# Differentiable Augmentation for Data-Efficient GAN Training[1]

September 2, 2020

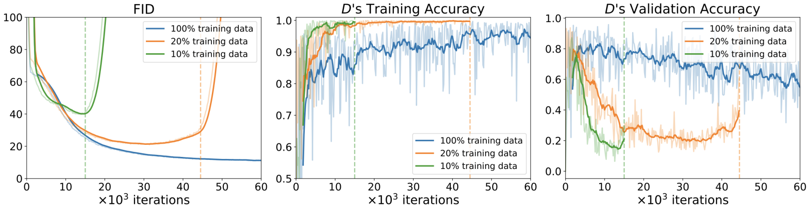
[1] *Differentiable Augmentation for Data-Efficient GAN Training* by Zhao et al., 2020

# Overview

- Training GANs generally requires a lot of data
- That's because it is too easy for D to memorize a small training set
- When D memorizes the training set, the training dynamics gets disrupted
- To alleviate this, people use augmentations
  - for reals
  - for reals and fakes for $D$ only
  - for reals and fakes for both $D$ and $G$
- Authors propose to backprop through augmentations of $G(z)$ and show that it works the best

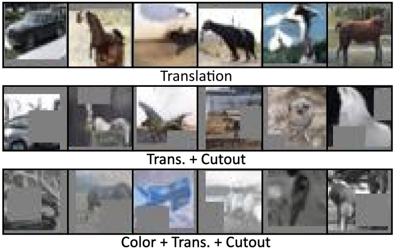
# Training without augmentations

$$\begin{aligned} L_D &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[f_D(-D(\boldsymbol{x}))\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[f_D(D(G(\boldsymbol{z})))\right], \\ L_G &= \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[f_G(-D(G(\boldsymbol{z})))\right] \end{aligned} \tag{1}$$



# Augmenting only reals

$$
\begin{aligned}
L_D &= \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\left[f_D(-D(\textcolor{red}{T}(\boldsymbol{x})))\right] + \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[f_D(D(G(\boldsymbol{z})))\right], \\
L_G &= \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}\left[f_G(-D(G(\boldsymbol{z})))\right]
\end{aligned} \tag{2}
$$

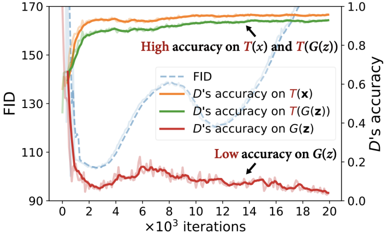



(a) "Augment reals only": the same augmentation artifacts appear on the generated images.

**Main problem**: $G$ starts producing augmentations as well (cutouts, color jitters, etc)

# Augmenting only $D$

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



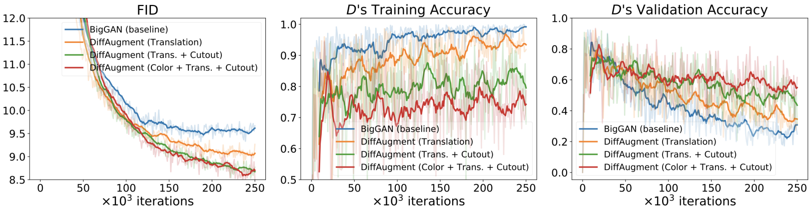

(b) "Augment $D$ only": the unbalanced optimization between $G$ and $D$ cripples training.

**Main problem**: it becomes too easy for $G$ to fool $D$:

$D$ recognizes reals well, but fakes very badly

# Using DiffAug decreases train accuracy and improves val accuracy

$$
\begin{aligned}
L_D &= \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}\left[f_D(-D(T(\boldsymbol{x})))\right] + \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}\left[f_D(D(T(G(\boldsymbol{z}))))\right], \\
L_G &= \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z})}\left[f_G(-D(T(G(\boldsymbol{z}))))\right]
\end{aligned} \tag{4}
$$




Core idea: to augment fakes *our augmentations must be differentiable*.

# Quantitive results on CIFAR-10 on 100% data

| Method | Where $T$? | | | Color + Trans. + Cutout | | Trans. + 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) | ✓ | ✓ | ✓ | **9.25** | **8.59** | **9.16** | **8.70** | **9.07** | **9.04** |

# Quantitative results on limited data

| Method | FFHQ | | | | |
|---|---|---|---|---|---|
| | Full (70k samples) | 30k samples | 10k samples | 5k samples | 1k samples |
| ADA [14] | 3.81 | 5.46 | 8.13 | 10.96 | **21.29** |
| StyleGAN2 [16] | **3.71** | 6.16 | 14.75 | 26.60 | 62.16 |
| + DiffAugment | 4.24 | **5.05** | **7.86** | **10.45** | 25.66 |

| Method | Pre-training? | 100-shot | | | AnimalFace [35] | |
|---|---|---|---|---|---|---|
| | | Obama | Grumpy cat | Panda | Cat | Dog |
| Scale/shift [29] | Yes | 50.72 | 34.20 | 21.38 | 54.83 | 83.04 |
| MineGAN [42] | Yes | 235.00 | 287.96 | 331.86 | 279.48 | 254.08 |
| TransferGAN [43] | Yes | 48.73 | 34.06 | 23.20 | 52.61 | 82.38 |
| + DiffAugment | Yes | **39.85** | **29.77** | **17.12** | **49.10** | **65.57** |
| FreezeD [28] | Yes | 41.87 | 31.22 | 17.95 | 47.70 | 70.46 |
| + DiffAugment | Yes | **35.75** | **29.34** | **14.50** | **46.07** | **61.03** |
| StyleGAN2 [16] | No | 80.20 | 48.90 | 34.27 | 71.71 | 130.19 |
| + DiffAugment | No | **46.87** | **27.08** | **12.06** | **42.44** | **58.85** |

# Qualitative results on limited data


Figure: Few-shot generation. Faces: 100-shot, cats: 160-shot, dogs: 389-shot. Resolution: $256 \times 256$

# Final thoughts

- DiffAugs are easy to implement and they are fast.
- Self-supervised learning is very simple to incorporate and boosts scores in many areas
  - Where else can we apply it?
- Next step: more powerful augmentations? FixMatch/SimCLR/etc have much more augmentations under the hood.
- Next step: parametrized differentiable augmentations?
  - It will be very similar to Spatial Transformer Networks then