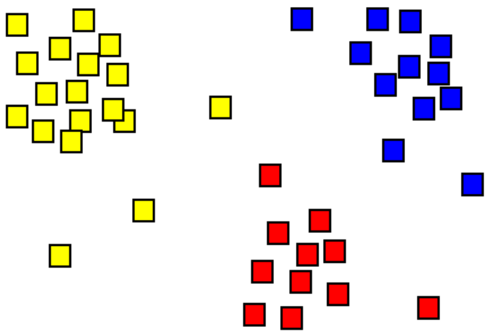
Unsupervised Learning
VAE & GAN

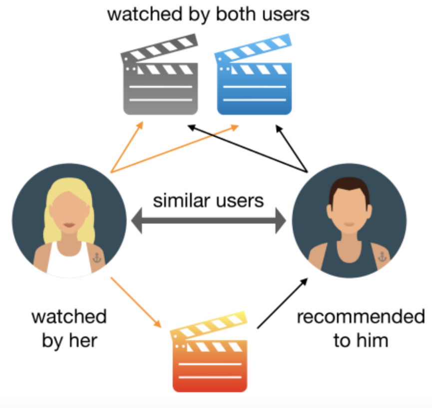
LEADINGINDIA.AI

NATIONWIDE AI SKILLING & RESEARCH INITIATIVE

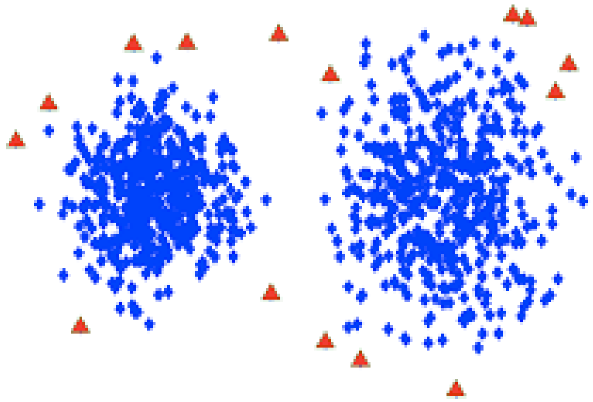
Applications of Unsupervised Learning



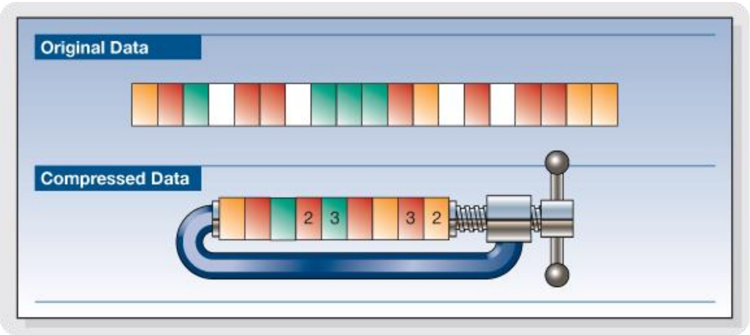
Clustering



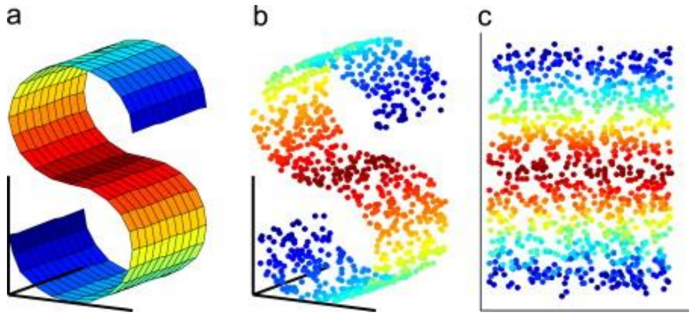
Recommender System



Anomaly Detection



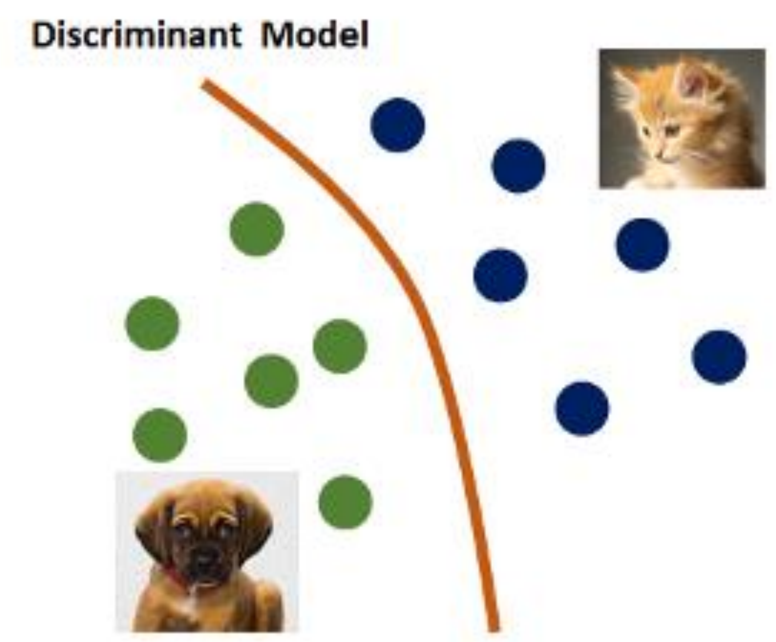
Data Compression

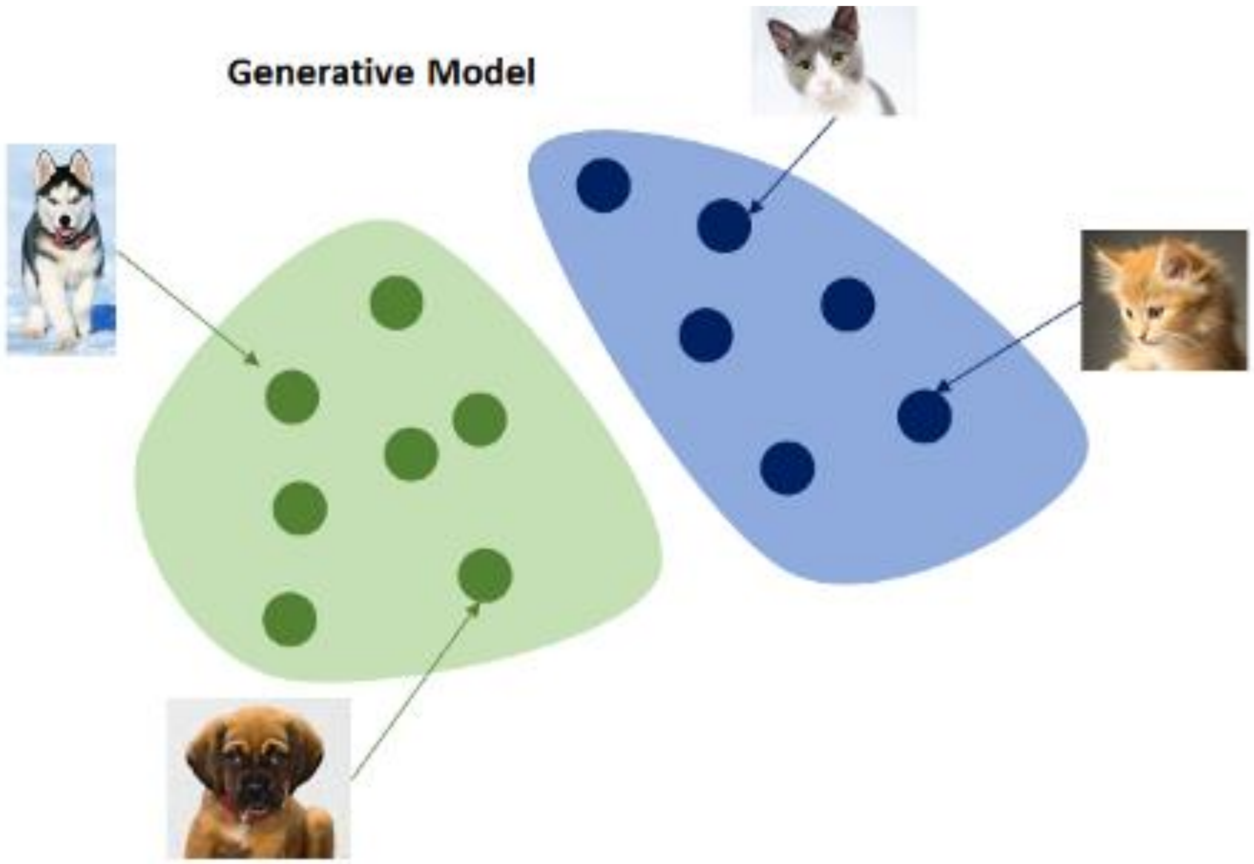


Dimensionality Reduction

Data Generation

Discriminative Vs Generative Models



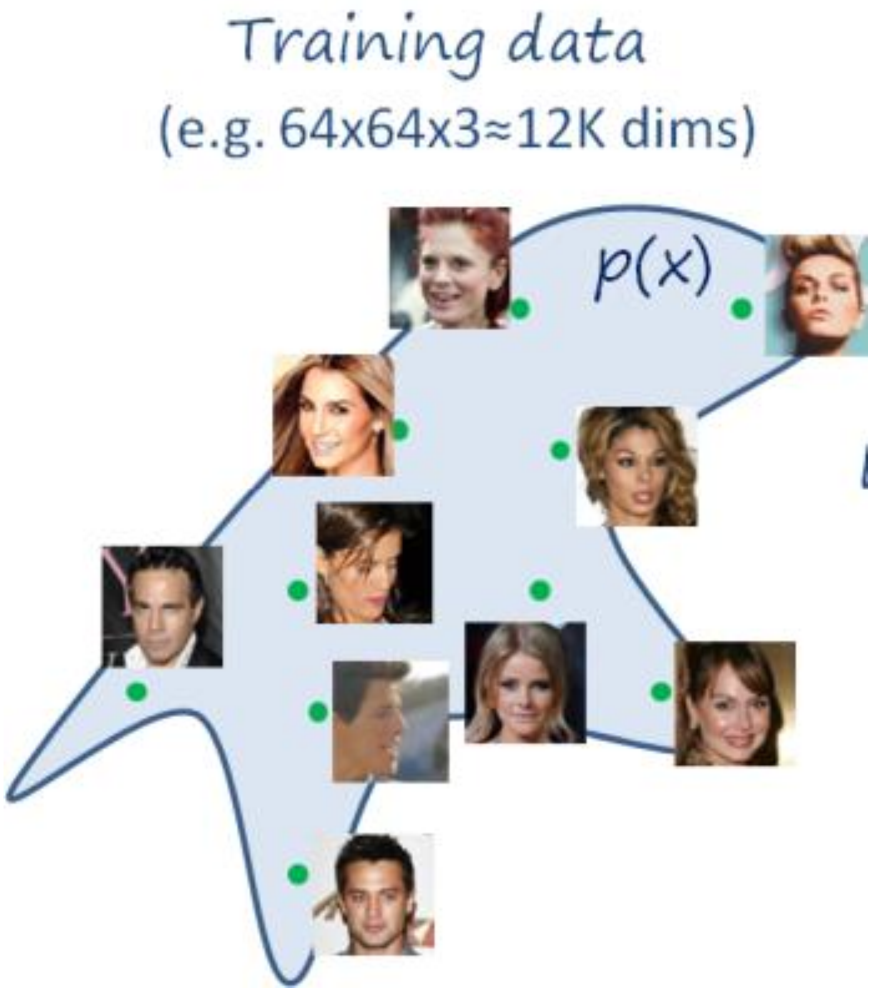


Discriminative Vs Generative Learning

- **Discriminative models**
- A discriminative model **does not** care how the data is generated. Here we just care about $P(y|x)$.
- **Example**: Classifying whether image contains cat or not?

- **Generative models**
- Generative models describe **how data is generated** using probabilistic models. They predict $P(y|x)$, the probability of y given x, calculating the $P(x,y)$, the probability of x and y.
- **Example:** Generating a new cat image

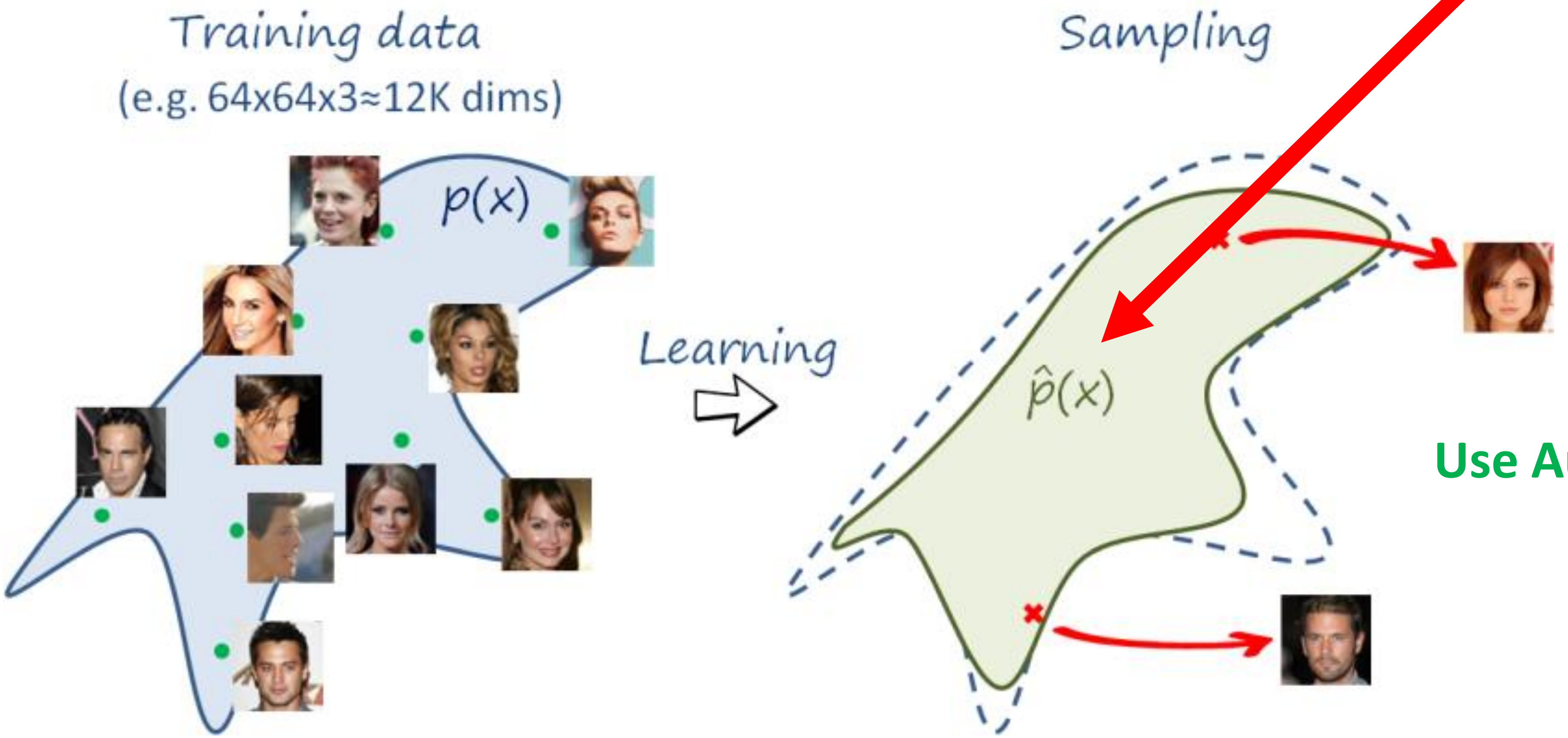
Sampling New Data



Dimensions are large...
then how?

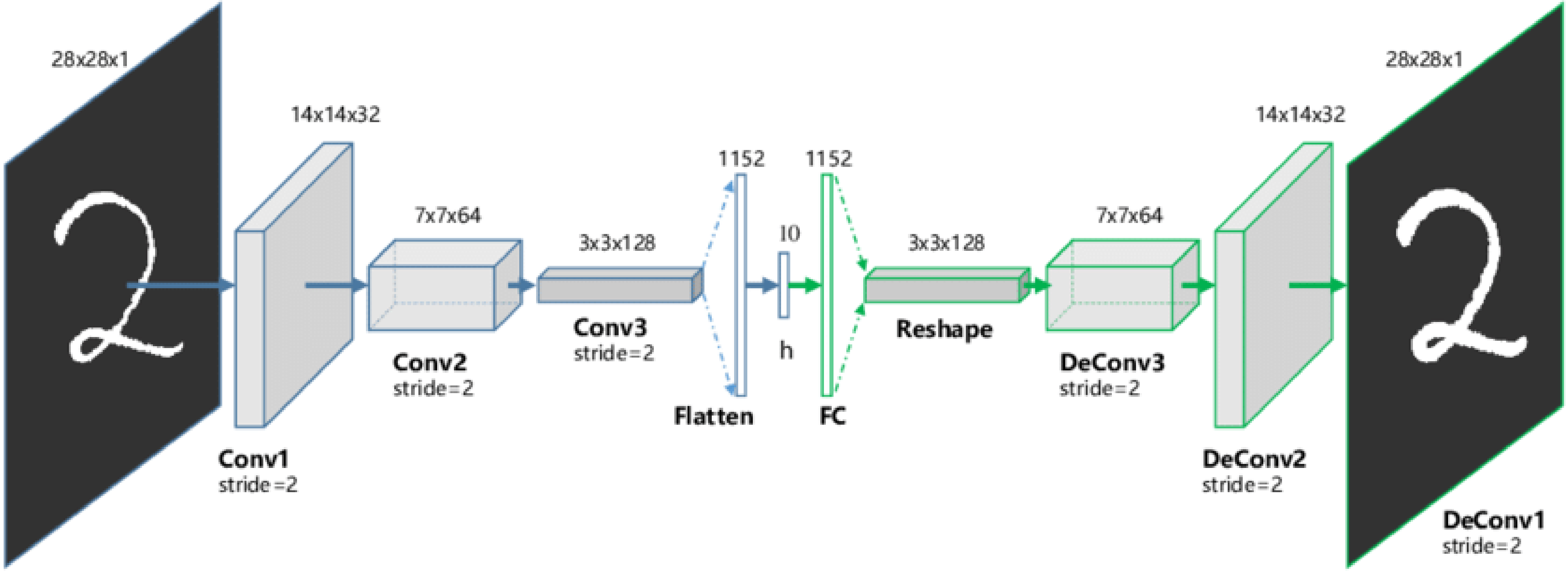
Sampling New Data

How to learn this p(x)?



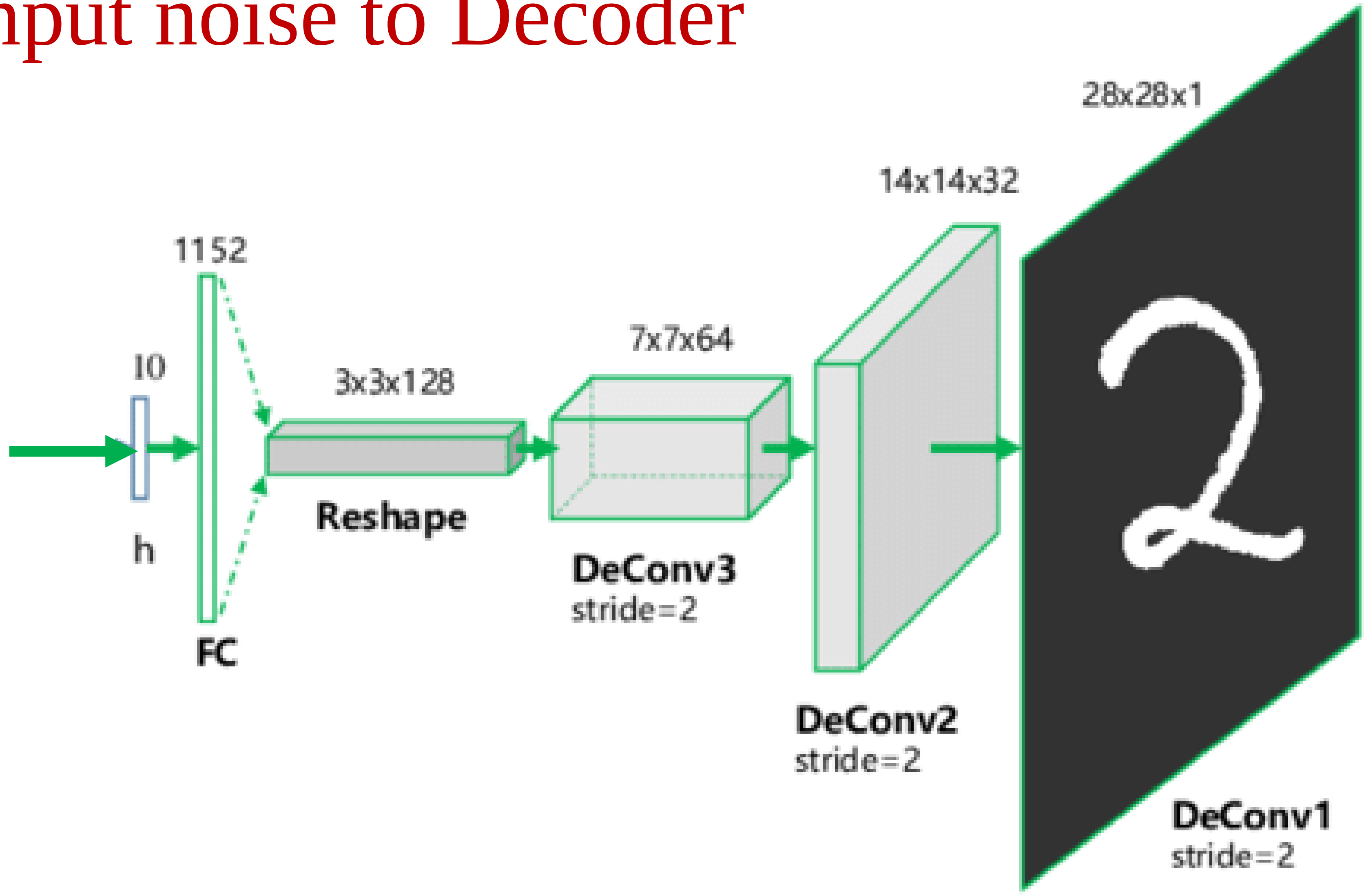
Use Autoencoder

Step 1: Train an autoencoder



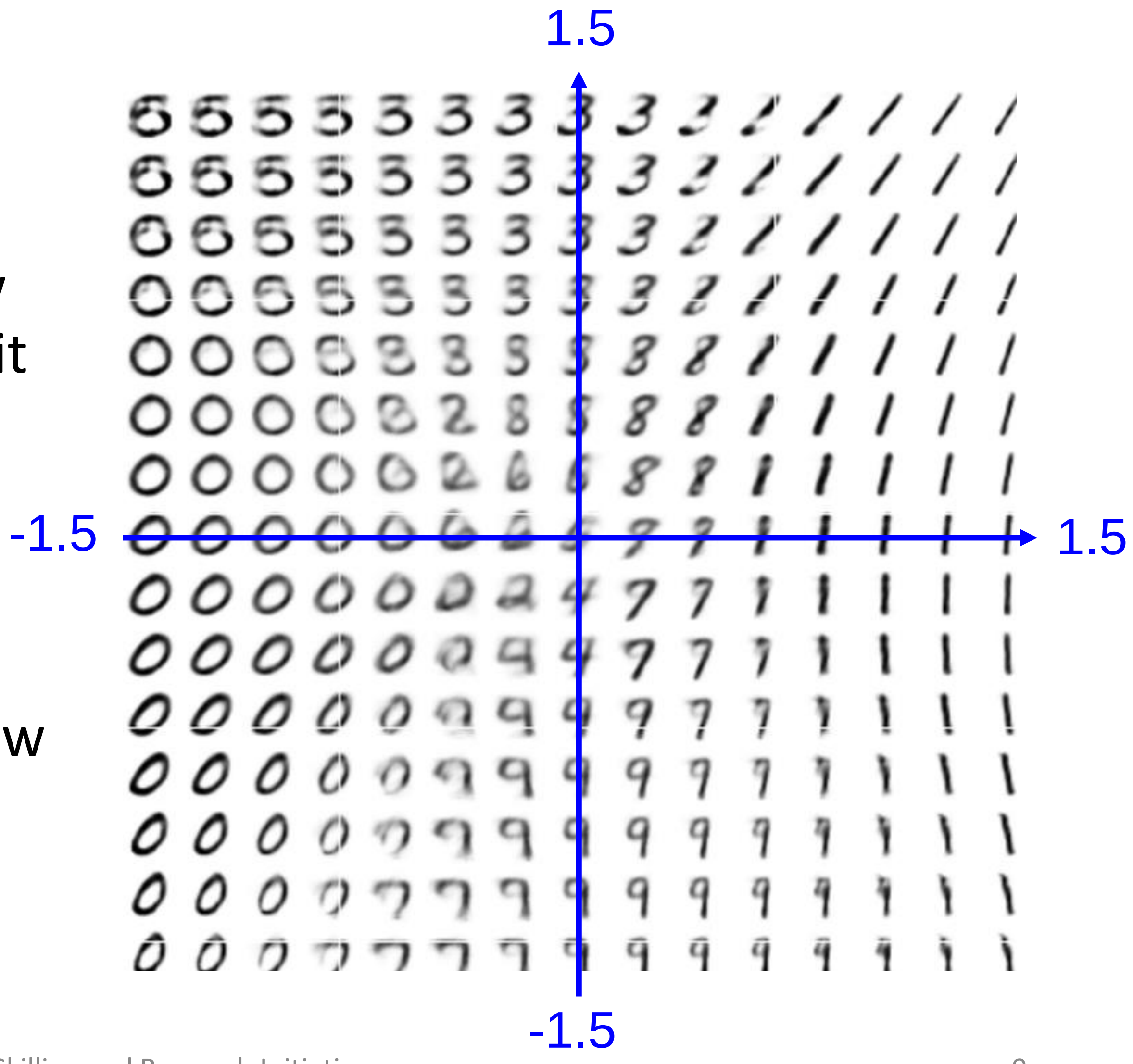
Step 2: Input noise to Decoder

Random Noise
10 float numbers

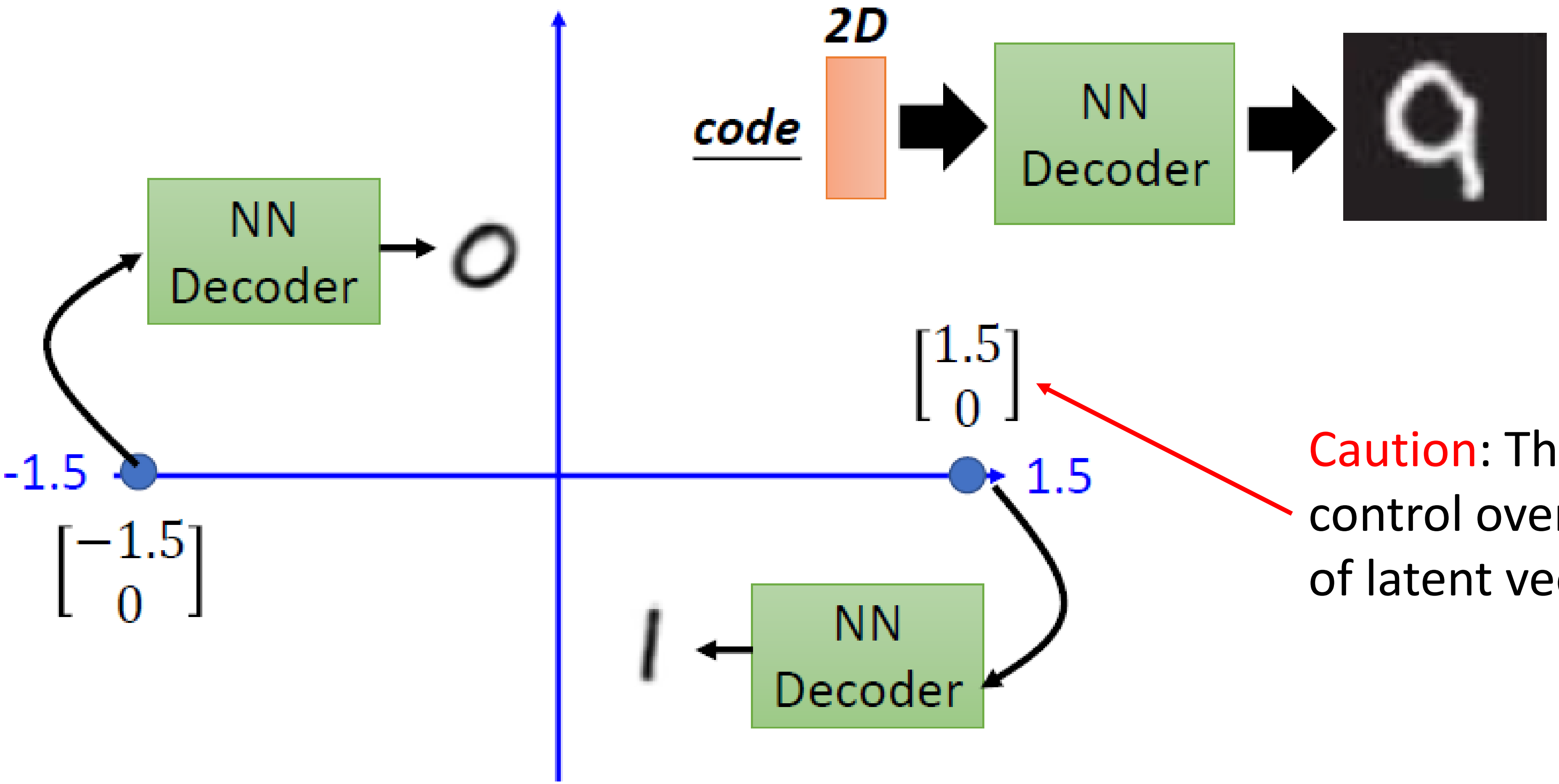


Autoencoder for Image Generation

- Lets imagine an autoencoder with **two** hidden nodes for MNIST
- Decoder can be used for generating new images if we give 2 numbers of input to it
- Each image in the dataset can be considered as a point in the 2-dimensional latent space
- New data can be sampled by picking a new point from then 2D space where new digit will look as interpolation between the neighbourhood points



Autoencoder for Image Generation



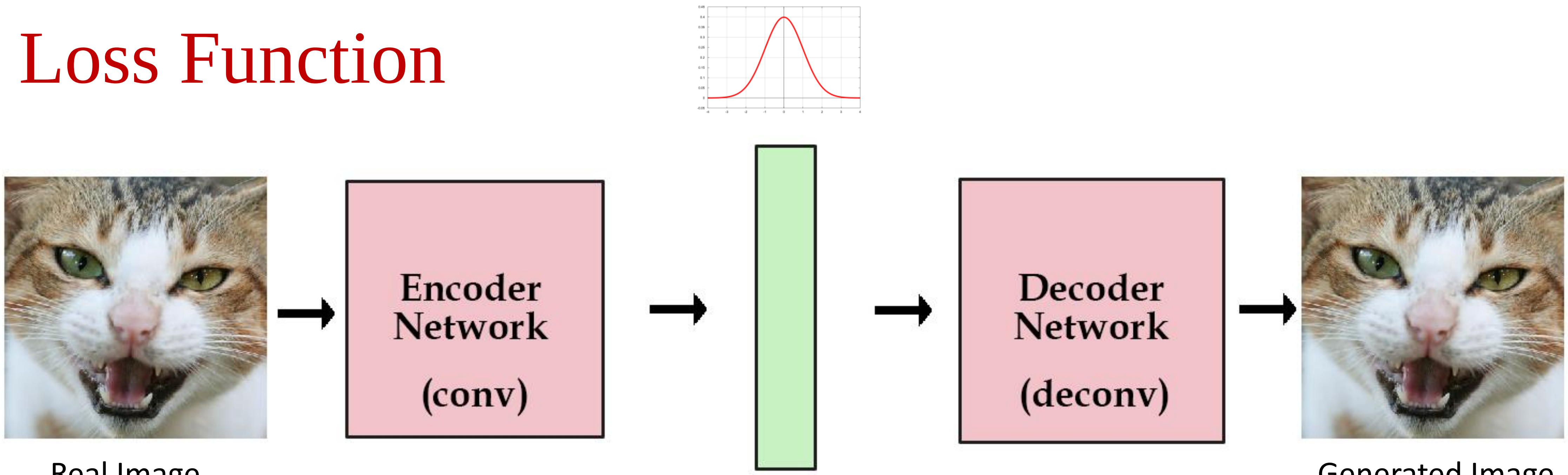
Caution: There is no control over the range of latent vectors!

From Autoencoder to Variational Autoencoder

- Autoencoders learn a “compressed representation” of input automatically by first compressing the input and decompressing it back
- AE has no control over the Latent activations of the network hence data generation becomes difficult
- Instead of learning a mapping function, Variational Auto Encoder learns the parameters of the probability distribution representing the data
- In VAE, we have the control over hidden activations because we keep in near the Unit Gaussian
- Add a Constraint in encoding network, that forces it to generate latent vector, that roughly follow the unit Gaussian distribution.
- This is the constraint that separates it with the standard auto encoder.
- For generation , it is easy now, sample a latent vector from unit Gaussian distribution, and pass it to the decoder. Decoder will generate the image for that latent vector.

Loss Function

Unit Gaussian Distribution



Real Image

Generated Image

latent vector / variables

Generation Loss = Mean(Square(Generated Image – Real Image))

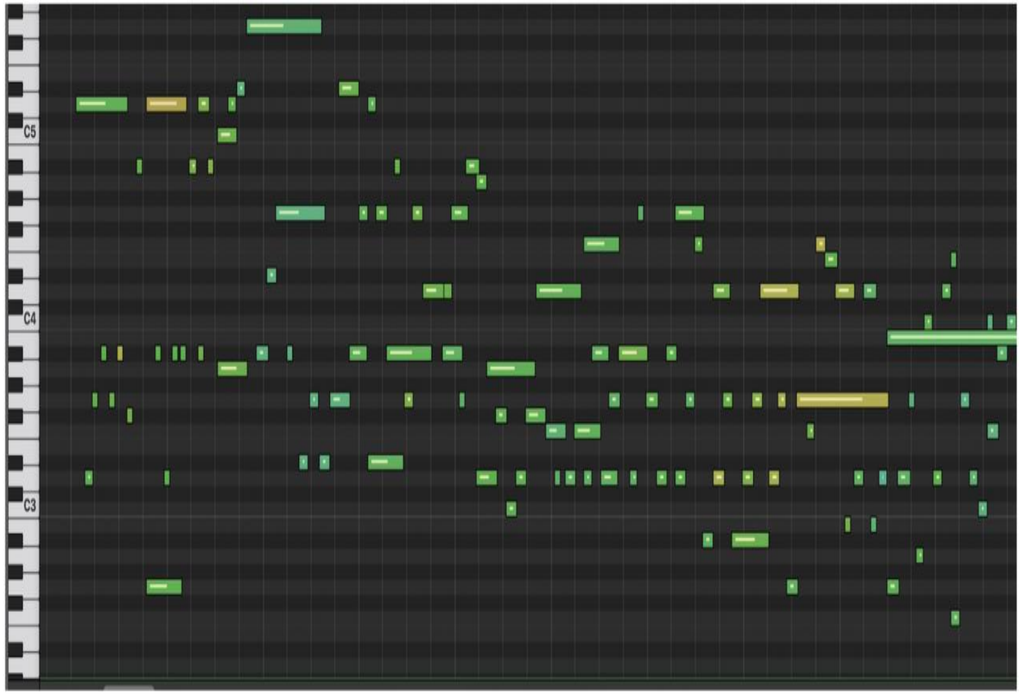
Latent Loss = KL-Divergence(Latent Variable, Unit Gaussian)

Loss = Generation Loss + Latent Loss

VAE Images

VAE for Music

- Google use's Autoencoders for creating new music and sounds



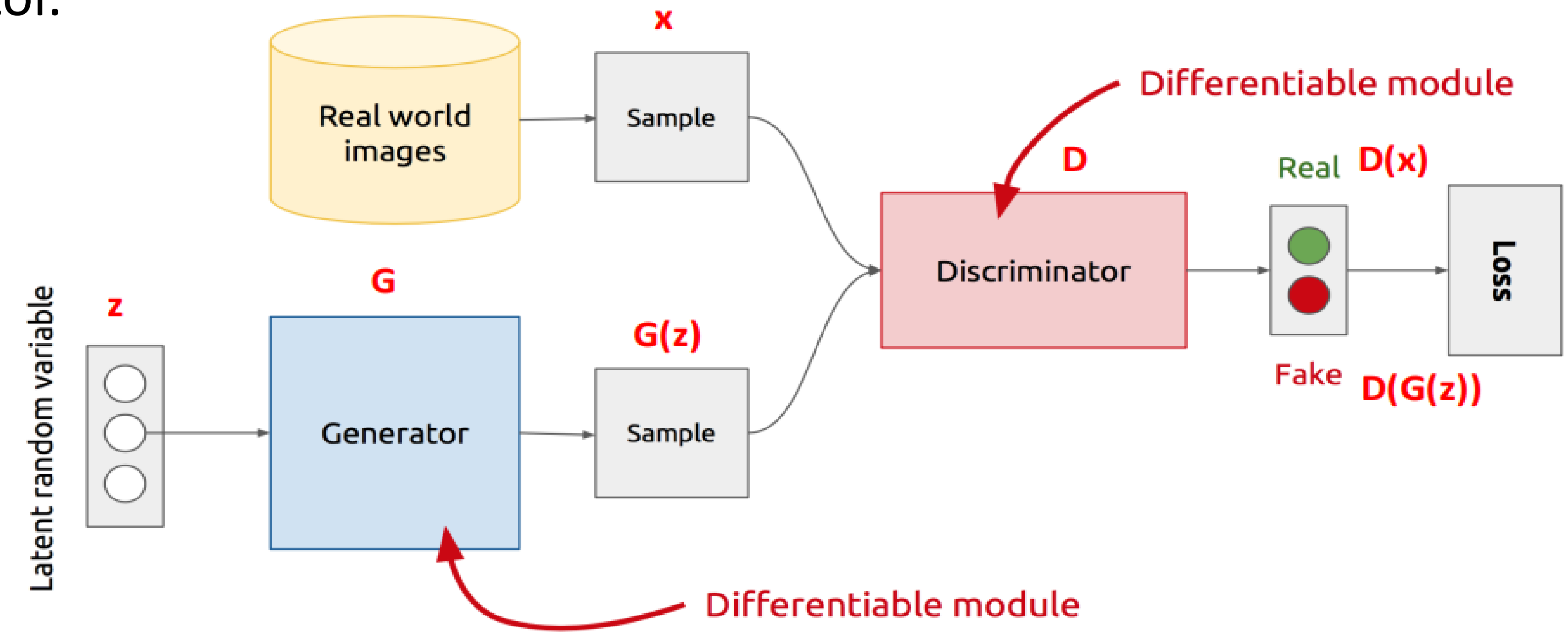
MusicVAE

NSynth Super

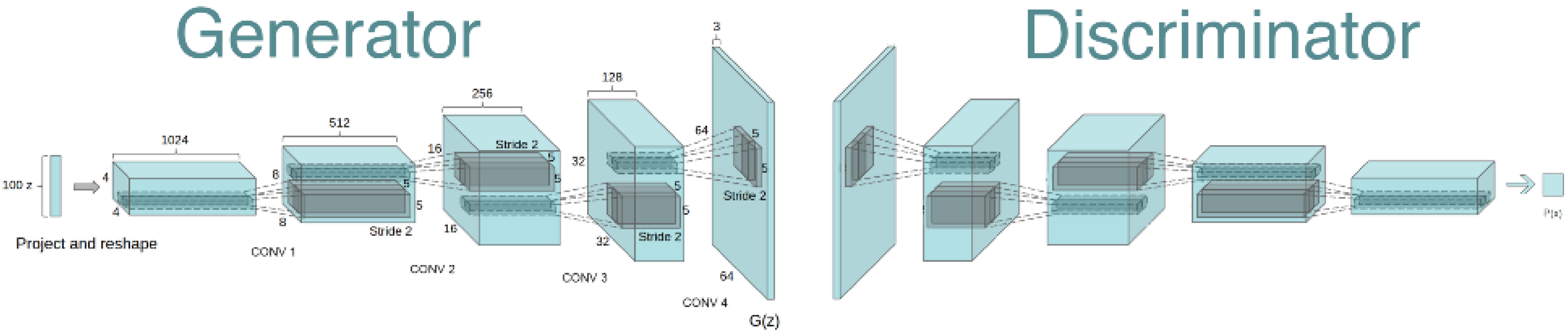
Magenta Project https://magenta.tensorflow.org/

Generative Adversarial Network

- A **Generator Model** G learns to capture the data distribution
- A **Discriminator Model** D estimates the probability that a sample came from the data distribution rather than model distribution.
- This can be think of the minimax game between two networks Discriminator and Generator.



Generator vs Discriminator



Objective of the **Generator** is to **cheat** the **Discriminator** by simulating **Realistic** images

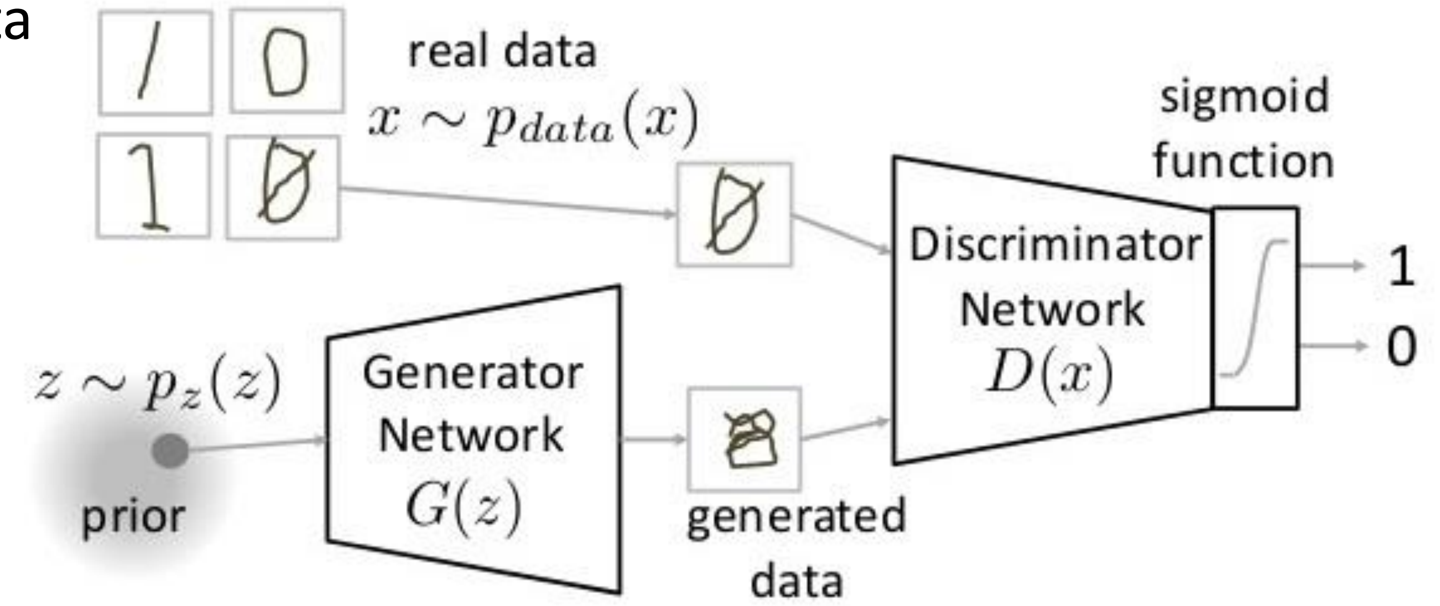
Objective of the **Discriminator** is to find out whether image is **Real** or **Fake**

Training a GAN

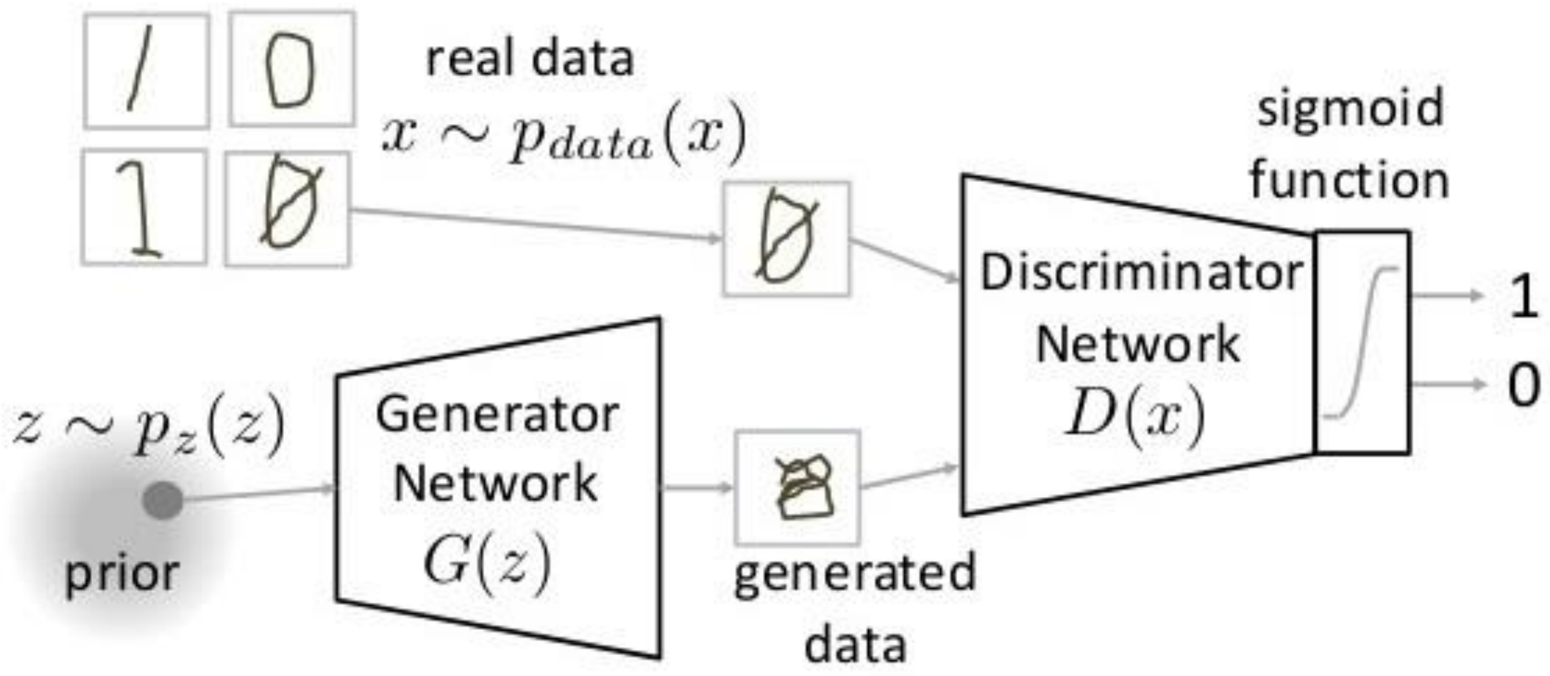
Pdata(x) - Distribution of real data
X - sample from pdata(x)

P(z) - distribution of generator
Z - sample from p(z)

G(z) - Generator Network
D(x) - Discriminator Network



Training a GAN

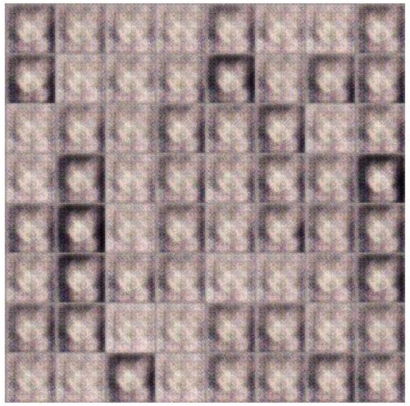


GANs objective is to solving a minimax problem

$$\min_G \max_D V(D, G)$$

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

GAN Training Example



100 rounds

1000 rounds

2000 rounds

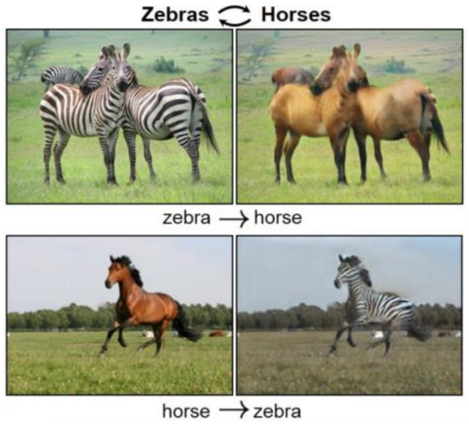
5000 rounds

10,000 rounds

50,000 rounds

Recent GAN Applications

Anime Characters



CycleGAN

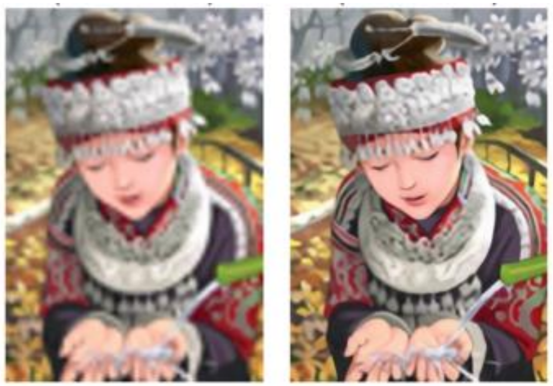


Image Super Resoultion

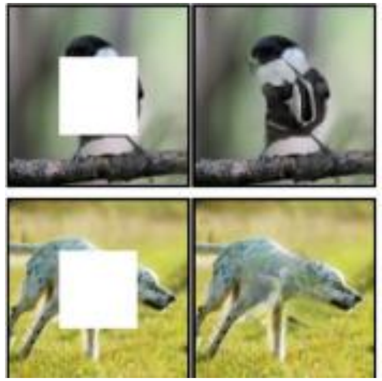


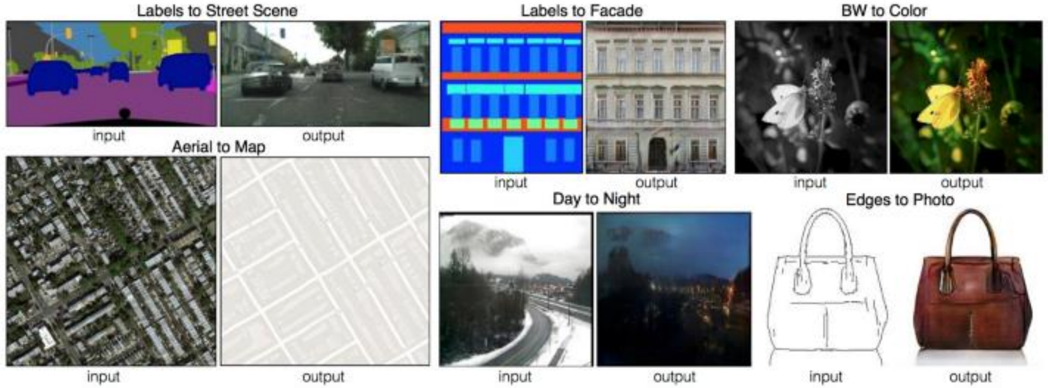
Image Inpainting

Pose Guided Person Image Generation

Recent GAN Applications

Fake Celebrities

Text to Image



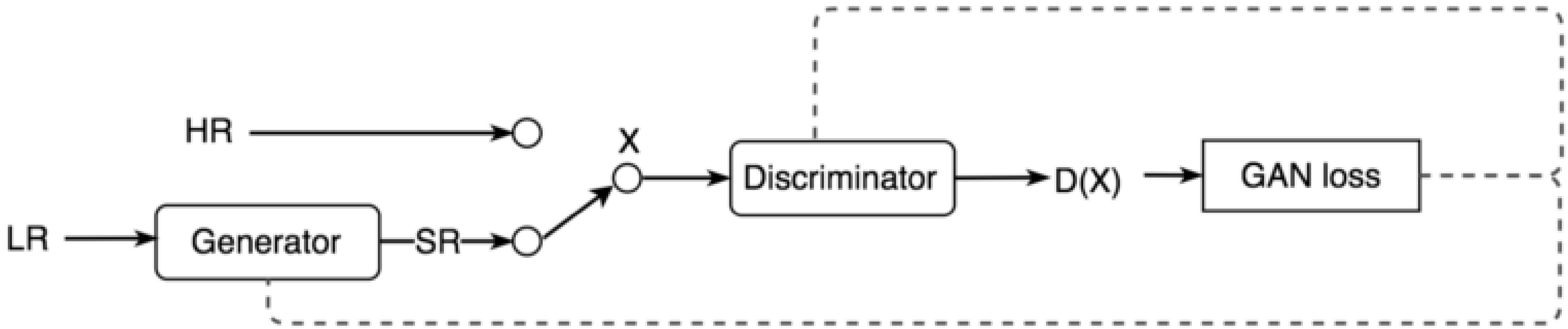
Pix2Pix

Next Frame Prediction

Emoji Generator

GANs for Image Super-resolution: SRGAN

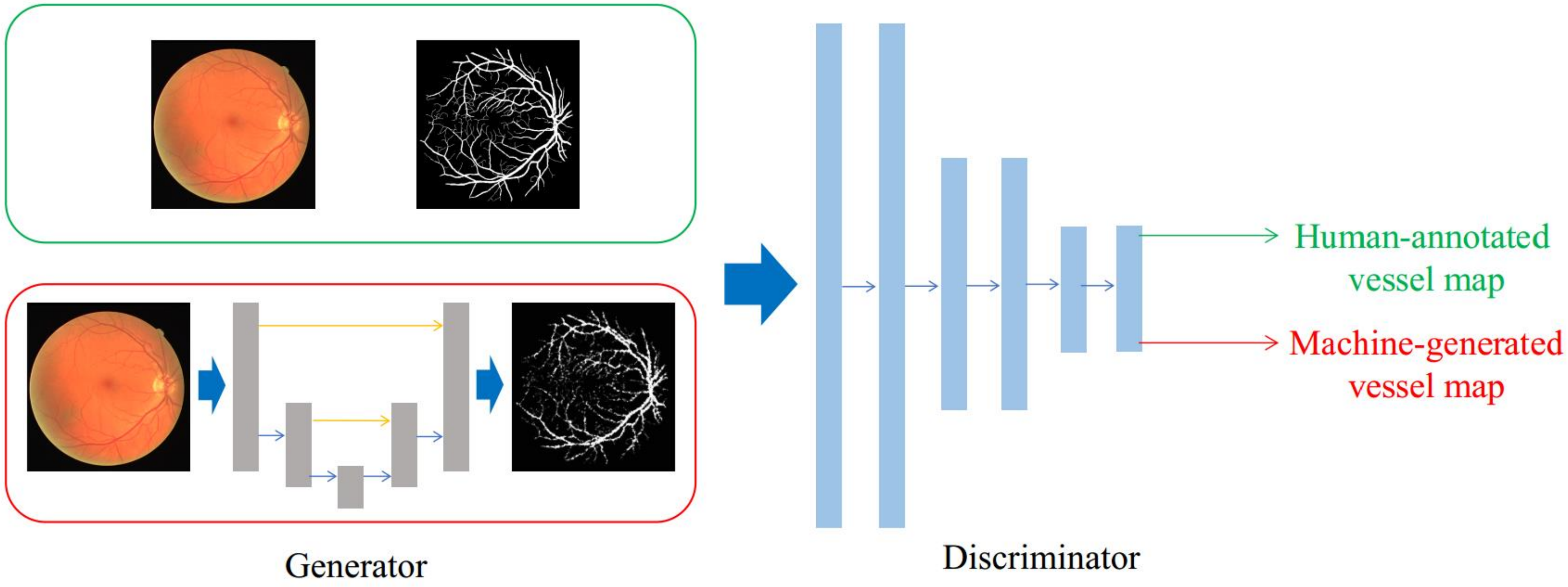
- Generator: gets low-resolution image as input and super-resolution image as output
- Discriminator: aims to differentiate between high-resolution image vs super-resolution image



Architecture of SRGAN

GANs for Image Segmentation

- Recently GANs are highly applied for medical image segmentation
- Generator is a segmentation network where discriminator will predict whether it is real segmentation or generated from segmentation



Questions ?

Thanks