KISTI 한국과학기술정보연구원

2022

Advanced Topic

in Research Data-centric Deep Learning

Lec 13: Generative Adaversal Networks : GAN

hsyi@kisti.re.kr

Hongsuk Yi (이홍석)

Reviewing the last class: AutoEncoder

Structured Learning and Generative Modeling

- Machine Learning is to find a function f
- Supervised
  - ✓ Data – X , y
  - ✓ Goal – Learn mapping from X -> Y
- Un-Supervised
  - ✓ Data – X
  - ✓ Goal – Learn Hidden structure of data
- Output
  - ✓ Regression
  - ✓ Classification
  - ✓ Prediction/Structure Learning

$$f : X \rightarrow Y$$

<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

Maximum Likehood Models

KISTI SINCE 1962

Maximum Likehood Models

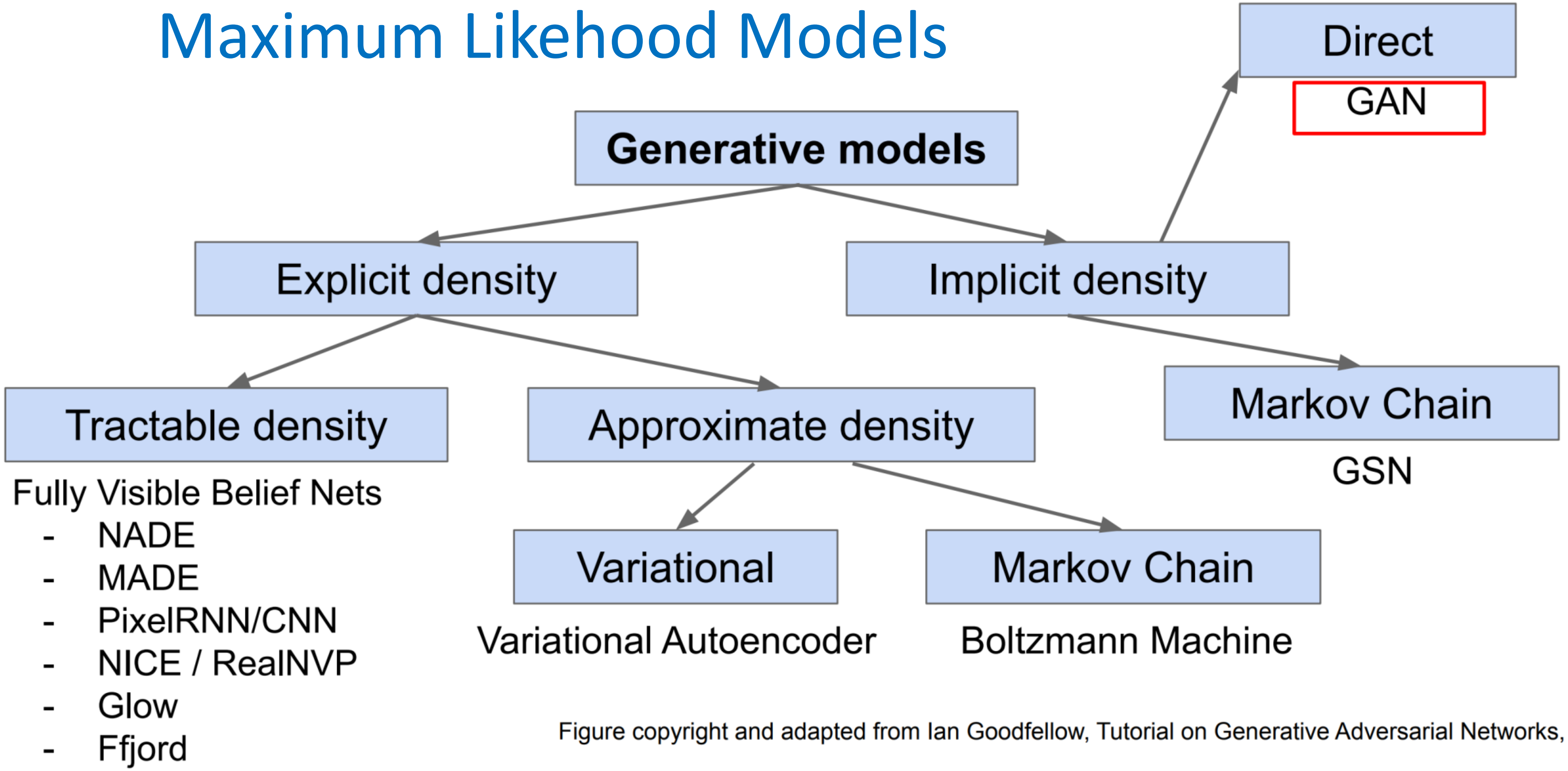


Figure copyright and adapted from Ian Goodfellow, Tutorial on Generative Adversarial Networks, 2017.

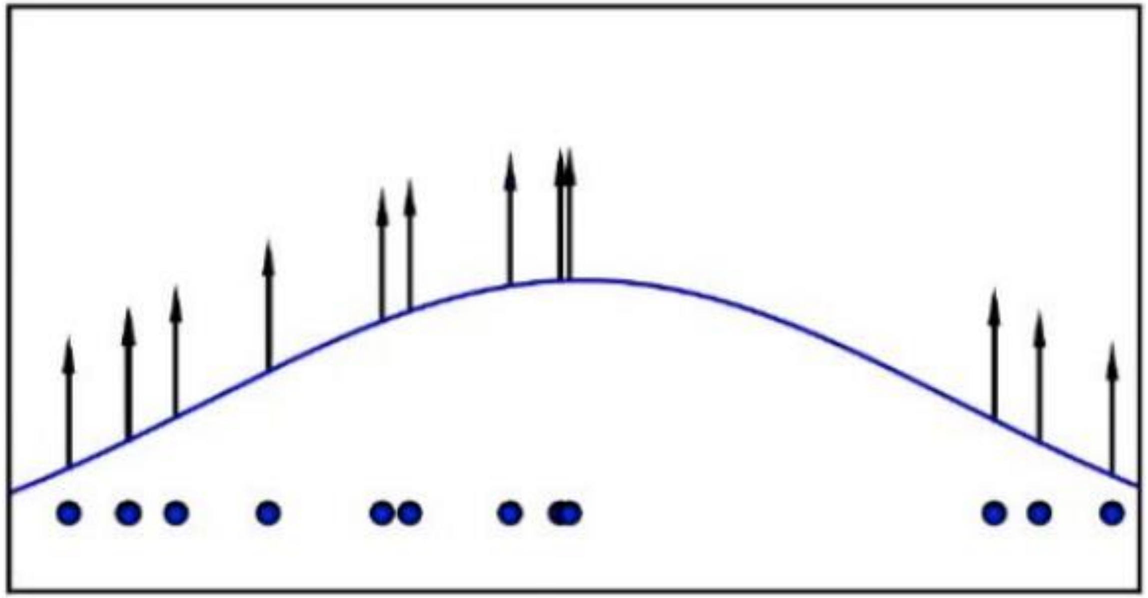
<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

- Maximum likelihood tries increase the likelihood of data given the parameters

P(x)

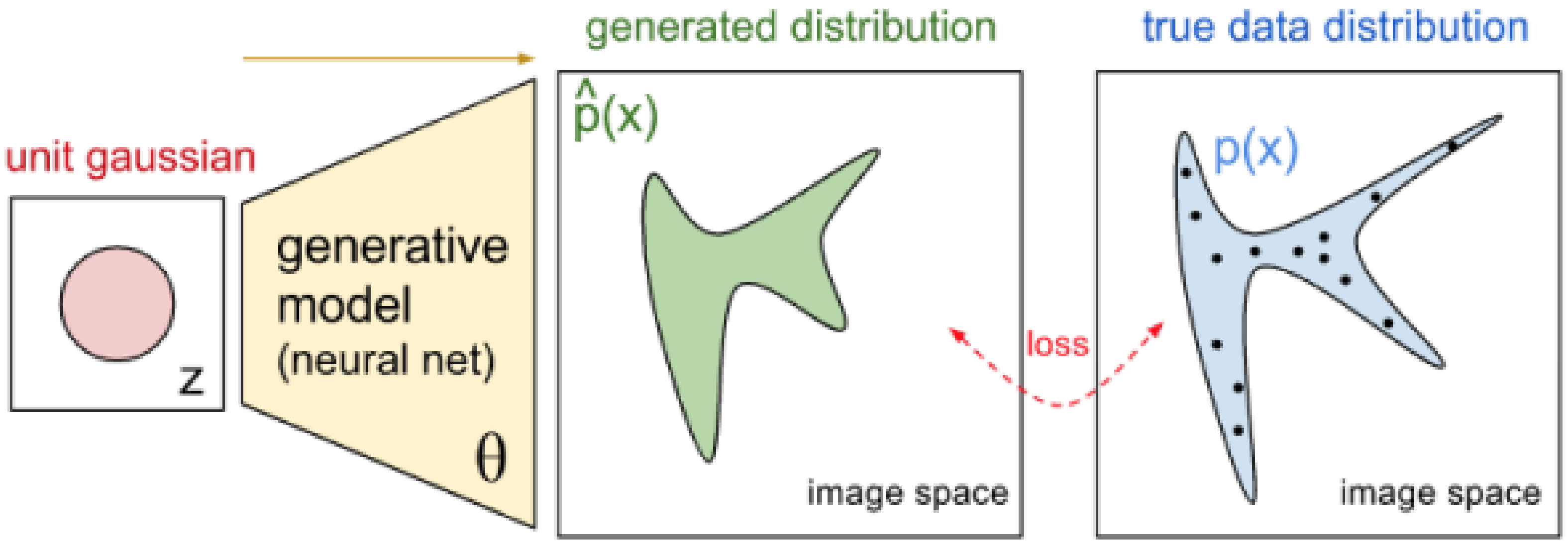
$$\theta^* = \arg\max_{\theta}$$

$$E_{x \sim Pdata} \log P\,(x/\theta)$$

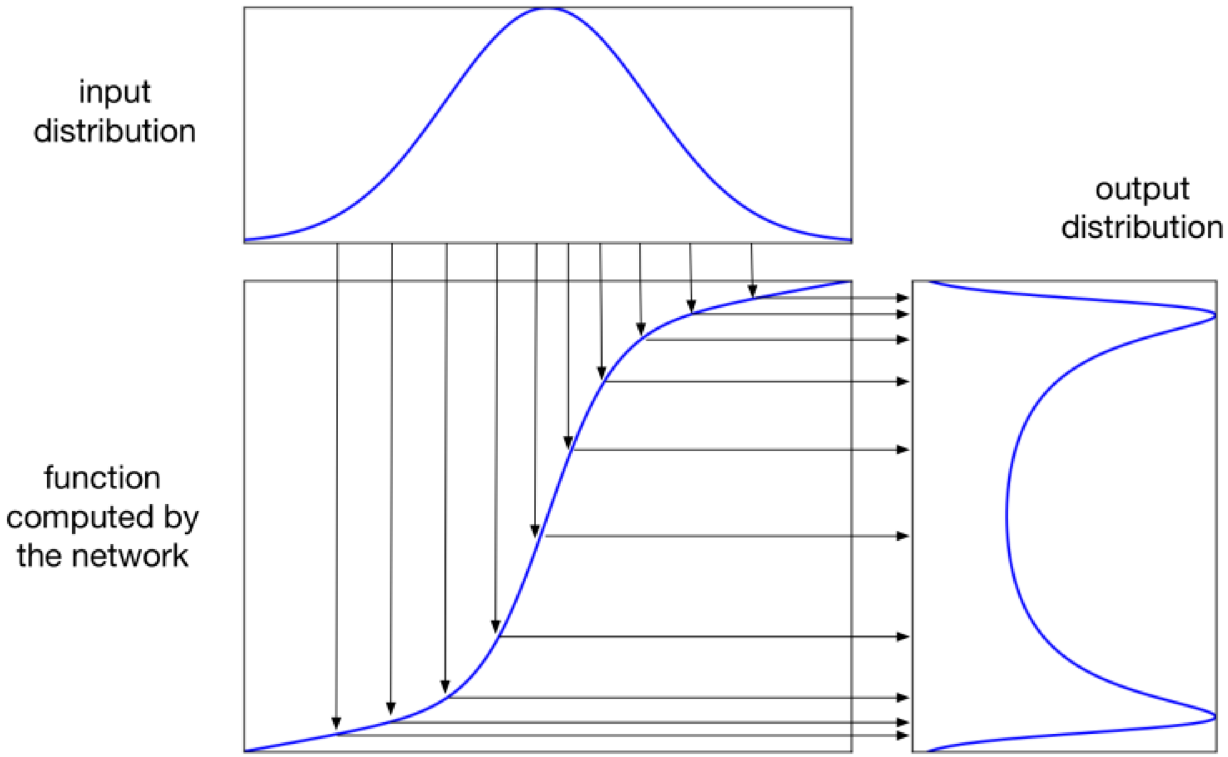


<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

- **Generative models define a probability distribution**
  - ✓ Start by sampling the code vector z from a fixed, simple distribution
  - ✓ The generator network computes a differentiable function G mapping z to an x in data space



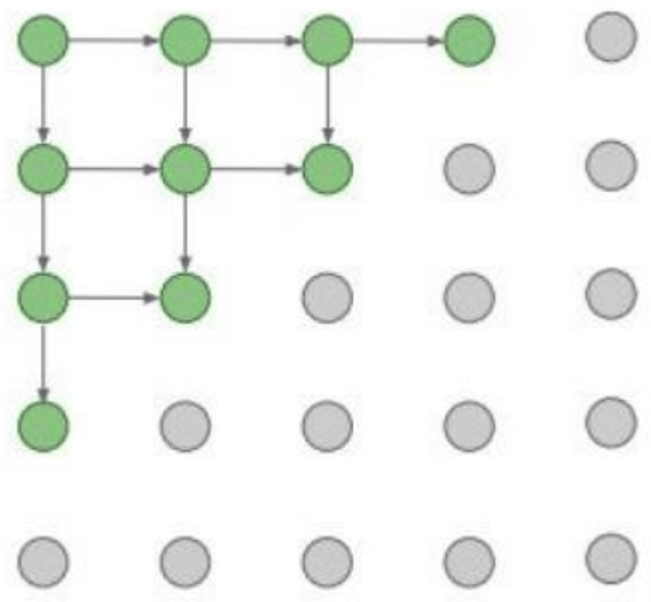
- 1Dimensional example



<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

Tractable Model - PixelRNN / PixelCNN

- Generate image pixels from corner
- Training Faster
- Generation Slow / Sequential
- Cannot generate samples based on some latent code



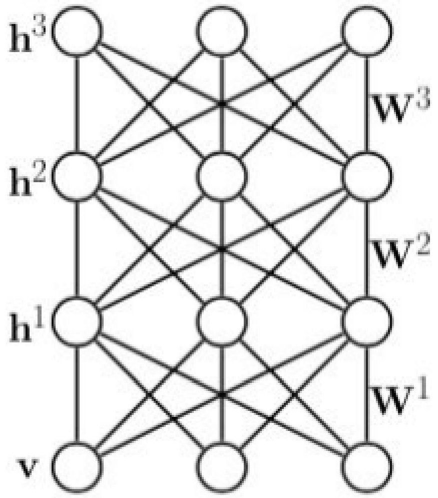
$$p(x) = \prod_{i=1}^{n} p(x_i \mid x_1, x_2, \ldots, x_{i-1})$$

Chain Rule

Maximum Likelihood based Training

<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

Boltzmann Machine



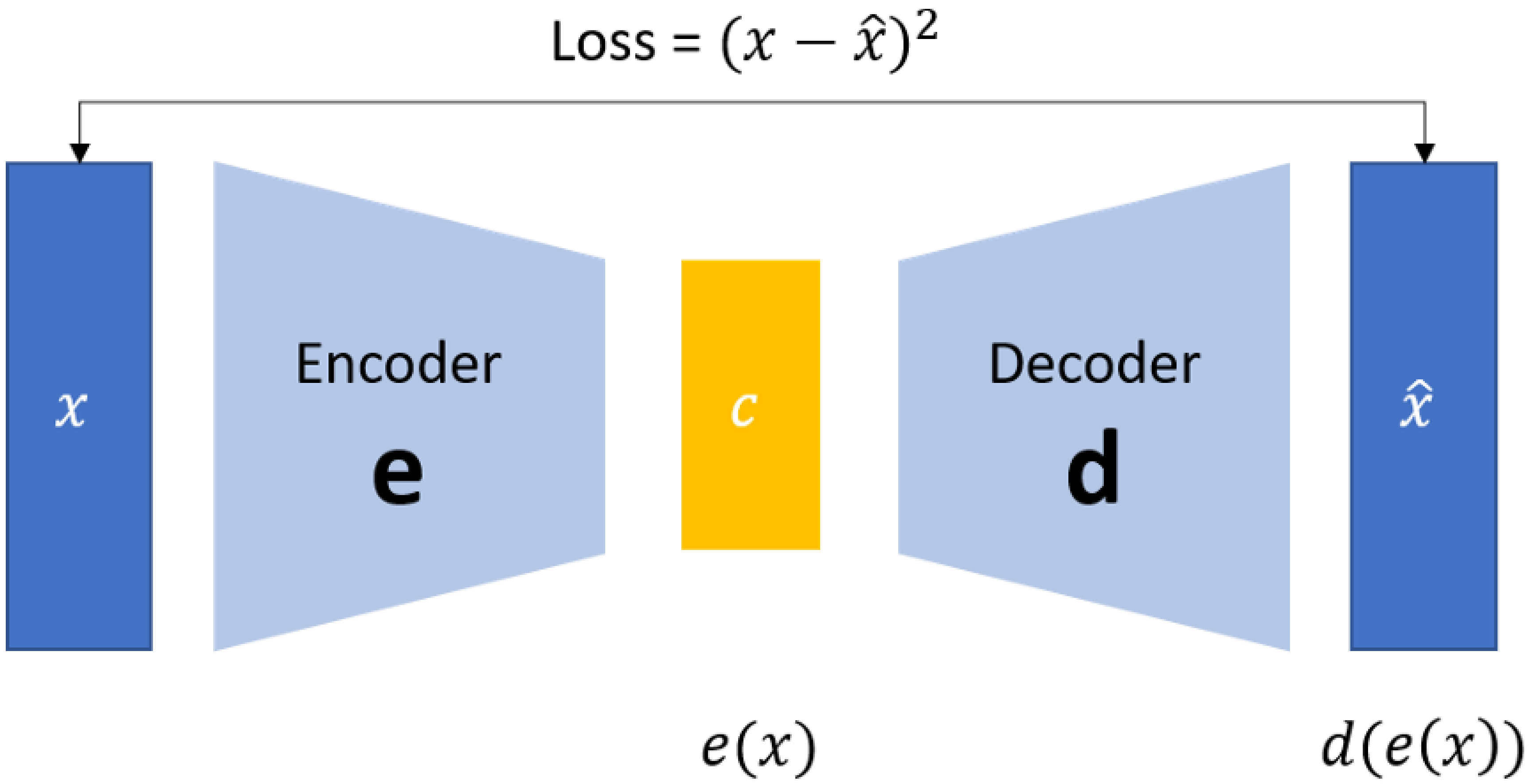
- Energy Function based models
- Markov chains don't work for long sequences
- Hard to scale on large dataset

$$p(x,h) = \exp(\,-E(x,h))\ |\ Z$$

$$Z = \sum_{x,h} \exp(\,-E(x,h))$$

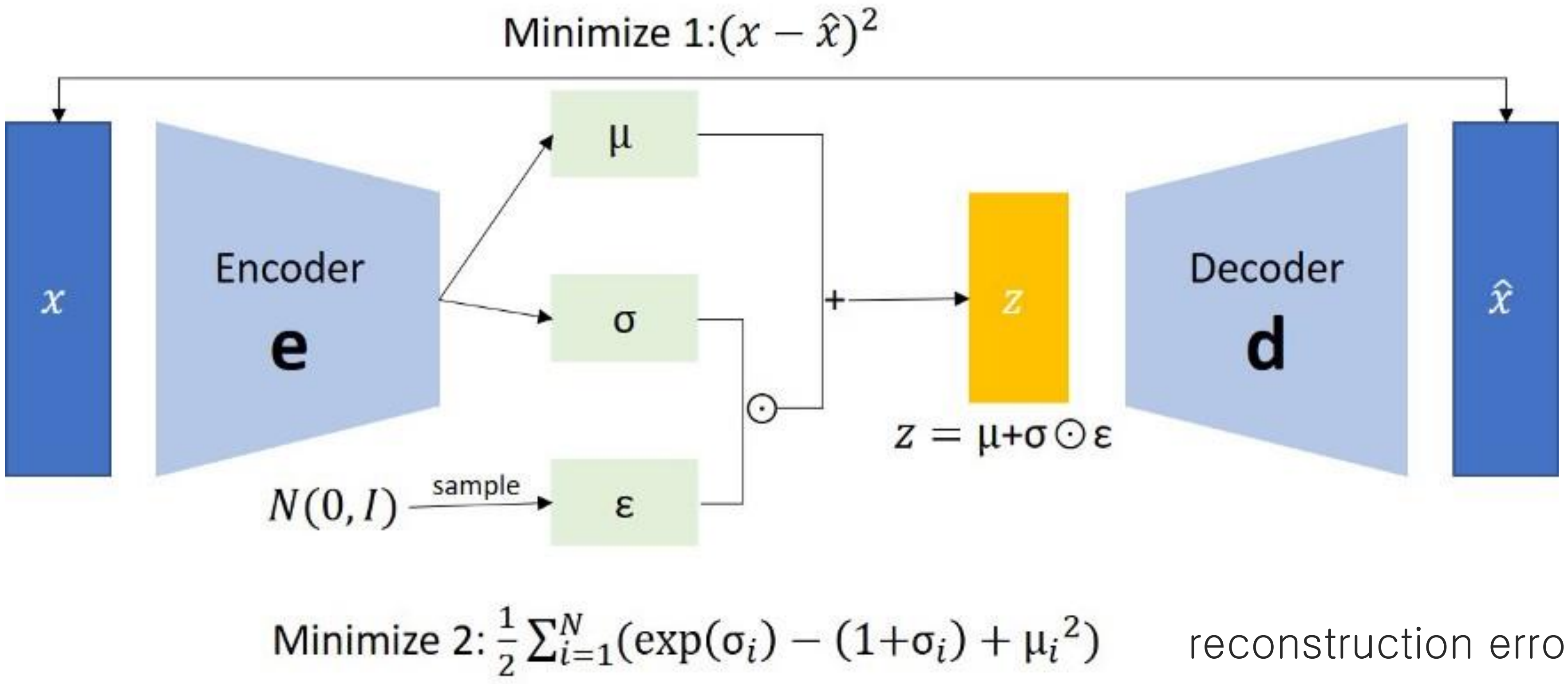
<source> https://people.cs.pitt.edu/~milos/courses/cs3750/lectures/class23.pdf

Non Tractable Model - Variational Auto-encoder



https://medium.com/geekculture/variational-autoencoder-vae-9b8ce5475f68

Variational Auto-encoder : VAE



reconstruction error

Reparameterization trick $z = \mu + \sigma \odot \varepsilon$

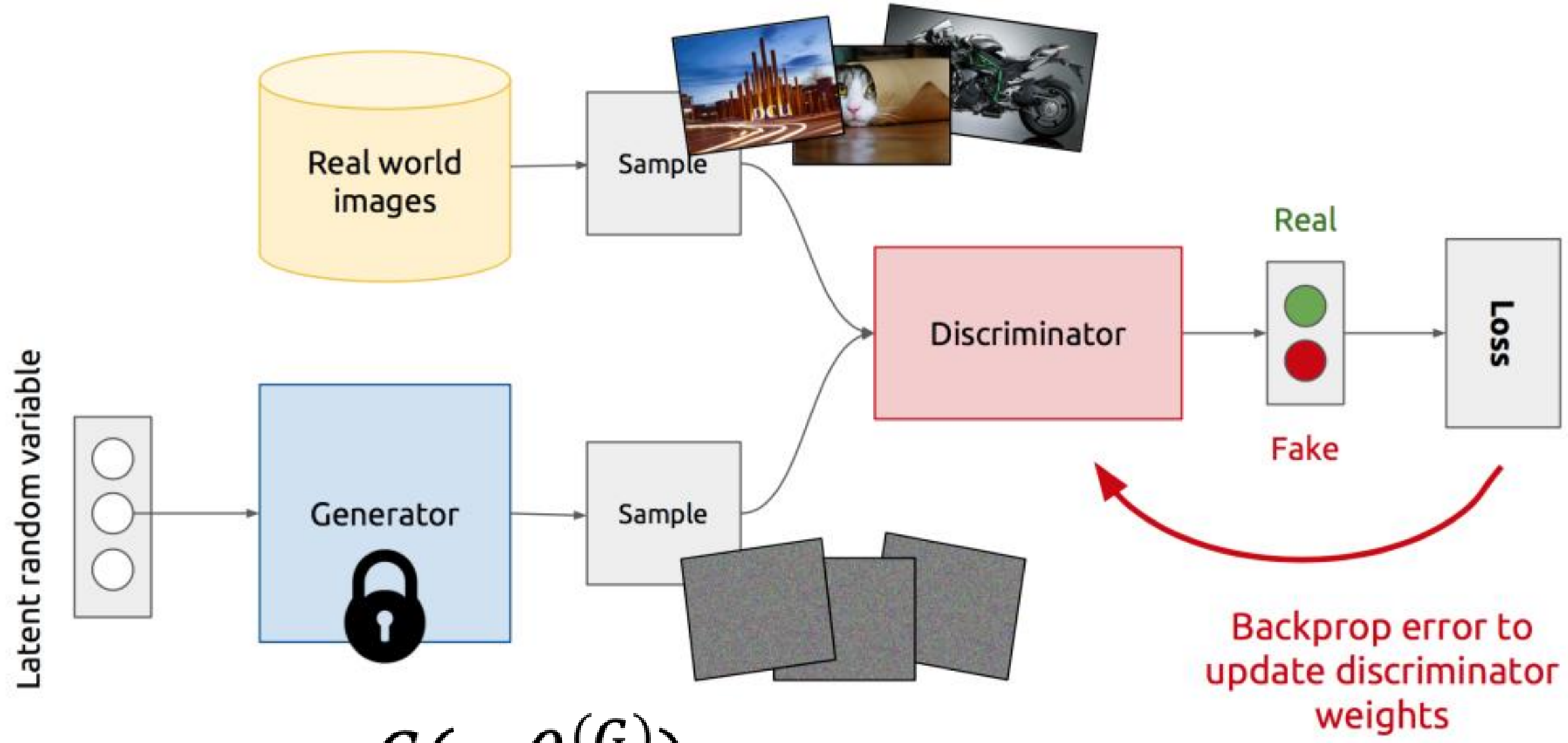
Generative Adversarial Networks (GANs)

Advantages of GANs

- Plenty of existing work on Deep Generative Models
  - ✓ Boltzmann Machine
  - ✓ Deep Belief Nets
  - ✓ Variational AutoEncoders (VAE)

- Why GANs?
  - ✓ Sampling (or generation) is straightforward
  - ✓ Training doesn't involve Maximum Likelihood estimation
  - ✓ Robust to Overfitting since Generator never sees the training data
  - ✓ Empirically, GANs are good at capturing the modes of the distribution.

➢ Classic GAN Framework



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

Adversarial Training

- GANs extend that idea to generative models:
  - ✓ generate adversarial samples to fool a discriminative model
  - ✓ use those adversarial samples to make models robust
  - ✓ Repeat this and we get better discriminative model
- Generator:
  - ✓ generate fake samples, tries to fool the Discriminator
- Discriminator:
  - ✓ tries to distinguish between real and fake samples
  - ✓ Train them against each other
  - ✓ Repeat this and we get better Generator and Discriminator

KISTI SINCE 1962

- D tries to identify real data from fakes created by the generator
- G tries to create imitations of data to trick the discriminator
- Objective function:

Train GAN jointly via **minimax** game:

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log \left(1 - D_{\theta_d} \left(G_{\theta_g}(z) \right) \right) \right]$$

Discriminator wants to maximize objective s.t. $D(x)$ close to 1, $D(G(z))$ close to 0.

Generator wants to minimize objective s.t. $D(G(z))$ close to 1.

Discriminator

- Discriminator is a function D (network, can deep)

$$D : X \to R$$

  - Input x: an object x (e.g. an image)
  - Output D(x): scalar which represents how “good” an object x is

Can we use the discriminator to generate objects?

Yes.

Discriminator

- Suppose we already have a good discriminator D(x) …

Inference

- Generate object $\tilde{x}$ that

$$\tilde{x} = \arg\max_{x \in X} D(x)$$

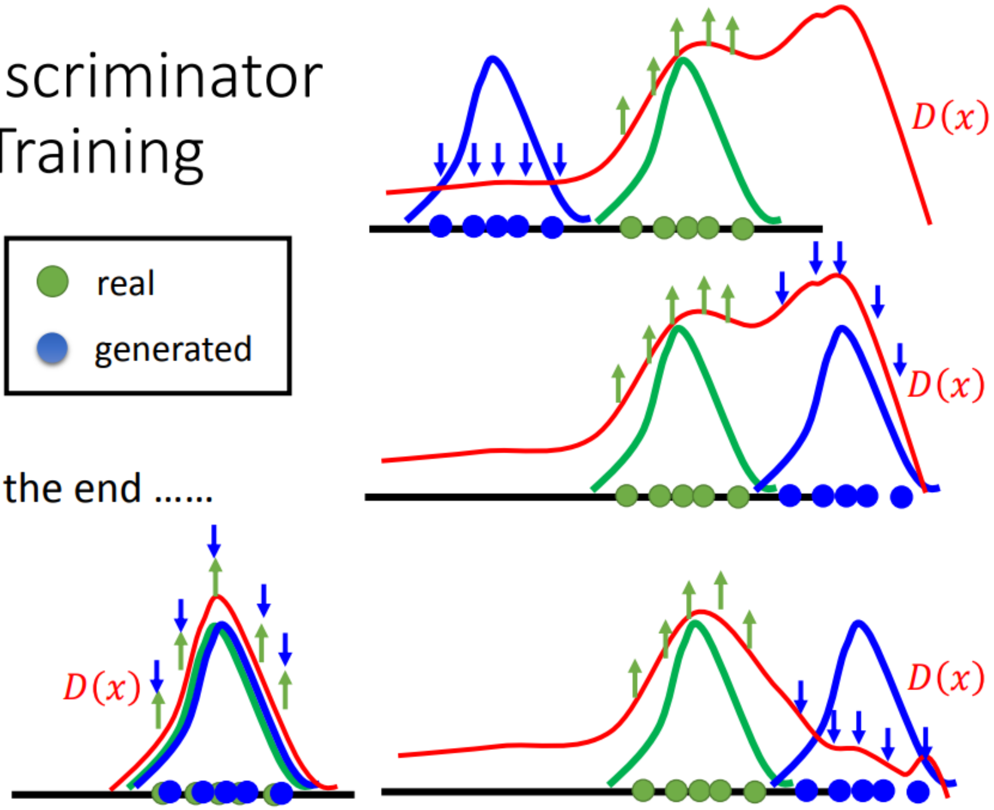
Enumerate all possible x !!!

It is feasible ???

How to learn the discriminator?

Discriminator - Training

Discriminator - Training



- **General Algorithm**

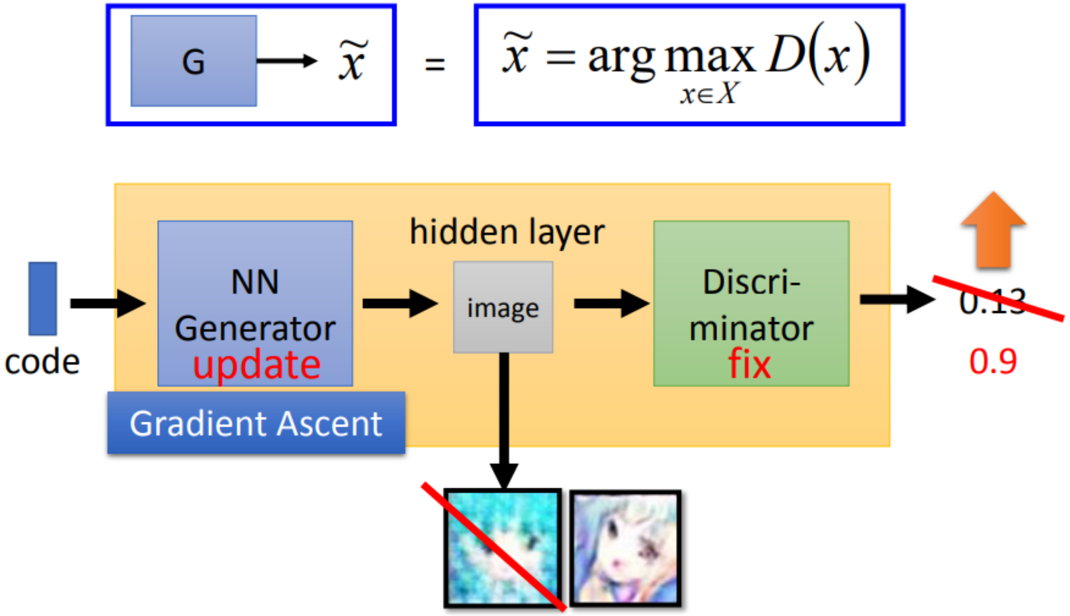
  - Given a set of positive examples, randomly generate a set of negative examples.

  - In each iteration
    - Learn a discriminator D that can discriminate positive and negative examples.

    - Generate negative examples by discriminator D

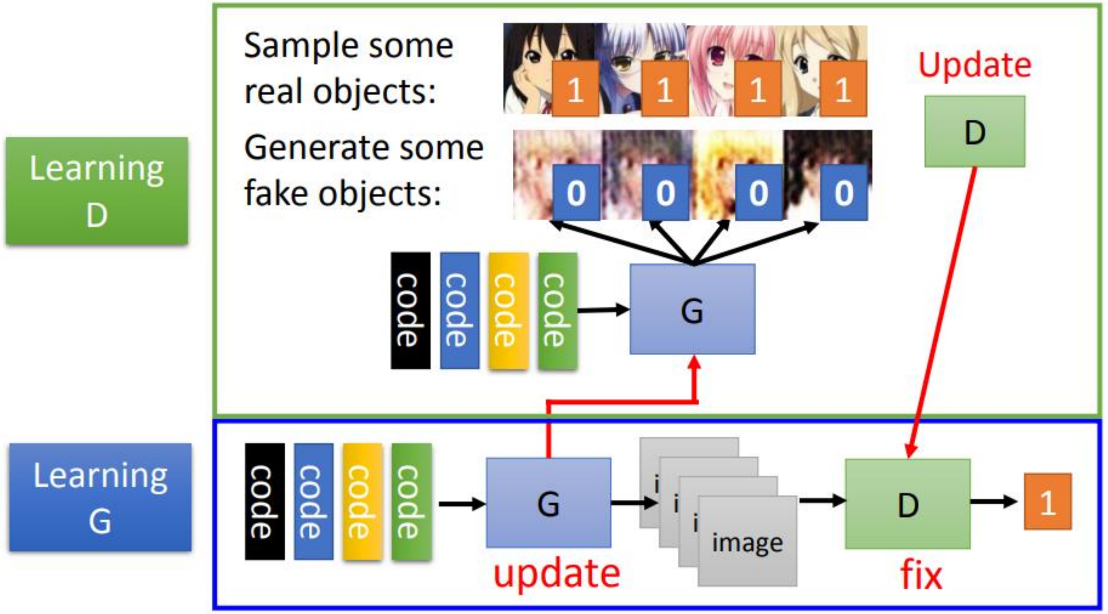
$$\boxed{G \rightarrow \tilde{x}} = \boxed{\tilde{x} = \arg\max_{x \in X} D(x)}$$

Generating Negative Examples

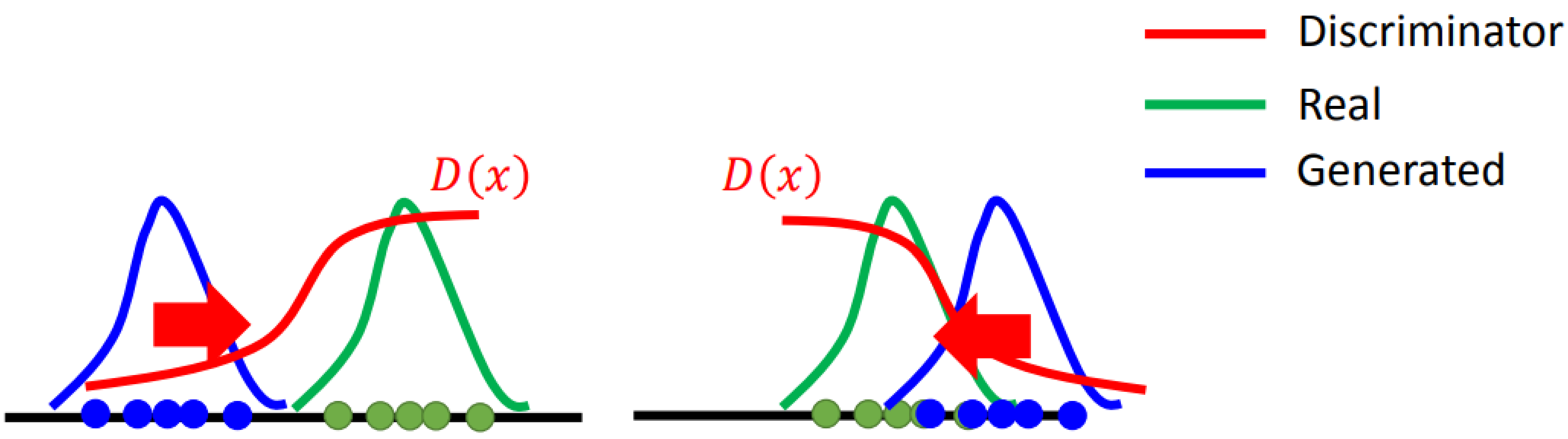


- Initialize generator and discriminator

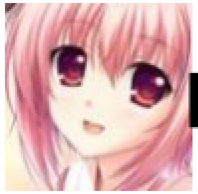
- In each training iteration:



- Discriminator leads the generator



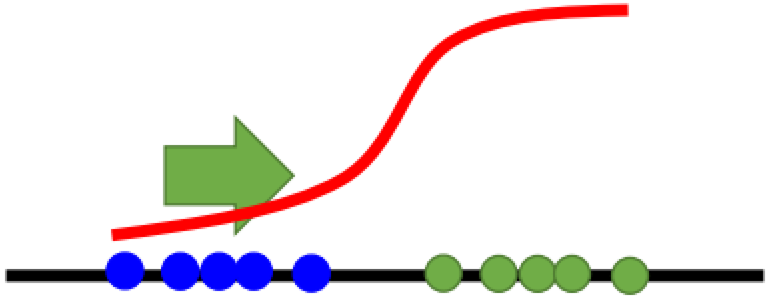
Binary Classifier as Discriminator

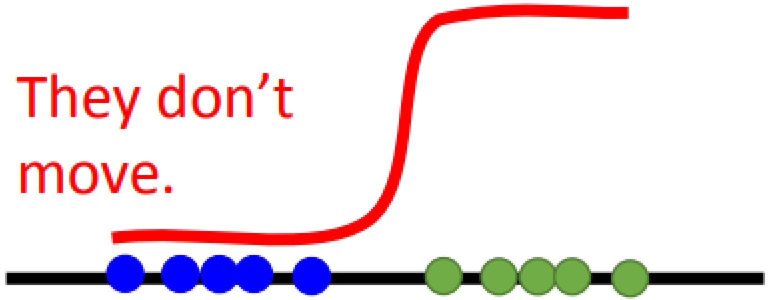
 → Binary Classifier → Real

 → Binary Classifier → Fake

Typical binary classifier uses sigmoid function at the output layer

1 is the largest, 0 is the smallest



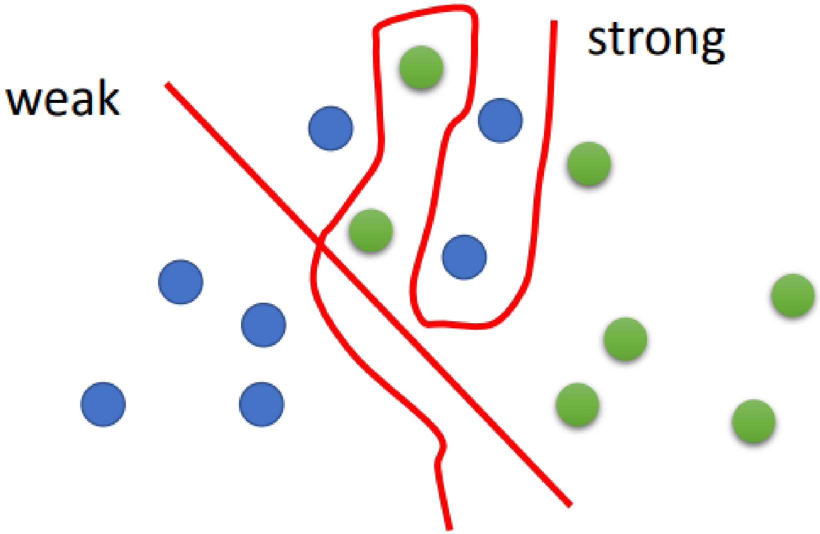


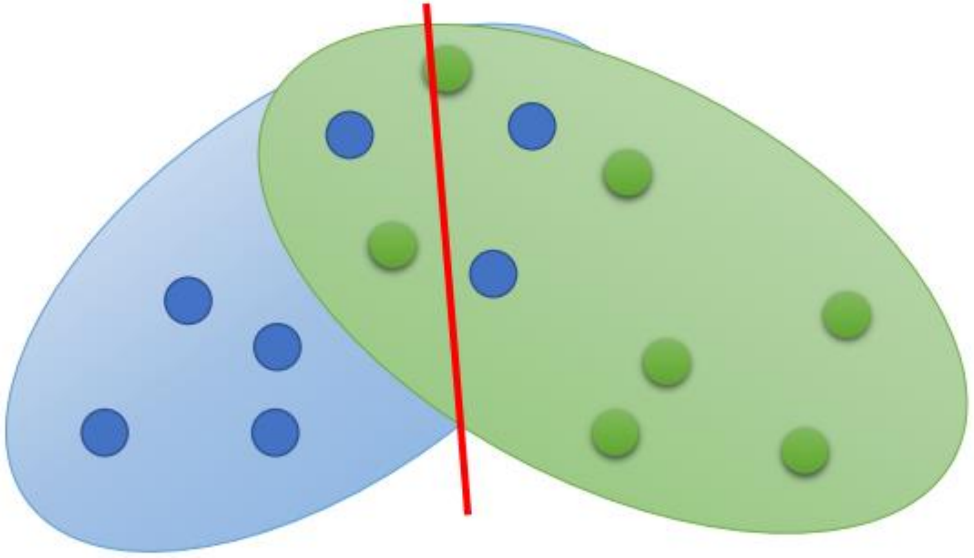
real
generated

You cannot train your classifier ***too good***……

Binary Classifier as Discriminator

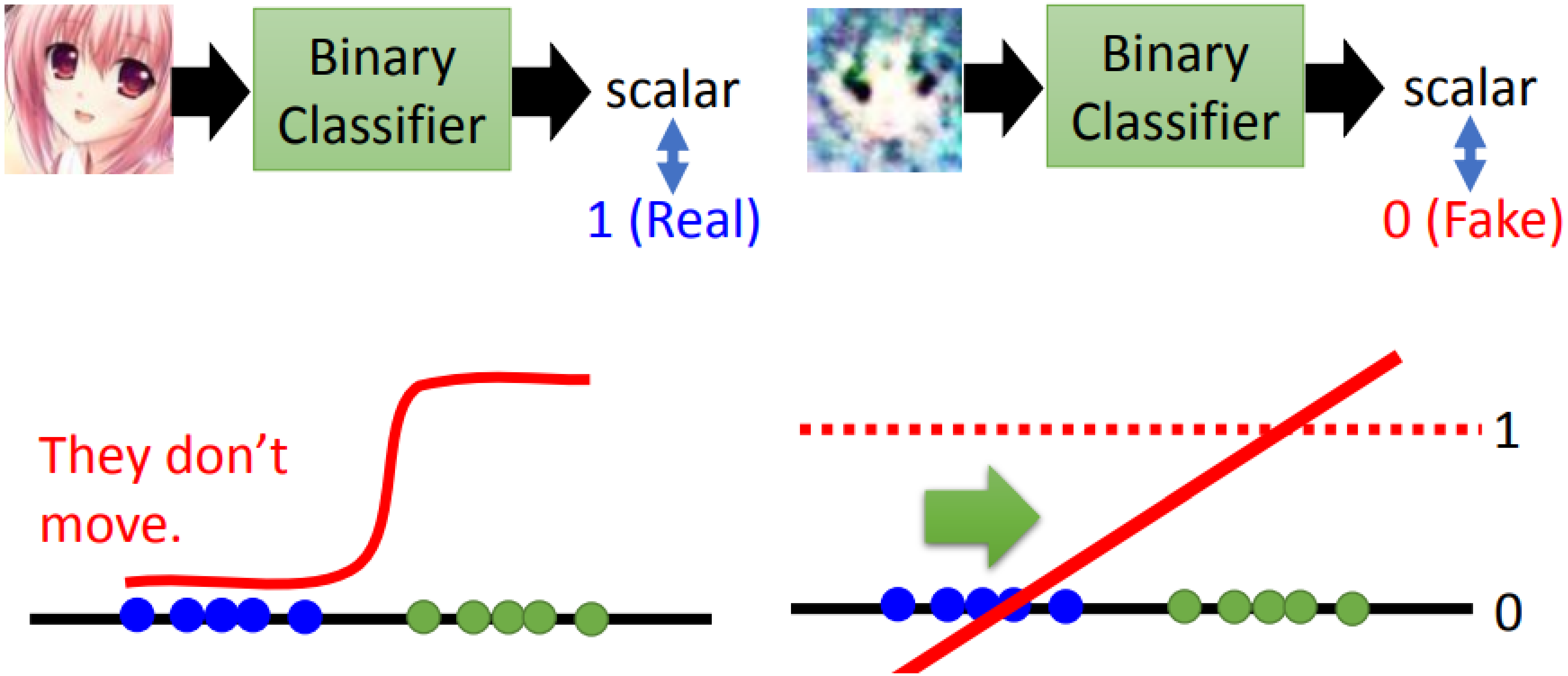
- Don't let the discriminator perfectly separate real and generated data
  - ✓ Add noise to input or label?





Least Square GAN (LSGAN)

- Replace sigmoid with linear (replace classification with regression)



The Difficulties of Training GANs

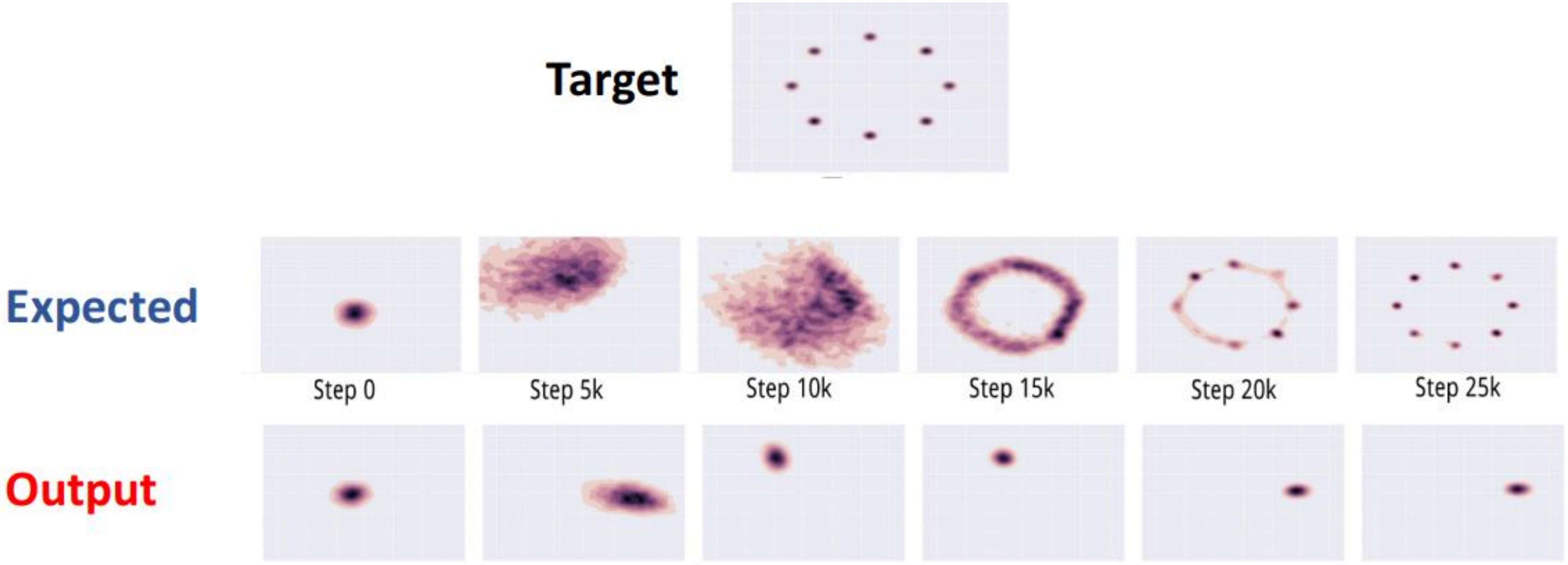
- zero-sum game:
  - ✓ During training, the generator and the discriminator constantly try to outsmart each other
- Nash equilibrium:
  - ✓ no player would be better offchanging their own strategy, assuming the other players do not change theirs
- The biggest difficulty : mode collapse
  - ✓ the generator’s outputs gradually become less diverse
- GANs are very sensitive to the hyperparameters:
  - ✓ you may have to spend a lot of effort fine-tuning them.

The Difficulties of Training GANs : Solutions

- Experience replay to avoid the mode collapse issue:
  - ✓ Google 2018 paper.
  - ✓ storing the images produced by the generator at each iteration in a replay buffer
- Mini-batch discrimination:
  - ✓ how similar images are across the batch and provides this statistic to the discriminator
  - ✓ it can easily reject a whole batch of fake images that lack diversity

Mode-Collapse and Solutions

- Generator fails to output diverse samples



- Basic Solutions:
  - ✓ Mini-Batch GANs

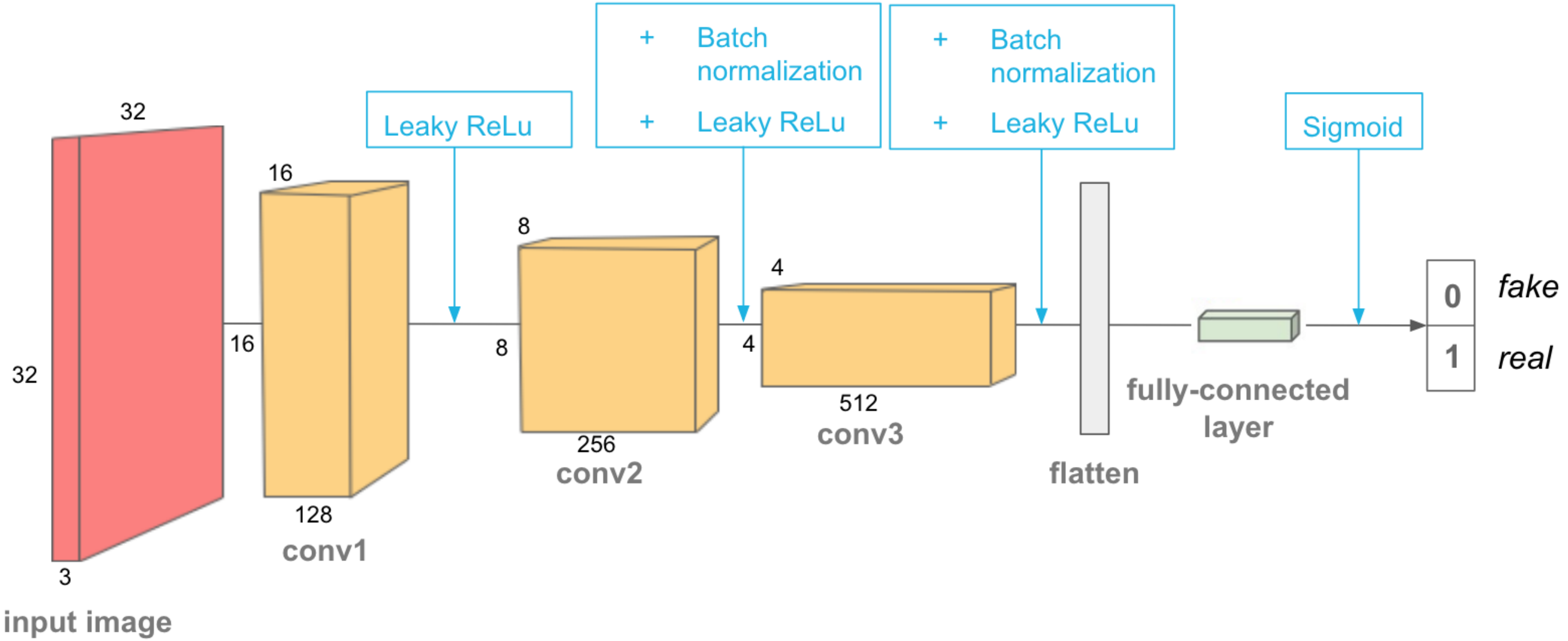
More Recent GANs

Deep Convolutional GANs

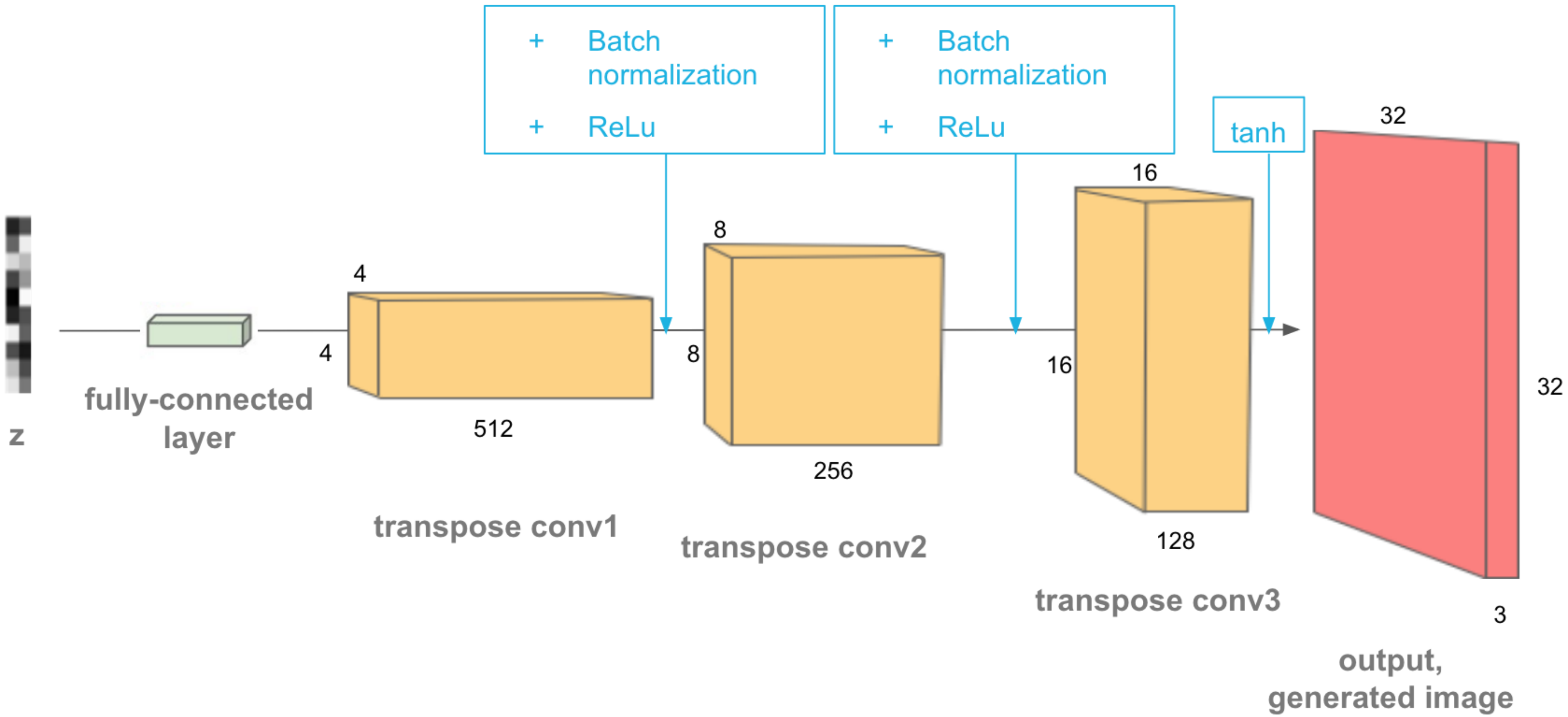
- The original GAN paper in 2014 experimented with convolutional layers:
  - ✓ But, only tried to generate small images
- DCGAN : Deep Convolutional GANs: 2015, Alec Radford
  - ✓ Replace any pooling layers with strided convolutions (in the discriminator) and transposed convolutions (in the generator).
  - ✓ Use Batch Normalization in both the generator and the discriminator, except in the generator’s output layer and the discriminator’s input layer.
  - ✓ Remove fully connected hidden layers for deeper architectures.
  - ✓ Use ReLU activation in the generator for all layers except the output layer, which should use tanh.
  - ✓ Use leaky ReLU activation in the discriminator for all layers.

➢ discriminator:

✓convolution > batch norm > leaky ReLU.



- The generator:
  - ✓transpose convolution > batch norm > ReLU.

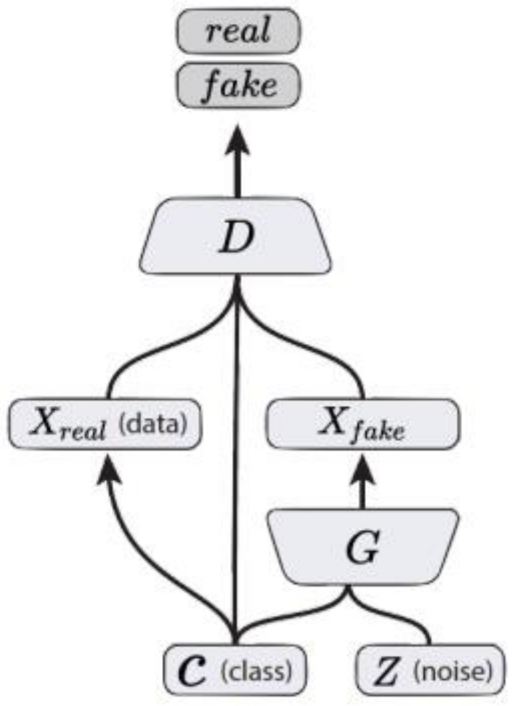


- Images generated by the DCGAN after 50 epochs of training
  - ✓ Fashion MNIST

Figure 17-17. Images generated by the DCGAN after 50 epochs of training

Conditional GANs

- Simple modification to the original GAN framework that conditions the model on additional information for better multi-modal learning
- Lends to many practical applications of GANs when we have explicit supervision available.



Conditional GAN
(Mirza & Osindero, 2014)

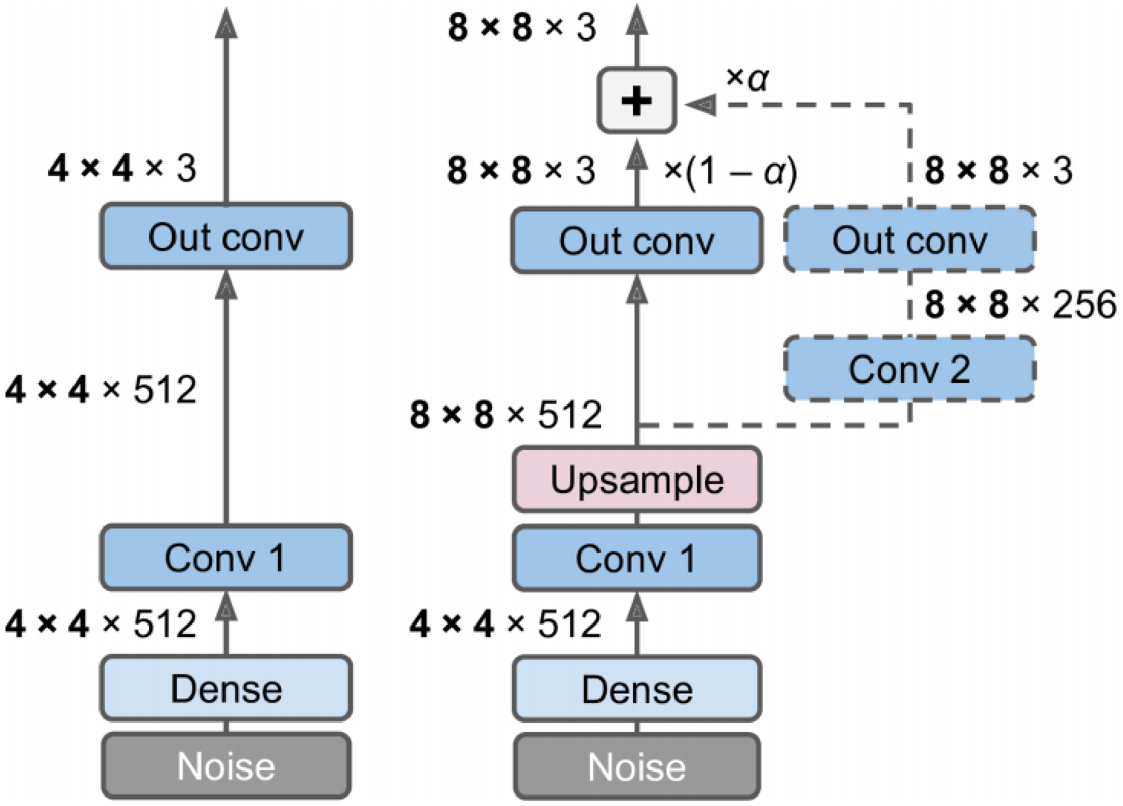
Conditional GANs

MNIST digits generated conditioned on their class label.

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] →
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0] →
[0, 0, 1, 0, 0, 0, 0, 0, 0, 0] →
[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] →
[0, 0, 0, 0, 1, 0, 0, 0, 0, 0] →
[0, 0, 0, 0, 0, 1, 0, 0, 0, 0] →
[0, 0, 0, 0, 0, 0, 1, 0, 0, 0] →
[0, 0, 0, 0, 0, 0, 0, 1, 0, 0] →
[0, 0, 0, 0, 0, 0, 0, 0, 1, 0] →
[0, 0, 0, 0, 0, 0, 0, 0, 0, 1] →

Figure 2 in the original paper.

- Progressive Growing of GANs: Nivida Tero Karras, in 2018 paper
  - ✓ a GAN generator outputs 4 × 4 color images (left)
  - ✓ extend it to output 8 × 8 images (right)



StyleGANs

- StyleGAN: Nvidia team
  - ✓ the state of the art in high-resolution image generation in 2018
  - ✓ style transfer techniques in the generator
- Adaptive Instance Normalization (AdaIN) :
  - ✓ each noise layer is followed by an AdaIN

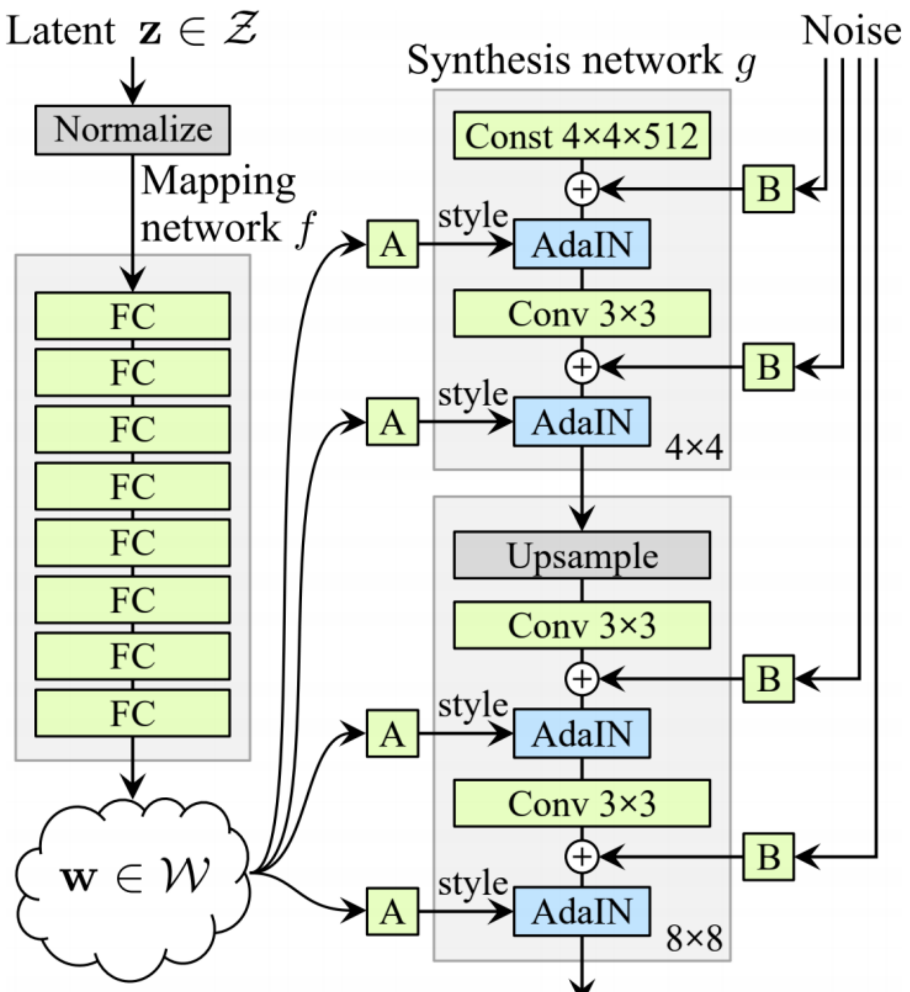


Figure 17-20. StyleGAN's generator architecture (part of figure 1 from the StyleGAN paper)[19]

Pix2pix

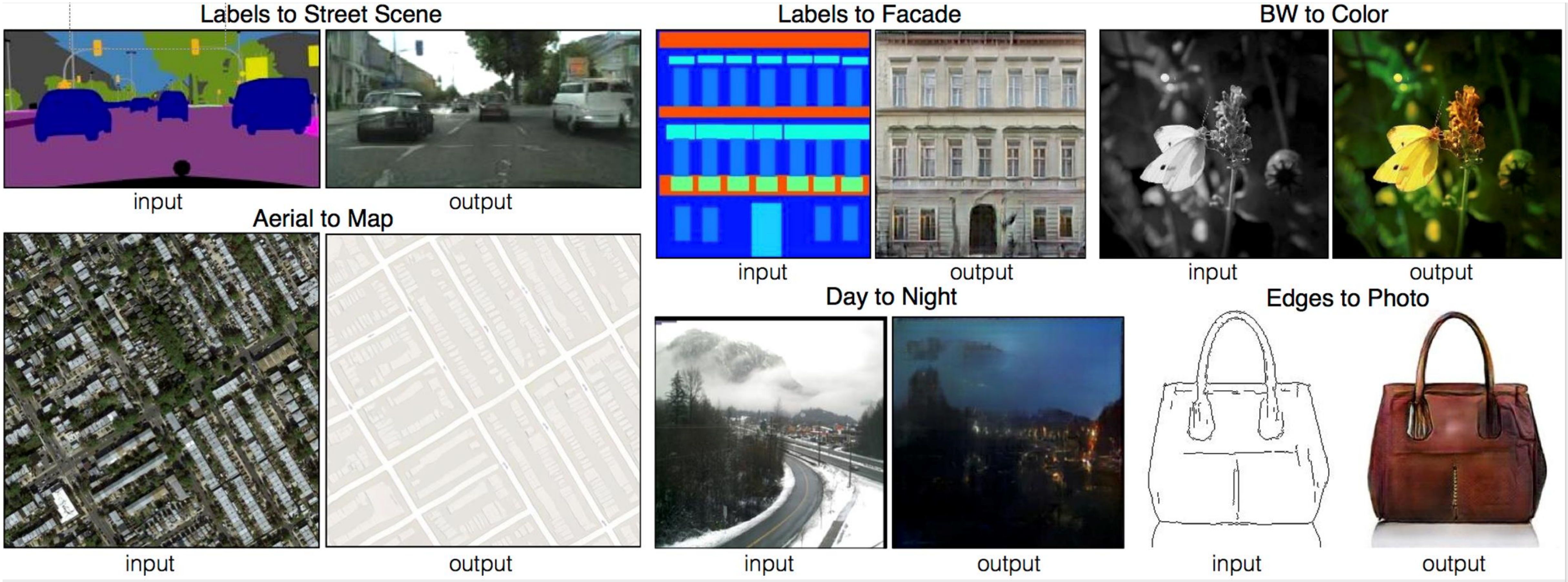

Labels to Street Scene
input
output
Aerial to Map
input
output
Labels to Facade
input
output
Day to Night
input
output
BW to Color
input
output
Edges to Photo
input
output

Image-to-Image Translation

- Architecture: *DCGAN*-based architecture
- Training is conditioned on the images from the source domain.
- Conditional GANs provide an effective way to handle many complex domains without worrying about designing *structured loss* functions explicitly.

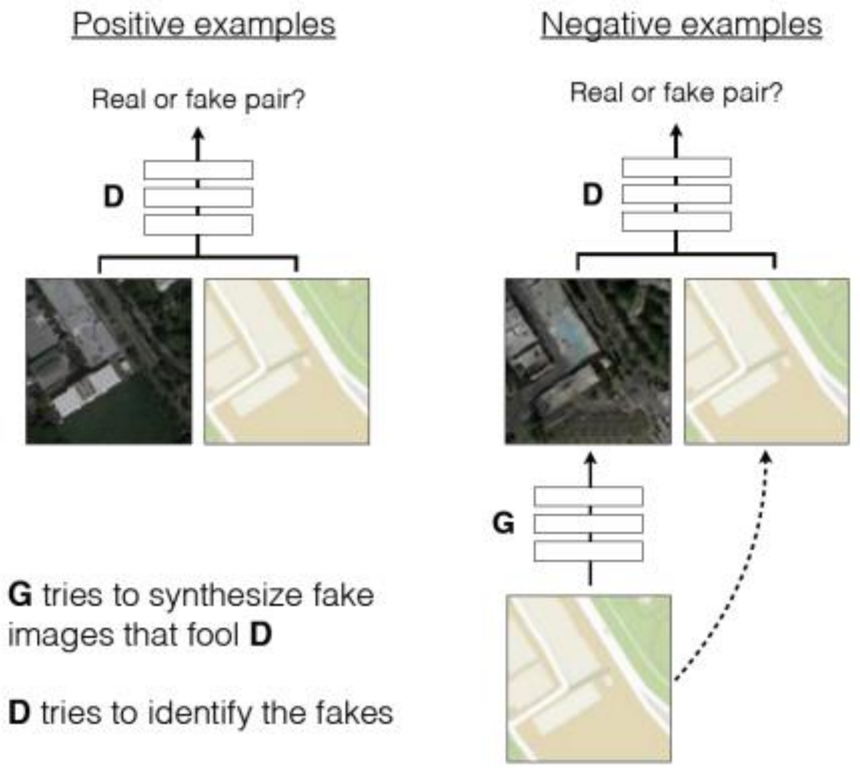
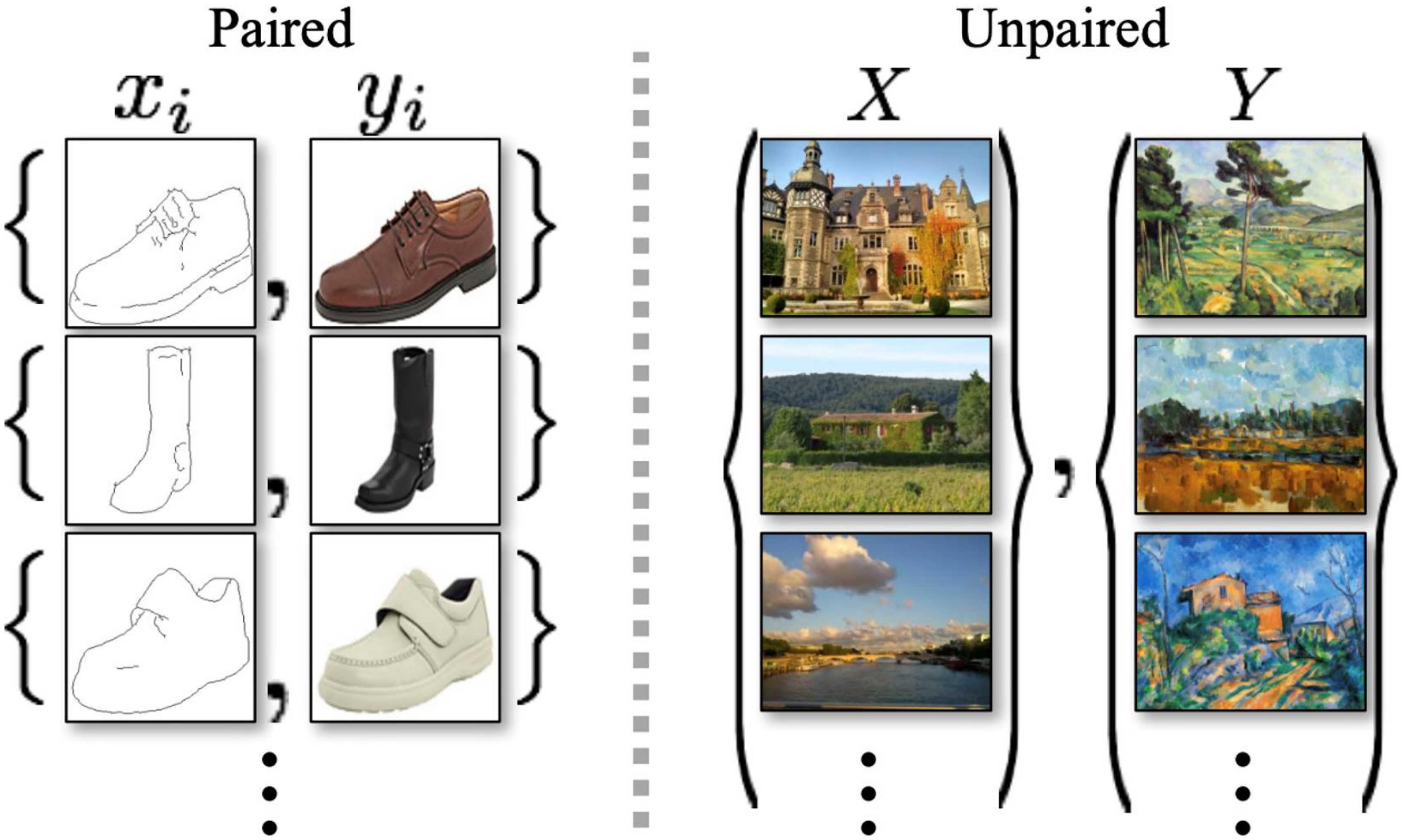


Figure 2 in the original paper.

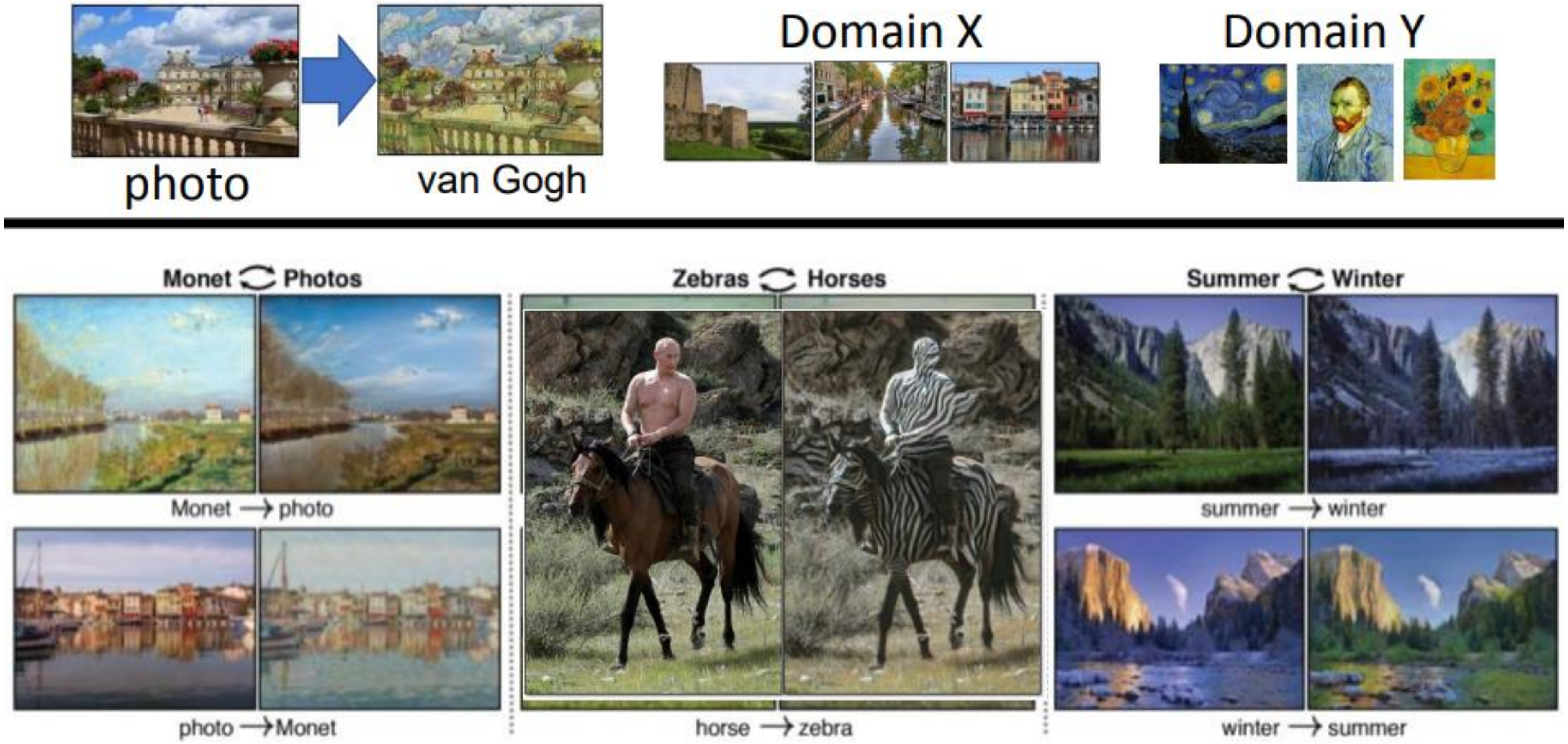
Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. "**Image-to-image translation with conditional adversarial networks**". arXiv preprint arXiv:1611.07004. (2016).



Paired
x_i
y_i
Unpaired
X
Y

CycleGAN: domain transformation

- Transform an object from one domain to another without paired data



- "Photo-realistic single image super-resolution using a generative adversarial network."
  - ✓ Ledig, Christian, et al.
  - ✓ arXiv preprint arXiv:1609.04802 (2016).

KISTI SINCE 1962

2022

TRUST KISTI

Korea Institute of Science
and Technology Information