang Hua, Guanxiong Sun, Hui Wang, and Seán McLoone. Improving the training of the gans with limited data via dual adaptive noise injection. In *ACMMM*, 2024. 1, 2

[64] Zhaoyu Zhang, Yang Hua, Guanxiong Sun, Hui Wang, and Seán McLoone. Improving the leaking of augmentations in data-efficient gans via adaptive negative data augmentation. In *WACV*, 2024. 1, 3, 4, 5

[65] Zhaoyu Zhang, Yang Hua, Hui Wang, and Seán McLoone. Improving the fairness of the min-max game in GANs training. In *WACV*, 2024. 1

[66] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient GAN training. In *NeurIPS*, 2020. 1, 2, 5, 6