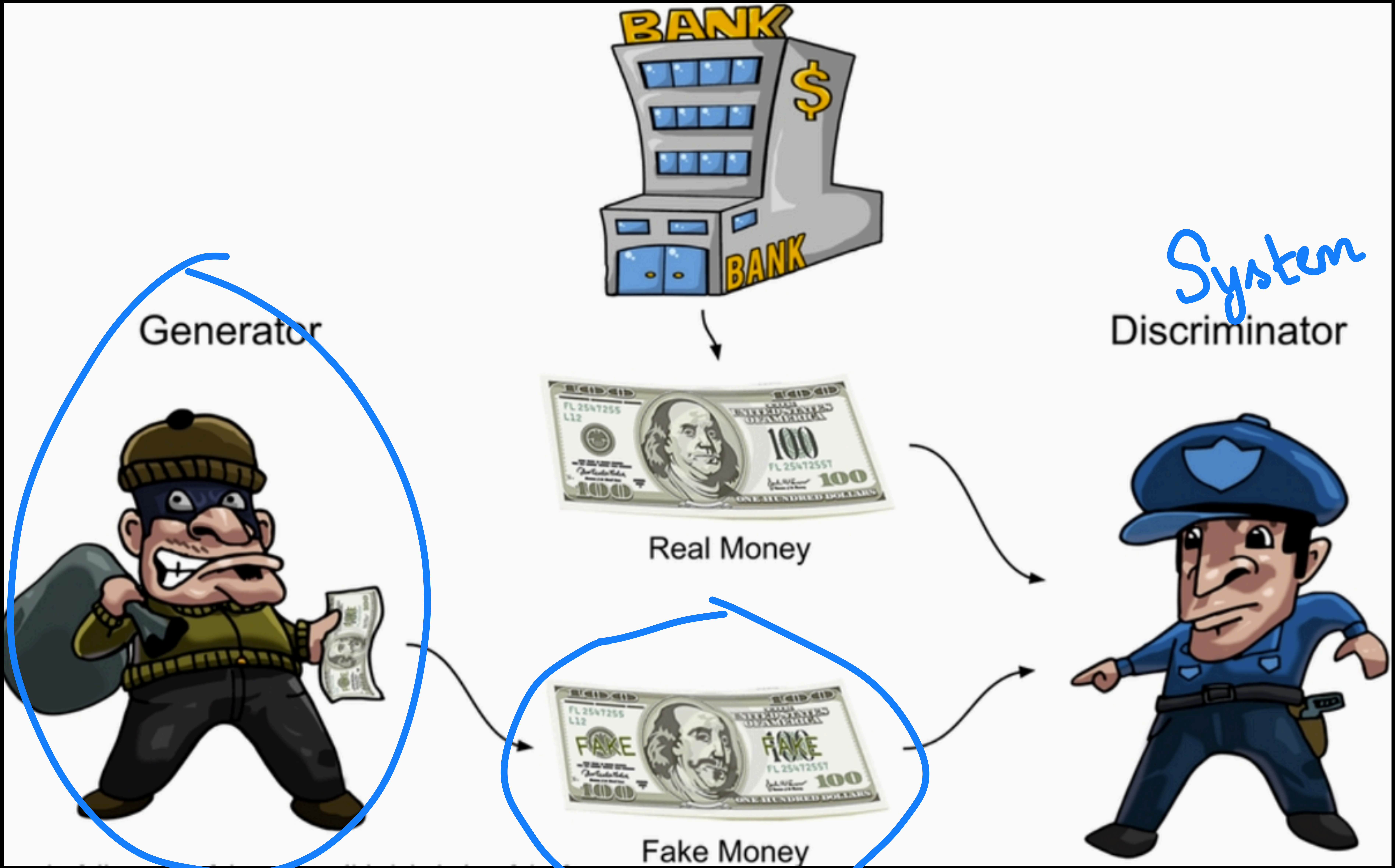
# E9: 309 ADL 09-12-2020

http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

# GAN - Overview



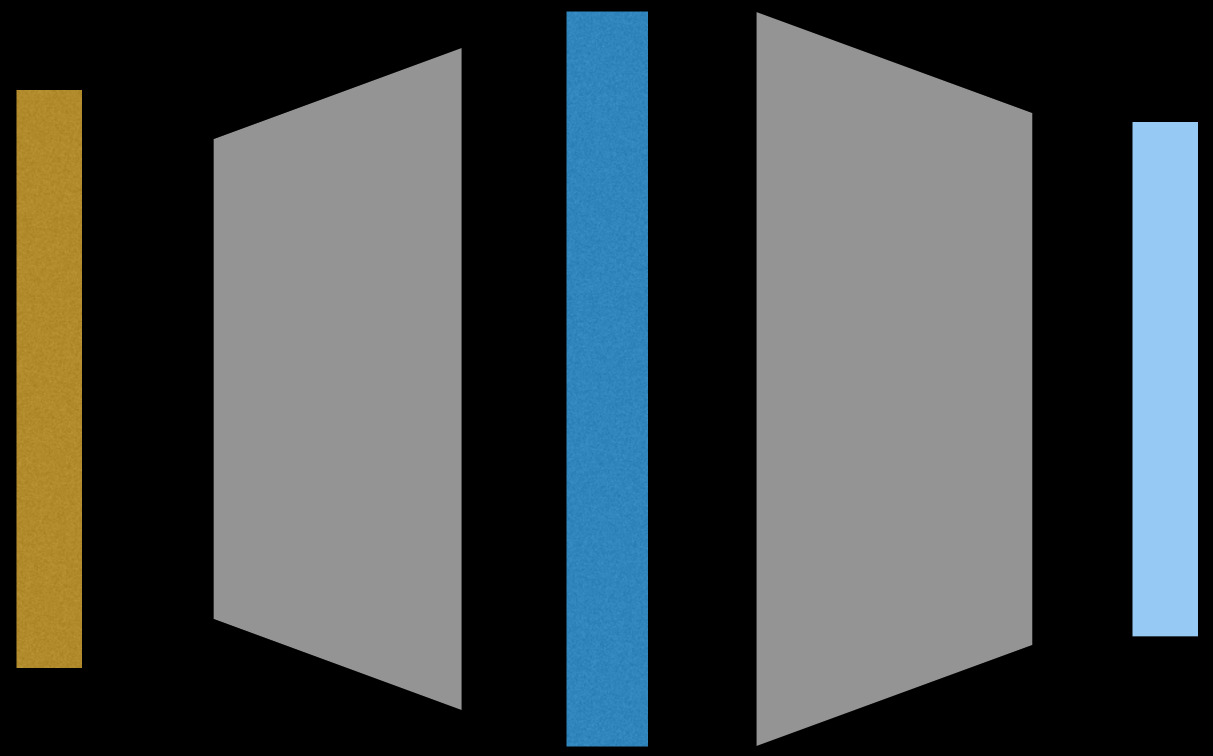
# Generative Adversarial Networks

- Generator networks
  - Differentiable network
  - Draws a random noise sample
    - ✓ Convert the noise to the data distribution

$\theta^G$

$$\mathbf{x}' \sim p_{\theta^G}(\mathbf{x}|\mathbf{z})$$

$\mathbf{z} \sim p(\mathbf{z})$

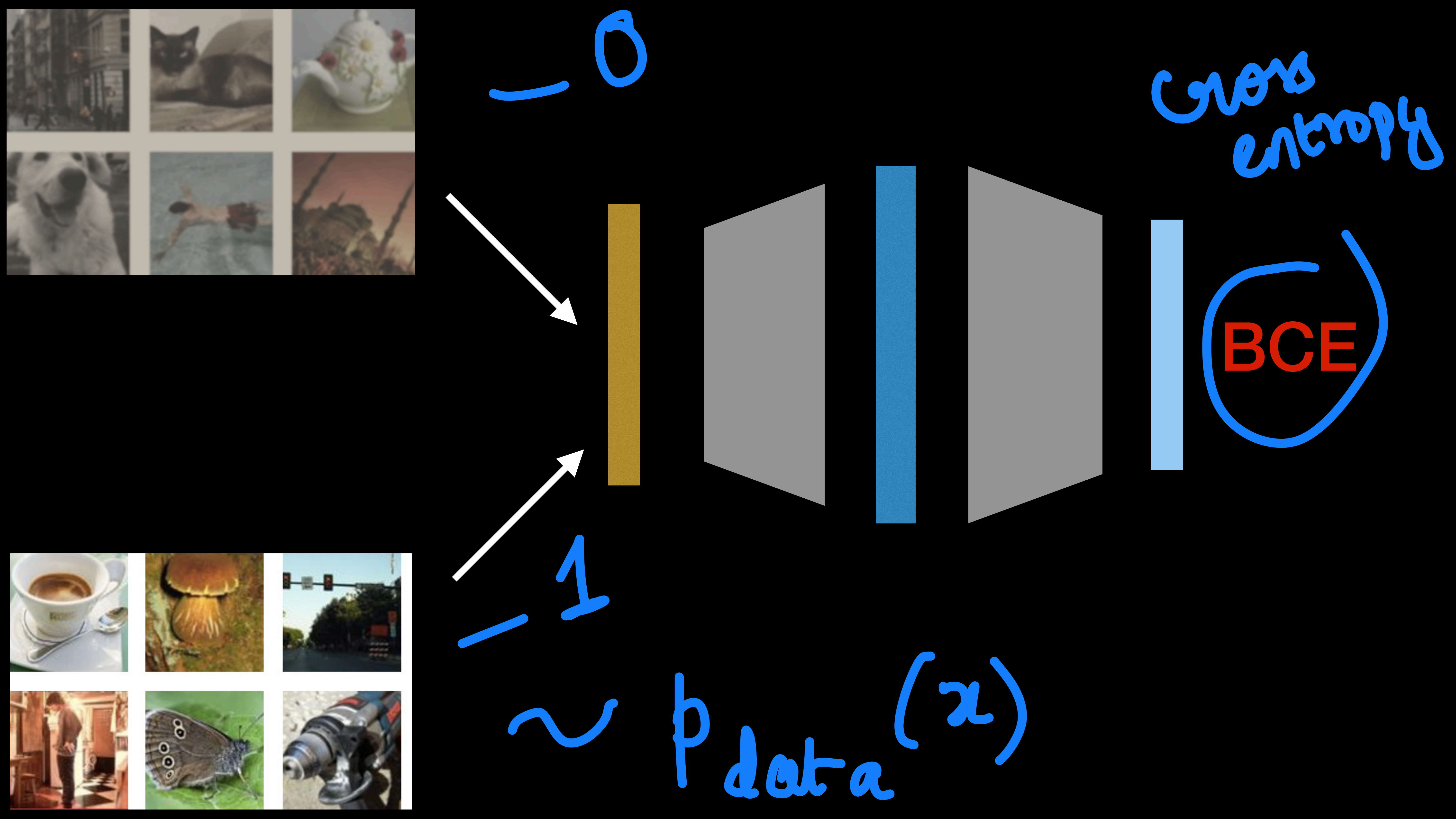


$\mathbf{x}' = G(\mathbf{z}, \boldsymbol{\theta}^G)$

# Generative Adversarial Networks

- Discriminator network
  - Two class classifier
    - ✓ Label 0 for $x'$
    - ✓ Label 1 for $x$
  - Model learning

Model generated samples




True samples

# Min-max game

- The binary cross entropy loss at the discriminator network is

class 1

$(1 - D(G(z))$

$$E_D = -\underline{\underline{\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log\ D(\mathbf{x})]}} - \underline{\underline{\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}}}[log\ (D(G(\mathbf{z})))]$$

$$E_G = -E_D$$

$\theta^G$ , $\theta^D$

class 0

- The discriminator is trained using B.C.E loss

$$\theta_D{}^n = \theta_D{}^{n-1} - \eta \frac{\partial E_D}{\partial \theta_D}$$

$$E = \sum 1 \underbrace{\log(D(x))}_{\text{class } 1} - (1-0)\log \underbrace{(1 - D(G(z)))}_{\text{class } 0}$$

# Min-max game

- The binary cross entropy loss at the discriminator network is

$$E_D = -\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log\ D(\mathbf{x})] - \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[log\ D(G(\mathbf{z}))]$$

$$E_G = -E_D$$

- The discriminator is trained using B.C.E loss

$$\theta_G{}^n = \theta_G{}^{n-1} - \eta \frac{\partial E_G}{\partial \theta_G}$$

# GAN - min-max game

$z \xrightarrow[\theta_G]{G} x'$ $\quad x \sim p_{model}(x)$

- Disriminator cost function

$$\underline{E_D} = min\left( \underbrace{-\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D(\mathbf{x}))]}_{\text{real}} - \underbrace{\mathbb{E}_{\mathbf{z}\sim \cancel{p_{model}}\,p(z)(\mathbf{z})}[log(1-D(G(\mathbf{z}))]}_{\text{fake}} \right)$$

$$= max\left( \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_{model}(\mathbf{x})}[log(1-D(\mathbf{x})] \right)$$

$p(z) \leftarrow$ no parameters

- Optimal cost function

$$E^* = \boxed{min_G}\ max_D\ E(D,G)$$

$$E(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_{model}(\mathbf{x})}[log(1-D(\mathbf{x})]$$

# GAN - min-max game

- Optimal disriminator

$$D^* = argmax_D \left( \underbrace{\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_{model}(\mathbf{x})}[log(1 - D(\mathbf{x})]}\right)$$

$$= argmax_D \int_x \left( log(D(\mathbf{x}))\, p_{data}(\mathbf{x}) + log(1 - D(\mathbf{x}))\, p_{model}(\mathbf{x}) \right) d\mathbf{x}$$

$$= argmax_D \underbrace{[log(D(\mathbf{x}))\, p_{data}(\mathbf{x}) + log(1 - D(\mathbf{x}))\, p_{model}(\mathbf{x})]}$$

- Functional derivative

$$\frac{p_{data}(\mathbf{x})}{D^*(\mathbf{x})} - \frac{p_{model}(\mathbf{x})}{1 - D^*(\mathbf{x})} = 0$$

$D(x)$

$\forall x$ ;

$$D^*(\mathbf{x}) = \frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_{model}(\mathbf{x})}$$

optimal
for a fixed generator

# GAN - min-max game

optimal discriminator

- Finding the best generator

$$G^* = argmin_G \left( \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[log(D^*(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim p_{model}(\mathbf{x})}[log(1 - D^*(\mathbf{x}))] \right)$$

$$= argmin_G \left( \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[log\frac{p_{data}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_{model}(\mathbf{x})}\right] + \mathbb{E}_{\mathbf{x}\sim p_{model}(\mathbf{x})}\left[log\frac{p_{model}(\mathbf{x})}{p_{data}(\mathbf{x}) + p_{model}(\mathbf{x})}\right] \right)$$

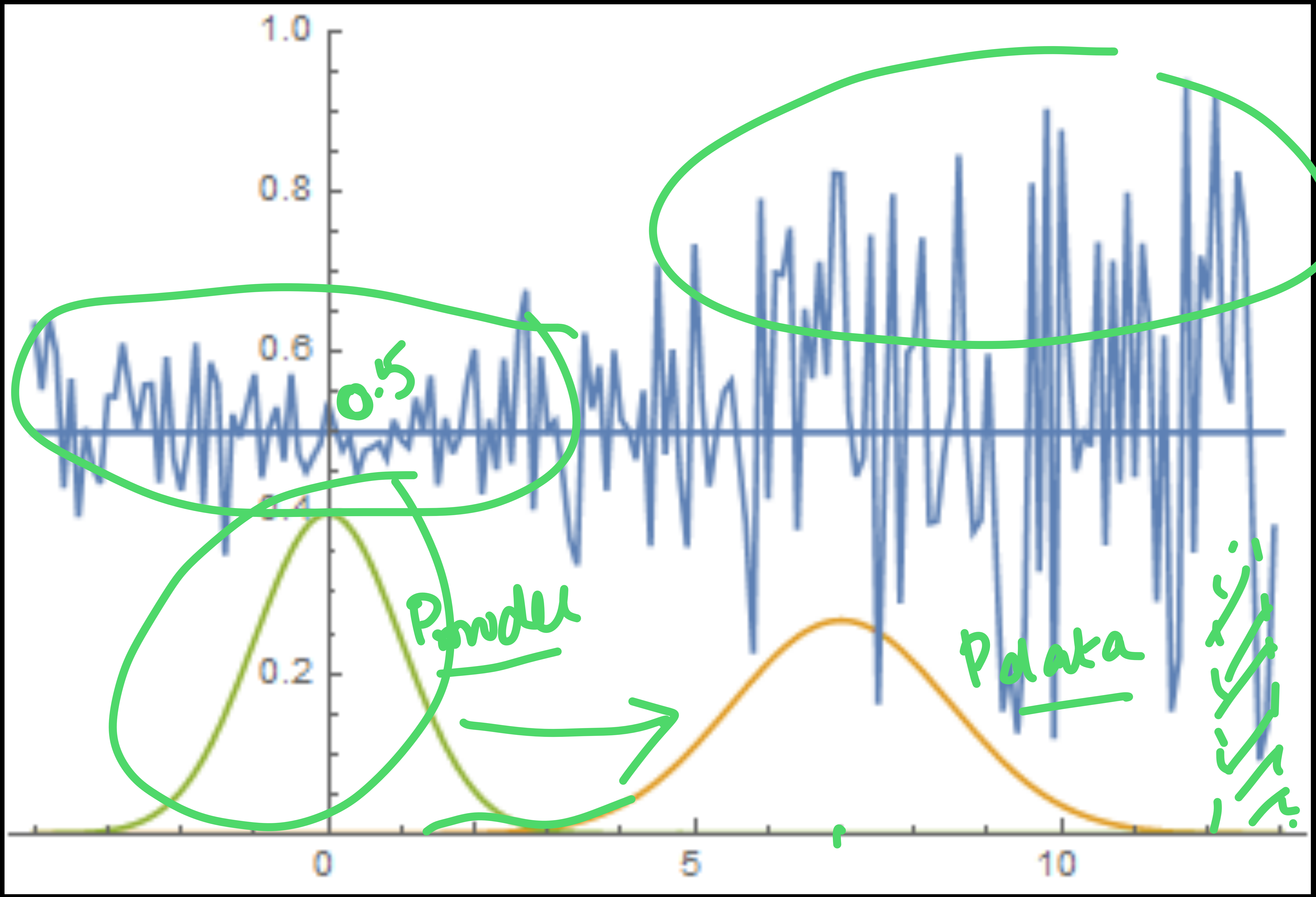
$$= argmin_G \left( KL\left(p_{data}(\mathbf{x})||\left(\frac{p_{data}(\mathbf{x}) + p_{model}(\mathbf{x})}{2}\right)\right) + KL\left(p_{model}(\mathbf{x})||\left(\frac{p_{data}(\mathbf{x}) + p_{model}(\mathbf{x})}{2}\right)\right) + 2log2 \right)$$

$$= argmin_{p_{model}(\mathbf{x})} \; \mathbb{D}_{JS}(p_{data}(\mathbf{x})||p_{model}(\mathbf{x}))$$

$p_{data} = p_{model}$

Jenson-Shannon divergence.

# GAN - Simple example

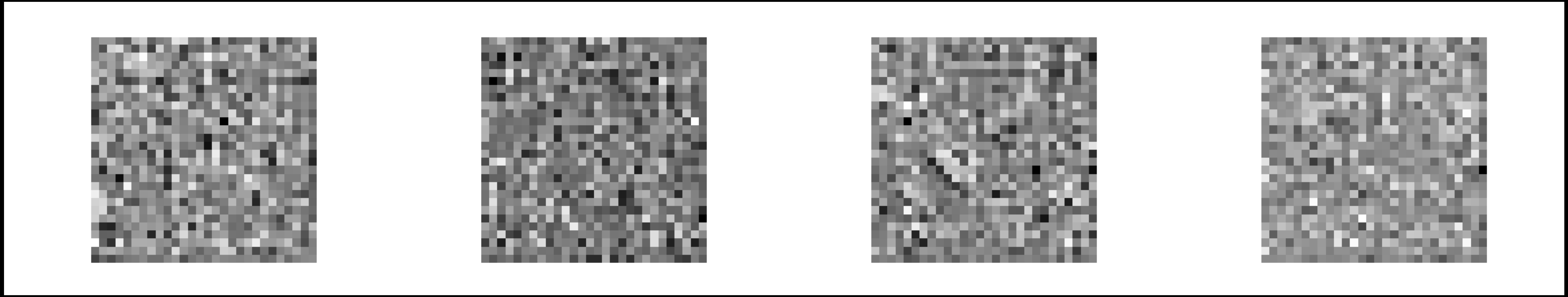



— discrimato
o/p

— Density
fn
of generato

# Generative adversarial networks - examples

- MNIST digits



$$\boxed{D(x) < 1/2}$$

# Deep convolutional GANs

(DCGANs)


deconvolution

convolution

x'

x'

# Conditional GANs

- Give the generator one hot encodings of classes (features)
  - This can also contain some attributes of the image.
  - Model learns the association between the images and the labels

x – real

$x^*$

1
0
:
0

# Conditional GANs

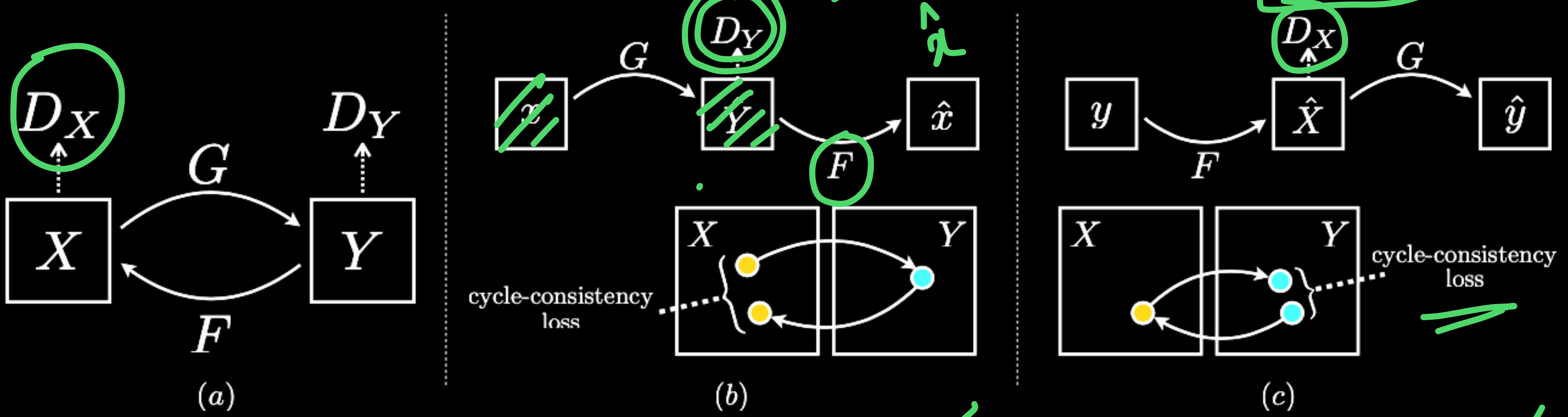
z

0

# Cycle GANs - Image to Image translation

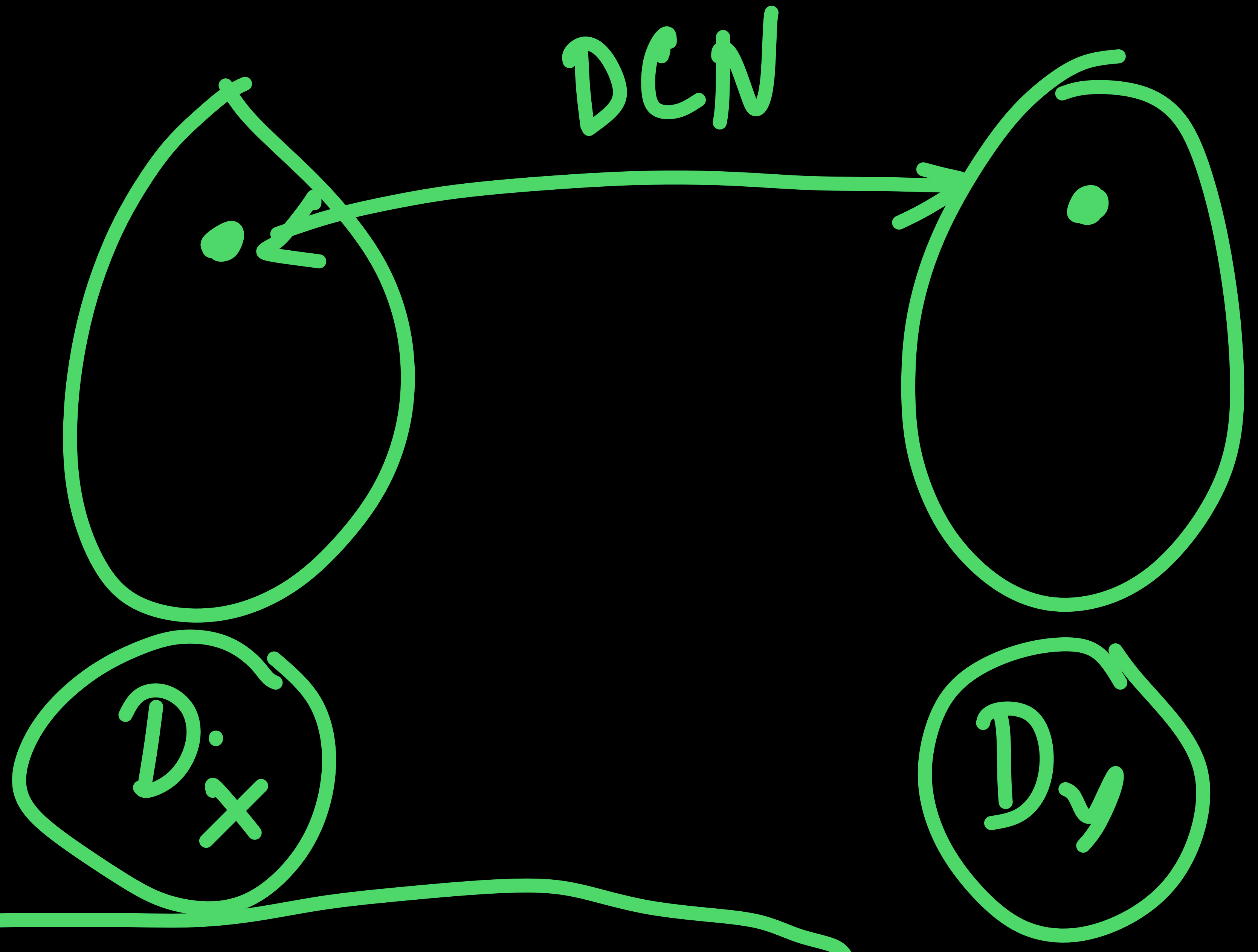
Discriminator

reverse

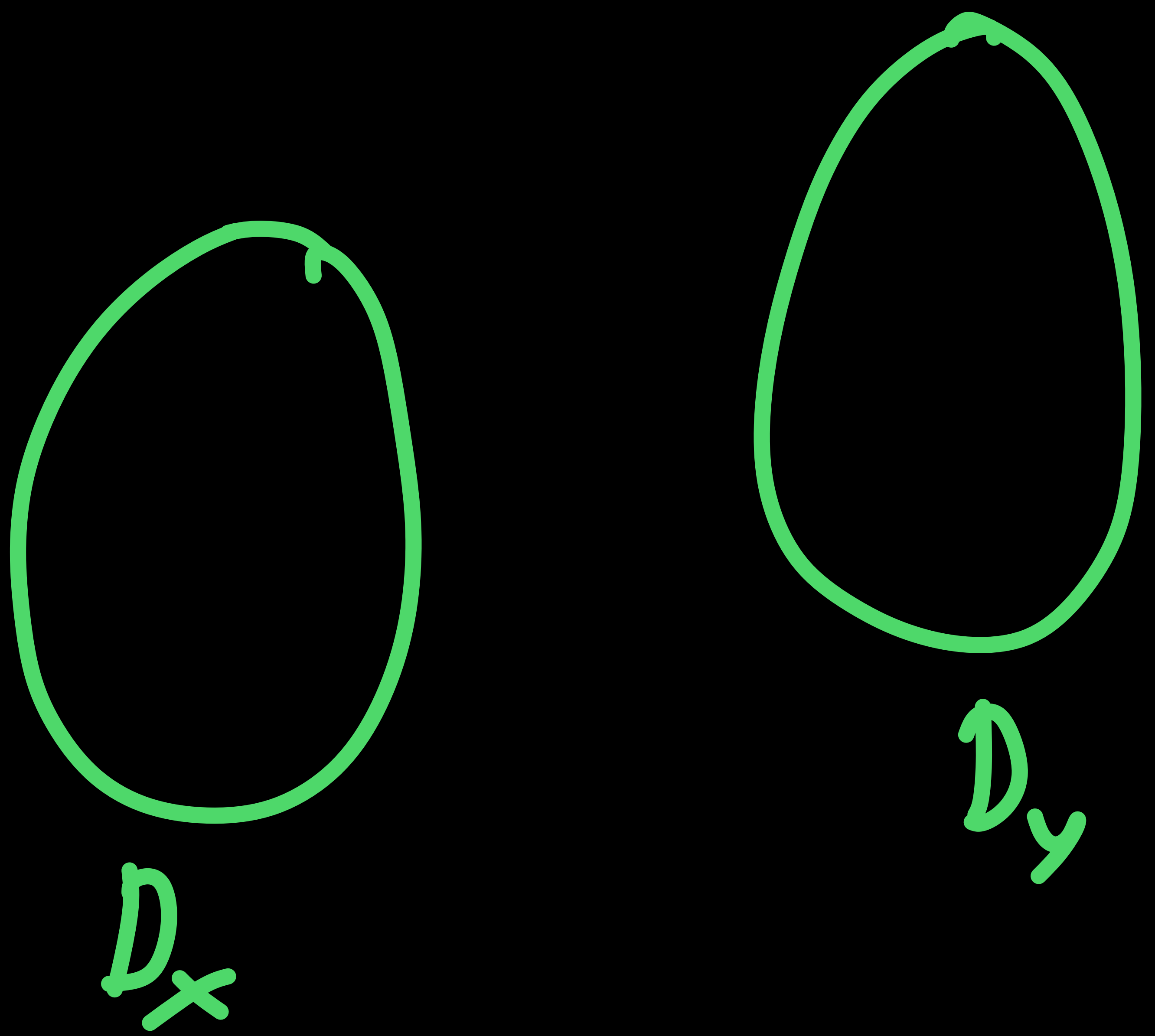



$D_x$ , $D_y$

# Paired Translation




# Unpaired translation



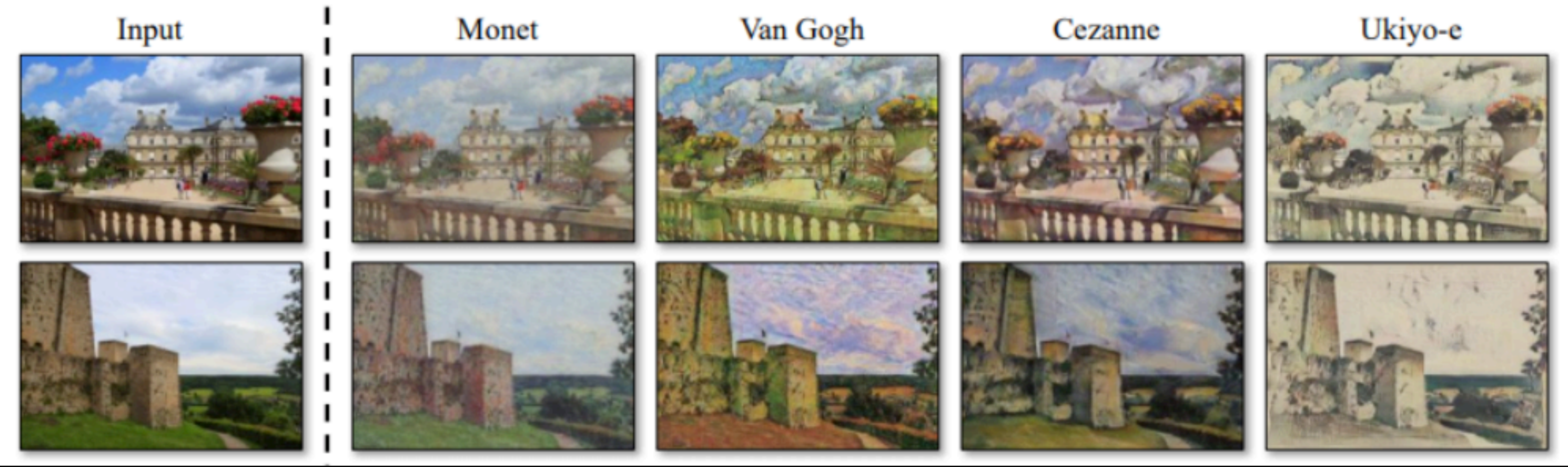
# Cycle GANs

painting style

horse to zebras

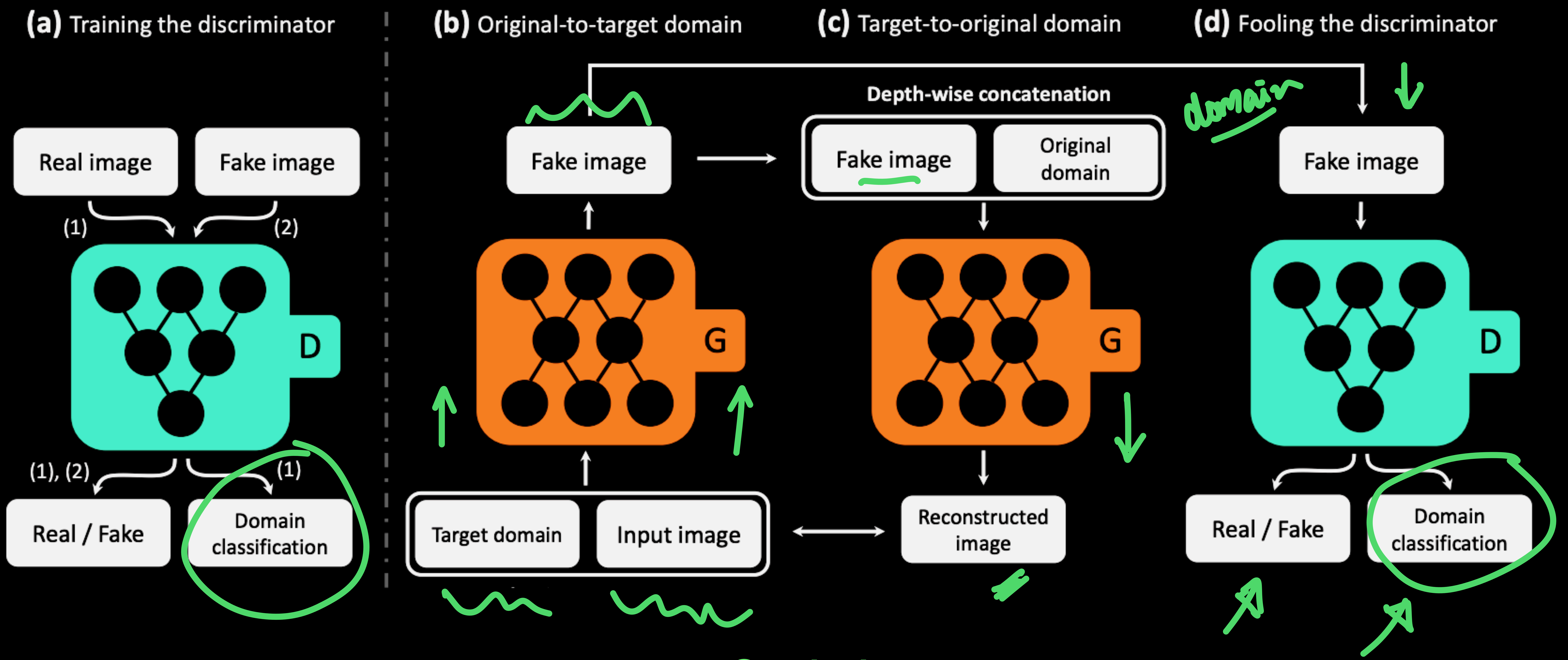
season change

# Cycle GANs - test data



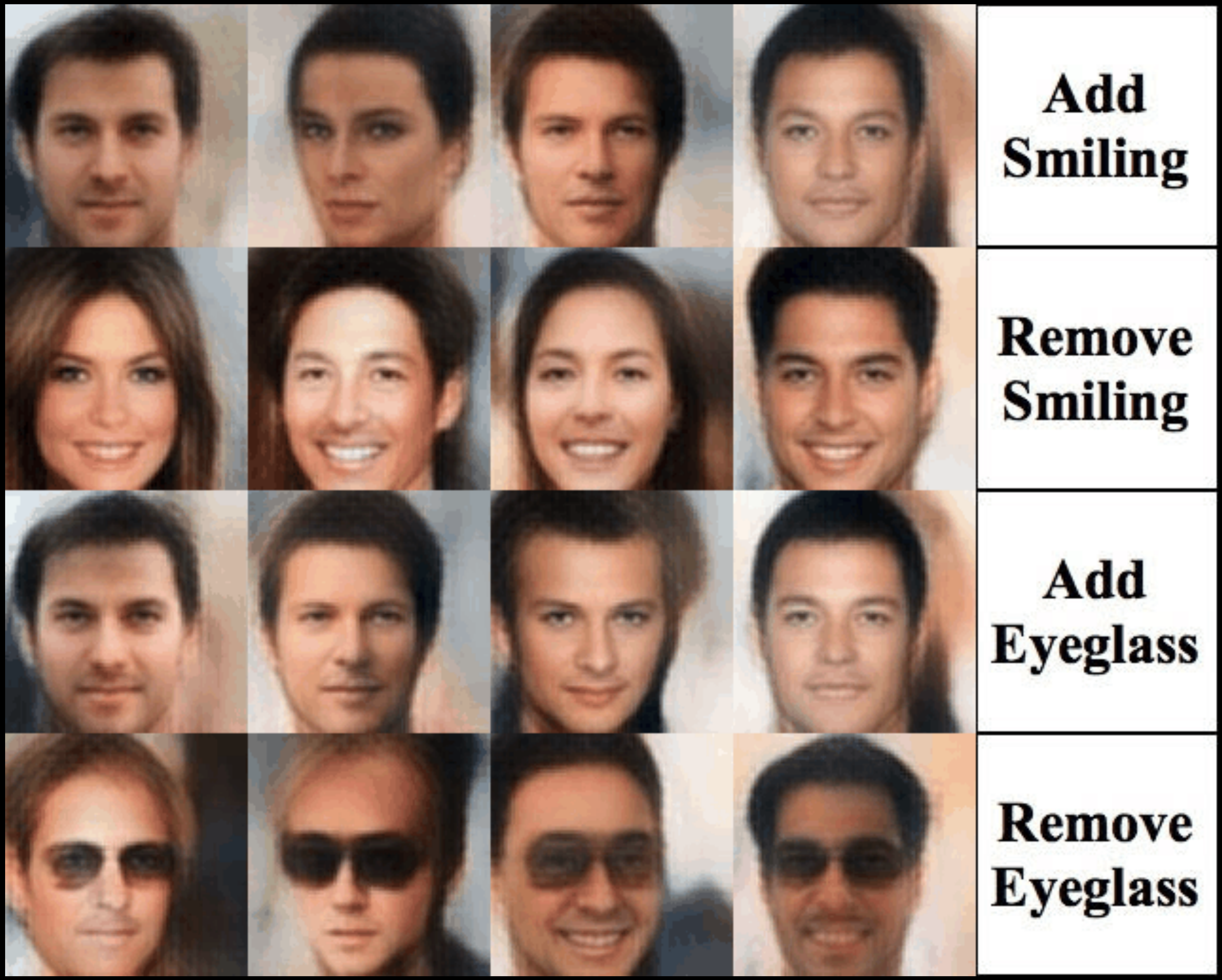
# Cycle GANs - test data

# StarGANs




Cycle loss

# StarGANs



# Comparing unsupervised learning methods

| | RBM | VAE | GAN |
|---|---|---|---|
| DISTRIBUTION | EXPLICIT | EXPLICIT CONDITIONAL DISTRIBUTION | IMPLICIT |
| LEARNING | APPROXIMATE ML | APPROXIMATE ML | DIRECT OPTIMIZATION |
| LATENT VARIABLE | BINARY | REAL | REAL |
| QUALITY | OKAY | MODERATE | GOOD |

# Topics thus far …

**Visual and Time Series Modeling:** Semantic Models, Recurrent neural models and LSTM models, Encoder-decoder models, Attention models.

**Representation Learning, Causality And Explainability:** t-SNE visualization, Hierarchical Representation, semantic embeddings, gradient and perturbation analysis, Topics in Explainable learning, Structural causal models.

**Unsupervised Learning:** Restricted Boltzmann Machines, Variational Autoencoders, Generative Adversarial Networks.

**New Architectures:** Capsule networks, End-to-end models, Transformer Networks.

**Applications:** Applications in in NLP, Speech, Image/Video domains in all modules.