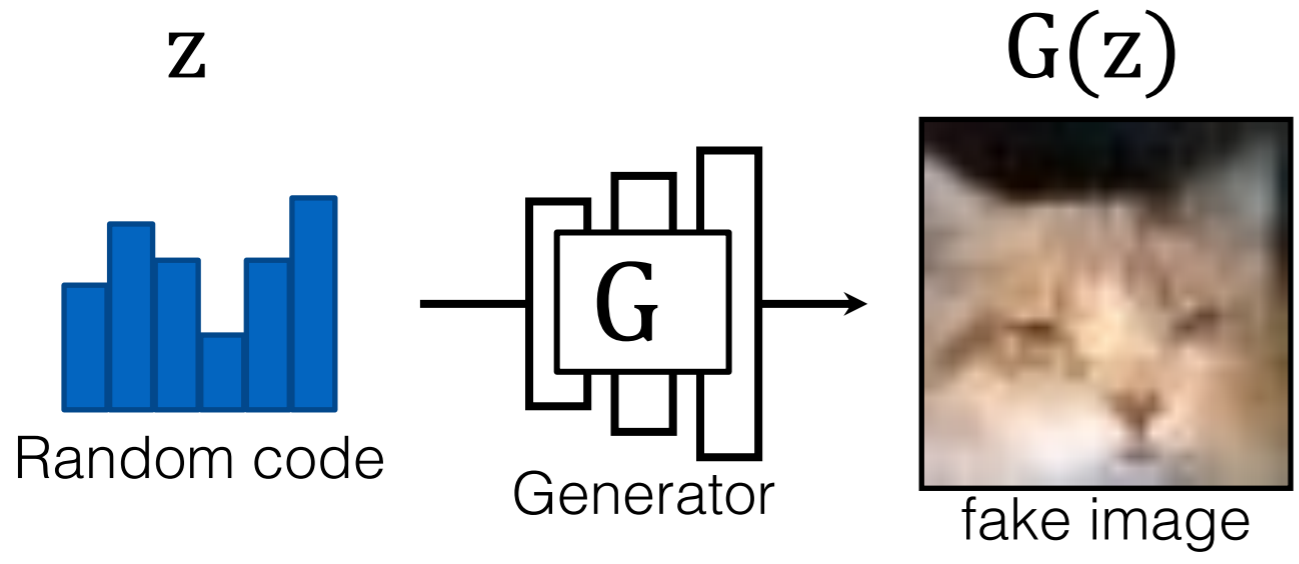


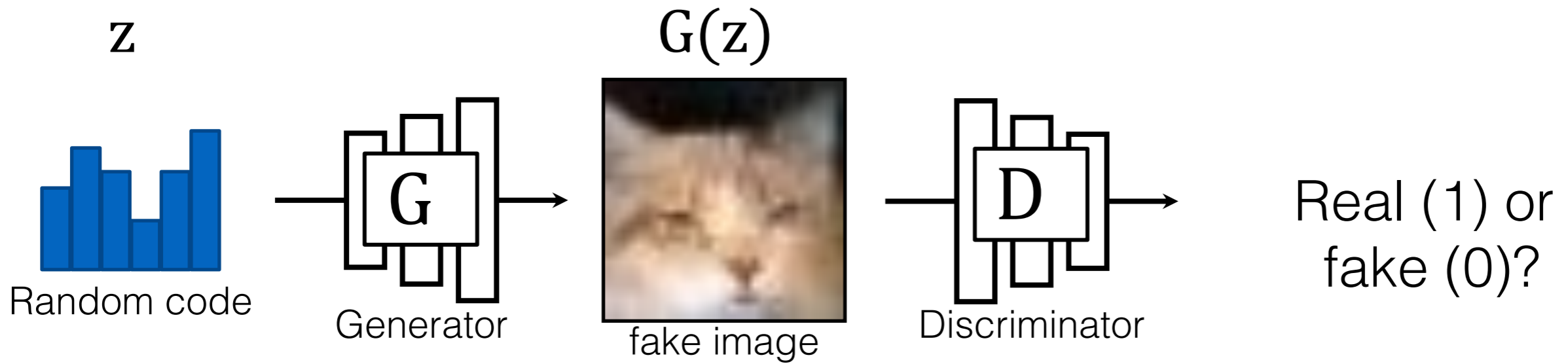


Generative Adversarial Networks (part 2)

Jun-Yan Zhu

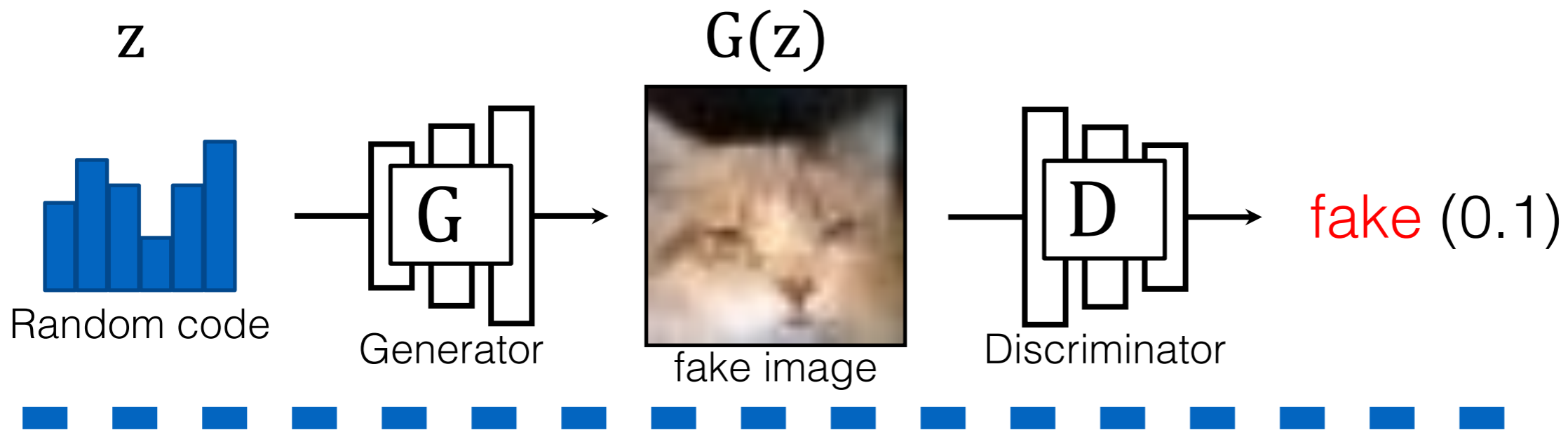
16-726 Learning-based Image Synthesis, Spring 2023





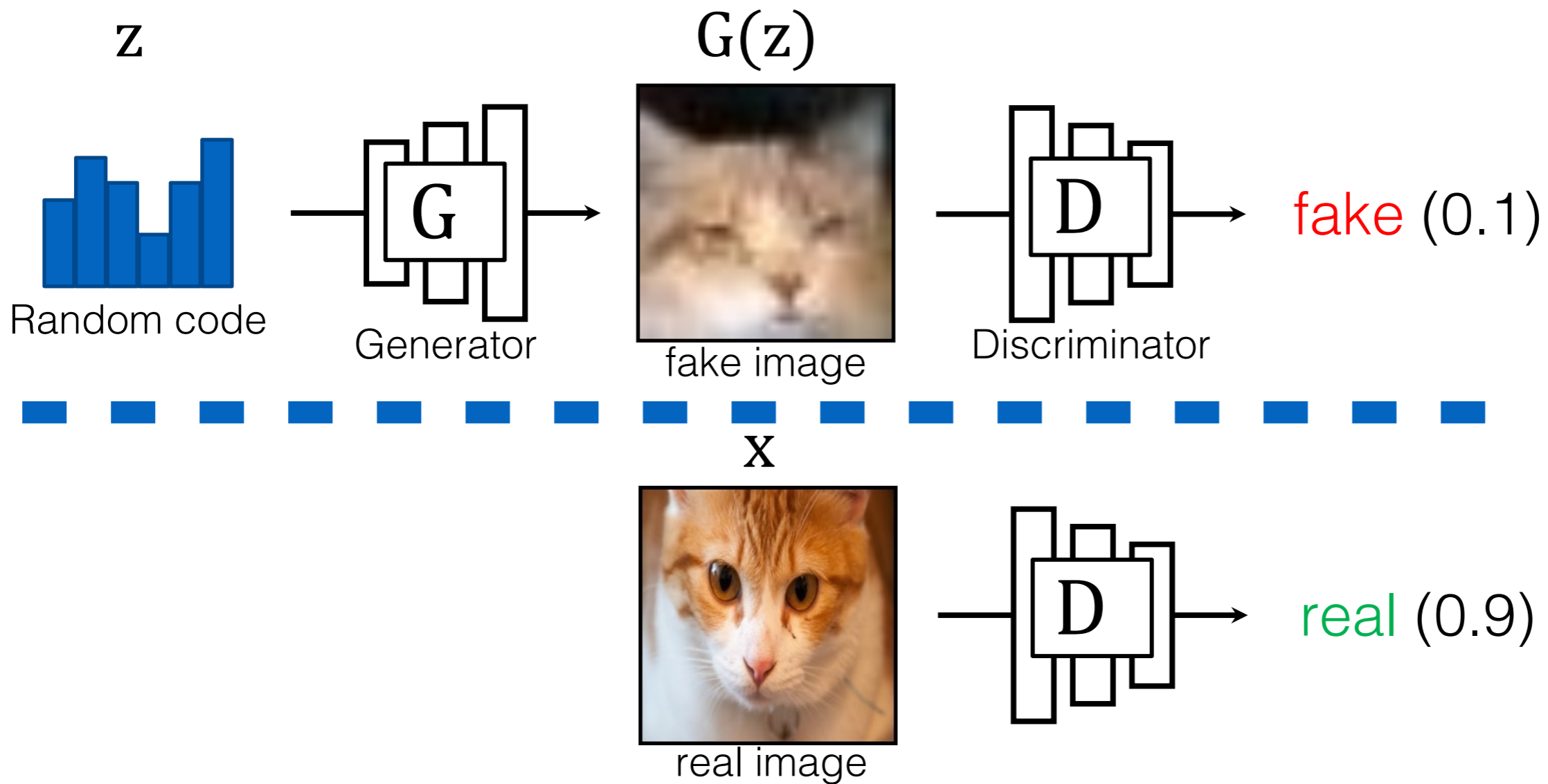
A two-player game:

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



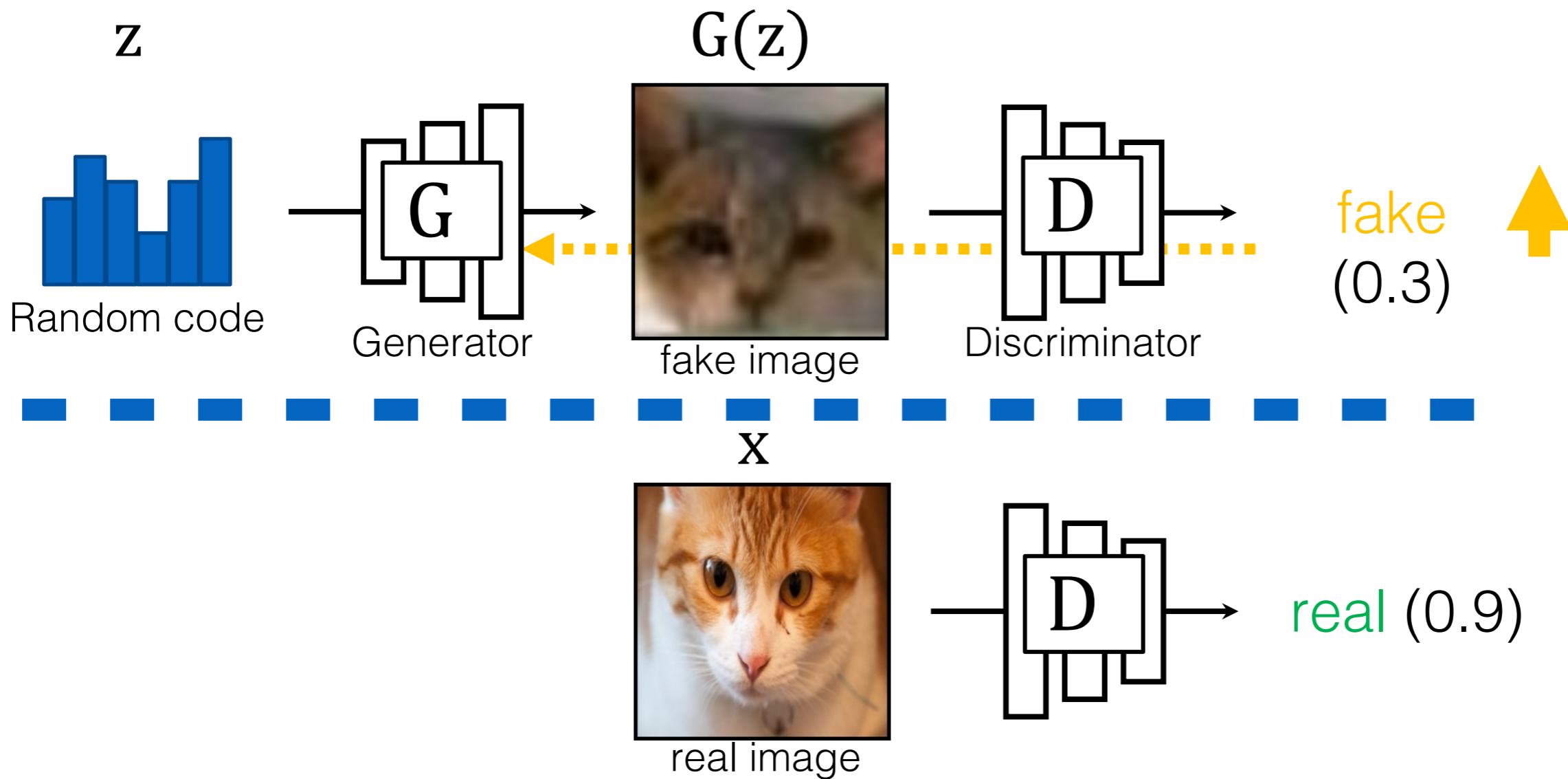
Learning objective (GANs)

$$\min_G \max_D \mathbb{E}_z [\log(1 - D(G(z)))]$$



Learning objective (GANs)

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



Learning objective (GANs)

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

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.

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

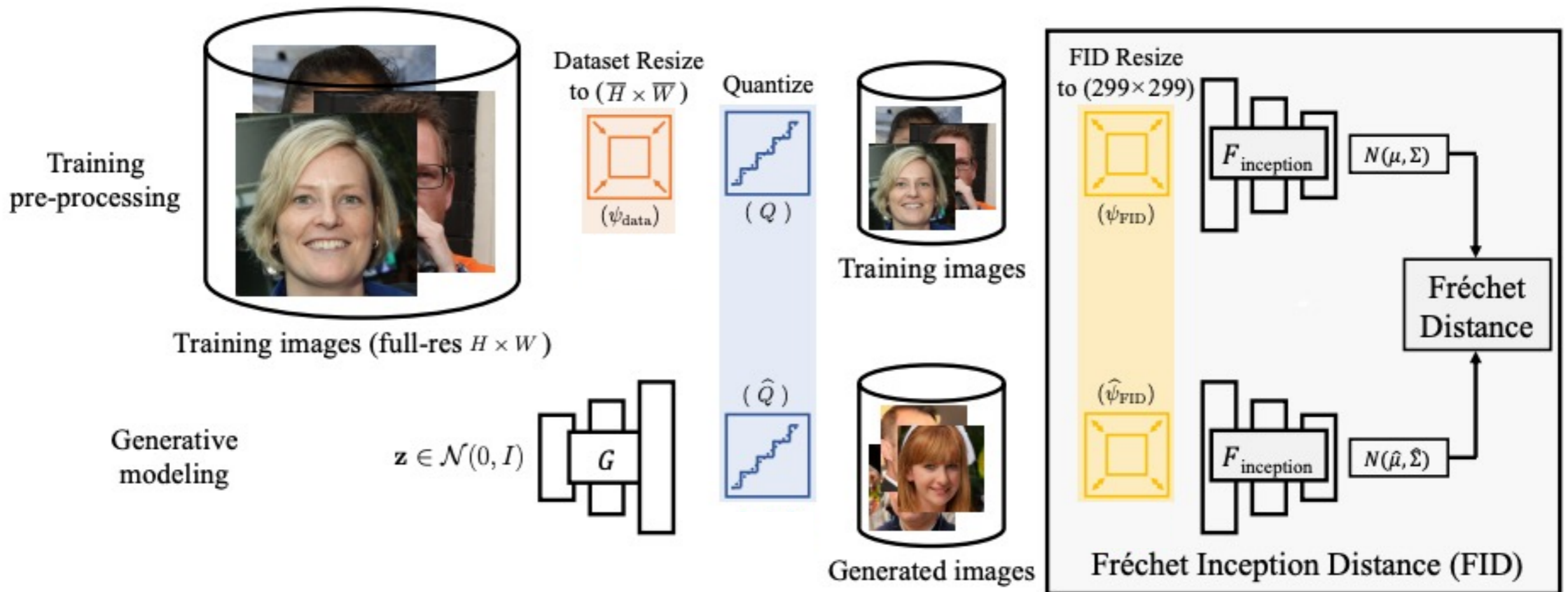


What has driven GAN progress?



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

GANs evaluation (FID)



Fréchet Inception Distance (FID)

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

GANs evaluation (FID)

Clean-fid libraries for evaluating generative models

```
Python 3.7.10 (default, Feb 26 2021, 18:47:35)
[GCC 7.3.0] :: Anaconda, Inc. on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

pip install clean-fid

Daily downloads (July, 2022): 100

Daily downloads (Feb, 2023) : 20, 000

Total downloads: 2, 600, 000

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)

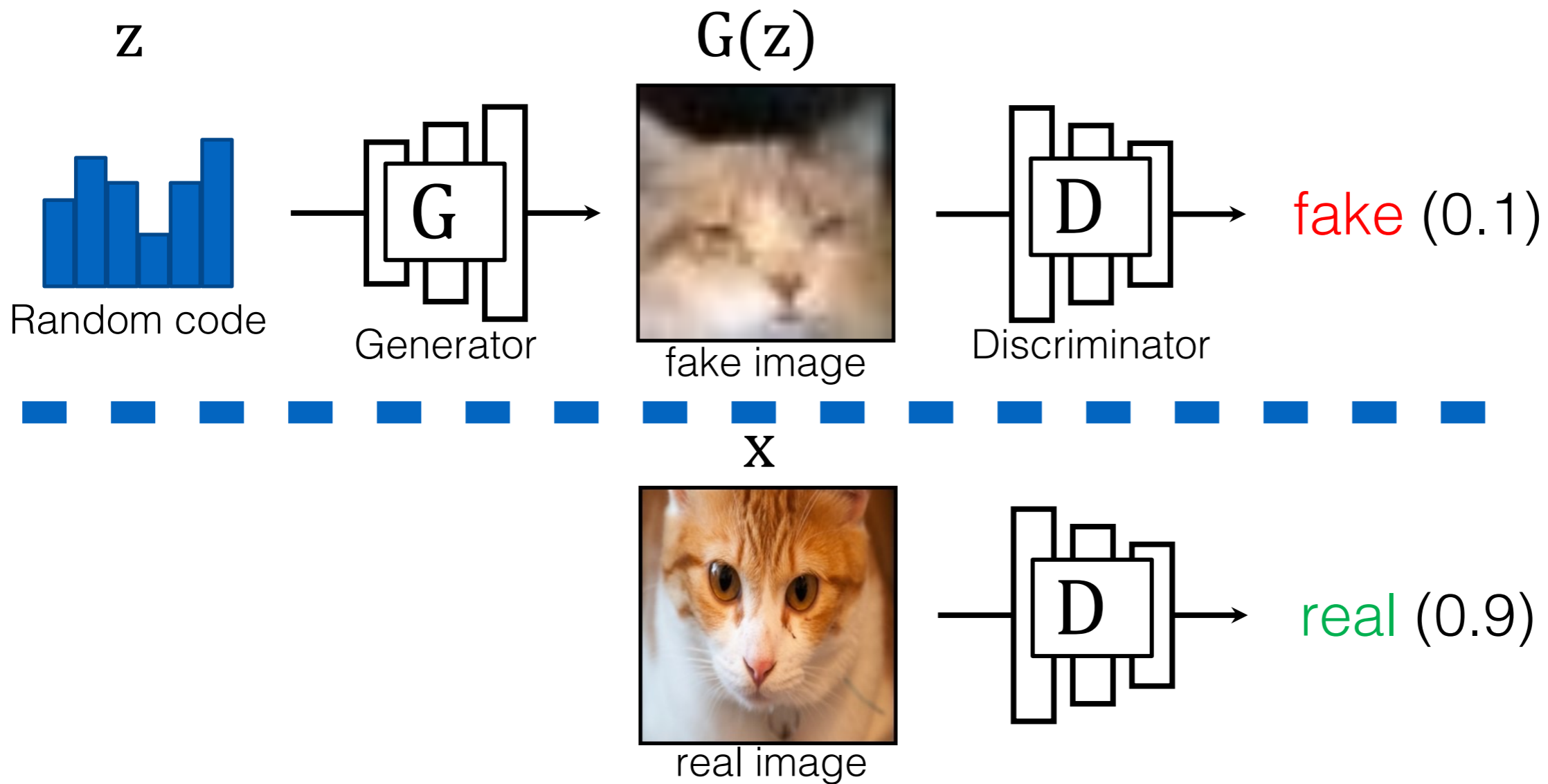
C. Training methods

D. Data

E. GPUs

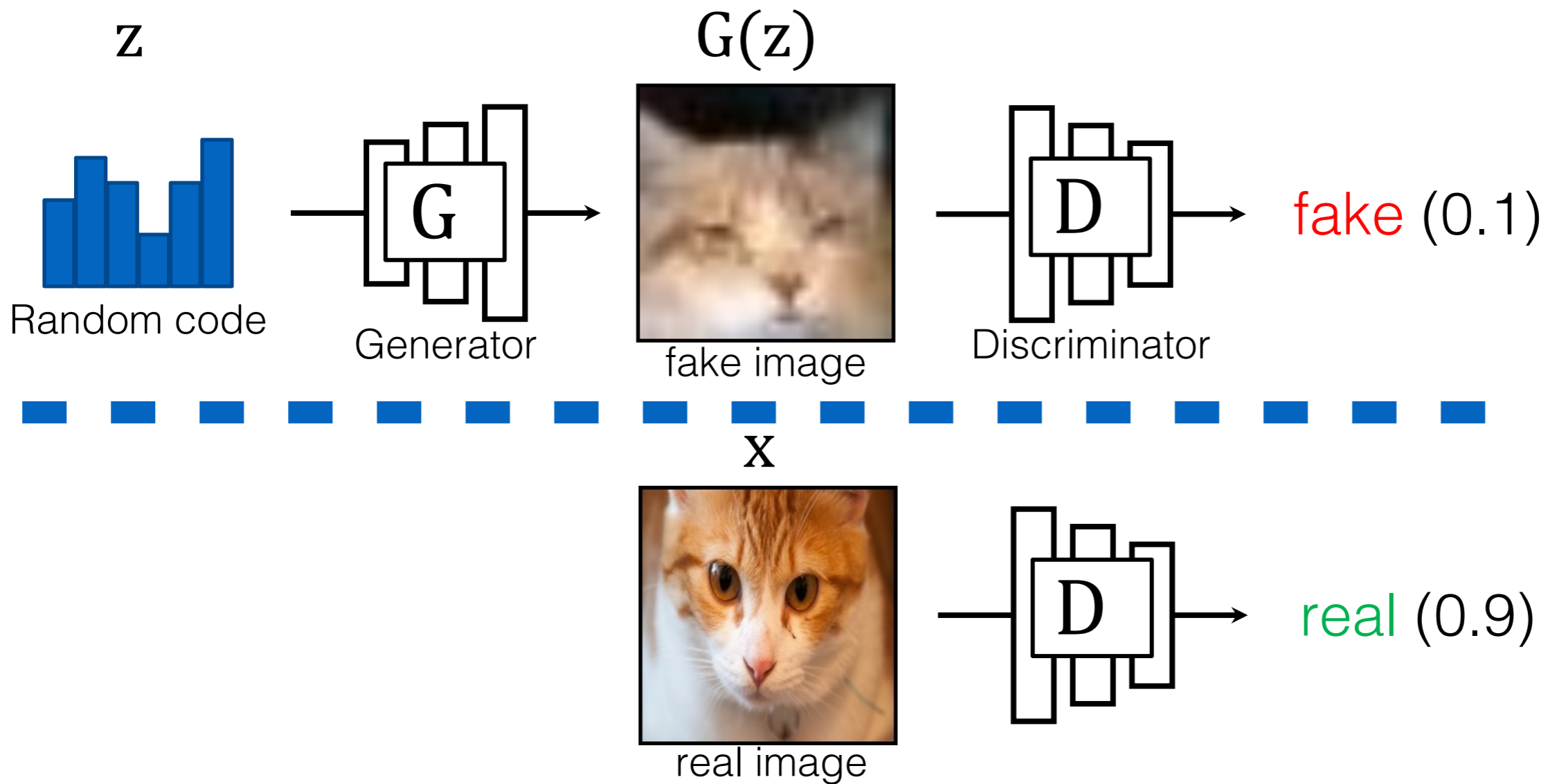
F. Funding

Loss functions



Learning objective (GANs)

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

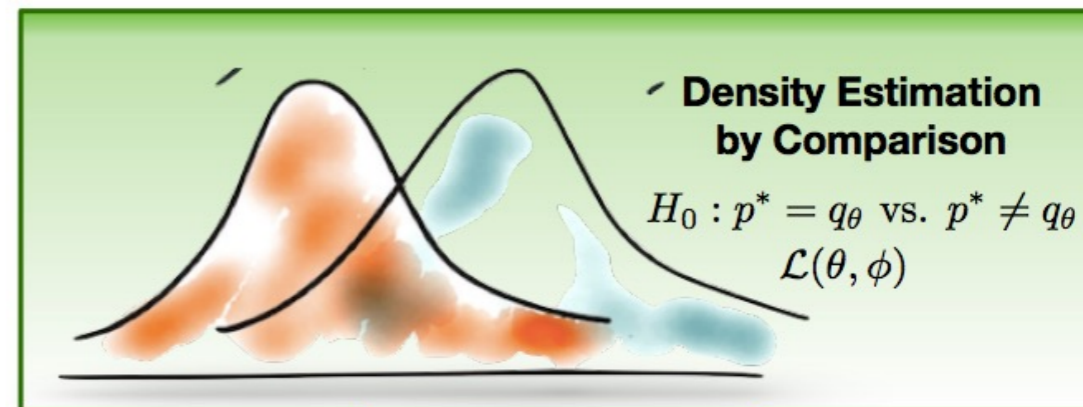


Learning objective (GANs variants)

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

EBGAN, WGAN, LSGAN, etc

Other divergences?



from [Mohamed & Lakshminarayanan 2017]

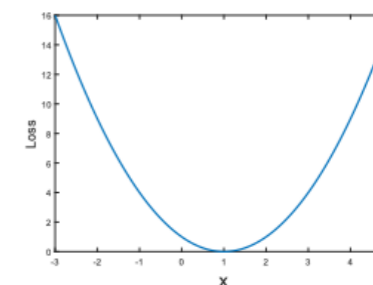
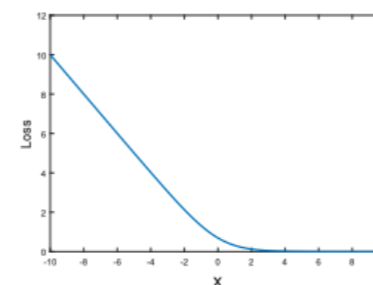
$$\min_G \max_{f_1, f_2} \mathbb{E}_z [f_1(G(z))] + \mathbb{E}_x [f_2(x)] \quad \begin{array}{l} \text{Convenient choice} \\ f_1 = -f \\ f_2 = f \end{array}$$

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

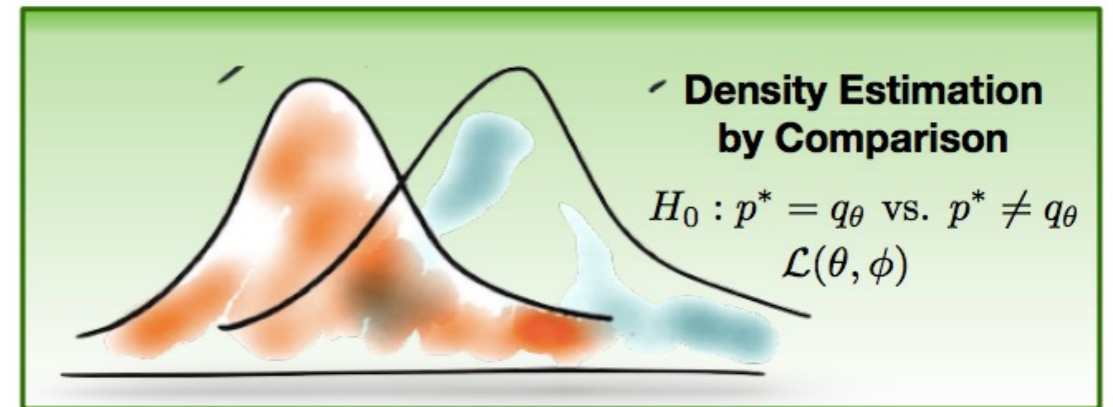
- Original GAN \rightarrow JSD
- Least-squares GAN \rightarrow Pearson chi-squared divergence

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

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



Other divergences?



from [Mohamed & Lakshminarayanan 2017]

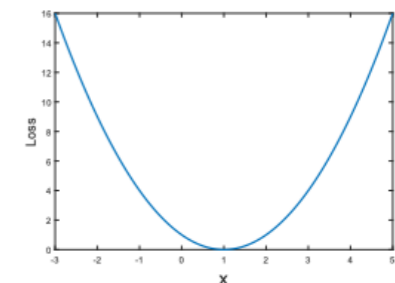
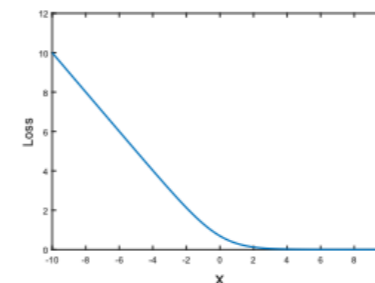
$$\min_G \max_{f_1, f_2} \mathbb{E}_z [f_1(G(z))] + \mathbb{E}_x [f_2(x)] \quad \begin{array}{l} \text{Convenient choice} \\ f_1 = -f \\ f_2 = f \end{array}$$

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

- Original GAN \rightarrow JSD
- Least-squares GAN \rightarrow Pearson chi-squared divergence

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

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



Other divergences?

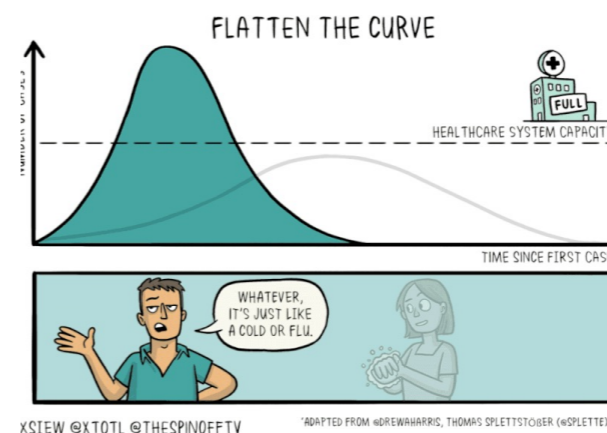
$$KL(p_{\text{data}} || p_{\theta}) \longleftarrow \mathbb{E}_{x \sim p_{\text{data}}} [\log p_{\theta}(x)]$$

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

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

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

Earth-Mover (EM) distance
/ Wasserstein distance



Wasserstein GAN

[Arjovsky, Chintala, Bottou 2017]

Lipschitz continuity

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

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

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

wGAN GP [Gulrajani et al., 2018]:

$$\arg \min_G \max_f \mathbb{E}_{\mathbf{z}, \mathbf{x}} [-f(G(\mathbf{z})) + f(\mathbf{x})] + \lambda \mathbb{E}_{\hat{\mathbf{x}} \sim P_{\hat{\mathbf{x}}}} [(\|\nabla_{\hat{\mathbf{x}}} f(\hat{\mathbf{x}})\|_2 - 1)^2]$$

Gradient penalty (GP)

Spectral Normalization

[Miyato, Kataoka, Koyama, Yoshida 2018]

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

- 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

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

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

Better objectives? optimizers?

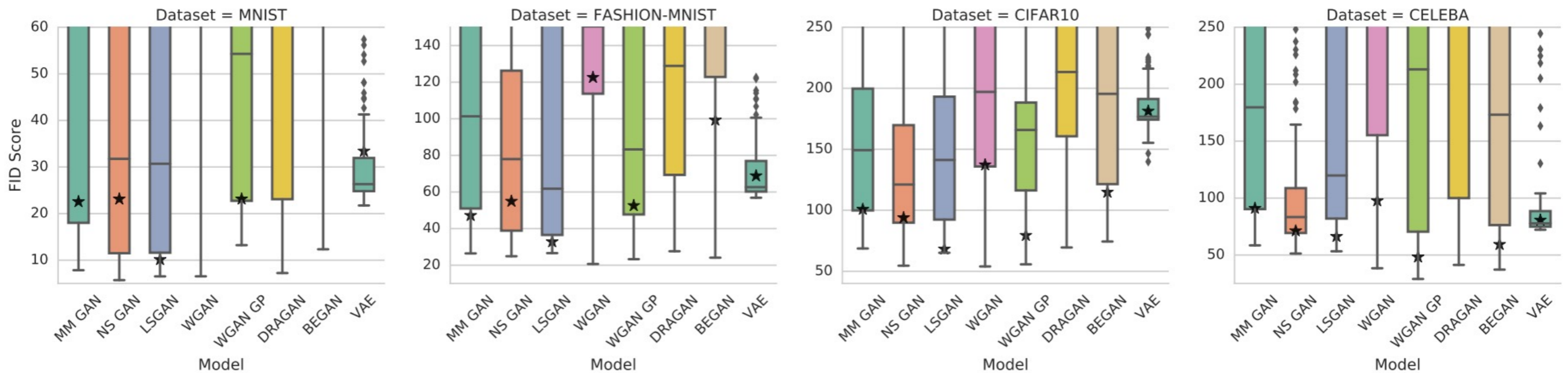


Figure 4: A *wide range* hyperparameter search (100 hyperparameter samples per model). Black stars indicate the performance of suggested hyperparameter settings. We observe that GAN training is extremely sensitive to hyperparameter settings and there is no model which is significantly more stable than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

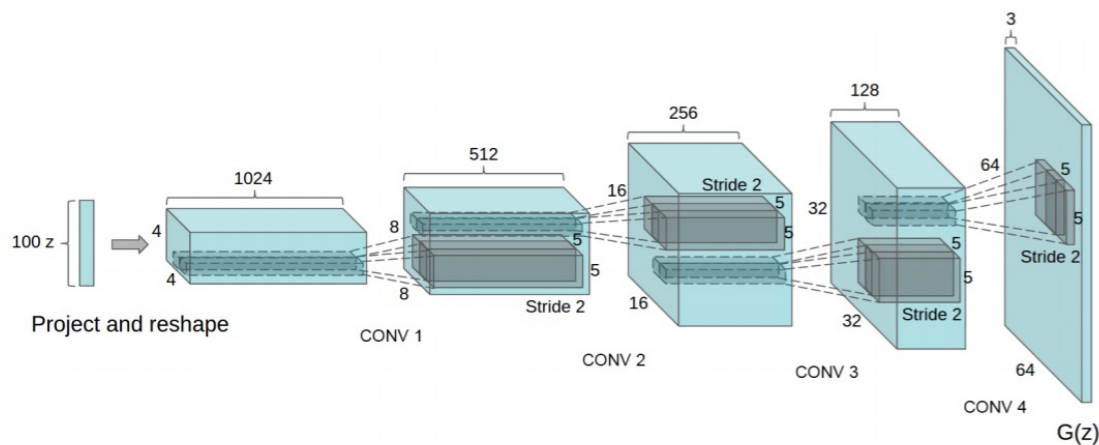
[Karras, Laine, Aila 2019]



Better Architectures!

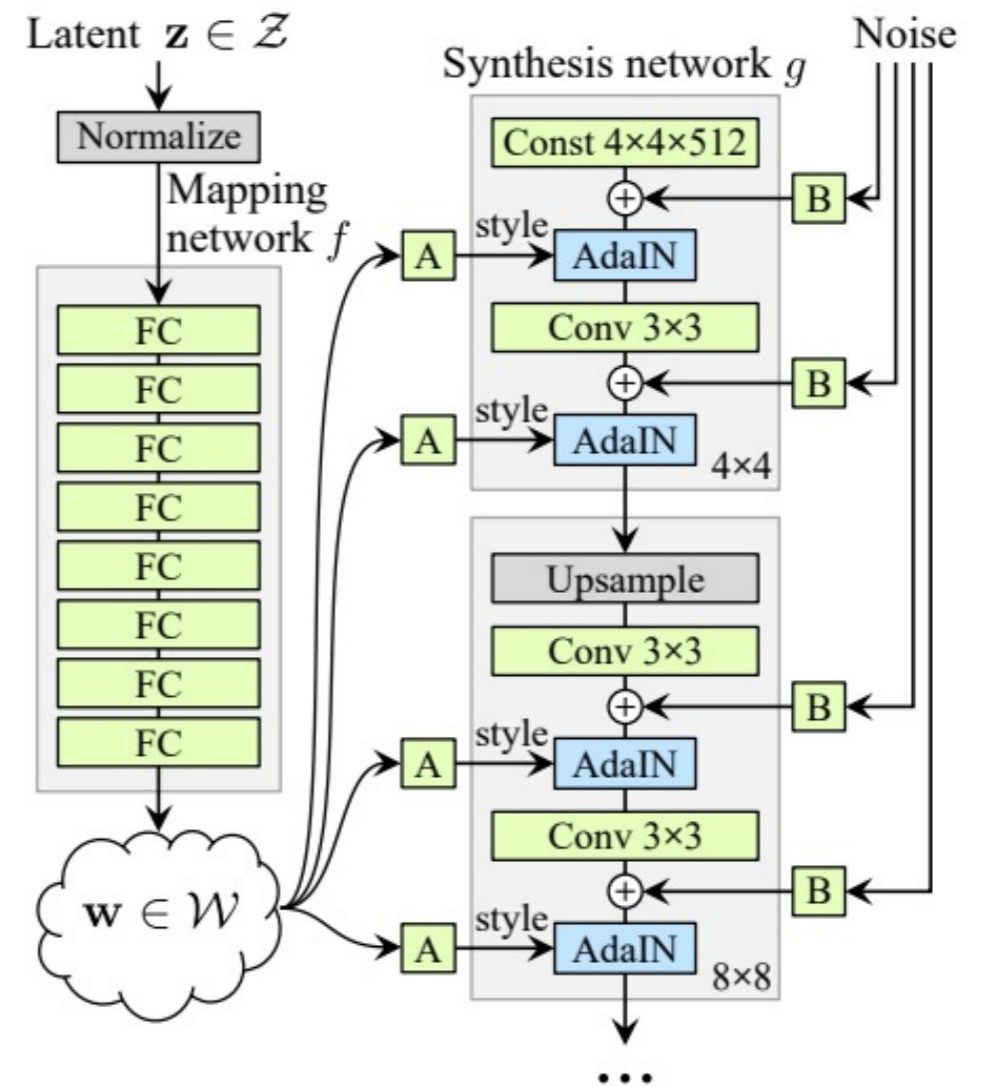
DCGAN

[Radford, Metz, Chintala 2016]



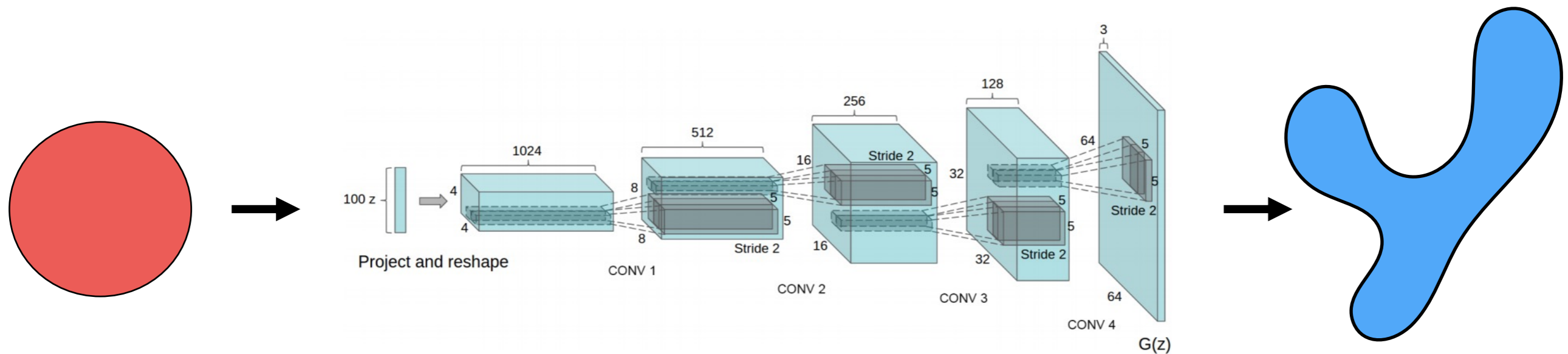
StyleGAN

[Karras, Laine, Aila 2019]



DCGAN

[Radford, Metz, Chintala 2015]

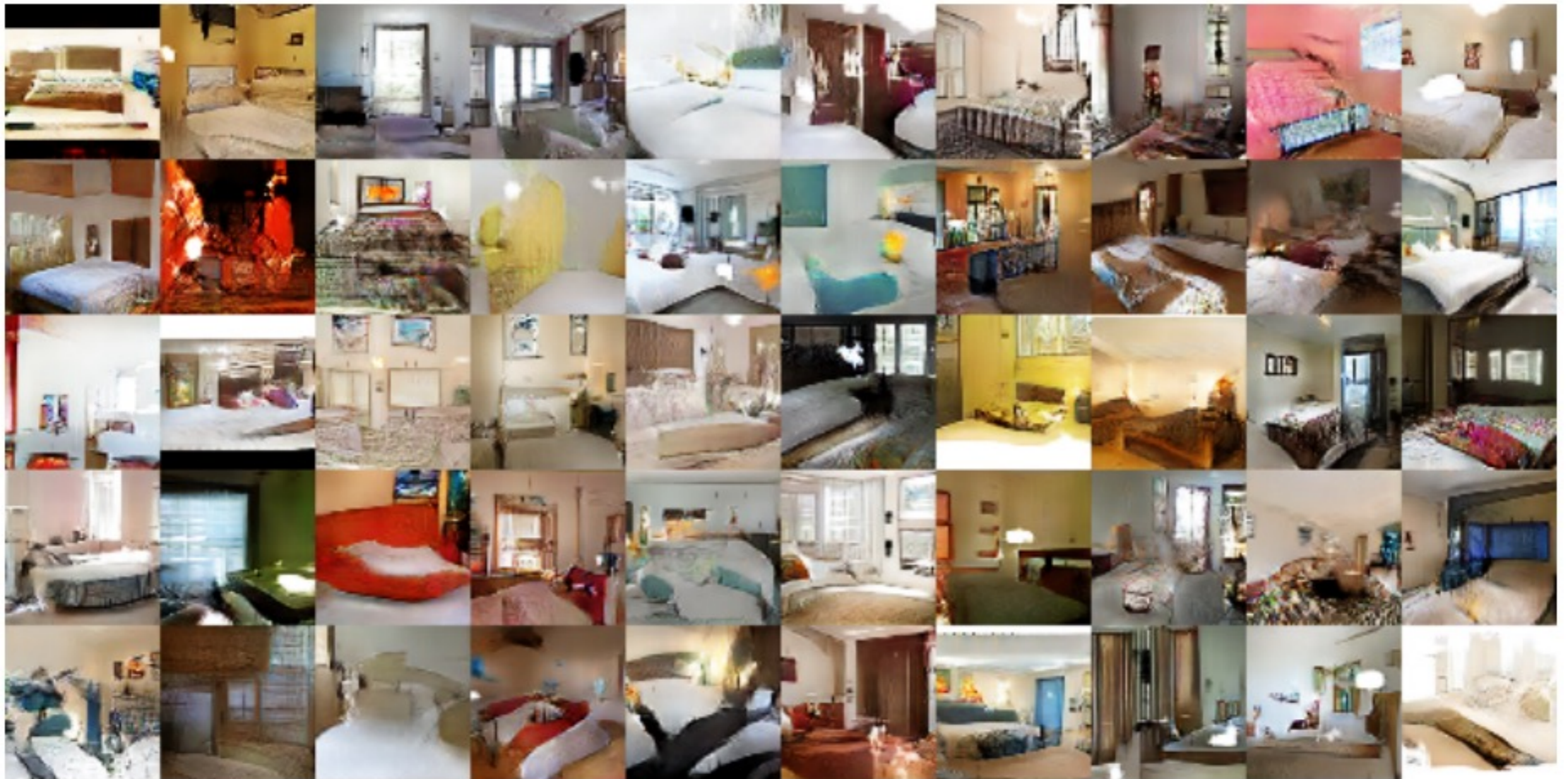


+ Convnet

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

DCGAN

[Radford, Metz, Chintala 2015]



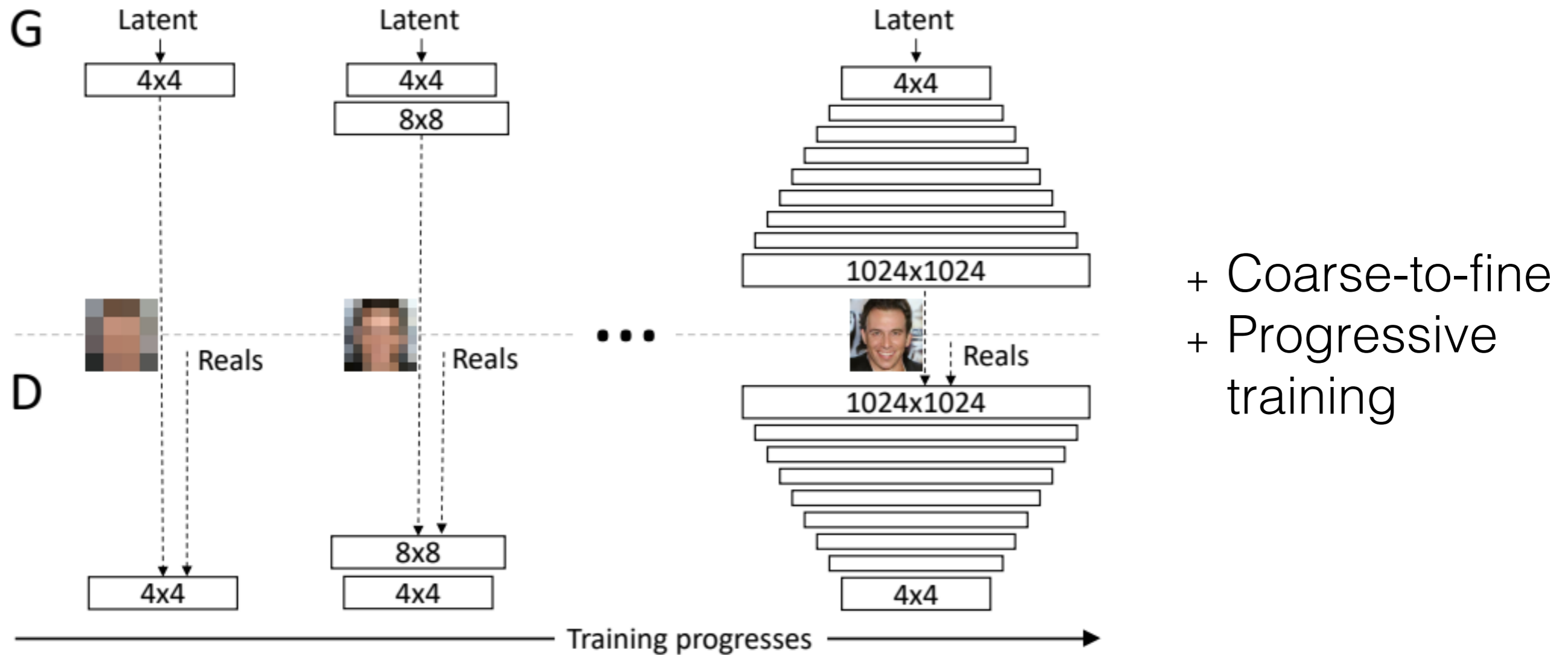
DCGAN

[Radford, Metz, Chintala 2015]



Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]



Computer vision can help quality: Gaussian Pyramid (HW1)

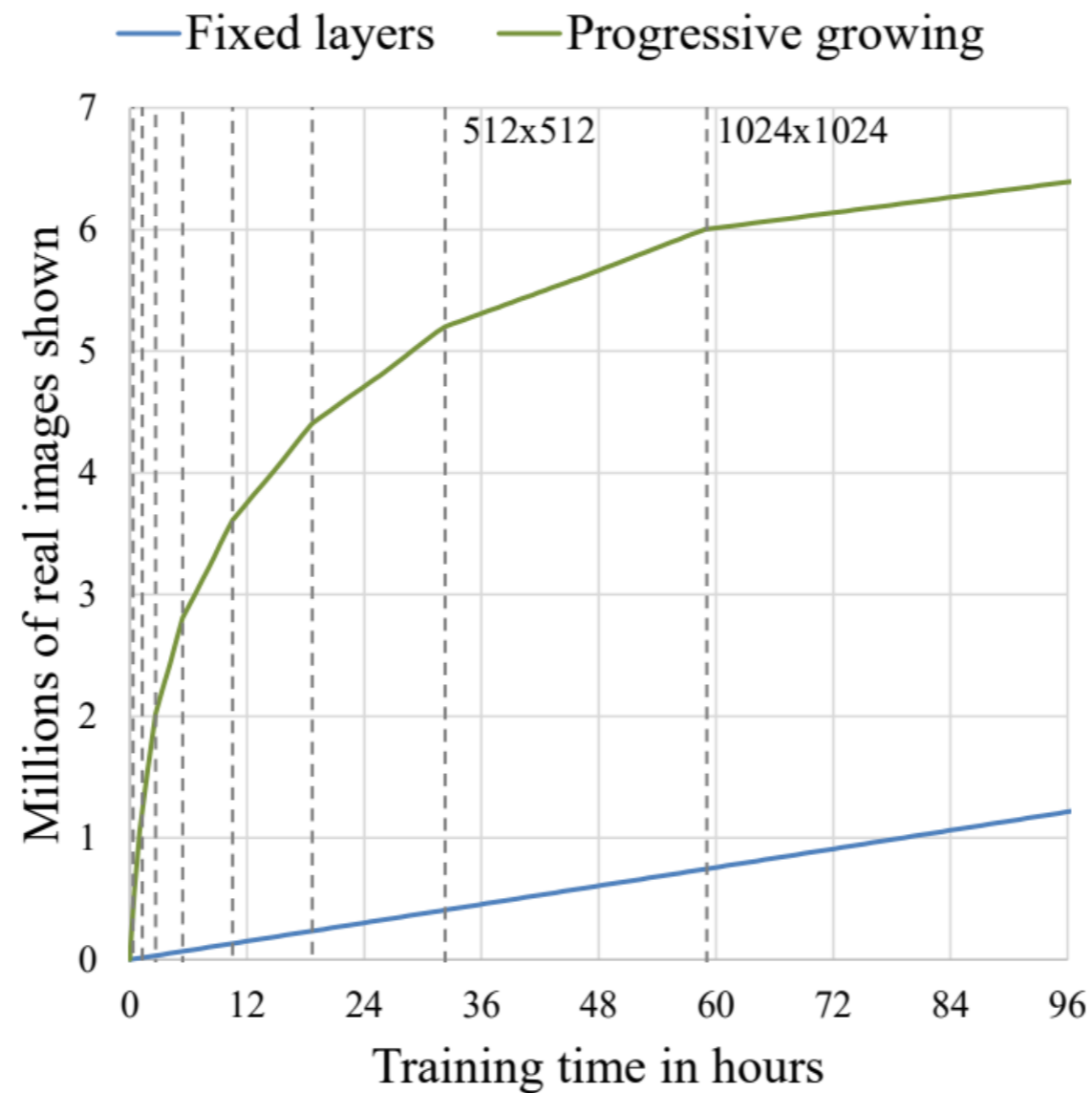
Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]



Progressive GAN: Better Training Scheme!

[Karras, Aila, Laine, Lehtinen 2018]

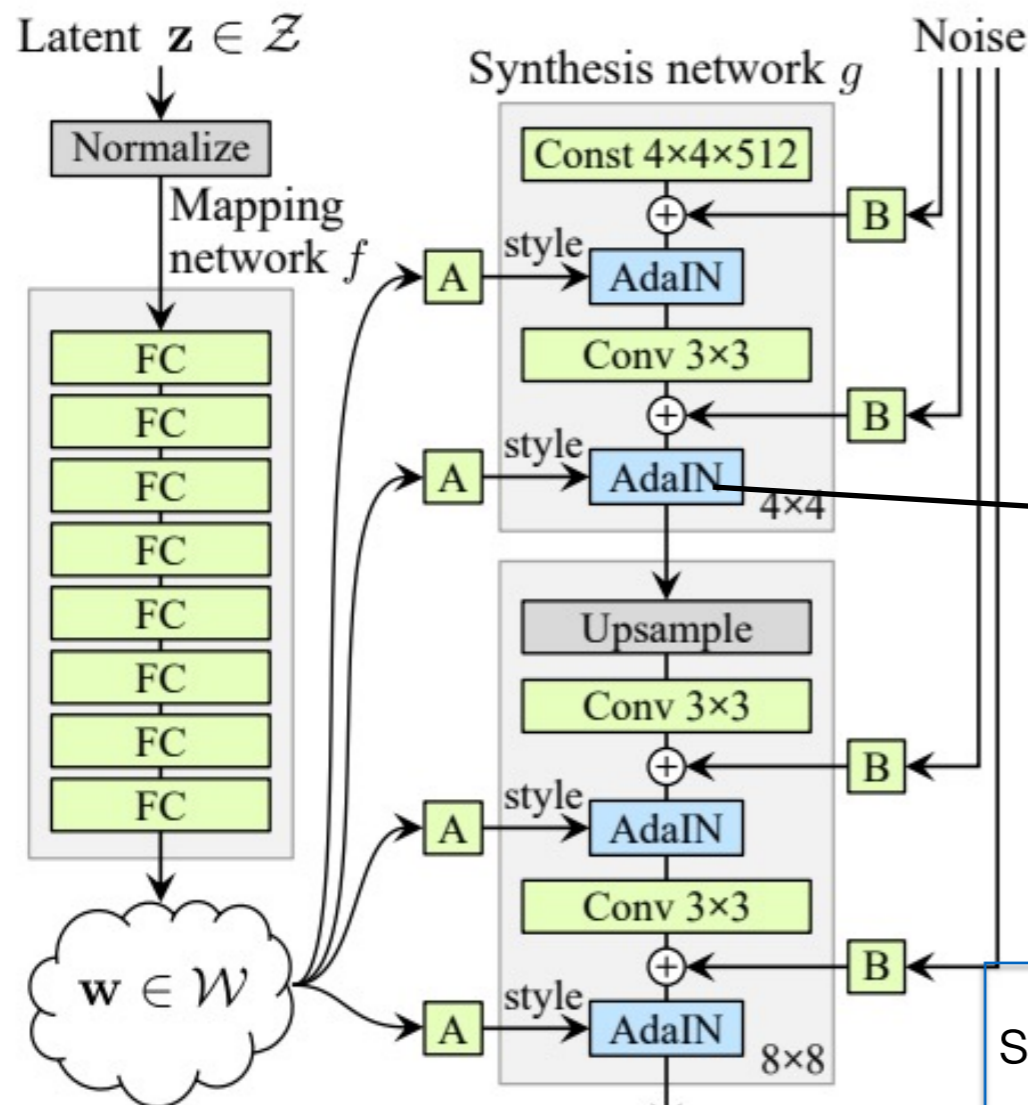


+ Coarse-to-fine
+ Progressive training

Computer vision can help speed: Gaussian Pyramid (HW1)

StyleGAN: Quality+ Control

[Karras, Laine, Aila. CVPR 2019]



- + Multiscale “style” (noise)
- + AdaIN layers

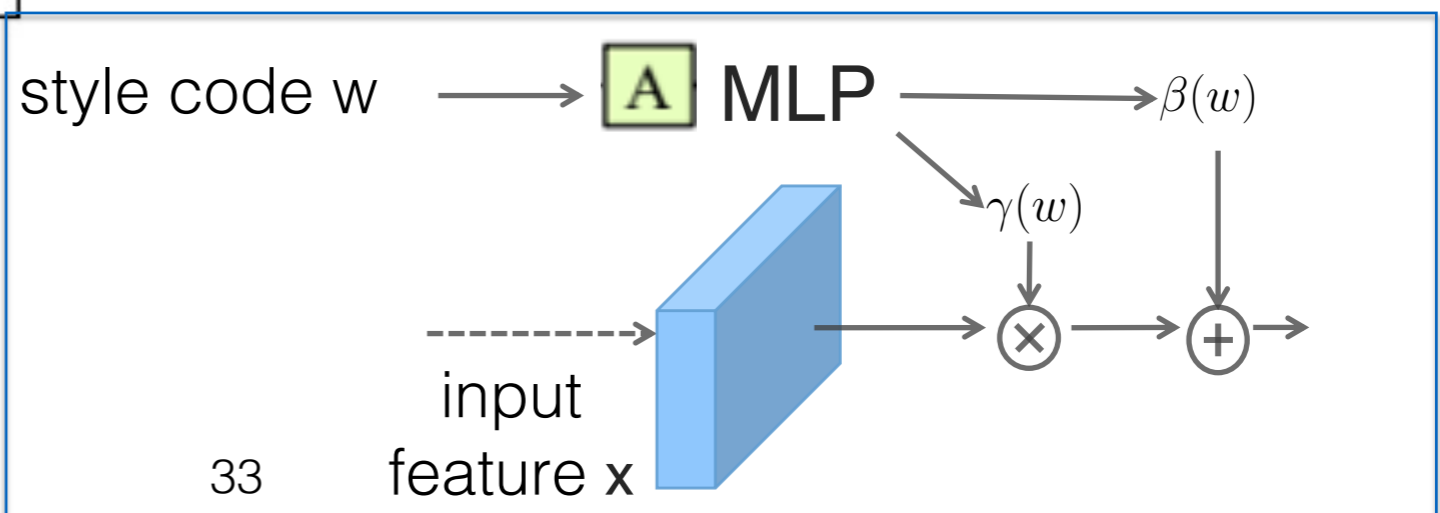
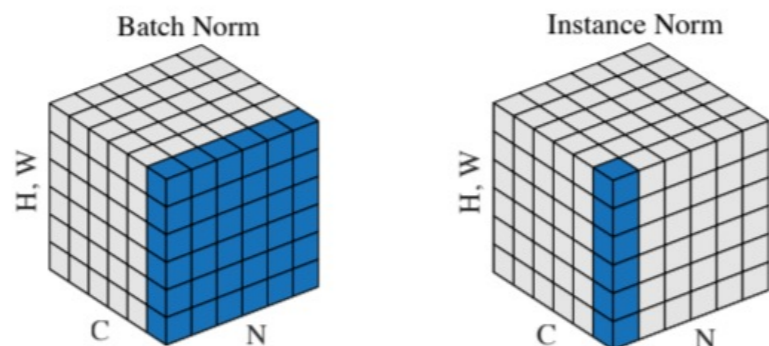
x : input feature

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

w : style code

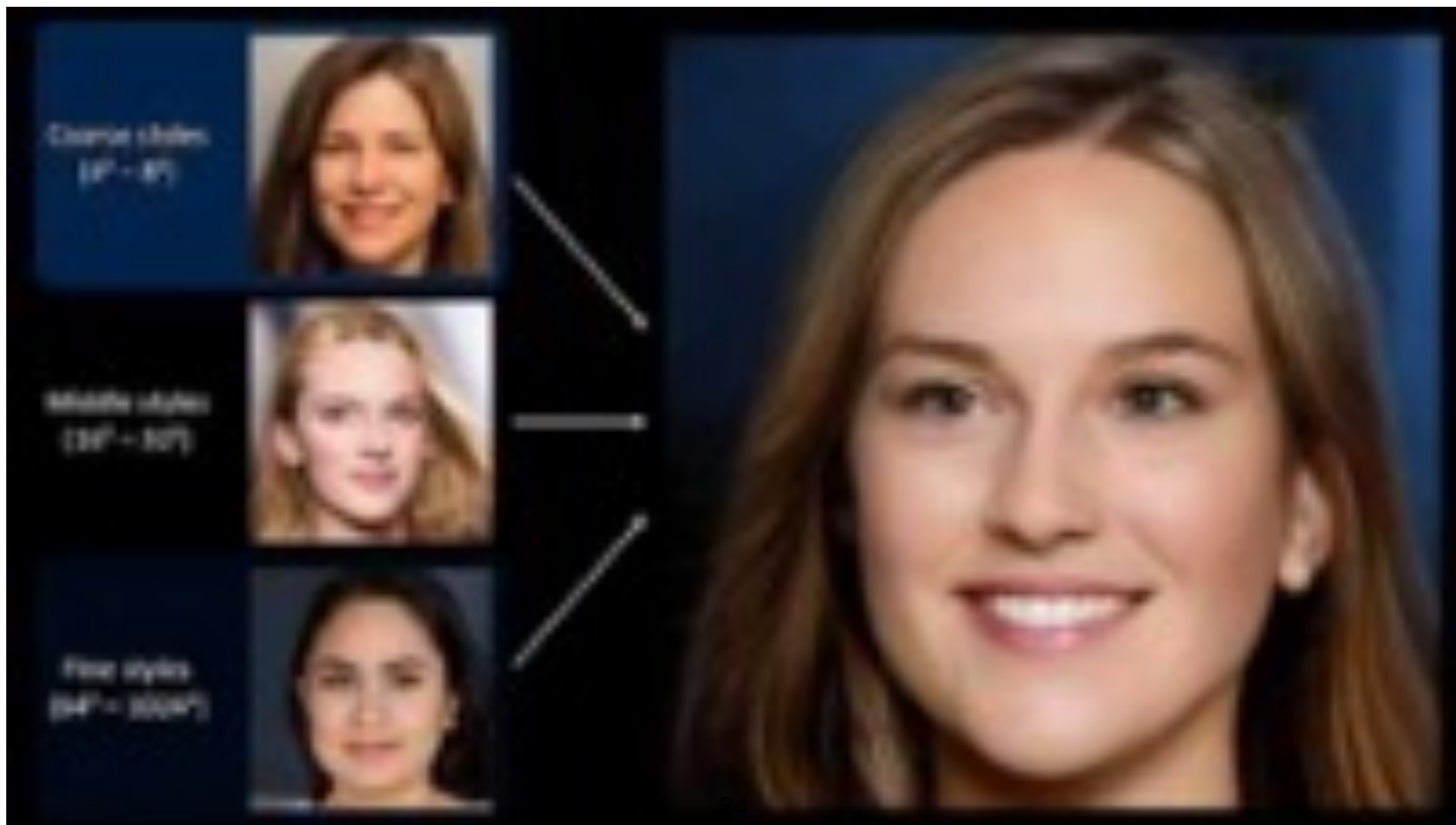
- batch/instance normalization:

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



StyleGAN: Quality+ Control

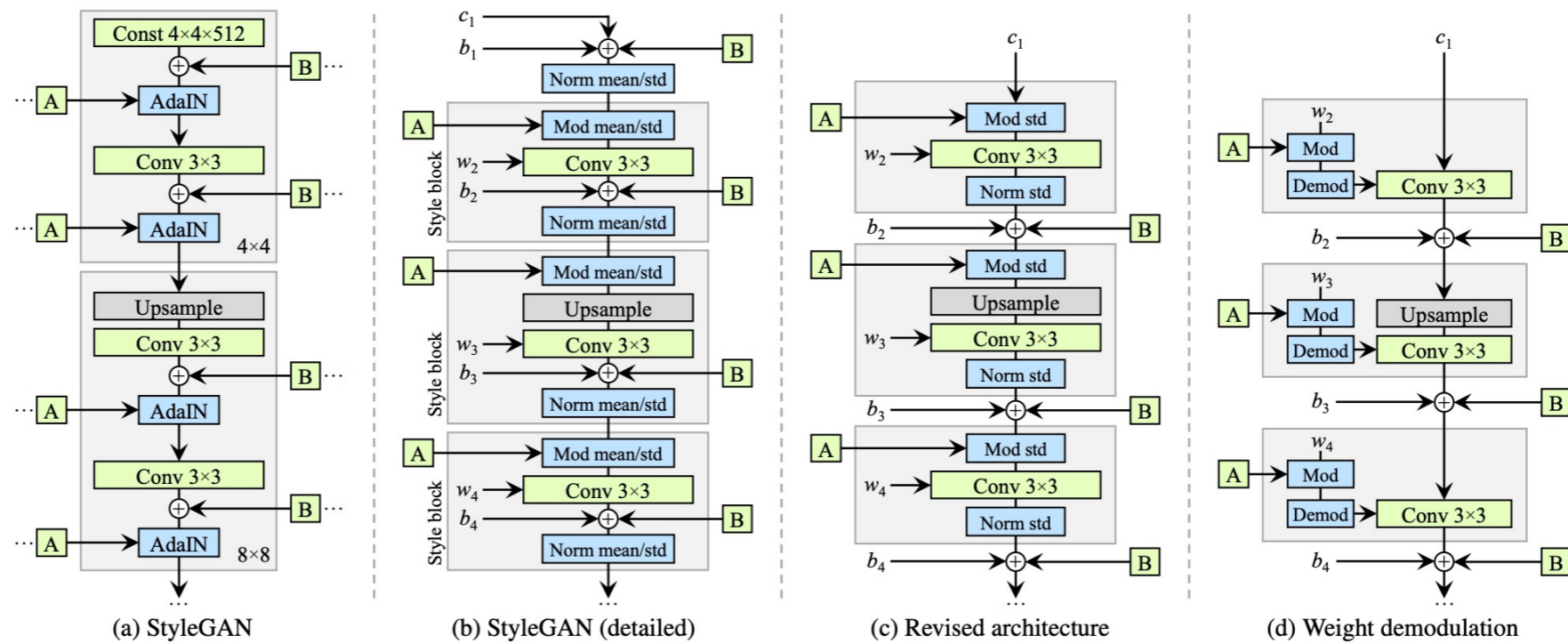
[Karras, Laine, Aila. CVPR 2019]



StyleGAN2 and StyleGAN3

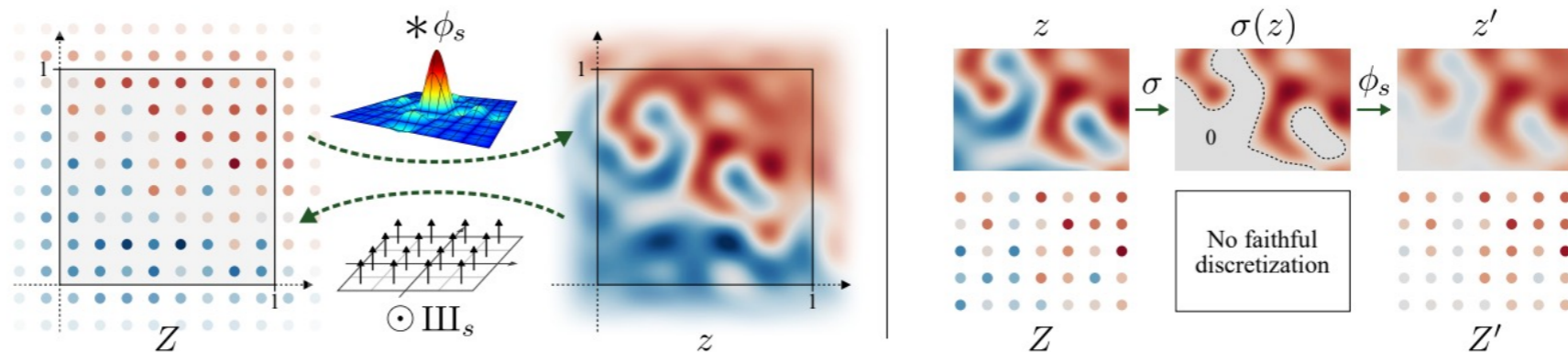
Analyzing and improving individual layers

Weight Modulation Layers



<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.

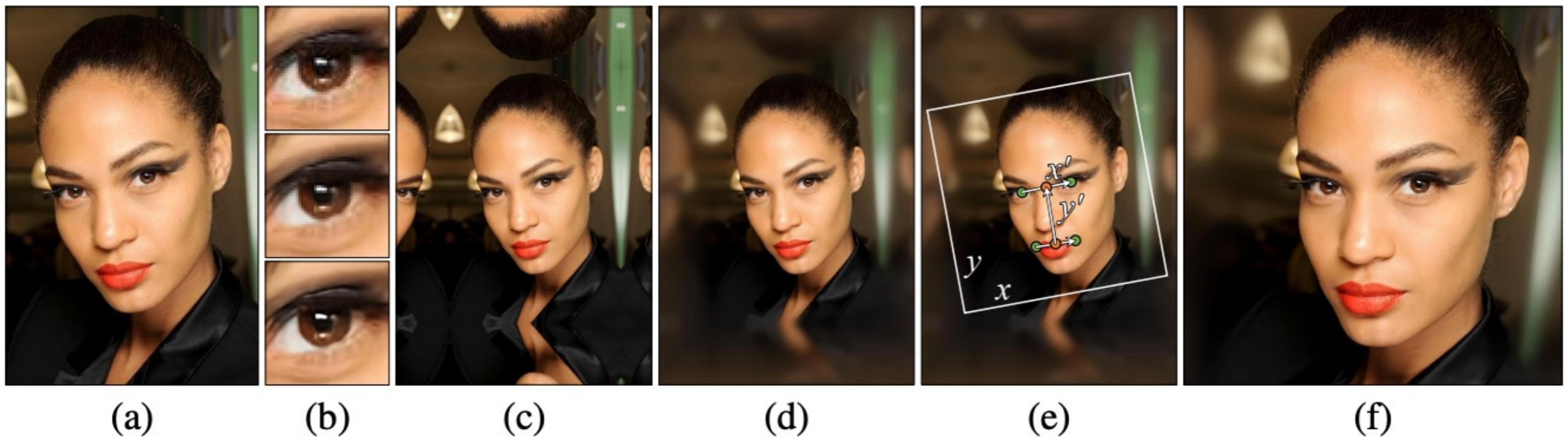


Figure 8: Creating the CELEBA-HQ dataset. We start with a JPEG image (a) from the CelebA in-the-wild dataset. We improve the visual quality (b,top) through JPEG artifact removal (b,middle) and 4x super-resolution (b,bottom). We then extend the image through mirror padding (c) and Gaussian filtering (d) 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).

Aligned vs. unaligned data



Real images from aligned FFHQ



StyleGAN2 samples

Aligned vs. unaligned data

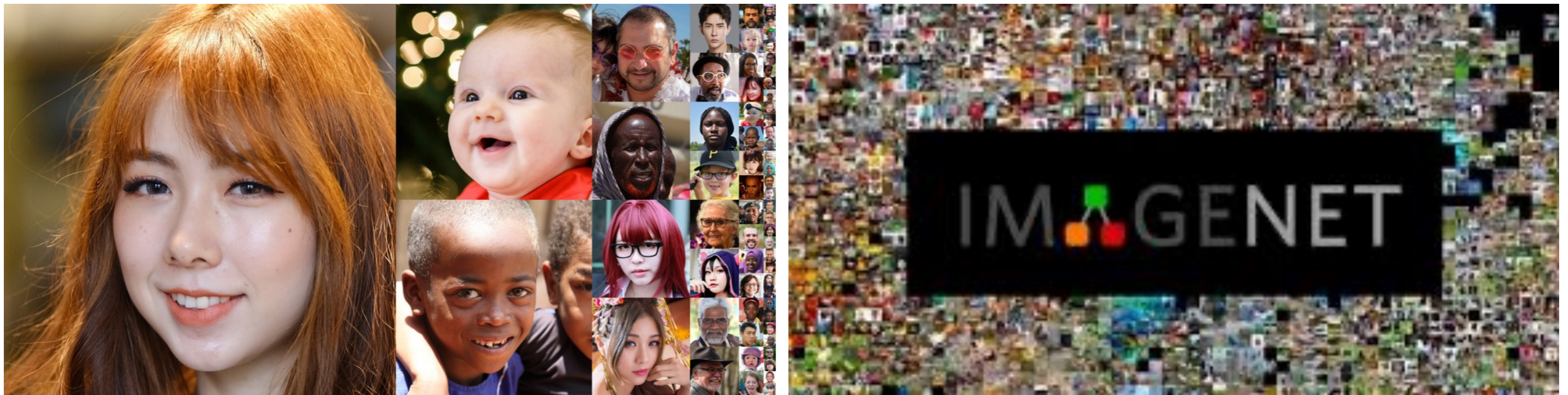


Real images from unaligned CelebA



StyleGAN2 samples

Data are Expensive



FFHQ dataset: 70,000 selective post-processed human faces ImageNet dataset: millions of images from diverse categories

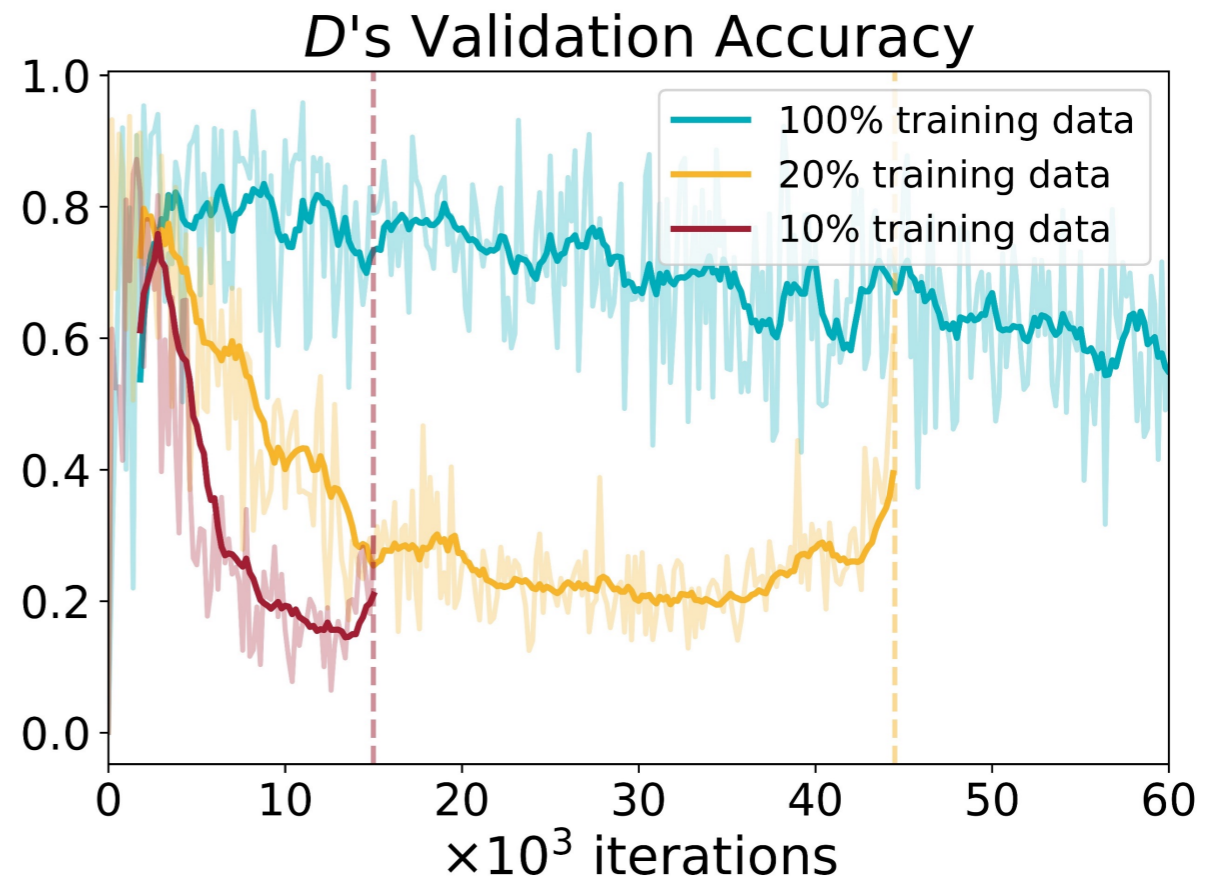
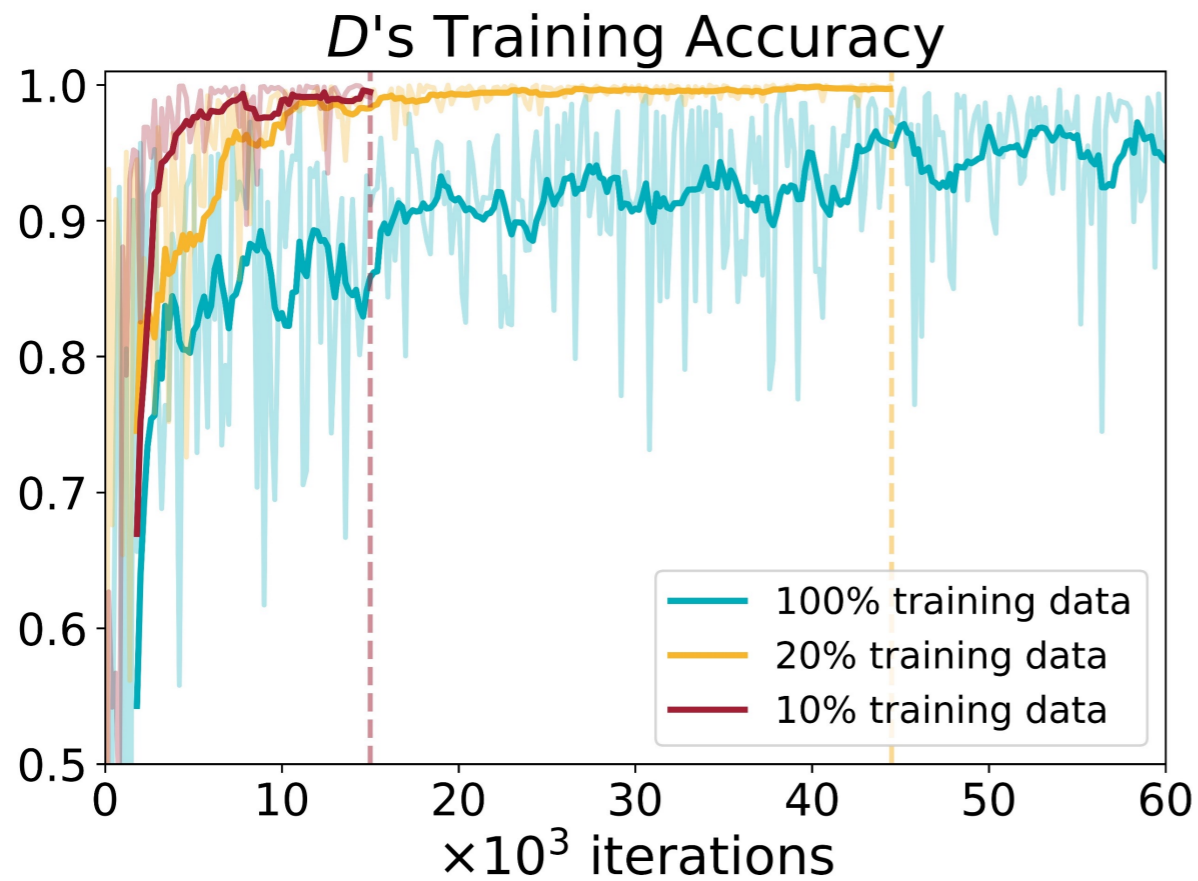
*Months or even years to collect the data,
along with *prohibitive* annotation costs.*

GANs Heavily Deteriorate Given Limited Data

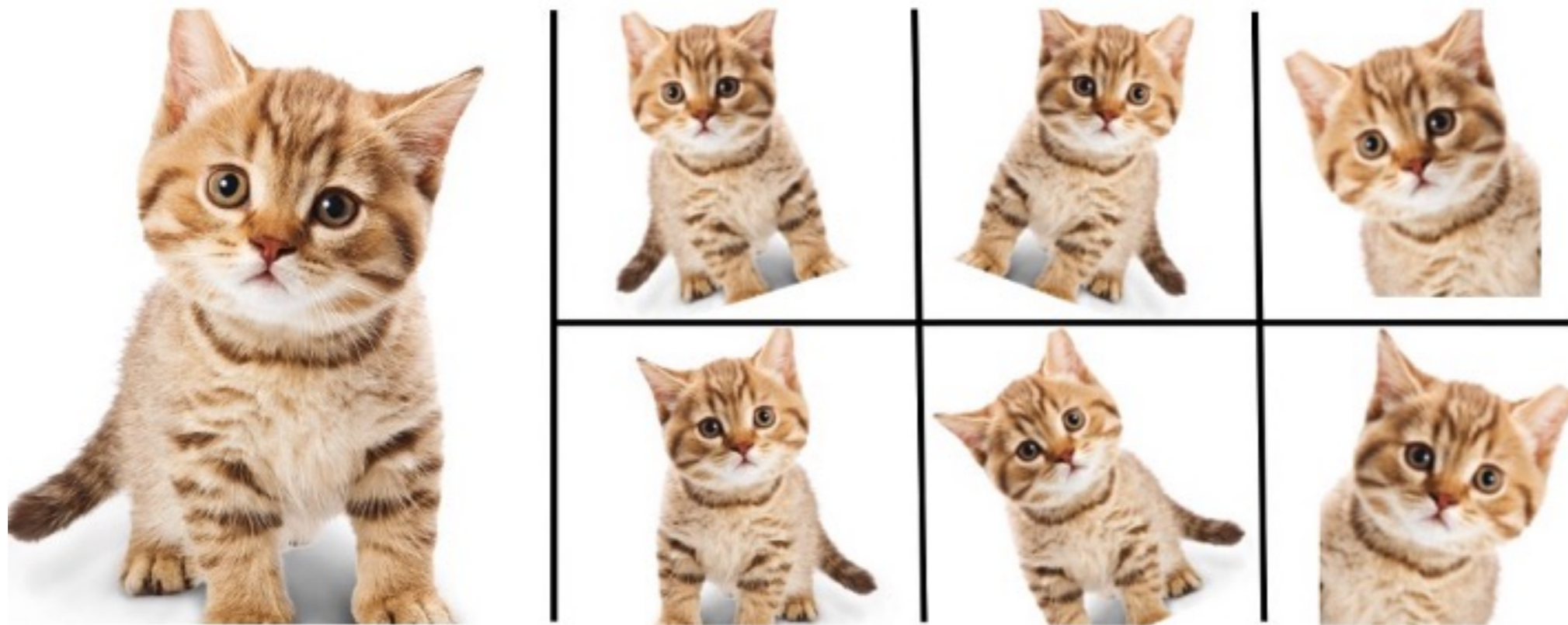


Generated samples of StyleGAN2 (Karras et al.)
using only hundreds of images

Discriminator Overfitting



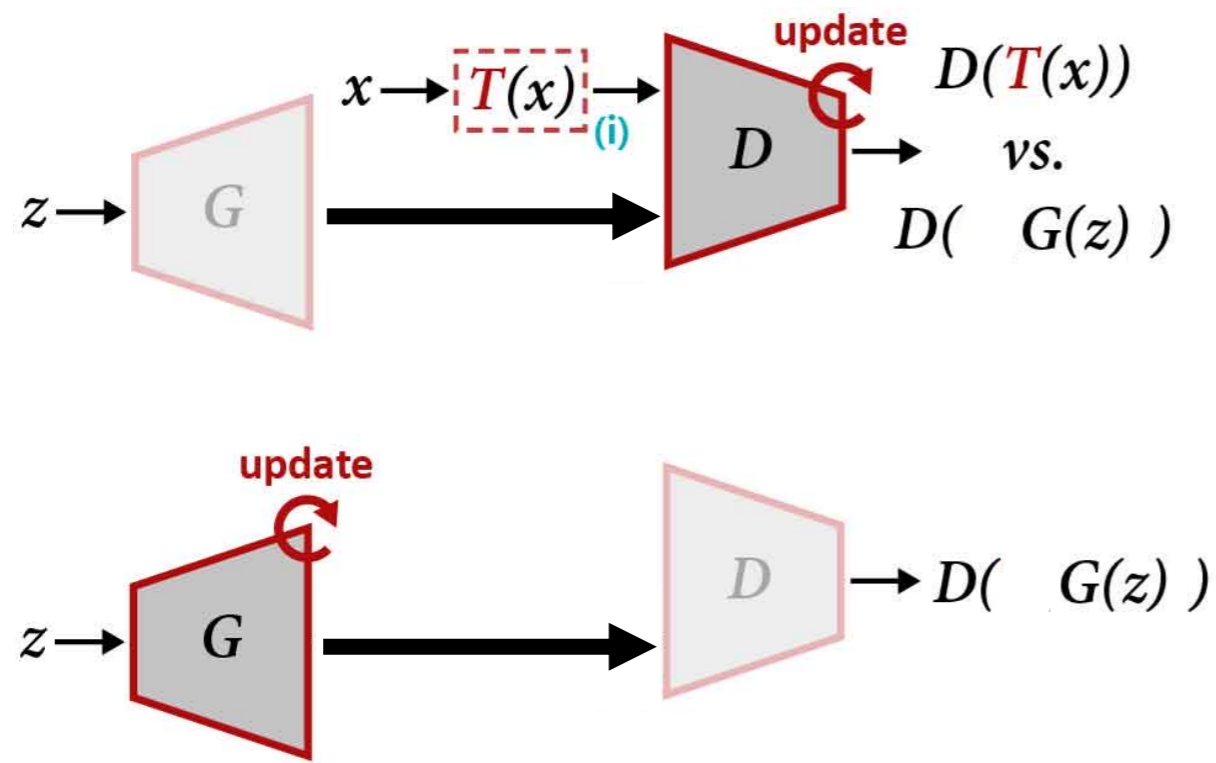
Data Augmentation



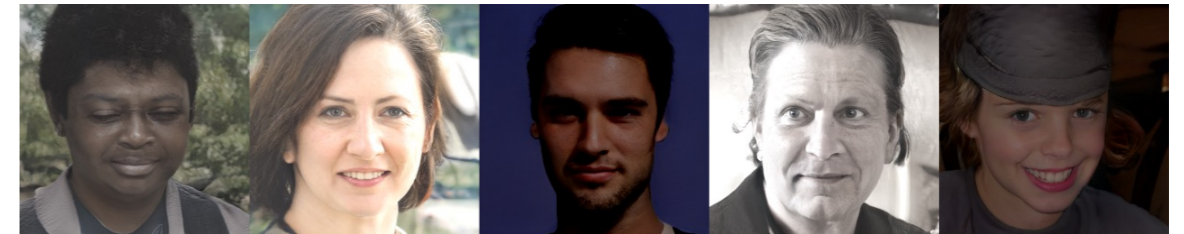
Data augmentation: enlarge datasets without collecting new samples.

How to Augment GANs?

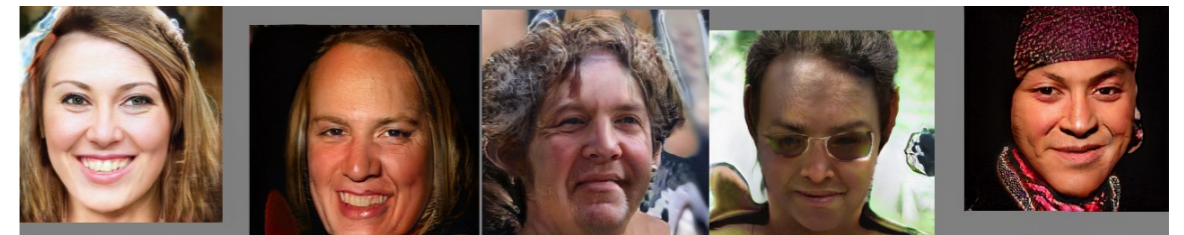
#1 Approach: Augment reals only



Generated images



Artifacts from Color jittering



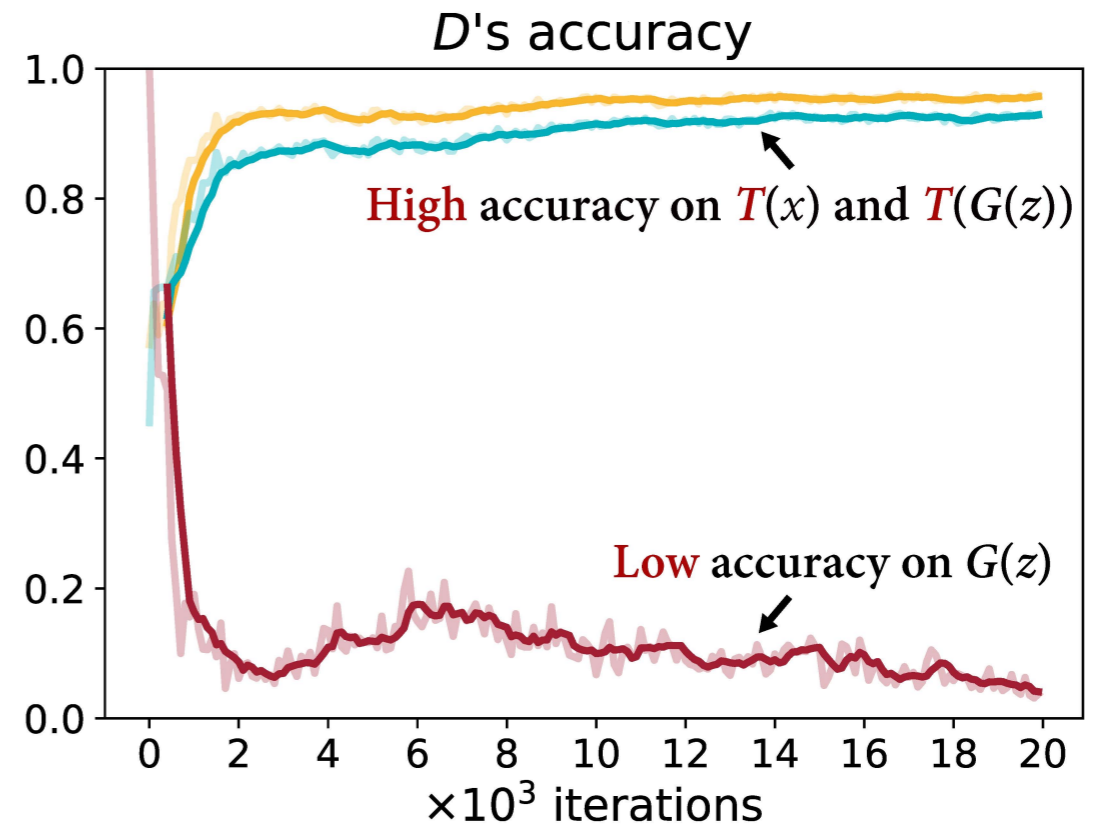
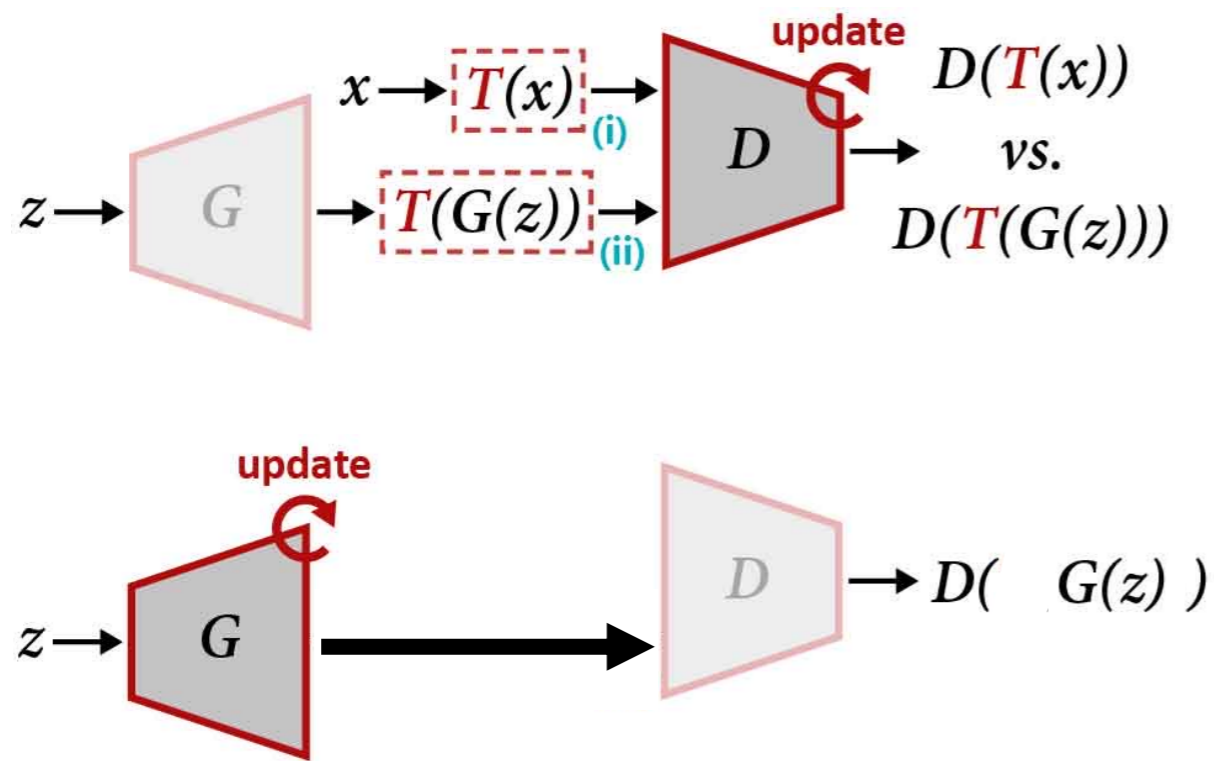
Artifacts from Translation



Artifacts from Cutout (DeVries et al.)

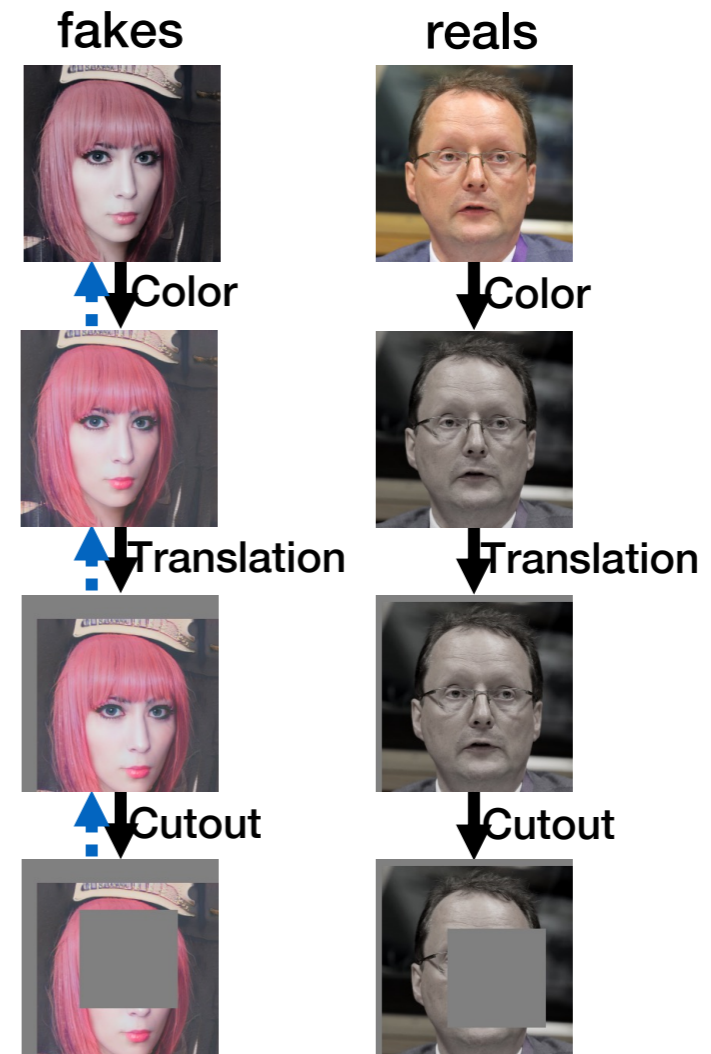
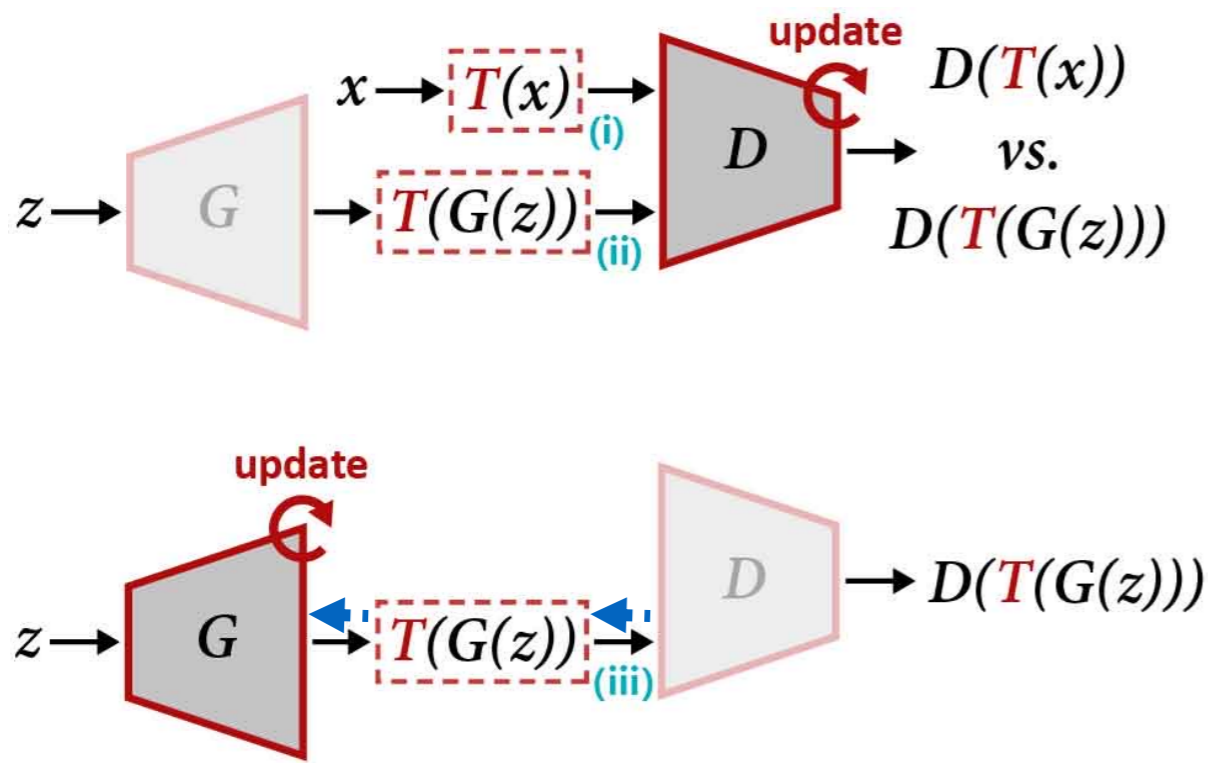
Augment reals only: the same artifacts appear on the generated images.

#2 Approach: Augment reals & fakes for D only



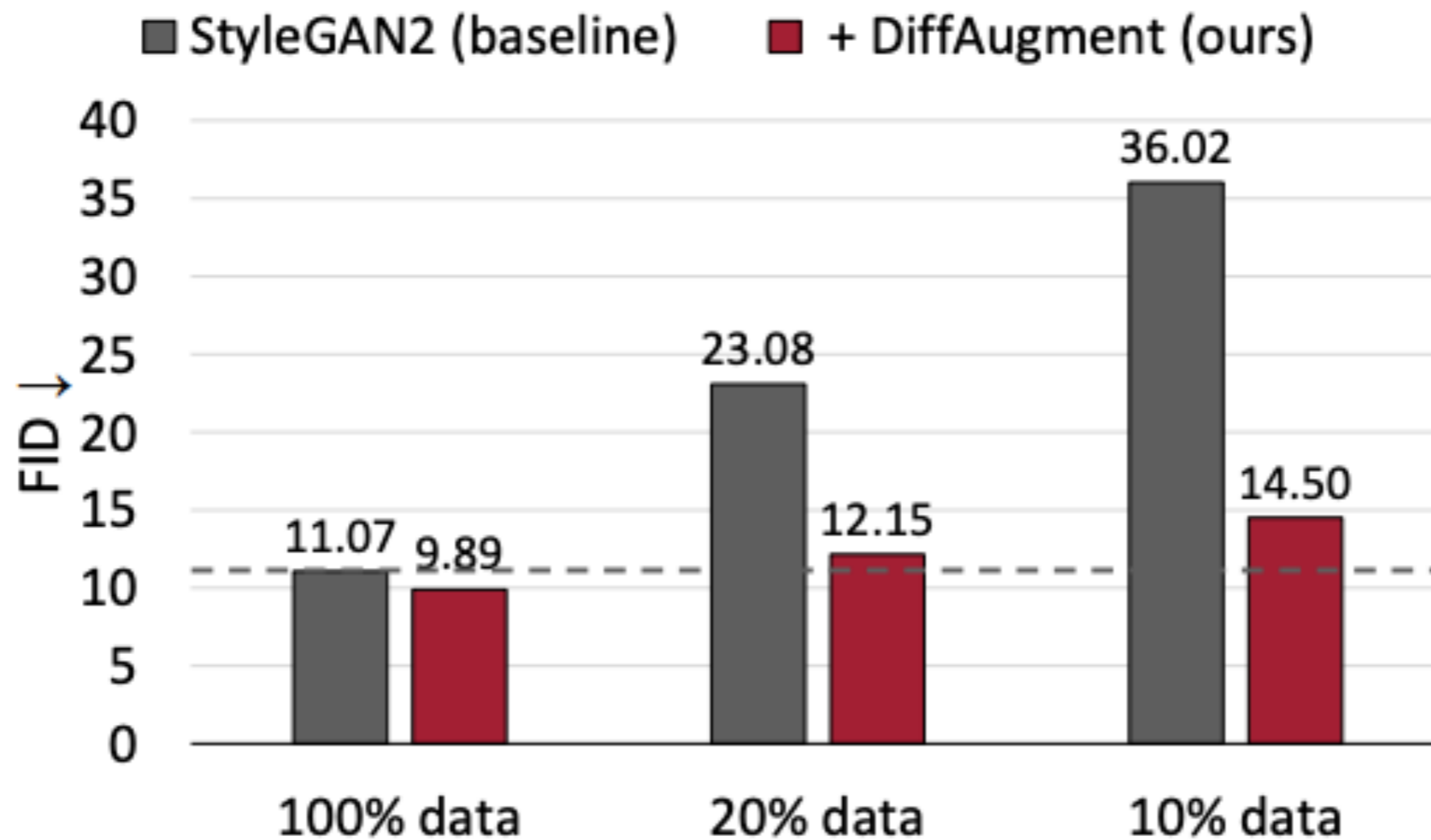
Augment D only: the unbalanced optimization cripples training.

#3 Approach: Differentiable Augmentation

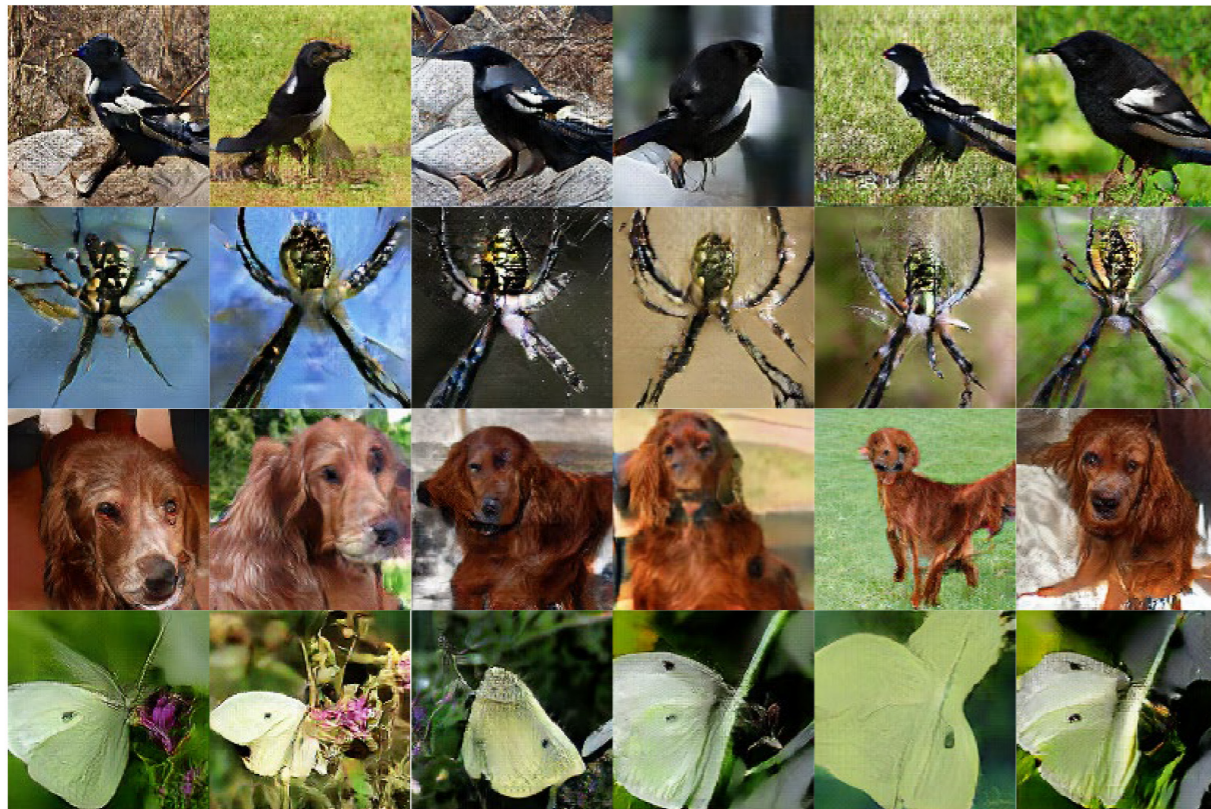


Our approach (DiffAugment): Augment reals + fakes for both D and G

CIFAR-10 (unconditional GANs)



ImageNet Generation (25% training data)



IS: 46.5 FID: 25.37

BigGAN (baseline)



IS: 74.2 FID: 13.28

+ DiffAugment (ours)

Low-Shot Generation

Obama
100 images



Cat (Simard et al.)
160 images



Dog (Simard et al.)
389 images



StyleGAN2 (baseline)

StyleGAN2 + DiffAugment (ours)

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.

```
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...
...
```



Data Augmentation for GANs

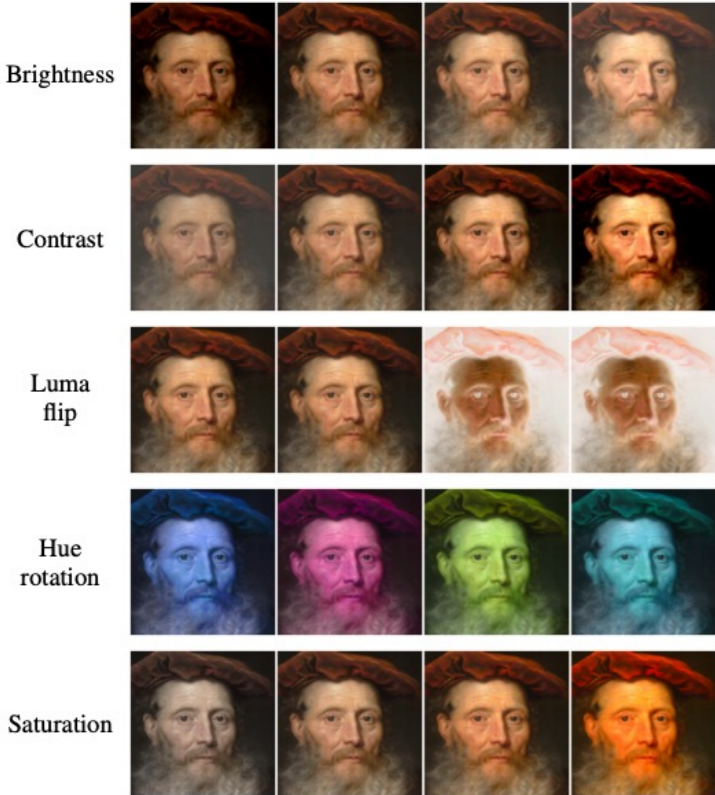
- *Differentiable Augmentation for Data-Efficient GAN Training (DiffAugment)*. Zhao et al., NeurIPS 2020.
- *Training Generative Adversarial Networks with Limited Data. (StyleGAN2-ADA)*. Karras et al., NeurIPS 2020.
- *On Data Augmentation for GAN Training*. Tran et al., IEEE TIP, 2020.
- *Image Augmentations for GAN Training*. Zhao et al., arXiv, 2020.

StyleGAN2-ADA

Pixel blitting



Color transformations



General geometric transformations



Image-space filtering

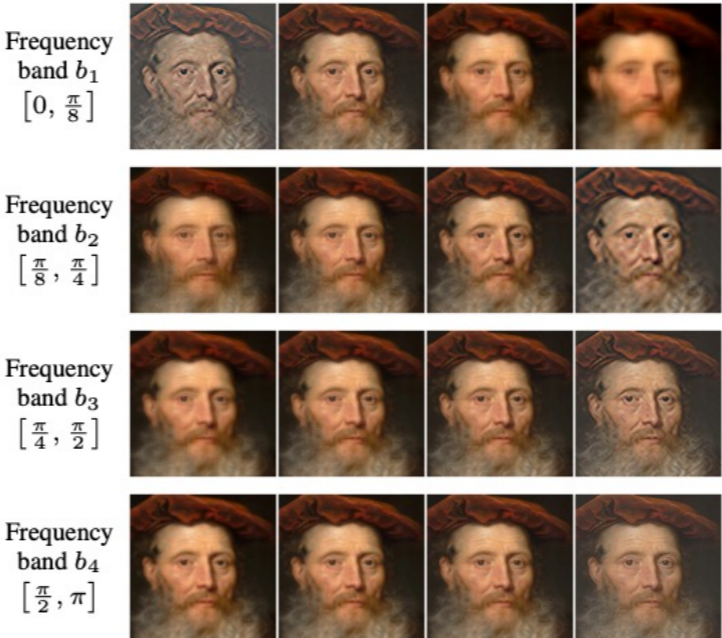
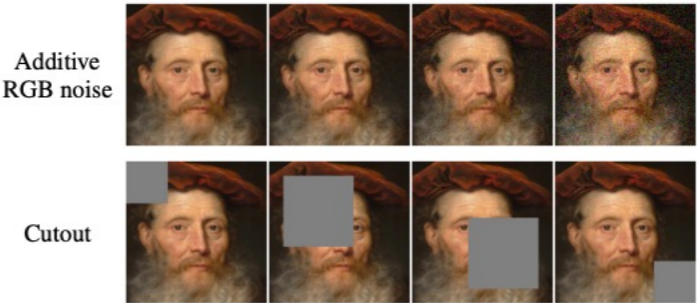


Image-space corruptions



StyleGAN2-ADA

Adaptative data augmentation

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

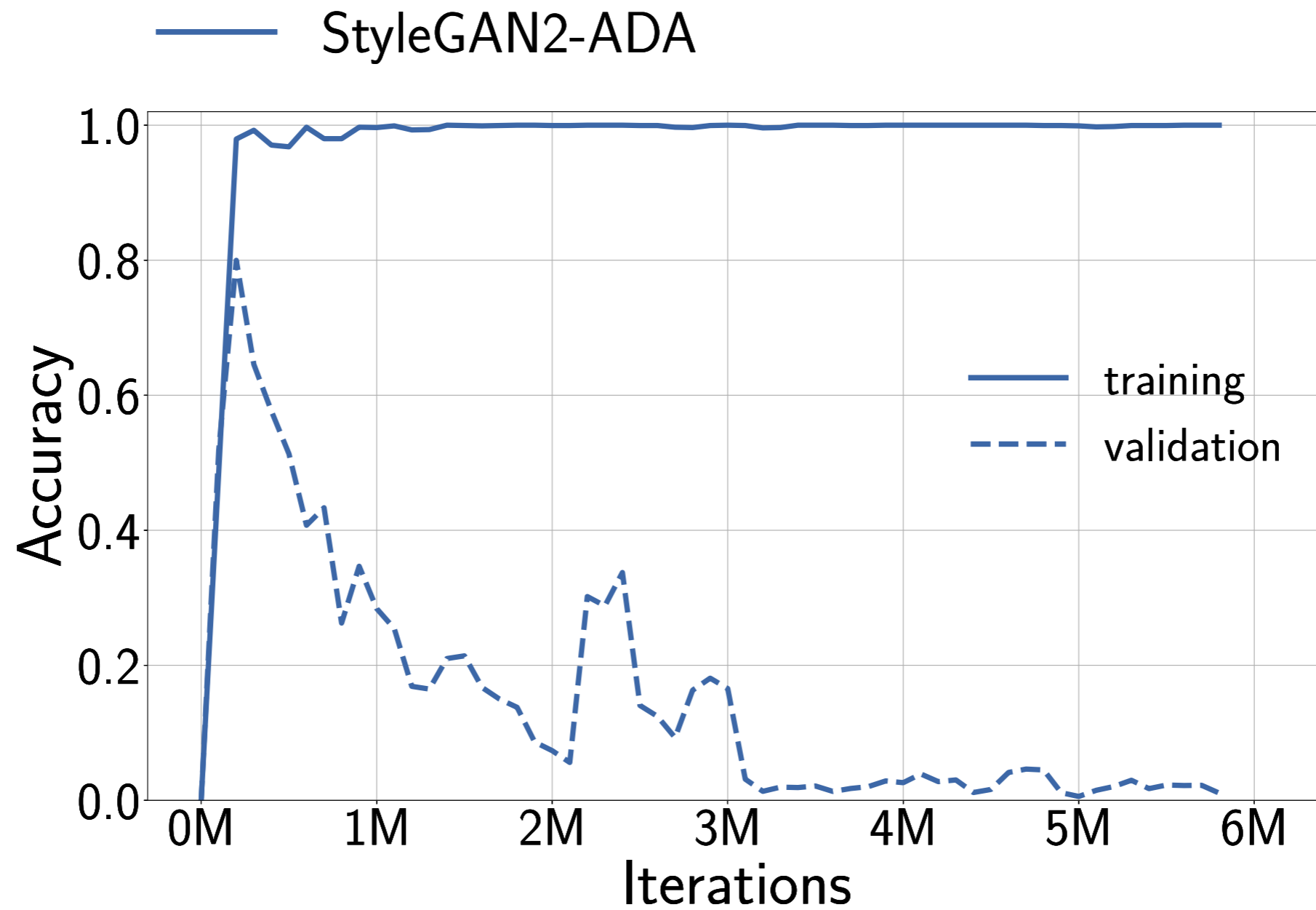
$r_t = 0$ no overfitting, decrease augmentation
 $r_t = 1$ complete overfitting, increase augmentation

Other metrics to consider:

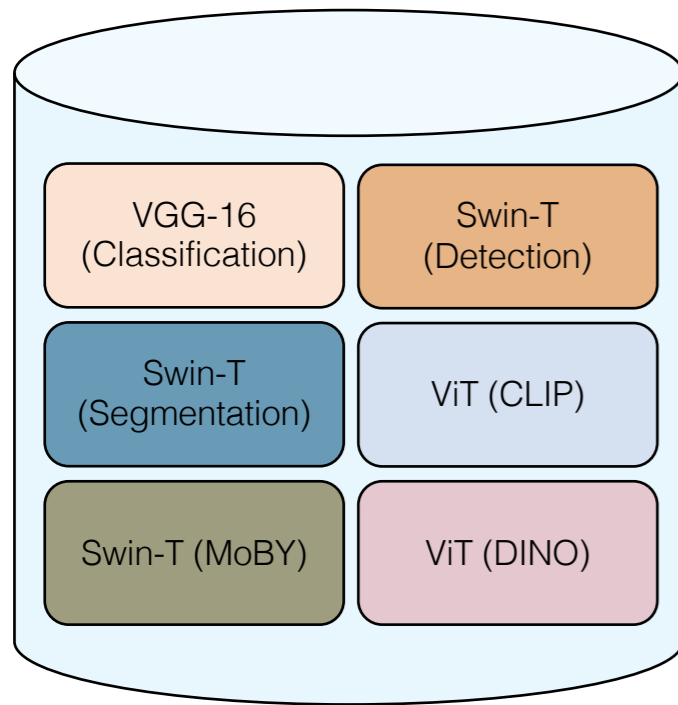
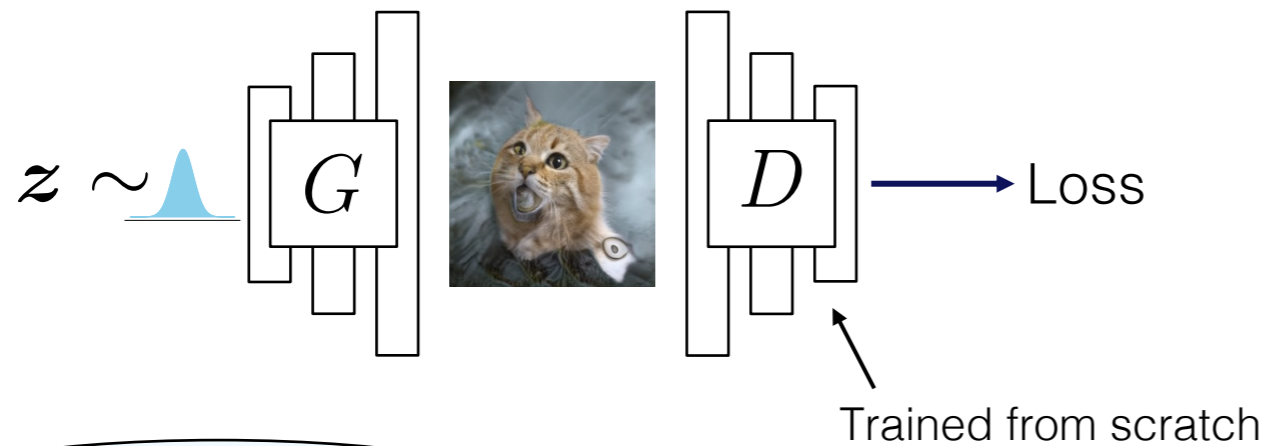
$$\frac{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{validation}}]}{\mathbb{E}[D_{\text{train}}] - \mathbb{E}[D_{\text{generated}}]} \quad \mathbb{E}[D_{\text{train}}]$$

Training methods

Discriminator is still Overfitting



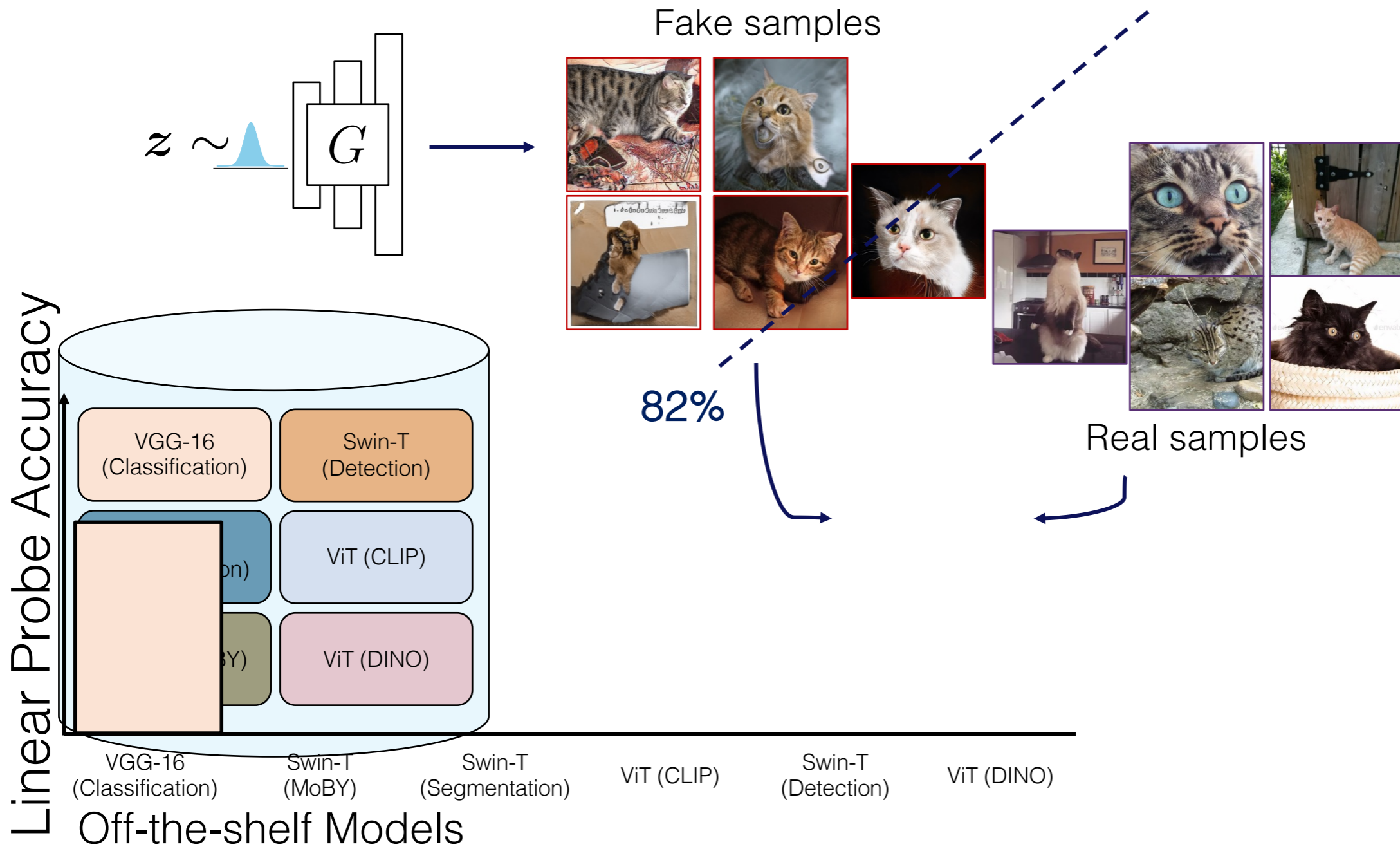
Standard GAN training



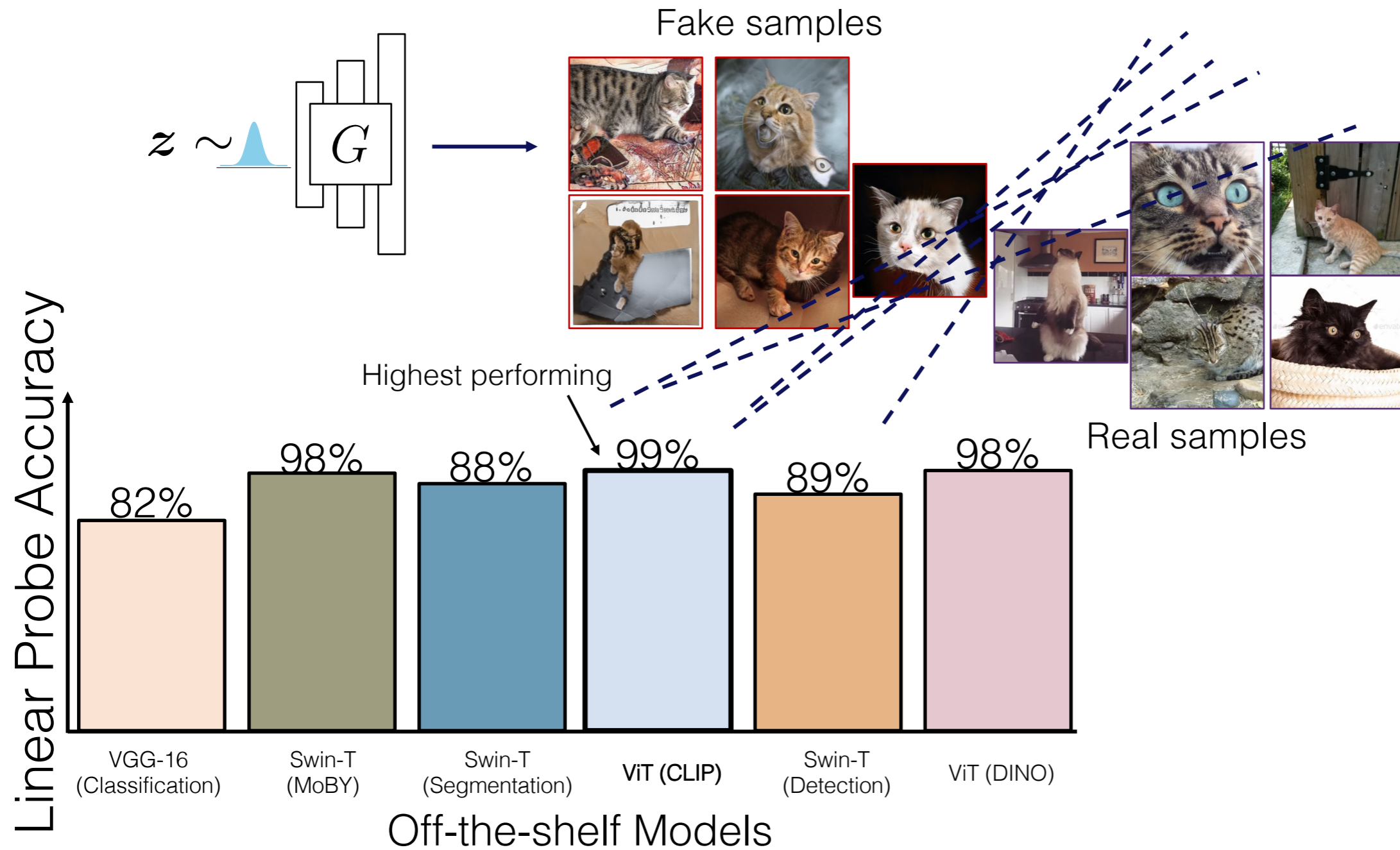
Off-the-shelf Models

Which pretrained models to use?

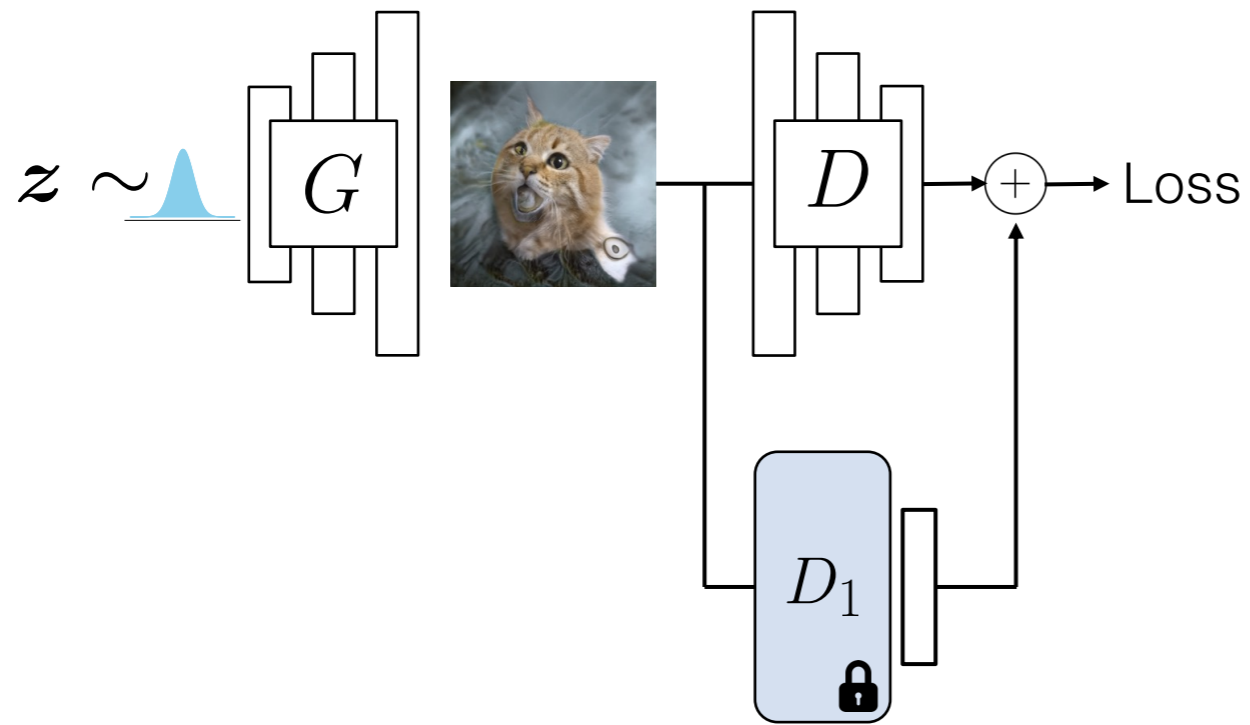
Model Selection



Model Selection



Vision-aided GAN training



VGG-16
(Classification)

Swin-T
(MoBY)

Swin-T
(Segmentation)

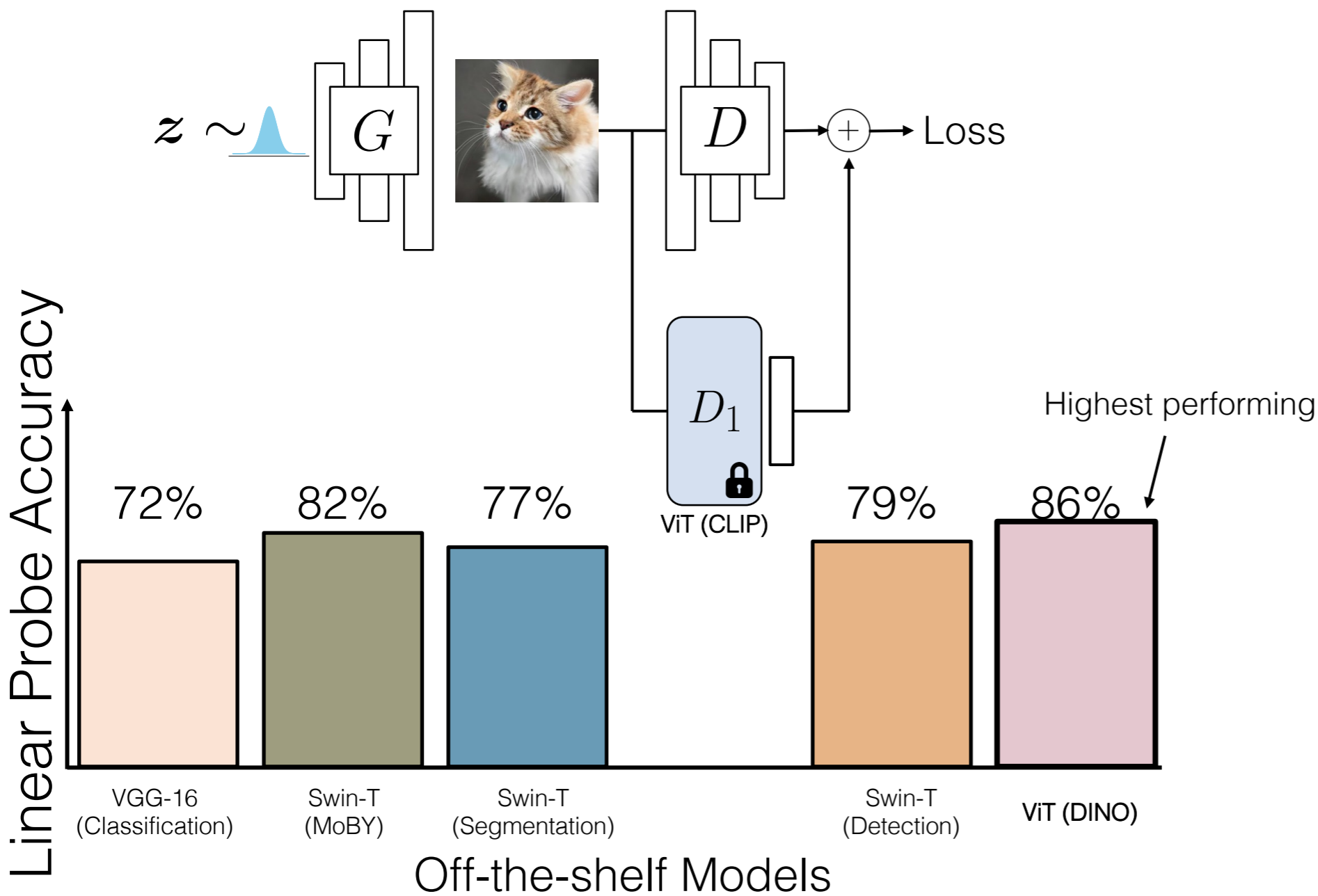
ViT (CLIP)

Swin-T
(Detection)

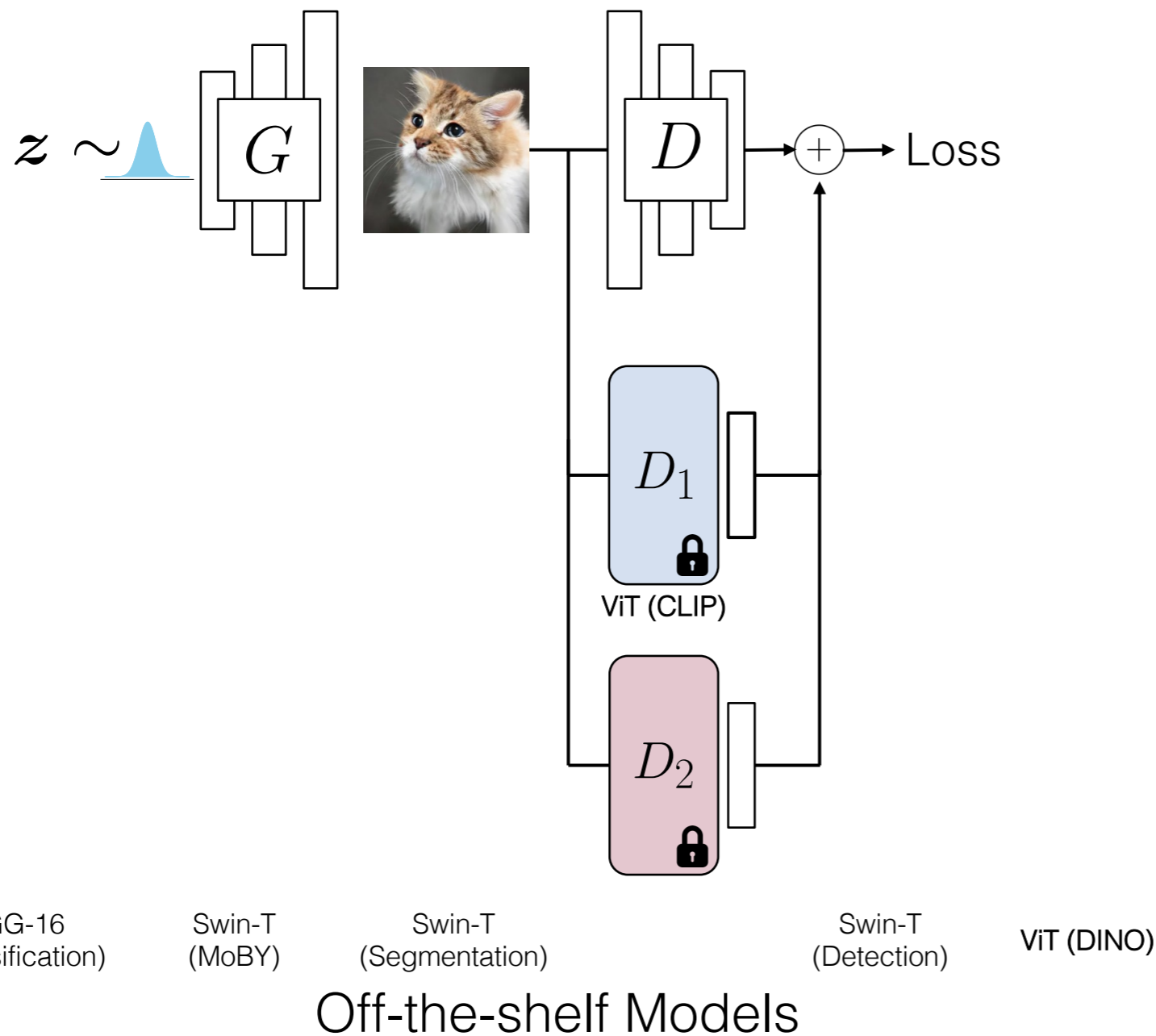
ViT (DINO)

Off-the-shelf Models

Add 2nd Vision-aided discriminator

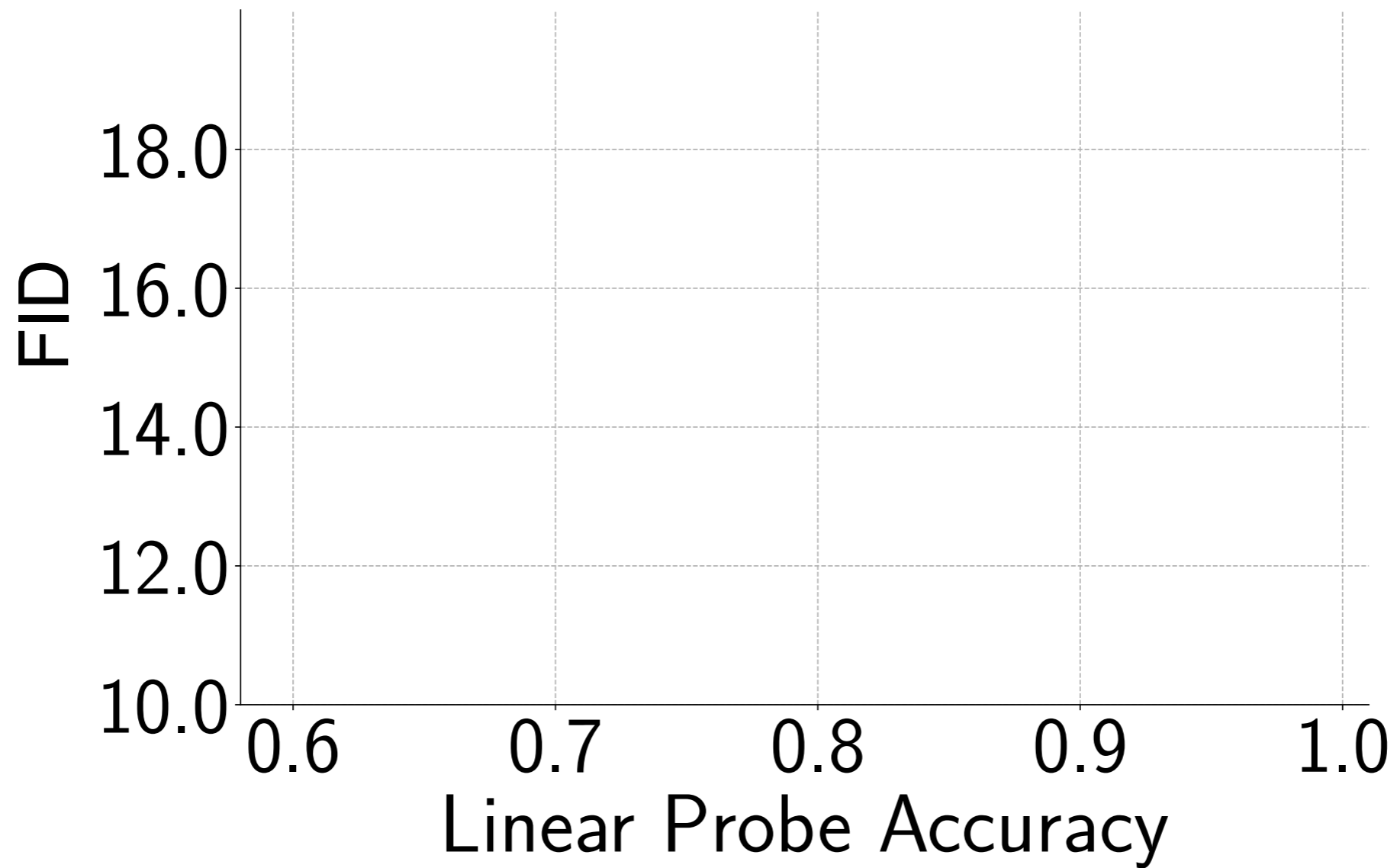


Add 2nd Vision-aided discriminator

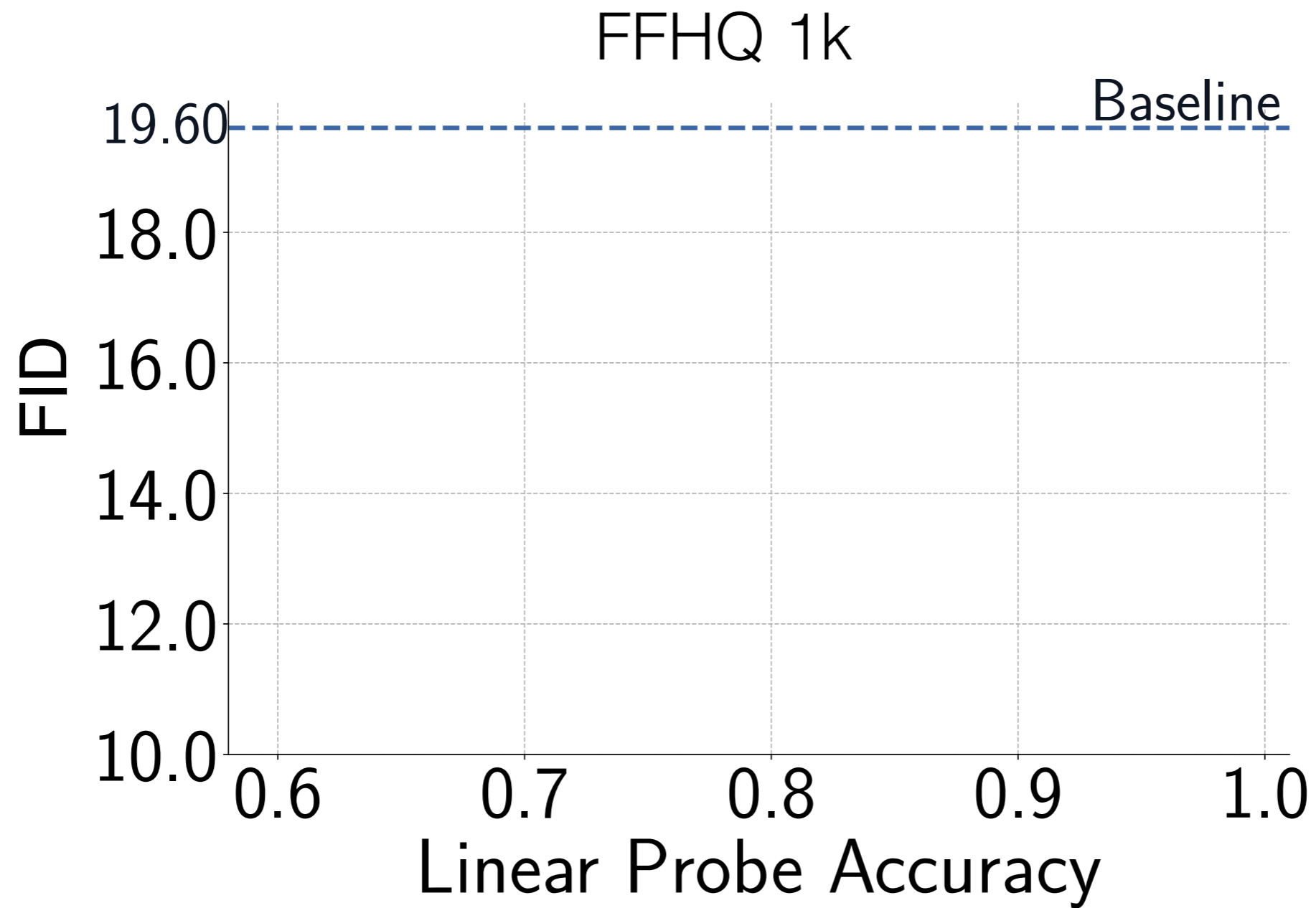


Better pretrained features lead to lower FID

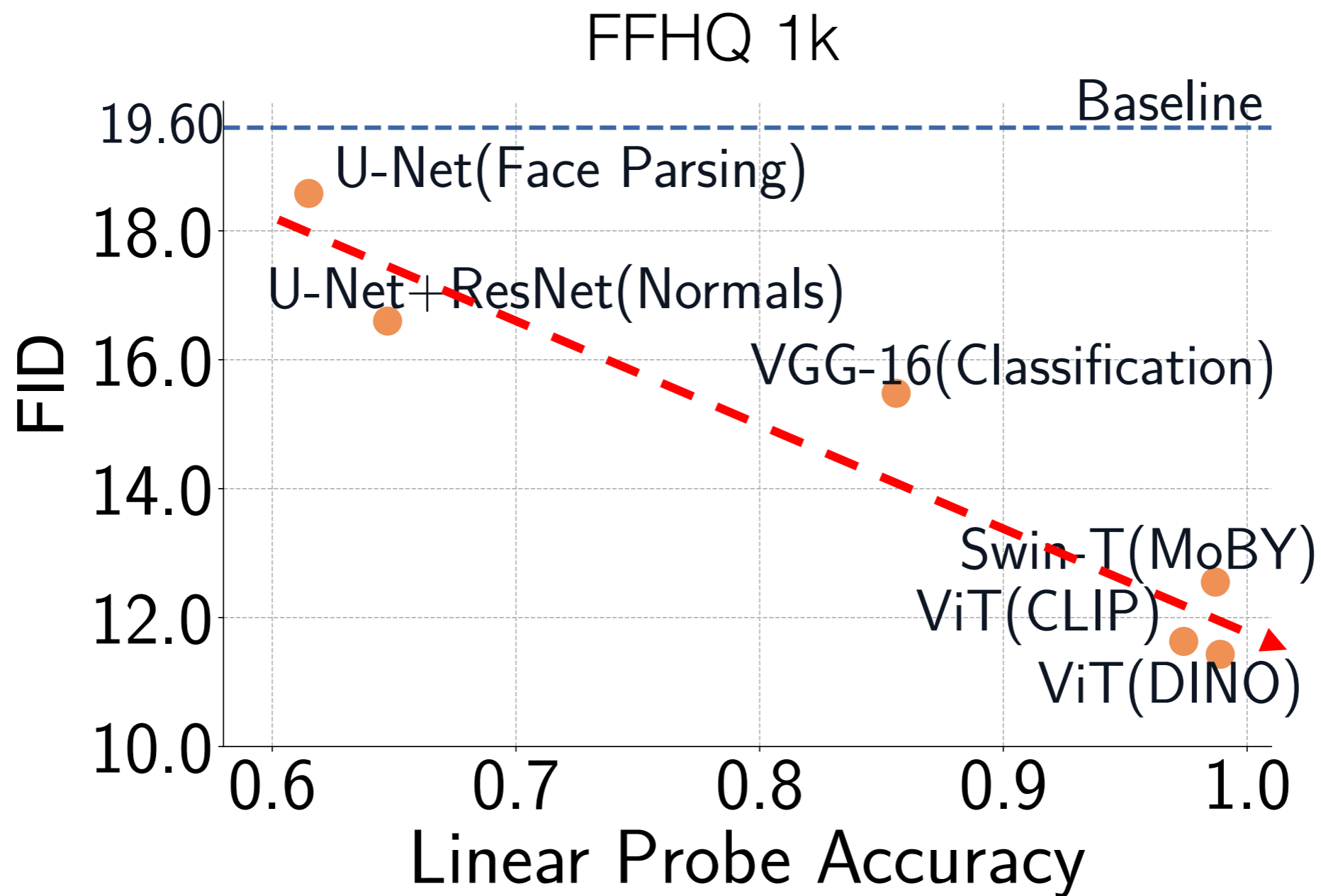
FFHQ 1k



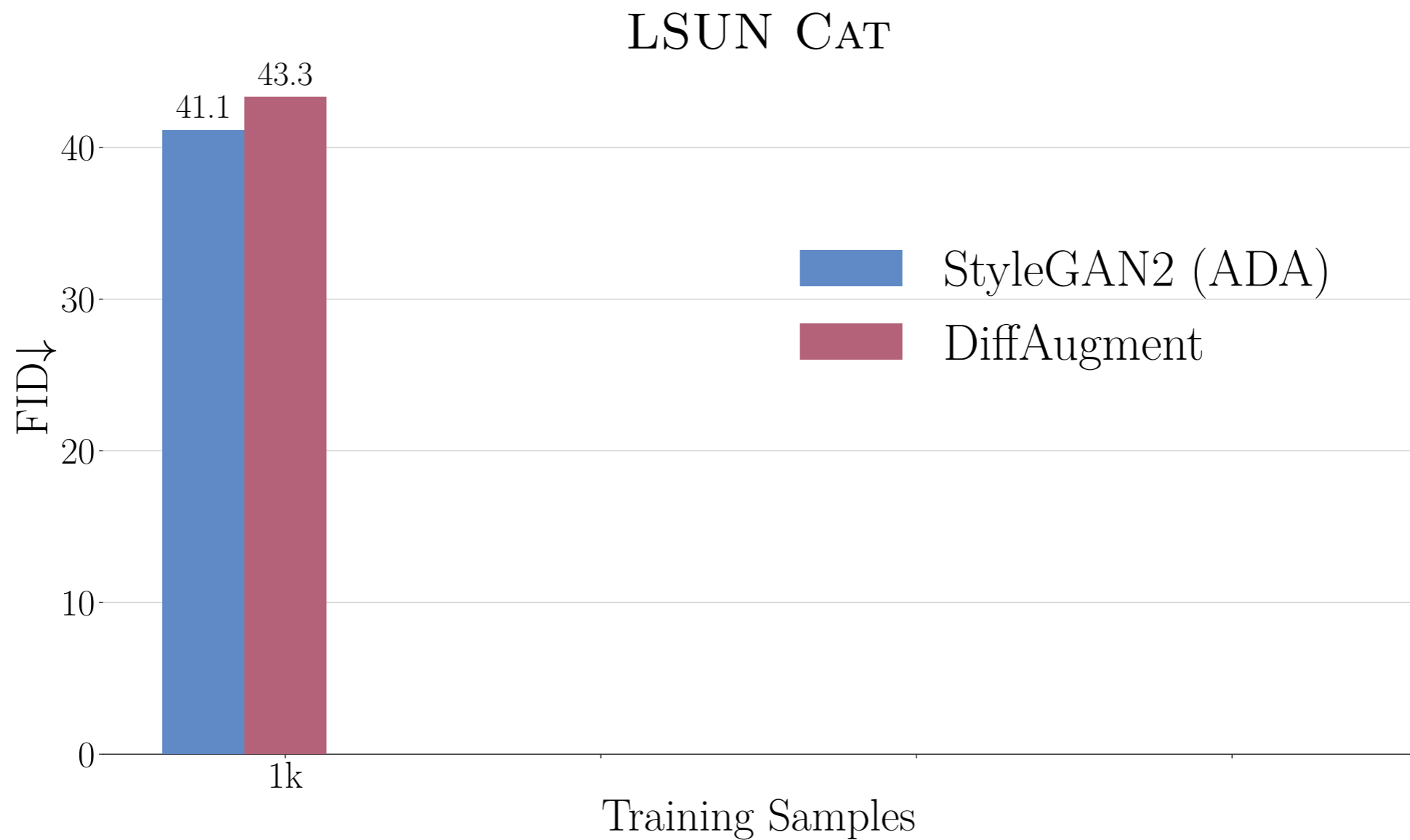
Better pretrained features lead to lower FID



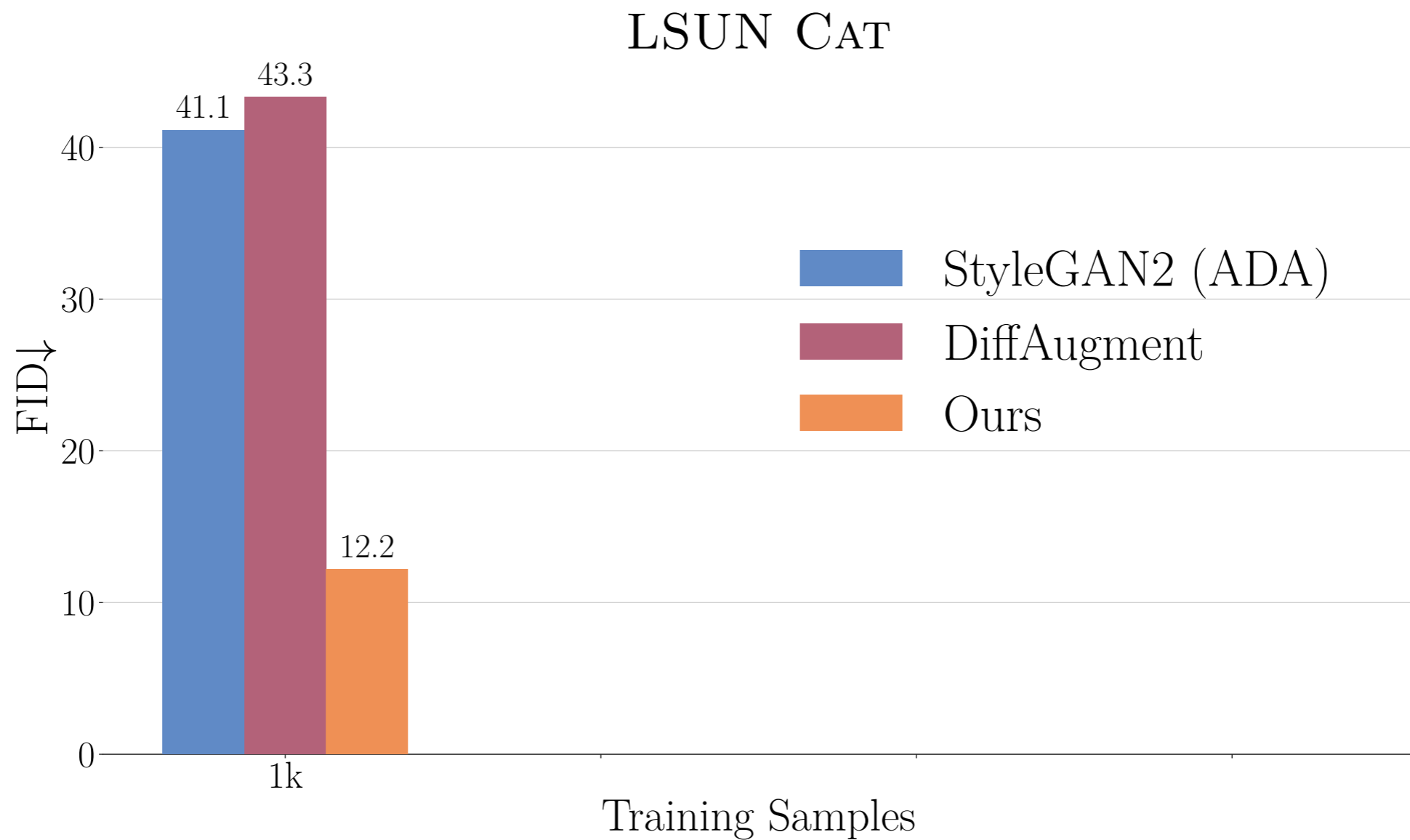
Better pretrained features lead to lower FID



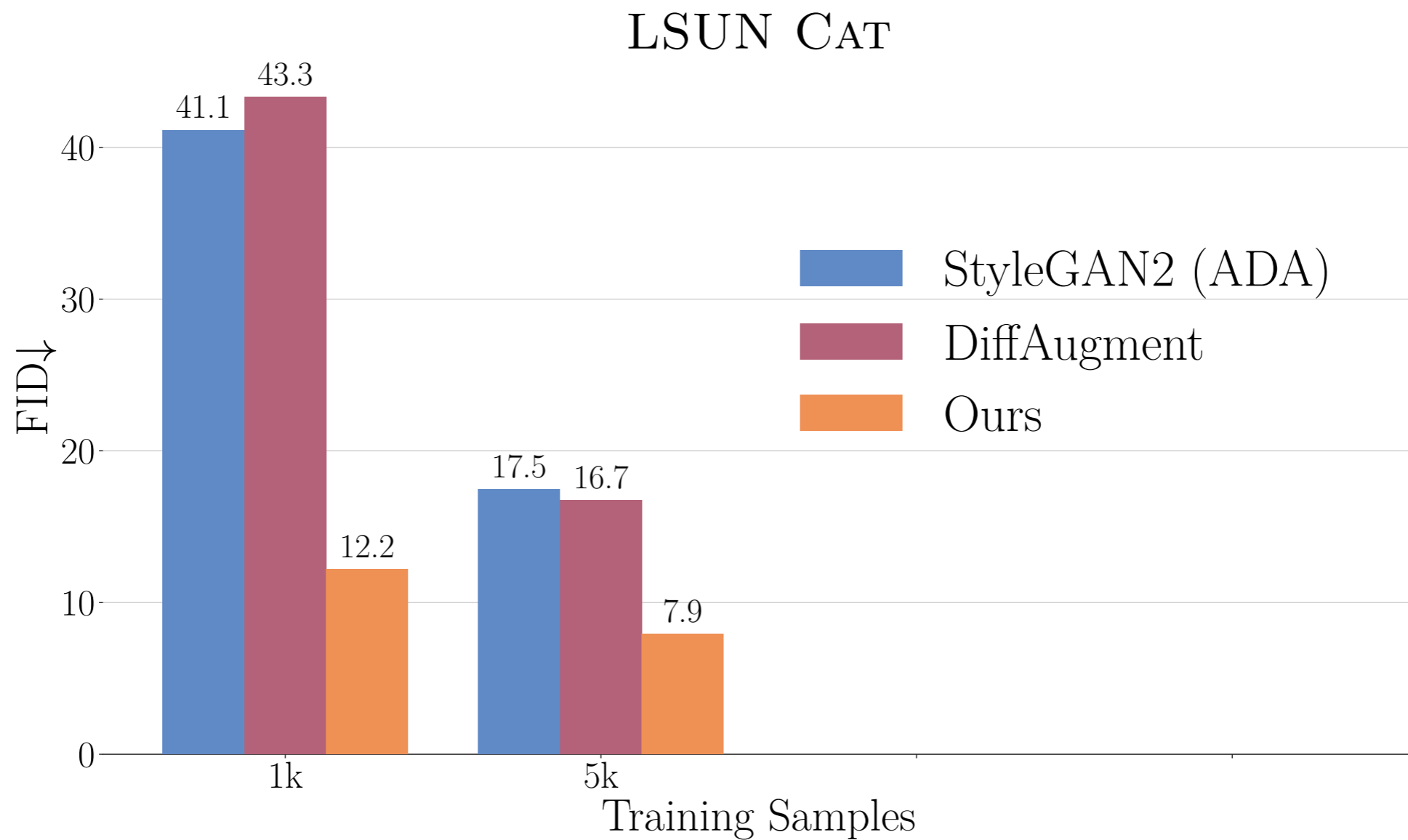
Benefit with varying training samples



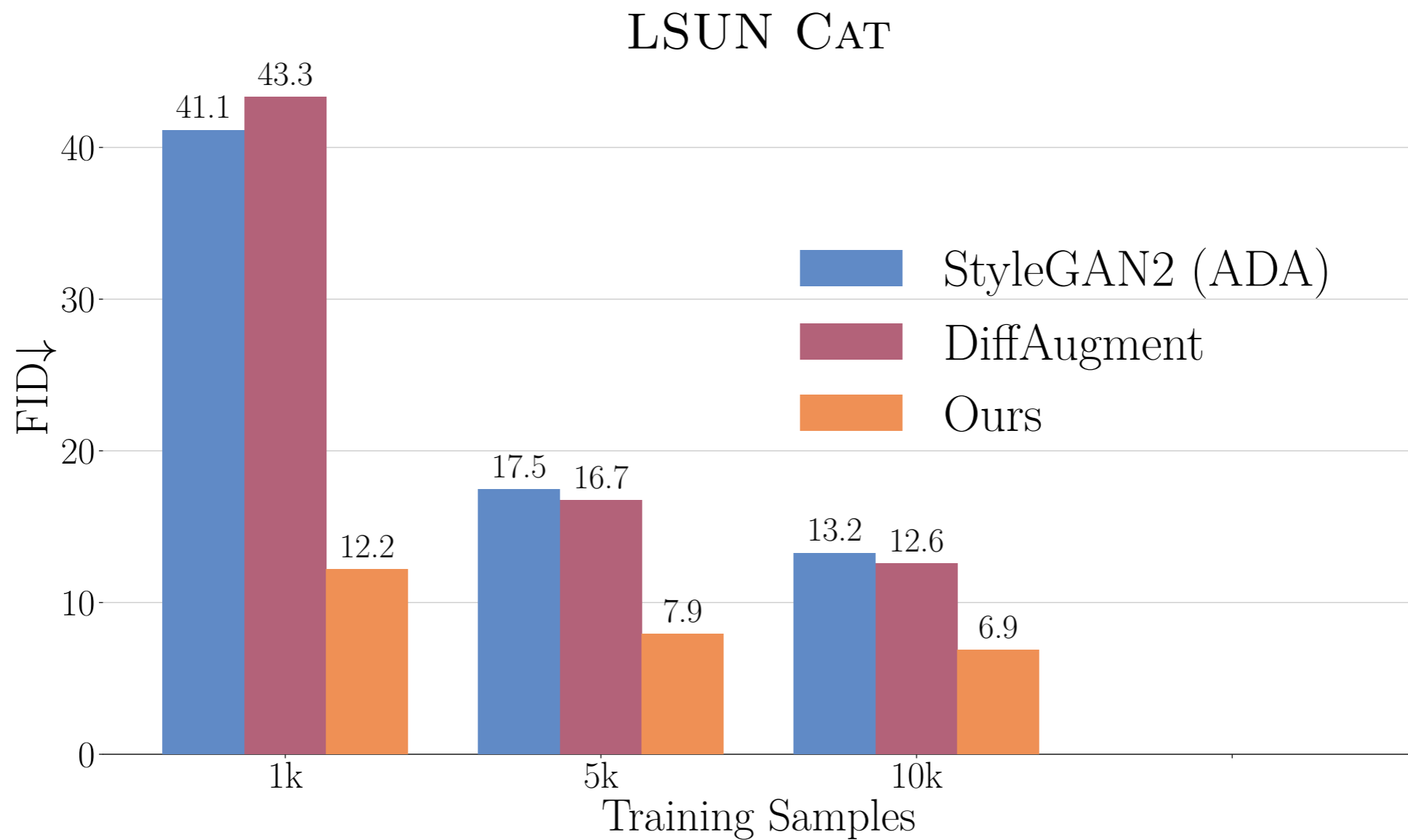
Benefit with varying training samples



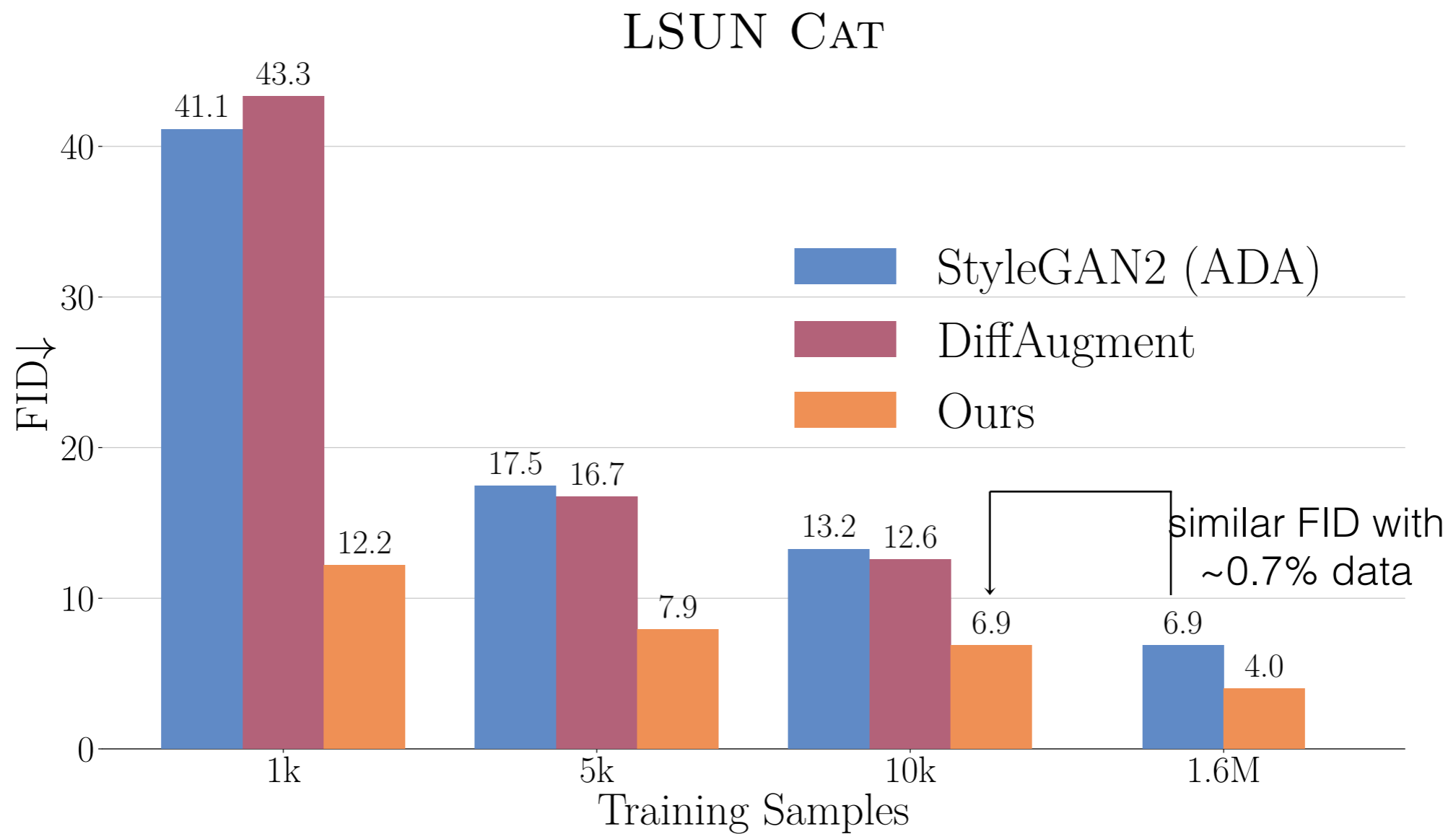
Benefit with varying training samples



Benefit with varying training samples



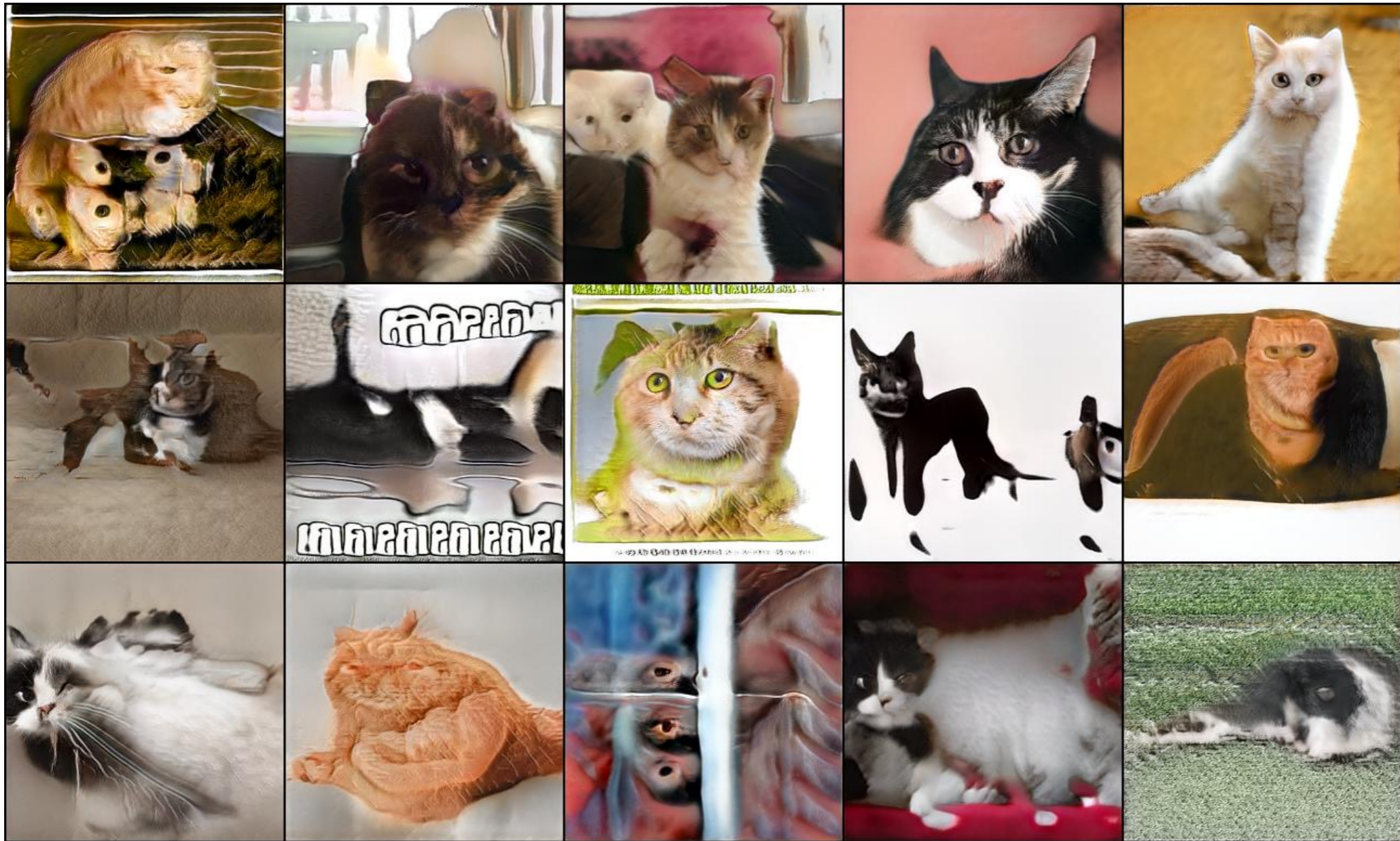
Benefit with varying training samples



StyleGAN2-ADA

LSUN CAT 1k

Improved Samples



Improved Samples

Ours

LSUN CAT 1k



Low-shot Generation with 100 samples

Bridge of Sighs

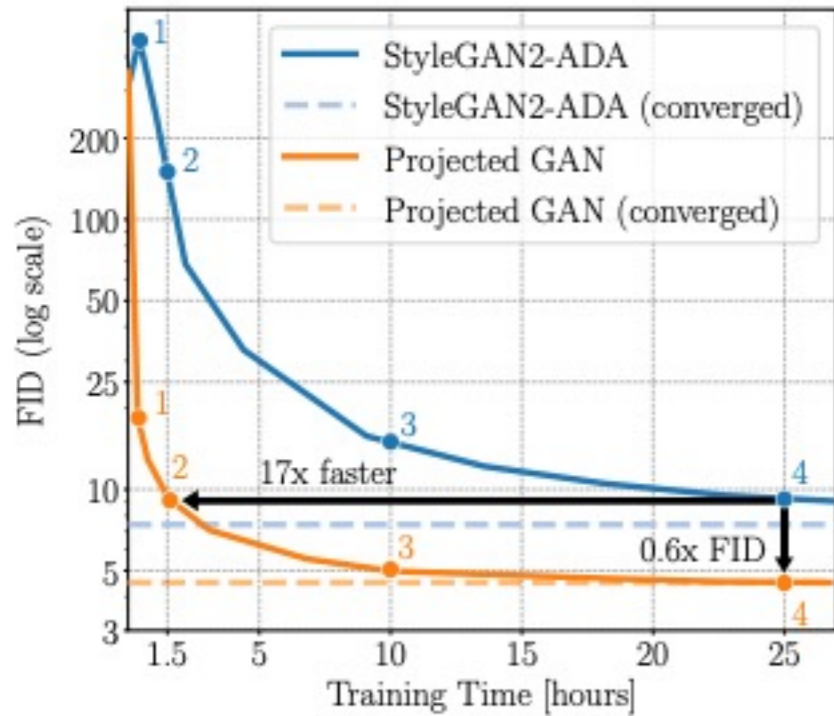


Low-shot Generation with 100 samples

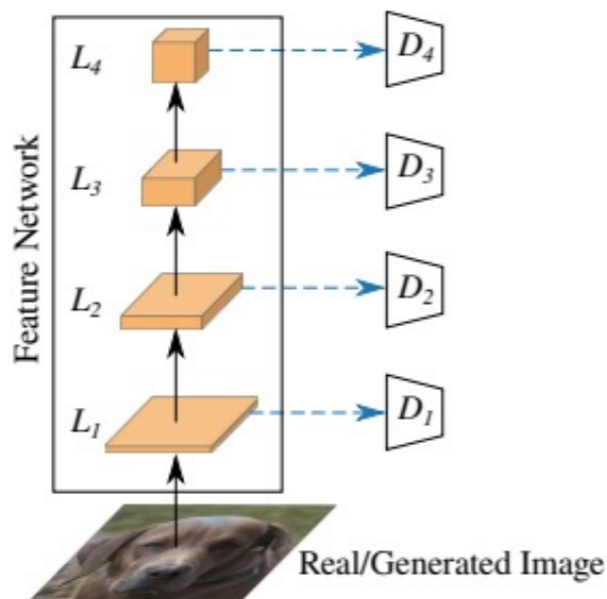
Bridge of Sighs



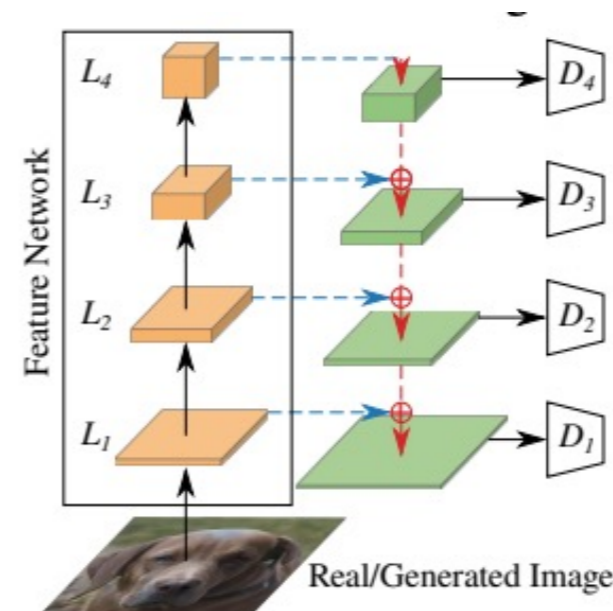
Faster Convergence with Projected GANs



Dashed blue arrows :
1x1 conv
with random weights



Dashed red arrows:
3x3 conv
with random weights



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

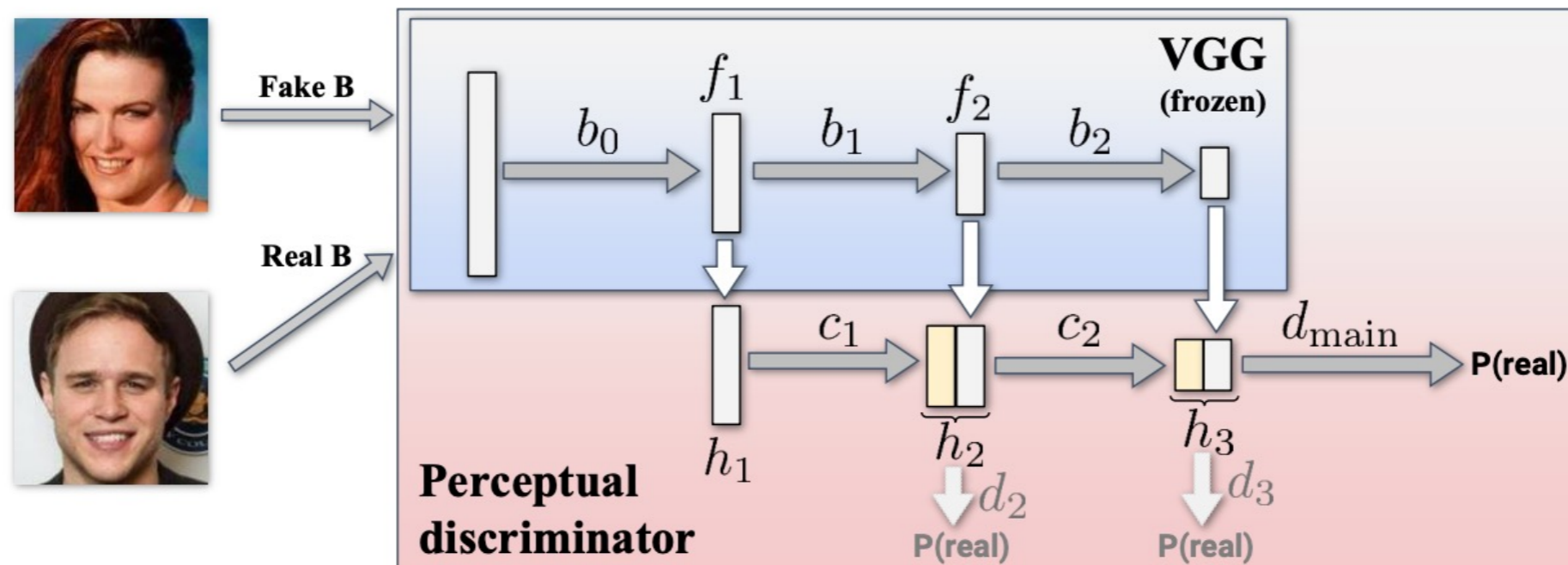


Image Manipulation with Perceptual Discriminators [Sungatullina et al. ECCV 2018]

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?

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

Thank You!



16-726, Spring 2023

<https://learning-image-synthesis.github.io/>