

---

# Differentiable Augmentation for Data-Efficient GAN Training

---

**Shengyu Zhao**  
IIIS, Tsinghua University and MIT      **Zhijian Liu**  
MIT      **Ji Lin**  
MIT      **Jun-Yan Zhu**  
Adobe and CMU      **Song Han**  
MIT

## Abstract

The performance of generative adversarial networks (GANs) heavily deteriorates given a limited amount of training data. This is mainly because the discriminator is memorizing the exact training set. To combat it, we propose *Differentiable Augmentation (DiffAugment)*, a simple method that improves the data efficiency of GANs by imposing various types of differentiable augmentations on both real and fake samples. Previous attempts to directly augment the training data manipulate the distribution of real images, yielding little benefit; DiffAugment enables us to adopt the differentiable augmentation for the generated samples, effectively stabilizes training, and leads to better convergence. Experiments demonstrate consistent gains of our method over a variety of GAN architectures and loss functions for both unconditional and class-conditional generation. With DiffAugment, we achieve a state-of-the-art FID of 6.80 with an IS of 100.8 on ImageNet  $128 \times 128$  and 2-4 $\times$  reductions of FID given 1,000 images on FFHQ and LSUN. Furthermore, with only 20% training data, we can match the top performance on CIFAR-10 and CIFAR-100. Finally, our method can generate high-fidelity images using only 100 images without pre-training, while being on par with existing transfer learning algorithms. Code is available at <https://github.com/mit-han-lab/data-efficient-gans>.

## 1 Introduction

Big data has enabled deep learning algorithms achieve rapid advancements. In particular, state-of-the-art generative adversarial networks (GANs) [11] are able to generate high-fidelity natural images of diverse categories [2, 18]. Many computer vision and graphics applications have been enabled [32, 43, 54]. However, this success comes at the cost of a tremendous amount of computation and data. Recently, researchers have proposed promising techniques to improve the *computational efficiency* of model inference [22, 36], while the *data efficiency* remains to be a fundamental challenge.

GANs heavily rely on vast quantities of diverse and high-quality training examples. To name a few, the FFHQ dataset [17] contains 70,000 selective post-processed high-resolution images of human faces; the ImageNet dataset [6] annotates more than a million of images with various object categories. Collecting such large-scale datasets requires *months or even years* of considerable human efforts along with prohibitive annotation costs. In some cases, it is not even possible to have that many examples, *e.g.*, images of rare species or photos of a specific person or landmark. Thus, it is of critical importance to eliminate the need of immense datasets for GAN training. However, reducing the amount of training data results in drastic degradation in the performance. For example in Figure 1, given only 10% or 20% of the CIFAR-10 data, the training accuracy of the discriminator saturates quickly (to nearly 100%); however, its validation accuracy keeps decreasing (to lower than 30%), suggesting that the discriminator is simply memorizing the entire training set. This severe over-fitting problem disrupts the training dynamics and leads to degraded image quality.

A widely-used strategy to reduce overfitting in image classification is data augmentation [20, 38, 42], which can increase the diversity of training data without collecting new samples. Transformations such

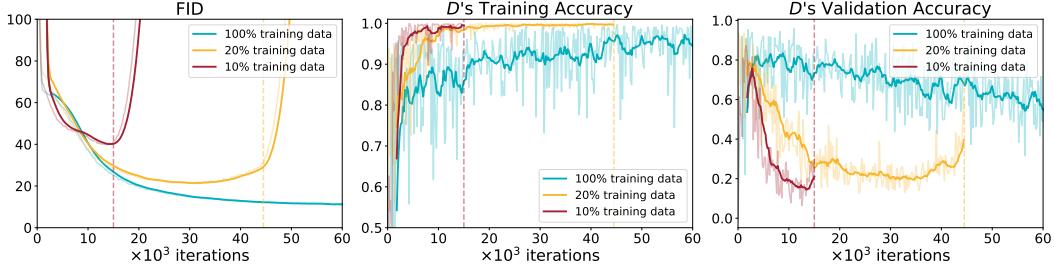


Figure 1: **BigGAN heavily deteriorates given a limited amount of data.** *left:* With 10% of CIFAR-10 data, FID increases shortly after the training starts, and the model then collapses (red curve). *middle:* the training accuracy of the discriminator  $D$  quickly saturates. *right:* the validation accuracy of  $D$  dramatically falls, indicating that  $D$  has memorized the exact training set and fails to generalize.

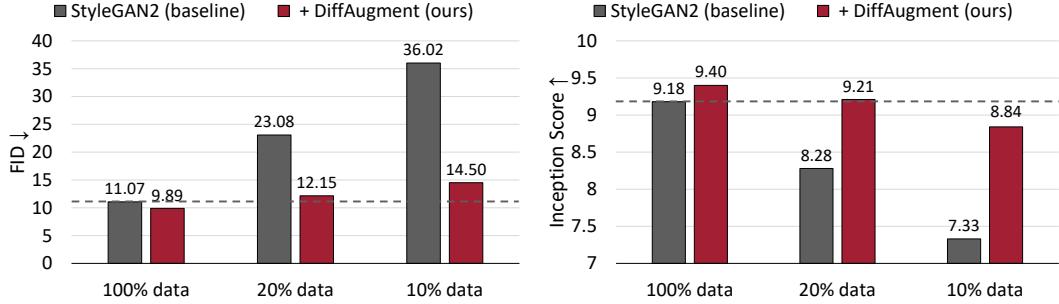


Figure 2: **Unconditional generation results on CIFAR-10.** StyleGAN2’s performance drastically degrades given less training data. With DiffAugment, we are able to roughly match its FID and outperform its Inception Score (IS) using only 20% training data. FID and IS are measured using 10k samples; the validation set is used as the reference distribution for FID calculation.

as cropping, flipping, scaling, color jittering [20], and region masking (Cutout) [8] are commonly-used augmentations for vision models. However, applying data augmentation to GANs is fundamentally different. If the transformation is only added to the real images, the generator would be encouraged to match the distribution of the *augmented* images. As a consequence, the outputs suffer from distribution shift and the introduced artifacts (*e.g.*, a region being masked, unnatural color, see Figure 5a). Alternatively, we can augment both the real and generated images when training the discriminator; however, this would break the subtle balance between the generator and discriminator, leading to poor convergence as they are optimizing completely different objectives (see Figure 5b).

To combat it, we introduce a simple but effective method, *DiffAugment*, which applies the same differentiable augmentation to both real and fake images for both generator and discriminator training. It enables the gradients to be propagated through the augmentation back to the generator, regularizes the discriminator without manipulating the target distribution, and maintains the balance of training dynamics. Experiments on a variety of GAN architectures and datasets consistently demonstrate the effectiveness of our method. With DiffAugment, we improve BigGAN and achieve a Fréchet Inception Distance (FID) of 6.80 with an Inception Score (IS) of 100.8 on ImageNet 128×128 without the truncation trick [2] and reduce the StyleGAN2 baseline’s FID by 2-4× given 1,000 images on the FFHQ and LSUN datasets. We also match the top performance on CIFAR-10 and CIFAR-100 using only 20% training data (see Figure 2 and Figure 10). Furthermore, our method can generate high-quality images with only 100 examples (see Figure 3). Without any pre-training, we achieve competitive performance with existing transfer learning algorithms that used to require tens of thousands of training images.

## 2 Related Work

**Generative Adversarial Networks.** Following the pioneering work of GAN [11], researchers have explored different ways to improve its performance and training stability. Recent efforts are centered on more stable objective functions [1, 12, 26, 27, 35], more advanced architectures [28, 29, 33, 48],



Figure 3: **Low-shot generation without pre-training.** Our method can generate high-fidelity images using only 100 Obama portraits (top) from our collected 100-shot datasets, 160 cats (middle) or 389 dogs (bottom) from the AnimalFace dataset [37] at  $256 \times 256$  resolution. See Figure 7 for the interpolation results; nearest neighbor tests are provided in Appendix A.4 (Figures 16–17).

and better training strategy [7, 15, 24, 49]. As a result, both the visual fidelity and diversity of generated images have increased significantly. For example, BigGAN [2] is able to synthesize natural images with a wide range of object classes at high resolution, and StyleGAN [17, 18] can produce photorealistic face portraits with large varieties, often indistinguishable from natural photos. However, the above work paid less attention to the data efficiency aspect. A recent attempt [3, 25] leverages semi- and self-supervised learning to reduce the amount of human annotation required for training. In this paper, we study a more challenging scenario where both data and labels are limited.

**Regularization for GANs.** GAN training often requires additional regularization as they are highly unstable. To stabilize the training dynamics, researchers have proposed several techniques including the instance noise [39], Jensen-Shannon regularization [34], gradient penalties [12, 27], spectral normalization [28], adversarial defense regularization [53], and consistency regularization [50]. All of these regularization techniques implicitly or explicitly penalize sudden changes in the discriminator’s output within a local region of the input. In this paper, we provide a different perspective, data augmentation, and we encourage the discriminator to perform well under different types of augmentation. In Section 4, we show that our method is complementary to the regularization techniques in practice.

**Data Augmentation.** Many deep learning models adopt label-preserving transformations to reduce overfitting: *e.g.*, color jittering [20], region masking [8], flipping, rotation, cropping [20, 42], data mixing [47], and local and affine distortion [38]. Recently, AutoML [40, 55] has been used to explore adaptive augmentation policies for a given dataset and task [4, 5, 23]. However, applying data augmentation to generative models, such as GANs, remains an open question. Different from the classifier training where the label is invariant to transformations of the input, the goal of generative models is to learn the data distribution itself. Directly applying augmentation would inevitably alter the distribution. We present a simple strategy to circumvent the above concern. Concurrent with our work, several methods [16, 41, 52] independently proposed data augmentation for training GANs. We urge the readers to check out their work for more details.

### 3 Method

Generative adversarial networks (GANs) aim to model the distribution of a target dataset via a generator  $G$  and a discriminator  $D$ . The generator  $G$  maps an input latent vector  $\mathbf{z}$ , typically drawn from a Gaussian distribution, to its output  $G(\mathbf{z})$ . The discriminator  $D$  learns to distinguish generated samples  $G(\mathbf{z})$  from real observations  $\mathbf{x}$ . The standard GANs training algorithm alternately optimizes the discriminator’s loss  $L_D$  and the generator’s loss  $L_G$  given loss functions  $f_D$  and  $f_G$ :

$$L_D = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f_D(-D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_D(D(G(\mathbf{z})))], \quad (1)$$

$$L_G = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_G(-D(G(\mathbf{z})))]. \quad (2)$$

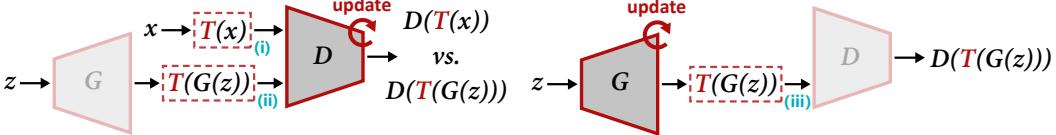


Figure 4: **Overview of DiffAugment** for updating  $D$  (left) and  $G$  (right). DiffAugment applies the augmentation  $T$  to both the real samples  $x$  and the generated output  $G(z)$ . When we update  $G$ , gradients need to be back-propagated through  $T$ , which requires  $T$  to be differentiable w.r.t. the input.

Method	Where $T$ ?			Color + Transl. + Cutout		Transl. + Cutout		Translation	
	(i)	(ii)	(iii)	IS	FID	IS	FID	IS	FID
BigGAN (baseline)				9.06	9.59	9.06	9.59	9.06	9.59
Aug. reals only	✓			5.94	49.38	6.51	37.95	8.40	19.16
Aug. reals + fakes ( $D$ only)	✓	✓		3.00	126.96	3.76	114.14	3.50	100.13
DiffAugment ( $D + G$ , ours)	✓	✓	✓	<b>9.25</b>	<b>8.59</b>	<b>9.16</b>	<b>8.70</b>	<b>9.07</b>	<b>9.04</b>

Table 1: **DiffAugment vs. vanilla augmentation strategies** on CIFAR-10 with 100% training data. “Augment reals only” applies augmentation  $T$  to (i) only (see Figure 4) and corresponds to Equations (3)-(4); “Augment  $D$  only” applies  $T$  to both reals (i) and fakes (ii), but not  $G$  (iii), and corresponds to Equations (5)-(6); “DiffAugment” applies  $T$  to reals (i), fakes (ii), and  $G$  (iii). (iii) requires  $T$  to be differentiable since gradients should be back-propagated through  $T$  to  $G$ . DiffAugment corresponds to Equations (7)-(8). IS and FID are measured using 10k samples; the validation set is the reference distribution. We select the snapshot with the best FID for each method. Results are averaged over 5 evaluation runs; all standard deviations are less than 1% relatively.

Here, different loss functions can be used, such as the non-saturating loss [11], where  $f_D(x) = f_G(x) = \log(1 + e^x)$ , and the hinge loss [28], where  $f_D(x) = \max(0, 1 + x)$  and  $f_G(x) = x$ .

Despite extensive ongoing efforts of better GAN architectures and loss functions, a fundamental challenge still exists: the discriminator tends to *memorize* the observations as the training progresses. An overfitted discriminator penalizes any generated samples other than the exact training data points, provides uninformative gradients due to poor generalization, and usually leads to training instability.

**Challenge: Discriminator Overfitting.** Here we analyze the performance of BigGAN [2] with different amounts of data on CIFAR-10. As plotted in Figure 1, even given 100% data, the gap between the discriminator’s training and validation accuracy keeps increasing, suggesting that the discriminator is simply memorizing the training images. This happens not only on limited data but also on the large-scale ImageNet dataset, as observed by Brock *et al.* [2]. BigGAN already adopts Spectral Normalization [28], a widely-used regularization technique for both generator and discriminator architectures, but still suffers from severe overfitting.

### 3.1 Revisiting Data Augmentation

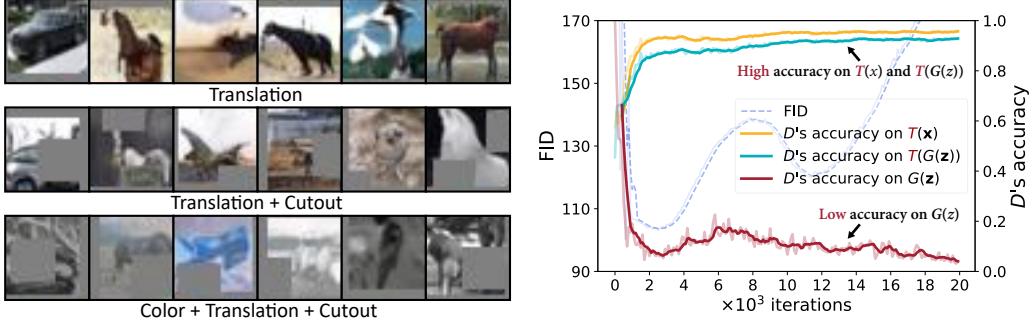
Data augmentation is a commonly-used strategy to reduce overfitting in many recognition tasks — it has an irreplaceable role and can also be applied in conjunction with other regularization techniques: *e.g.*, weight decay. We have shown that the discriminator suffers from a similar overfitting problem as the binary classifier. However, data augmentation is seldom used in the GAN literature compared to the explicit regularizations on the discriminator [12, 27, 28]. In fact, a recent work [50] observes that directly applying data augmentation to GANs does not improve the baseline. So, we would like to ask the questions: what prevents us from simply applying data augmentation to GANs? Why is augmenting GANs not as effective as augmenting classifiers?

**Augment reals only.** The most straightforward way of augmenting GANs would be directly applying augmentation  $T$  to the real observations  $x$ , which we call “Augment reals only”:

$$L_D = \mathbb{E}_{x \sim p_{\text{data}}(x)}[f_D(-D(T(x)))] + \mathbb{E}_{z \sim p(z)}[f_D(D(G(z)))], \quad (3)$$

$$L_G = \mathbb{E}_{z \sim p(z)}[f_G(-D(G(z)))]. \quad (4)$$

However, “Augment reals only” deviates from the original purpose of generative modeling, as the model is now learning a different data distribution of  $T(x)$  instead of  $x$ . This prevents us from



(a) “Augment reals only”: the same augmentation artifacts appear on the generated images.

(b) “Augment  $D$  only”: the unbalanced optimization between  $G$  and  $D$  cripples training.

**Figure 5: Understanding why vanilla augmentation strategies fail:** (a) “Augment reals only” mimics the same data distortion as introduced by the augmentations, e.g., the translation padding, the Cutout square, and the color artifacts; (b) “Augment  $D$  only” diverges because of the unbalanced optimization —  $D$  perfectly classifies the augmented images (both  $T(\mathbf{x})$  and  $T(G(\mathbf{z}))$ ) but barely recognizes  $G(\mathbf{z})$  (i.e., fake images without augmentation) from which  $G$  receives gradients.

applying any augmentation that significantly alters the distribution of the real images. The choices that meet this requirement, although strongly dependent on the specific dataset, can only be horizontal flips in most cases. We find that applying random horizontal flips does increase the performance moderately, and we use it in all our experiments to make our baselines stronger. We demonstrate the side effects of enforcing stronger augmentations quantitatively in Table 1 and qualitatively in Figure 5a. As expected, the model learns to produce unwanted color and geometric distortion (e.g., unnatural color, cutout holes) as introduced by these augmentations, resulting in a significantly worse performance (see “Augment reals only” in Table 1).

**Augment  $D$  only.** Previously, “Augment reals only” applies one-sided augmentation to the real samples, and hence the convergence can be achieved only if the generated distribution matches the manipulated real distribution. From the discriminator’s perspective, it may be tempting to augment both real and fake samples when we update  $D$ :

$$L_D = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f_D(-D(\mathbf{T}(\mathbf{x})))] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_D(D(\mathbf{T}(G(\mathbf{z}))))], \quad (5)$$

$$L_G = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_G(-D(G(\mathbf{z})))]. \quad (6)$$

Here, the same function  $T$  is applied to both real samples  $\mathbf{x}$  and fake samples  $G(\mathbf{z})$ . If the generator successfully models the distribution of  $\mathbf{x}$ ,  $T(G(\mathbf{z}))$  and  $T(\mathbf{x})$  should be indistinguishable to the discriminator as well as  $G(\mathbf{z})$  and  $\mathbf{x}$ . However, this strategy leads to even worse results (see “Augment  $D$  only” in Table 1). Figure 5b plots the training dynamics of “Augment  $D$  only” with *Translation* applied. Although  $D$  classifies the augmented images (both  $T(G(\mathbf{z}))$  and  $T(\mathbf{x})$ ) perfectly with an accuracy of above 90%, it fails to recognize  $G(\mathbf{z})$ , the generated images without augmentation, with an accuracy of lower than 10%. As a result, the generator completely fools the discriminator by  $G(\mathbf{z})$  and cannot obtain useful information from the discriminator. This suggests that any attempts that break the delicate balance between the generator  $G$  and discriminator  $D$  are prone to failure.

### 3.2 Differentiable Augmentation for GANs

The failure of “Augment reals only” motivates us to augment both *real* and *fake* samples, while the failure of “Augment  $D$  only” warns us that the *generator* should not neglect the augmented samples. Therefore, to propagate gradients through the augmented samples to  $G$ , the augmentation  $T$  must be *differentiable* as depicted in Figure 4. We call this *Differentiable Augmentation (DiffAugment)*:

$$L_D = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[f_D(-D(\mathbf{T}(\mathbf{x})))] + \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_D(D(\mathbf{T}(G(\mathbf{z}))))], \quad (7)$$

$$L_G = \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[f_G(-D(\mathbf{T}(G(\mathbf{z}))))]. \quad (8)$$

Note that  $T$  is required to be the same (random) function but not necessarily the same random seed across the three places illustrated in Figure 4. We demonstrate the effectiveness of DiffAugment using three simple choices of transformations and its composition, throughout the paper: *Translation*

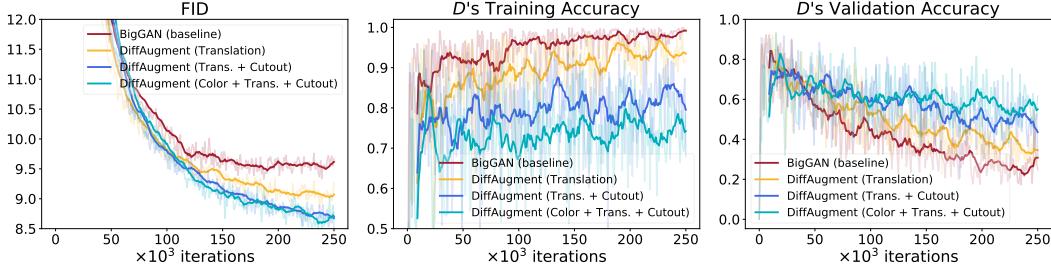


Figure 6: **Analysis of different types of DiffAugment** on CIFAR-10 with 100% training data. A stronger DiffAugment can dramatically reduce the gap between the discriminator’s training accuracy (middle) and validation accuracy (right), leading to a better convergence (left).

Method	100% training data		50% training data		25% training data	
	IS	FID	IS	FID	IS	FID
BigGAN [2]	$94.5 \pm 0.4$	$7.62 \pm 0.02$	$89.9 \pm 0.2$	$9.64 \pm 0.04$	$46.5 \pm 0.4$	$25.37 \pm 0.07$
+ DiffAugment	<b><math>100.8 \pm 0.2</math></b>	<b><math>6.80 \pm 0.02</math></b>	<b><math>91.9 \pm 0.5</math></b>	<b><math>8.88 \pm 0.06</math></b>	<b><math>74.2 \pm 0.5</math></b>	<b><math>13.28 \pm 0.07</math></b>

Table 2: **ImageNet**  $128 \times 128$  results without the truncation trick [2]. IS and FID are measured using 50k samples; the validation set is used as the reference distribution for FID. We select the snapshot with the best FID for each method. We report means and standard deviations over 3 evaluation runs.

(within  $[-1/8, 1/8]$  of the image size, padded with zeros), *Cutout* [8] (masking with a random square of half image size), and *Color* (including random brightness within  $[-0.5, 0.5]$ , contrast within  $[0.5, 1.5]$ , and saturation within  $[0, 2]$ ). As shown in Table 1, BigGAN can be improved using the simple *Translation* policy and further boosted using a composition of *Cutout* and *Translation*; it is also robust to the strongest policy when *Color* is used in combined. Figure 6 analyzes that stronger DiffAugment policies generally maintain a higher discriminator’s validation accuracy at the cost of a lower training accuracy, alleviate the overfitting problem, and eventually achieve better convergence.

## 4 Experiments

We conduct extensive experiments on ImageNet [6], CIFAR-10 [19], CIFAR-100, FFHQ [17], and LSUN-Cat [46] based on the leading class-conditional BigGAN [2] and unconditional StyleGAN2 [18]. We use the common evaluation metrics Fréchet Inception Distance (FID) [13] and Inception Score (IS) [35]. In addition, we apply our method to low-shot generation both with and without pre-training in Section 4.4. Finally, we perform analysis studies in Section 4.5.

### 4.1 ImageNet

We follow the top-performing model BigGAN [2] on ImageNet dataset at  $128 \times 128$  resolution. Additionally, we augment real images with random horizontal flips, yielding the best reimplementation of BigGAN to our knowledge (FID: ours 7.6 vs. 8.7 in the original paper [2]). We use the simple *Translation* DiffAugment for all the data percentage settings. In Table 2, our method achieves significant gains especially under the 25% data setting, in which the baseline model undergoes an early collapse, and advances the state-of-the-art FID and IS with 100% data available.

### 4.2 FFHQ and LSUN-Cat

We further experiment with StyleGAN2 [18] on the FFHQ portrait dataset [17] and the LSUN-Cat dataset [46] at  $256 \times 256$  resolution. We investigate different limited data settings, with 1k, 5k, 10k, and 30k training images available. We apply the strongest *Color + Translation + Cutout* DiffAugment to all the StyleGAN2 baselines without any hyperparameter changes. The real images are also augmented with random horizontal flips as commonly applied in StyleGAN2 [18]. Results are shown in Table 3. Our performance gains are considerable under all the data percentage settings. Moreover, with the fixed policies used in DiffAugment, our performance is on par with ADA [16], a concurrent work based on the adaptive augmentation strategy.

Method	FFHQ				LSUN-Cat			
	30k	10k	5k	1k	30k	10k	5k	1k
ADA [16]	5.46	8.13	10.96	<b>21.29</b>	10.50	13.13	16.95	43.25
StyleGAN2 [18]	6.16	14.75	26.60	62.16	10.12	17.93	34.69	182.85
+ DiffAugment	<b>5.05</b>	<b>7.86</b>	<b>10.45</b>	25.66	<b>9.68</b>	<b>12.07</b>	<b>16.11</b>	<b>42.26</b>

Table 3: **FFHQ and LSUN-Cat** results with 1k, 5k, 10k, and 30k training samples. With the fixed *Color + Translation + Cutout* DiffAugment, our method improves the StyleGAN2 baseline and is on par with a concurrent work ADA [16]. FID is measured using 50k generated samples; the full training set is used as the reference distribution. We select the snapshot with the best FID for each method. Results are averaged over 5 evaluation runs; all standard deviations are less than 1% relatively.

Method	CIFAR-10			CIFAR-100		
	100% data	20% data	10% data	100% data	20% data	10% data
BigGAN [2]	9.59	21.58	39.78	12.87	33.11	66.71
+ DiffAugment	<b>8.70</b>	<b>14.04</b>	<b>22.40</b>	<b>12.00</b>	<b>22.14</b>	<b>33.70</b>
CR-BigGAN [50]	9.06	20.62	37.45	11.26	36.91	47.16
+ DiffAugment	<b>8.49</b>	<b>12.84</b>	<b>18.70</b>	<b>11.25</b>	<b>20.28</b>	<b>26.90</b>
StyleGAN2 [18]	11.07	23.08	36.02	16.54	32.30	45.87
+ DiffAugment	<b>9.89</b>	<b>12.15</b>	<b>14.50</b>	<b>15.22</b>	<b>16.65</b>	<b>20.75</b>

Table 4: **CIFAR-10 and CIFAR-100** results. We select the snapshot with the best FID for each method. Results are averaged over 5 evaluation runs; all standard deviations are less than 1% relatively. We use 10k samples and the validation set as the reference distribution for FID calculation, as done in prior work [50]. Concurrent works [14, 16] use a different protocol: 50k samples and the training set as the reference distribution. If we adopt this evaluation protocol, our BigGAN + DiffAugment achieves an FID of 4.61, CR-BigGAN + DiffAugment achieves an FID of 4.30, and StyleGAN2 + DiffAugment achieves an FID of 5.79.

### 4.3 CIFAR-10 and CIFAR-100

We experiment on the class-conditional BigGAN [2] and CR-BigGAN [50] and unconditional StyleGAN2 [18] models. For a fair comparison, we also augment real images with random horizontal flips for all the baselines. The baseline models already adopt advanced regularization techniques, including Spectral Normalization [28], Consistency Regularization [50], and  $R_1$  regularization [27]; however, none of them achieves satisfactory results under the 10% data setting. For DiffAugment, we adopt *Translation + Cutout* for the BigGAN models, *Color + Cutout* for StyleGAN2 with 100% data, and *Color + Translation + Cutout* for StyleGAN2 with 10% or 20% data. As summarized in Table 4, our method improves all the baselines independently of the baseline architectures, regularizations, and loss functions (hinge loss in BigGAN and non-saturating loss in StyleGAN2) without any hyperparameter changes. We refer the readers to the appendix (Tables 6-7) for the complete tables with IS. The improvements are considerable especially when limited data is available. This is, to our knowledge, the new state of the art on CIFAR-10 and CIFAR-100 for both class-conditional and unconditional generation under all the 10%, 20%, and 100% data settings.

### 4.4 Low-Shot Generation

For a certain person, an object, or a landmark, it is often tedious, if not completely impossible, to collect a large-scale dataset. To address this, researchers recently exploit few-shot learning [9, 21] in the setting of image generation. Wang *et al.* [45] use fine-tuning to transfer the knowledge of models pre-trained on external large-scale datasets. Several works propose to fine-tune only part of the model [30, 31, 44]. Below, we show that our method not only produces competitive results without using external datasets or models but also is orthogonal to the existing transfer learning methods.

We replicate the recent transfer learning algorithms [30, 31, 44, 45] using the same codebase as Mo *et al.* [30] on their datasets (AnimalFace [37] with 160 cats and 389 dogs), based on the pre-trained StyleGAN model from the FFHQ face dataset [17]. To further demonstrate the data efficiency,

Method	Pre-training?	100-shot			AnimalFace [37]	
		Obama	Grumpy cat	Panda	Cat	Dog
Scale/shift [31]	Yes	50.72	34.20	21.38	54.83	83.04
MineGAN [44]	Yes	50.63	34.54	14.84	54.45	93.03
TransferGAN [45] + DiffAugment	Yes	48.73	34.06	23.20	52.61	82.38
	Yes	<b>39.85</b>	<b>29.77</b>	<b>17.12</b>	<b>49.10</b>	<b>65.57</b>
FreezeD [30] + DiffAugment	Yes	41.87	31.22	17.95	47.70	70.46
	Yes	<b>35.75</b>	<b>29.34</b>	<b>14.50</b>	<b>46.07</b>	<b>61.03</b>
StyleGAN2 [18] + DiffAugment	No	80.20	48.90	34.27	71.71	130.19
	No	<b>46.87</b>	<b>27.08</b>	<b>12.06</b>	<b>42.44</b>	<b>58.85</b>

Table 5: **Low-shot generation** results. With only **100** (Obama, Grumpy cat, Panda), **160** (Cat), or **389** (Dog) training images, our method is on par with the transfer learning algorithms that are pre-trained with **70,000** images. FID is measured using 5k generated samples; the training set is the reference distribution. We select the snapshot with the best FID for each method.



Figure 7: **Style space interpolation** of our method for low-shot generation without pre-training. The smooth interpolation results suggest little overfitting of our method even given small datasets.

we collect the 100-shot Obama, grumpy cat, and panda datasets, and train the StyleGAN2 model on each dataset using only 100 images without pre-training. For DiffAugment, we adopt *Color + Translation + Cutout* for StyleGAN2, *Color + Cutout* for both the vanilla fine-tuning algorithm TransferGAN [45] and FreezeD [30] that freezes the first several layers of the discriminator. Table 5 shows that DiffAugment achieves consistent gains independently of the training algorithm on all the datasets. Without any pre-training, we still achieve results on par with the existing transfer learning algorithms that require tens of thousands of images, with an exception on the 100-shot Obama dataset where pre-training with human faces clearly leads to better generalization. See Figure 3 and the appendix (Figures 18-22) for qualitative comparisons. While there might be a concern that the generator is likely to overfit the tiny datasets (*i.e.*, generating identical training images), Figure 7 suggests little overfitting of our method via linear interpolation in the style space [17] (see also Figure 15 in the appendix); please refer to the appendix (Figures 16-17) for the nearest neighbor tests.

#### 4.5 Analysis

Below, we investigate whether smaller model or stronger regularization would similarly reduce overfitting and whether DiffAugment still helps. Finally, we analyze additional choices of DiffAugment.

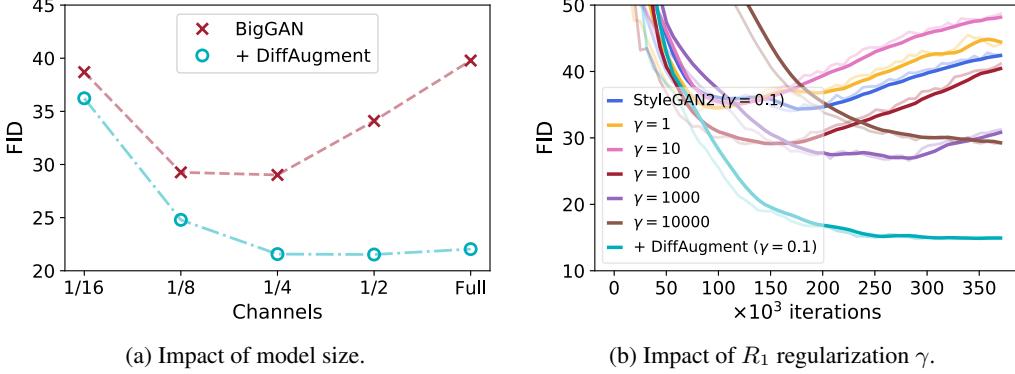


Figure 8: **Analysis of smaller models or stronger regularization** on CIFAR-10 with 10% training data. (a) Smaller models reduce overfitting for the BigGAN baseline, while our method dominates its performance at all model capacities. (b) Over a wide sweep of the  $R_1$  regularization  $\gamma$  for the baseline StyleGAN2, its best FID (26.87) is still much worse than ours (14.50).

**Model Size Matters?** We reduce the model capacity of BigGAN by progressively halving the number of channels for both  $G$  and  $D$ . As plotted in Figure 8a, the baseline heavily overfits on CIFAR-10 with 10% training data when using the full model and achieves a minimum FID of 29.02 at 1/4 channels. However, it is surpassed by our method over all model capacities. With 1/4 channels, our model achieves a significantly better FID of 21.57, while the gap is monotonically increasing as the model becomes larger. We refer the readers to the appendix (Figure 11) for the IS plot.

**Stronger Regularization Matters?** As StyleGAN2 adopts the  $R_1$  regularization [27] to stabilize training, we increase its strength from  $\gamma = 0.1$  to up to  $10^4$  and plot the FID curves in Figure 8b. While we initially find that  $\gamma = 0.1$  works best under the 100% data setting, the choice of  $\gamma = 10^3$  boosts its performance from 34.05 to 26.87 under the 10% data setting. When  $\gamma = 10^4$ , within 750k iterations, we only observe a minimum FID of 29.14 at 440k iteration and the performance deteriorates after that. However, its best FID is still  $1.8\times$  worse than ours (with the default  $\gamma = 0.1$ ). This shows that DiffAugment is more effective compared to explicitly regularizing the discriminator.

**Choice of DiffAugment Matters?** We investigate additional choices of DiffAugment in Figure 9, including random 90° rotations ( $\{-90^\circ, 0^\circ, 90^\circ\}$  each with 1/3 probability), Gaussian noise (with a standard deviation of 0.1), and general geometry transformations that involve bilinear interpolation, such as bilinear translation (within  $[-0.25, 0.25]$ ), bilinear scaling (within  $[0.75, 1.25]$ ), bilinear rotation (within  $[-30^\circ, 30^\circ]$ ), and bilinear shearing (within  $[-0.25, 0.25]$ ). While all these policies consistently outperform the baseline, we find that the *Color + Translation + Cutout* DiffAugment is especially effective. The simplicity also makes it easier to deploy.

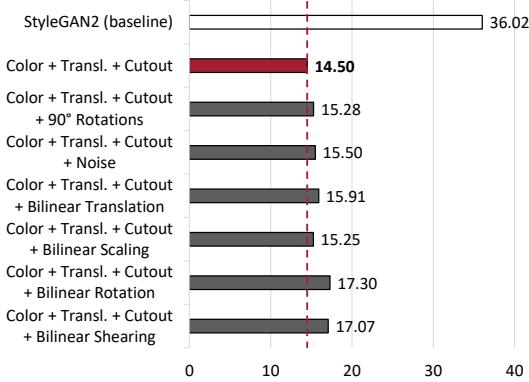


Figure 9: Various types of DiffAugment consistently outperform the baseline. We report StyleGAN2’s FID on CIFAR-10 with 10% training data.

## 5 Conclusion

We present *DiffAugment* for data-efficient GAN training. DiffAugment reveals valuable observations that augmenting both real and fake samples effectively prevents the discriminator from over-fitting, and that the augmentation must be differentiable to enable both generator and discriminator training. Extensive experiments consistently demonstrate its benefits with different network architectures (StyleGAN2 and BigGAN), supervision settings, and objective functions, across multiple datasets (ImageNet, CIFAR, FFHQ, LSUN, and 100-shot datasets). Our method is especially effective when limited data is available. Our [code](#), [datasets](#), and [models](#) are available for future comparisons.

## Broader Impact

In this paper, we investigate GANs from the data efficiency perspective, aiming to make generative modeling accessible to more people (e.g., visual artists and novice users) and research fields who have no access to abundant data. In the real-world scenarios, there could be various reasons that lead to limited amount of data available, such as rare incidents, privacy concerns, and historical visual data [10]. DiffAugment provides a promising way to alleviate the above issues and make AI more accessible to everyone.

## Acknowledgments

We thank NSF Career Award #1943349, MIT-IBM Watson AI Lab, Google, Adobe, and Sony for supporting this research. Research supported with Cloud TPUs from Google’s TensorFlow Research Cloud (TFRC). We thank William S. Peebles and Yijun Li for helpful comments.

## References

- [1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*, 2017. 2
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 3, 4, 6, 7, 12, 13, 16
- [3] Ting Chen, Xiaohua Zhai, Marvin Ritter, Mario Lucic, and Neil Houlsby. Self-supervised gans via auxiliary rotation loss. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [4] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning Augmentation Policies from Data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3
- [5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space. *arXiv*, 2019. 3
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 1, 6
- [7] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015. 3
- [8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv*, 2017. 2, 3, 6
- [9] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(4):594–611, 2006. 7
- [10] Shiry Ginosar, Kate Rakelly, Sarah Sachs, Brian Yin, and Alexei A Efros. A century of portraits: A visual historical record of american high school yearbooks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015. 10
- [11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2014. 1, 2, 4
- [12] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 2, 3, 4, 14
- [13] 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 *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 6
- [14] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv*, 2020. 7, 15
- [15] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*, 2018. 3
- [16] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv*, 2020. 3, 6, 7, 14, 15
- [17] Tero Karras, Samuli Laine, and Timo Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 3, 6, 7, 8, 14
- [18] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and Improving the Image Quality of StyleGAN. *arXiv*, 2019. 1, 3, 6, 7, 8, 13, 14, 17, 18

- [19] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 6
- [20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2012. 1, 2, 3
- [21] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. 7
- [22] Muyang Li, Ji Lin, Yaoyao Ding, Zhijian Liu, Jun-Yan Zhu, and Song Han. GAN Compression: Efficient Architectures for Interactive Conditional GANs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1
- [23] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast AutoAugment. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 3
- [24] Steven Liu, Tongzhou Wang, David Bau, Jun-Yan Zhu, and Antonio Torralba. Diverse image generation via self-conditioned gans. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3
- [25] Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-Fidelity Image Generation With Fewer Labels. In *International Conference on Machine Learning (ICML)*, 2019. 3
- [26] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2
- [27] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International Conference on Machine Learning (ICML)*, 2018. 2, 3, 4, 7, 9
- [28] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2018. 2, 3, 4, 7
- [29] Takeru Miyato and Masanori Koyama. cgans with projection discriminator. In *International Conference on Learning Representations (ICLR)*, 2018. 2
- [30] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Freeze Discriminator: A Simple Baseline for Fine-tuning GANs. *arXiv*, 2020. 7, 8, 14, 15
- [31] Atsuhiro Noguchi and Tatsuya Harada. Image generation from small datasets via batch statistics adaptation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 7, 8, 14
- [32] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1
- [33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2016. 2
- [34] Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *Conference on Neural Information Processing Systems (NeurIPS)*, pages 2018–2028, 2017. 3
- [35] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2016. 2, 6
- [36] Han Shu, Yunhe Wang, Xu Jia, Kai Han, Hanting Chen, Chunjing Xu, Qi Tian, and Chang Xu. Co-Evolutionary Compression for Unpaired Image Translation. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. 1
- [37] Zhangzhang Si and Song-Chun Zhu. Learning Hybrid Image Templates (HIT) by Information Projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2011. 3, 7, 8, 15, 22
- [38] Patrice Y Simard, David Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of International Conference on Document Analysis and Recognition*, 2003. 1, 3
- [39] Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. *arXiv*, 2016. 3
- [40] Kenneth O Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary computation*, 10(2):99–127, 2002. 3
- [41] Ngoc-Trung Tran, Viet-Hung Tran, Ngoc-Bao Nguyen, Trung-Kien Nguyen, and Ngai-Man Cheung. Towards good practices for data augmentation in gan training. *arXiv*, 2020. 3
- [42] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning (ICML)*, 2013. 1, 3
- [43] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-renn: Hard positive generation via adversary for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

- [44] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. [7](#), [8](#), [14](#)
- [45] Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. Transferring gans: generating images from limited data. In *European Conference on Computer Vision (ECCV)*, 2018. [7](#), [8](#), [14](#)
- [46] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv*, 2015. [6](#)
- [47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations (ICLR)*, 2018. [3](#)
- [48] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-Attention Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*, 2019. [2](#)
- [49] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2017. [3](#)
- [50] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2020. [3](#), [4](#), [7](#), [12](#), [13](#), [14](#)
- [51] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [15](#), [21](#)
- [52] Zhengli Zhao, Zizhao Zhang, Ting Chen, Sameer Singh, and Han Zhang. Image augmentations for gan training. *arXiv*, 2020. [3](#)
- [53] Brady Zhou and Philipp Krähenbühl. Don’t let your discriminator be fooled. In *International Conference on Learning Representations (ICLR)*, 2019. [3](#)
- [54] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2017. [1](#)
- [55] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2017. [3](#)

## Appendix A Hyperparameters and Training Details

### A.1 ImageNet Experiments

The Compare GAN codebase\* suffices to replicate BigGAN’s FID on ImageNet dataset at  $128 \times 128$  resolution but has some small differences to the original paper [2]. First, the codebase uses a learning rate of  $10^{-4}$  for  $G$  and  $5 \times 10^{-4}$  for  $D$ . Second, it processes the raw images into  $128 \times 128$  resolution with random scaling and random cropping. Since we find that random cropping leads to a worse IS, we process the raw images with random scaling and center cropping instead. We additionally augment the images with random horizontal flips, yielding the best re-implementation of BigGAN to our knowledge. With DiffAugment, we find that  $D$ ’s learning rate of  $5 \times 10^{-4}$  often makes  $D$ ’s loss stuck at a high level, so we reduce  $D$ ’s learning rate to  $4 \times 10^{-4}$  for the 100% data setting and  $2 \times 10^{-4}$  for the 10% and 20% data settings. However, we note that the baseline model does not benefit from this reduced learning rate: if we reduce  $D$ ’s learning rate from  $5 \times 10^{-4}$  to  $2 \times 10^{-4}$  under the 50% data setting, its performance degrades from an FID/IS of 9.64/89.9 to 10.79/75.7. All the models achieve the best FID within 200k iterations and deteriorate after that, taking up to 3 days on a TPU v2/v3 Pod with 128 cores.

See Figure 12 for a qualitative comparison between BigGAN and BigGAN + DiffAugment. Our method improves the image quality of the samples in both 25% and 100% data settings. The visual difference is more clear under the 25% data setting.

**Notes on CR-BigGAN [50].** CR-BigGAN [50] reports an FID of 6.66, which is slightly better than ours 6.80 (BigGAN + DiffAugment) with 100% data. However, the code and pre-trained models of CR-BigGAN [50] are not available, while its IS is not reported either. Our reimplemented CR-BigGAN only achieves an FID of 7.95 with an IS of 82.0, even worse than the baseline BigGAN. Nevertheless, our CIFAR experiments suggest the potential of applying DiffAugment on top of CR.

---

\*[https://github.com/google/compare\\_gan](https://github.com/google/compare_gan)

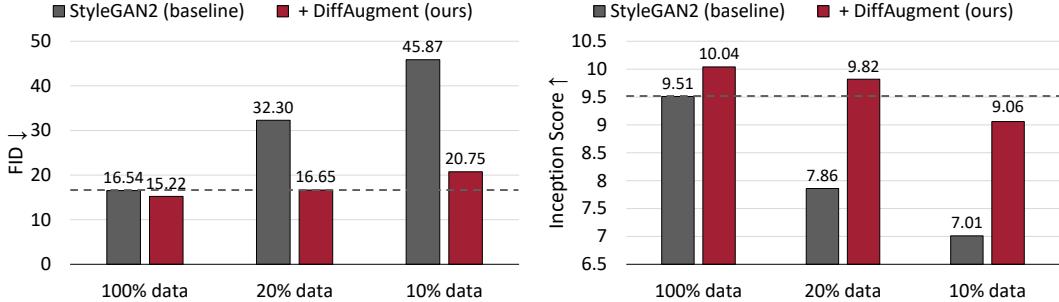


Figure 10: Unconditional generation results on **CIFAR-100**. We are able to roughly match StyleGAN2’s FID and outperform its IS using only 20% training data.

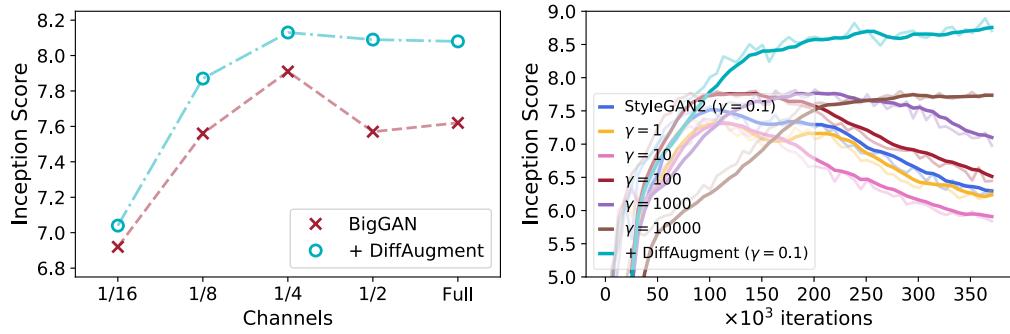


Figure 11: **Analysis of smaller models or stronger regularization** on CIFAR-10 with **10%** training data. *left*: Smaller models reduce overfitting for the BigGAN baseline, while our method outperforms it at all the model capacities. *right*: Over a wide sweep of the  $R_1$  regularization  $\gamma$  for the baseline StyleGAN2, its best IS (7.75) is still **12%** worse than ours (8.84).

Method	100% training data		20% training data		10% training data	
	IS	FID	IS	FID	IS	FID
BigGAN [2]	9.06	9.59	8.41	21.58	7.62	39.78
+ DiffAugment	<b>9.16</b>	<b>8.70</b>	<b>8.65</b>	<b>14.04</b>	<b>8.09</b>	<b>22.40</b>
CR-BigGAN [50]	<b>9.20</b>	9.06	8.43	20.62	7.66	37.45
+ DiffAugment	9.17	<b>8.49</b>	<b>8.61</b>	<b>12.84</b>	<b>8.49</b>	<b>18.70</b>
StyleGAN2 [18]	9.18	11.07	8.28	23.08	7.33	36.02
+ DiffAugment	<b>9.40</b>	<b>9.89</b>	<b>9.21</b>	<b>12.15</b>	<b>8.84</b>	<b>14.50</b>

Table 6: **CIFAR-10 results**. IS and FID are measured using 10k samples; the validation set is the reference distribution for FID calculation. We select the snapshot with the best FID for each method. Results are averaged over 5 evaluation runs; all standard deviations are less than 1% relatively.

Method	100% training data		20% training data		10% training data	
	IS	FID	IS	FID	IS	FID
BigGAN [2]	<b>10.92</b>	12.87	9.11	33.11	5.94	66.71
+ DiffAugment	10.66	<b>12.00</b>	<b>9.47</b>	<b>22.14</b>	<b>8.38</b>	<b>33.70</b>
CR-BigGAN [50]	<b>10.95</b>	11.26	8.44	36.91	7.91	47.16
+ DiffAugment	10.81	<b>11.25</b>	<b>9.12</b>	<b>20.28</b>	<b>8.70</b>	<b>26.90</b>
StyleGAN2 [18]	9.51	16.54	7.86	32.30	7.01	45.87
+ DiffAugment	<b>10.04</b>	<b>15.22</b>	<b>9.82</b>	<b>16.65</b>	<b>9.06</b>	<b>20.75</b>

Table 7: **CIFAR-100 results**. IS and FID are measured using 10k samples; the validation set is the reference distribution for FID calculation. We select the snapshot with the best FID for each method. Results are averaged over 5 evaluation runs; all standard deviations are less than 1% relatively.

## A.2 FFHQ and LSUN-Cat Experiments

We use the official TensorFlow implementation of StyleGAN2<sup>†</sup> and the default network configuration at  $256 \times 256$  resolution with an  $R_1$  regularization  $\gamma$  of 1, but without the path length regularization and the lazy regularization since they do not improve FID [18]. The number of feature maps at shallow layers ( $64 \times 64$  resolution and above) is halved to match the architecture of ADA [16]. All the models in our experiments are augmented with random horizontal flips, trained on 8 GPUs with a maximum training length of 25,000k images.

See Figure 13–14 for qualitative comparisons between StyleGAN2 and StyleGAN2 + DiffAugment. Our method considerably improves the image quality with limited data available.

## A.3 CIFAR-10 and CIFAR-100 Experiments

We replicate BigGAN and CR-BigGAN baselines on CIFAR using the PyTorch implementation<sup>‡</sup>. All hyperparameters are kept unchanged from the default CIFAR-10 configuration, including the batch size (50), the number of  $D$  steps (4) per  $G$  step, and a learning rate of  $2 \times 10^{-4}$  for both  $G$  and  $D$ . The hyperparameter  $\lambda$  of Consistency Regularization (CR) is set to 10 as recommended [50]. All the models are run on 2 GPUs with a maximum of 250k training iterations on CIFAR-10 and 500k iterations on CIFAR-100.

For StyleGAN2, we use the official TensorFlow implementation<sup>§</sup> but include some changes to make it work better on CIFAR. The number of channels is 128 at  $32 \times 32$  resolution and doubled at each coarser level with a maximum of 512 channels. We set the half-life of the exponential moving average of the generator’s weights to 1,000k instead of 10k images since it stabilizes the FID curve and leads to consistently better performance. We set  $\gamma = 0.1$  instead of 10 for the  $R_1$  regularization, which significantly improves the baseline’s performance under the 100% data setting on CIFAR. The path length regularization and the lazy regularization are also disabled. The baseline model can already achieve the best FID and IS to our knowledge for unconditional generation on the CIFAR datasets. All StyleGAN2 models are trained on 4 GPUs with the default batch size (32) and a maximum training length of 25,000k images.

We apply DiffAugment to BigGAN, CR-BigGAN, and StyleGAN2 without changes to the baseline settings. There are several things to note when applying DiffAugment in conjunction with gradient penalties [12] or CR [50]. The  $R_1$  regularization penalizes the gradients of  $D(\mathbf{x})$  w.r.t. the input  $\mathbf{x}$ . With DiffAugment, the gradients of  $D(T(\mathbf{x}))$  can be calculated w.r.t. either  $\mathbf{x}$  or  $T(\mathbf{x})$ . We choose to penalize the gradients of  $D(T(\mathbf{x}))$  w.r.t.  $T(\mathbf{x})$  for the CIFAR, FFHQ, and LSUN experiments since it slightly outperforms the other choice in practice; for the low-shot generation experiments, we penalize the gradients of  $D(T(\mathbf{x}))$  w.r.t.  $\mathbf{x}$  instead from which we observe better diversity of the generated images. As CR has already used image translation to calculate the consistency loss, we only apply *Cutout* DiffAugment on top of CR under the 100% data setting. For the 10% and 20% data settings, we exploit stronger regularization by directly applying CR between  $\mathbf{x}$  and  $T(\mathbf{x})$ , i.e., before and after the *Translation* + *Cutout* DiffAugment.

We match the top performance for unconditional generation on CIFAR-100 as well as CIFAR-10 using only 20% data (see Figure 10). See Figure 11 for the analysis of smaller models or stronger regularization in terms of IS. See Table 6 and Table 7 for quantitative results.

## A.4 Low-Shot Generation Experiments

We compare our method to transfer learning algorithms using the FreezeD’s codebase<sup>¶</sup> (for TransferGAN [45], Scale/shift [31], and FreezeD [30]) and the newly released MineGAN [44] code<sup>||</sup>. All the models are transferred from a pre-trained StyleGAN model from the FFHQ dataset [17] at  $256 \times 256$  resolution. FreezeD reports the best performance when freezing the first 4 layers of  $D$  [30]; when applying DiffAugment to FreezeD, we only freeze the first 2 layers of  $D$ . All other hyperparameters

<sup>†</sup><https://github.com/NVlabs/stylegan2>

<sup>‡</sup><https://github.com/ajbrock/BigGAN-PyTorch>

<sup>§</sup><https://github.com/NVlabs/stylegan2>

<sup>¶</sup><https://github.com/sangwoomo/FreezeD>

<sup>||</sup><https://github.com/yaxingwang/MineGAN>

are kept unchanged from the default settings. All the models are trained on 1 GPU with a maximum of 10k training iterations on our 100-shot datasets and 20k iterations on the AnimalFace [37] datasets.

When training the StyleGAN2 model from scratch, we use their default network configuration at  $256 \times 256$  resolution with an  $R_1$  regularization  $\gamma$  of 10 but without the path length regularization and the lazy regularization. We use a smaller batch size of 16, which improves the performance of both the StyleGAN2 baseline and ours, compared to the default batch size of 32. All the models are trained on 4 GPUs with a maximum training length of 300k images on our 100-shot datasets and 500k images on the AnimalFace datasets.

See Figure 15 for the additional interpolation results, Figure 16 and Figure 17 for the nearest neighbor tests of our method without pre-training both in pixel space and in the LPIPS feature space [51]. See Figures 18-22 for qualitative comparisons.

## Appendix B Evaluation Metrics

We measure FID and IS using the official Inception v3 model in TensorFlow for all the methods and datasets. Note that some papers using PyTorch implementations, including FreezeD [30], report different numbers from the official TensorFlow implementation of FID and IS. On ImageNet, CIFAR-10, and CIFAR-100, we inherit the setting from the Compare GAN codebase that the number of samples of generated images equals the number of real images in the validation set, and the validation set is used as the reference distribution for FID calculation. For the low-shot generation experiments, we sample 5k generated images and we use the training set as the reference distribution. For the FFHQ and LSUN experiments, we use the same evaluation setting as ADA [16].

## Appendix C 100-Shot Generation Benchmark

We collect the 100-shot datasets from the Internet. We then manually filter and crop each image as a pre-processing step. The full datasets are available [here](#).

## Appendix D Changelog

**v1** Initial preprint release.

**v2** (a) Add StyleGAN2 results on FFHQ (Table 3 and Figure 13). (b) For low-shot generation, we rerun the StyleGAN2 and StyleGAN + DiffAugment models with a batch size of 16 (Table 5). Both our method and the baseline are improved with a batch size of 16 over 32 (used in v1). (c) Add interpolation results on the additional 100-shot landmark datasets (Figure 15). (d) Update the CR-BigGAN notes in Appendix A.1.

**v3** (a) Add StyleGAN2 results on the LSUN-Cat dataset (Table 3 and Figure 14). (b) Rerun the MineGAN models using their newly released code (Table 5), and add qualitative samples (Figures 18-22). (c) Analyze additional choices of DiffAugment (Figure 9).

**v4** Add notes in Table 4 regarding a different evaluation protocol for FID calculation as used in concurrent works [14, 16].



Figure 12: Qualitative comparison on **ImageNet**  $128 \times 128$  without the truncation trick [2].

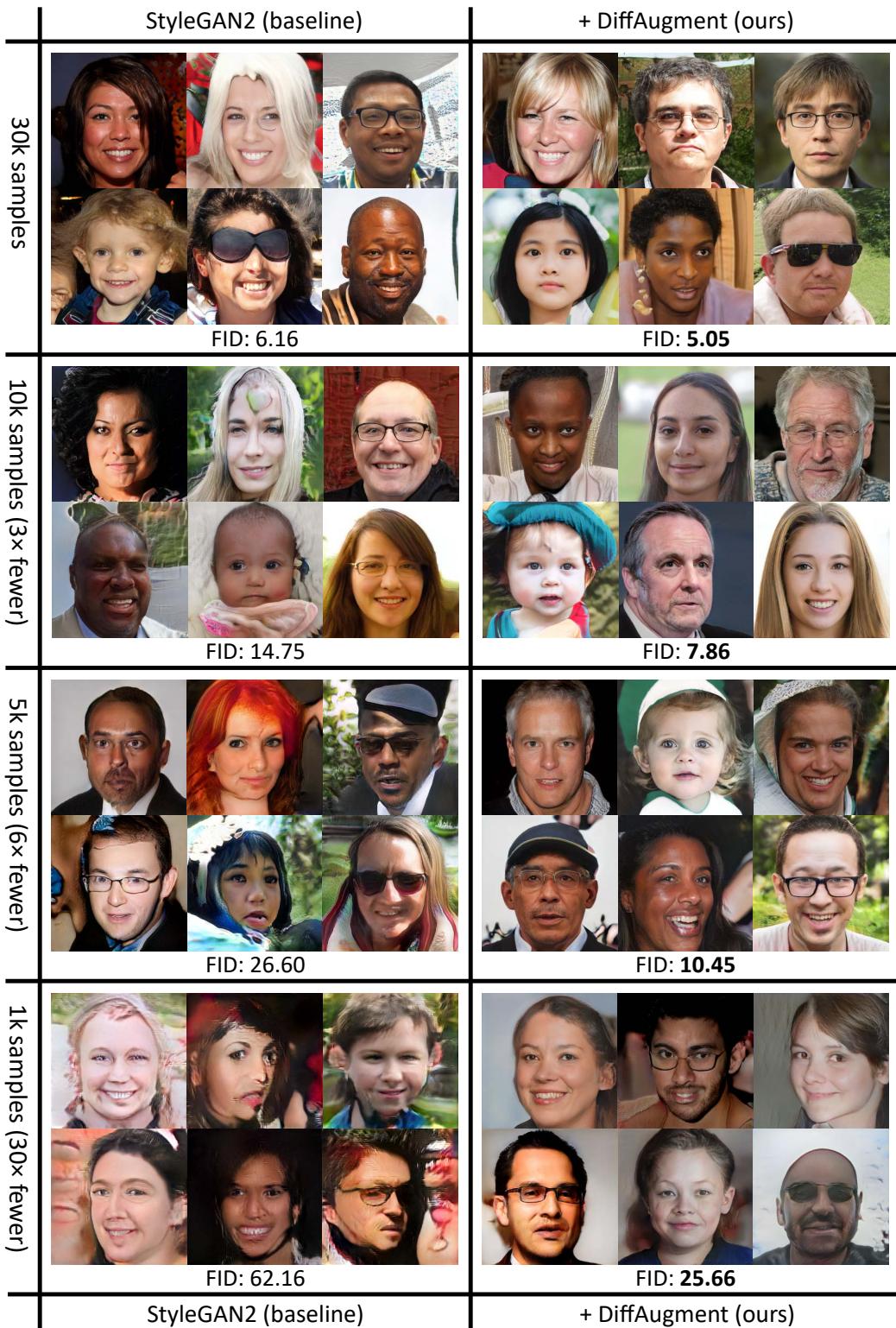


Figure 13: Qualitative comparison on **FFHQ** at 256×256 resolution with 1k, 5k, 10k, and 30k training images. Our method consistently outperforms the StyleGAN2 baselines [18] under different data percentage settings.



Figure 14: Qualitative comparison on **LSUN-cat** at 256×256 resolution with 1k, 5k, 10k, and 30k training images. Our method consistently outperforms the StyleGAN2 baselines [18] under different data percentage settings.

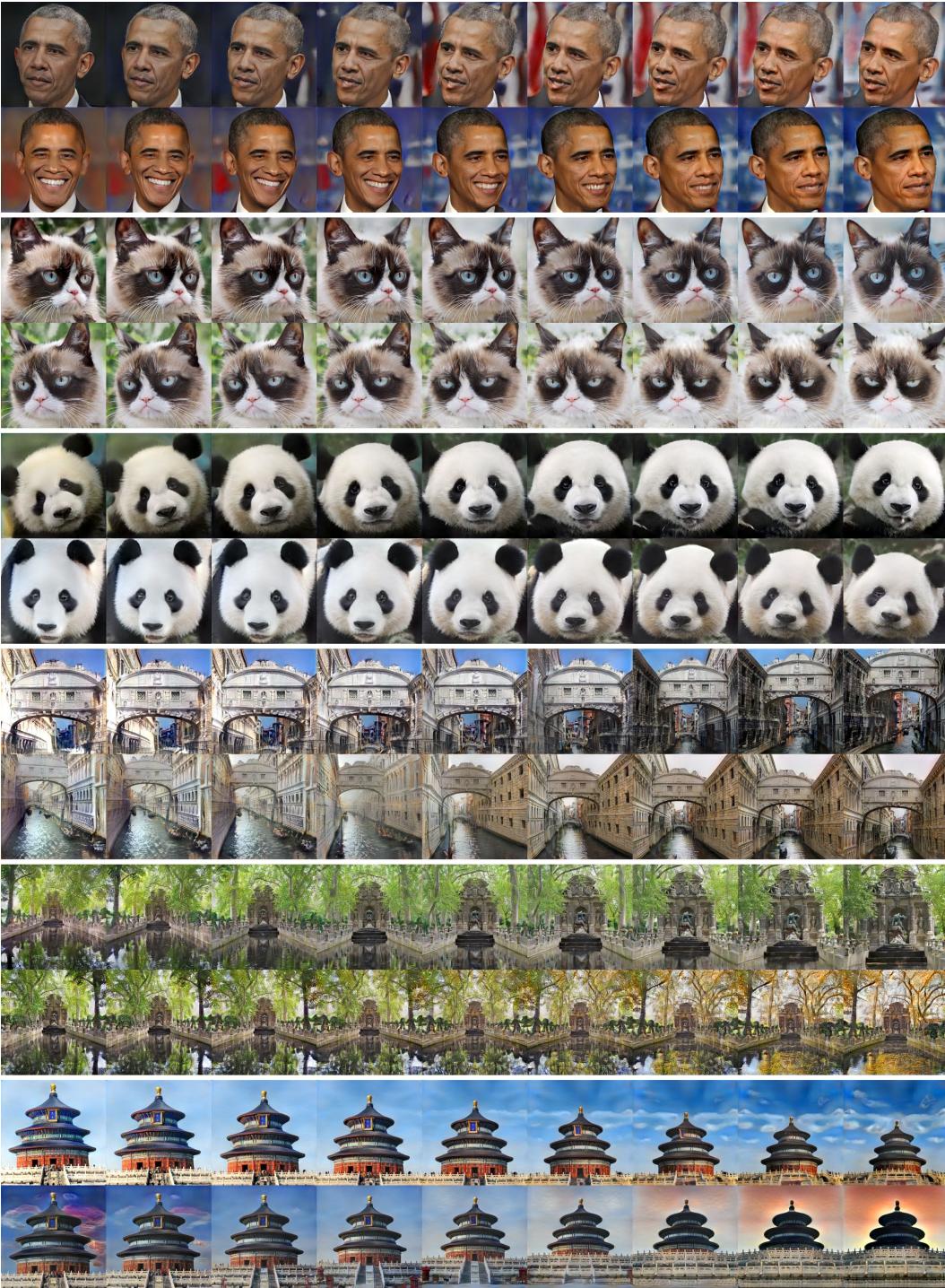


Figure 15: **Style space interpolation** of our method on the 100-shot Obama, grumpy cat, panda, the Bridge of Sighs, the Medici Fountain, and the Temple of Heaven datasets without pre-training. The smooth interpolation results suggest little overfitting of our method even given small datasets.



Figure 16: **Nearest neighbors in pixel space** measured by the pixel-wise  $L_1$  distance. Each query (on the left of the dashed lines) is a generated image of our method without pre-training (StyleGAN2 + DiffAugment) on the 100-shot or AnimalFace generation datasets. Each nearest neighbor (on the right of the dashed lines) is an original image queried from the training set with horizontal flips. The generated images are different from the training set, indicating that our model does not simply memorize the training images or overfit even given small datasets.



Figure 17: **Nearest neighbors in feature space** measured by the Learned Perceptual Image Patch Similarity (LPIPS) [51]. Each query (on the left of the dashed lines) is a generated image of our method without pre-training (StyleGAN2 + DiffAugment) on the 100-shot or AnimalFace generation datasets. Each nearest neighbor (on the right of the dashed lines) is an original image queried from the training set with horizontal flips. The generated images are different from the training set, indicating that our model does not simply memorize the training images or overfit even given small datasets.

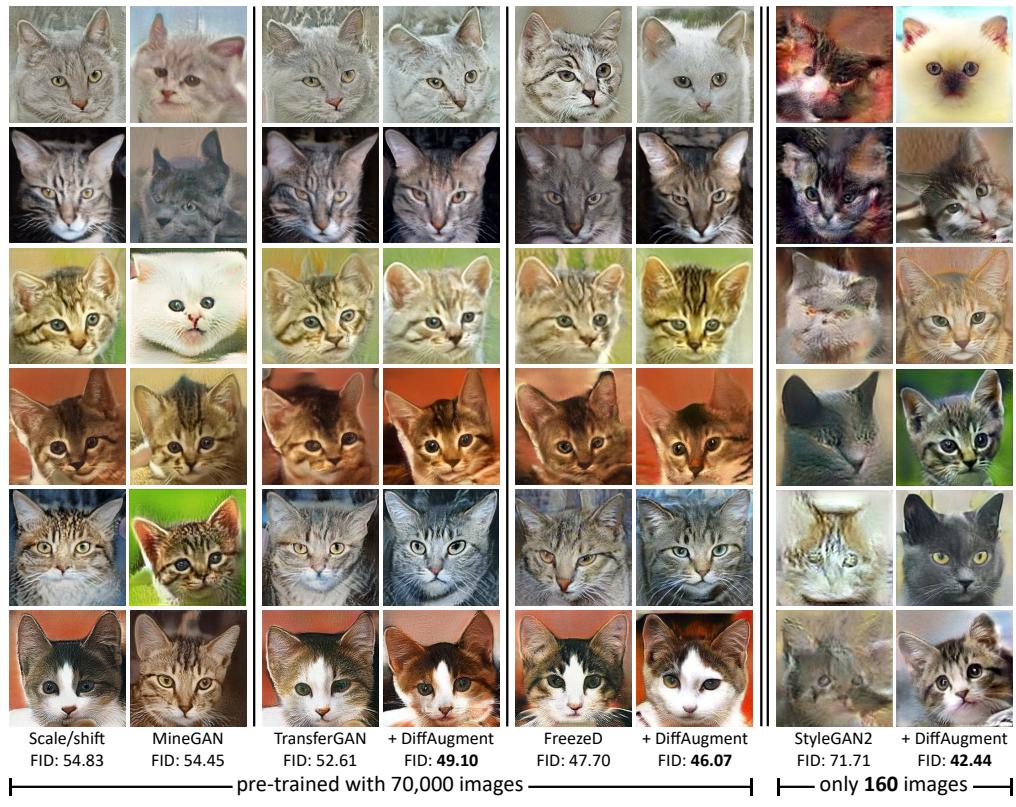


Figure 18: Qualitative comparison on the **AnimalFace-cat** [37] dataset.

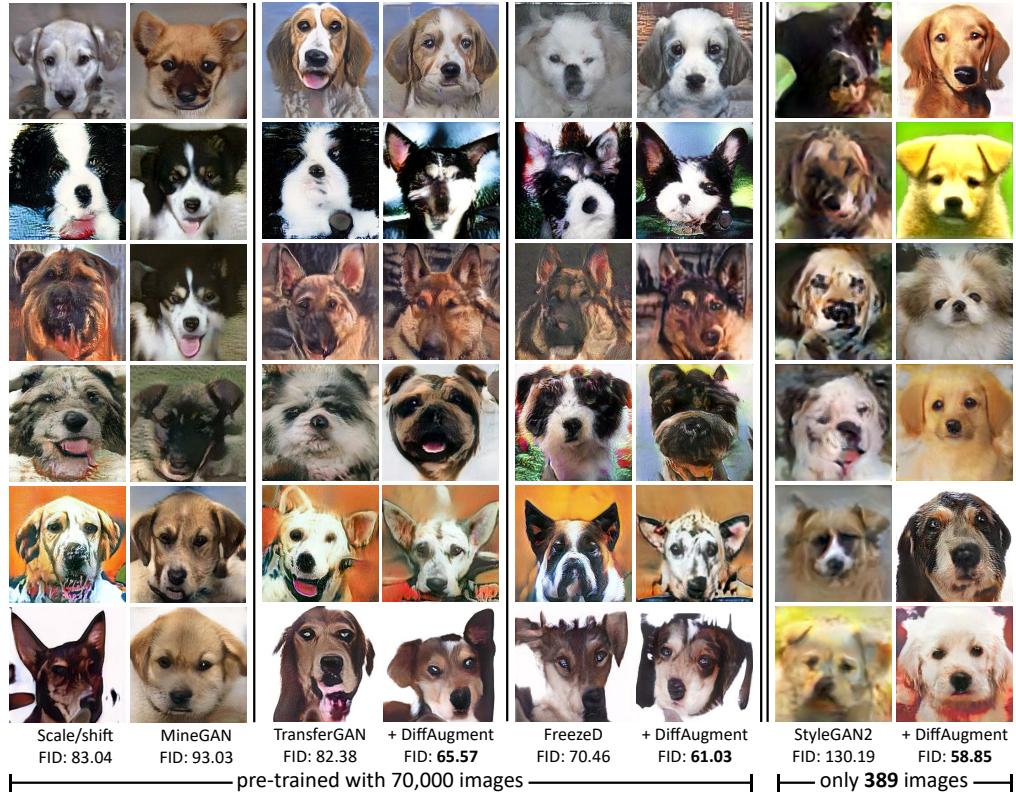


Figure 19: Qualitative comparison on the **AnimalFace-dog** [37] dataset.

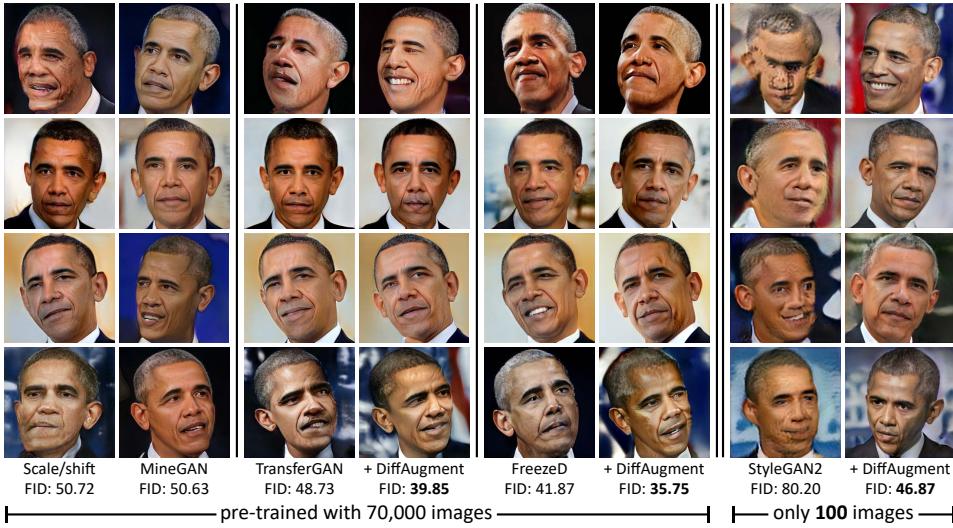


Figure 20: Qualitative comparison on the **100-shot Obama** dataset.

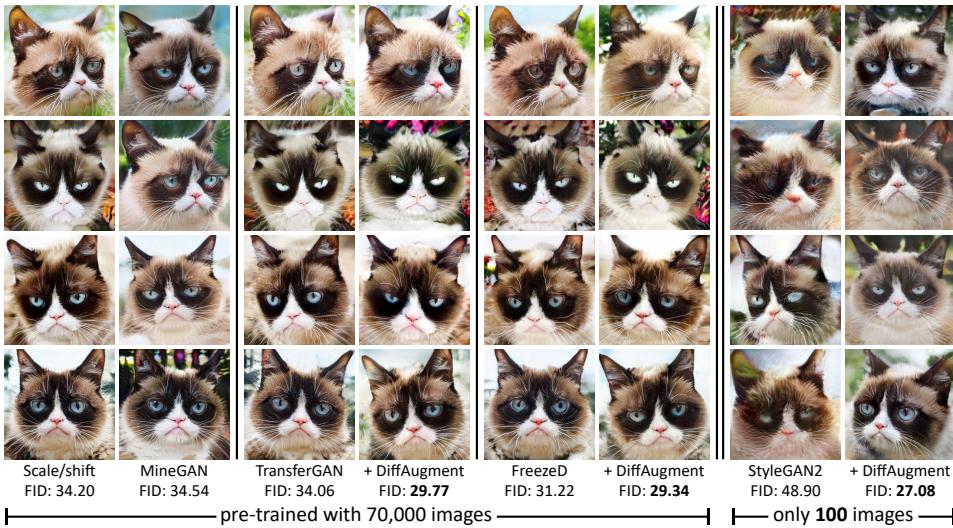


Figure 21: Qualitative comparison on the **100-shot grumpy cat** dataset.

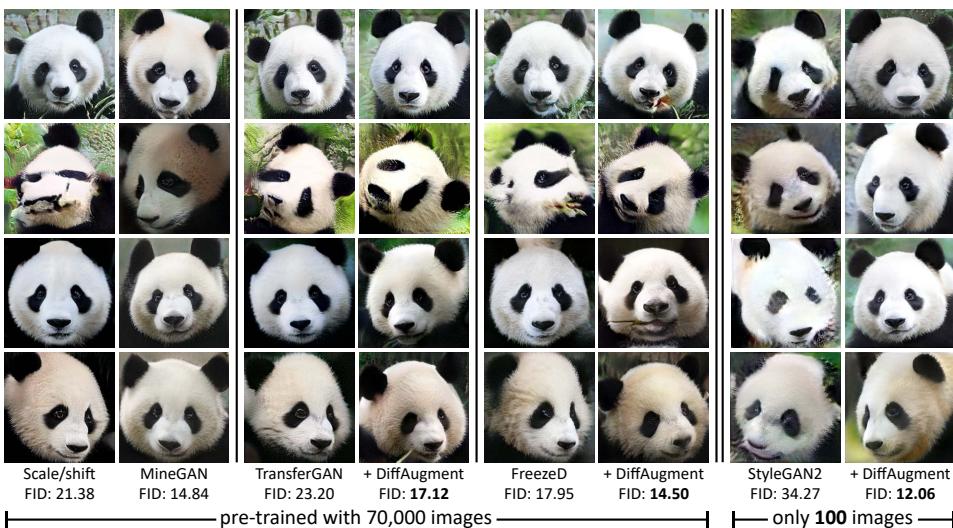


Figure 22: Qualitative comparison on the **100-shot panda** dataset.