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

Zhaoyu Zhang[1], Yang Hua[1], Guanxiong Sun[1,2], Hui Wang[1], Seán McLoone[1]

[1]Queen's University Belfast [2]Huawei UKRD

{zzhang55,Y.Hua,gsun02,h.wang,s.mcloone}@qub.ac.uk

# 1 Proofs

## 1.1 Proofs of Lemma 1

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

$$\mathbb{E}_{x\sim\beta P+\gamma Q}[f(x)]=\beta\mathbb{E}_{x\sim P}[f(x)]+\gamma\mathbb{E}_{x\sim Q}[f(x)]. \tag{S1}$$

*Proof.*

$$\begin{aligned}\mathbb{E}_{x\sim\beta P+\gamma Q}[f(x)]&=\int_x(\beta P+\gamma Q)f(x)\mathrm{d}x\\&=\beta\int_x Pf(x)\mathrm{d}x+\gamma\int_x Qf(x)\mathrm{d}x\\&=\beta\mathbb{E}_{x\sim P}[f(x)]+\gamma\mathbb{E}_{x\sim Q}[f(x)].\end{aligned} \tag{S2}$$

That concludes the proof. □

## 1.2 Proofs of Proposition 1

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

$$D^*(T(x))=\frac{P_R^T(T(x))}{P_R^T(T(x))+\lambda(1-\alpha)\hat{P}_R^T(T(x))+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))}. \tag{S3}$$

*Proof.* Given any fixed $G$, the training object of the discriminator $D$ is to maximize the $V_D(G,D)$ in Eq.(7) in the main paper, we have that

$$\begin{aligned}V_D(G,D)=\int_{T(x)}&[P_R^T(T(x))\log(D(T(x))+\lambda(1-\alpha)\hat{P}_R^T(T(x))\log(1-D(T(x))\\&+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))\log(1-D(T(x))]\mathrm{d}T(x),\end{aligned} \tag{S4}$$

which can be simplified as:

$$\begin{aligned}V_D(G,D)=\int_{T(x)}&[P_R^T(T(x))\log(D(T(x))\\&+(\lambda(1-\alpha)\hat{P}_R^T(T(x))+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x)))\log(1-D(T(x))]\mathrm{d}T(x).\end{aligned} \tag{S5}$$

Following the proof in APA [1], for any $(m,n)\in\mathbb{R}^2\setminus\{0,0\}$, the function $f(x)=m\log(x)+n\log(1-x)$ achieves its maximum value in the range $[0,1]$ at $\frac{m}{m+n}$. Besides, the discriminator $D$ is defined only inside of $\textbf{supp}(P_R^T)\cup\textbf{supp}(P_G^T)$, where **supp** is the set-theoretic support. Therefore, we conclude the proof for Proposition 1. □

## 1.3 Proofs of Proposition 2

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

$$C(G) = -2\log 2 + 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T). \tag{S6}$$

*Proof.* For $C(G)$ in Eq.(10) in the main paper, we have that

$$\begin{aligned} C(G) = -2\log 2 + \int_{T(x)} & [P_R^T(T(x))\log 2 \times D^*(T(x)) \\ & + (\lambda(1-\alpha)\hat{P}_R^T(T(x)) + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x)))\log 2 \times (1-D^*(T(x))]\mathrm{d}T(x). \end{aligned} \tag{S7}$$

By substituting Eq.(S3) into Eq.(S7), we can achieve:

$$\begin{aligned} C(G) = -2\log 2 + & \mathbf{KL}(P_R^T \,||\, \frac{P_R^T + \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}{2}) \\ & + \mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, \frac{P_R^T + \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}{2}). \end{aligned} \tag{S8}$$

By simplifying Eq.(S8), we have that

$$C(G) = -2\log 2 + 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T). \tag{S9}$$

That concludes the proof. □

## 1.4 Proofs of Theorem 1

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

$$\begin{aligned} V_G(G, D^*) = \frac{1}{(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)} & [\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T) \\ & - 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T)]. \end{aligned} \tag{S10}$$

*Proof.* By investigating the item $\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T)$, we have that

$$\begin{aligned} & \mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T) \\ & = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log \frac{\lambda(1-\alpha)\hat{P}_R^T(T(x)) + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))}{P_R^T(T(x))}] \\ & = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log \frac{\frac{\lambda(1-\alpha)\hat{P}_R^T(T(x))+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))}{P_R^T(T(x))+\lambda(1-\alpha)\hat{P}_R^T(T(x))+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))}}{\frac{P_R^T(T(x))}{P_R^T(T(x))+\lambda(1-\alpha)\hat{P}_R^T(T(x))+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T(T(x))}}] \\ & = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log \frac{1-D^*(T(x))}{D^*(T(x))}] \\ & = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(1-D^*(T(x)))] - \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T+(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(D^*(T(x)))]. \end{aligned} \tag{S11}$$

Based on Lemma 1, Eq.(S11) can be formulated as:

$$\begin{aligned} & \mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T) \\ & = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(1-D^*(T(x)))] + \mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(1-D^*(T(x)))] \\ & - \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(D^*(T(x)))] - \mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(D^*(T(x)))], \end{aligned} \tag{S12}$$

where $\mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(1-D^*(T(x)))]$ and $-\mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(D^*(T(x)))]$ have no contribution to update the $G$. Thus, these two items can be ignored when we update the $G$. Then, by applying the Lemma 1 and Proposition 2 in Eq.(S12), we can obtain that

$$
\begin{aligned}
&-\mathbb{E}_{T(x)\sim P_G^T}[\log D^*(T(x))] = -\frac{1}{(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)}\mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(D^*(T(x)))] \\
&= \frac{1}{(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)}[\mathbf{KL}(\lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T \,||\, P_R^T) \\
&\qquad - 2\mathbf{JS}(P_R^T \,||\, \lambda(1-\alpha)\hat{P}_R^T + (1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T)].
\end{aligned}
\tag{S13}
$$

That concludes the proof. □

The training algorithm of ANDA is shown in Algorithm 1.

**Algorithm 1** Training algorithm for ANDA.

**Require:** The number of $D$ iterations $n_D$, batchsize $m$, and functions $f_w(x) = \mathbb{E}_{T(x)\sim P_R^T}[\log(D_w(T(x)))]$, $f_w^1(x) = \mathbb{E}_{T(x)\sim\lambda(1-\alpha)\hat{P}_R^T}[\log(1-D_w(T(x)))]]$, $f_w^2(x) = \mathbb{E}_{T(x)\sim(1-\lambda)(1+\frac{\lambda}{1-\lambda}\alpha)P_G^T}[\log(1-D_w(T(x)))]]$ and $g_\theta(x) = -\mathbb{E}_{T(x)\sim P_G^T}[\log(D(T(x)))]$, where $T$ is one augmentation method used to transform data, $\lambda$ and $\alpha$ have the same meaning in section 3.3, $\hat{P}_R^T$ be the distribution of transformed NDA real samples. $\theta$ and $w$ are the parameters of $G$ and $D$, respectively. $z$ is the input noise of $G$.

**while** $\theta$ has not converged **do**
  **for** $t$=1, ..., $n_D$ **do**
    Samples $\{x^{(i)}\}_{i=1}^m \sim P_R$
    Samples $\{z^{(i)}\}_{i=1}^m \sim P_z$
    Update $w$ using SGD by ascending with:
      $\nabla_w \frac{1}{m}\sum_{i=1}^m [f_w(x^{(i)}) + f_w^1(x^{(i)}) + f_w^2((G_\theta(z^{(i)}))]$
  **end for**
  Samples $\{x^{(i)}\}_{i=1}^m \sim P_R$
  Samples $\{z^{(i)}\}_{i=1}^m \sim P_z$
  Update $\theta$ using SGD by ascending with:
    $\nabla_\theta \frac{1}{m}\sum_{i=1}^m [g_\theta(G_\theta(z^{(i)}))]$
**end while**

# 2 More Details about Experiments

## 2.1 Experiments Requirements

Hardware: NVIDIA DGX with 4 Tesla V100 (32G) GPUs.
Software: Linux with 64-bit Python 3.7 and PyTorch 1.7.1, CUDA toolkit 11.0 and GCC version is 6.1.0.
Python libraries: click, requests, tqdm, pyspng, ninja, imageio-ffmpeg 0.4.3 and h5py.

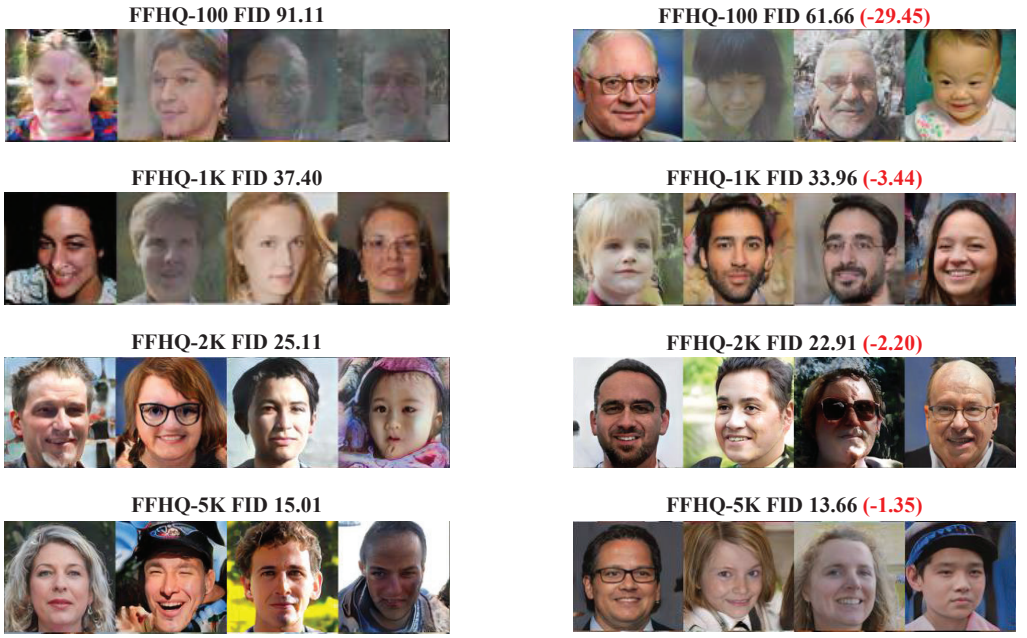
## 2.2 More details about Implement

In the main paper, we implement our ANDA on four different DE-GANs, i.e., StyleGAN2 + Diff-Augment [2], StyleGAN2 + ADA [3], Diffusion-GAN (StyleGAN2 backbone) [4] and InsGen [5], with selecting StyleGAN2 as the backbone. For the implementation, the training regularization applied in the StyleGAN2 [6] is preserved, including path length regularization, lazy regularization, and style mixing regularization. Furthermore, the Exponential moving average of generator weights, non-saturating logistic loss with $R_1$ regularization, and Adam [7] optimizer are also adopted. For a fair comparison, we select the same training iterations as the baseline. All the results in the paper are based on Mixed-precision training, i.e., FP16.

| Method | FFHQ-100 | FFHQ-1K | FFHQ-2K | FFHQ-5K |
|---|---|---|---|---|
| Diffusion-GAN (StyleGAN2 backbone) [4] | 91.11 | 37.40 | 25.11 | 15.01 |
| **+ANDA** | **61.66** | **33.96** | **22.91** | **13.66** |

Table S1: FID score (lower is better) on $256 \times 256$ FFHQ dataset. Following FakeCLR [8], we perform experiments on 100, 1K, 2K and 5K training samples on the FFHQ dataset. Massive Augmentation (MA) is applied in all of the methods. For a fair comparison, FID is measured using 50K generated samples; the full training set (70K) is used as the reference distribution. The FIDs are averaged over three runs; all standard deviations are less than $1\%$, relatively. The results of Diffusion-GAN (StyleGAN2 backbone) are run by ourselves based on the official open-source codes.

## 2.3 More Experiments on the FFHQ Dataset




**(a) Diffusion-GAN (StyGAN2 backbone)** **(b) Diffusion-GAN (StyGAN2 backbone) + ANDA**

Fig. S1: The comparison of generated images with (a) Diffusion-GAN (StyleGAN2 backbone) and (b) Diffusion-GAN (StyleGAN2 backbone) + ANDA on the FFHQ dataset. Adding ANDA to Diffusion-GAN (StyleGAN2 backbone) can effectively improve the leaking of augmentations problem, especially in the low-shot data setting, thus leading to a better quality of generated images. *Best viewed in color.*

We show the additional experimental results on the FFHQ dataset [6] by comparing with the latest Diffusion-GAN (StyleGAN2 backbone) [4]. Following the experiments in FakeCLR [8], the subset of the training set, i.e., less than 5K, is used for training DE-GANs, and the full training set (70K) is used as the reference distribution to calculate the FID. The results are shown in Table S1. Adding ANDA can achieve further improvement compared with the baseline. To further demonstrate this, the compared generated images on the FFHQ dataset with Diffusion-GAN (StyleGAN2 backbone) are shown in Figure S1.

## 2.4 More Generated Images Results on the Low-shot Datasets

To further demonstrate the superiority of the proposed ANDA, the additional compared generated images on AnimalFace-cat and AnimalFace-dog datasets with Diffusion-GAN (StyleGAN2 backbone) are shown in Figure S2. The generated images by adding ANDA on different other DE-GAN methods, i.e., StyleGAN2 + Diff-Augment, StyleGAN2 + ADA and InsGen, on Low-shot datasets are shown in Figures S3, S4 and S5, respectively.


Fig. S2: The comparison of generated images with (a) Diffusion-GAN (StyleGAN2 backbone) and (b) Diffusion-GAN (StyleGAN2 backbone) + ANDA on the Animal-Face Dog and Animal-Face Cat datasets. Adding ANDA to Diffusion-GAN (StyleGAN2 backbone) can effectively improve the leaking of augmentations problem, thus leading to a better quality of generated images. *Best viewed in color.*

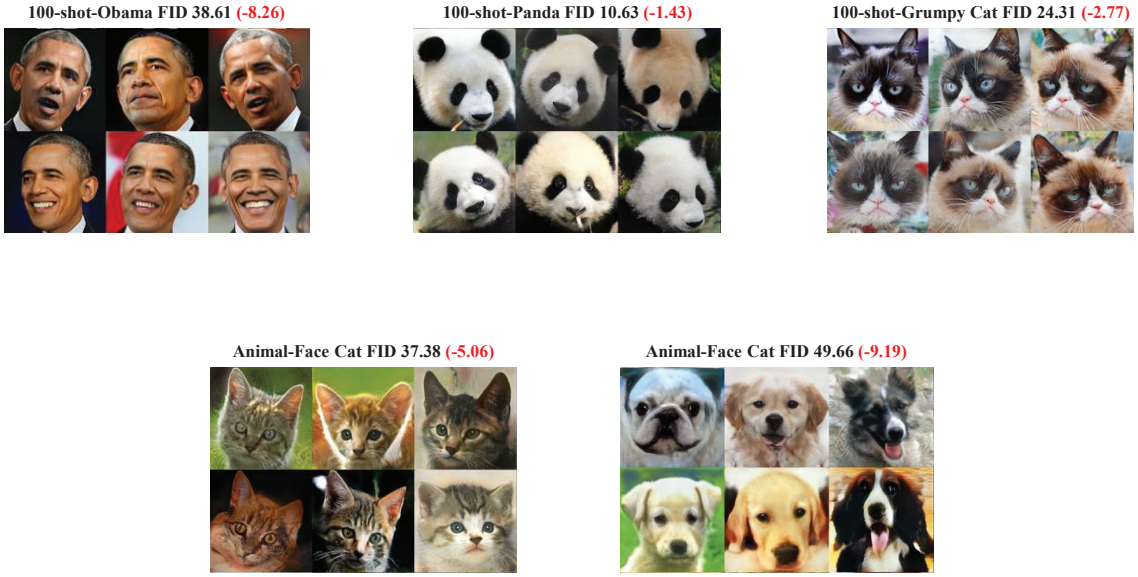



Fig. S3: Images generated by StyleGAN2 + Diff-Augment + ANDA on 100-shot Obama, 100-shot Panda, 100-shot Grumpy Cat, Animal-Face Cat and Animal-Face Dog datasets. The red value shows the improvements made by adding ANDA to the baseline StyleGAN2 + Diff-Augment. *Best viewed in color.*

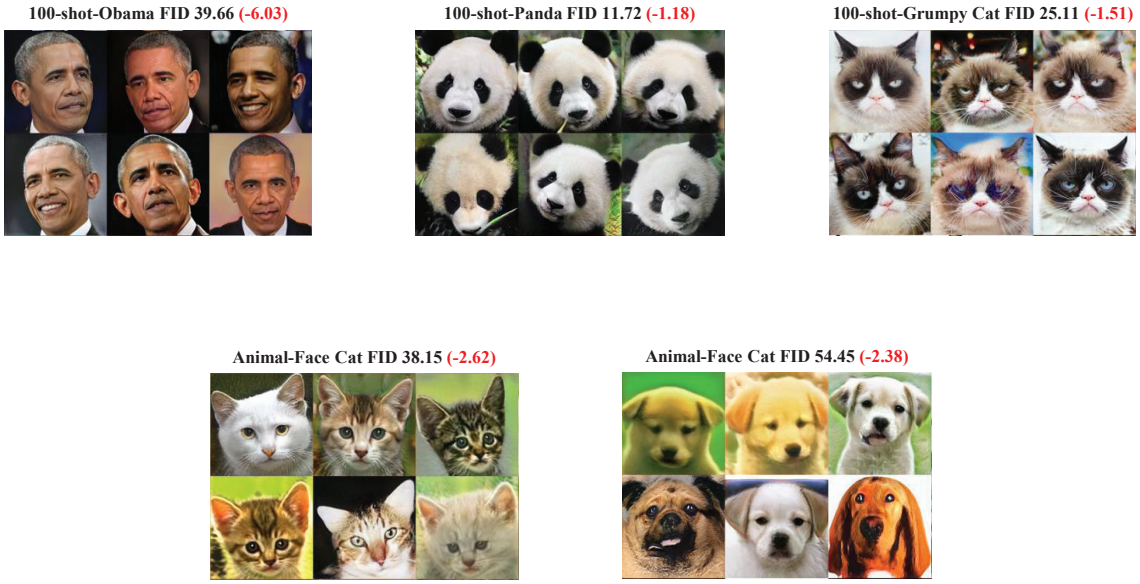



Fig. S4: Images generated by StyleGAN2 + ADA + ANDA on 100-shot Obama, 100-shot Panda, 100-shot Grumpy Cat, Animal-Face Cat and Animal-Face Dog datasets. The red value shows the improvements made by adding ANDA to the baseline StyleGAN2 + ADA. *Best viewed in color.*

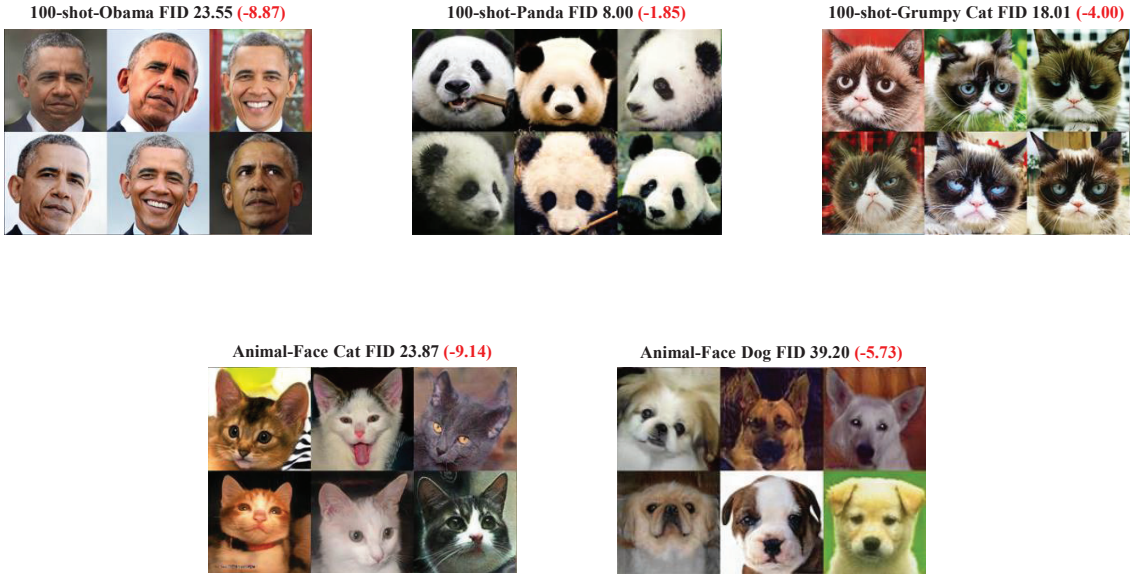



Fig. S5: Images generated by InsGen + ANDA on 100-shot Obama, 100-shot Panda, 100-shot Grumpy Cat, Animal-Face Cat and Animal-Face Dog datasets. The red value shows the improvements made by adding ANDA to the baseline InsGen. *Best viewed in color.*

# 3 Discussion of Boarder Impact and Limitations

**Boarder Impact.** This paper proposes a novel adaptive negative augmentation (ANDA) method for DE-GANs to benefit the practical deployment of training GANs with limited data with negligible computational cost. The technical contributions of this paper do not raise any particular ethical challenges. However, because technology is usually a double-edged sword, our work may also bring potential social risks when applying GANs with limited data. For example, it may ease the fake media synthesis using only limited data.

**Limitations.** The proposed ANDA can alleviate the leaking of augmentations problem in DE-GANs, but it cannot solve this problem completely. As shown in Figure 1, adding ANDA in Diffusion-GAN (StyleGAN2 backbone) can also produce slightly noise-based images. However, with the rapid technical development in recent years, we think this limitation will be well solved in the future.

# References


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

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

[3] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *NeurIPS*, 2020.

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

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

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

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[8] 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.