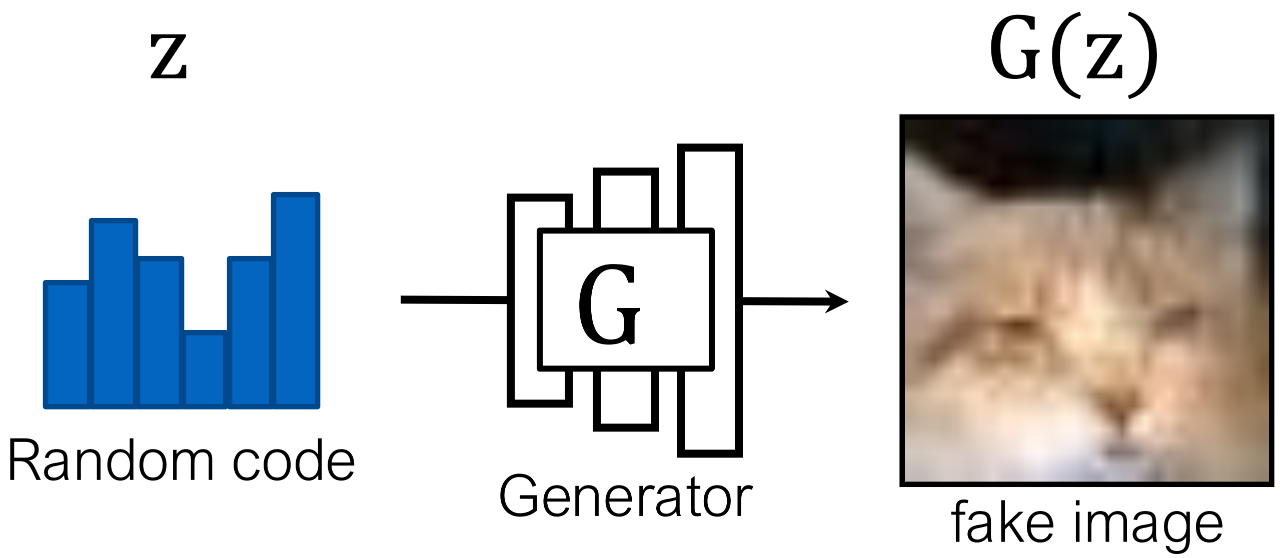
Generative Adversarial Networks (part 2)

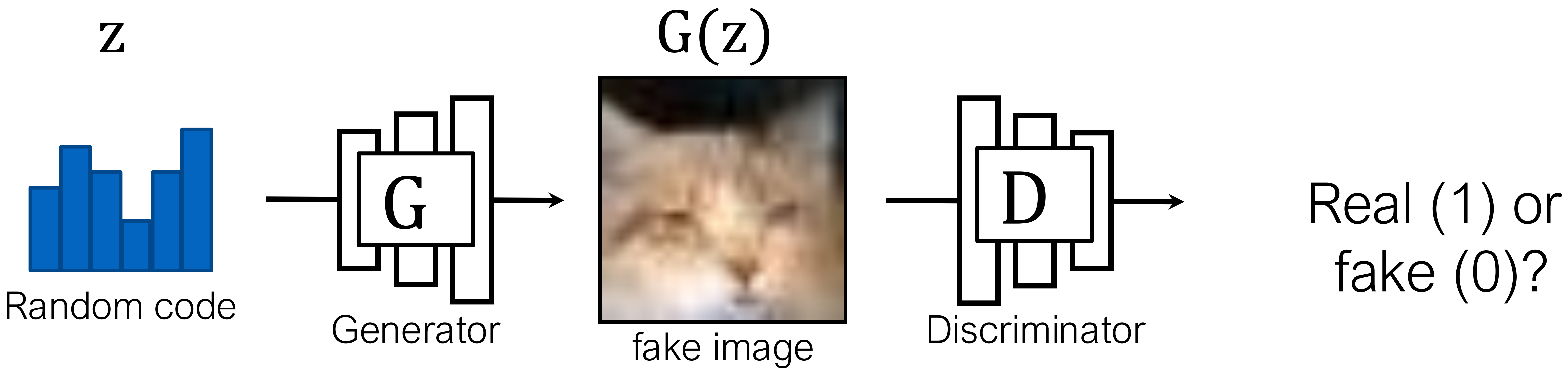
Jun-Yan Zhu

16-726 Learning-based Image Synthesis, Spring 2025

many slides from Phillip Isola, Richard Zhang, Alyosha Efros



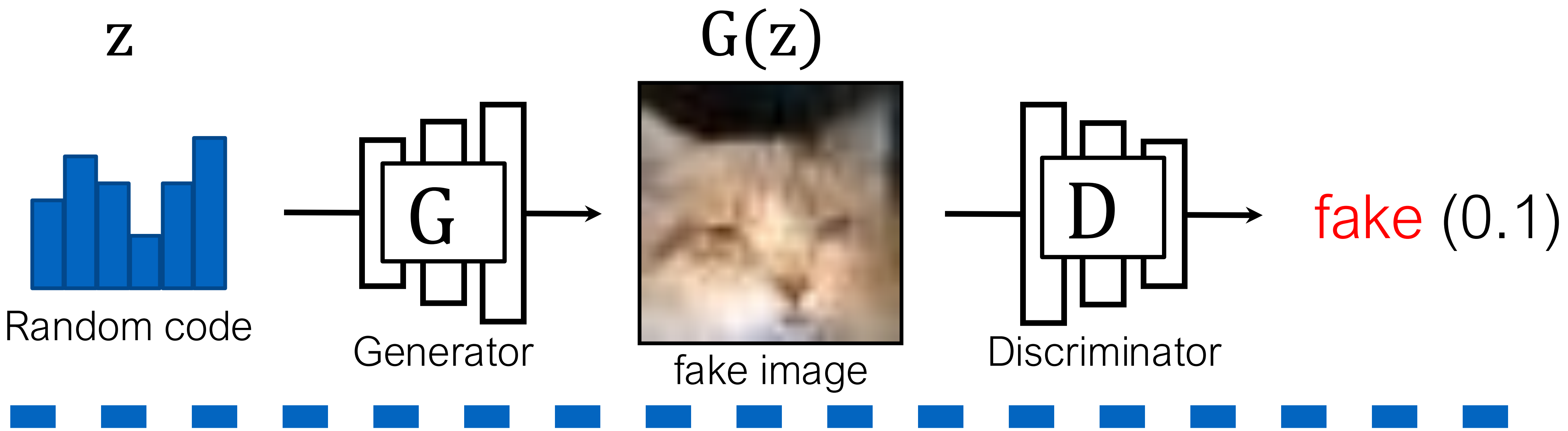
[Goodfellow et al. 2014]



A two-player game:

- G tries to generate fake images that can fool D.
- D tries to detect fake images.

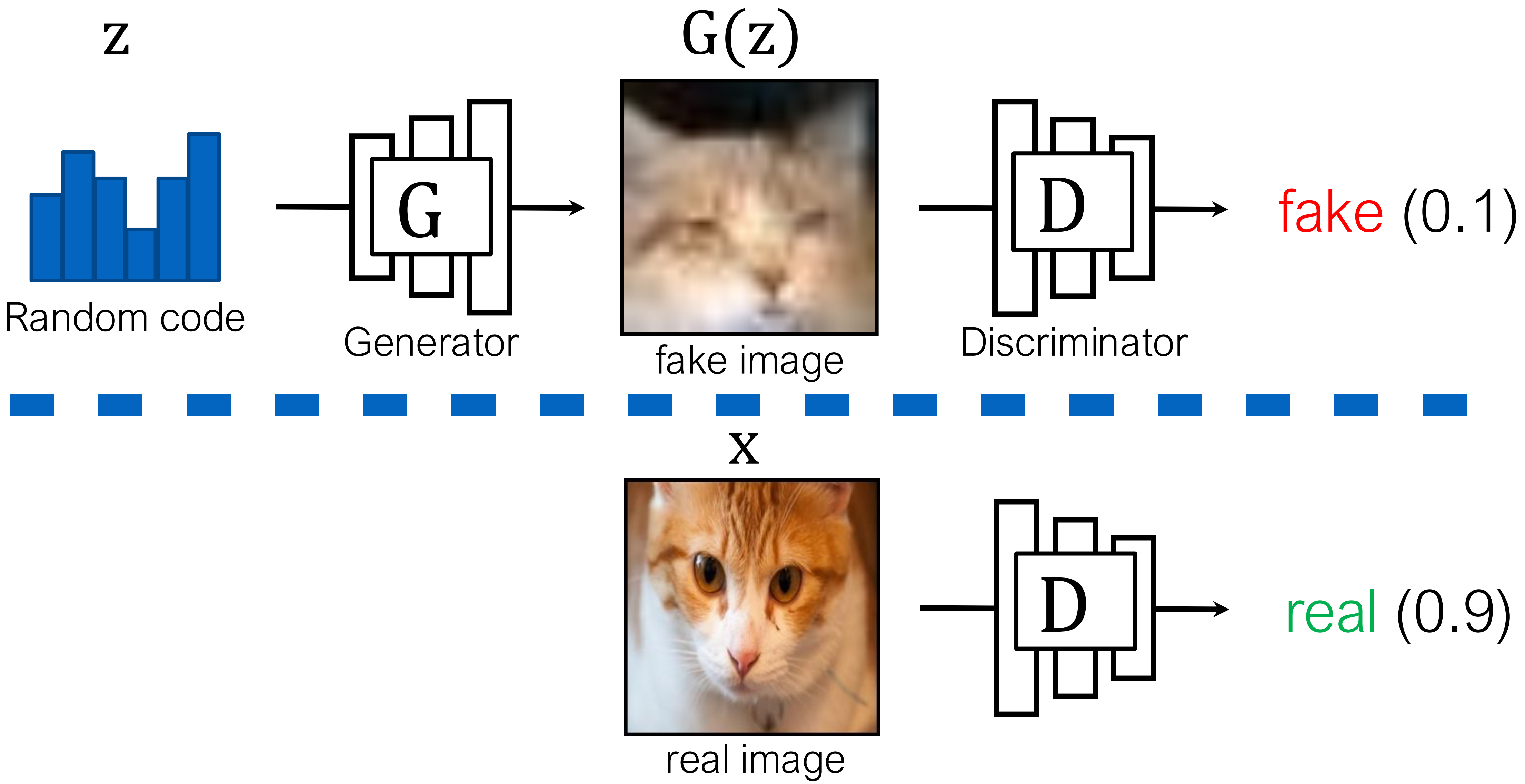
[Goodfellow et al. 2014]



Learning objective (GANs)

$$\min_{G} \max_{\boxed{D}} \boxed{\mathbb{E}_z[\log(1 - D(G(z)))]}$$

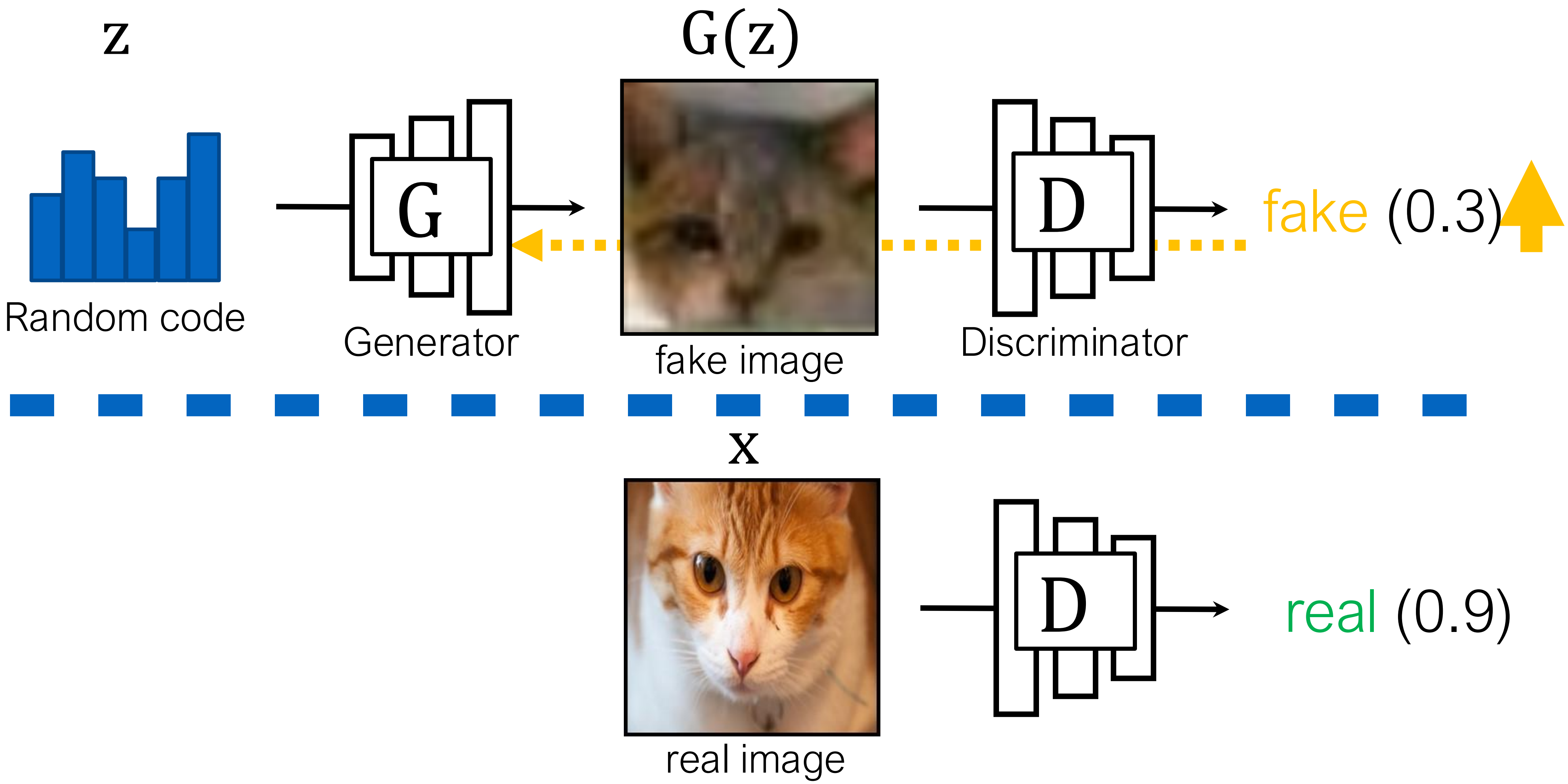
[Goodfellow et al. 2014]



Learning objective (GANs)

$$\min_{G} \max_{\boxed{D}} \boxed{\mathbb{E}_z[\log(1-D(G(z)))]} + \boxed{\mathbb{E}_x[\log D(x)]}$$

[Goodfellow et al. 2014]

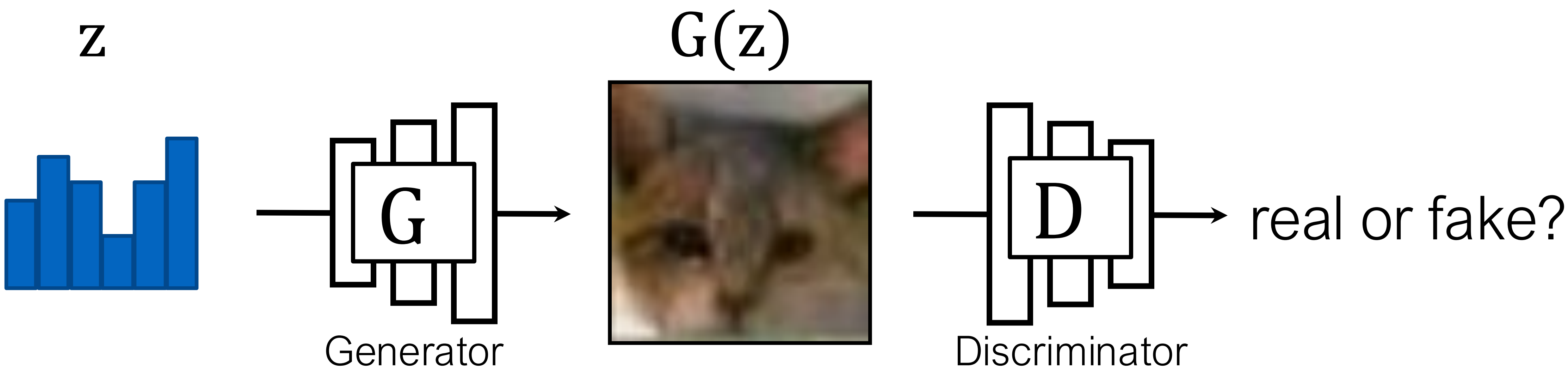


Learning objective (GANs)

$$\min_{\boxed{G}} \max_{D} \boxed{\mathbb{E}_z[\log(1-D(G(z))]} + \mathbb{E}_x[\log D(x)]$$

[Goodfellow et al. 2014]

GANs Training Breakdown



G tries to synthesize fake images that fool D

D tries to identify the fakes

- Training: iterate between training D and G with backprop.
- Global optimum when G reproduces data distribution.

[Goodfellow et al., 2014]

What has driven GAN progress?

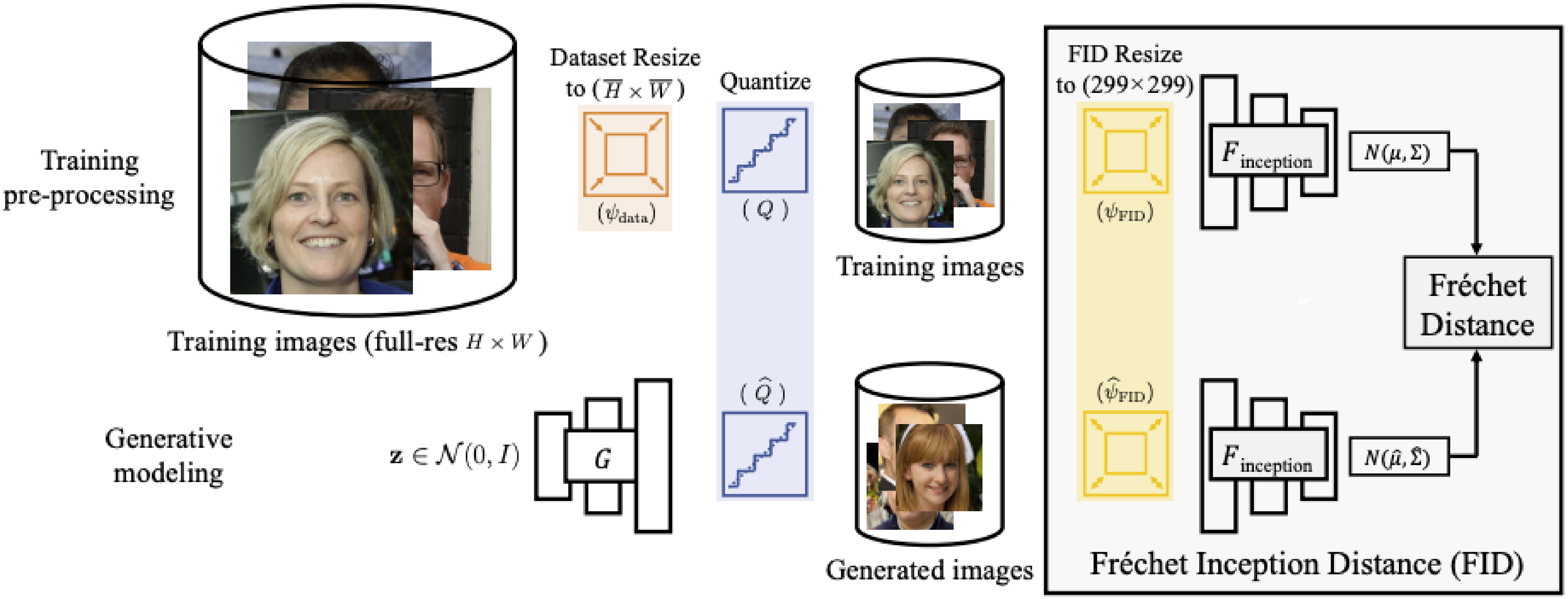
Ian Goodfellow @goodfellow_ian · Jan 14

4.5 years of **GAN progress** on face generation. arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434 arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196 arxiv.org/abs/1812.04948

What has driven GAN progress?

Samples from StyleGAN2 [Karras et al., CVPR 2020]

GANs evaluation (FID)



Fréchet Inception Distance (FID)

$$\mathbf{FID} = ||\mu - \hat{\mu}||_2^2 + \mathbf{Tr}(\Sigma + \widehat{\Sigma} - 2(\Sigma\widehat{\Sigma})^{1/2})$$

Also called 2-Wasserstein distance

What has driven GAN progress?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding

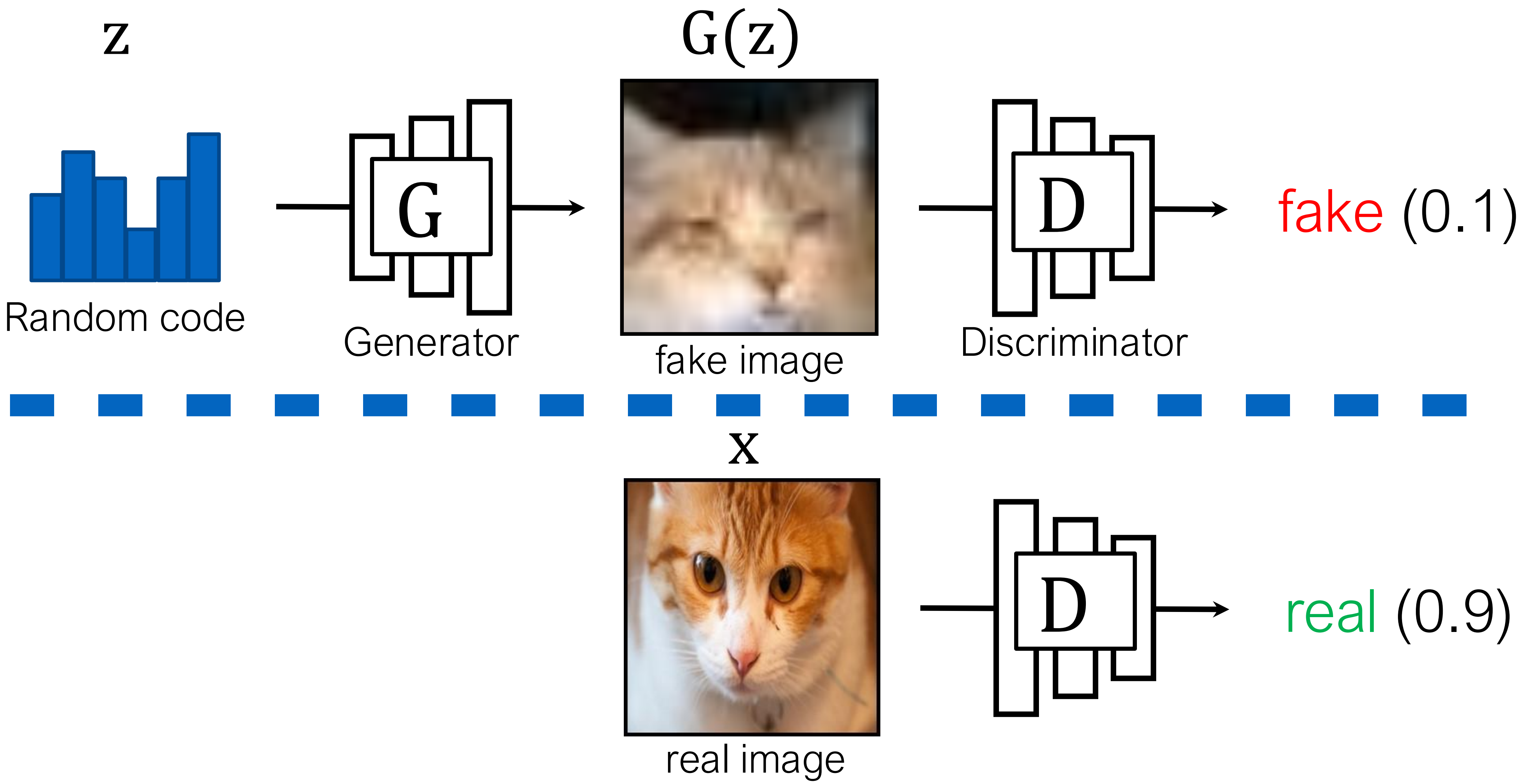
Which topics are easy to publish?

A. Loss functions
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding

Which topics are easy to publish?

A. **Loss functions**
B. Network architectures (G/D)
C. Training methods
D. Data
E. GPUs
F. Funding

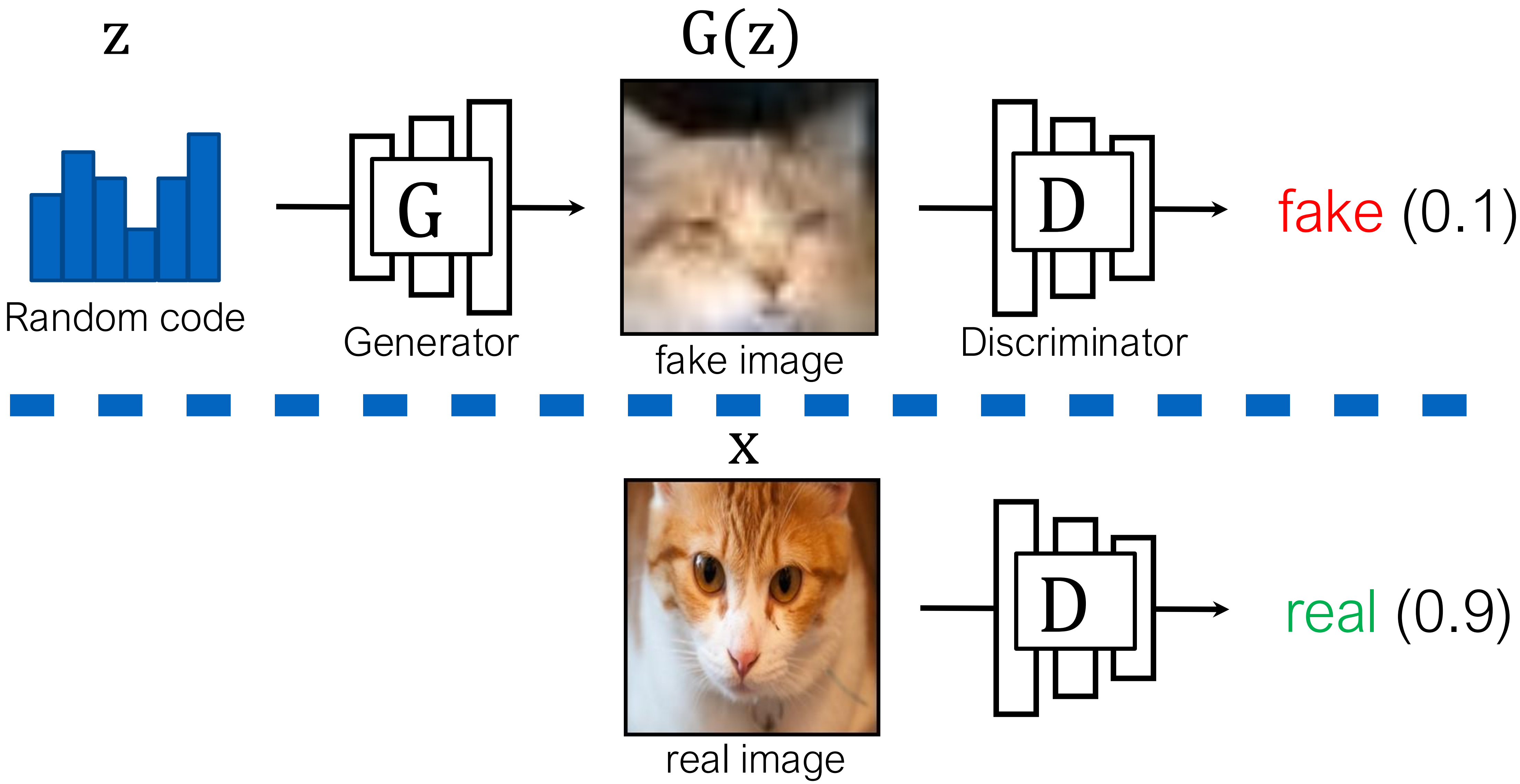
Loss functions



Learning objective (GANs)

$$\min_{G}\max_{D}\boxed{\mathbb{E}_z[\log(1-D(G(z)))]}+\boxed{\mathbb{E}_x[\log D(x)]}$$

[Goodfellow et al. 2014]



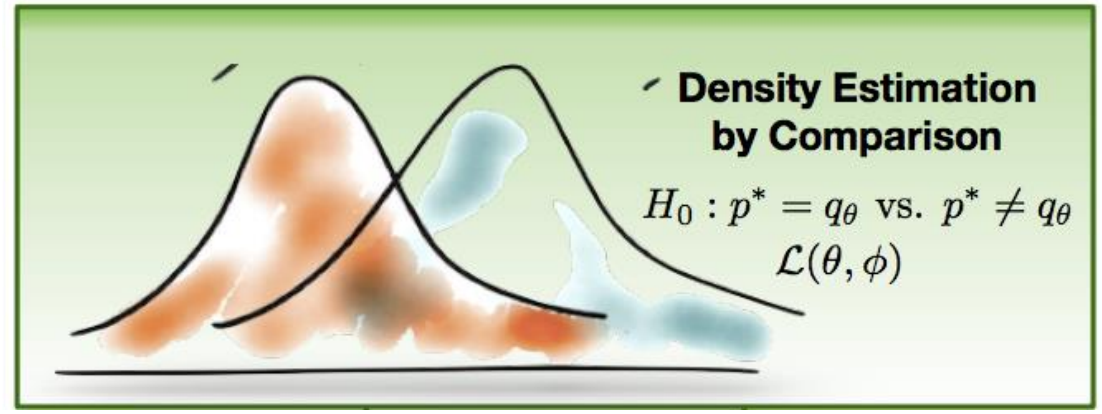
Learning objective (GANs variants)

$$\min_{G} \max_{f_1,f_2} \boxed{\mathbb{E}_z[f_1(G(z))]} + \mathbb{E}_x \boxed{[f_2(x)]}$$

EBGAN, WGAN, LSGAN, etc

[Goodfellow et al. 2014]

Other divergences?



from [Mohamed & Lakshminarayanan 2017]

Convenient choice

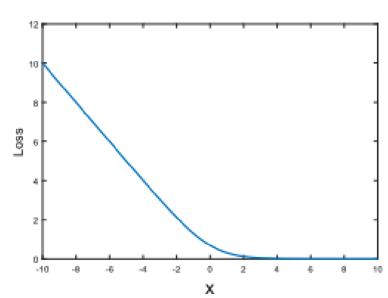
$$\min_G \max_{f_1, f_2} \boxed{\mathbb{E}_z[f_1(G(z))]} + \mathbb{E}_x \boxed{[f_2(x)]} \qquad \begin{aligned} f1 &= -f \\ f_2 &= f \end{aligned}$$

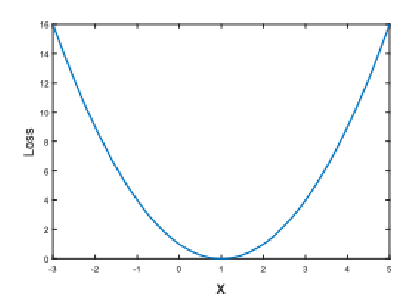
Different choices of f_1 and f_2 correspond to different divergence measures:

- Original GAN —> JSD
- Least-squares GAN —> Pearson chi-squared divergence

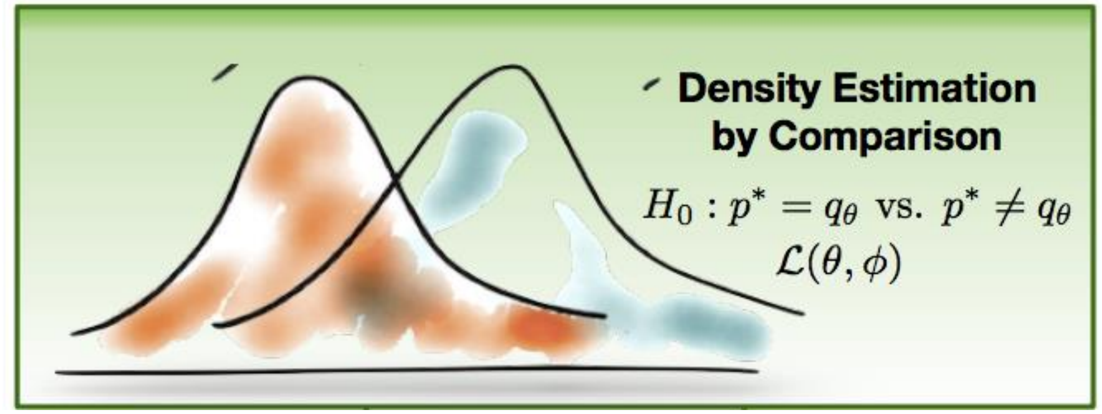
$$\min_D V_{\text{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}\big[(D(\boldsymbol{x}) - 1)^2\big] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_z}$$

$$\min_G V_{\text{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}\big[(D(G(\boldsymbol{z})) - 1)^2\big].$$





Other divergences?



from [Mohamed & Lakshminarayanan 2017]

Convenient choice

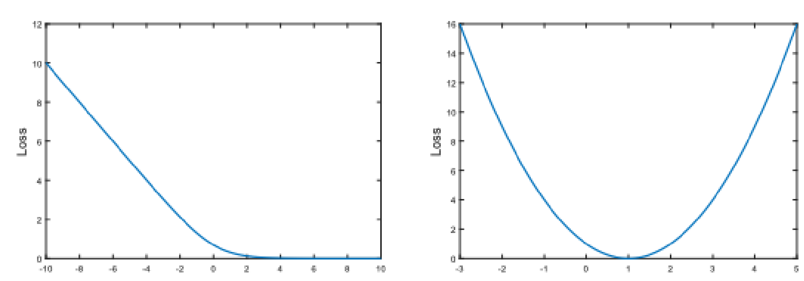
$$\min_{G} \max_{f_1, f_2} \boxed{\mathbb{E}_z[f_1(G(z))]} + \mathbb{E}_x \boxed{[f_2(x)]} \qquad \begin{aligned} f1 &= -f \\ f_2 &= f \end{aligned}$$

Different choices of f_1 and f_2 correspond to different divergence measures:

- Original GAN —> JSD
- Least-squares GAN —> Pearson chi-squared divergence

$$\min_{D} V_{\mathrm{LSGAN}}(D) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}(\boldsymbol{x})}\left[(D(\boldsymbol{x})-1)^2\right] + \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_z}$$

$$\min_{G} V_{\mathrm{LSGAN}}(G) = \frac{1}{2}\mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}\left[(D(G(\boldsymbol{z}))-1)^2\right].$$



Other divergences?

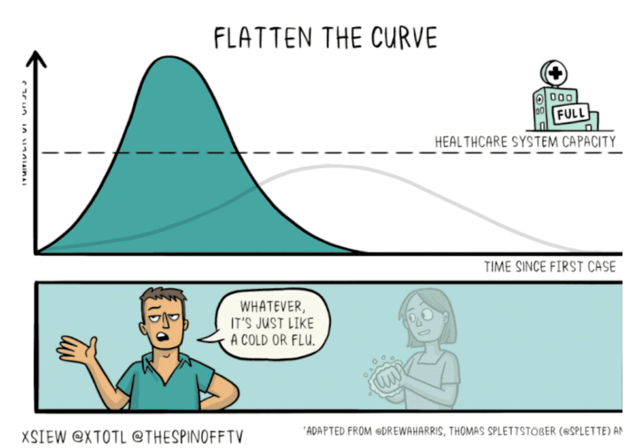
$$KL(p_{\texttt{data}}||p_\theta) \longleftarrow \mathbb{E}_{x\sim p_{\texttt{data}}}[\log p_\theta(x)]$$

$KL(p_\theta||p_{\texttt{data}})$ ⟵ Reverse KL — mode seeking, intractable

$JS(p_{\texttt{data}}, p_\theta)$ ⟵ Jensen-Shannon, original GAN

$$W(p_{\texttt{data}}, p_\theta) = \inf_{\gamma\in\Pi(p_{\texttt{data}}, p_\theta)} \mathbb{E}_{(x,y)\sim\gamma}[||x-y||]$$ ⟵ Wasserstein

Earth-Mover (EM) distance
/ Wasserstein distance



Wasserstein GAN

[Arjovsky, Chintala, Bottou 2017]

Lipschitz continuity

$|f(x) - f(y)| \leq |x - y|$

$$\arg\min_{G} \max_{\|f\|_L \leq 1} \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\boxed{-f(G(\mathbf{z}))} + \boxed{f(\mathbf{x})} \right]$$

$$W(p_{\texttt{data}}, p_\theta) = \inf_{\gamma \in \Pi(p_{\texttt{data}}, p_\theta)} \mathbb{E}_{(x,y)\sim\gamma}[\|x - y\|]$$

wGAN GP [Gulrajani et al., 2018]:

$$\arg\min_{G} \max_{f} \mathbb{E}_{\mathbf{z},\mathbf{x}}[\boxed{-f(G(\mathbf{z}))} + \boxed{f(\mathbf{x})}] + \underbrace{\lambda \mathbb{E}_{\hat{\mathbf{x}}\sim\mathbb{P}_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})\|_2 - 1)^2]}_{\text{Gradient penalty (GP)}}$$

Spectral Normalization

[Miyato, Kataoka, Koyama, Yoshida 2018]

$$\bar{W}_{\mathrm{SN}}(W) := W/\sigma(W) \quad \sigma(A) := \max_{\boldsymbol{h}:\boldsymbol{h}\neq\boldsymbol{0}} \frac{\|A\boldsymbol{h}\|_2}{\|\boldsymbol{h}\|_2}$$

- W is the weight of one layer in the discriminator
- $\sigma(A)$ (spectral norm) is the largest singular value of A
  (If A is a square matrix, the largest eigenvalue)
- Effect: limit the amount of changes each layer introduces

$$\sigma(\hat{W}_{SN}) = 1$$

+ Lipschitz discriminator regularization (c.f. Wasserstein GAN)

Better objectives? optimizers?

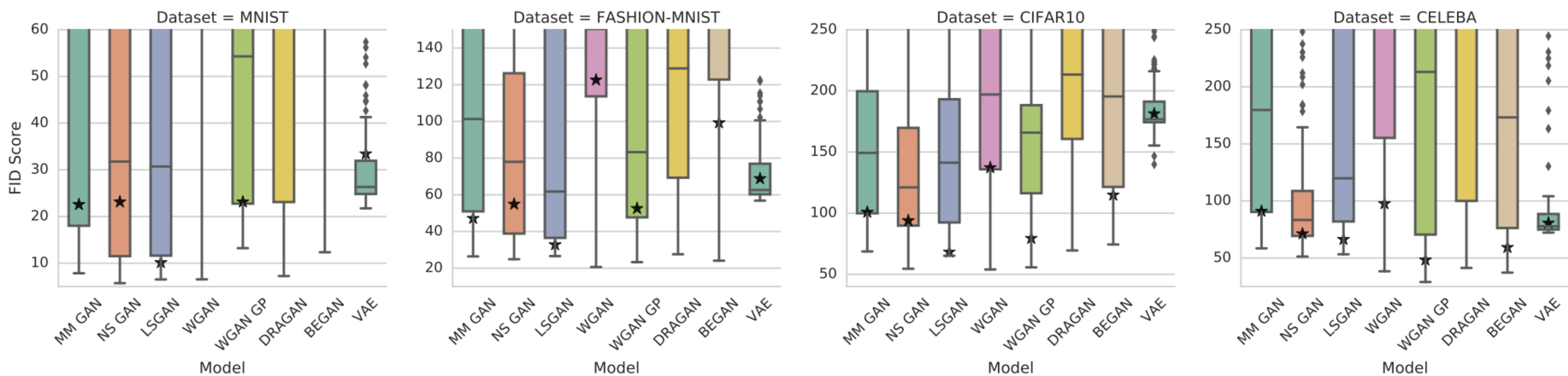


Figure 4: A *wide range* hyperparameter search (100 hyperparameter samples per model). Black stars indicate the performance of suggested hyperparameter settings. We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than others.

[“Are all GANs Created Equal?”, Lucic*, Kurach*, et al. 2018]

Original GAN loss/Hinge Loss/Least Square Loss
+ R1 gradient penalty (use 0 rather than 1)

Network architectures & Training methods

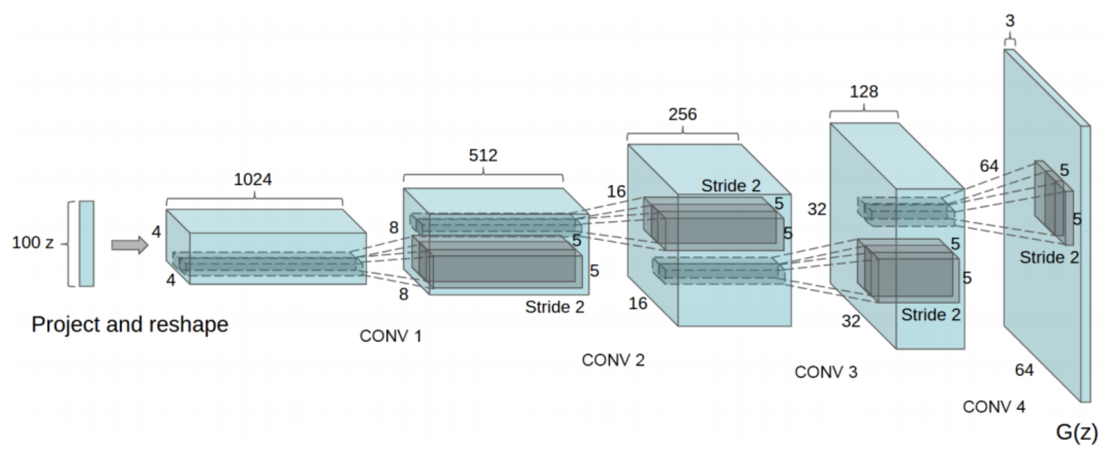
Better Architectures!

DCGAN
[Radford, Metz, Chintala 2016]

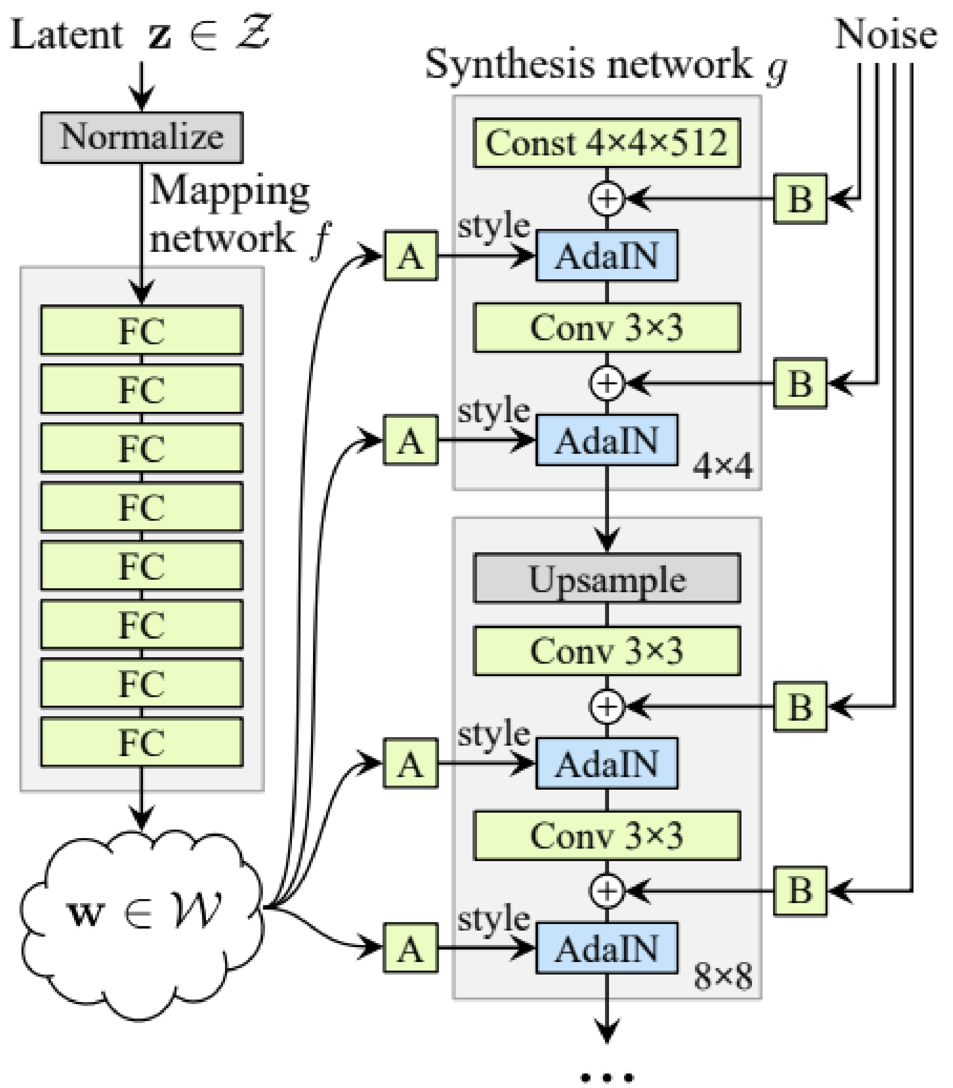
StyleGAN
[Karras, Laine, Aila 2019]

Better Architectures!

DCGAN
[Radford, Metz, Chintala 2016]

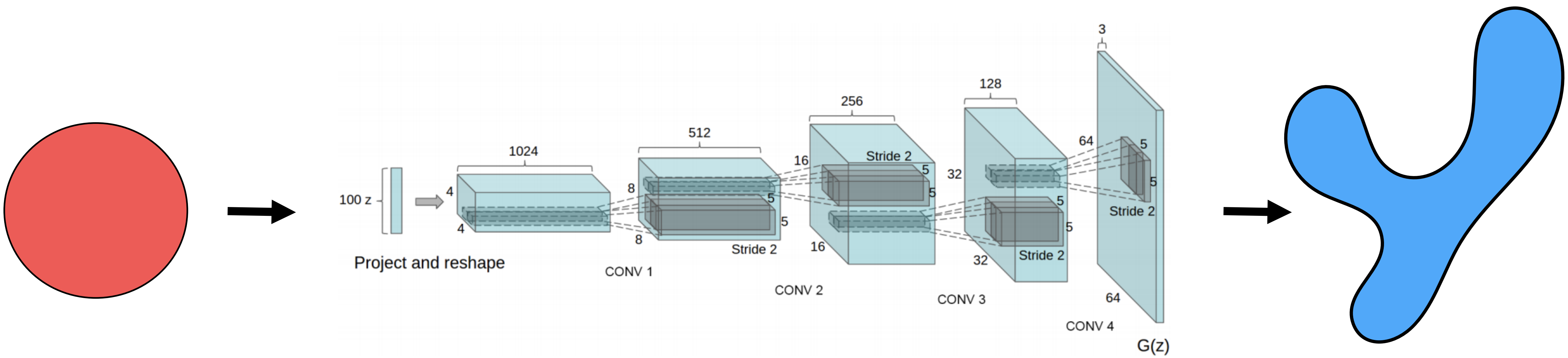


StyleGAN
[Karras, Laine, Aila 2019]



DCGAN

[Radford, Metz, Chintala 2015]



+ Convnet

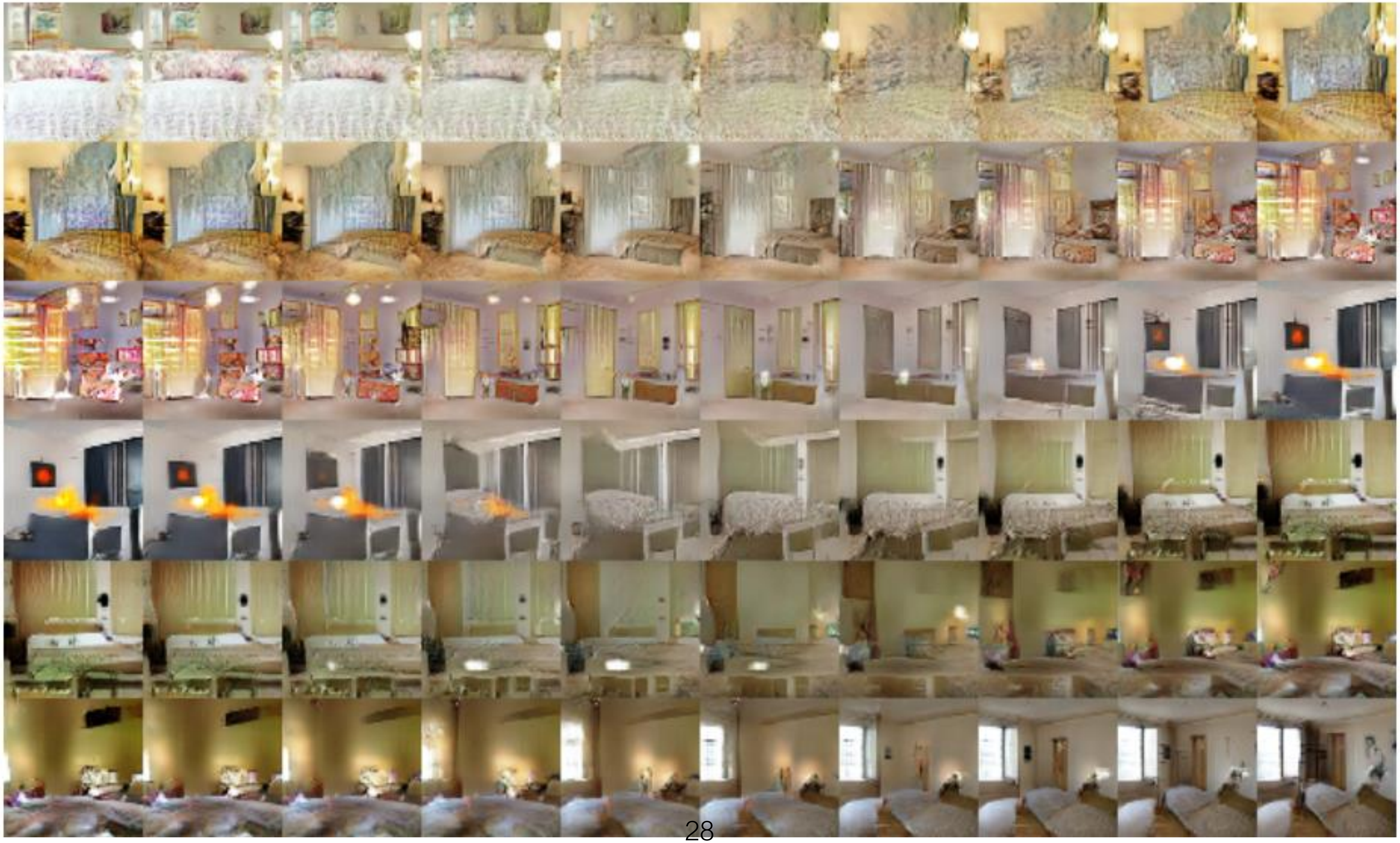
also see LAPGAN [Denton*, Chintala*, Szlam, Fergus 2015], which used a convnet

DCGAN

[Radford, Metz, Chintala 2015]

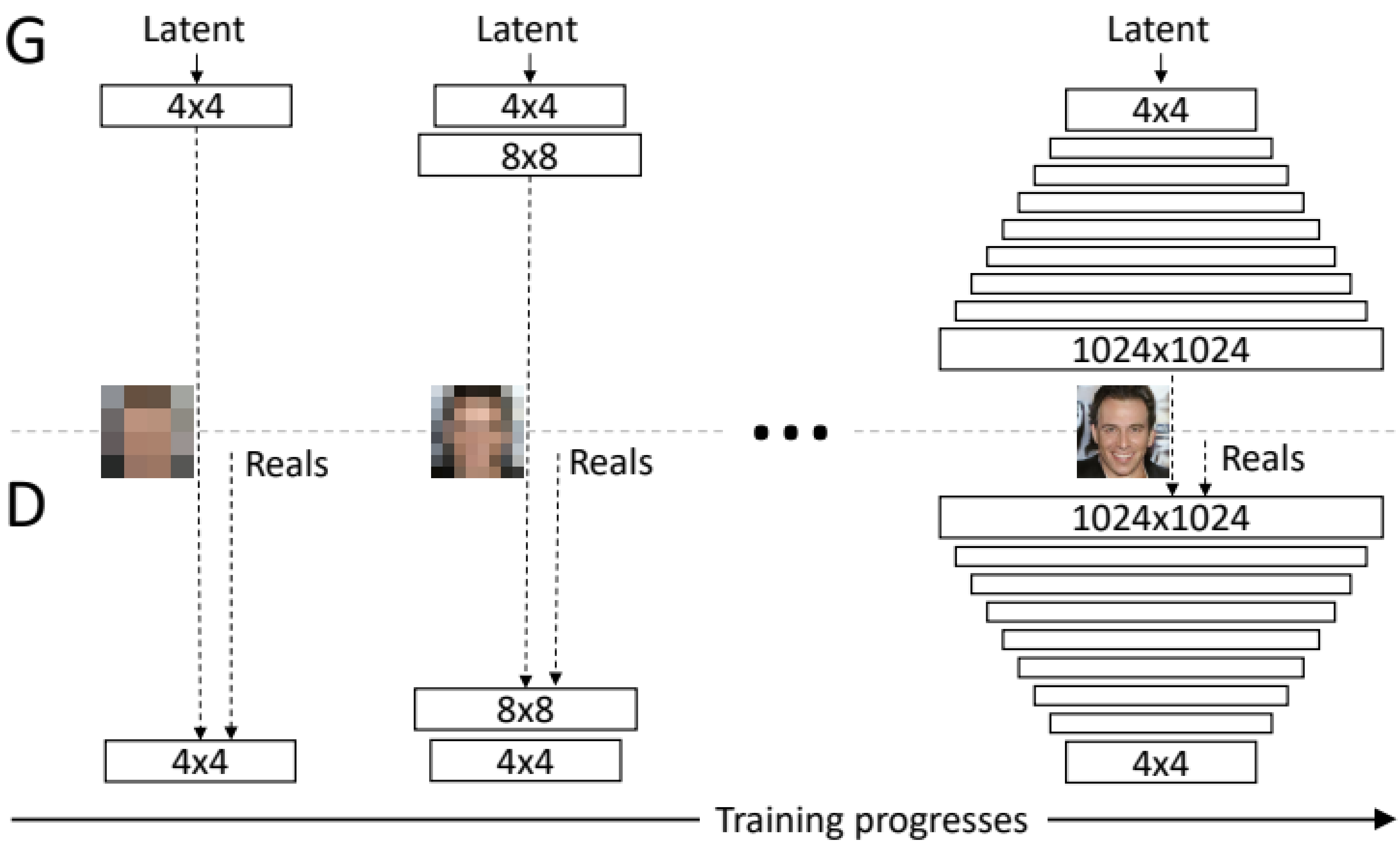
DCGAN

[Radford, Metz, Chintala 2015]



Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]



+ Coarse-to-fine
+ Progressive training

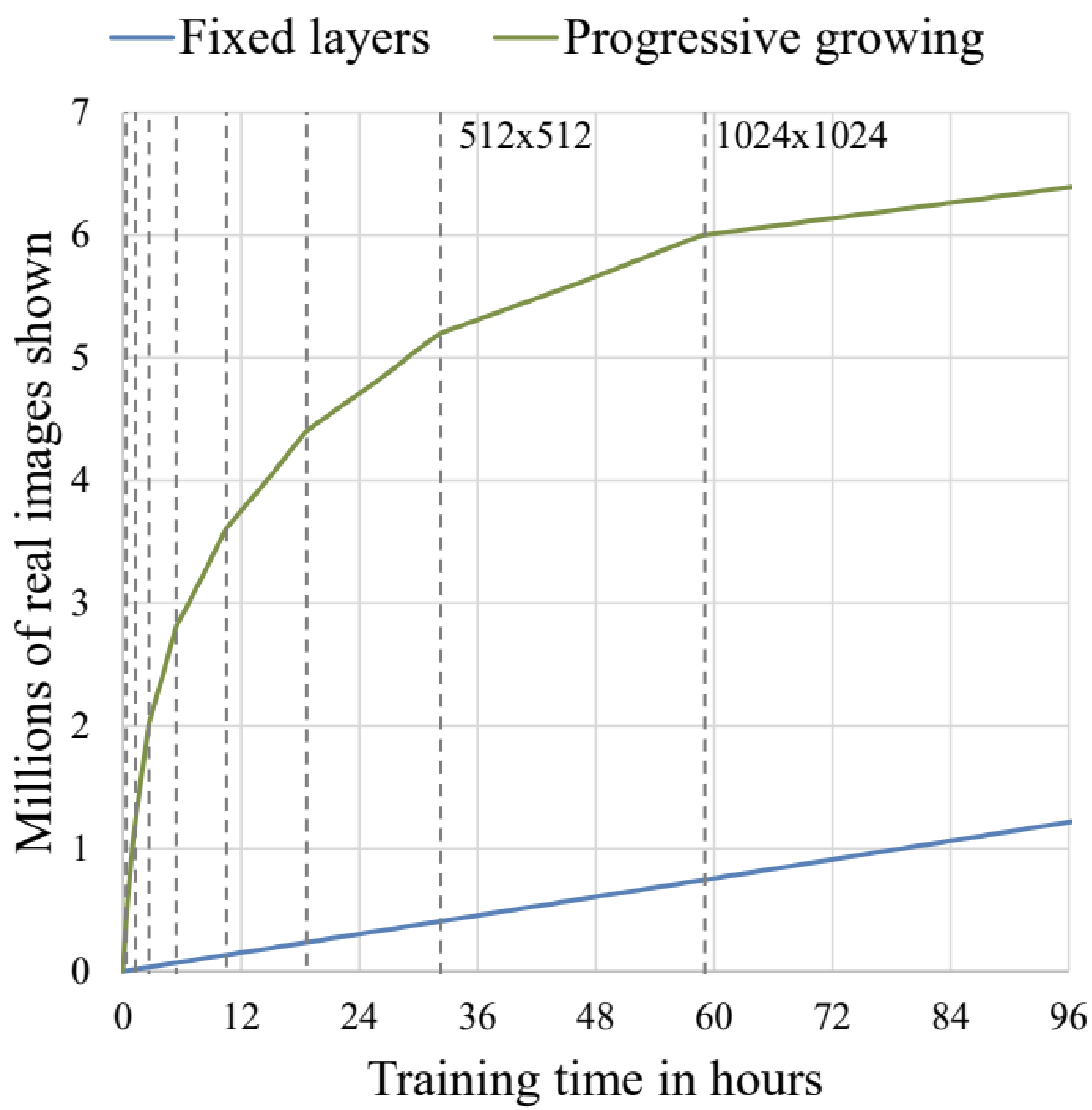
Computer vision can help quality: Gaussian Pyramid (HW1)

Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]

Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]

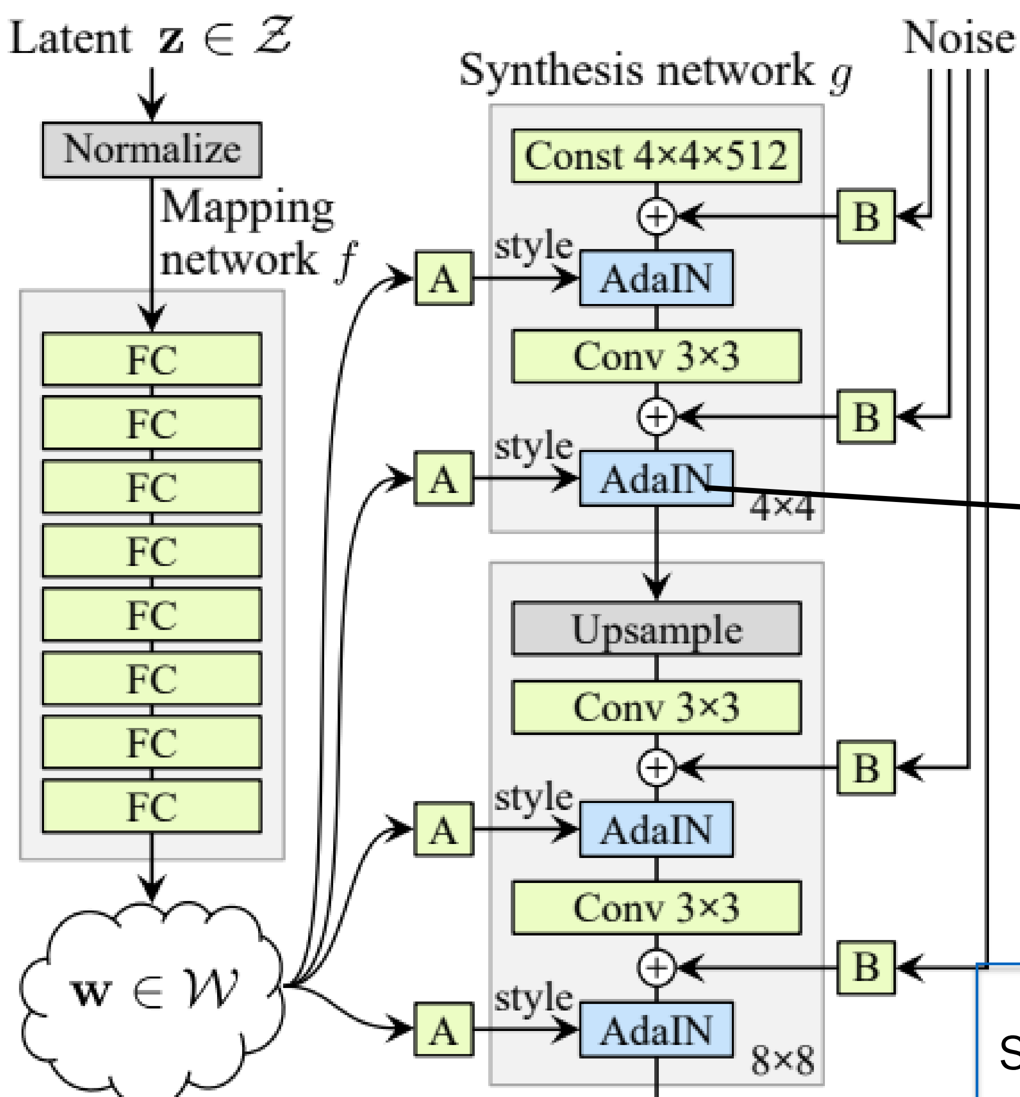


+ Coarse-to-fine
+ Progressive training

Computer vision can help speed: Gaussian Pyramid (HW1)

StyleGAN: Quality+ Control

[Karras, Laine, Aila. CVPR 2019]



- Multiscale "style" (noise)
- AdaIN layers

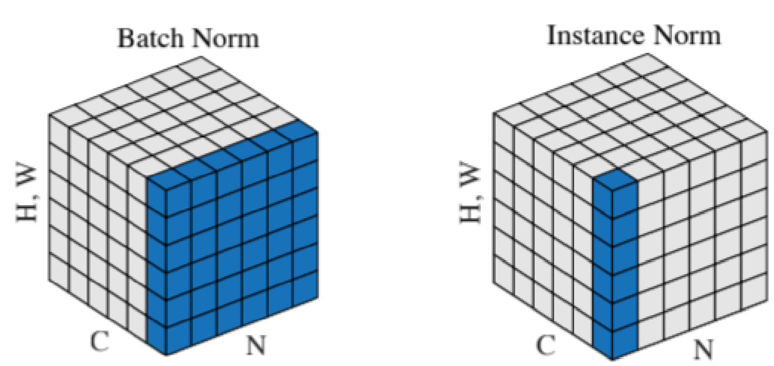
x: input feature

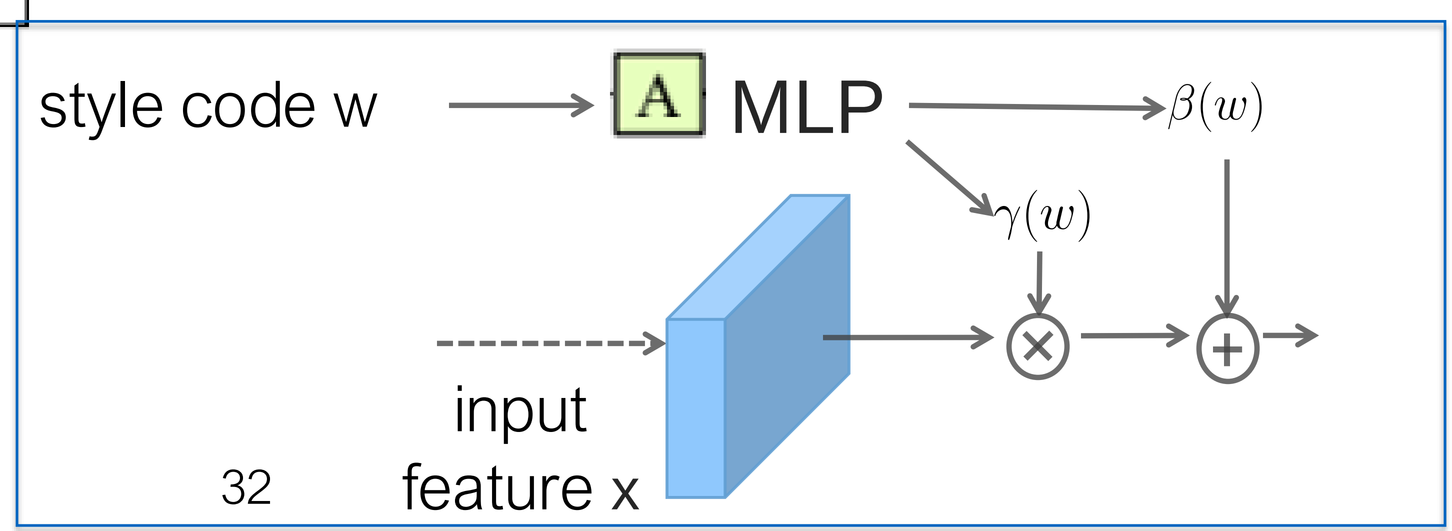
$$\mathrm{AdaIN}(x) = \gamma(w)\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta(w)$$

w: style code

- batch/instance normalization:

$$\mathrm{BN}(x) = \gamma\left(\frac{x-\mu(x)}{\sigma(x)}\right) + \beta$$





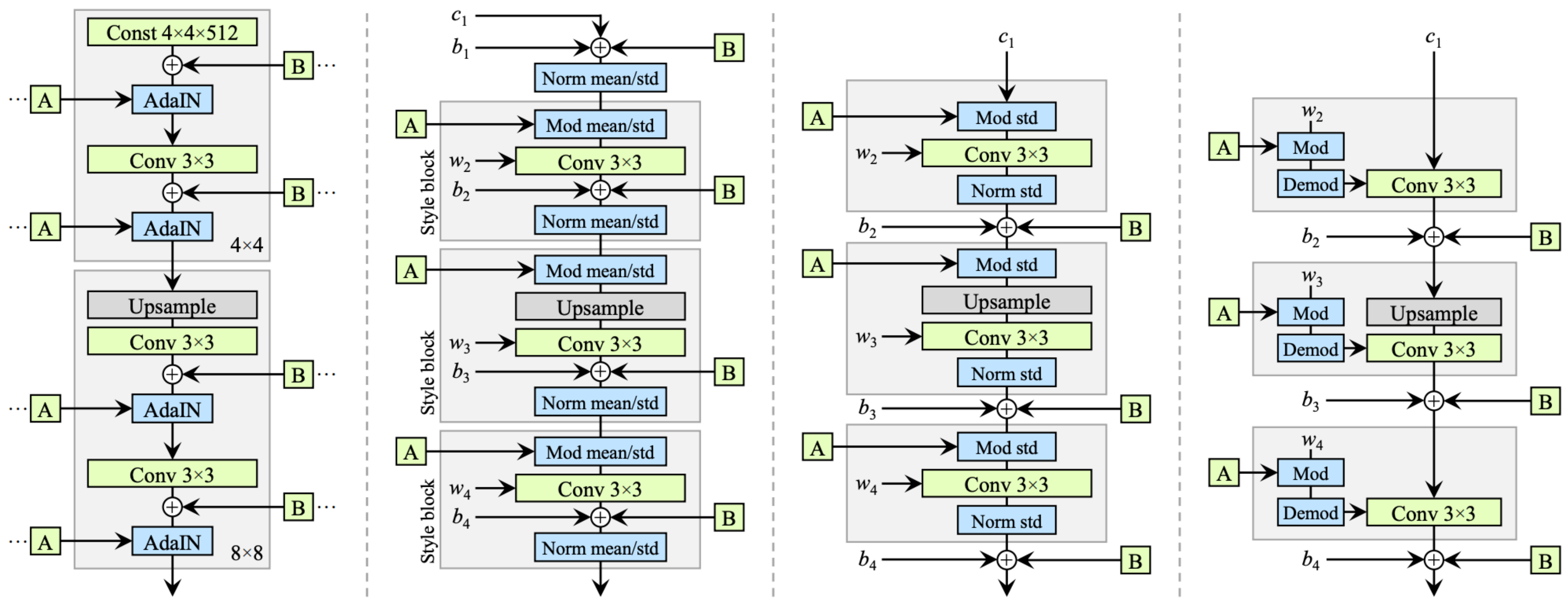
StyleGAN: Quality+ Control

[Karras, Laine, Aila. CVPR 2019]

StyleGAN2 and StyleGAN3

Analyzing and improving individual layers

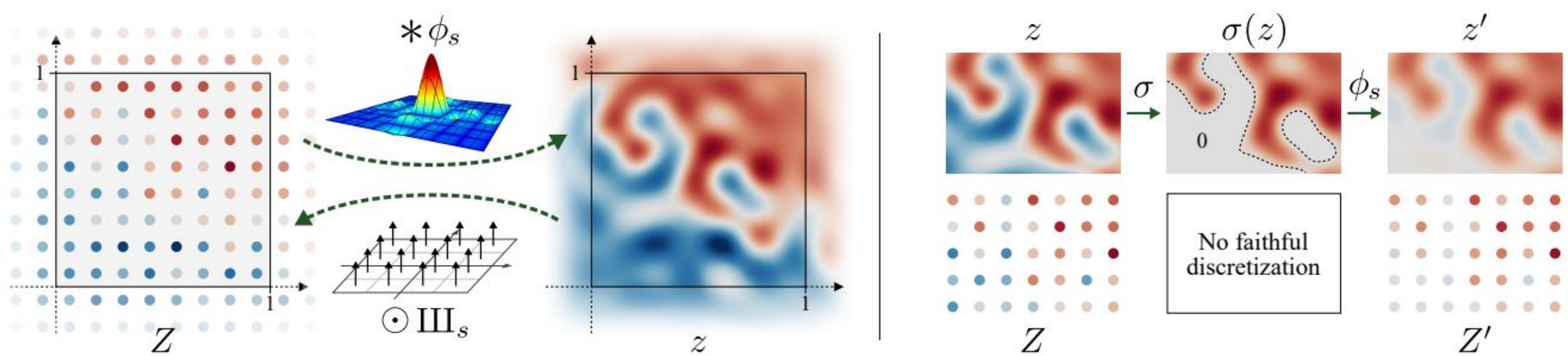
Weight Modulation Layers



(a) StyleGAN (b) StyleGAN (detailed) (c) Revised architecture (d) Weight demodulation

https://arxiv.org/abs/1912.04958

Alias-free layers



https://arxiv.org/abs/2106.12423

Data

Data alignment

- Work well for well-aligned objects and landscapes.

(a) (b) (c) (d) (e) (f)

Figure 8: Creating the CELEBA-HQ dataset. We start with a JPEG image (a) from the CelebA in-the-wild dataset. We improve the visual quality (b,top) through JPEG artifact removal (b,middle) and 4x super-resolution (b,bottom). We then extend the image through mirror padding (c) and Gaussian filtering (d) to produce a visually pleasing depth-of-field effect. Finally, we use the facial landmark locations to select an appropriate crop region (e) and perform high-quality resampling to obtain the final image at 1024×1024 resolution (f).

See more details in Appendix of Progressive GANs [Karras et al., ICLR 2018]

Aligned vs. unaligned data

Real images from aligned FFHQ

StyleGAN2 samples

Photo credit: StyleGAN2 [Karras et al., CVPR 2020]

Aligned vs. unaligned data

Real images from unaligned CelebA

StyleGAN2 samples

Photo credit: William Peebles (UC Berkeley)

Data are Expensive

FFHQ dataset: 70,000 selective post-processed human faces

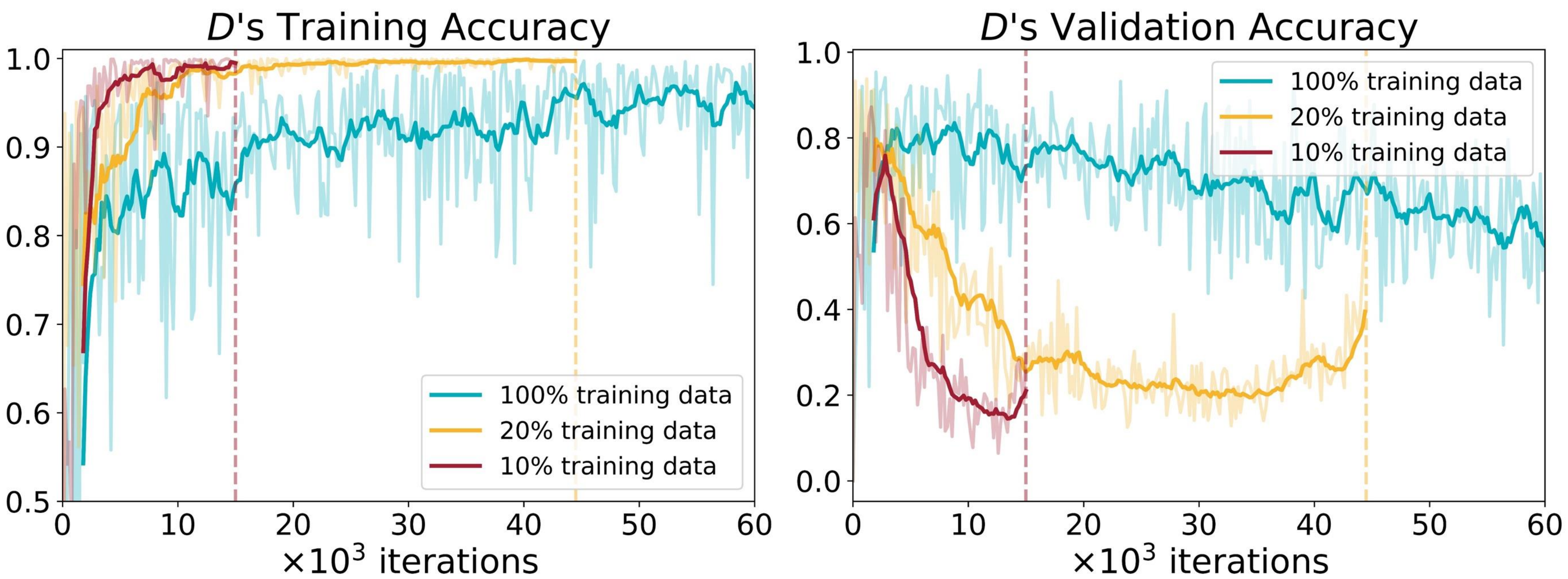
ImageNet dataset: millions of images from diverse categories

Months or even years to collect the data,
along with *prohibitive* annotation costs.

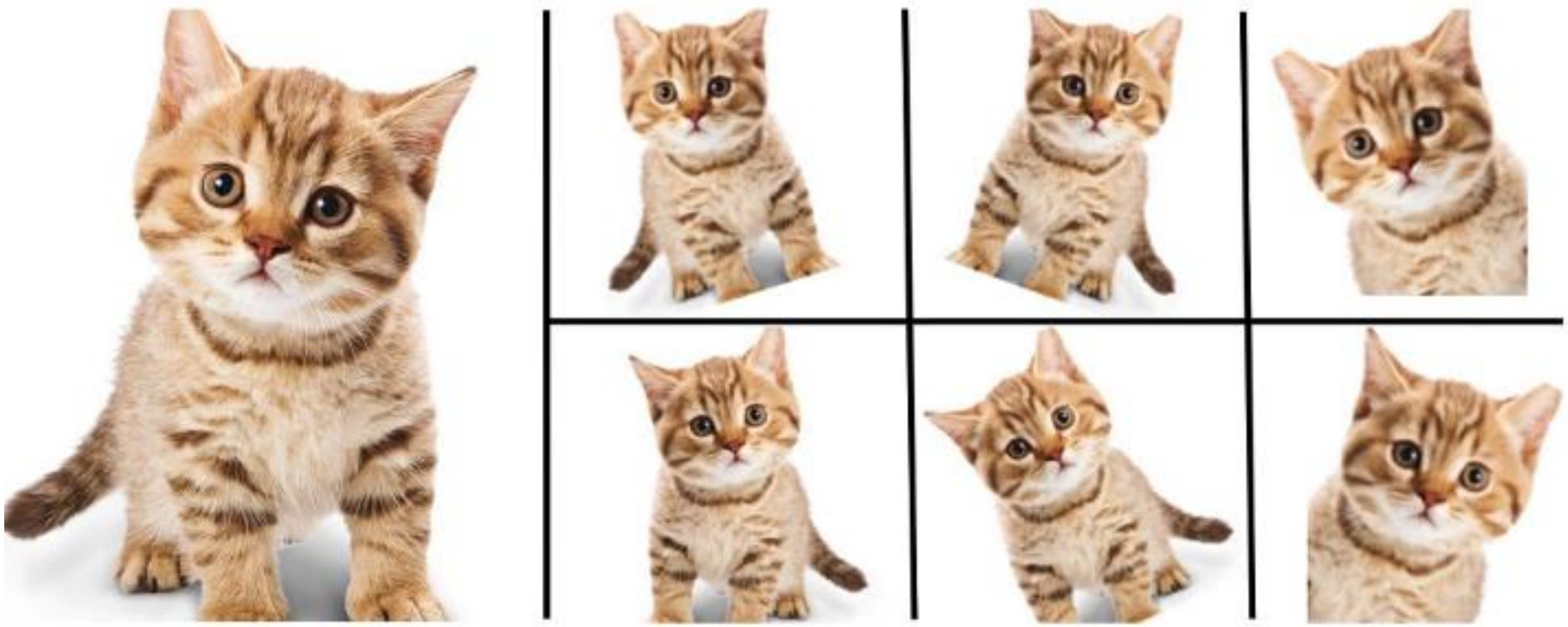
GANs Heavily Deteriorate Given Limited Data

Generated samples of StyleGAN2 (Karras et al.)
using only hundreds of images

Discriminator Overfitting



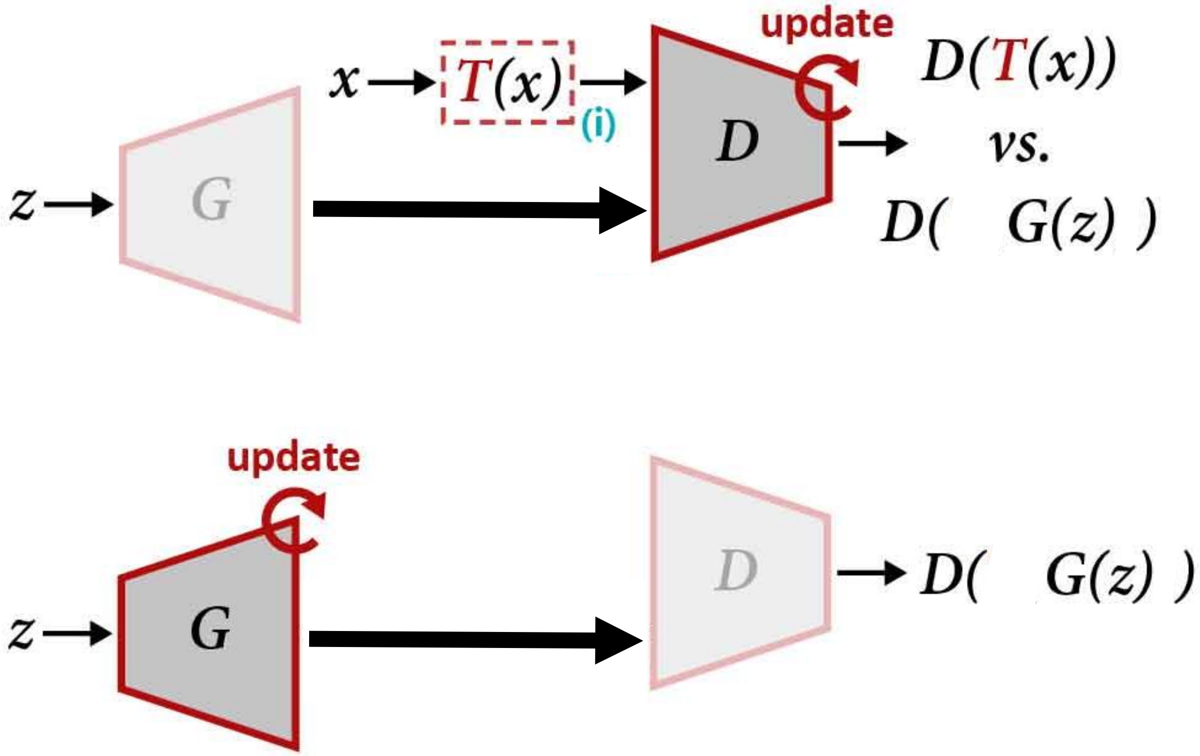
Data Augmentation



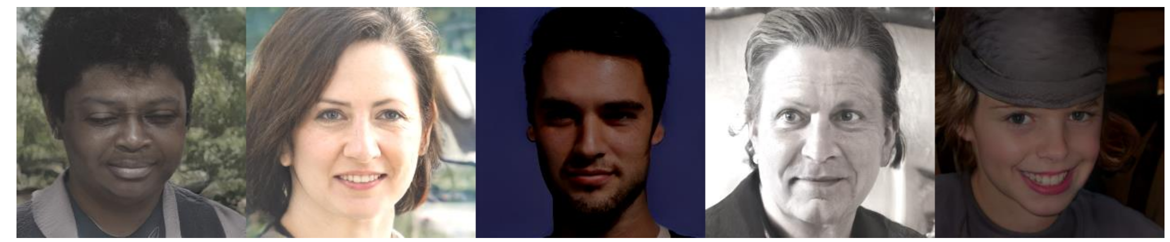
Data augmentation: enlarge datasets without collecting new samples.

How to Augment GANs?

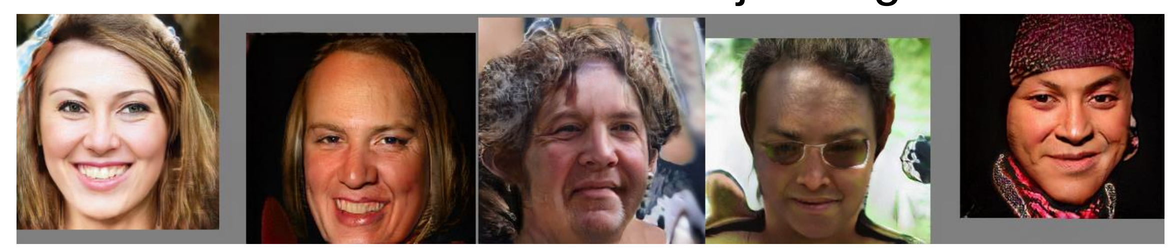
#1 Approach: Augment reals only



Generated images



Artifacts from Color jittering

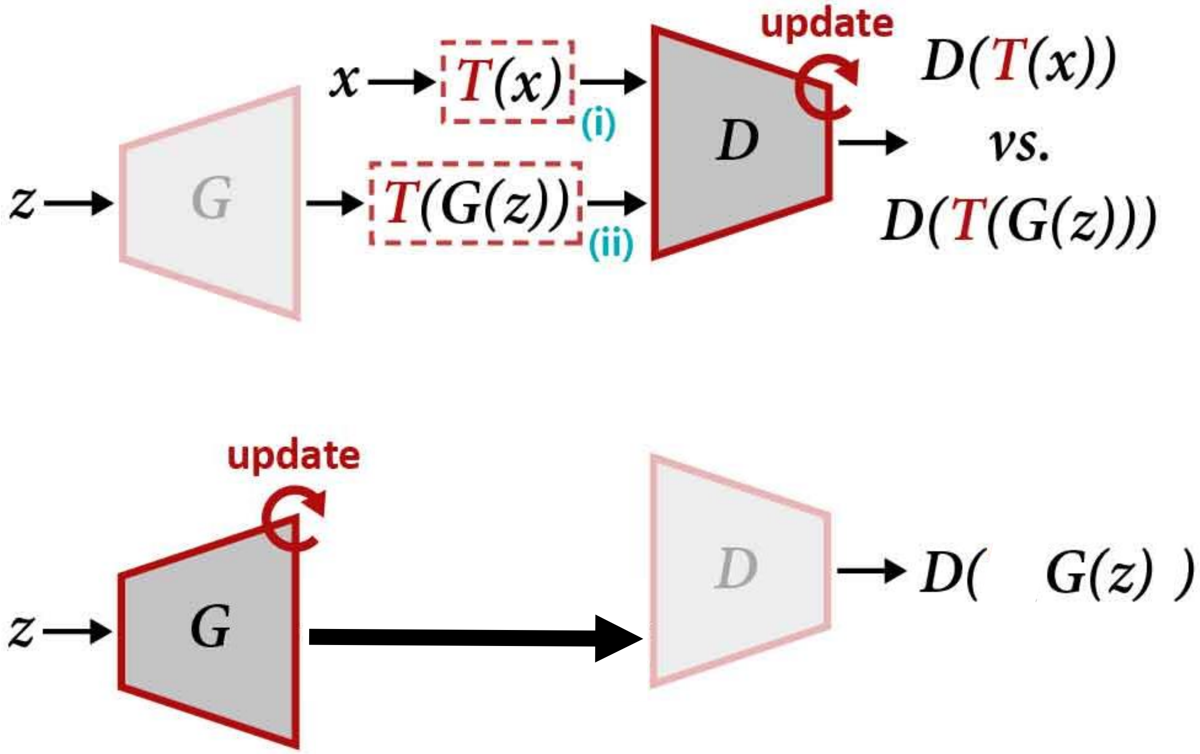


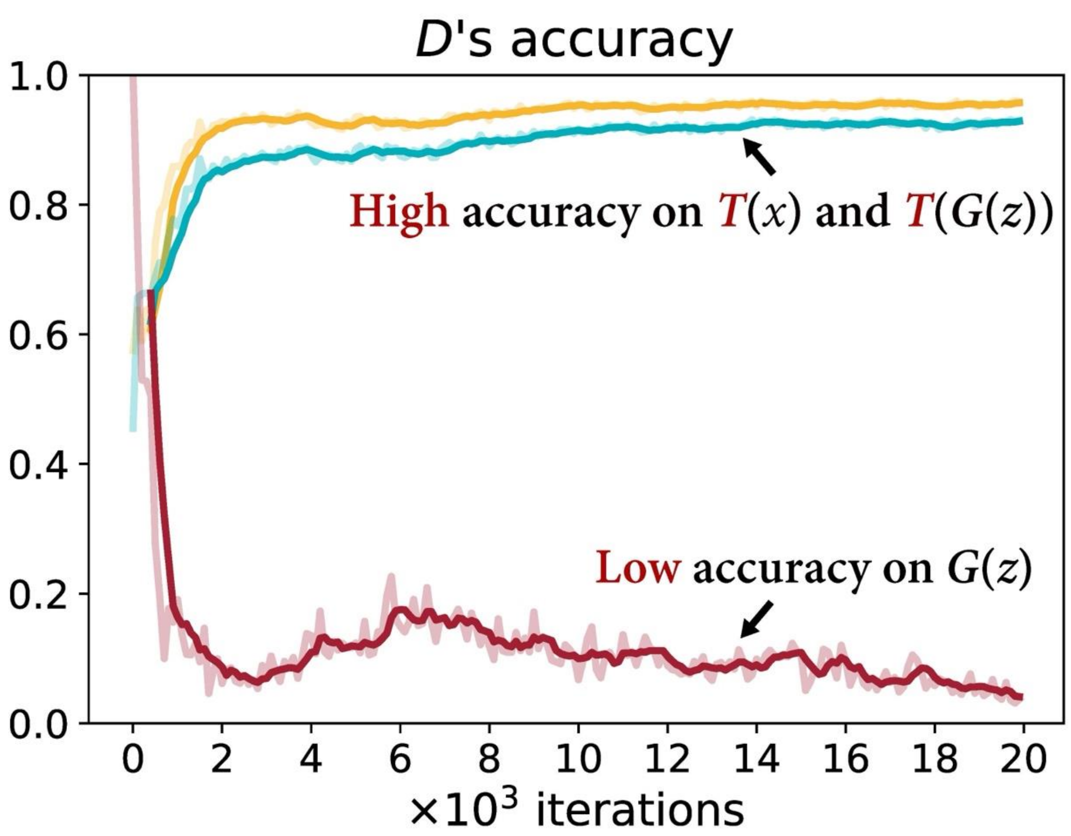
Artifacts from Translation

Artifacts from Cutout (DeVries et al.)

Augment reals only: the same artifacts appear on the generated images.

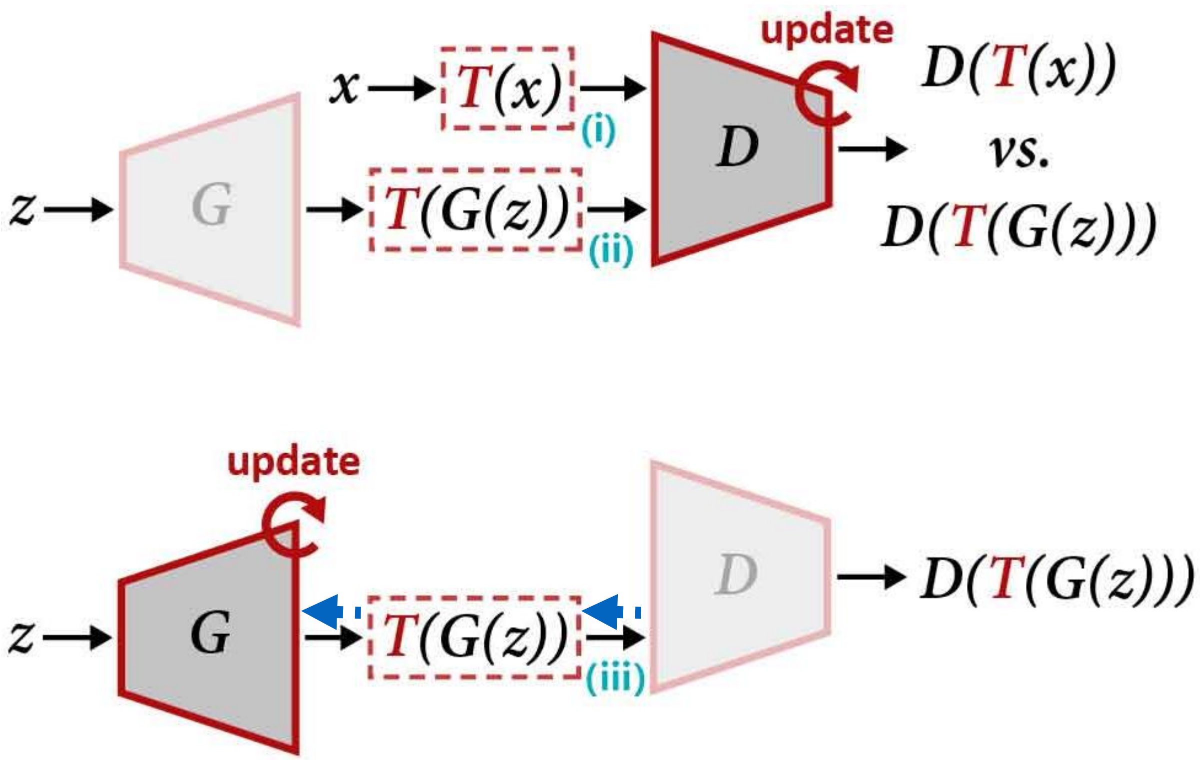
#2 Approach: Augment reals & fakes for ***D*** only

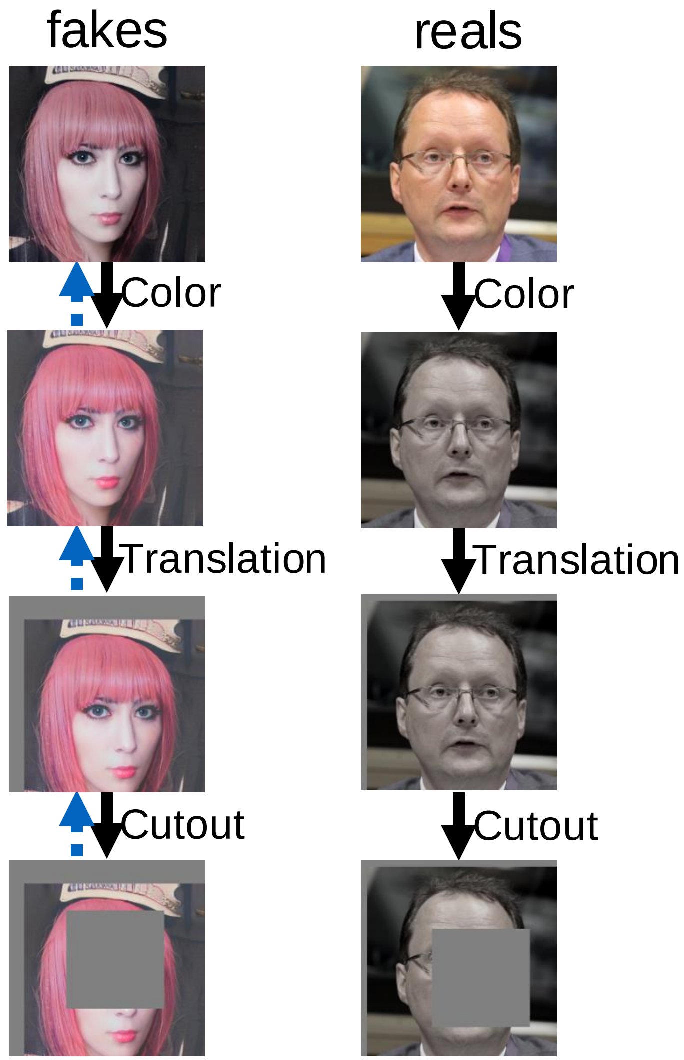




Augment ***D*** only: the unbalanced optimization cripples training.

#3 Approach: Differentiable Augmentation

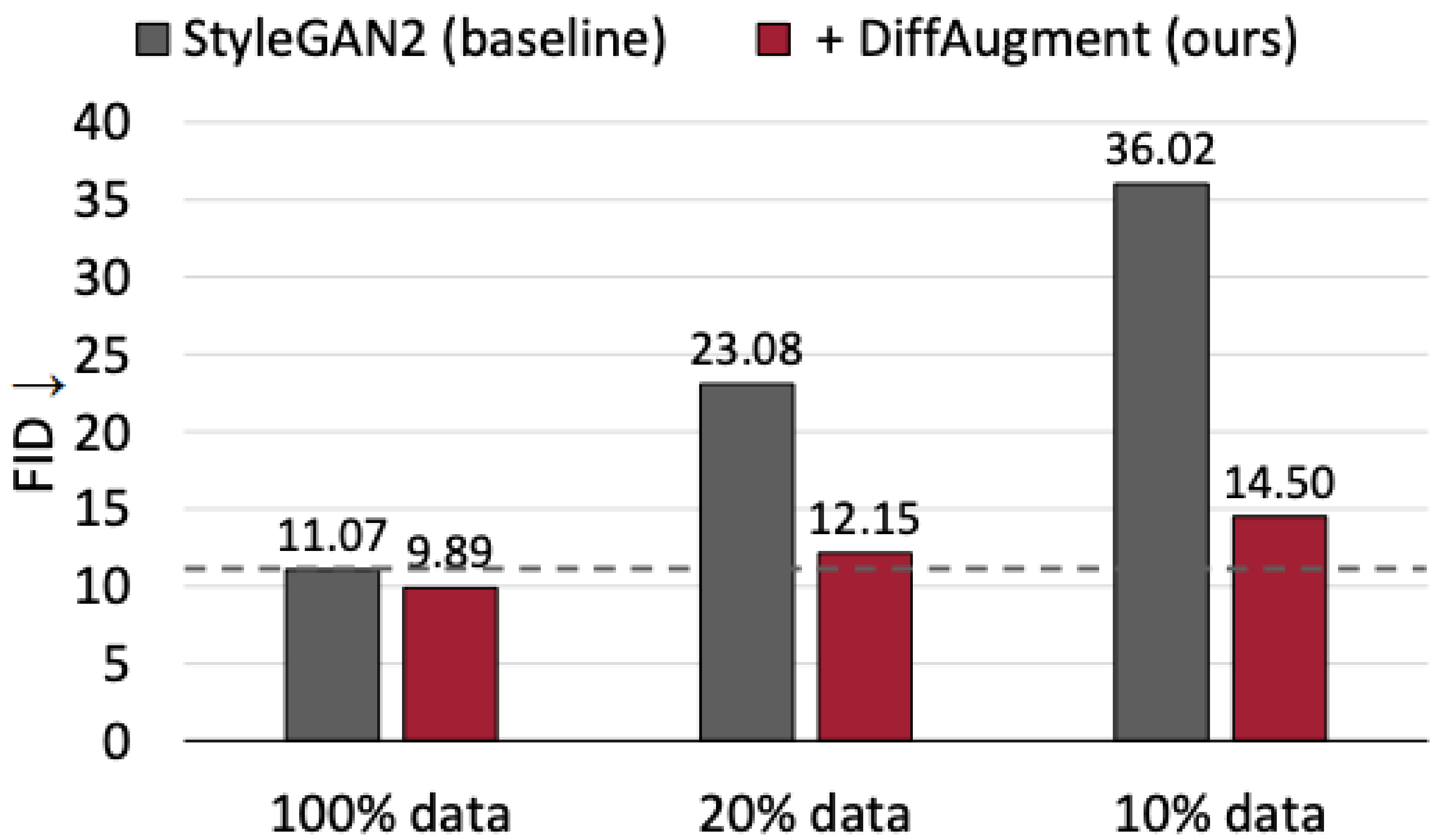




Our approach (DiffAugment): Augment reals + fakes for both $\boldsymbol{D}$ and $\boldsymbol{G}$

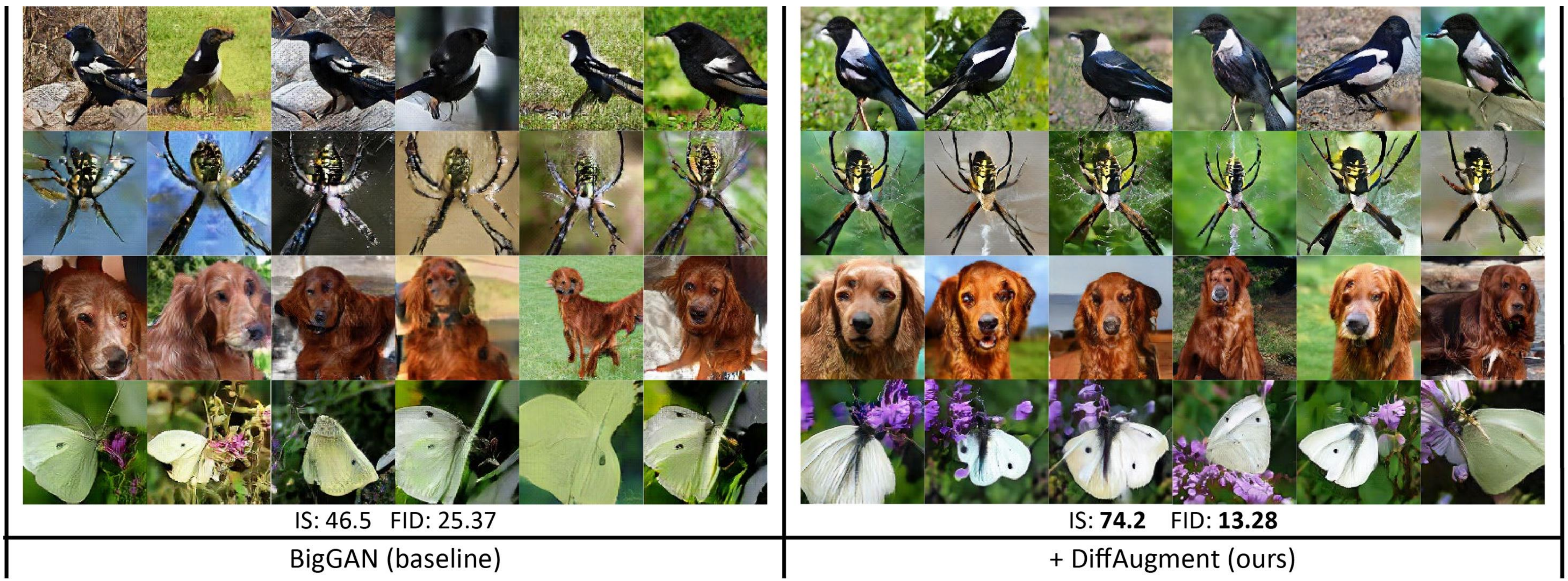
Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

CIFAR-10 (unconditional GANs)



Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

ImageNet Generation (25% training data)



Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

Low-Shot Generation

100-Shot Interpolation

The smooth interpolation results suggest little overfitting of our method even given *only 100 images of Obama, grumpy cat, panda, the Bridge of Sighs, the Medici Fountain, the Temple of Heaven, and Wuzhen.*

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

```python
from DiffAugment_pytorch import DiffAugment
# from DiffAugment_tf import DiffAugment
policy = 'color,translation,cutout' # If your dataset is as small as ours (e.g.,
# hundreds of images), we recommend using the strongest Color + Translation + Cutout.
# For large datasets, try using a subset of transformations in ['color', 'translation', 'cutout'].
# Welcome to discover more DiffAugment transformations!

...
# Training loop: update D
reals = sample_real_images() # a batch of real images
z = sample_latent_vectors()
fakes = Generator(z) # a batch of fake images
real_scores = Discriminator(DiffAugment(reals, policy=policy))
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating D's loss based on real_scores and fake_scores...
...

...
# Training loop: update G
z = sample_latent_vectors()
fakes = Generator(z) # a batch of fake images
fake_scores = Discriminator(DiffAugment(fakes, policy=policy))
# Calculating G's loss based on fake_scores...
...
```

Differentiable Augmentation for Data-Efficient GAN Training. Zhao et al., NeurIPS 2020

Data Augmentation for GANs

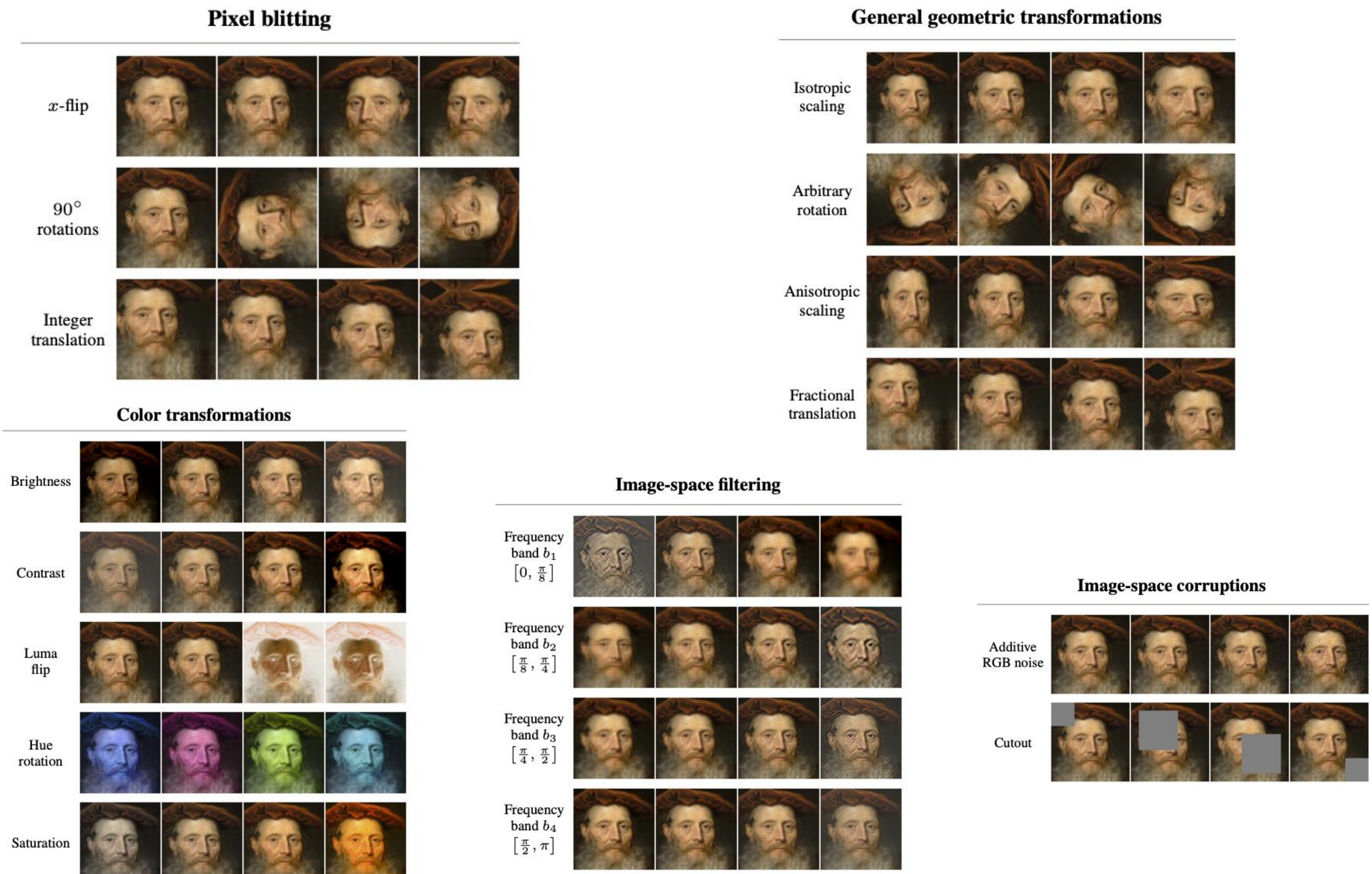
- *Differentiable Augmentation for Data-Efficient GAN Training* (DiffAugment). Zhao et al., NeurIPS 2020.

- *Training Generative Adversarial Networks with Limited Data.* (StyleGAN2-ADA). Karras et al., NeurIPS 2020.

- *On Data Augmentation for GAN Training*. Tran et al., IEEE TIP, 2020.

- *Image Augmentations for GAN Training*. Zhao et al., arXiv, 2020.

StyleGAN2-ADA



Training Generative Adversarial Networks with Limited Data [Karras et al., NeurIPS 2020]

StyleGAN2-ADA

Adaptative data augmentation

$$r_t = \mathbb{E}[\mathbf{sign}(D_{\text{train}})]$$

$r_t = 0$ no overfitting, decrease augmentation
r=1 complete overfitting, increase augmentation

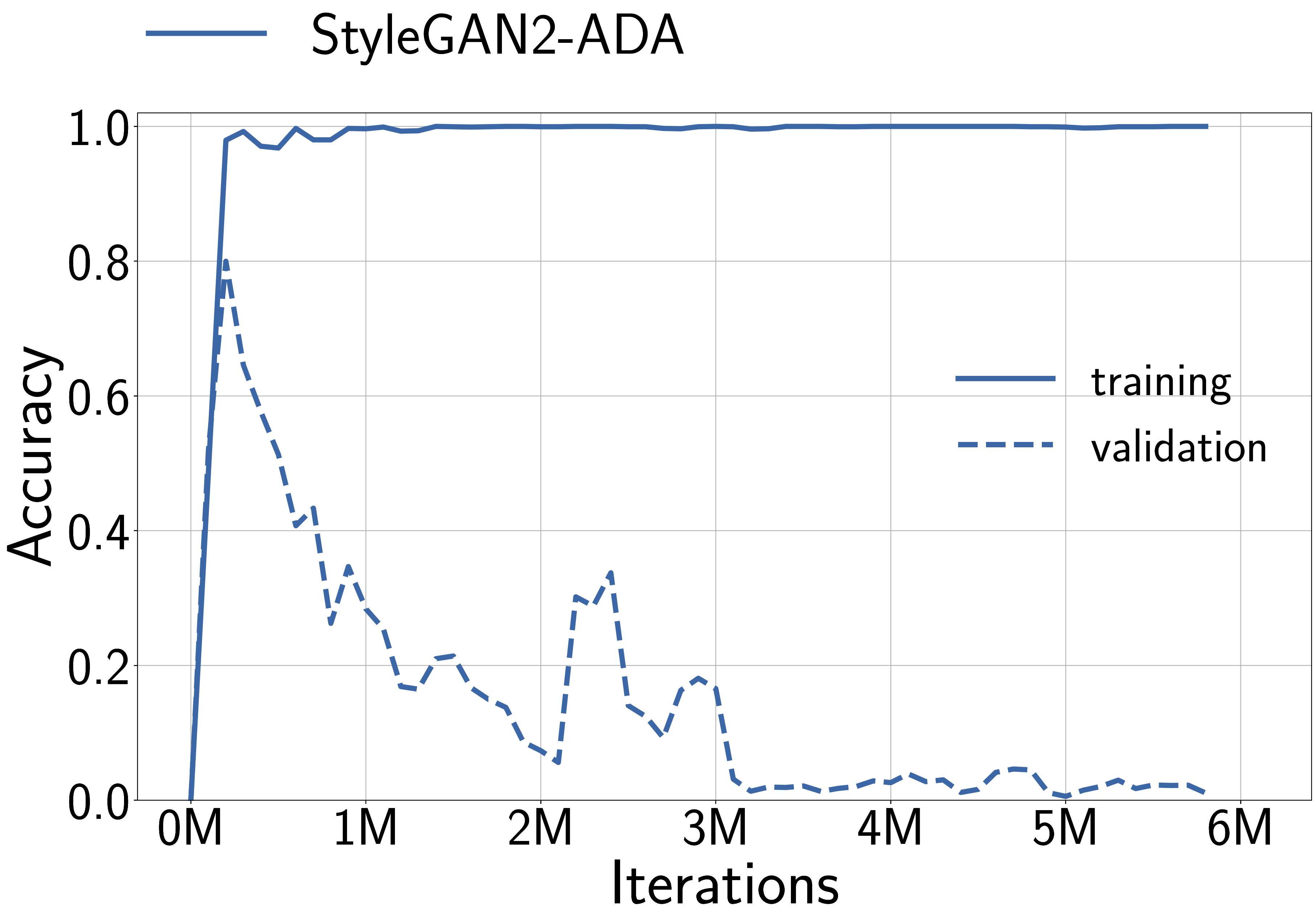
Other metrics to consider:

$$\frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]} \qquad \mathbb{E}[D_{\text{train}}]$$

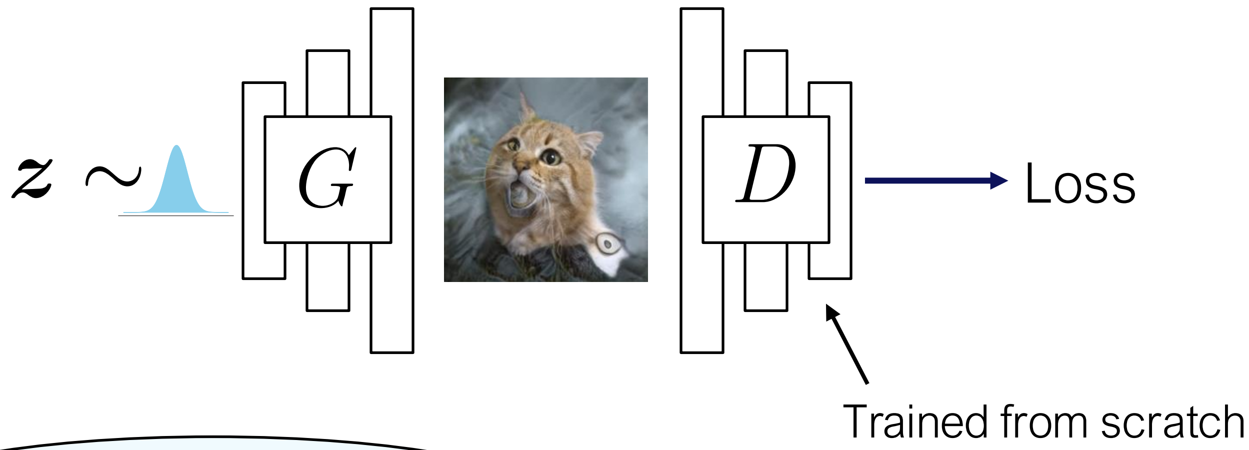
Training Generative Adversarial Networks with Limited Data [Karras et al., NeurIPS 2020]

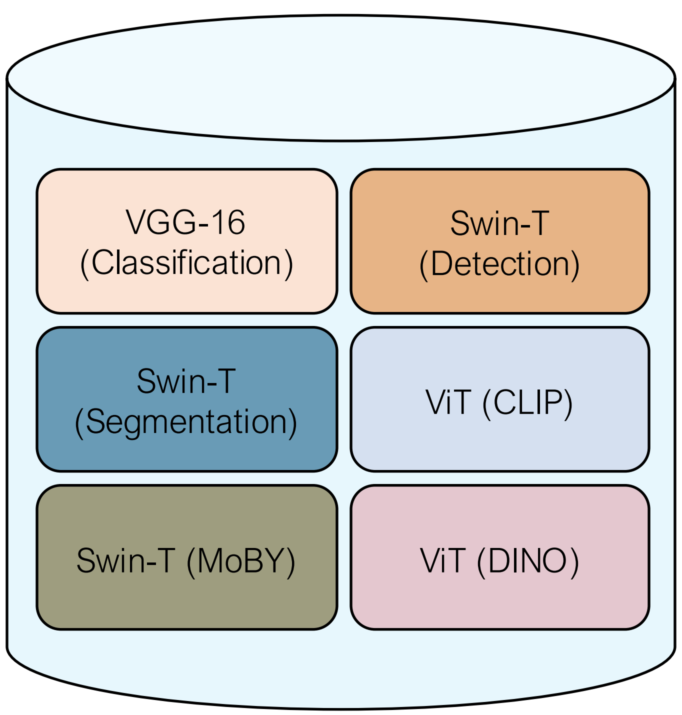
Training methods

Discriminator is still Overfitting



Standard GAN training



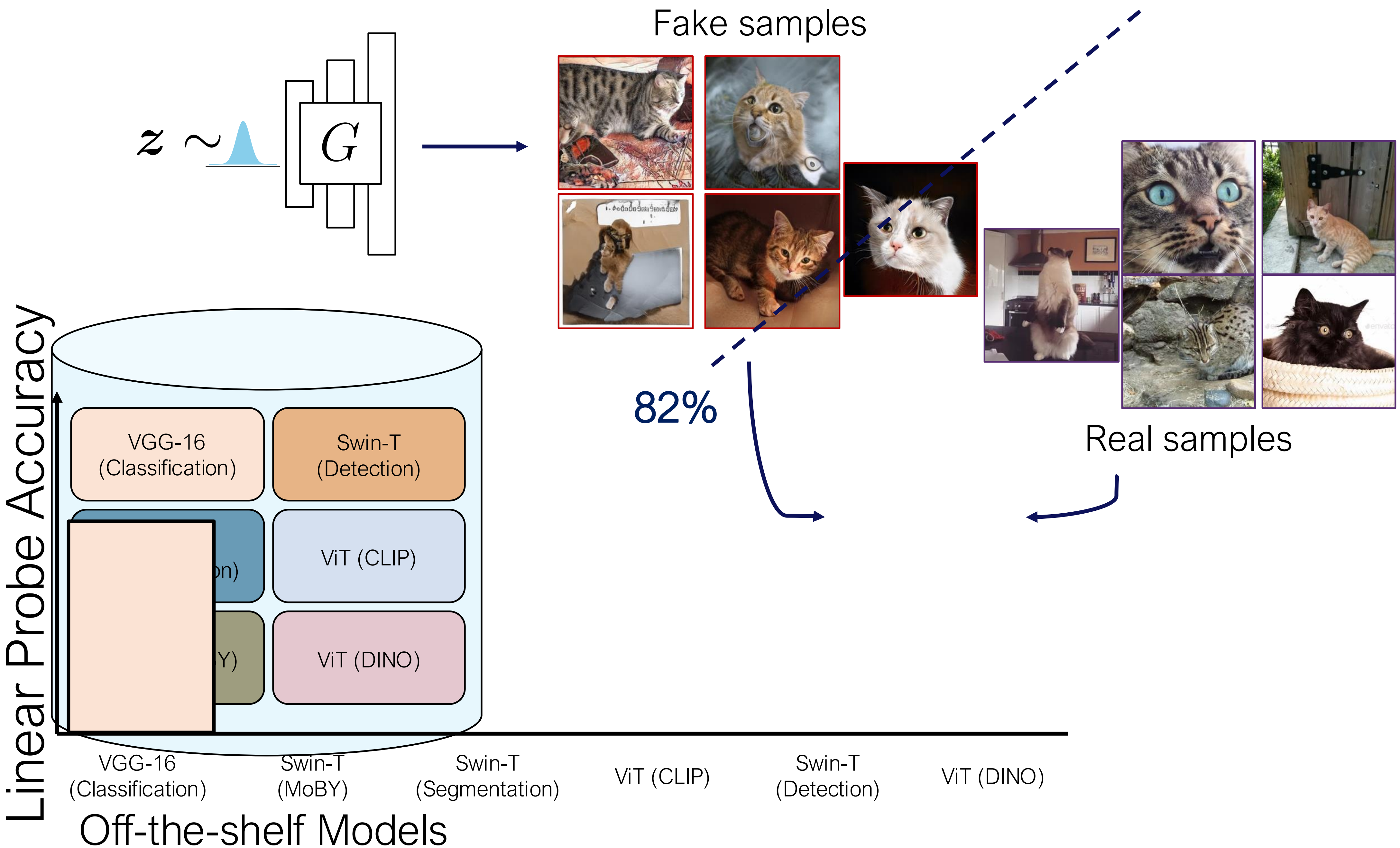


Off-the-shelf Models

Which pretrained models to use?

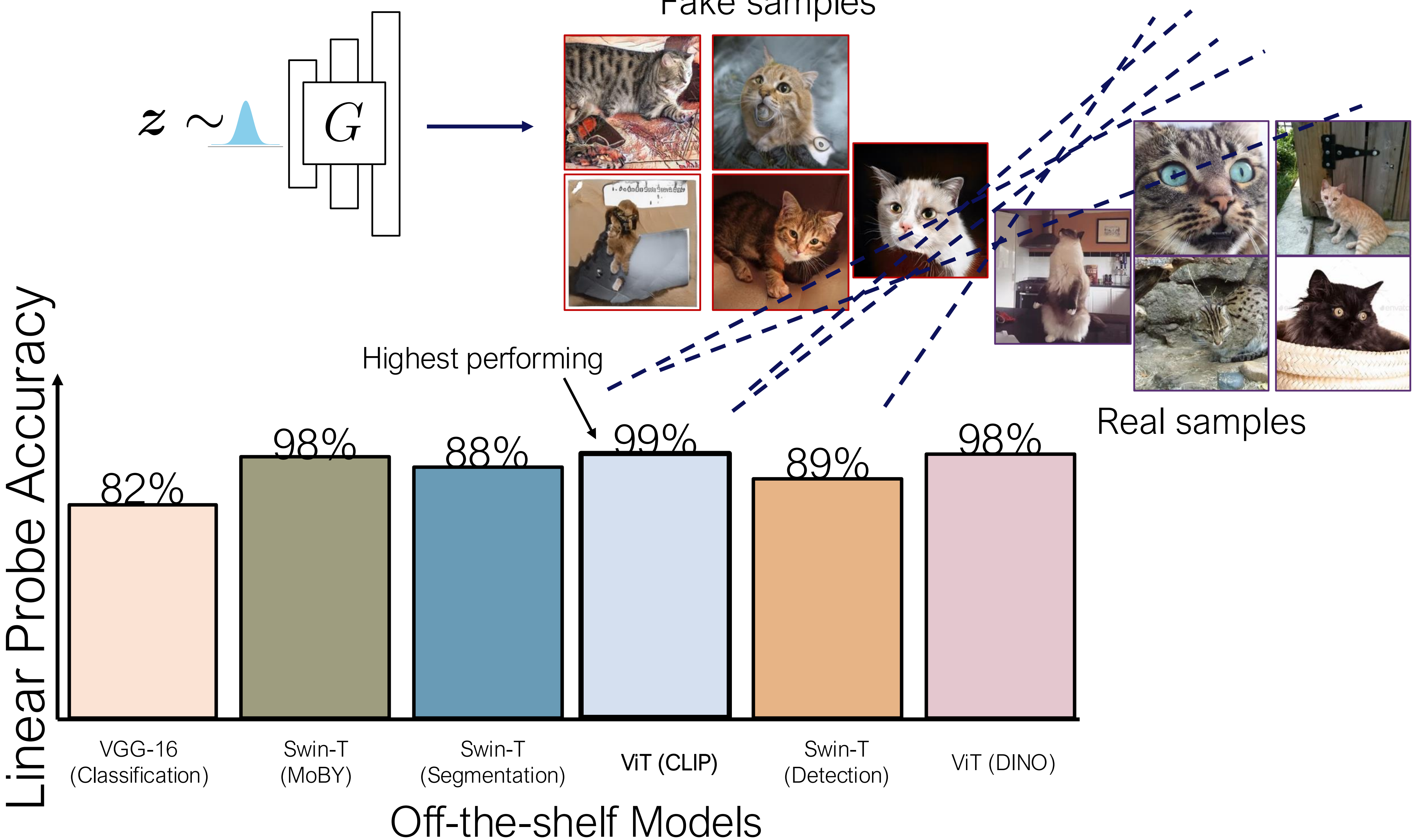
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Model Selection



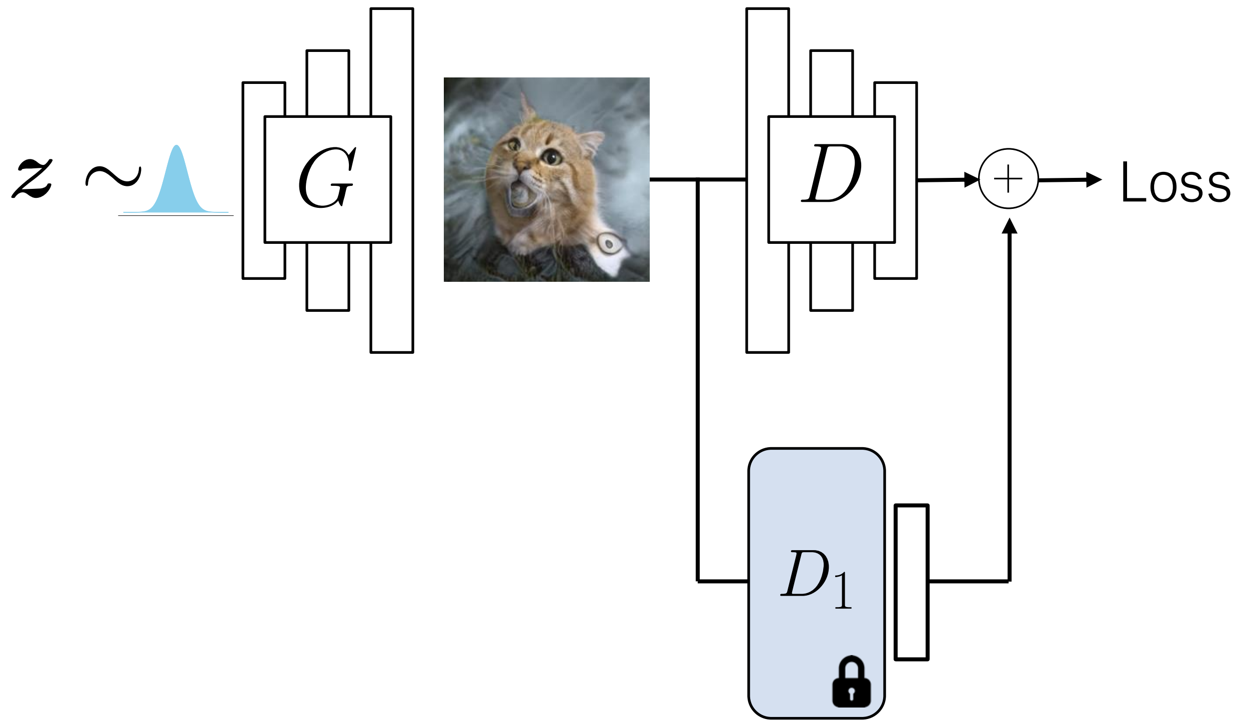
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Model Selection



Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Vision-aided GAN training

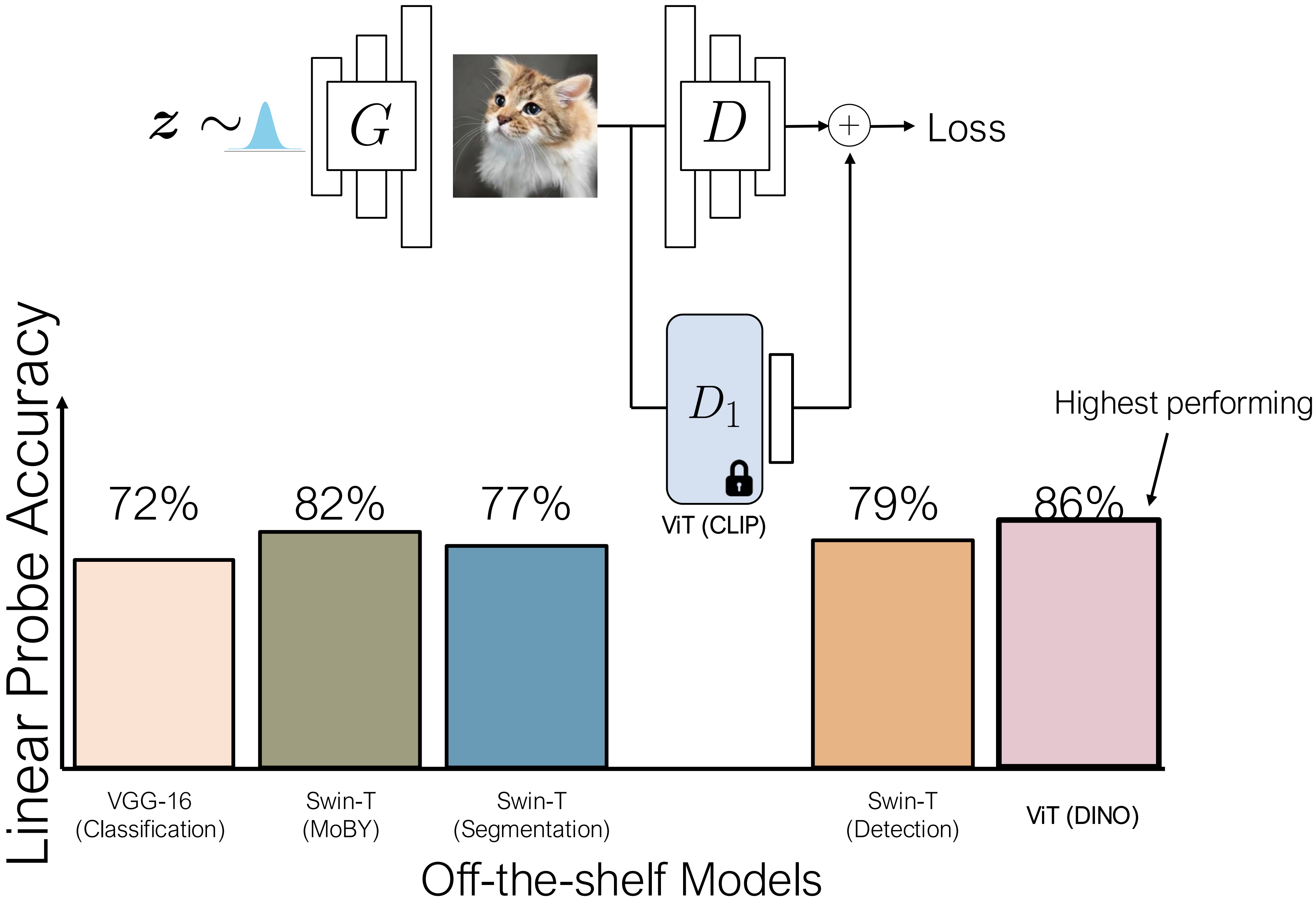


VGG-16 (Classification) | Swin-T (MoBY) | Swin-T (Segmentation) | **ViT (CLIP)** | Swin-T (Detection) | ViT (DINO)

Off-the-shelf Models

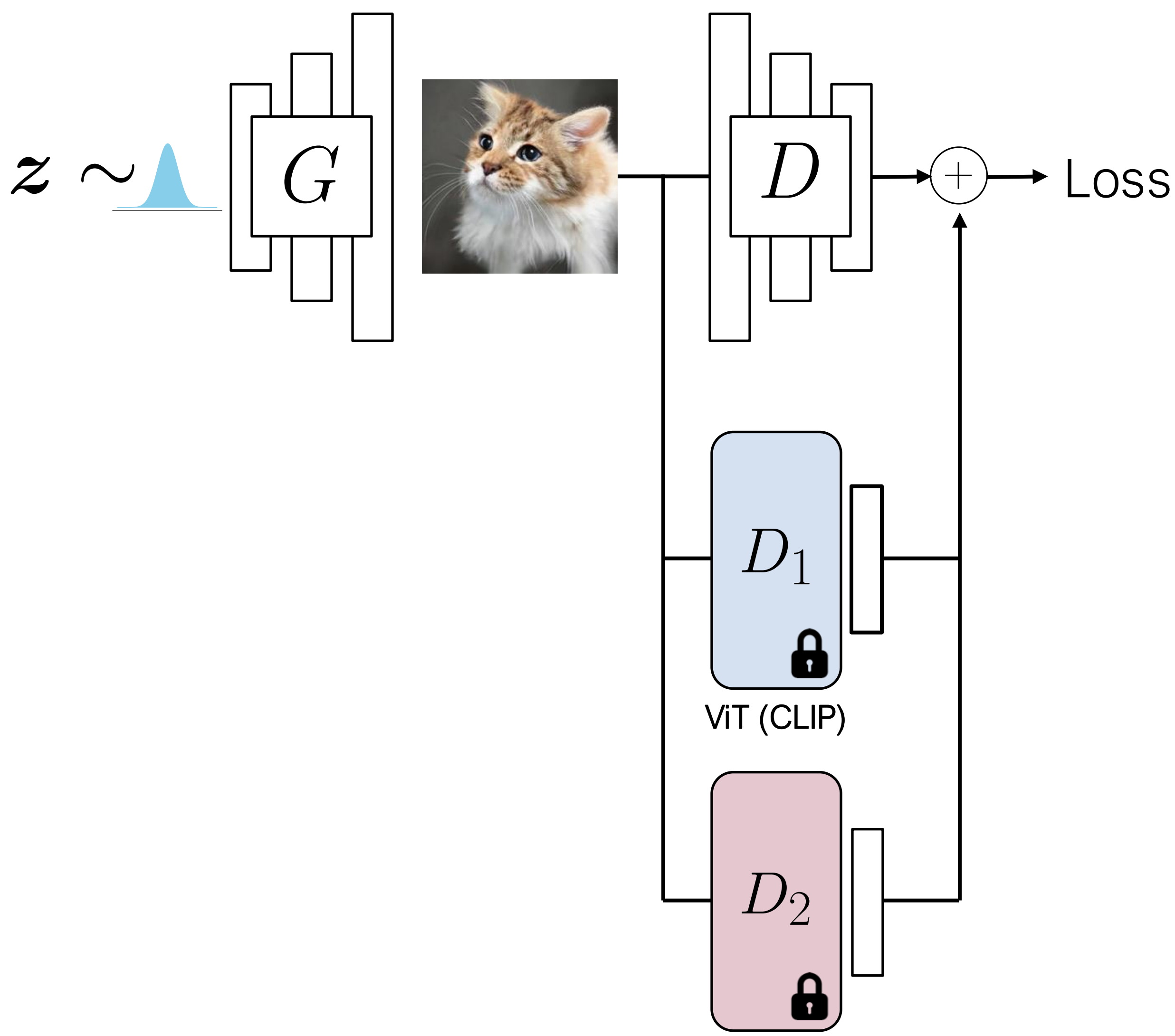
Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Add 2nd Vision-aided discriminator



Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

Add 2nd Vision-aided discriminator



VGG-16 (Classification)

Swin-T (MoBY)

Swin-T (Segmentation)

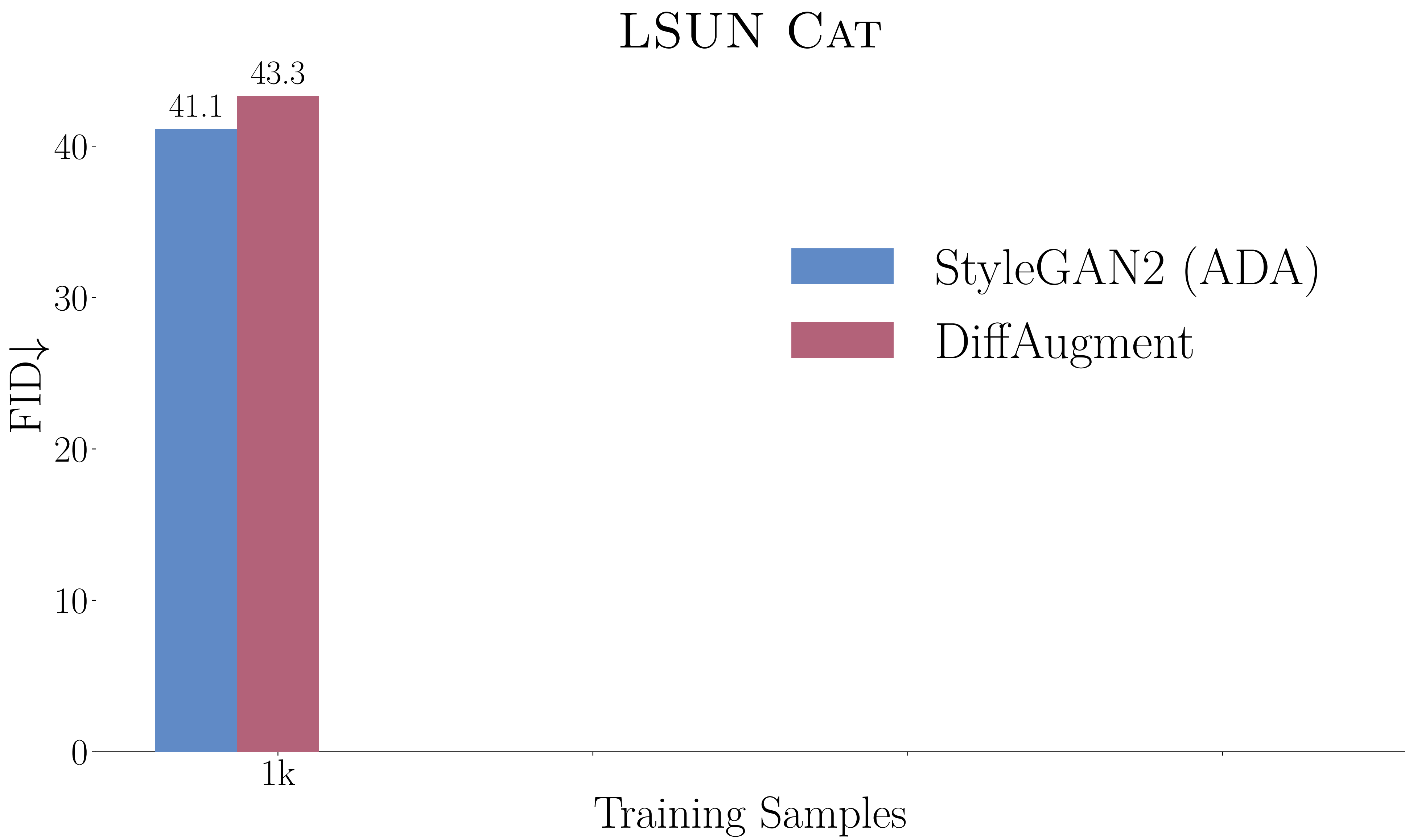
Swin-T (Detection)

ViT (DINO)

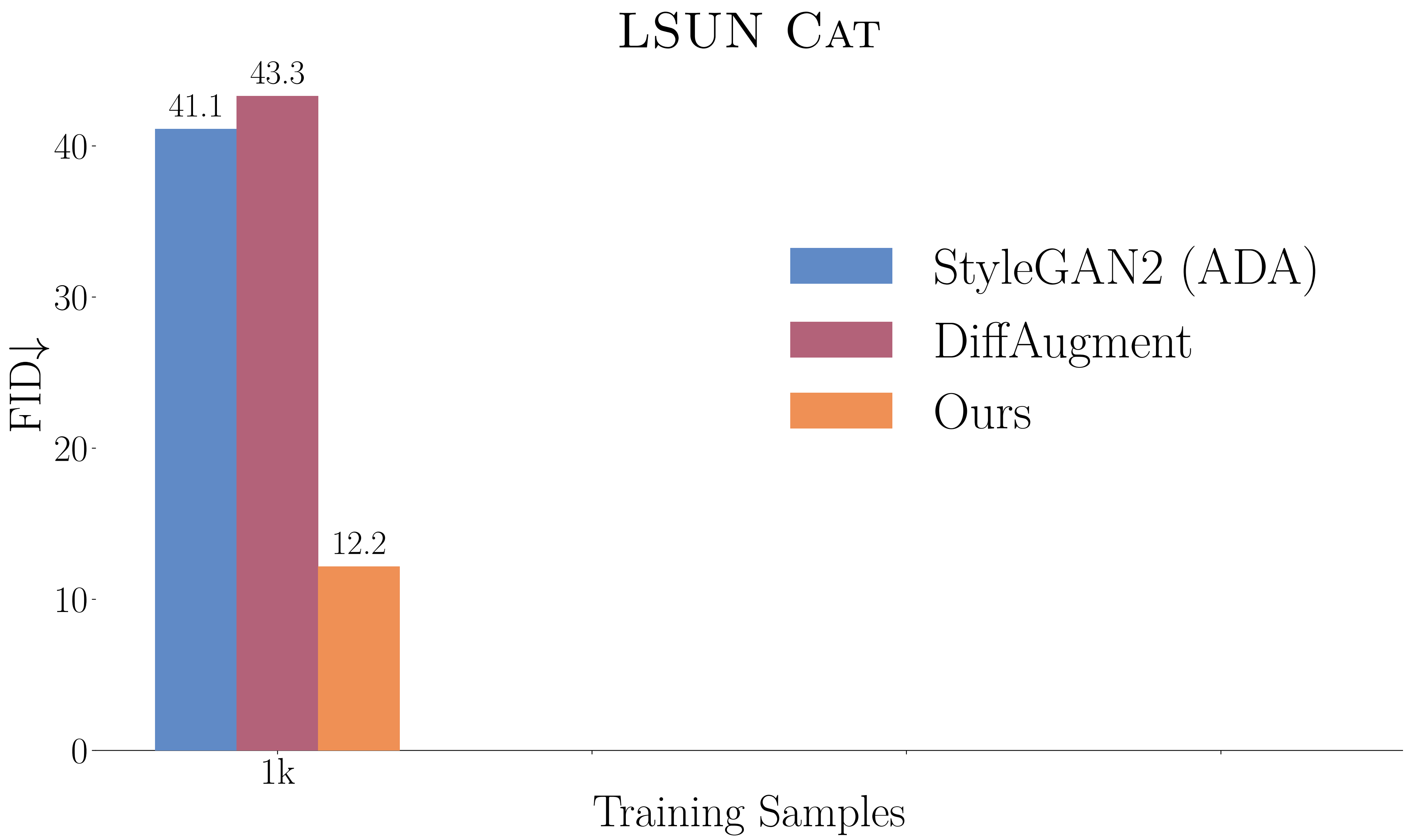
Off-the-shelf Models

Ensembling Off-the-shelf Models for GAN Training. Kumari et al., arXiv 2021

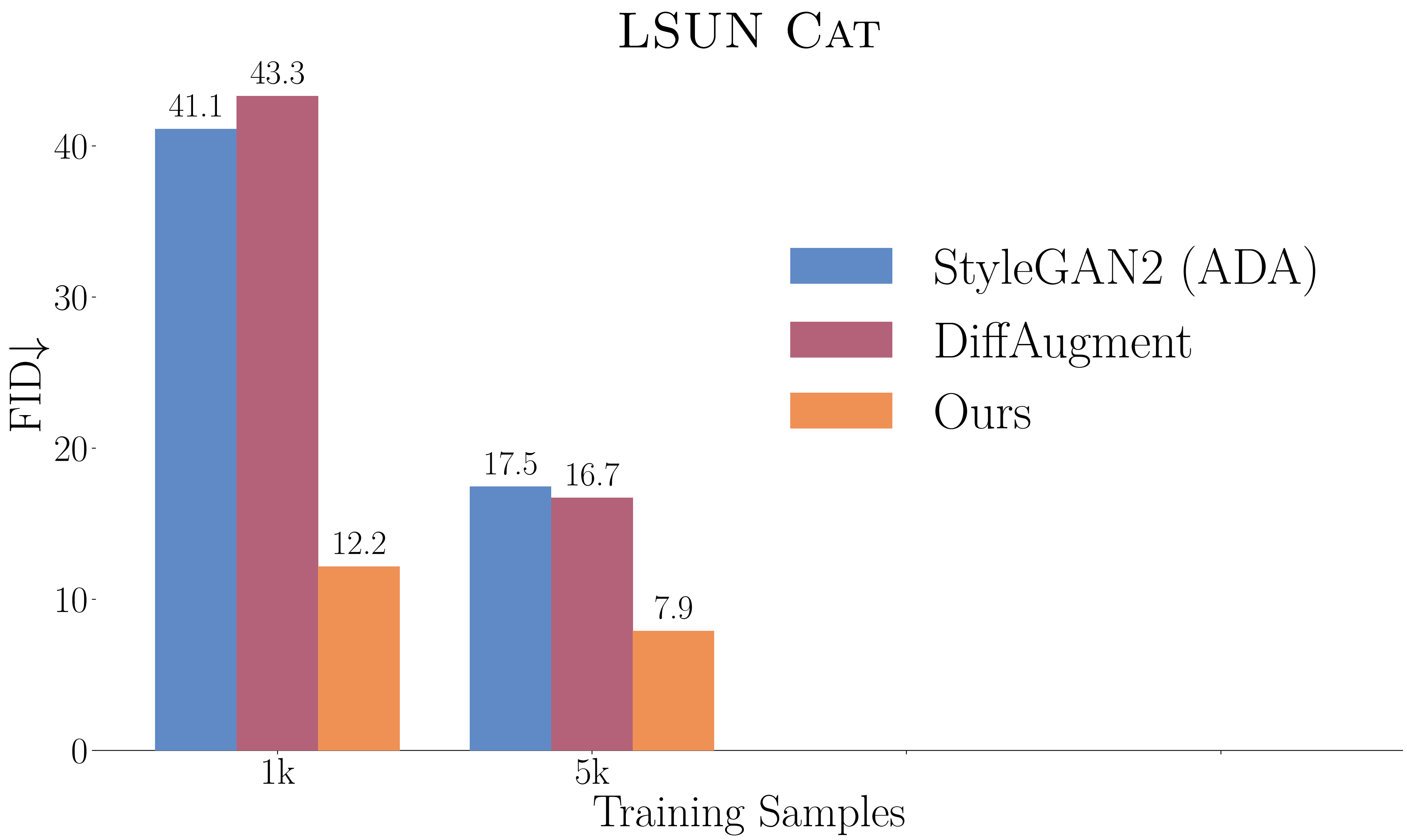
Benefit with varying training samples



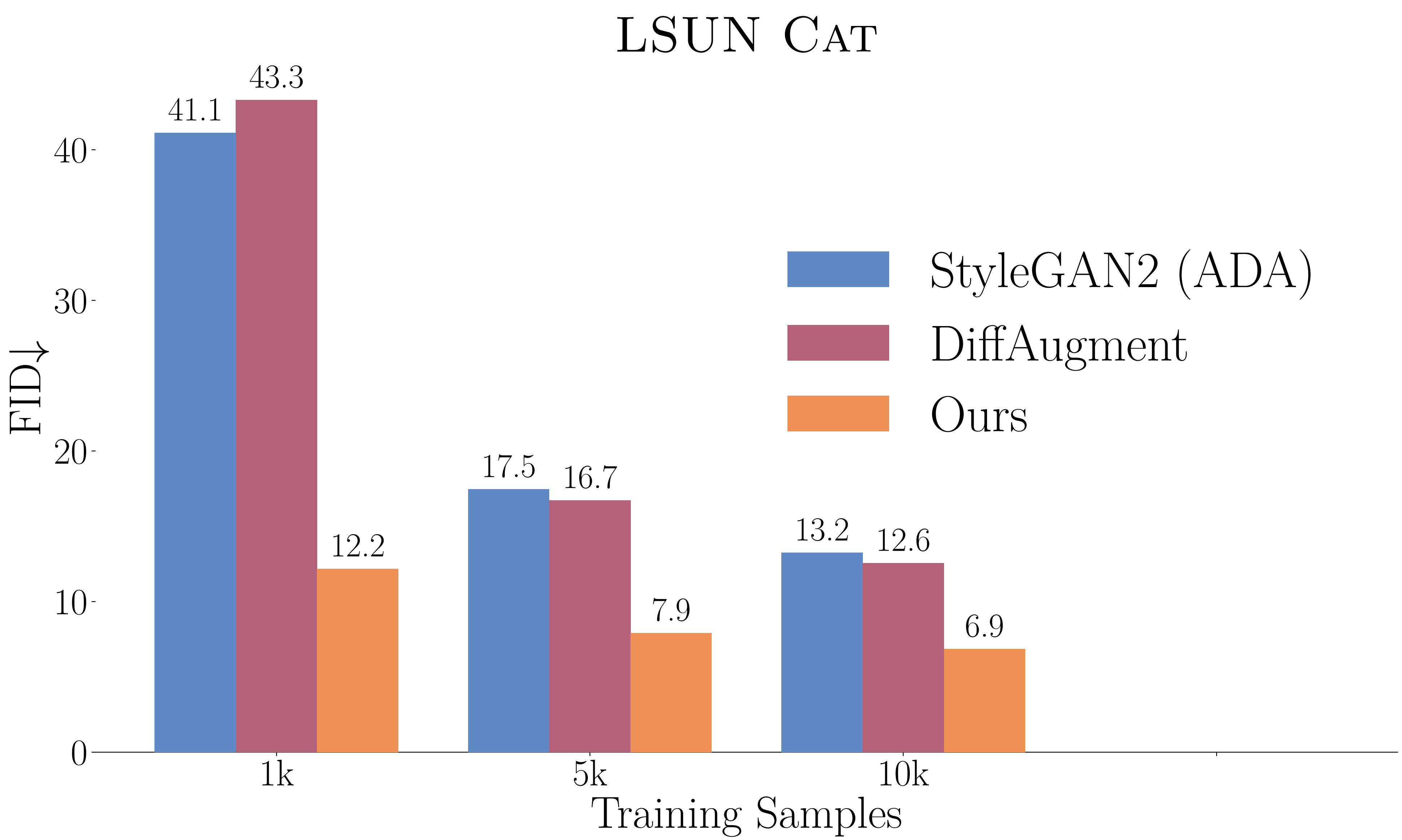
Benefit with varying training samples



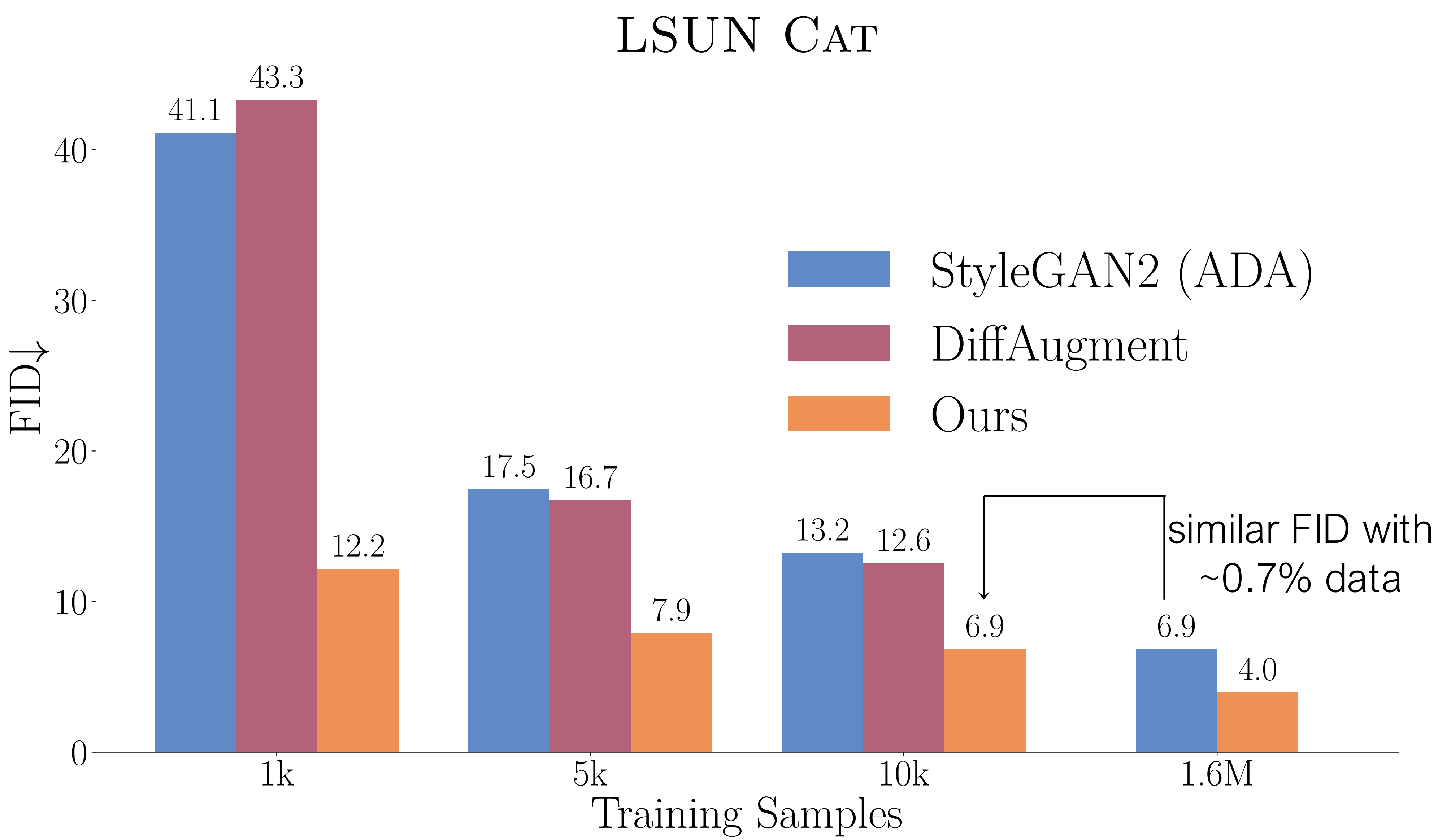
Benefit with varying training samples



Benefit with varying training samples



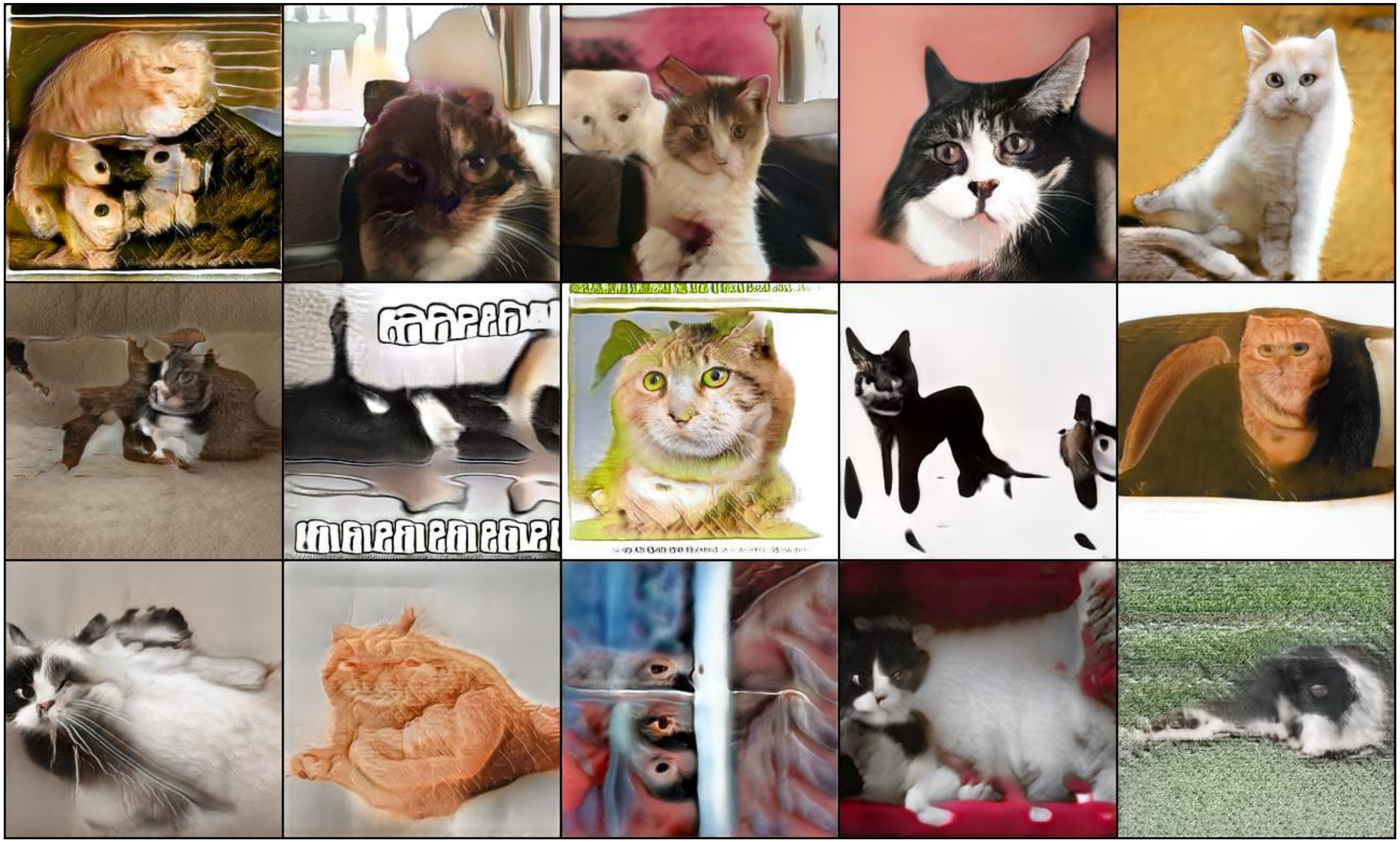
Benefit with varying training samples



Improved Samples

StyleGAN2-ADA

LSUN CAT 1k

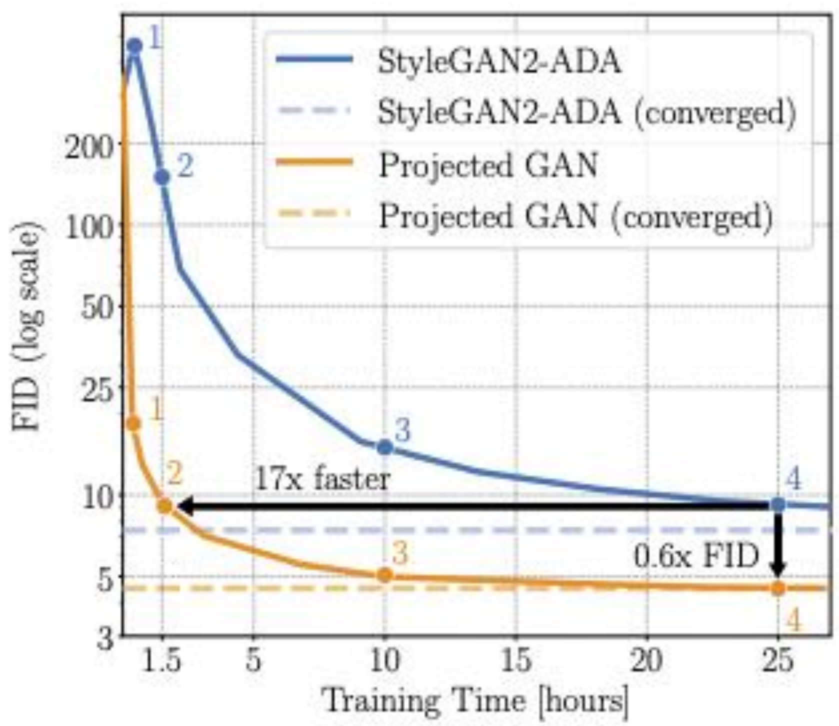


Improved Samples

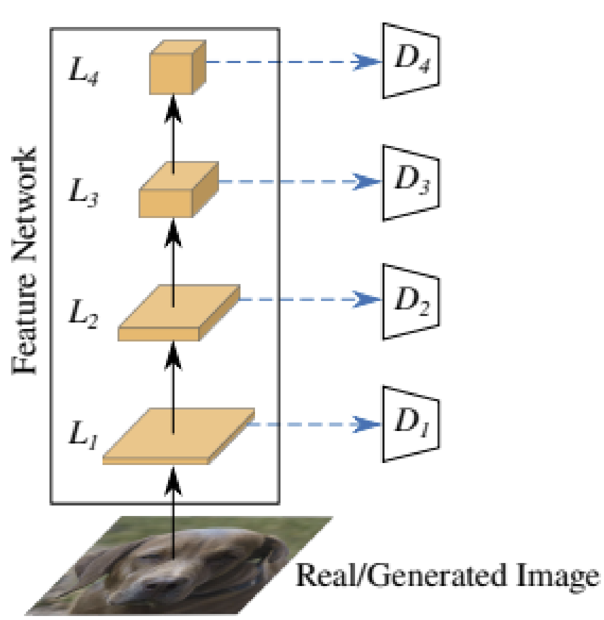
Ours

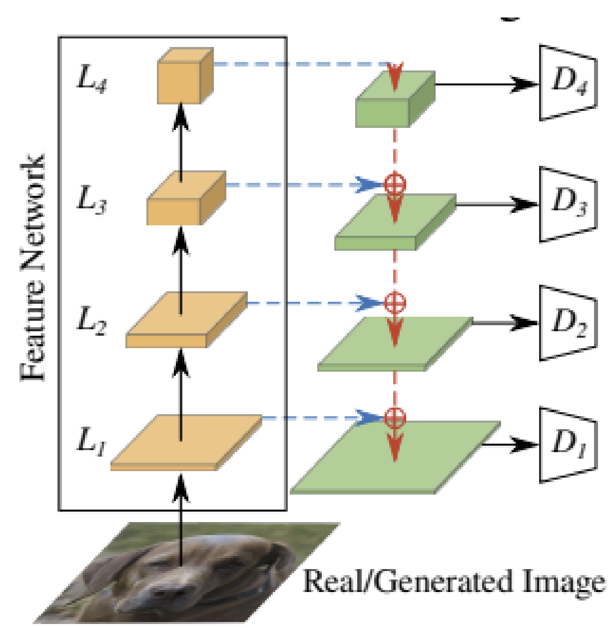
LSUN CAT 1k

Faster Convergence with Projected GANs



Dashed blue arrows :
1x1 conv
with random weights





Dashed red arrows:
3x3 conv
with random weights

use a FastGAN generator (https://arxiv.org/abs/2101.04775)

Projected GANs Converge Faster. Sauer et al., NeurIPS 2021

Combining Perceptual Loss and GAN Loss

Idea 1: add them together (many papers did that. It works)

Idea 2: Pre-trained features + trainable MLP layers
= Perceptual Discriminator

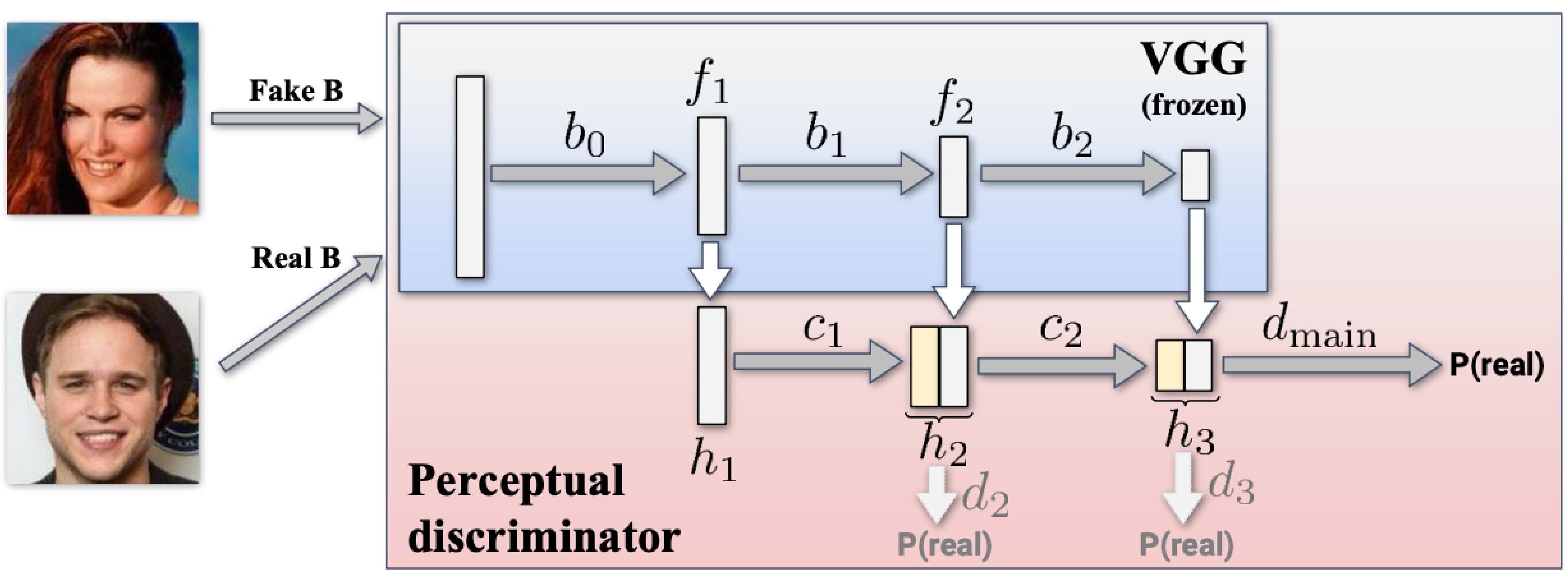


Image Manipulation with Perceptual Discriminators [Sungatullina et al. ECCV 2018]
Using multiple pre-trained models: Vision-aided GANs [Kumari et al., 2021]
Using random projection head: Projected GANs [Sauer et al., NeurIPS 2021]
Conditional discriminator: Enhancing photorealism enhancement [Richter et al., 2020]

What has driven GAN progress?

Ian Goodfellow @goodfellow_ian · Jan 14

4.5 years of **GAN progress** on face generation. arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434 arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196 arxiv.org/abs/1812.04948

What has driven GAN progress?

- Loss functions:

  cross-entropy, least square, Wasserstein loss, gradient penalty, Hinge loss, …
- Network architectures (G/D)

  Conv layers, Transposed Conv layers, modulation layers (AdaIN, spectral norm)
  mapping networks, …
- Training methods

  1. coarse-to-fine progressive training
  2. using pre-trained classifiers (multiple classifiers, random projection)
- Data

  data alignment, data filtering, differentiable augmentation
- GPUs

  bigger GPUs = bigger batch size (stable training) + higher resolution

Thank You!