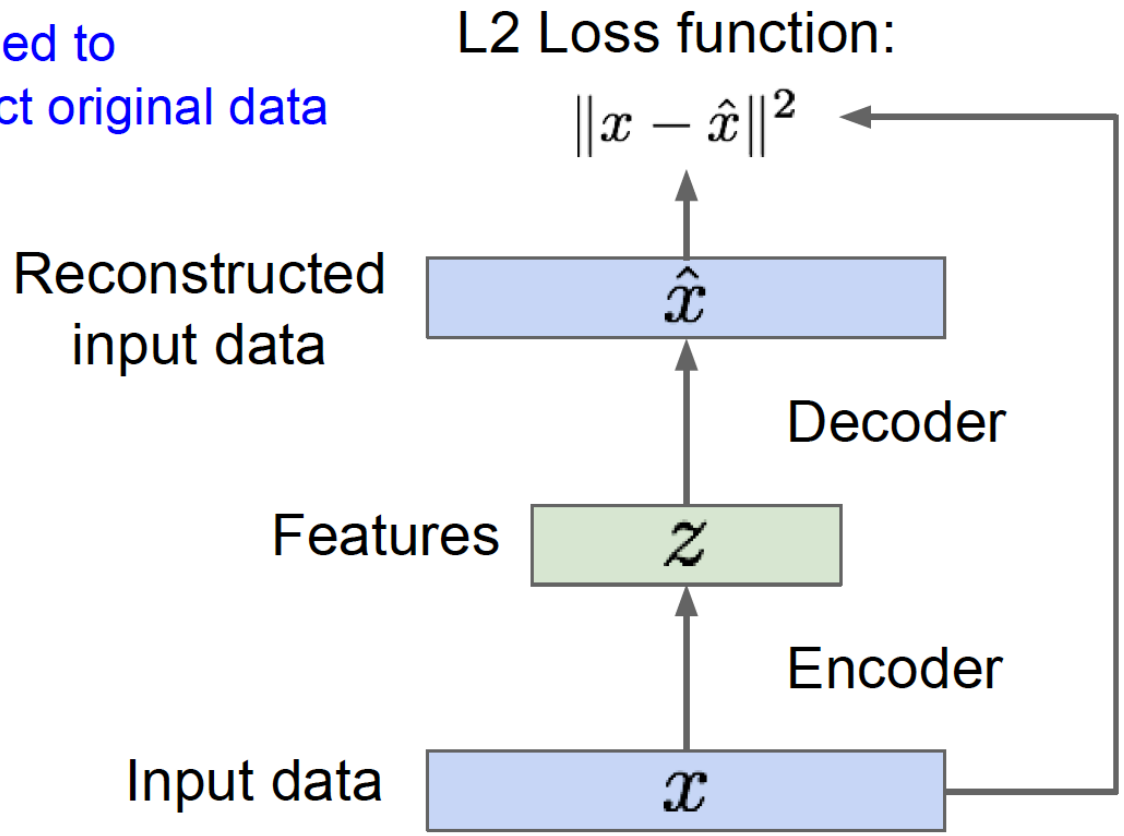
Generative Adversarial Networks And Their Applications

Bindita Chaudhuri

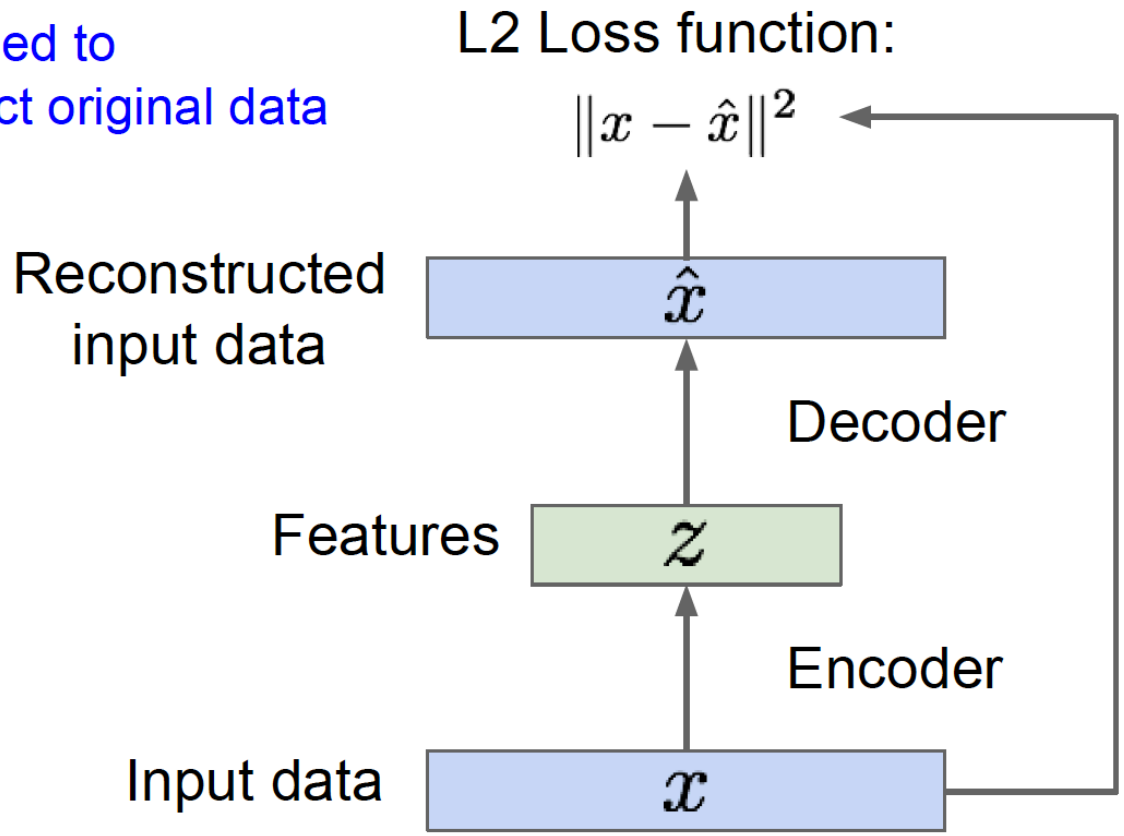
Unsupervised Learning: Autoencoders

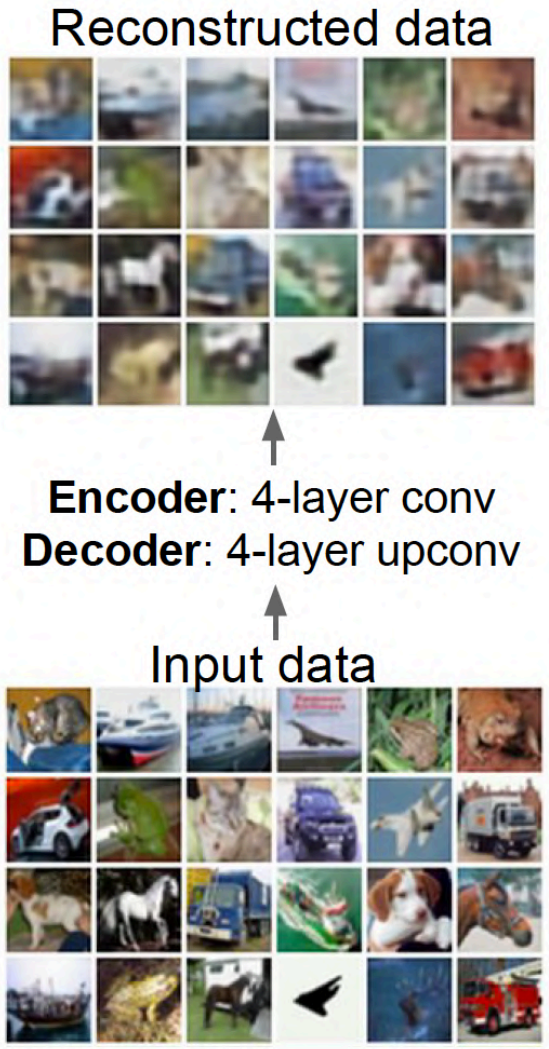
Train such that features can be used to reconstruct original data



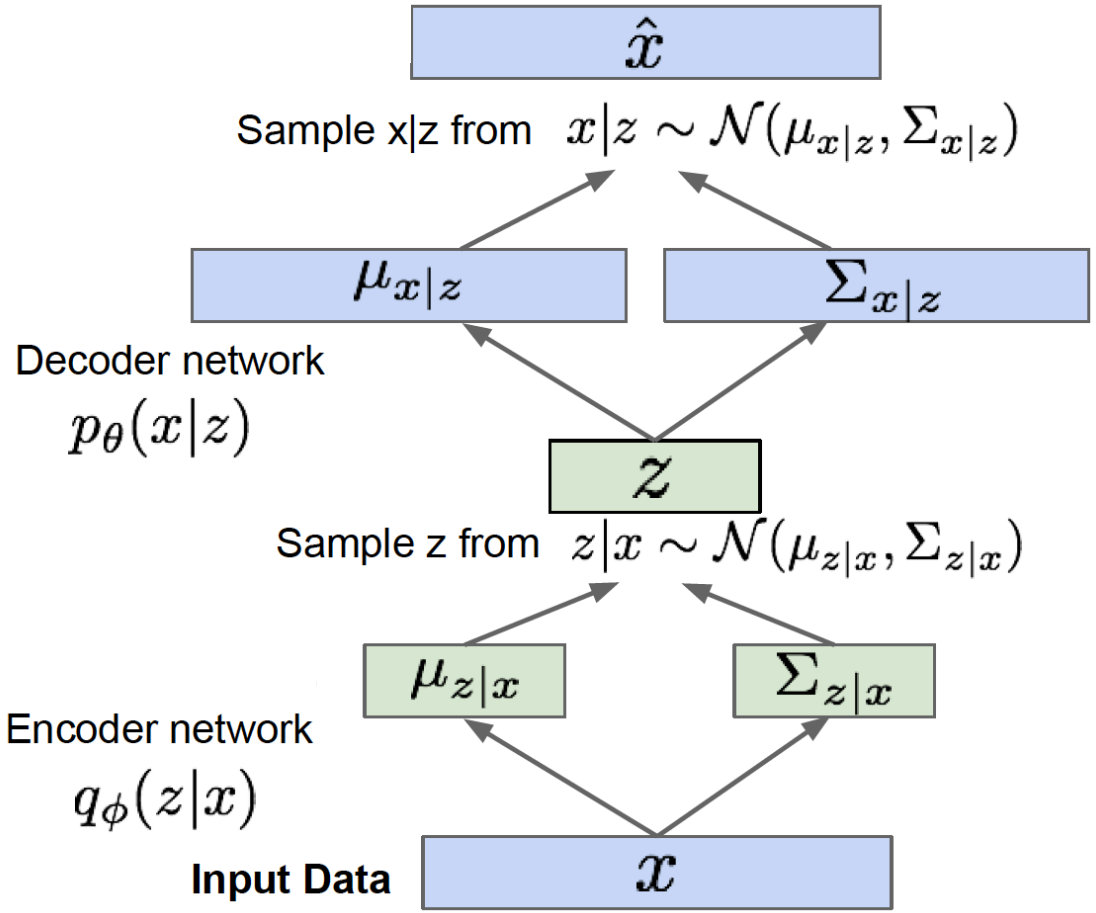
Unsupervised Learning: Autoencoders

Train such that features can be used to reconstruct original data

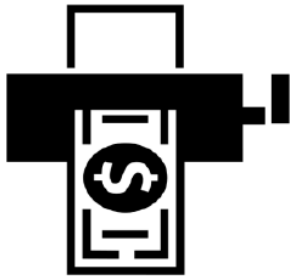




Unsupervised Learning: Variational Autoencoders

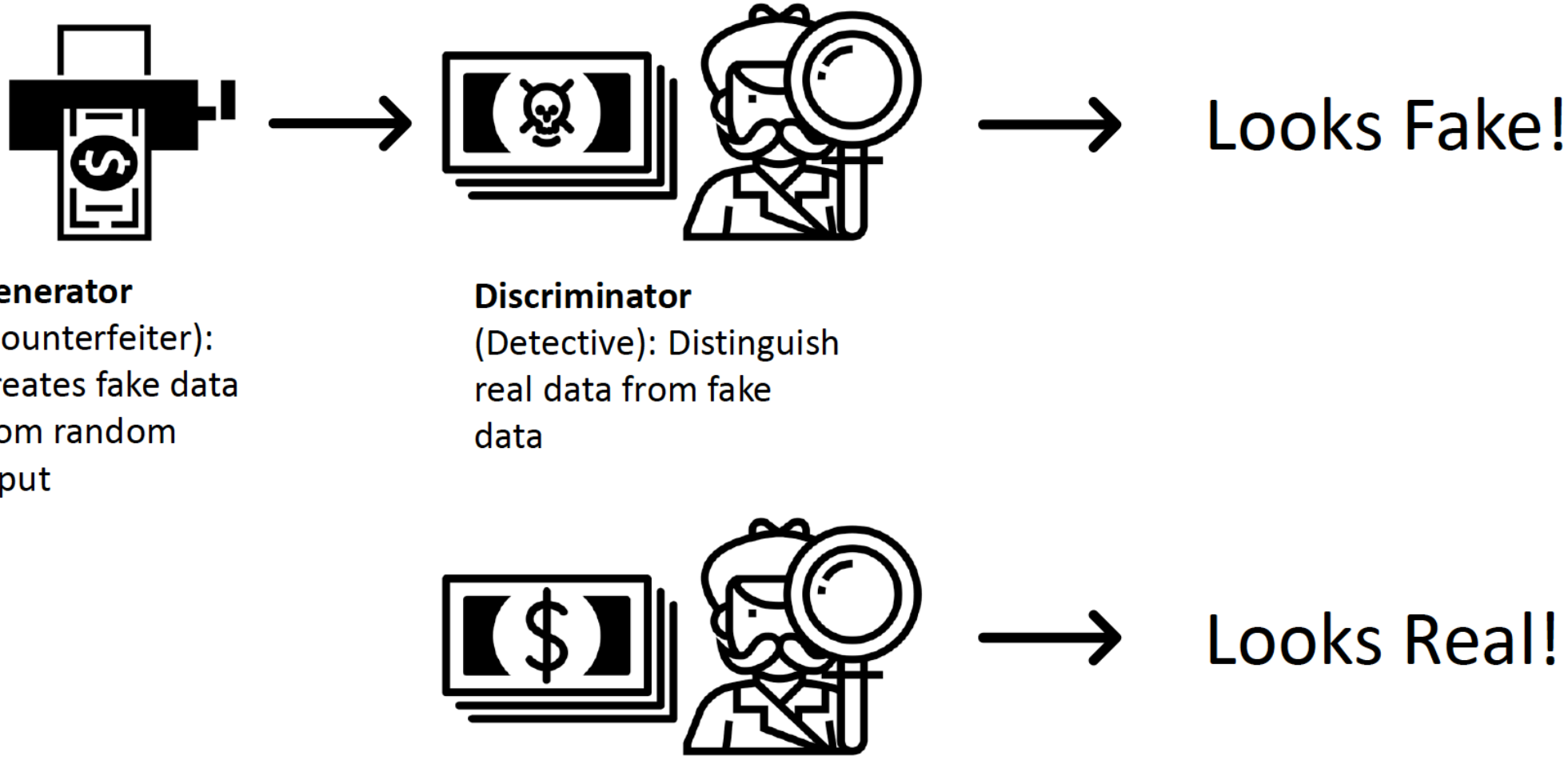


Generative Adversarial Networks: Idea



Generator (Counterfeiter): Creates fake data from random input

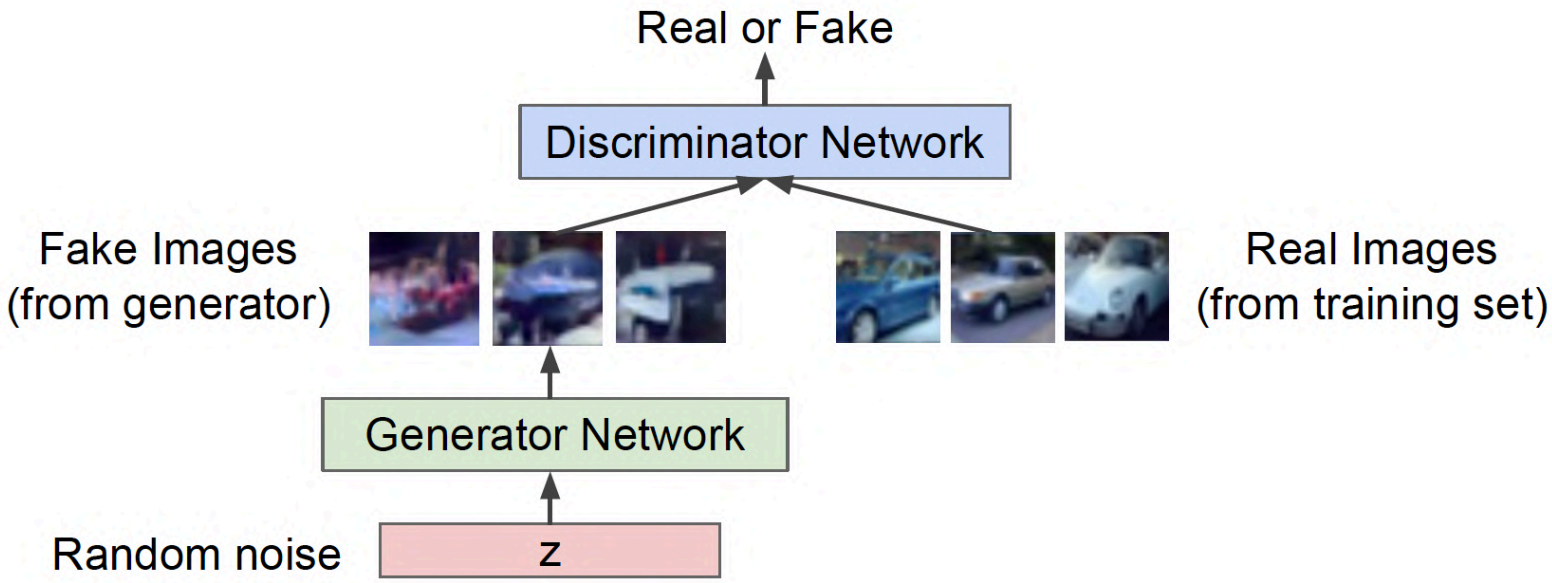
Generative Adversarial Networks: Idea



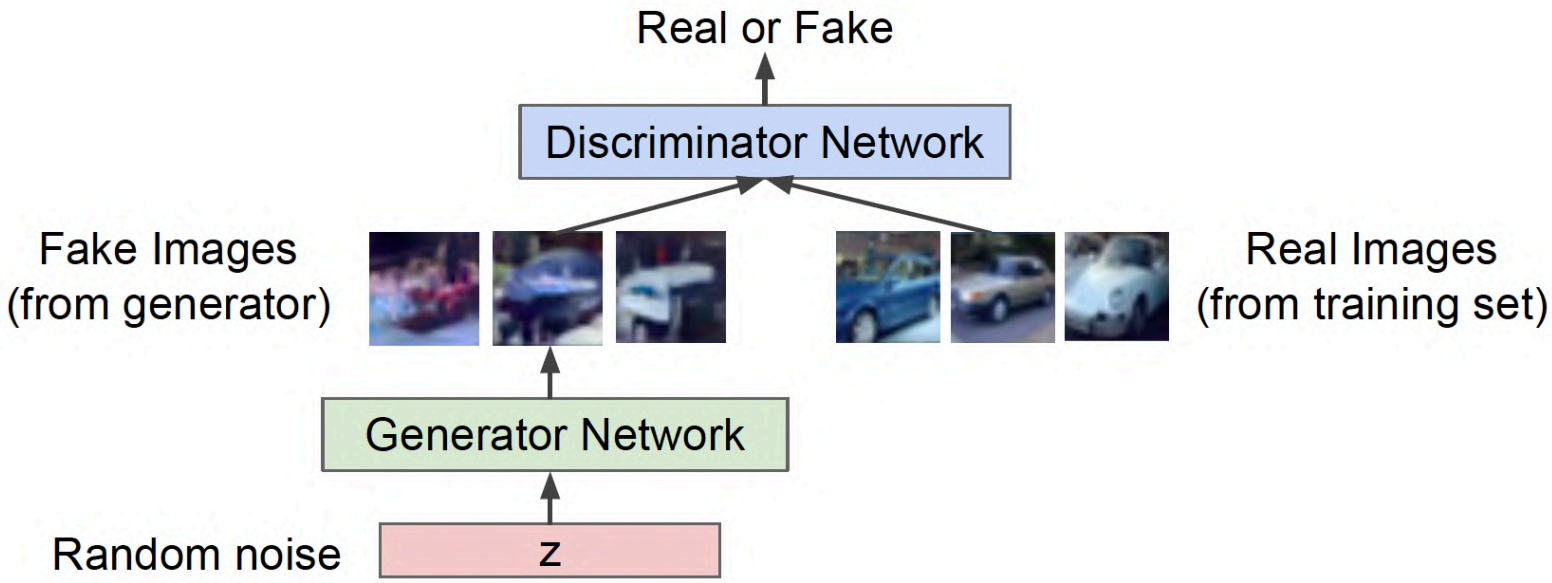
Generator (Counterfeiter): Creates fake data from random input

Discriminator (Detective): Distinguish real data from fake data

Generative Adversarial Networks



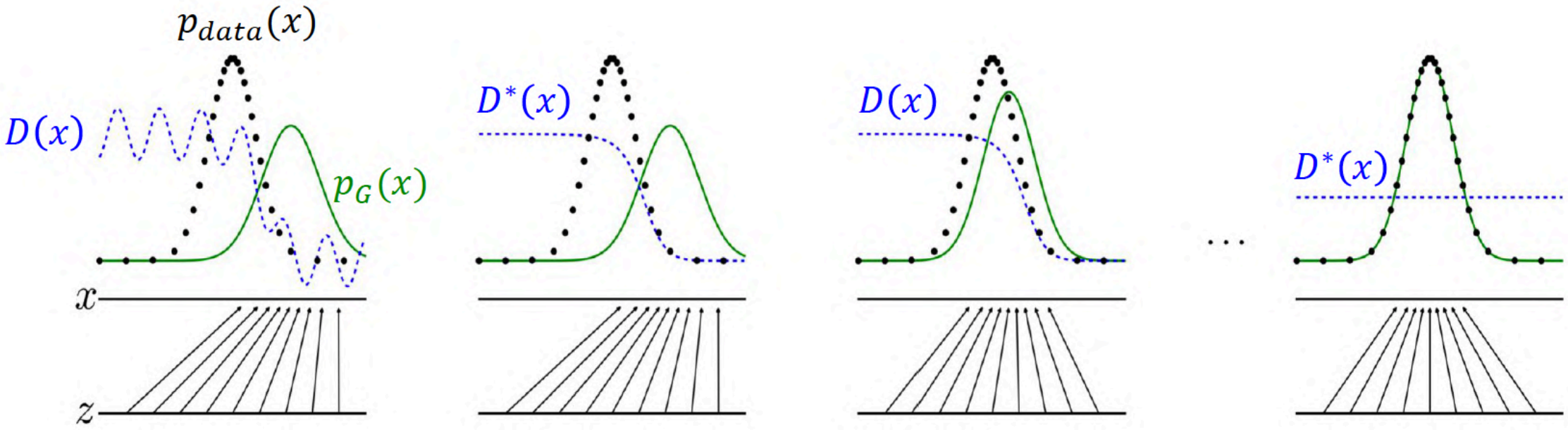
Generative Adversarial Networks



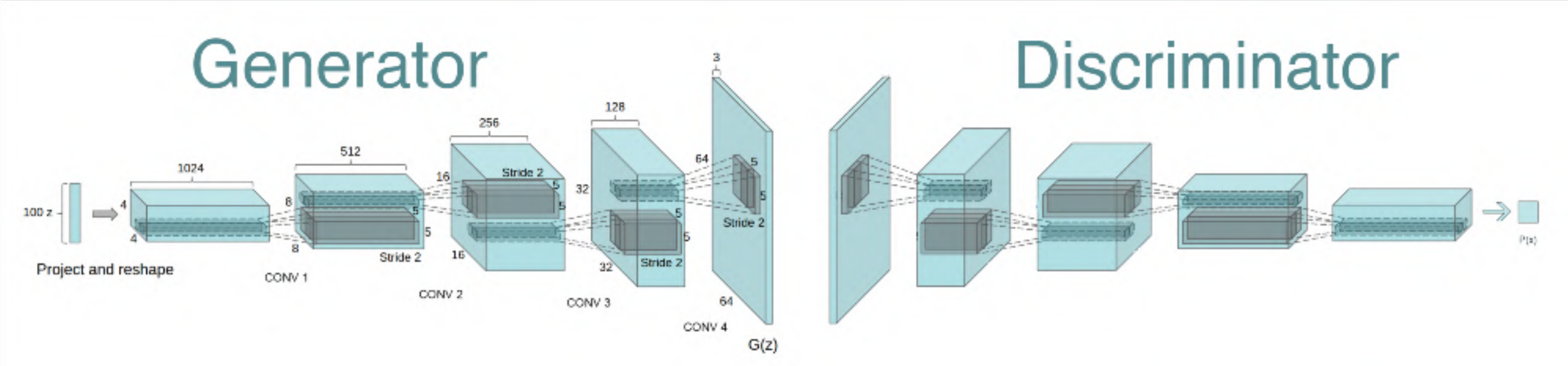
Minimax objective function:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Distributions during training

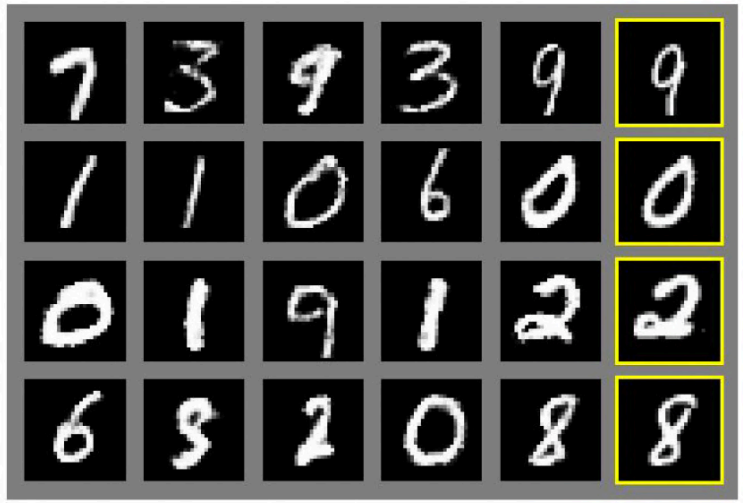


GAN: Sample Architecture (DC-GAN)



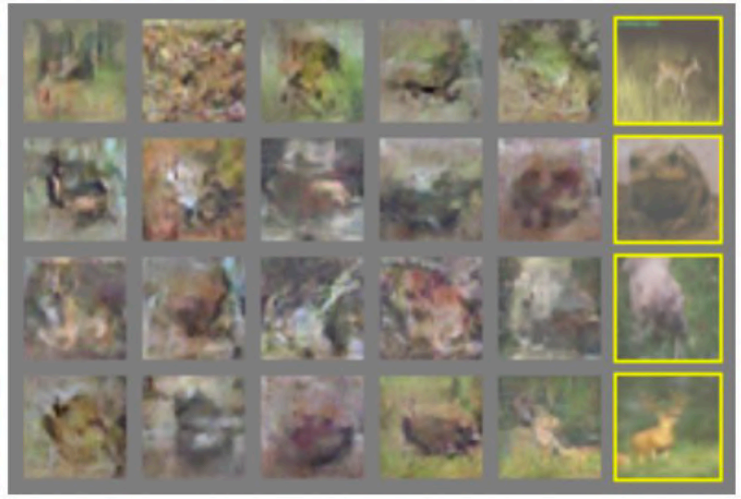
Generated Samples

MNIST

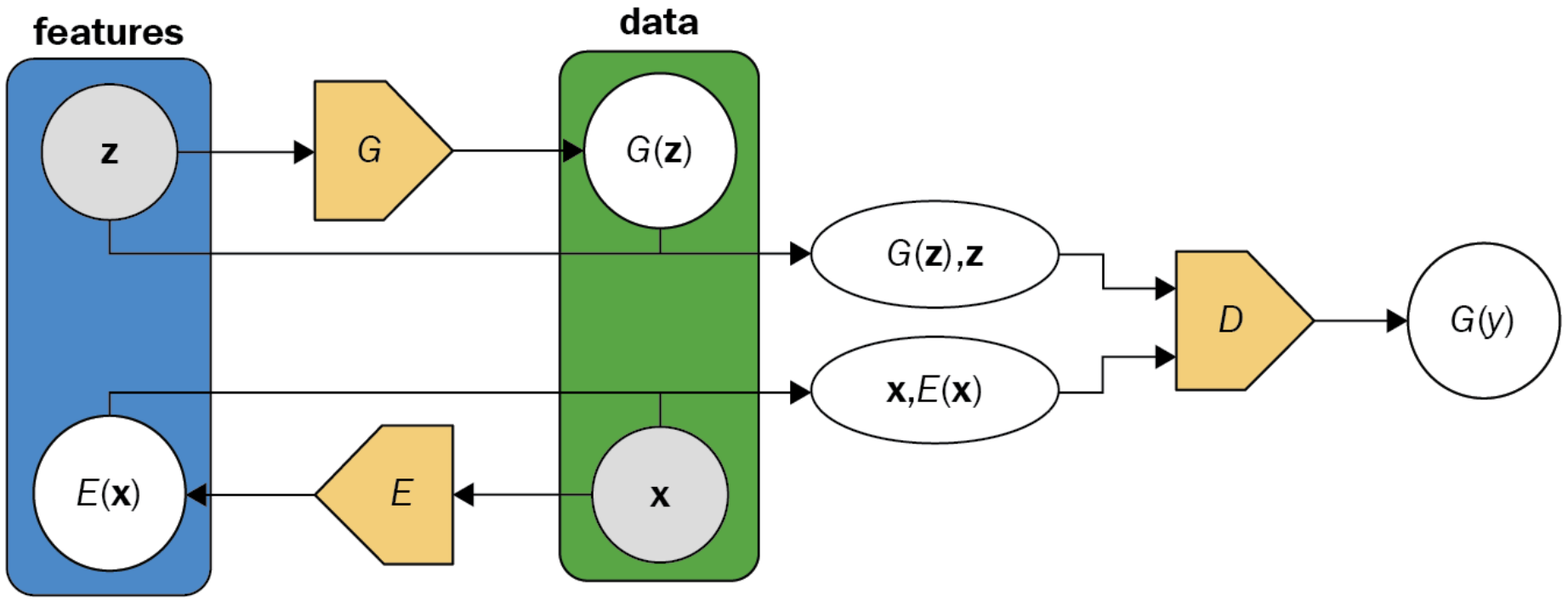


Faces

CIFAR 10

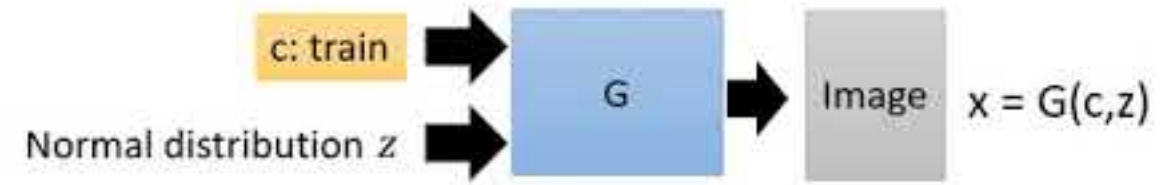


Bidirectional GAN (BiGAN)



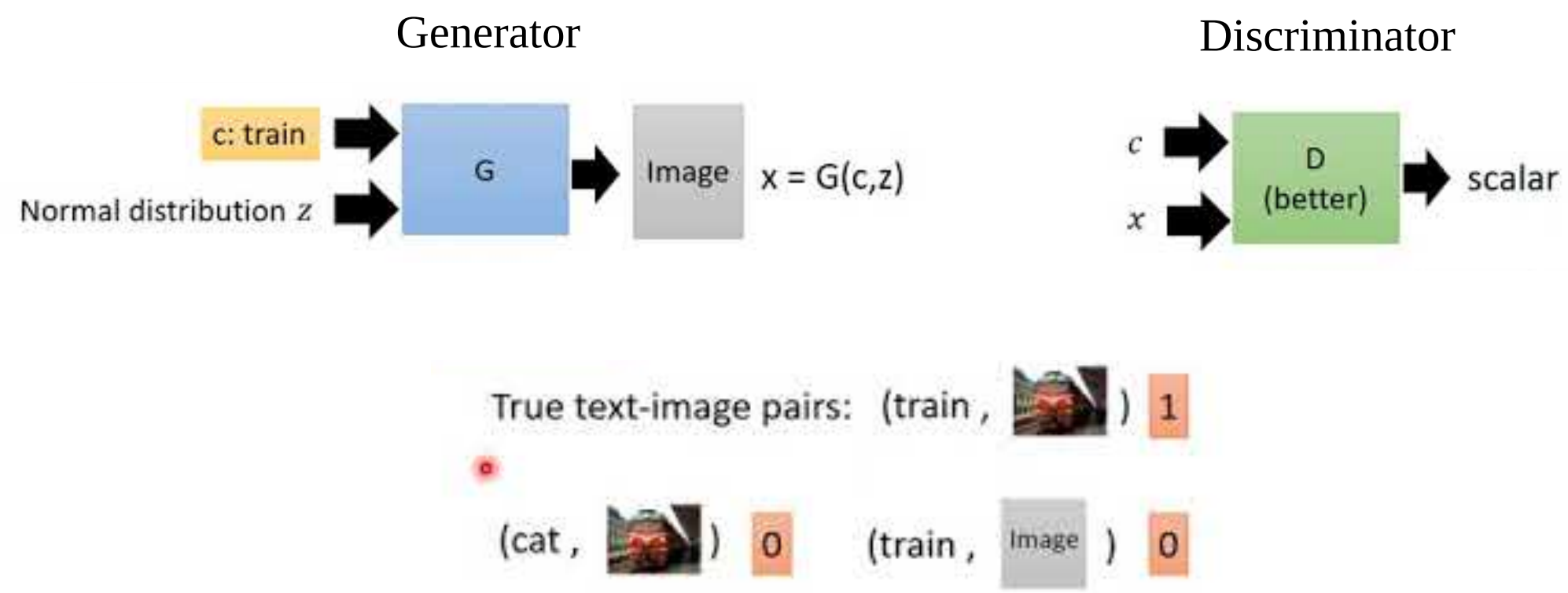
Conditional GAN (cGAN)

Generator

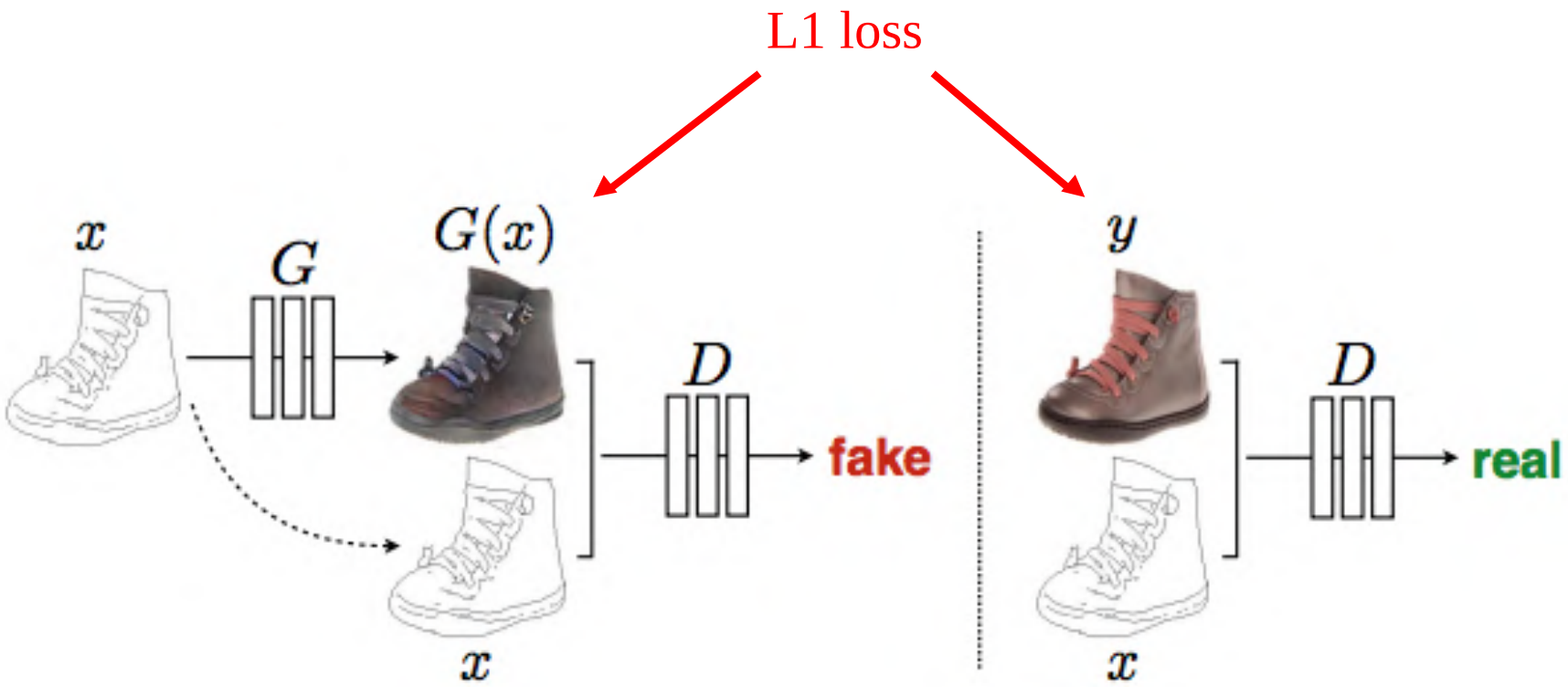


Discriminator

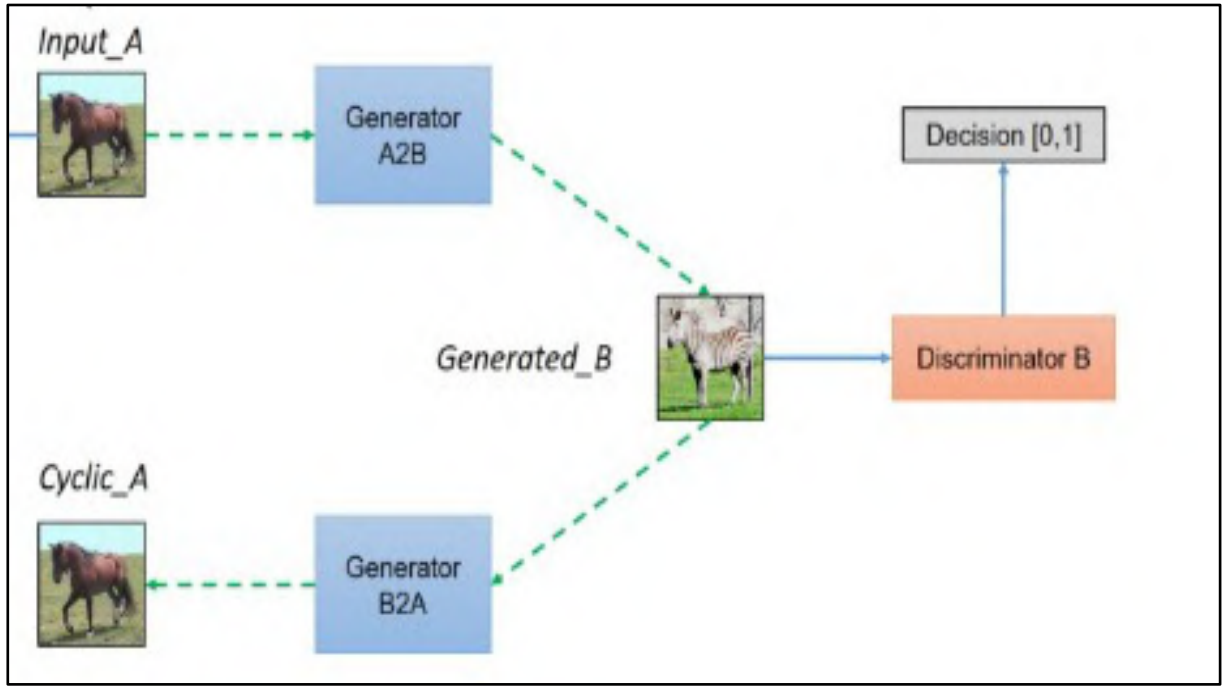
Conditional GAN (cGAN)



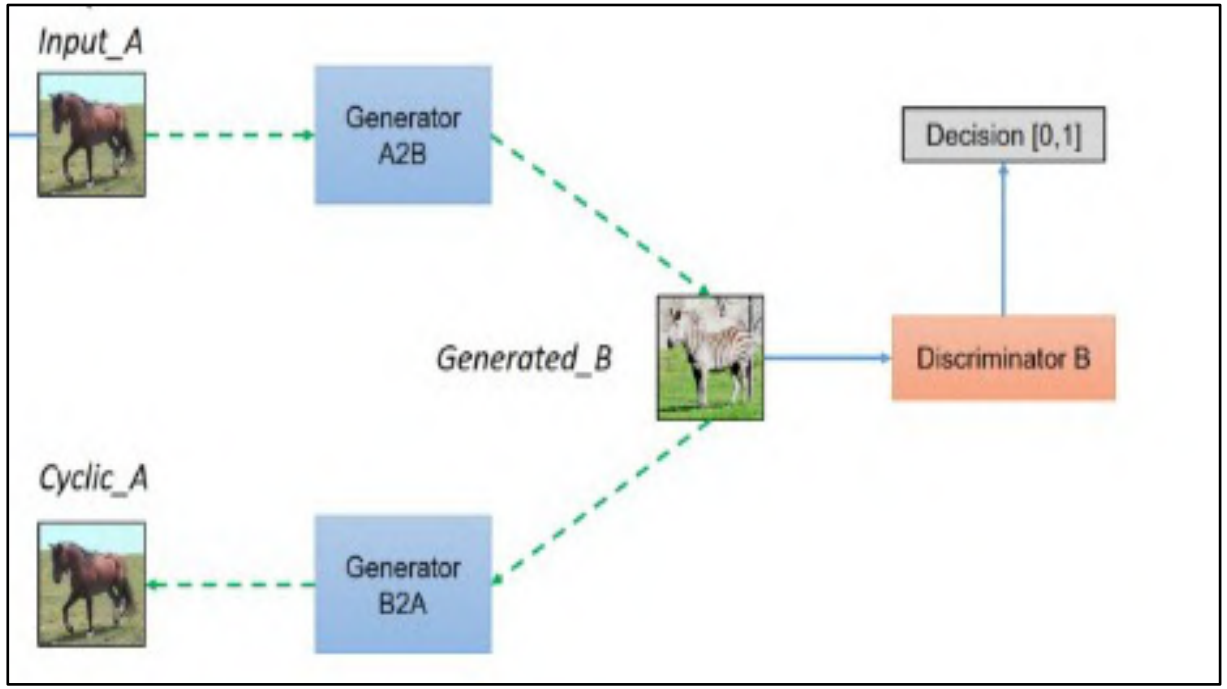
Pix2Pix: Type of cGAN

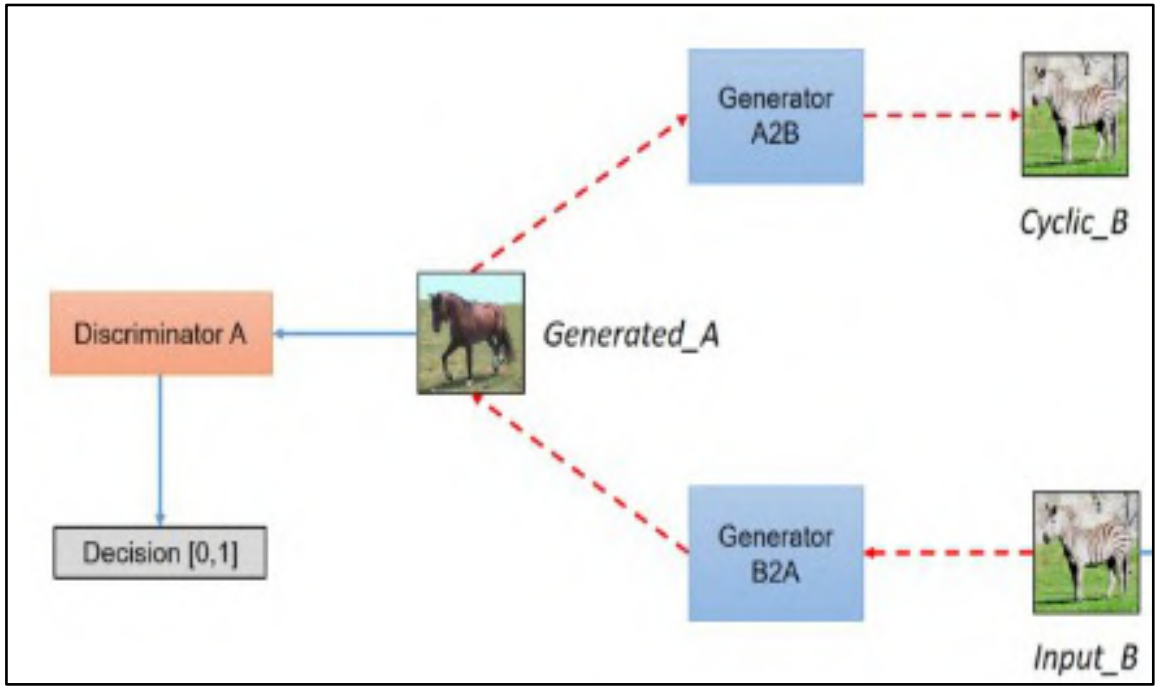


CycleGAN: Unsupervised Pix2Pix

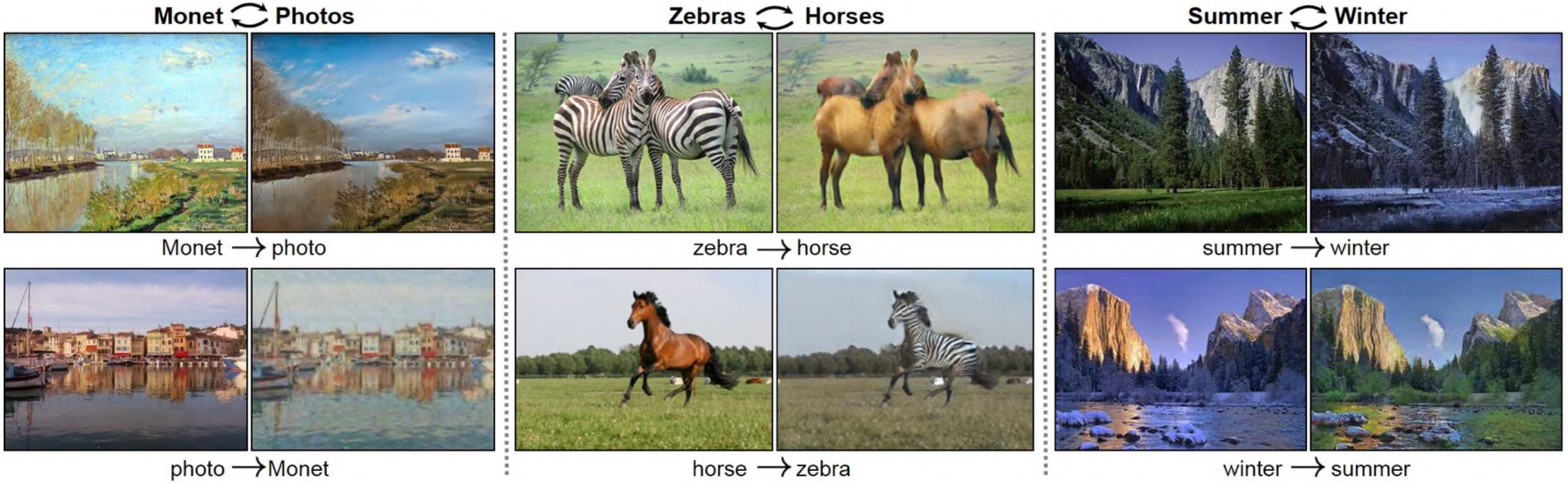


CycleGAN: Unsupervised Pix2Pix

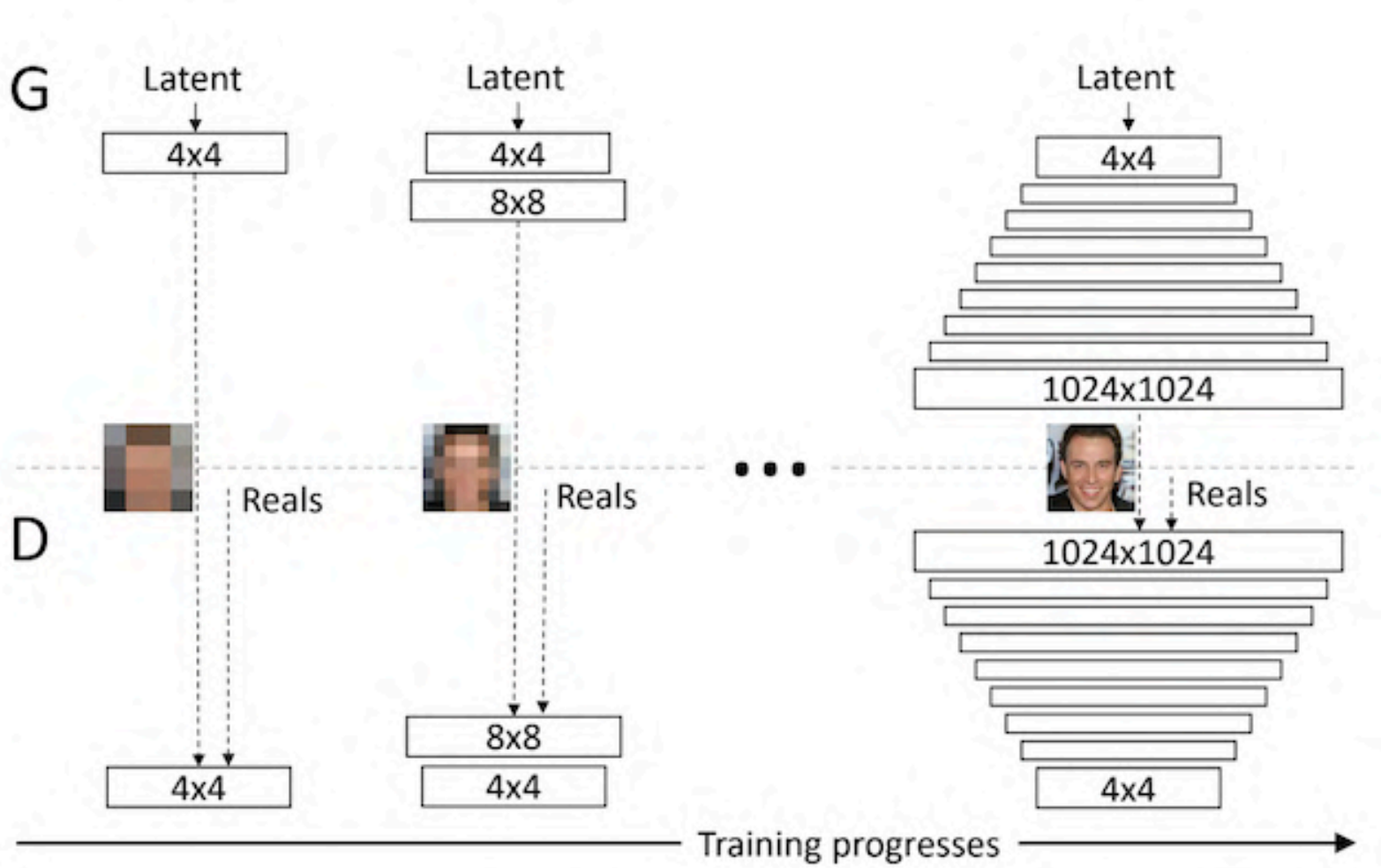




CycleGAN Results



Progressive Growing of GANs



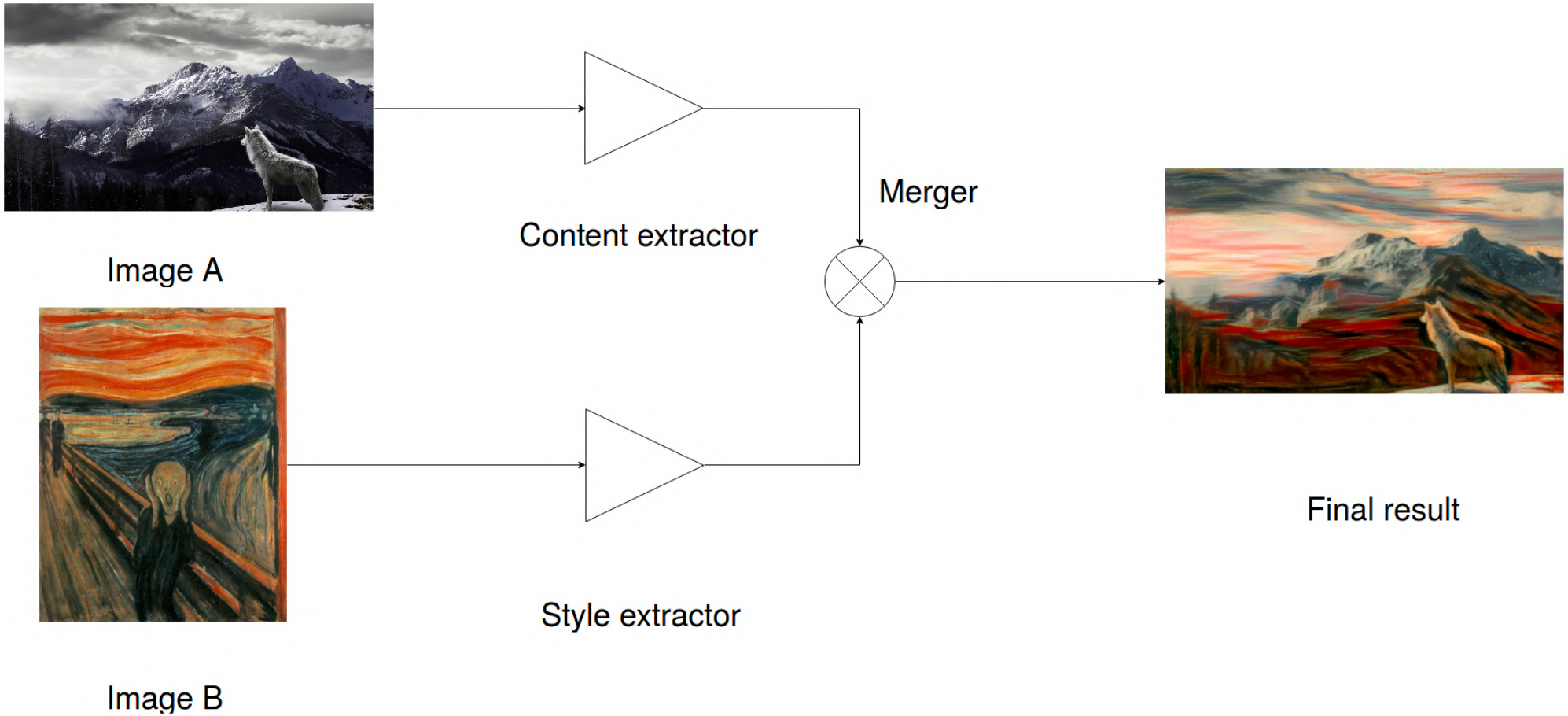
Progressive GAN Results

Celebrities

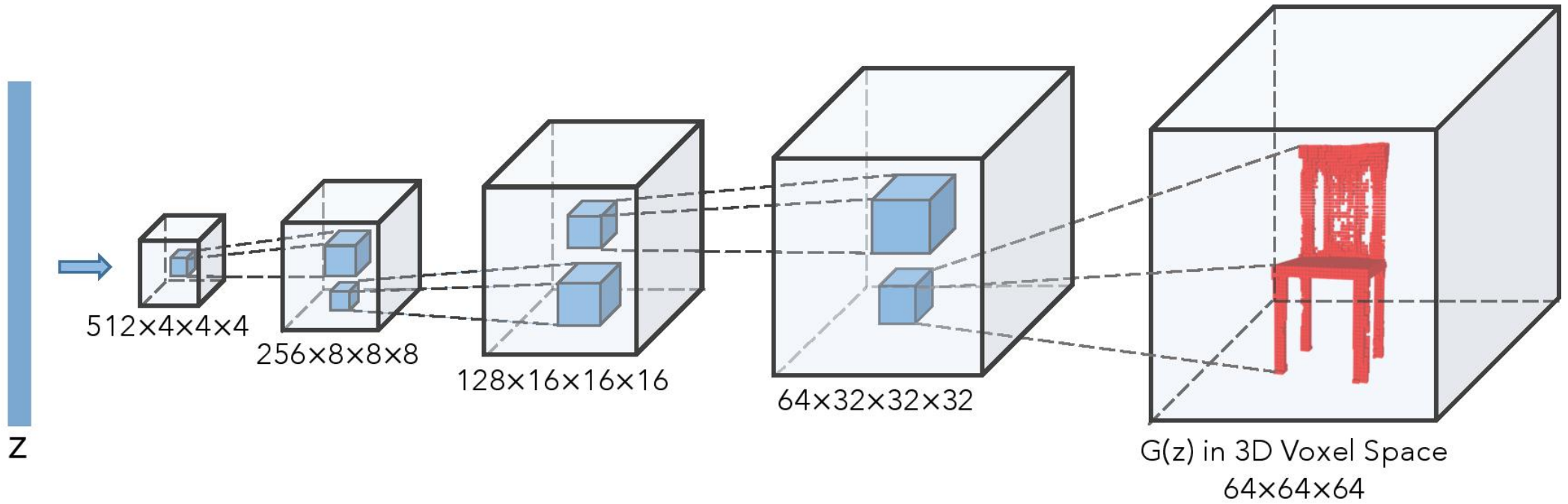
Bedrooms

Objects

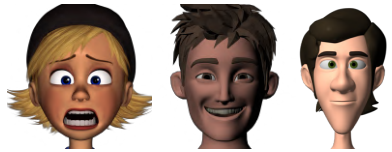
Application: Neural Style transfer



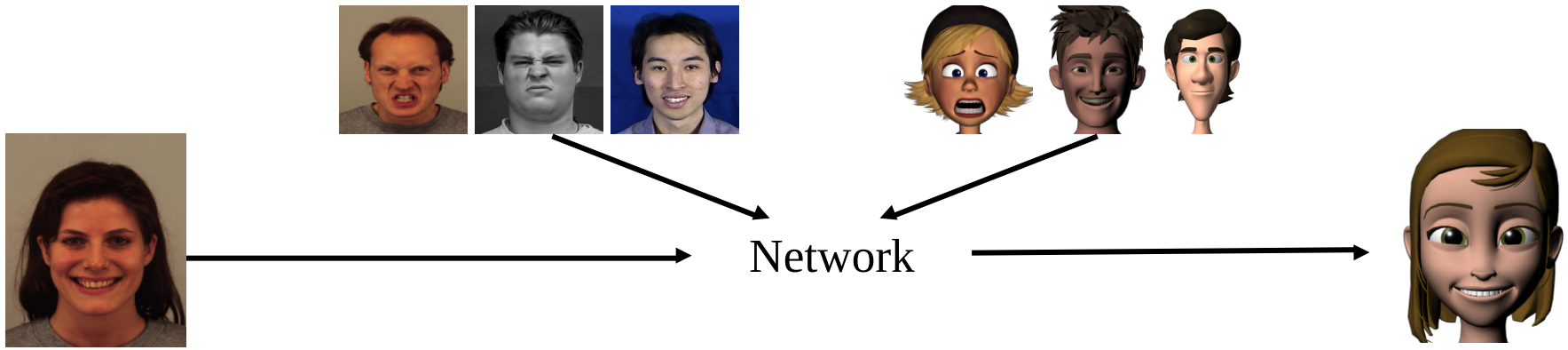
Application: 3D GAN



My Project: Facial Motion Retargeting

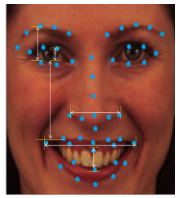


My Project: Facial Motion Retargeting



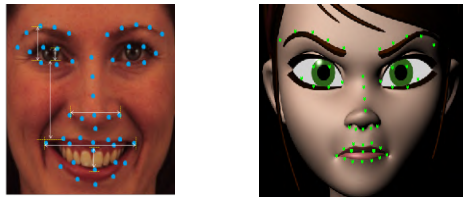
2D-to-3D CycleGAN

- Compute facial landmarks:



2D-to-3D CycleGAN

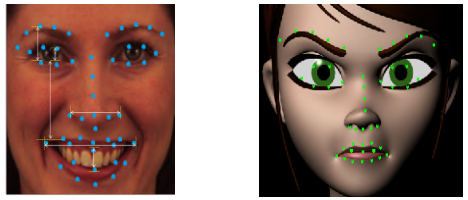
- Compute facial landmarks:



- Convert 3D model to 2D position map:

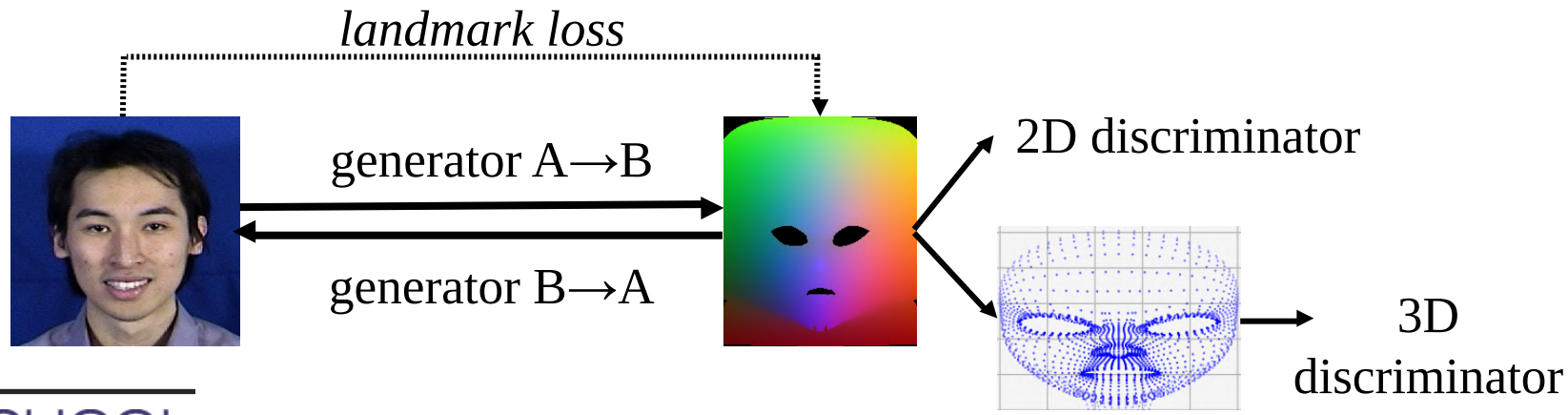
2D-to-3D CycleGAN

- Compute facial landmarks:

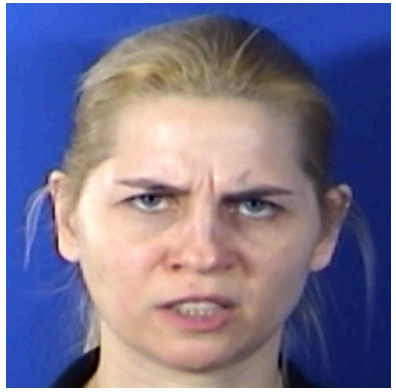
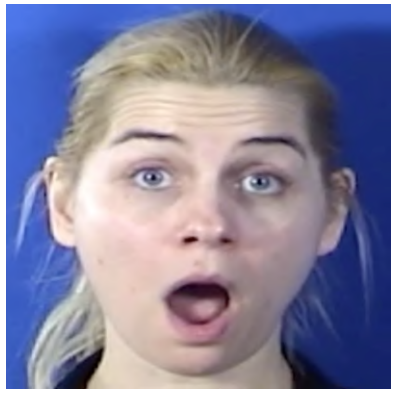


- Convert 3D model to 2D position map:

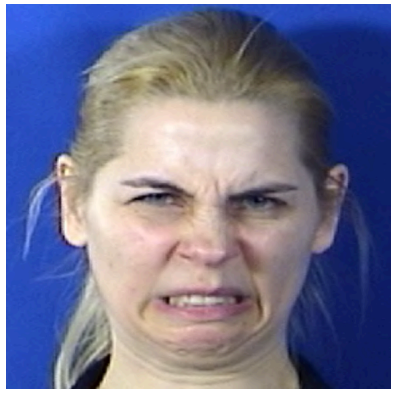
- Train CycleGAN:



Results

Input	Landmark only	CycleGAN
		
		

Results

Input	Landmark only	CycleGAN
		
		

Useful links

- GAN Zoo: https://github.com/hindupuravinash/the-gan-zoo
- GAN hacks: https://github.com/soumith/ganhacks
- **Code Bases**:
  - Tensorflow: https://www.tensorflow.org/tutorials/generative/dcgan
  - Keras: https://github.com/eriklindernoren/Keras-GAN
  - Pytorch: https://github.com/pytorch/examples/tree/master/dcgan
- **References:**
  - http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf
  - https://www.cs.toronto.edu/~rgrosse/courses/csc321_2018/slides/lec19.pdf
  - https://www.cs.cmu.edu/~bhiksha/courses/deeplearning/Fall.2015/slides/lec13.GAN.pdf

Questions