

Deep Learning II: Interpretability & Adversarial Methods

8DM50 Machine Learning in Medical Imaging and Biology

Jelmer Wolterink

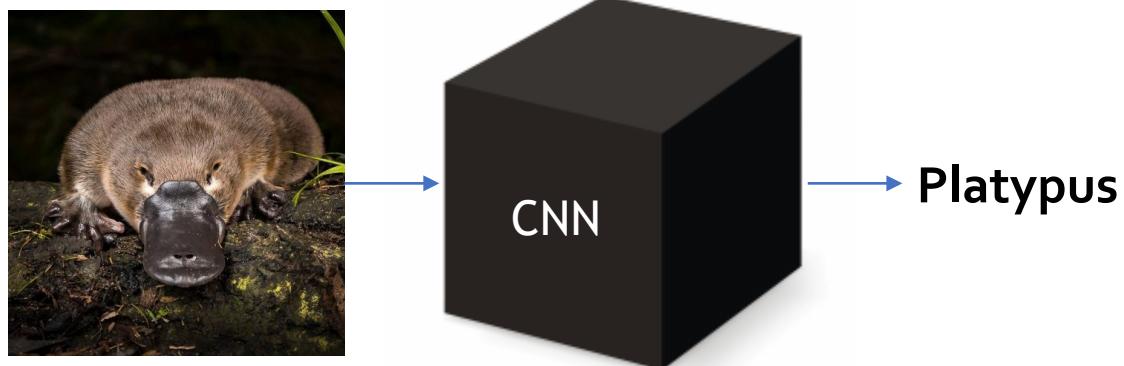
28-09-2020

Deep Learning II

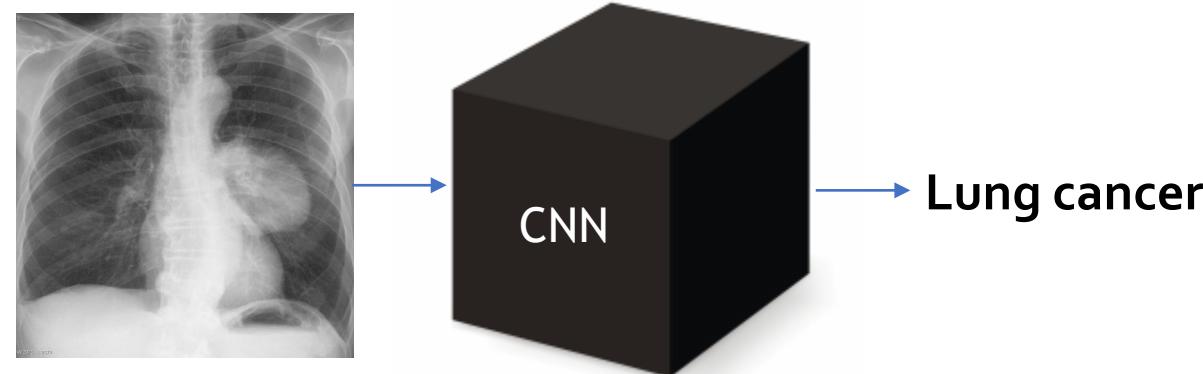
Today

1. Advanced neural network architectures
2. Interpretability and generative adversarial networks
3. Practical assignment in Keras

Machine learning as a black box



Machine learning as a black box



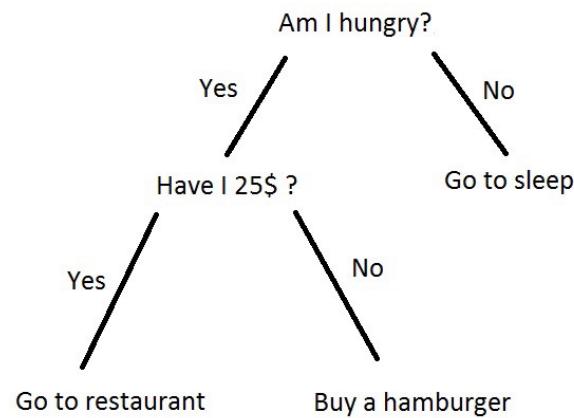
Interpretability: Can we predict what the model will do if the input changes?

Explainability: Do we know exactly what the model is doing?

Uncertainty: How sure is the network of its prediction?

Important questions for ethical use and liability

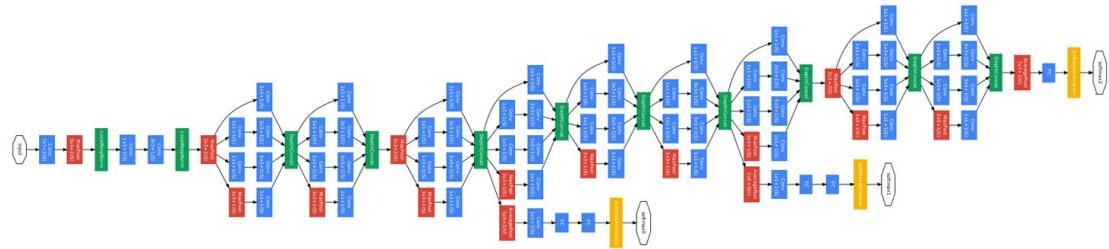
Complexity/performance trade-off



Decision tree

Explainable/interpretable

Performance



CNN

Explainable/interpretable

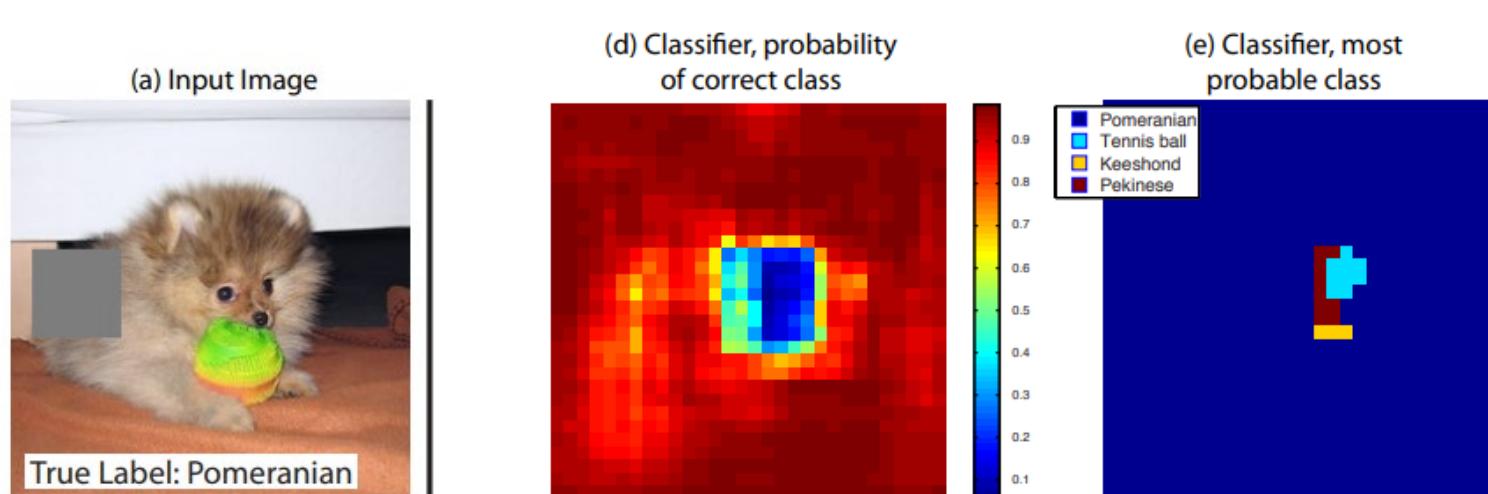
Performance

Interpreting CNNs: saliency maps

- Millions of parameters, impossible to know in detail what each parameter does
- Substitute: what part of the input is associated with a CNN feature or output?
- Popular approaches
 - Occlusion
 - Deconvolution
 - Class activation mapping

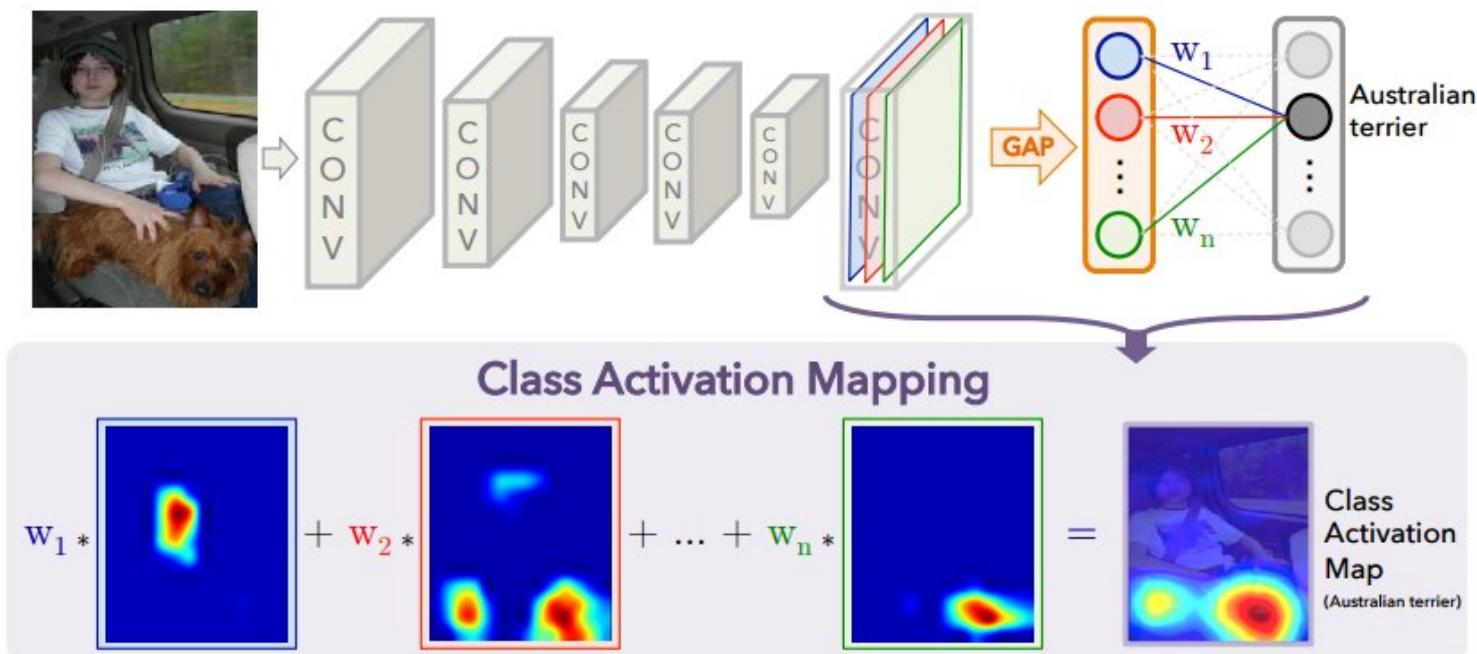
Occlusion

1. Move grey square over image (a)
2. See how classifier output for correct class changes (d)



Class activation maps (CAM)

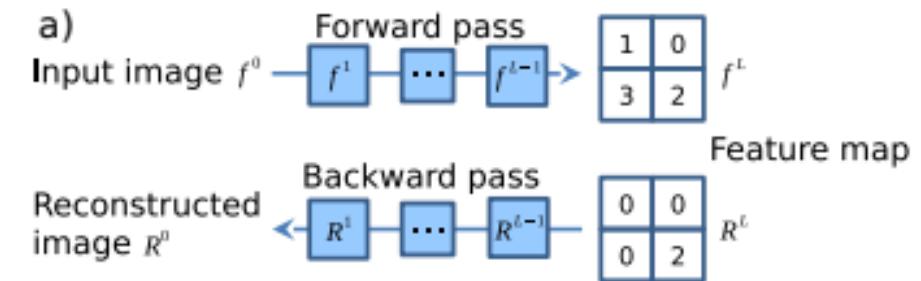
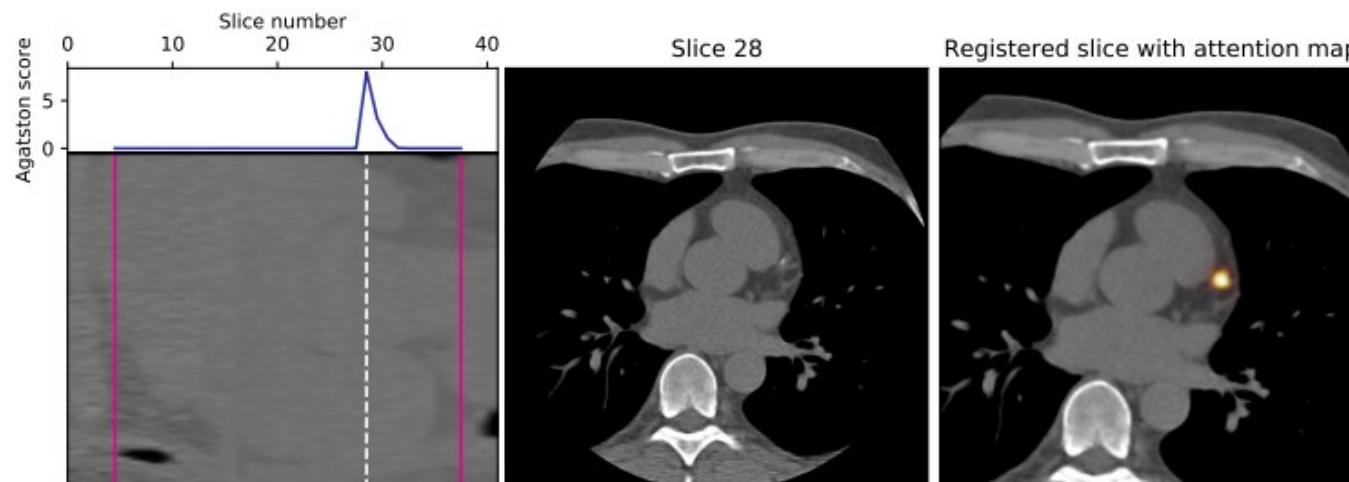
- Global average pooling: simply compute average of whole feature map
- Feature maps represent heat maps, weighting gives evidence



Deconvolution

Inversion: find input corresponding to a particular feature

1. Push input through CNN
2. Fix value for one ‘neuron’, set all others to zero
3. Perform inverse of convolution, pooling, activation functions



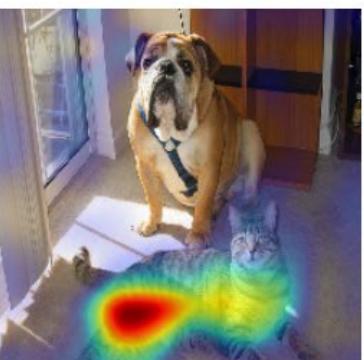
Comparison



(a) Original Image



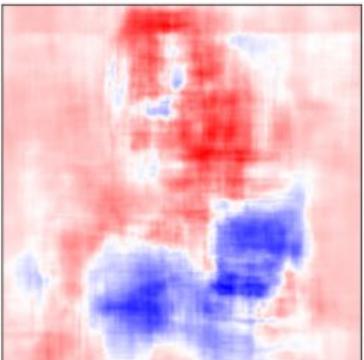
(b) Guided Backprop ‘Cat’



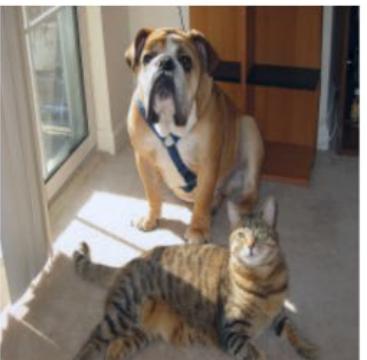
(c) Grad-CAM ‘Cat’



(d) Guided Grad-CAM ‘Cat’



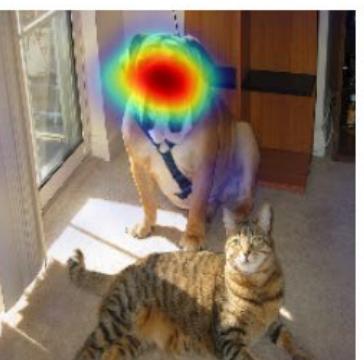
(e) Occlusion map for ‘Cat’



(g) Original Image



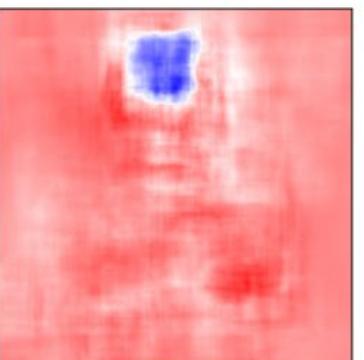
(h) Guided Backprop ‘Dog’



(i) Grad-CAM ‘Dog’



(j) Guided Grad-CAM ‘Dog’



(k) Occlusion map for ‘Dog’ (

Discriminative vs. generative models

Discriminative models

- Go from high-dimensional sample to low-dimensional prediction
- E.g. from an image to the label of that image



Container ship

Generative models

- Go from low-dimensional input to high-dimensional sample
- E.g. from an image label to an image sample

Container ship



But why would we do this?

- **Unsupervised learning:** we don't need labels for our training data
- Learning more about your data set
- Synthesizing new data for discriminative models
- Advanced image manipulation

Generative adversarial networks (GANs)

Instead of **one** neural network, we train **two** neural networks

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples

The **generator** network tries to fool the discriminator



Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples



Generator
Counterfeiter



Discriminator
Detective

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples



Generator
Counterfeiter



Discriminator
Detective



UNIVERSITY
OF TWENTE.

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples



Generator
Counterfeiter



Discriminator
Detective



UNIVERSITY
OF TWENTE.

Generative adversarial networks (GANs)

1. A **generator** network generates samples of images
2. A **discriminator** network distinguishes generated samples from real samples



Generator
Counterfeiter



Discriminator
Detective



UNIVERSITY
OF TWENTE.

Generative adversarial networks (GANs)

A bit more formally...

We are given samples x_1, \dots, x_n from a data distribution p_{data}

We want to generate new samples similar to p_{data}

We can sample from p_{data} efficiently – examples are available



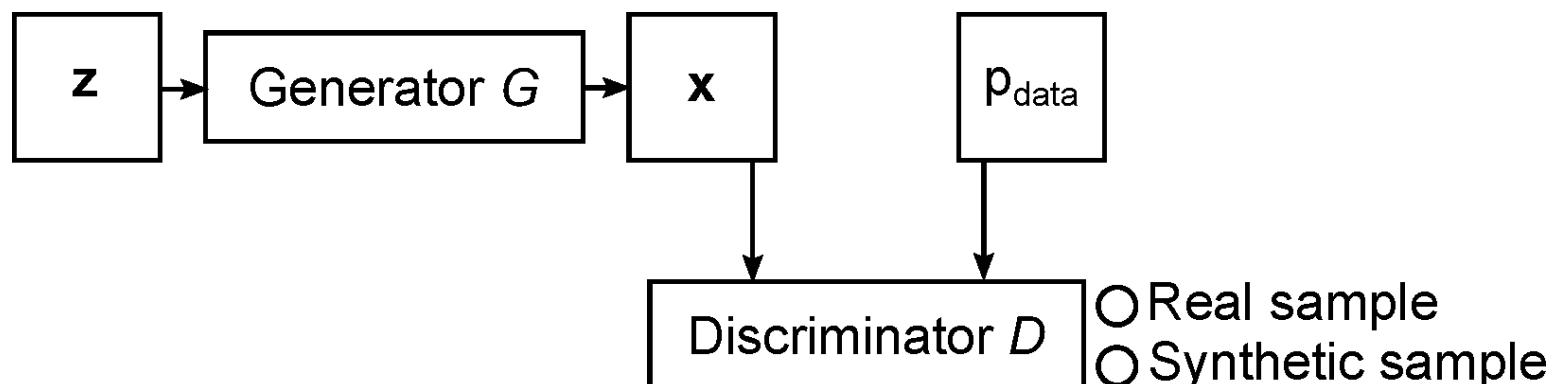
Generative adversarial networks (GANs)

Aim: Find a sample distribution p_{gen} that closely resembles p_{data}

Generator G maps points from a normal/uniform distribution z to samples x in p_{gen}

Discriminator D tells samples from p_{data} and p_{gen} apart

Generators G and D are networks with trainable parameters ϑ_G and ϑ_D



Generative adversarial networks (GANs)

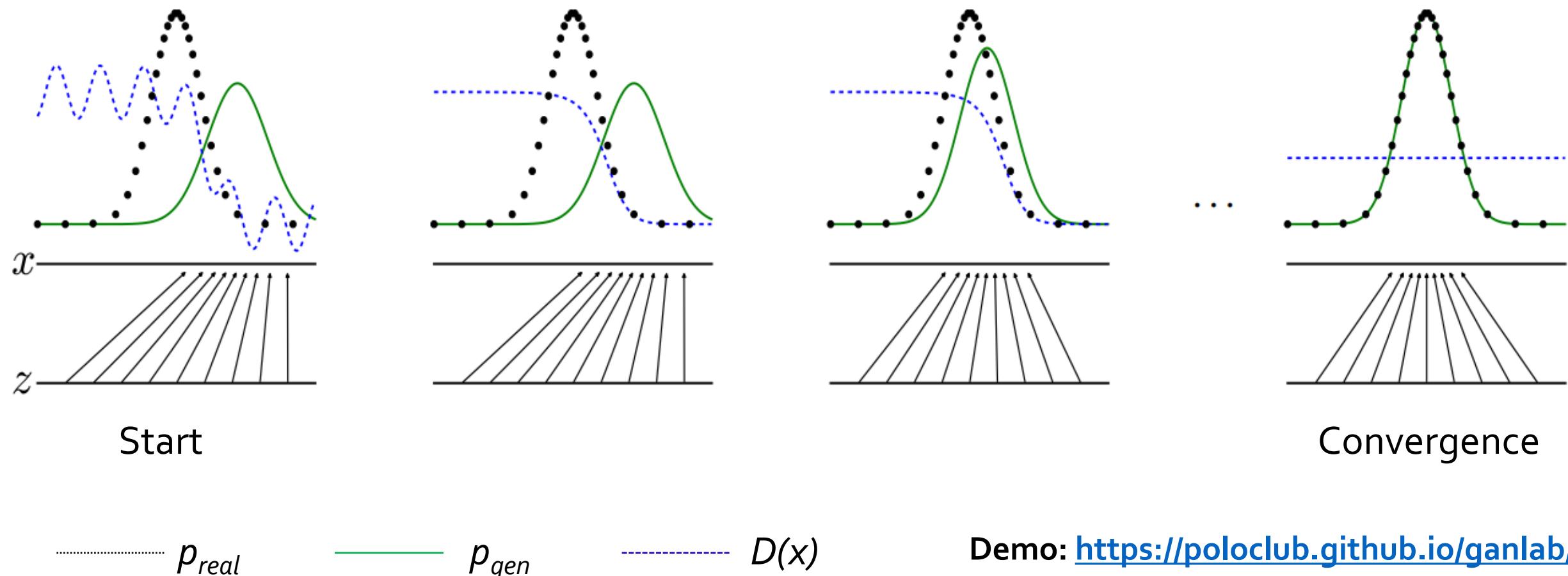
Discriminator D maximizes an **objective function** by assigning a high **probability to real samples** and a low **probability to synthetic samples**

$$V^{(D)}(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

At the same time, G tries to minimize this **objective function** by generating samples that have a high **probability of being real**

$$V^{(G)}(D, G) = \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

GANs in theory



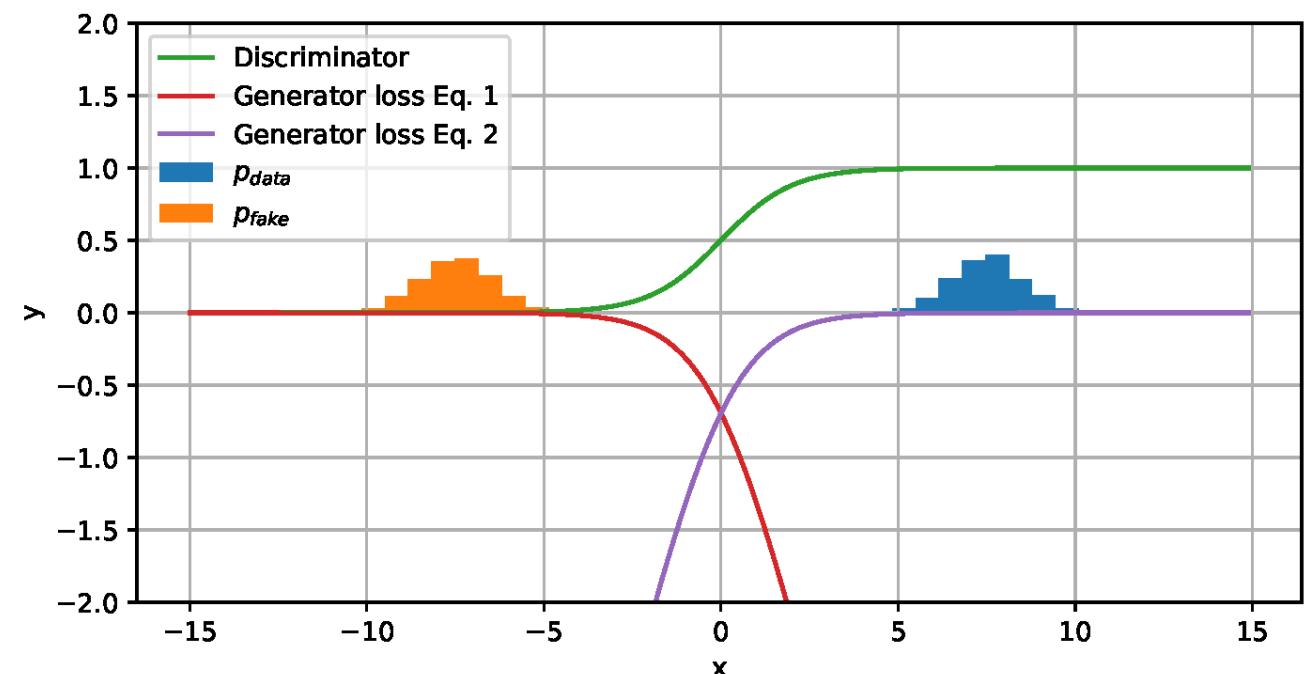
Challenges when training GANs

Poor gradients from discriminator D to generator G

Especially when the discriminator D can easily distinguish fake and real samples

$$1) \quad V^{(G)}(D, G) = \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

$$2) \quad V^{(G)}(D, G) = - \mathbb{E}_{z \sim p_z} [\log (D(G(z)))]$$



Challenges when training GANs

1. Poor gradients from discriminator D to generator G

- Especially when the discriminator D can easily distinguish fake and real samples

2. Mode collapse

- Generator G maps all noise vectors to similar samples $G(z)$
- E.g. generator only synthesizes 1 in MNIST
- Generator and discriminator chase each other from mode to mode

3. Unclear when training is finished

Synthesizing MNIST

Generator

z 10

256

512

1024

x $28 \times 28 = 784$

Discriminator

1024

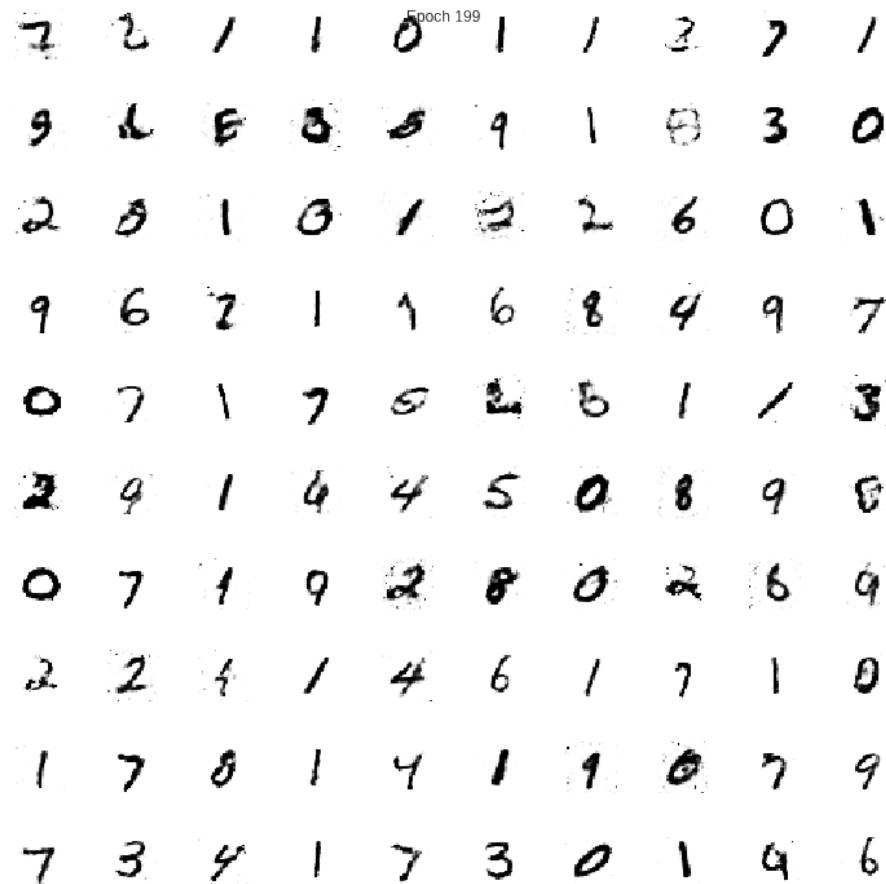
512

256

$D(x)$

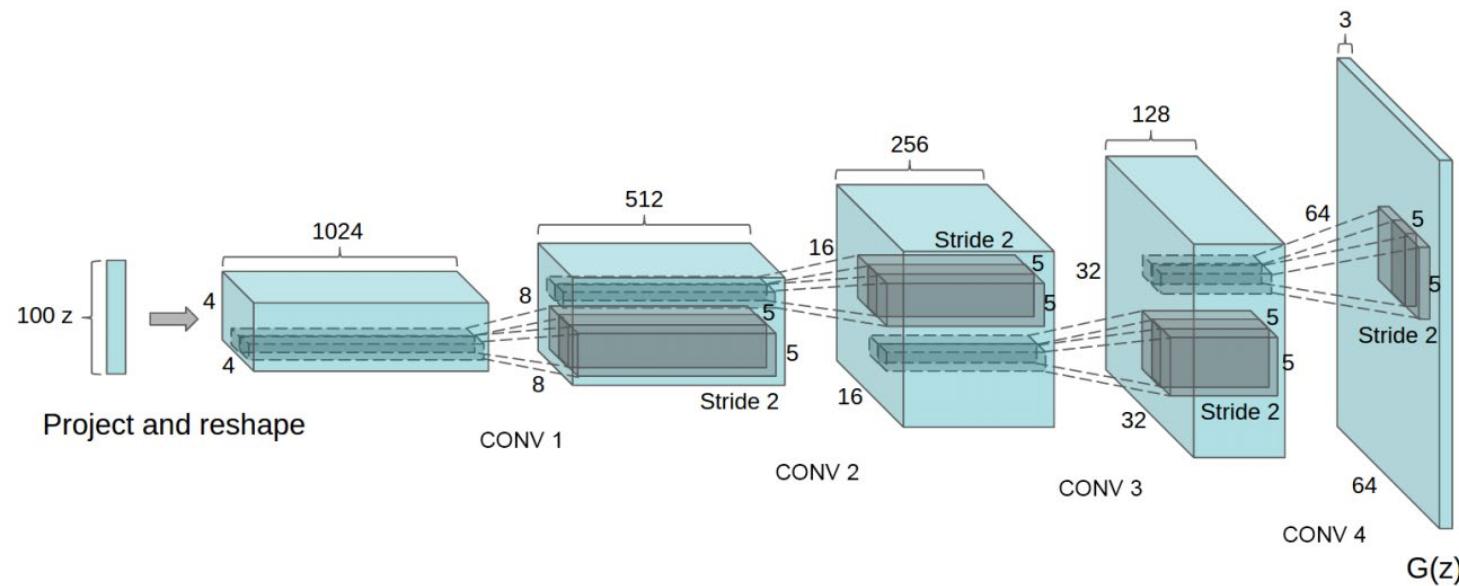
1

Synthesizing MNIST

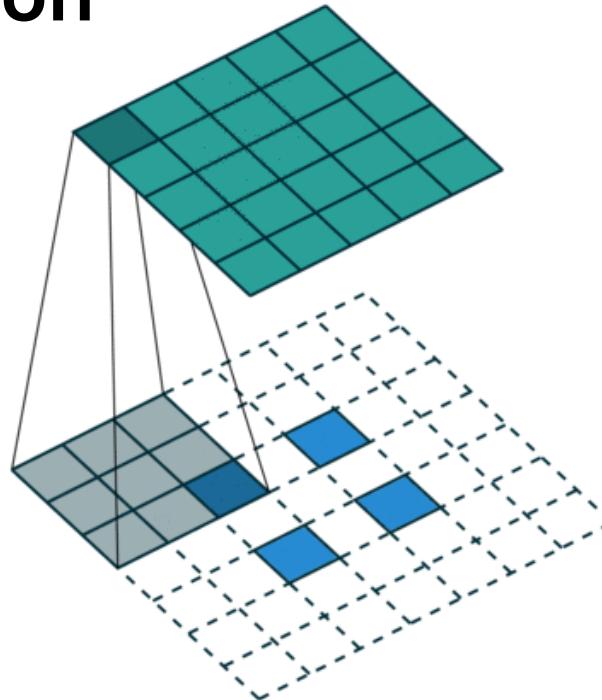


Synthesizing images

- Image synthesis requires convolutional generators and discriminators
- We know how to reduce the image size
- Now we need some way to increase the image size



Transposed convolution



- Deconvolution
- Fractionally-strided convolution
- Transposed convolution



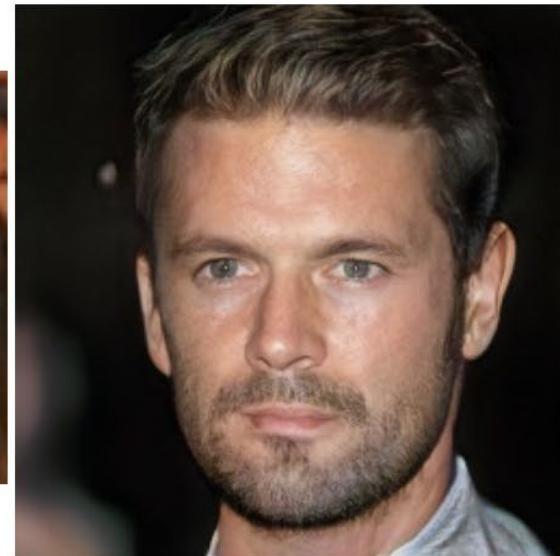
2014



2015



2016



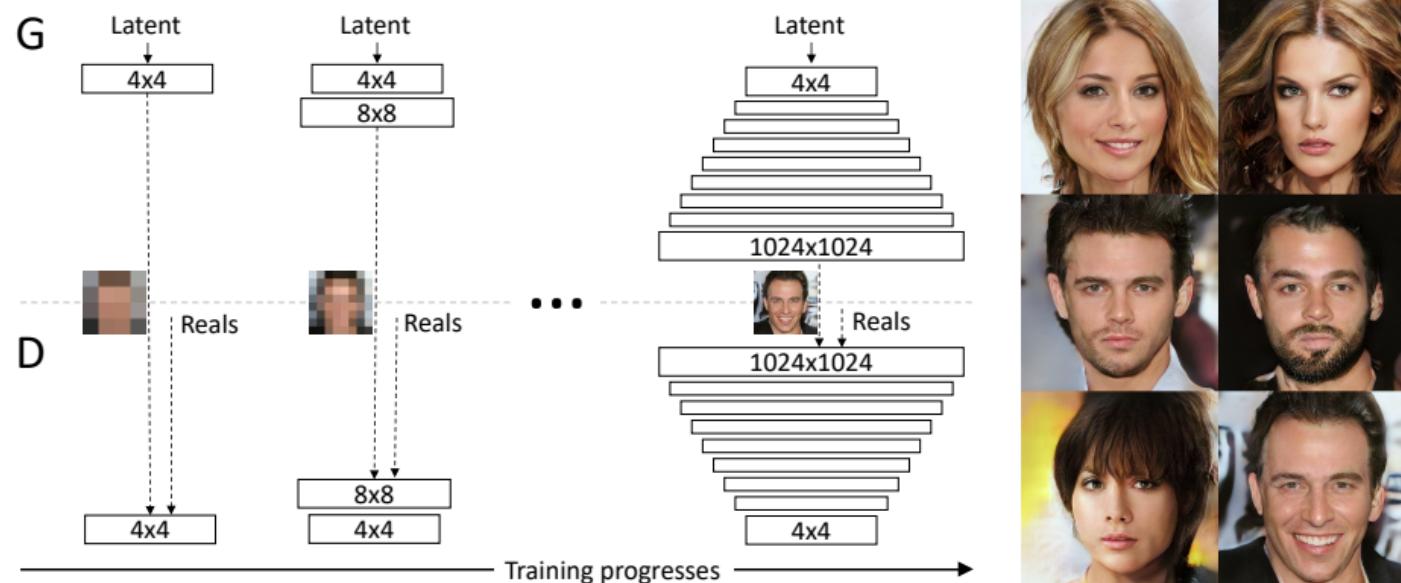
2017



2018

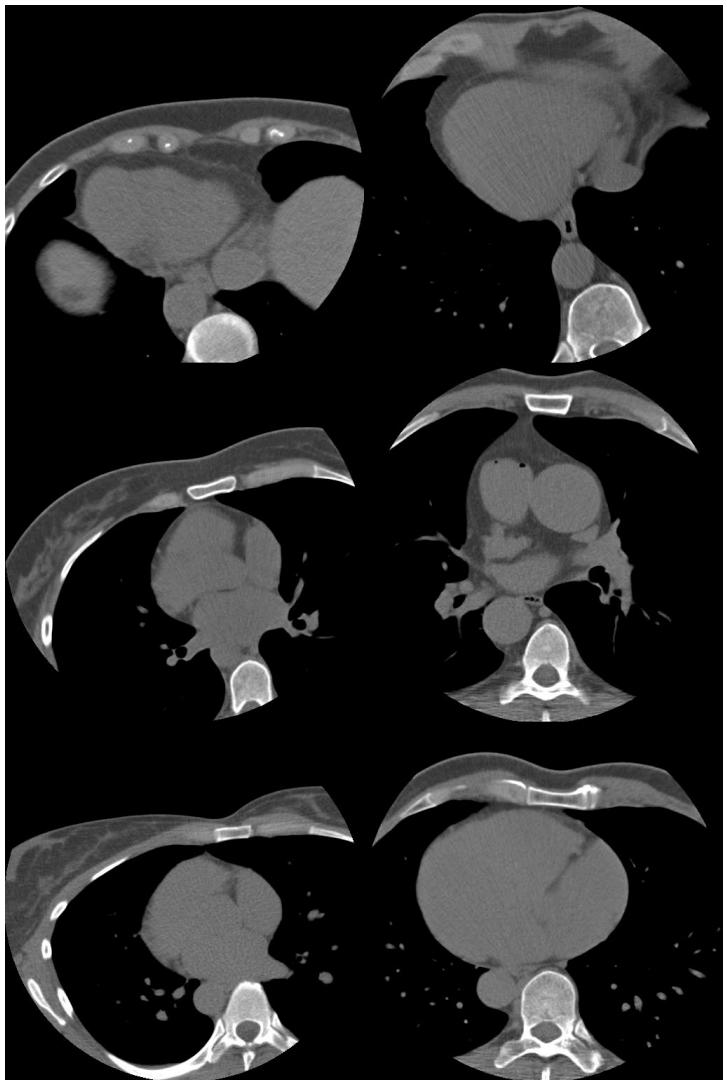
Progressive GAN

- Goal: high-resolution image synthesis (1024×1024 pixels)
- Problem: optimizing large and deep networks can lead to stability issues
- Trick: periodically increase resolution in G and D

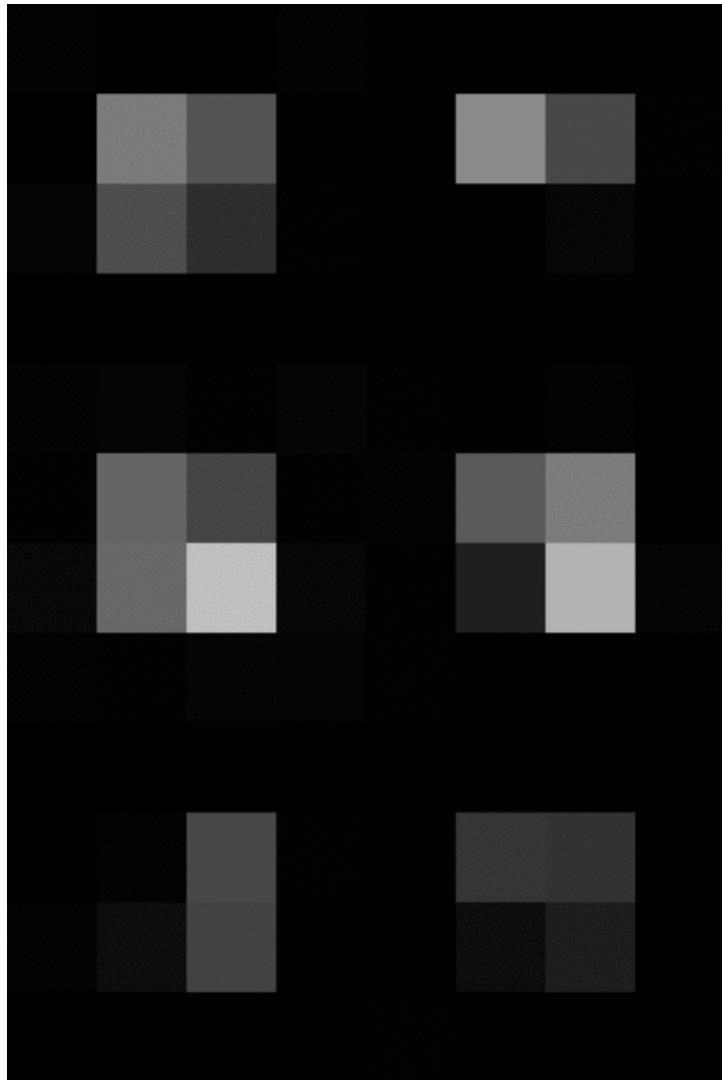


UNIVERSITY
OF TWENTE.

Real



Fake



UNIVERSITY
OF TWENTE.

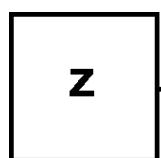
StyleGAN



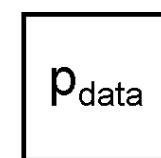
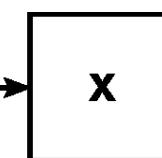
Spaces

- GANs can help us learn more about our data
- Different latent space points correspond to different sample space points
- The latent space is structured
- Interpolation in the latent space leads to smooth transitions

The latent space



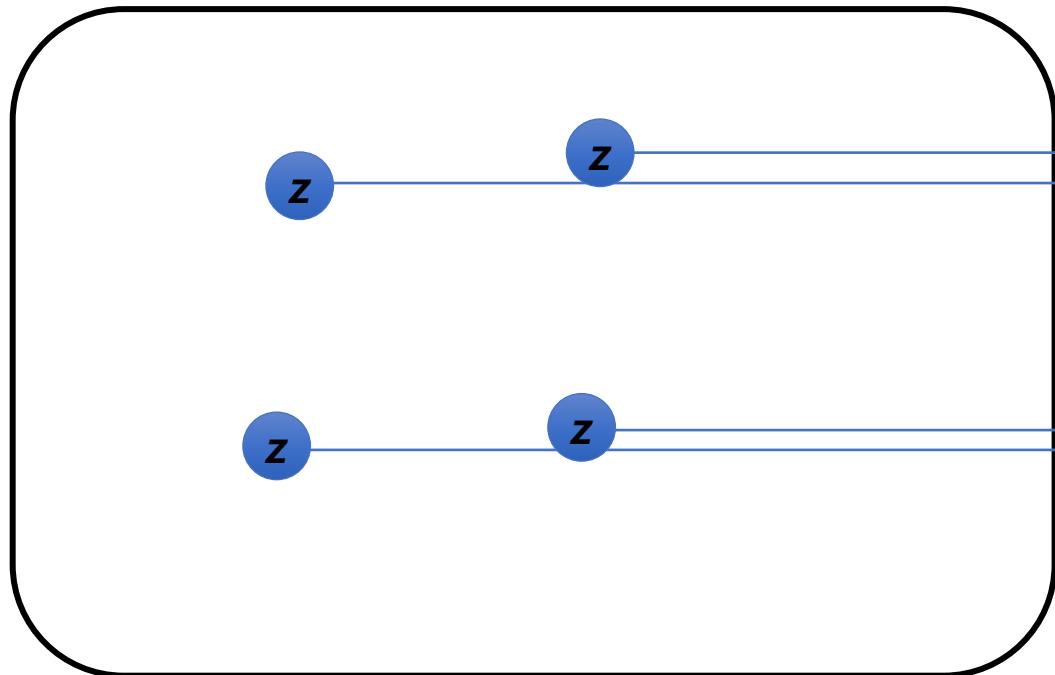
The sample space



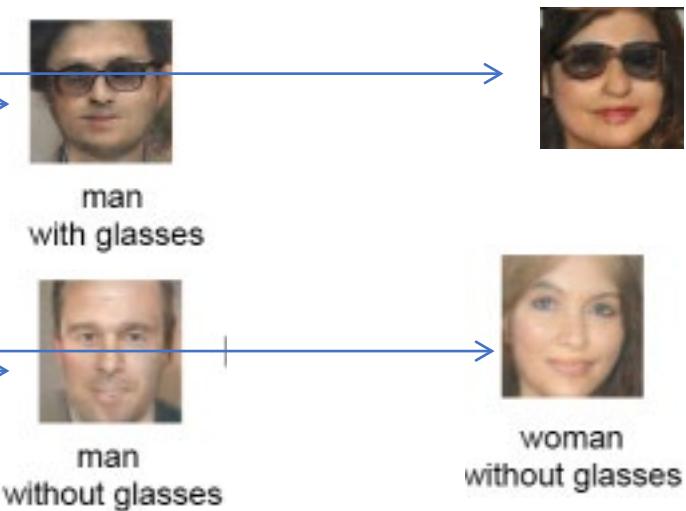
○ Real sample
○ Synthetic sample

Spaces

The latent space

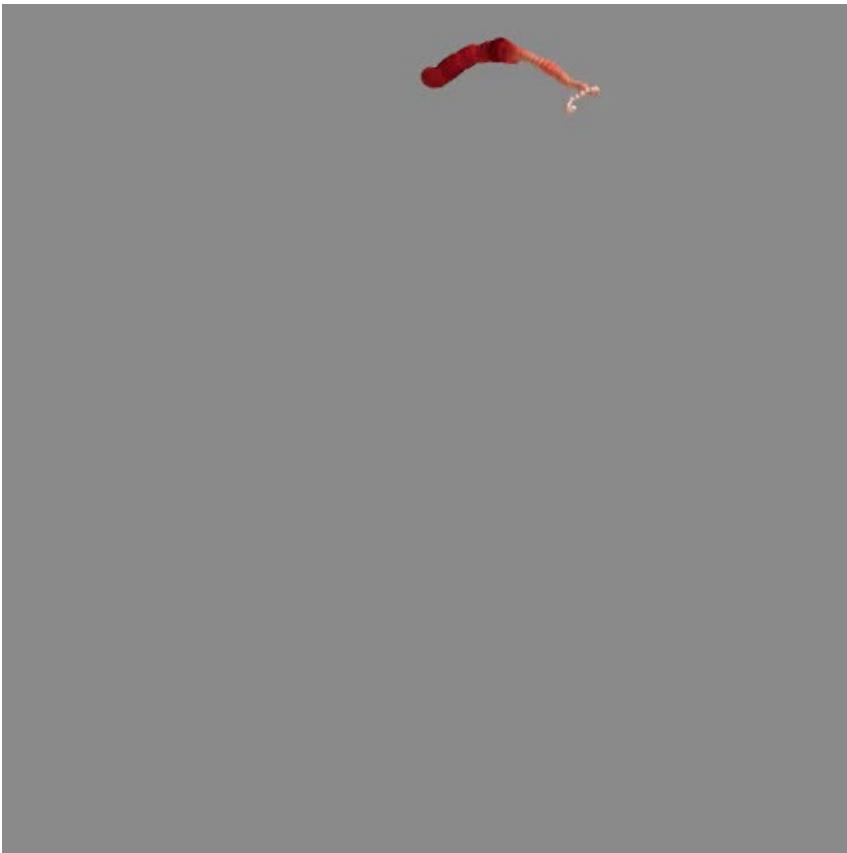


The sample space

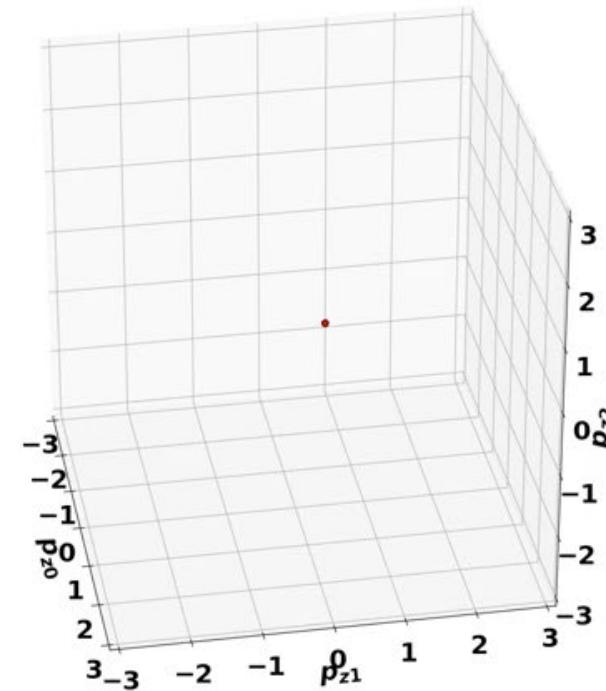


Example: coronary artery synthesis

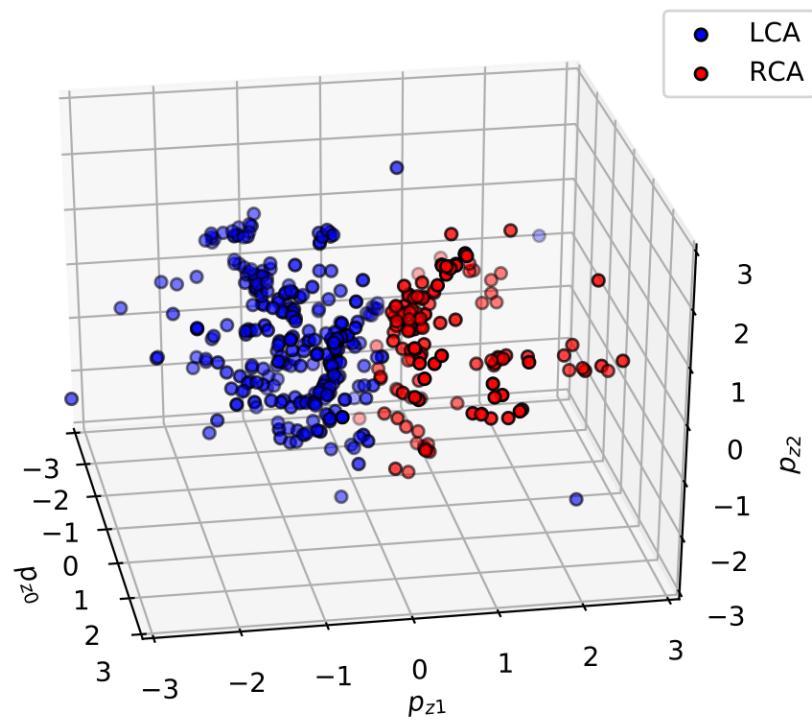
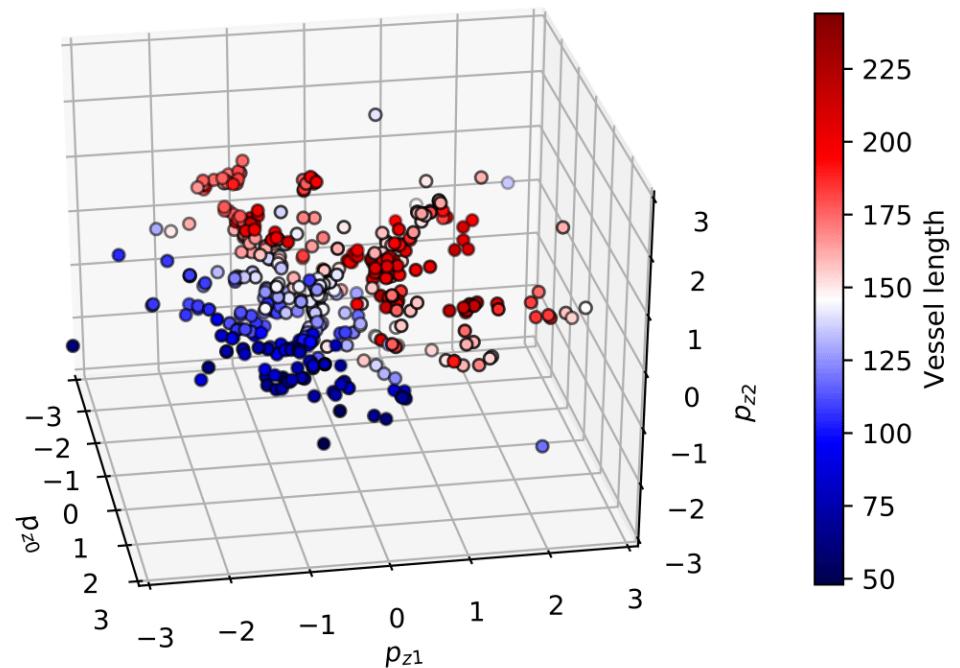
The sample space



The latent space



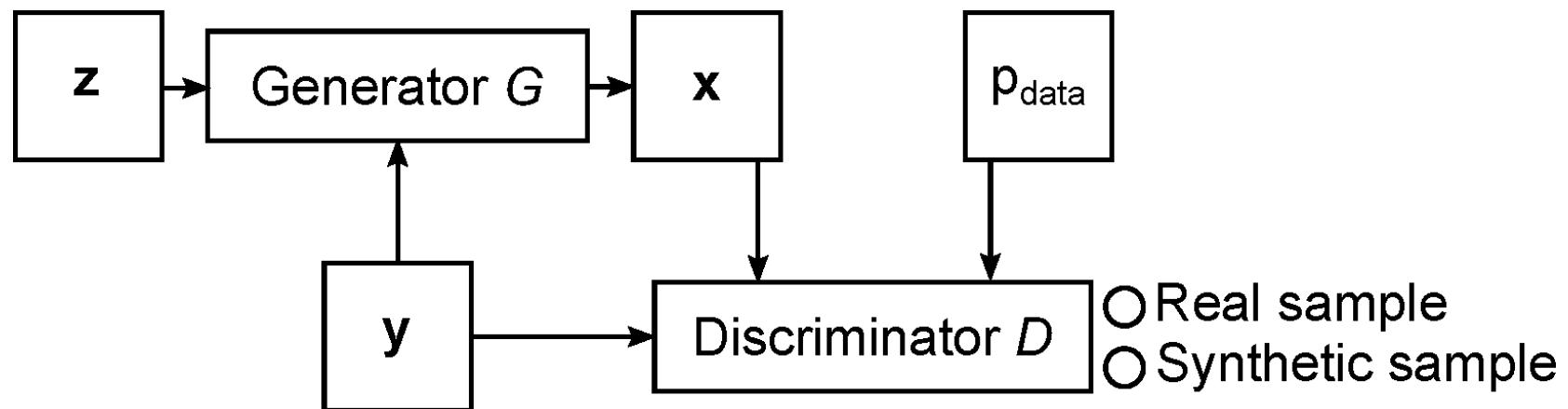
Example: coronary artery synthesis



UNIVERSITY
OF TWENTE.

Conditional GAN (cGAN)

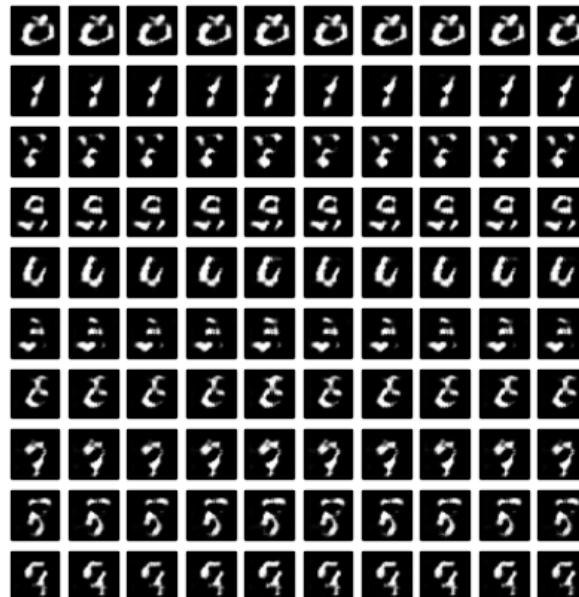
- GAN: difficult to control what is generated
- Condition GAN on extra input y



Conditional GAN (cGAN)

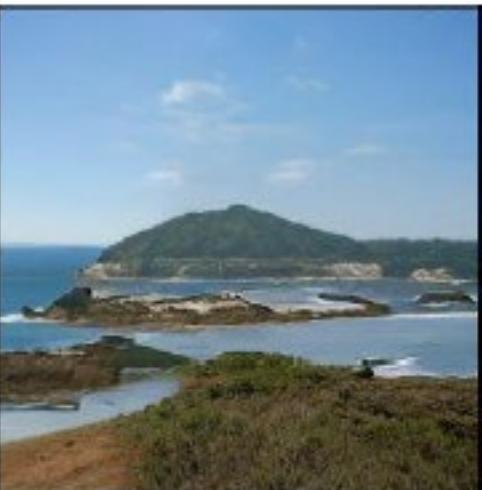
The generator and discriminator optimize an **objective function** based on the predictions for **real samples** and **synthetic samples**, given **extra information**

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}} [\log D(x|y)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z|y)|y))]$$



BigGAN

- State of the art in conditional GAN training
- Trained on ImageNet (~million images)
- Get any class you want



UNIVERSITY
OF TWENTE.

In medical imaging

We typically have some image to condition on, 'image-to-image' applications

- Segmentation
- Synthesis
- Image quality enhancement

It's not always easy to determine a loss function that captures what we want

- Noise-free segmentations?
- Sharp images?

Use a conditional GAN where y is the input image

Image-to-image translation

Domain A: input image y

Domain B: output image x

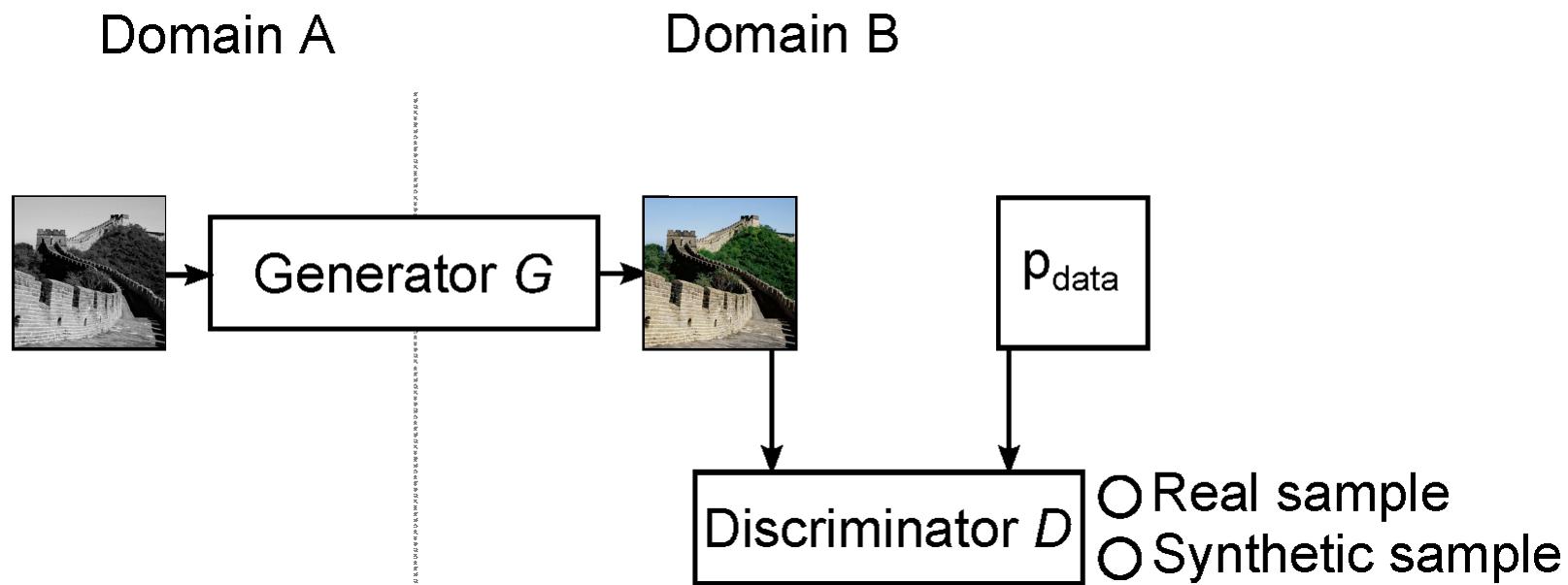
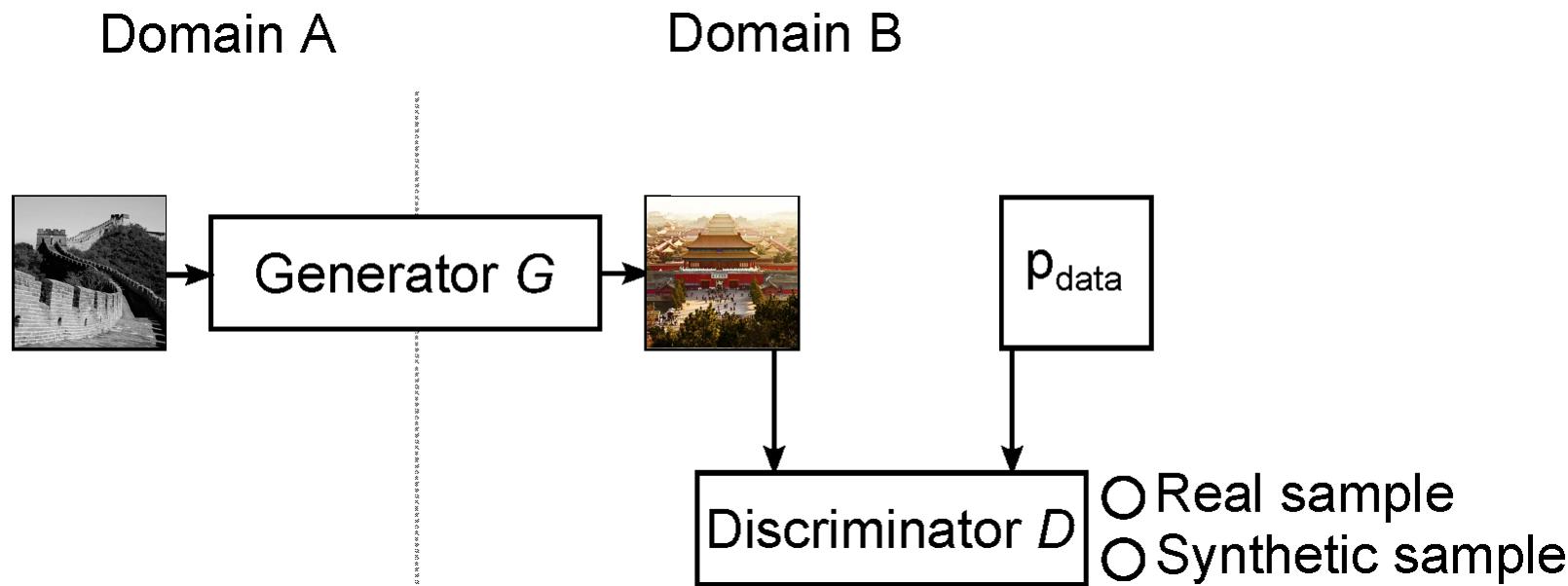


Image-to-image translation

The output image x could be very realistic but 'wrong'

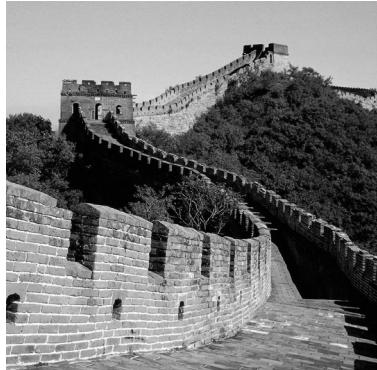
Regularize to make sure that it corresponds to the input image y



Paired vs. unpaired training data

Paired

Domain A



Domain B



Unpaired

Domain A



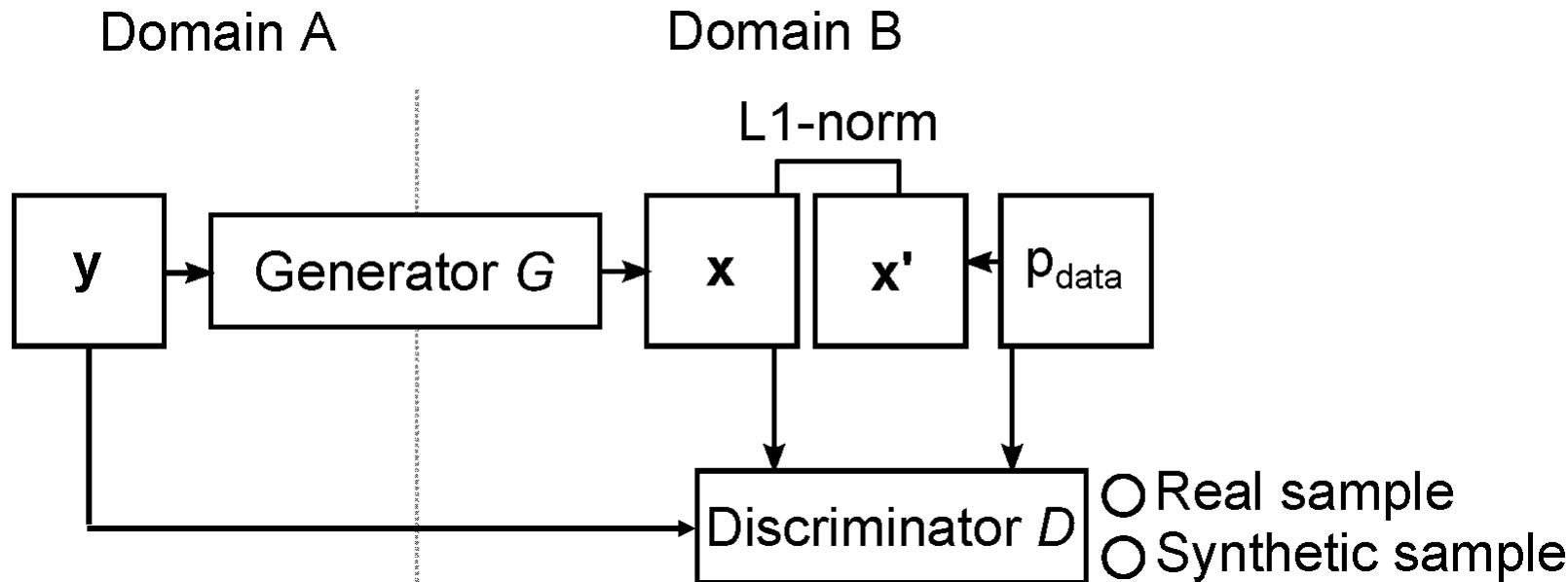
Domain B



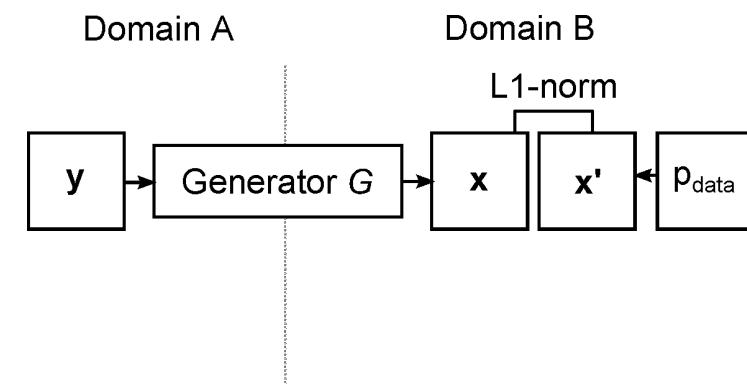
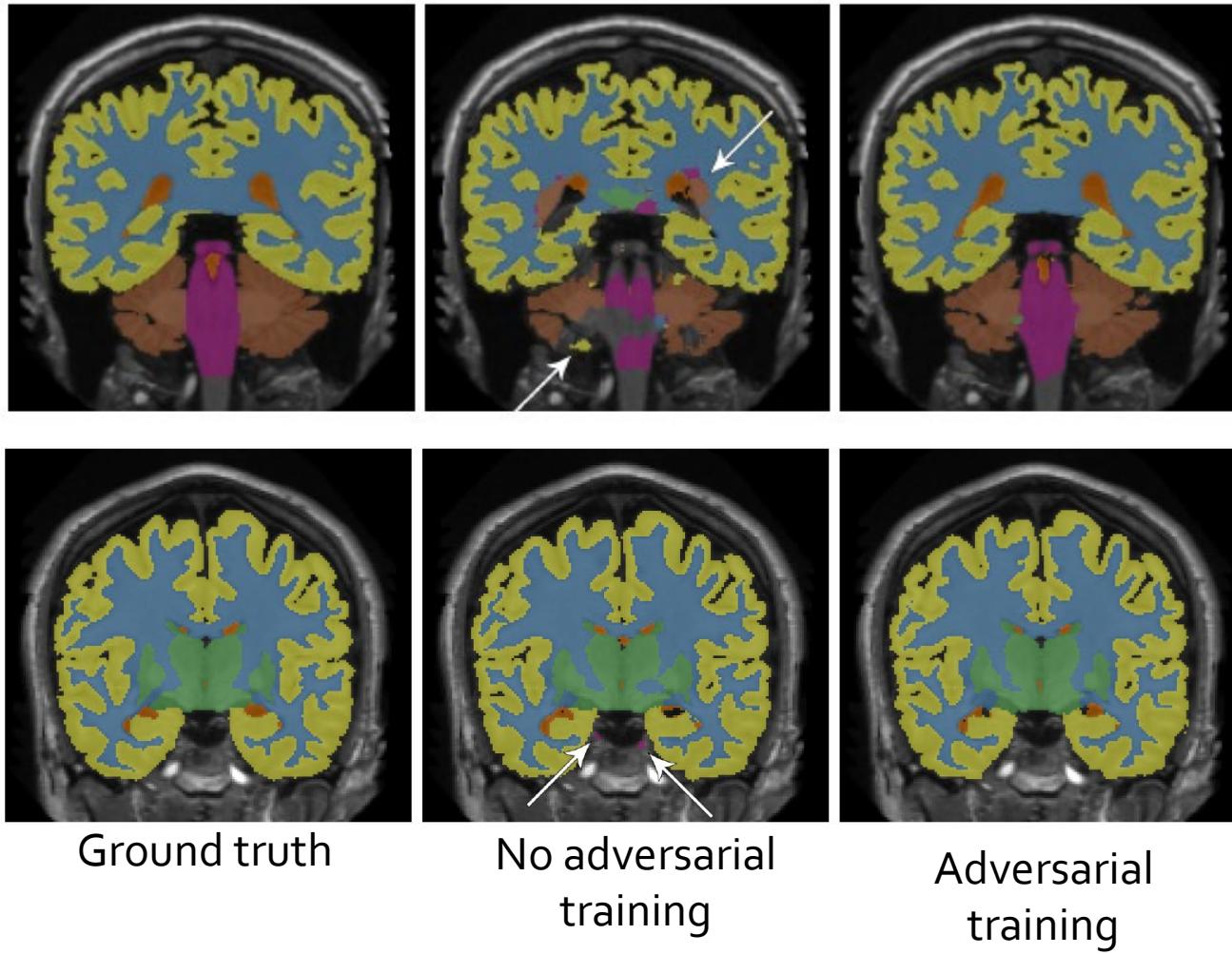
Paired image-to-image translation

The discriminator and generator optimize an **objective function** based on predictions for **real** and **synthetic** images in domain B, given input images in domain A while minimizing the **difference between real and synthetic images in domain B**

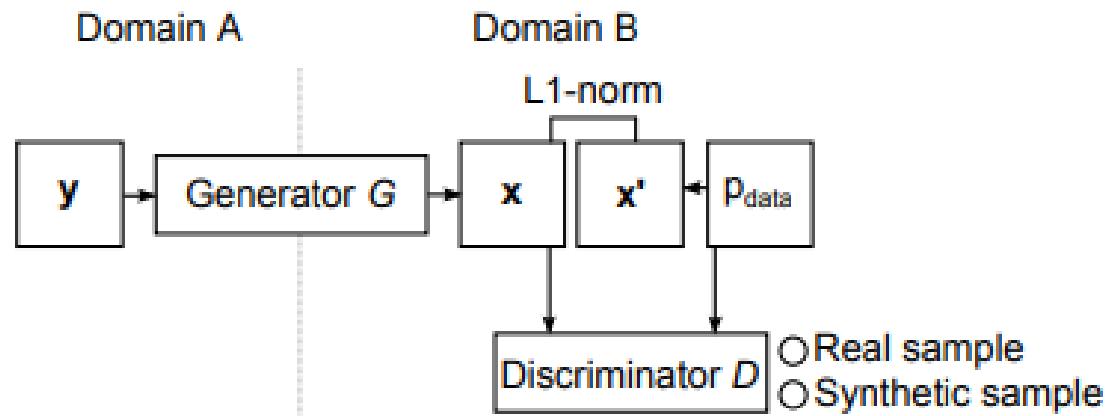
$$\min_G \max_D V(D, G) = \mathbb{E}_{x, y \sim p_{data_{B,A}}} [\log D(x, y)] + \mathbb{E}_{y \sim p_{data_A}} [\log (1 - D(G(y), y))] + \lambda \mathbb{E}_{x, y \sim p_{data_{B,A}}} [|x - G(y)|]_1$$



Brain MR segmentation



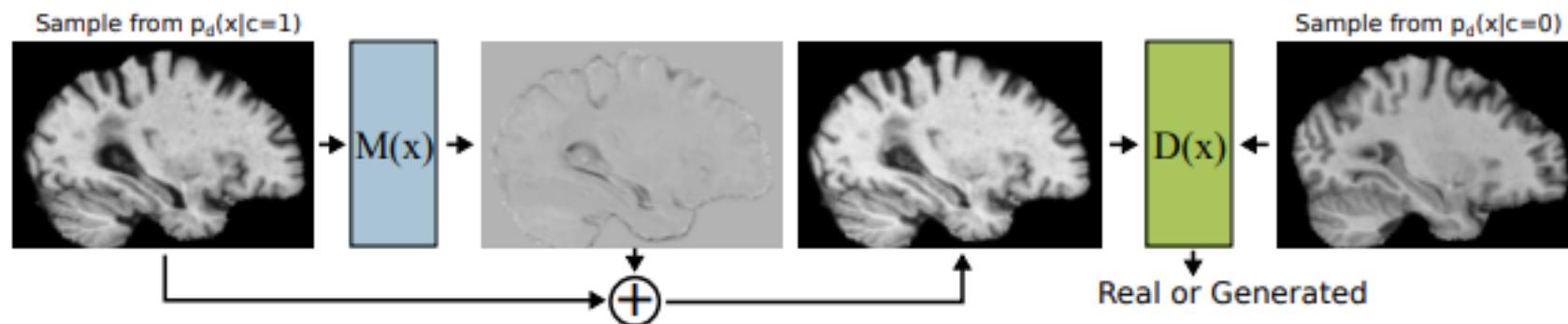
Unpaired training: self-regularization



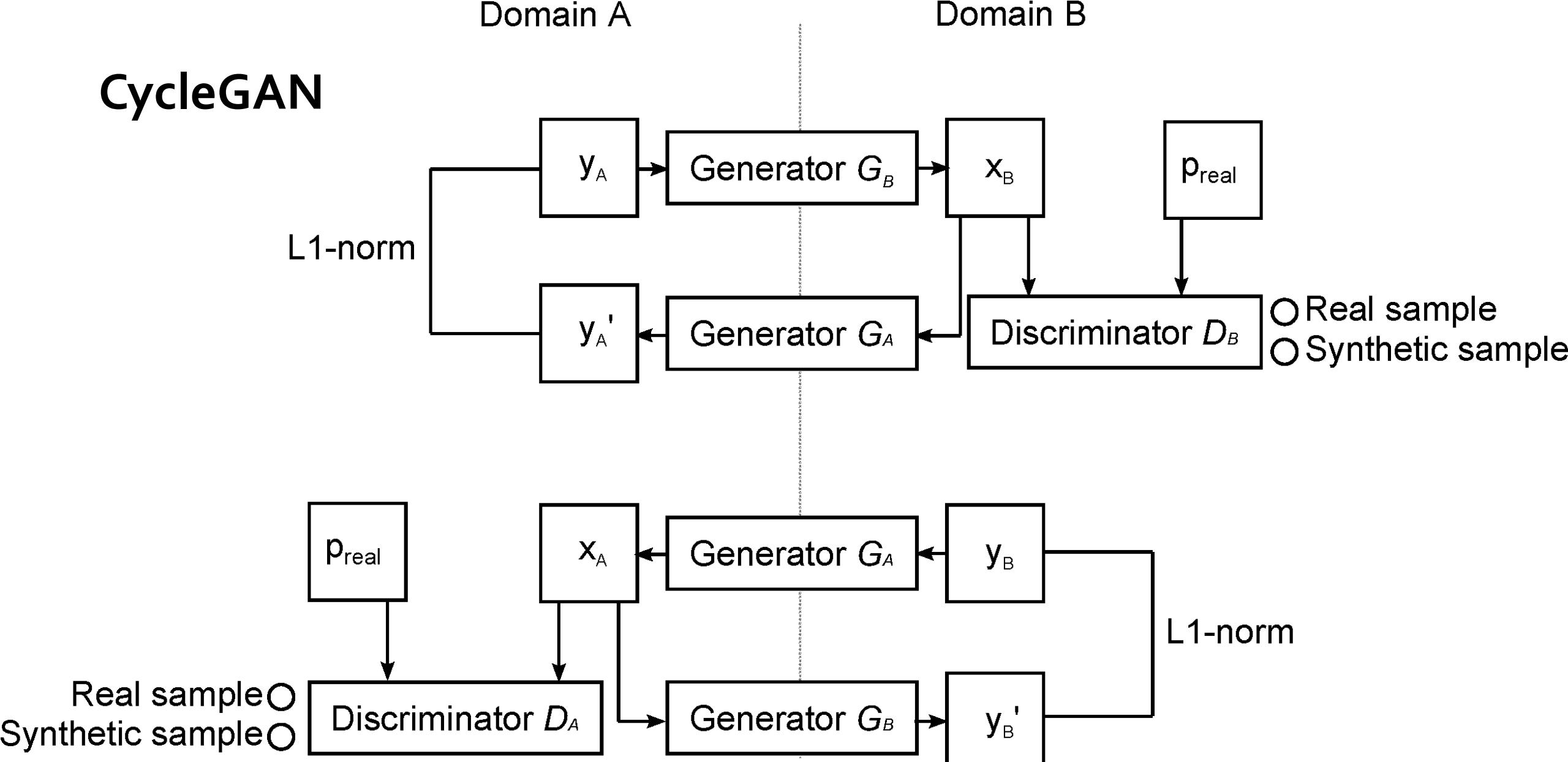
Unpaired training: self-regularization

Domain A
Brain MR images of patients with Alzheimer's

Domain B
Brain MR images of healthy patients



CycleGAN



CycleGAN

GAN objective functions in domain B and domain A and a cycle consistency loss

$$V_B(D_B, G_B) = \mathbb{E}_{x \sim p_{data_B}} [\log D_B(x)] + \mathbb{E}_{y \sim p_{data_A}} [\log (1 - D_B(G_B(y)))]$$

$$V_A(D_A, G_A) = \mathbb{E}_{y \sim p_{data_A}} [\log D_A(y)] + \mathbb{E}_{x \sim p_{data_B}} [\log (1 - D_A(G_A(x)))]$$

$$V_{Cycle}(G_A, G_B) = \mathbb{E}_{y \sim p_{data_A}} [||G_A(G_B(y)) - y||_1] + \mathbb{E}_{x \sim p_{data_B}} [||G_B(G_A(x)) - x||_1]$$

$$\min_{G_A, G_B} \max_{D_A, D_B} V(G_A, G_B, D_A, D_B) = V_B(D_B, G_B) + V_A(D_A, G_A) + \lambda V_{Cycle}(G_A, G_B)$$

CycleGAN

Domain A



Domain B



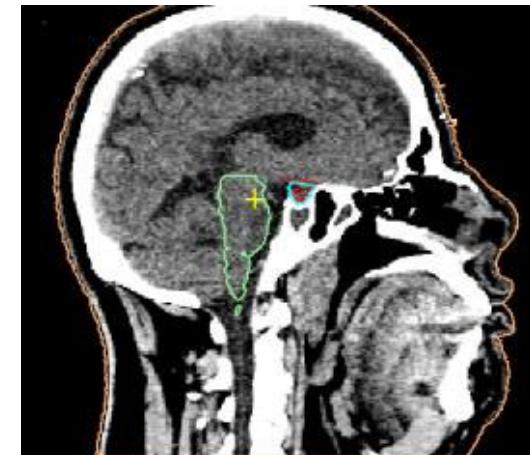
UNIVERSITY
OF TWENTE.

Example: MR to CT synthesis

- Radiotherapy treatment planning requires
 - MR volume
 - Soft tissue contrast
 - Tissue delineation
 - CT volume
 - Electron density
 - Dose calculation
- Acquisition of both volumes leads to
 - Increase in time and money
 - Decrease in patient comfort

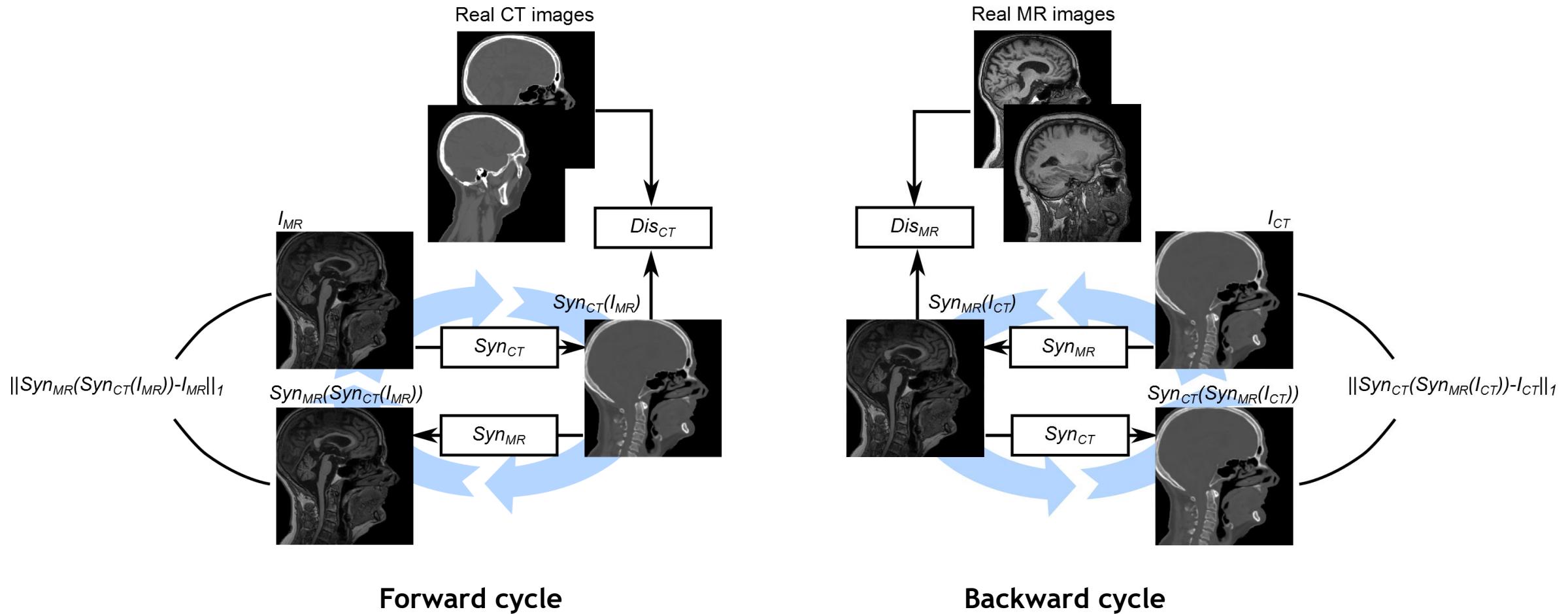


MR



CT

Example: MR to CT synthesis

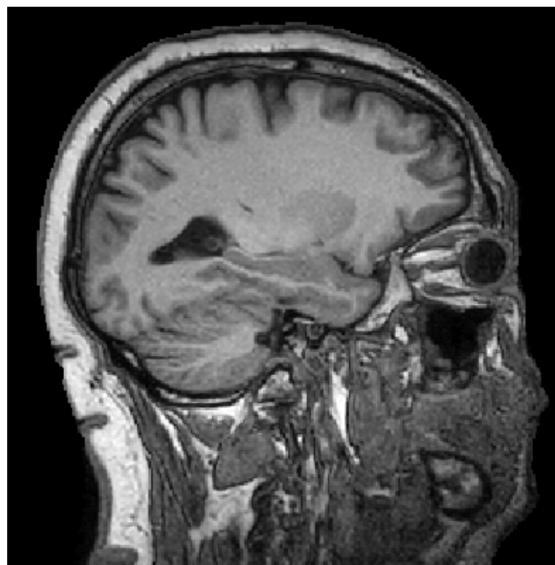


Forward cycle

Backward cycle

UNIVERSITY
OF TWENTE.

Example: MR to CT synthesis



I_{MR}



$Syn_{CT}(I_{MR})$



I_{CT}



Regression

UNIVERSITY
OF TWENTE.

Example: MR to CT synthesis



I_{MR}

$Syn_{CT}(I_{MR})$

I_{CT}

$|Syn_{CT}(I_{MR}) - I_{CT}|$

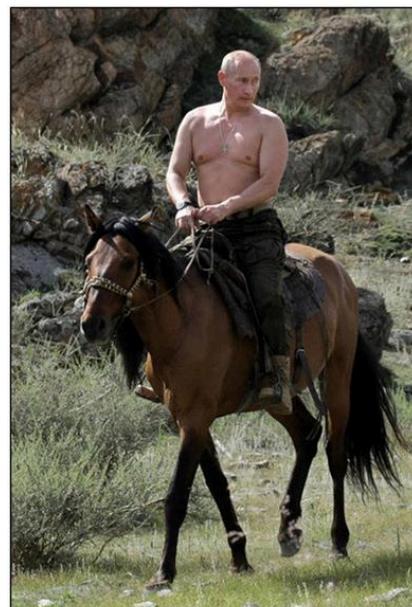
UNIVERSITY
OF TWENTE.

CycleGAN

Domain A



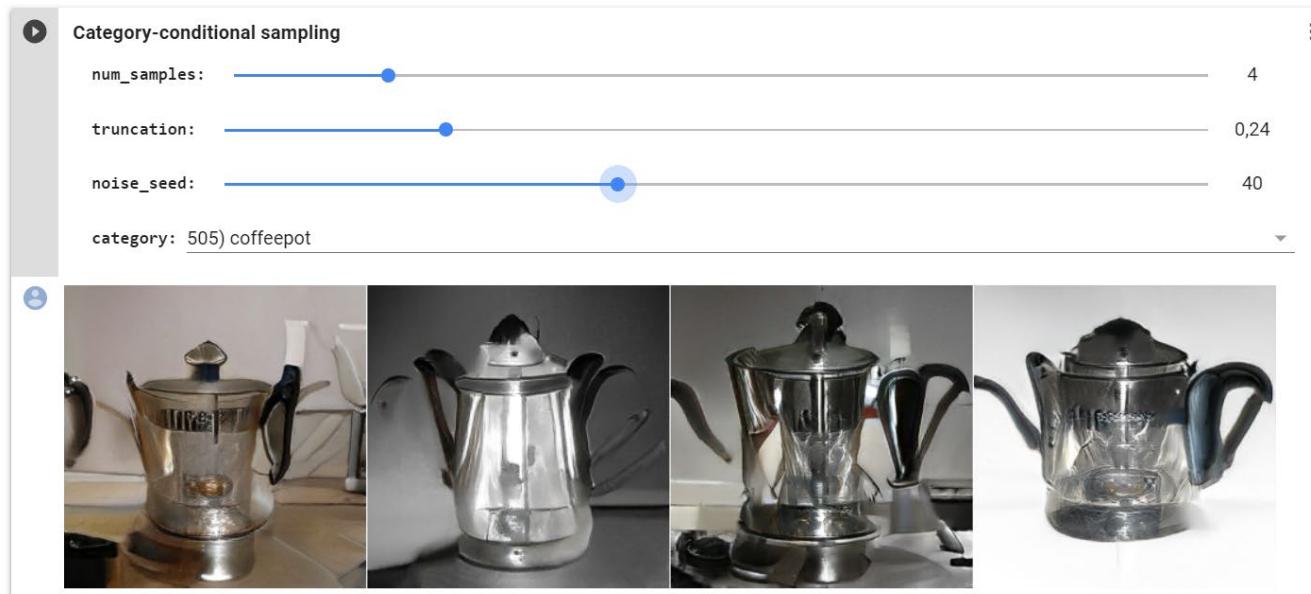
Domain B



UNIVERSITY
OF TWENTE.

Summary

- Interpretability + explainability important for practical ML use
- Generative adversarial networks are SOTA for image synthesis
- Conditional GANs synthesize samples with particular characteristics
- Adversarial networks can define loss functions that we cannot
- Many applications in medical image analysis



Practical assignment

Train your own GAN! <https://tinyurl.com/capitaselectacolab>

- Match a normal distribution
- MNIST digits (unconditional + conditional)
- Histopathology images: <https://github.com/basveeling/pcam>
- Run with free GPU: in Playground (no saving) or make copy in Google Drive (log in)
- Experiments with BigGAN <https://tinyurl.com/y8yuqyqv>

