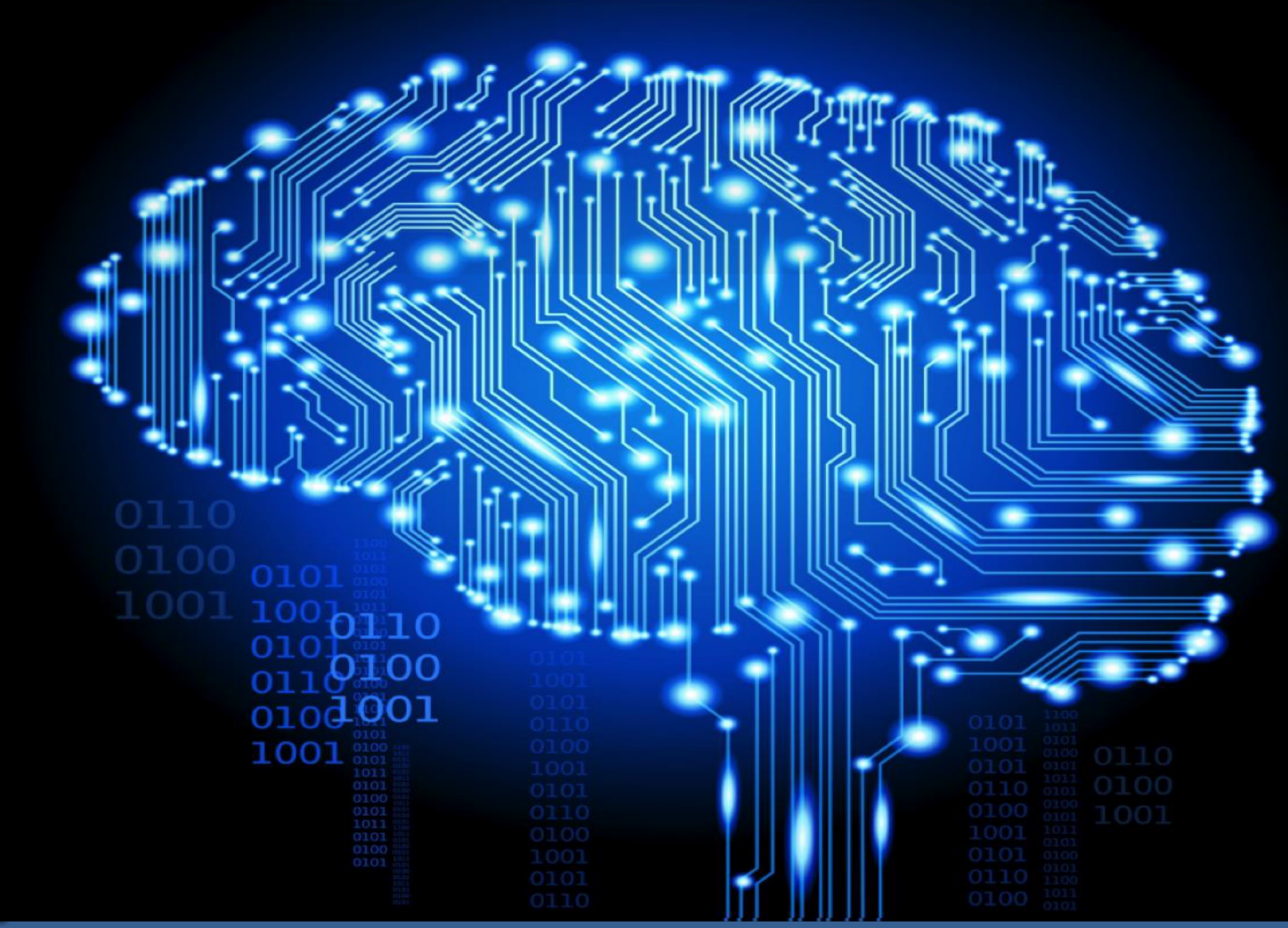


Prof. Saerom Park

Department of Convergence Security Engineering
psr6275@sungshin.ac.kr

Deep Generative Models

Introduction

Prof. Saerom Park

Reference

- [Goodfellow et al. (2016)] I. Goodfellow, Y. Bengio, and A. Courville, Deep learning, The MIT Press, 2016.
- [Gavves. (2017)] E. Gavves, UvA Deep Learning Course.

  http://uvadlc.github.io/
- [Grover. (2018)] A. Grover, S. Ermon, Tutorial on Deep Generative Models.

  https://ermongroup.github.io/generative-models/
- Borrows slides (some modified) from Grover & Gavves

Introduction

Richard Feynman: "*What I cannot create, I do not understand*“

Assumption for today: "*What I understand, I can create*“

Source: https://ermongroup.github.io/generative-models/

Statistical Generative Models

Data

+

Model family, loss function, optimization algorithm, etc.

Learning

Prior Knowledge

Image x

A probability distribution p(x)

New sample x’

Sampling from p(x) generates new images:

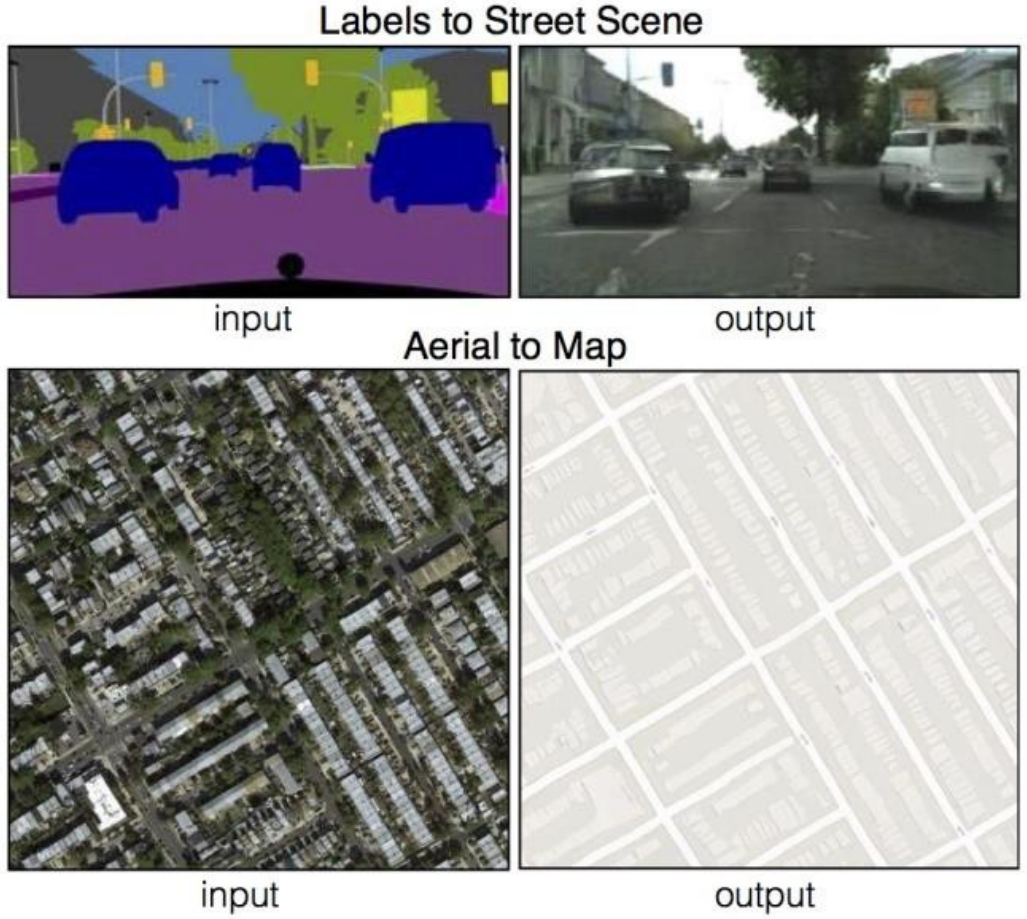
Source: https://ermongroup.github.io/generative-models/

Why generative modeling?

- Act as a regularizer in discriminative learning
  - Discriminative learning often too goal-oriented
  - Overfitting to the observations
- Semi-supervised learning
  - Missing data
- Simulating “possible futures” for Reinforcement Learning
- Data-driven generation/sampling/simulation

Source: http://uvadlc.github.io/

Applications: Cross-model translation



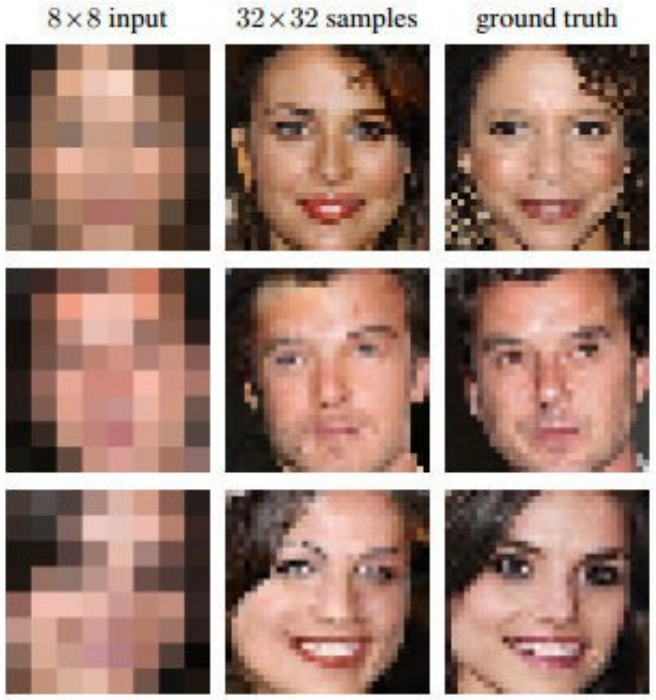
Source: http://uvadlc.github.io/

WaveNet

- Speech synthesis or text-to-speech (TTS)
  - Concatenative TTS: a very large database of short speech fragments are recorded from one speaker → recombined to from complete utterances
  - But, difficult to modify the voice (speaker, emotion, …)
  - Parametric TTS: all the information required to generate data is stored in the parameters of the model
- WaveNet improves the state of the art
  - Using Google's TTS datasets
  - Chinese, English, …
  - Music
  - https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

Image Super Resolution

Conditional generative model P(high res image | low res image)

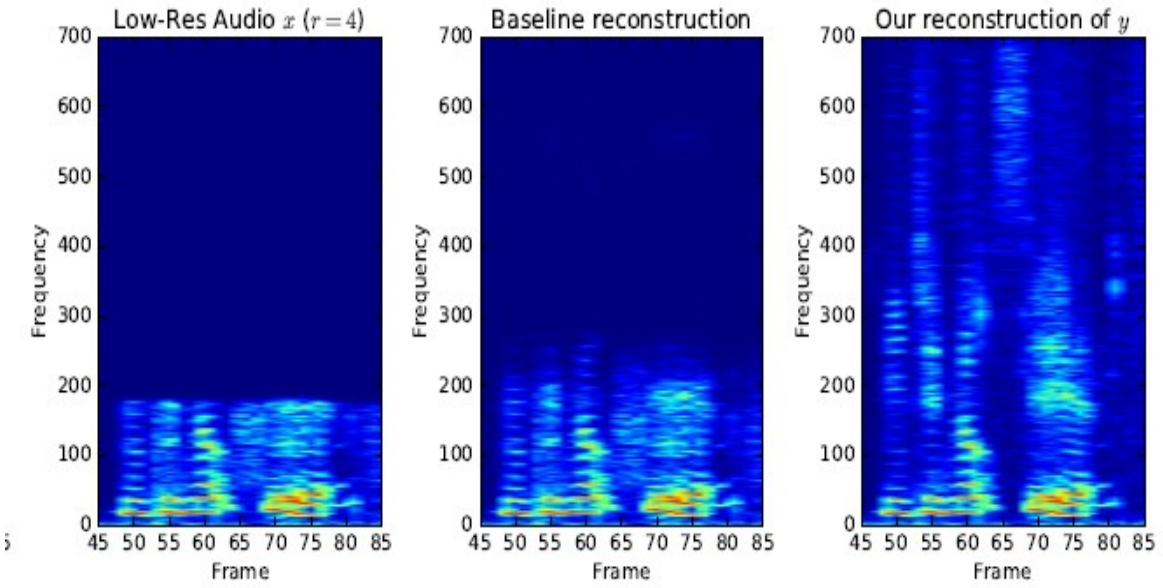


Ledig et al., 2017

Source: https://ermongroup.github.io/generative-models/

Audio Super Resolution

Conditional generative model P(high-res signal | low-res audio signal)



Low res signal

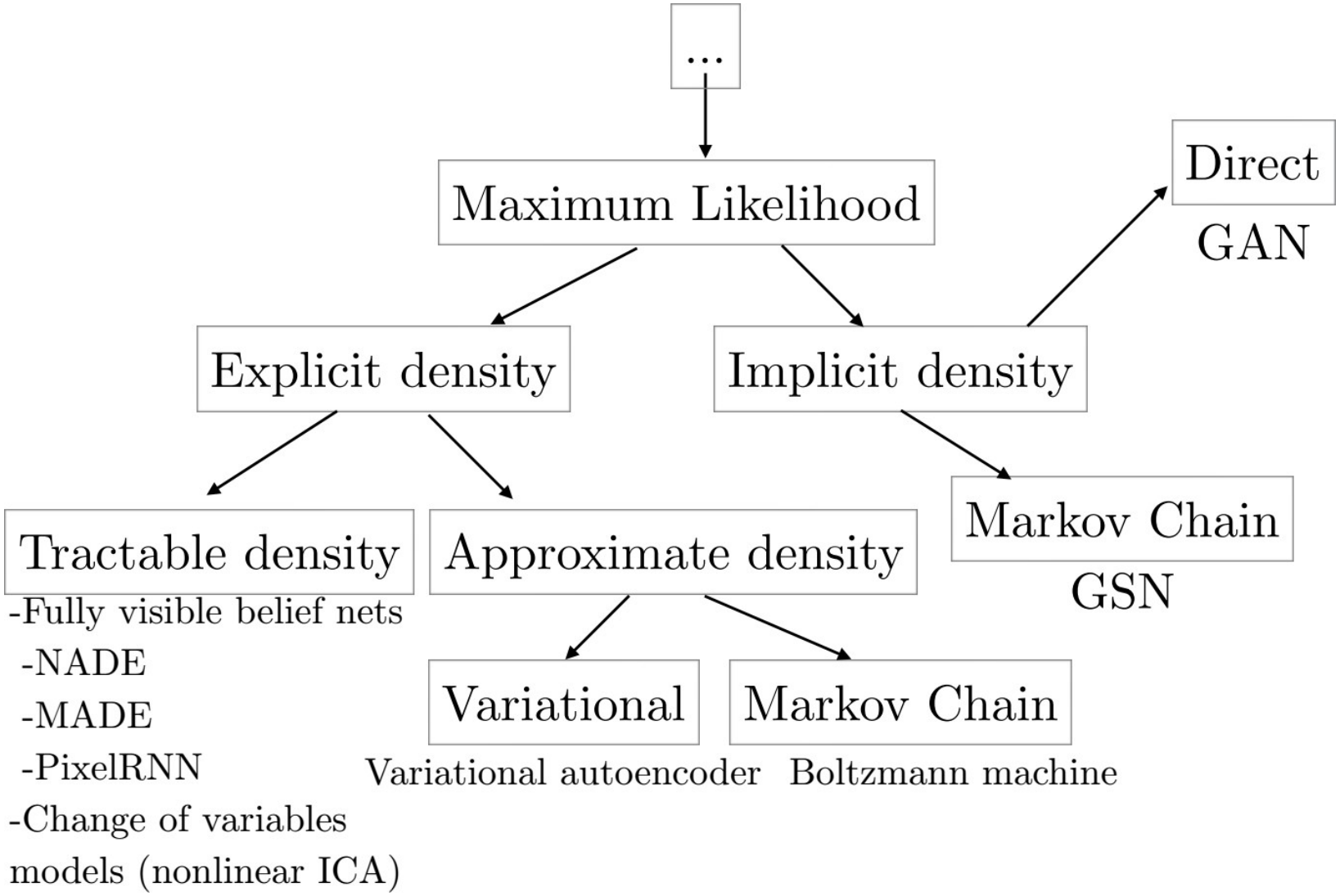
High res audio signal

Kuleshov et al., 2017

https://kuleshov.github.io/audio-super-res/

Source: https://ermongroup.github.io/generative-models/

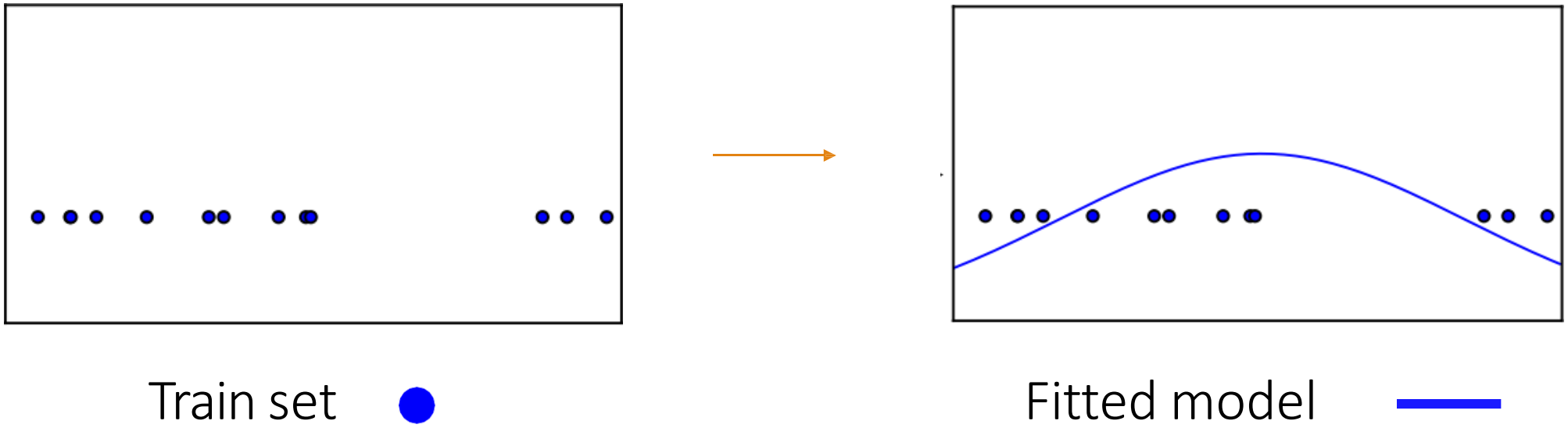
A map of generative models



Source: http://uvadlc.github.io/

Generative modeling: Case I

- Density estimation

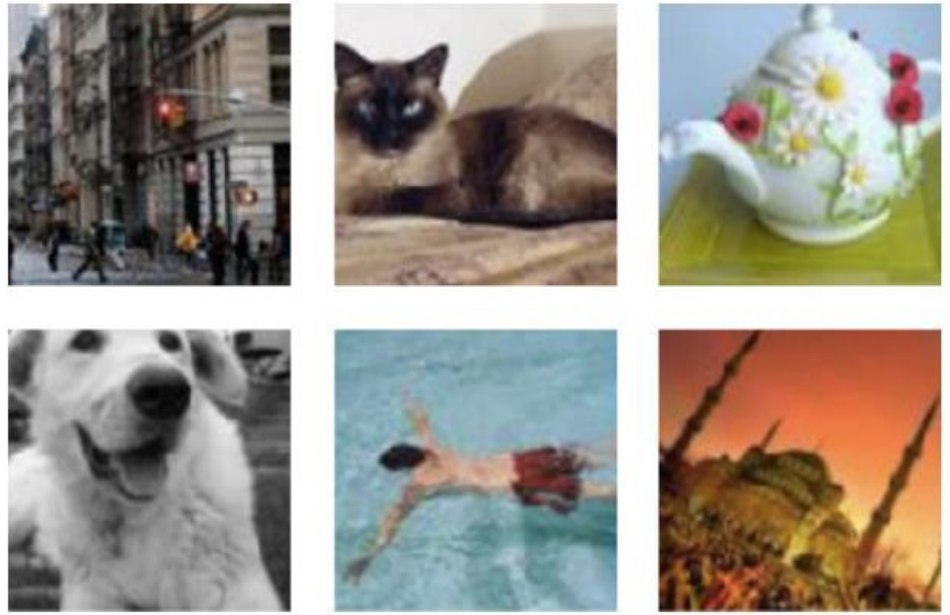


Source: http://uvadlc.github.io/

Generative modeling: Case II

- Sample Generation

Train examples



New samples (ideally)

Source: http://uvadlc.github.io/

Statistical Generative Models

Model family, loss function, optimization algorithm, etc.

Learn

Data

Generative model

$p(x)$

Desirable properties:

1. Sampling
2. Evaluating p(x)
3. Extracting features

as efficiently as possible

Source: https://ermongroup.github.io/generative-models/

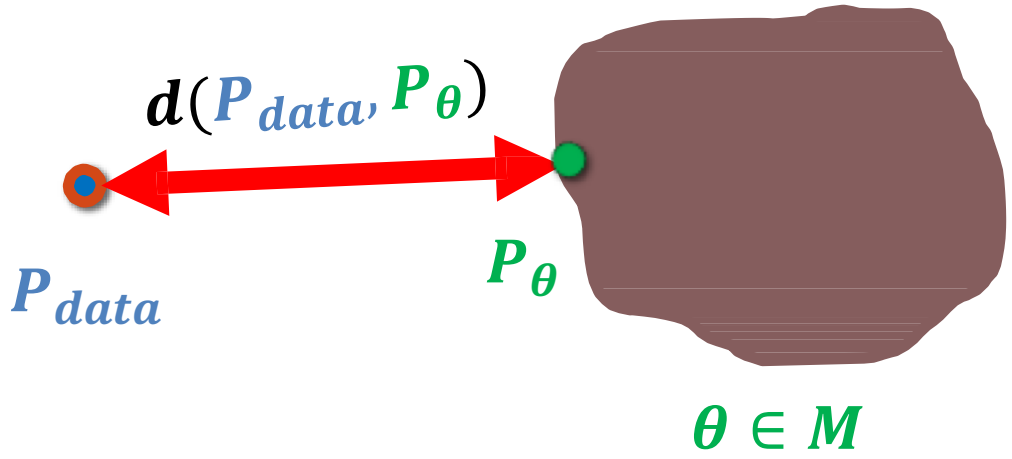
Learning in Generative Models

Given: Samples from a data distribution and **model family**

Goal: Approximate a data distribution as closely as possible

$$\min_{\theta \in M} d(P_{data}, P_\theta)$$

$x_i \sim P_{data}$
$i = 1, \ldots, n$

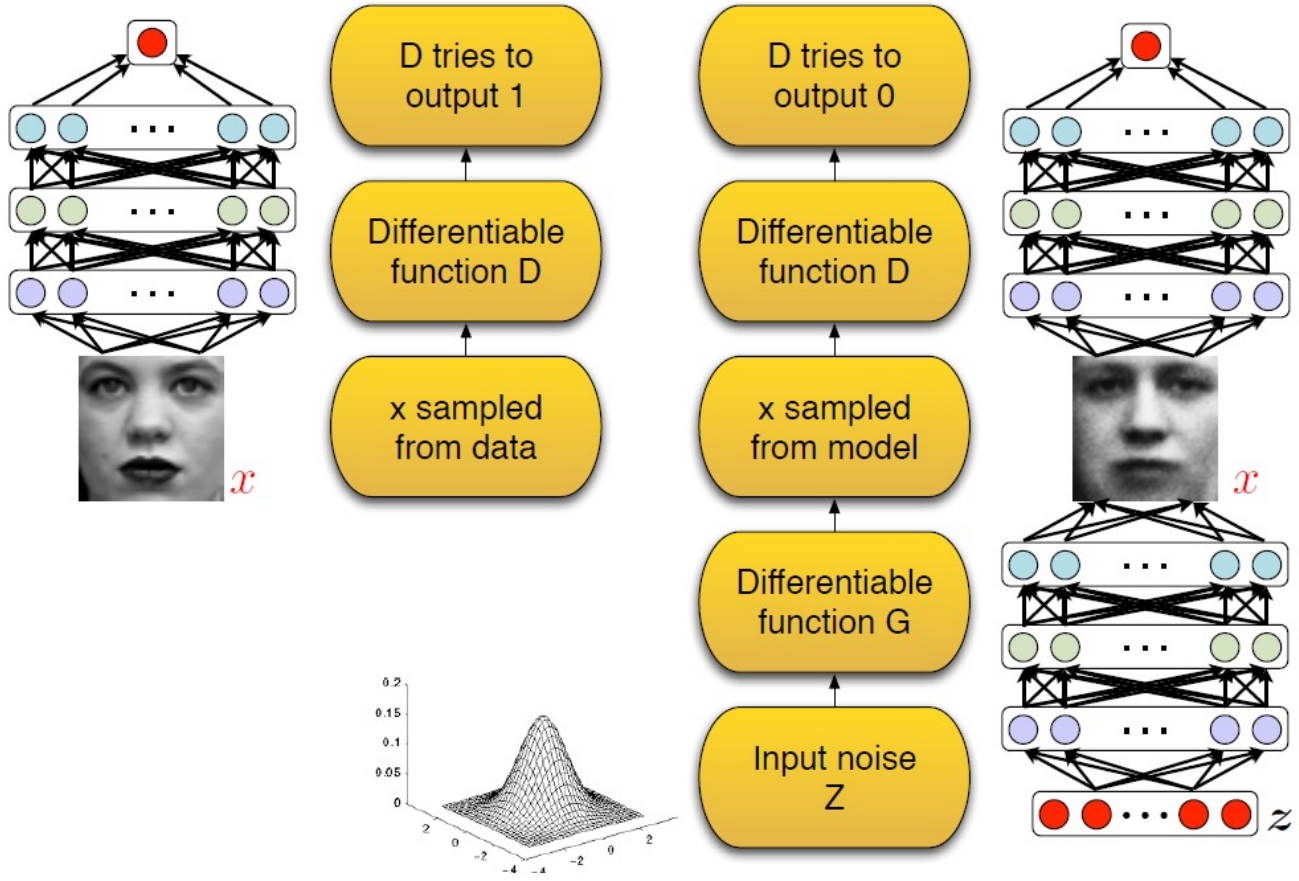


Challenge: How should we evaluate and optimize closeness between the data distribution and the model distribution?

Source: https://ermongroup.github.io/generative-models/

Generative Adversarial Networks

Prof. Saerom Park



What is a GAN?

- Generative
  - You can sample novel input samples
  - E.g., you can literally "create" images that never existed
- Adversarial
  - Our generative model G learns adversarially, by fooling an discriminative oracle model D
- Network
  - Implemented typically as a (deep) neural network
  - Easy to incorporate new modules
  - Easy to learn via backpropagation

Source: http://uvadlc.github.io/

Generative Adversarial Networks

- So far we were trying to express the pdf of the data $p(x)$
- Why bother? Is that really necessary? Is that limiting?
- Can we model directly the sampling of data?
- Yes, by modelling sampling as a “Thief vs. Police” game

GAN: Intuition

- Assume you have two parties
  - Police: wants to recognize fake money as reliably as possible
  - Counterfeiter: wants to make as realistic fake money as possible
- The police forces the counterfeiter to get better (and vice versa)
- Solution relates to Nash equilibrium

Source: http://uvadlc.github.io/

“Police vs Thief”

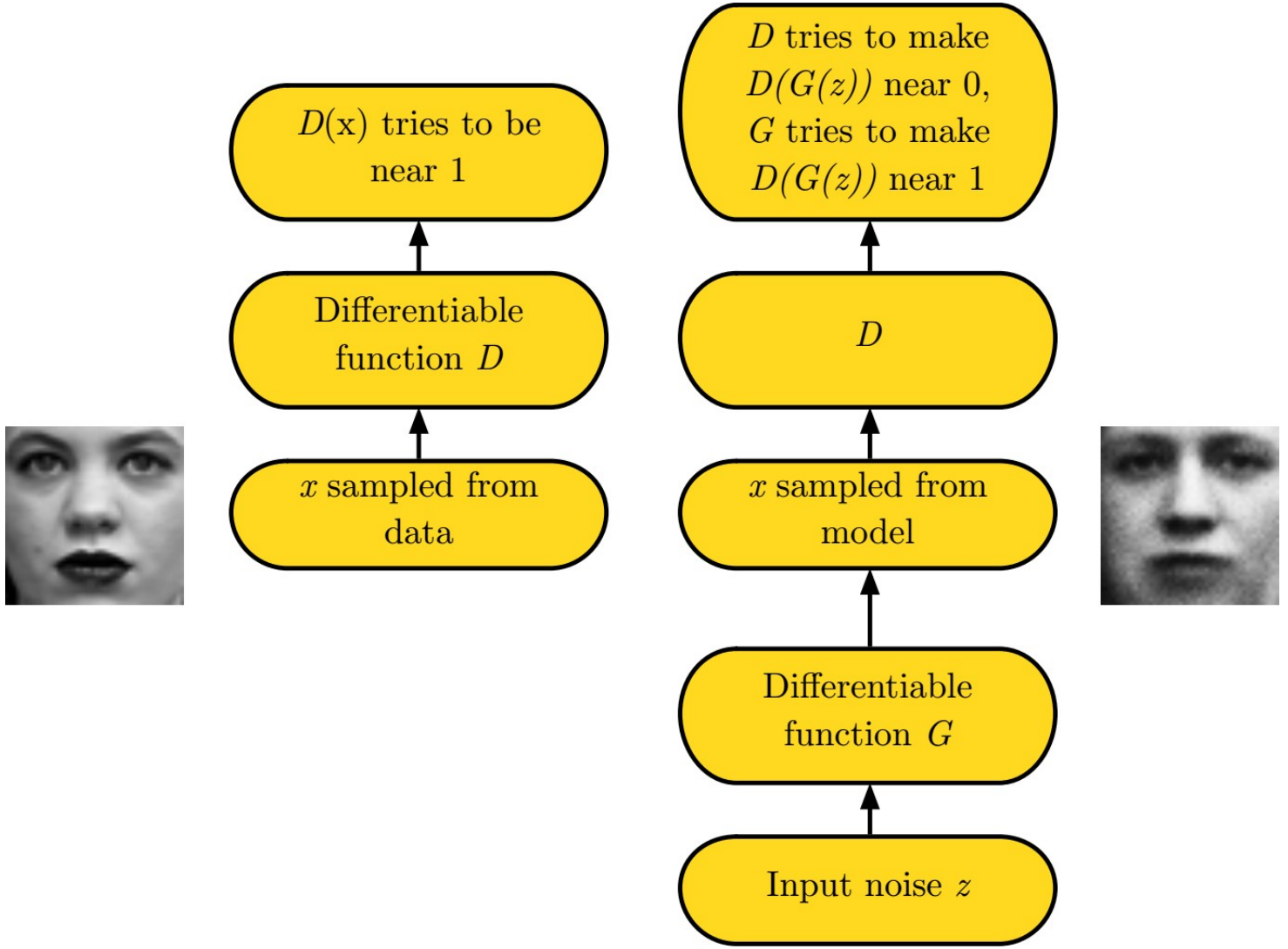
- Generator and discriminator networks optimized together
  - The generator (thief) tries to fool the discriminator
  - The discriminator (police) tries to not get fooled by the generator
- Key idea: A two player generator-discriminator game
  - **Discriminator** distinguishes between real dataset samples and fake samples from the generator.
  - **Generator** generates samples that can fool the discriminator
- Mathematically

$$\min_G \max_D V(G,D) = \mathbf{E}_{x\sim p_{data}(x)} \log D(x) + \mathbf{E}_{z\sim p_Z(z)} \log(1 - D(G(z)))$$

  - Discriminator tries to maximize its reward $V(G,D)$
  - Generator is tries to minimize Discriminator’s reward

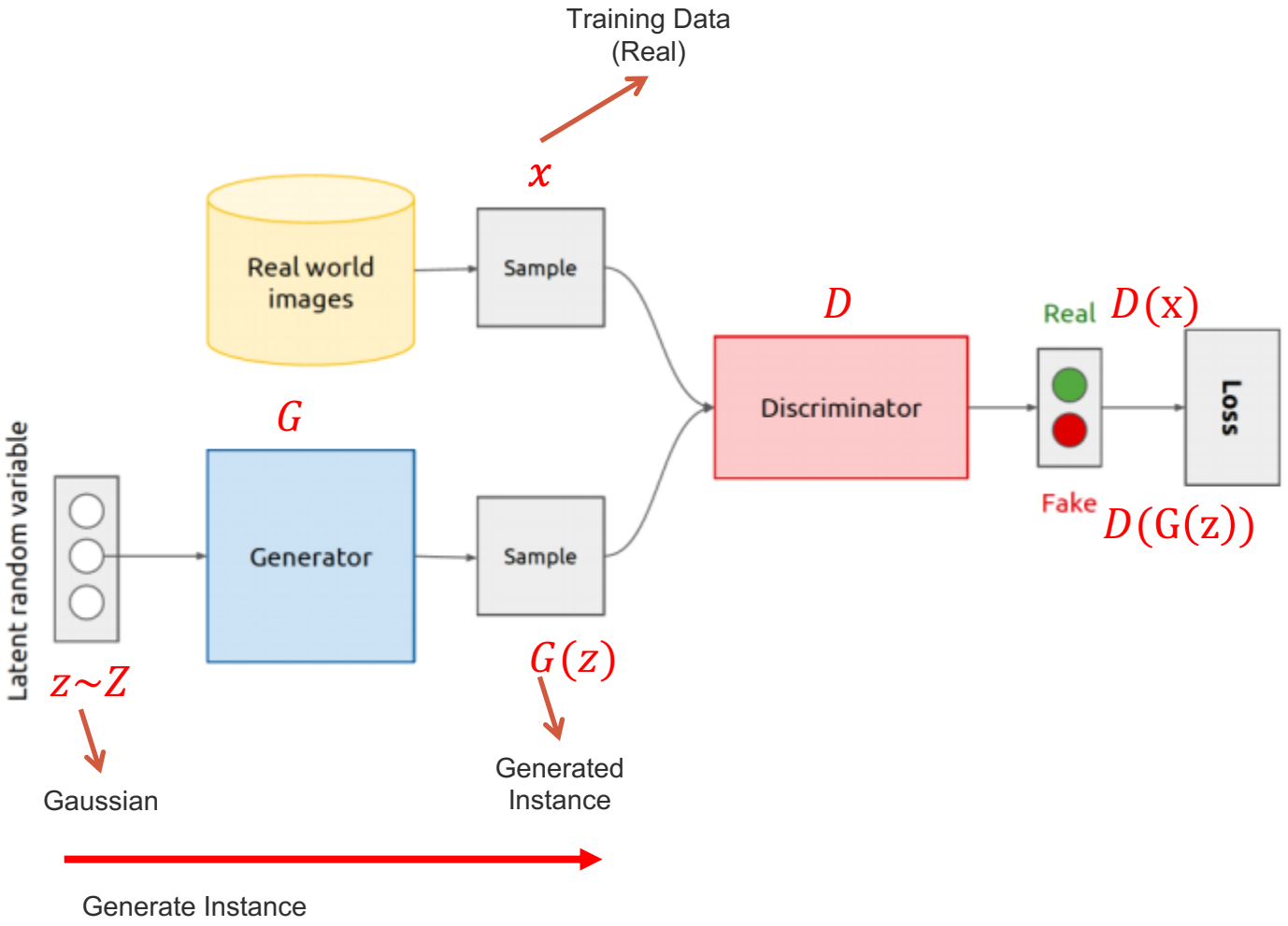
Source: http://uvadlc.github.io/

GAN: Pipeline



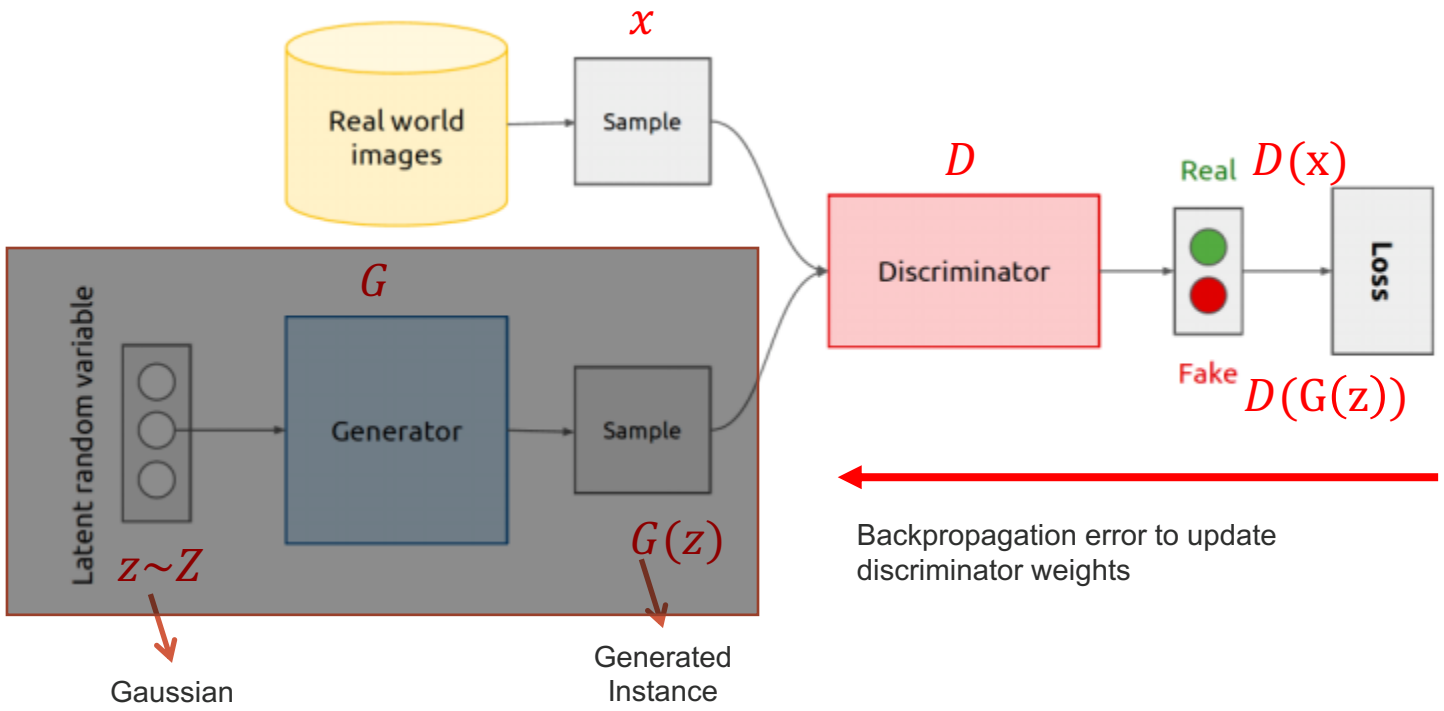
Source: http://uvadlc.github.io/

GAN's architecture



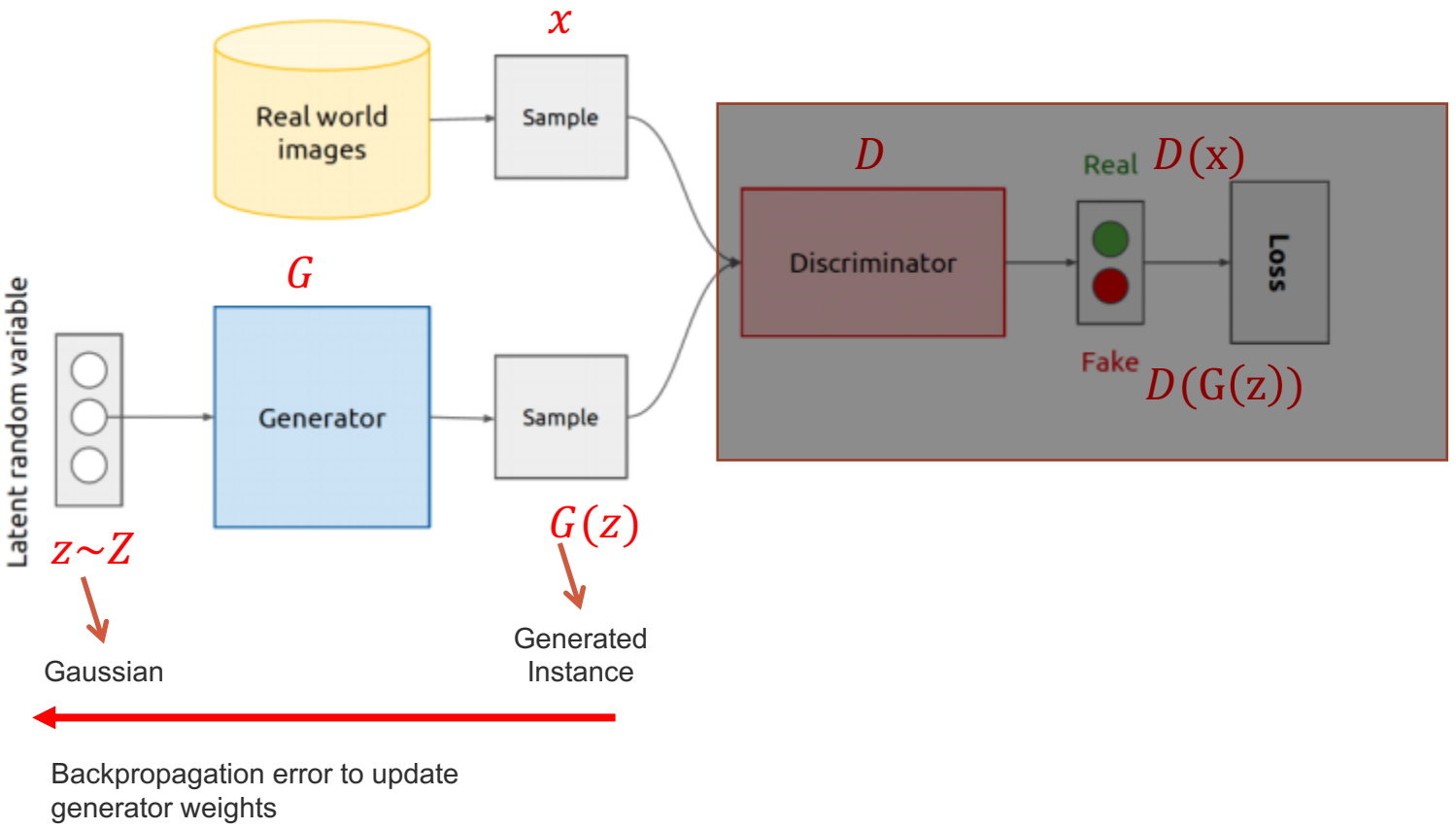
Source: http://slazebni.cs.illinois.edu

GAN's architecture



Source: http://slazebni.cs.illinois.edu

GAN's architecture



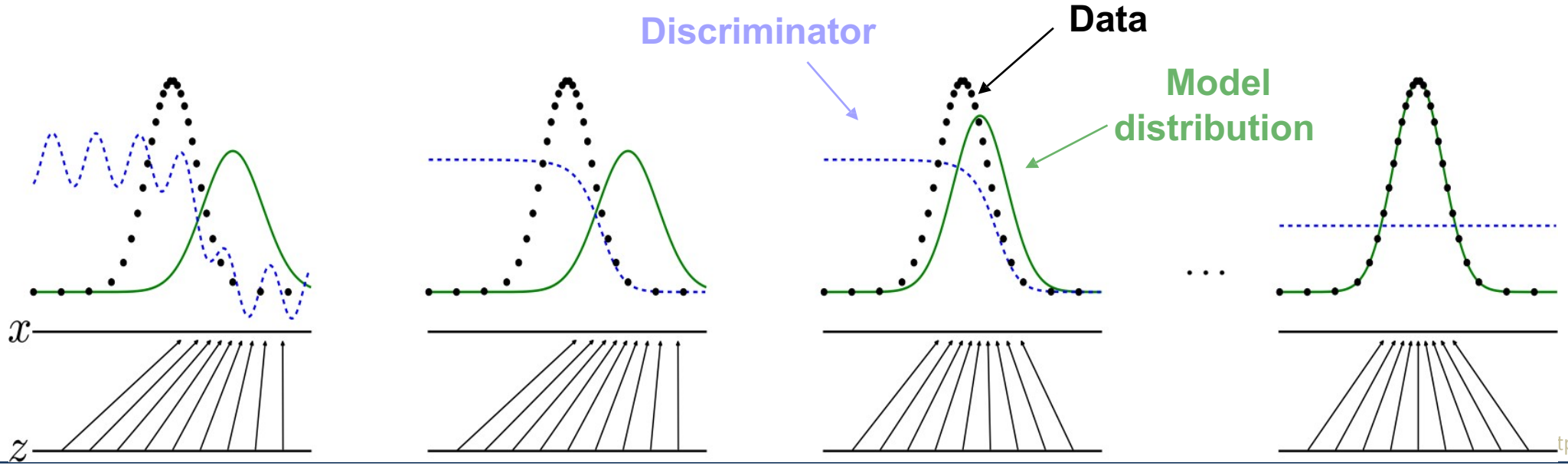
Source: http://slazebni.cs.illinois.edu

Optimal discriminator

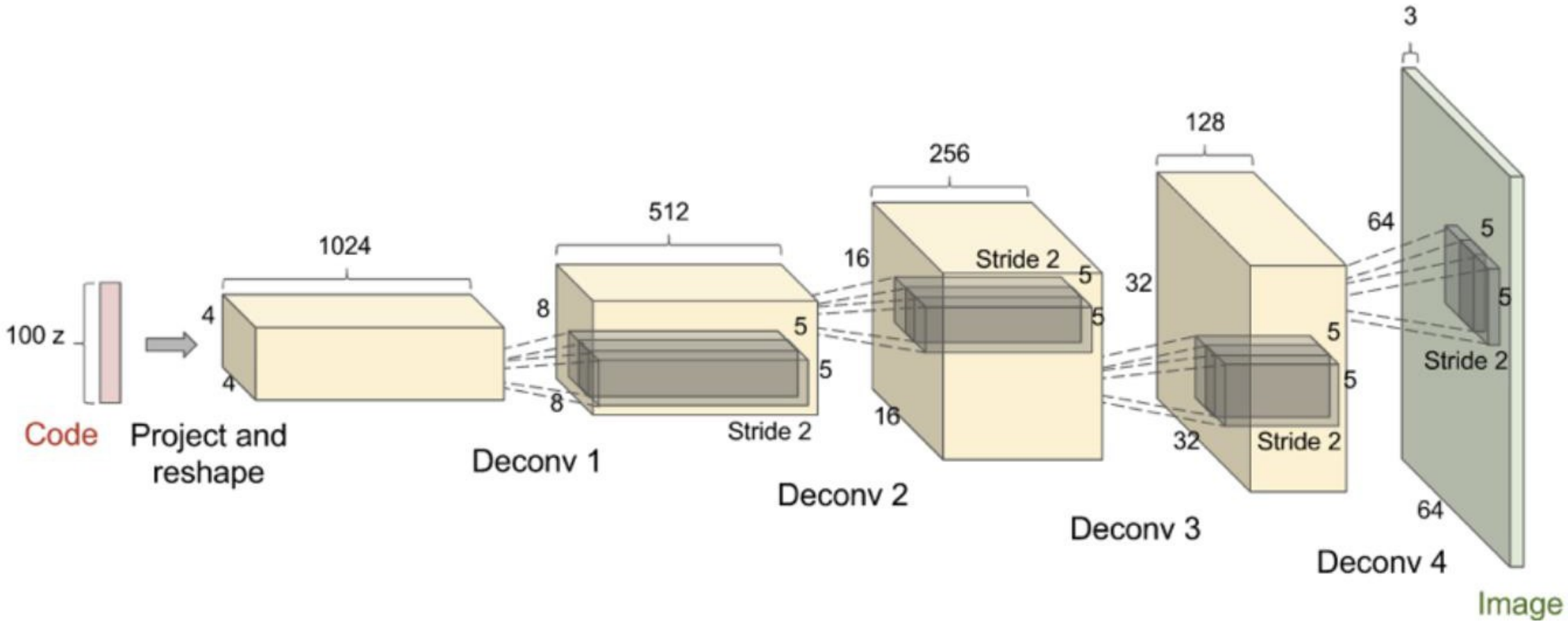
- $\min_G \max_D V(G,D) = E_{x\sim p_{data}(x)} \log D(x) + E_{z\sim p_Z(z)} \log(1 - D(G(z)))$
- Optimal $D(x)$ for any $p_{data}(x)$ and $p_{model}(x)$ is always

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

- Estimating this ratio with supervised learning (discriminator) is the key



DCGAN Architecture

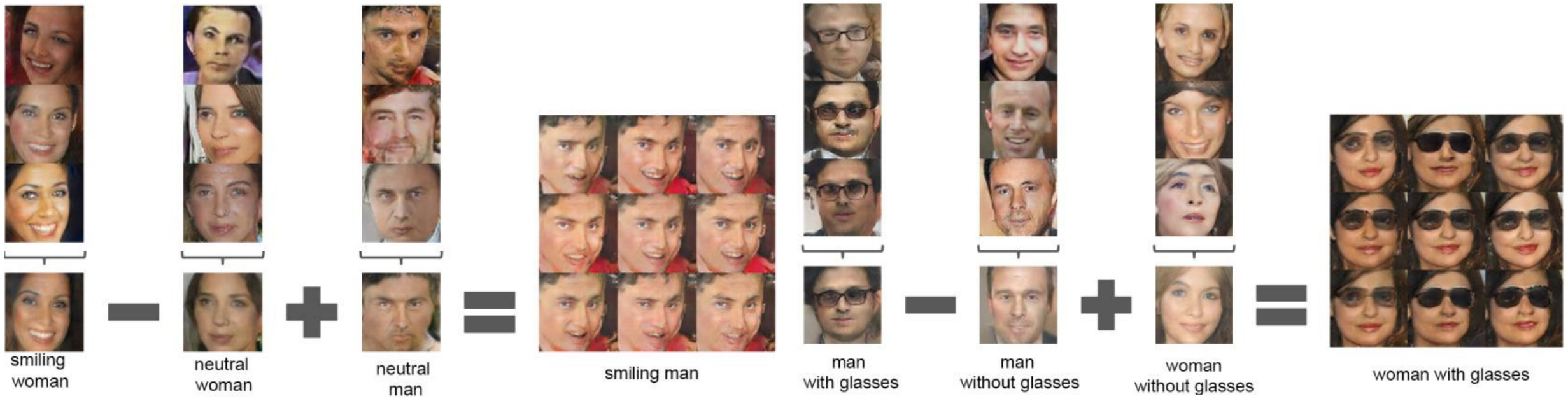


Source: http://uvadlc.github.io/

Examples

Source: http://uvadlc.github.io/

Image “arithmetics”



Source: http://uvadlc.github.io/

Inferring latent representations

- Prefinal layer of discriminator (Salimans et al., 2016)
- **Controlled generation:** InfoGAN (Chen et al., 2016)

(a) Rotation (b) Width

Source: https://ermongroup.github.io/generative-models/

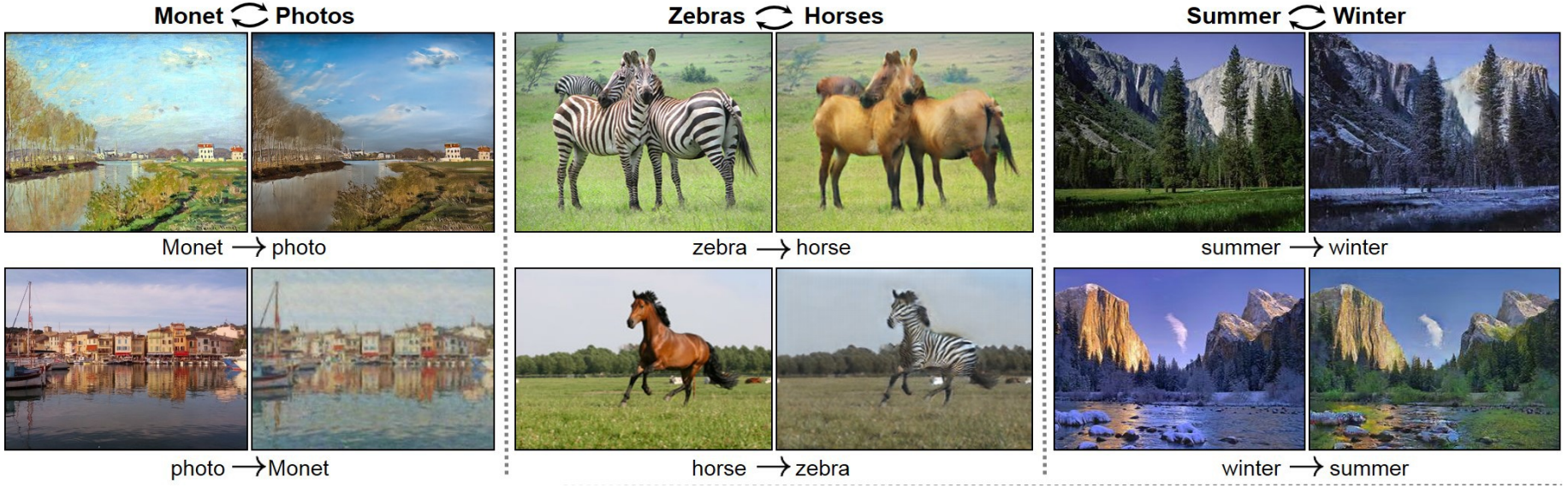
Progressive Growing of GANs

Karras et al., 2018

Source: https://ermongroup.github.io/generative-models/

Inferring latent representations

- Learn **explicit encoders** along with generators: Adversarially Learned Inference(Dumoulin et al., 2017), BiGAN (Donahue et al., 2017)
- **Conditional GANs** for image translation: Pix2Pix (Isola et al., 2017), CycleGAN (Zhu et al., 2017)



Source: https://ermongroup.github.io/generative-models/

References of GAN

- Evaluation
  - **Likelihoods** may not be defined or tractable (Wu et al., 2017, Grover et al., 2018, Salimans et al., 2016, Heusel et al., 2018)
  - Directed model permits **ancestral sampling**
    - For labelled datasets, metrics such as inception scores and Frechet inception distance quantify sample diversity and quality using pretrained classifiers
- Stabilizing training
  - **Theory of GAN convergence:** Arjovsky & Bottou, 2017, Arora et al., 2017, Mescheder et al., 2017, Roth et al., 2017, Nagarajan & Kolter, 2018, Li et al., 2018
  - JSD can be replaced **f-divergences and integral probability metrics** (Nowozoin et al., 2016, Li et al., 2017, Arjovsky et al., 2017)
  - **Practical architectures, regularizers, optimization algorithms:** Radford et al., 2015, Poole et al., 2016, Salimans et al., 2016, Gulrajani et al., 2017, Metz et al., 2017, Miyato et al., 2018, Karras et al., 2018 and many more! https://github.com/soumith/ganhacks
    - *How to Train a GAN? Tips and tricks to make GANs work* by Soumith Chintala

Source: https://ermongroup.github.io/generative-models/