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 128×128 and 2-4× 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, 17]. Many computer vision and graphics applications have been enabled [31, 42, 53]. 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 [21, 35], 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 [16] 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 [19, 37, 41], which can increase the diversity of training data without collecting new samples. Transformations such
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 (green 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 1: **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 [19], 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 and 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 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, 25, 26, 34], more advanced architectures [27, 28, 32, 47],
and better training strategy [7, 14, 23, 48]. 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 [16, 17] 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, 24] 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 [38], Jensen-Shannon regularization [33], gradient penalties [12, 26], spectral normalization [27], adversarial defense regularization [52], and consistency regularization [49]. 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 [19], region masking [8], flipping, rotation, cropping [19, 41], data mixing [46], and local and affine distortion [37]. Recently, AutoML [39, 54] has been used to explore adaptive augmentation policies for a given dataset and task [4, 5, 22]. 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 [15, 40, 51] 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 $z$, typically drawn from a Gaussian distribution, to its output $G(z)$. The discriminator $D$ learns to distinguish generated samples $G(z)$ from real observations $x$. 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 $f_G$:

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

(1)

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

(2)
An overfitted discriminator penalizes any generated samples other than the exact training data points, 
(iii) requires T(G(z)) to be differentiable since gradients should be back-propagated through T, which requires T to be differentiable w.r.t. the input. 

### 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, 26, 27]. In fact, a recent work [49] 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 samples x, which we call “Augment reals only”:

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

$$
L_G = E_{z \sim p(z)}[f_G(-D(G(z)))] \quad \text{(4)}
$$

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
Therefore, to propagate gradients through the augmented samples to \( G \), the failure of “Augment reals only” motivates us to augment both real and fake samples when we update \( D \). Although \( D \) classifies the augmented images (both \( T(x) \) and \( T(G(z)) \)) perfectly with an accuracy of above 90%, it fails to recognize \( G(z) \) (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 \( D \):

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

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

Here, the same function \( T \) is applied to both real samples \( x \) and fake samples \( G(z) \). If the generator successfully models the distribution of \( x \), \( T(G(z)) \) and \( T(x) \) should be indistinguishable to the discriminator as well as \( G(z) \) and \( x \). 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 \( T(G(z)) \) and \( T(x) \)) perfectly with an accuracy of above 90%, it fails to recognize \( G(z) \), the generated images without augmentation, with an accuracy of lower than 10%. As a result, the generator completely fools the discriminator by \( G(z) \) 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):

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

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

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
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).

<table>
<thead>
<tr>
<th>Method</th>
<th>100% training data</th>
<th>50% training data</th>
<th>25% training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IS</td>
<td>FID</td>
<td>IS</td>
</tr>
<tr>
<td>BigGAN [2]</td>
<td>94.5</td>
<td>7.62</td>
<td>89.9</td>
</tr>
<tr>
<td>+ DiffAugment</td>
<td><strong>100.8</strong></td>
<td><strong>6.80</strong></td>
<td><strong>91.9</strong></td>
</tr>
</tbody>
</table>

Table 2: **ImageNet** 128×128 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.

using three simple choices of transformations and its composition, throughout the paper: *Translation* (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 $[0, 1.5]$, 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 [18], CIFAR-100, FFHQ [16], and LSUN-Cat [45] based on the leading class-conditional BigGAN [2] and unconditional StyleGAN2 [17]. We use the common evaluation metrics Fréchet Inception Distance (FID) [13] and Inception Score (IS) [34]. 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 128×128 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 [17] on the FFHQ portrait dataset [16] and the LSUN-Cat dataset [45] at 256×256 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 [17]. Results are shown in Table 3. Our performance gains are considerable under all the data percentage settings.
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 [15]. 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>FFHQ</th>
<th>LSUN-Cat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>30k</td>
<td>10k</td>
</tr>
<tr>
<td>StyleGAN2 [17] (+ DiffAugment)</td>
<td>6.16</td>
<td>14.75</td>
</tr>
<tr>
<td></td>
<td>5.05</td>
<td>7.86</td>
</tr>
</tbody>
</table>

Table 4: **CIFAR-10 and CIFAR-100** results. FID is measured using 10k samples; the validation 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.

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100% data</td>
<td>20% data</td>
</tr>
<tr>
<td>CR-BigGAN [49] + DiffAugment</td>
<td>9.06</td>
<td>20.62</td>
</tr>
<tr>
<td>StyleGAN2 [17] + DiffAugment</td>
<td>11.07</td>
<td>23.08</td>
</tr>
</tbody>
</table>

Moreover, with the fixed policies used in DiffAugment, our performance is on par with ADA [15], a concurrent work based on the adaptive augmentation strategy.

### 4.3 CIFAR-10 and CIFAR-100

We experiment on the class-conditional BigGAN [2] and CR-BigGAN [49] and unconditional StyleGAN2 [17] 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 [27], Consistency Regularization [49], and $R_1$ regularization [26]; 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, 20] in the setting of image generation. Wang et al. [44] 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 [29,30,43]. 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 [29,30,43,44] using the same codebase as Mo et al. [29] on their datasets (AnimalFace [36] with 160 cats and 389 dogs), based on the pre-trained StyleGAN model from the FFHQ face dataset [16]. To further demonstrate the data efficiency, we collect the 100-shot Obama, grumpy cat, and panda datasets, and train the StyleGAN2 model
<table>
<thead>
<tr>
<th>Method</th>
<th>Pre-training?</th>
<th>100-shot</th>
<th>AnimalFace [36]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Obama</td>
<td>Grumpy cat</td>
</tr>
<tr>
<td>Scale/shift [30]</td>
<td>Yes</td>
<td>50.72</td>
<td>34.20</td>
</tr>
<tr>
<td>MineGAN [43]</td>
<td>Yes</td>
<td>50.63</td>
<td>34.54</td>
</tr>
<tr>
<td>TransferGAN [44]</td>
<td>Yes</td>
<td>39.85</td>
<td>29.77</td>
</tr>
<tr>
<td>+ DiffAugment</td>
<td>Yes</td>
<td>38.72</td>
<td>34.06</td>
</tr>
<tr>
<td>FreezeD [29]</td>
<td>Yes</td>
<td>41.87</td>
<td>31.22</td>
</tr>
<tr>
<td>+ DiffAugment</td>
<td>Yes</td>
<td>35.75</td>
<td>29.34</td>
</tr>
<tr>
<td>StyleGAN2 [17]</td>
<td>No</td>
<td>80.20</td>
<td>48.90</td>
</tr>
<tr>
<td>+ DiffAugment</td>
<td>No</td>
<td>46.87</td>
<td>27.08</td>
</tr>
</tbody>
</table>

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.

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 [44] and FreezeD [29] 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 [16] (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.
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 [26] 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.

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**


[17] 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, 18, 19


