Department of Computer Engineering
University of Kurdistan

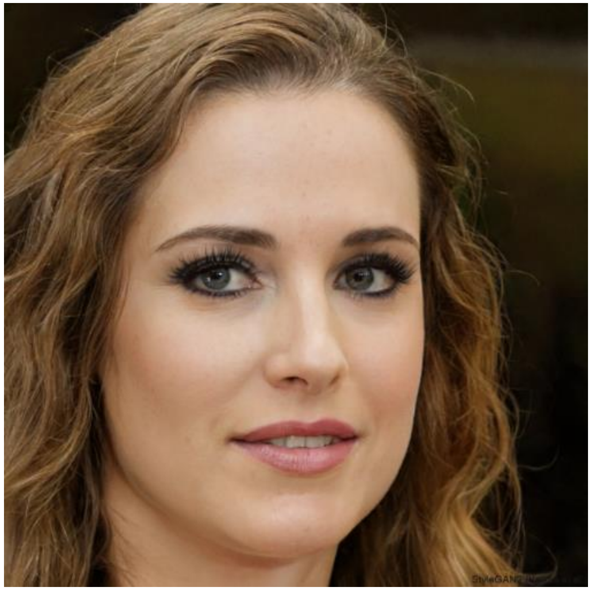
Deep Learning (Graduate level)

Generative Adversarial Networks (GAN)

By: Dr. Alireza Abdollahpouri

Which face is real?

A



B

C

This Person Does Not Exist

website ThisPersonDoesNotExist.com. In the image below you can see the several fake faces that are generated by the GANs.

Generative Adversarial Networks

Ian Goodfellow and co-authors
(NeurIPS) conference 2014

What is Generative AI?

- Generative AI is a type of artificial intelligence that uses neural networks to create text, images, and other content.

- Generative models are trained on large amounts of data, and they learn to identify patterns in the data.

- Once a generative model has been trained, it can be used to generate new data samples.

Generative AI Models

- **Generative Adversarial Networks** (GANs): Two neural networks, a generator and a discriminator, are trained in opposition. Applications: Image generation, style transfer, data augmentation.

- **Variational Autoencoders** (VAEs): Learn latent representations of data. Applications: Image generation, anomaly detection, data compression.

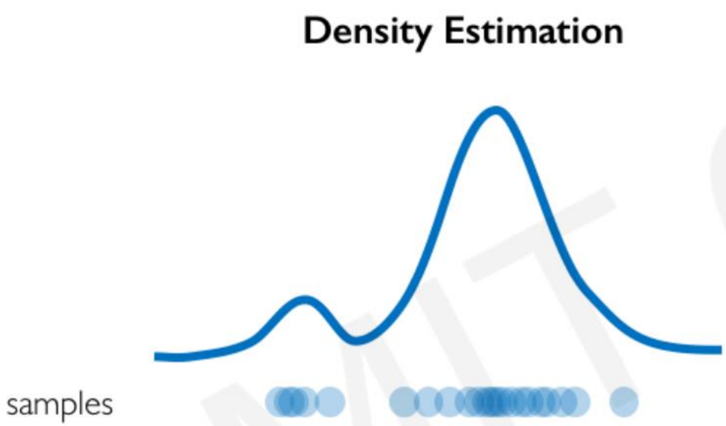
- **Large Language Models** (LLMs):Process and generate human-like text. Applications: Writing, translation, coding assistants, chatbots.

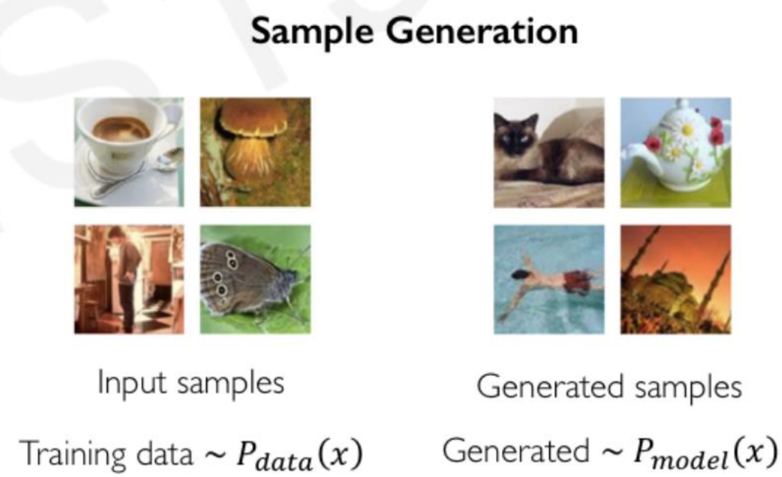
- **Diffusion Models**: Gradually add noise to an image and then denoise it. Applications: Image generation, image editing, text-to-image generation.

Generative Modeling

Goal: take as input training samples from some distribution and learn a model that represents the distribution.

Two operations:





How can we learn $P_{model}(x)$ similar to $P_{data}(x)$?

Generative Modeling- Debiasing

Capable of uncovering **underlying features** in a dataset

VS

Homogeneous skin color, pose

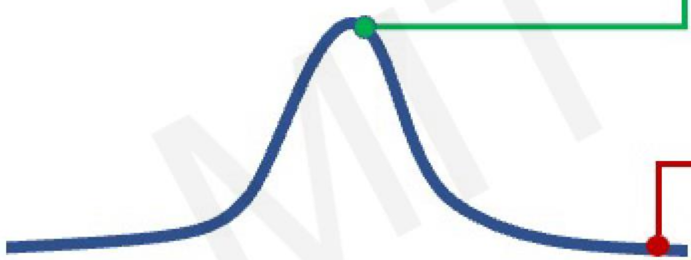
Diverse skin color, pose, illumination

How can we use this information to create fair and representative datasets?

Generative Modeling- Outlier detection

- **Problem:** How can we detect when we encounter something new or rare?
- **Strategy:** Leverage generative models, detect outliers in the distribution
- Use outliers during training to improve even more!

95% of Driving Data:
(1) sunny, (2) highway, (3) straight road



Detect outliers to avoid unpredictable behavior when training

Edge Cases

Harsh Weather

Pedestrians

Generative Adversarial Networks

Catch Me If You Can

generator discriminator

Two agents play a **minimax** game

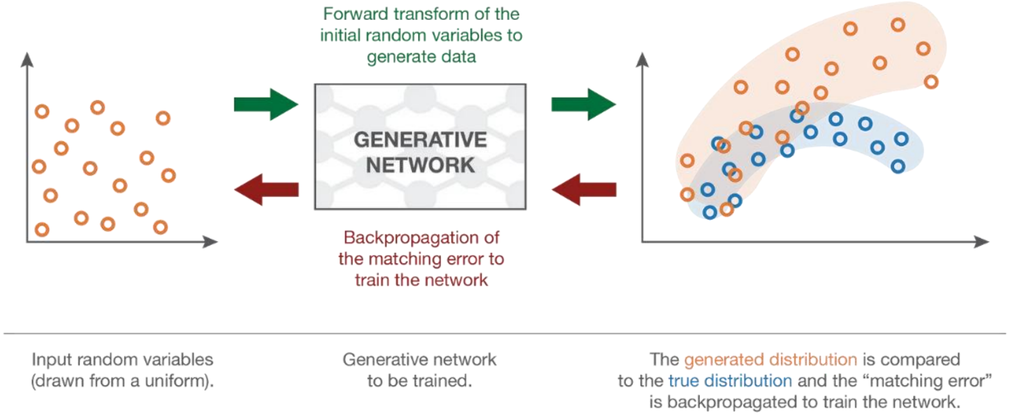
Generative Adversarial Networks (GAN)

- GANs are a framework where two neural networks, a generator and a discriminator, are trained adversarially.
- The generator learns to create synthetic data that mimics the real data distribution, aiming to fool the discriminator.
- Simultaneously, the discriminator learns to distinguish between real samples and the generator's fakes.
- Through this competition, both networks improve until the generator produces highly realistic outputs.

Generative Adversarial Networks

Generator

Generates candidates/images (from a probability distribution). It's objective is to 'fool' the discriminator by producing novel synthesized instances that appear to come from the true data.



Discriminator

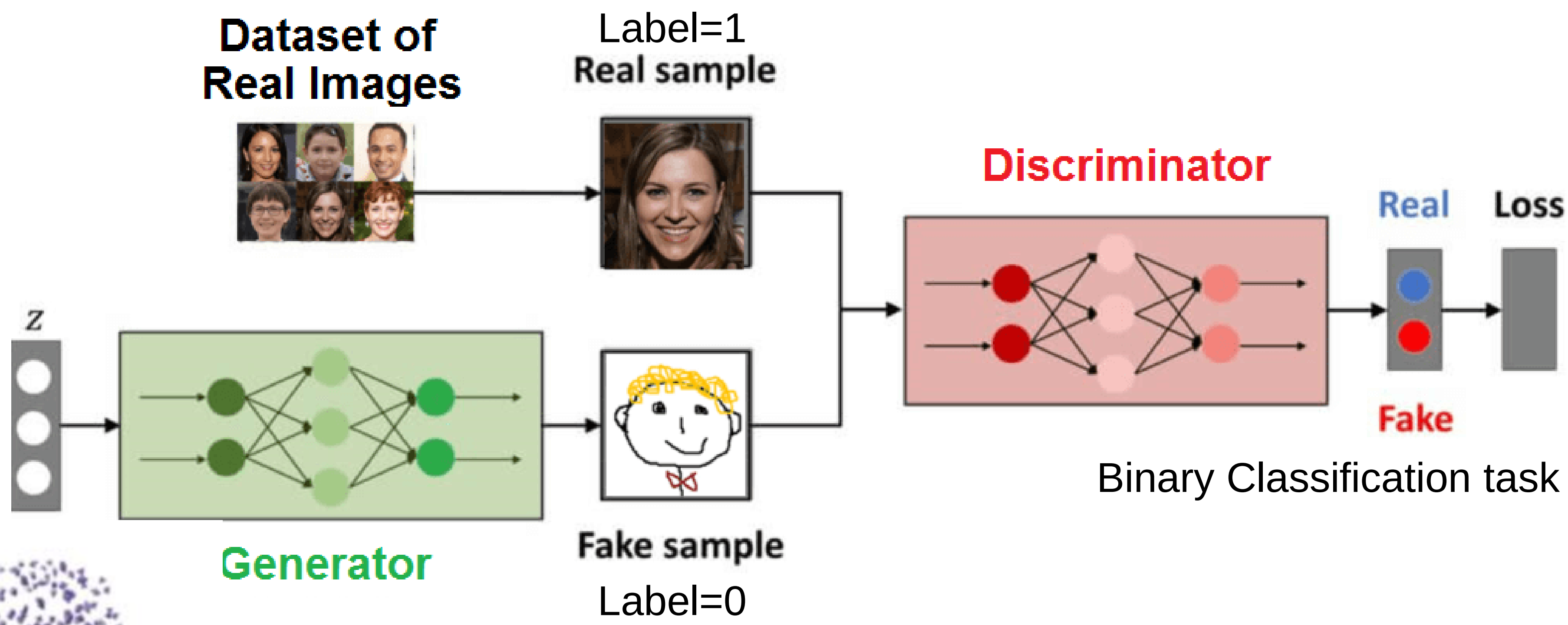
Evaluates the generated images to see if they come from the true data or not

Backpropagation applied to both networks:

•Generator to produce better images

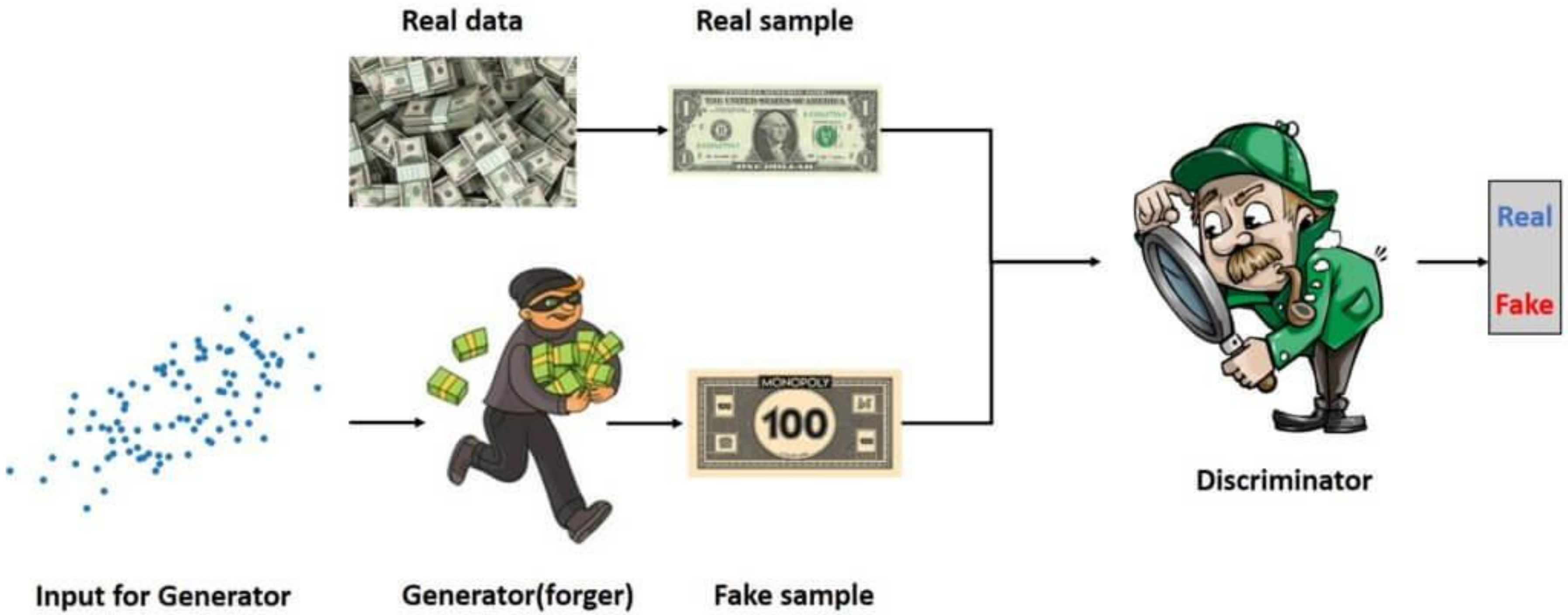
•Discriminator to be more skilled at evaluating generated images

Generative Adversarial Networks

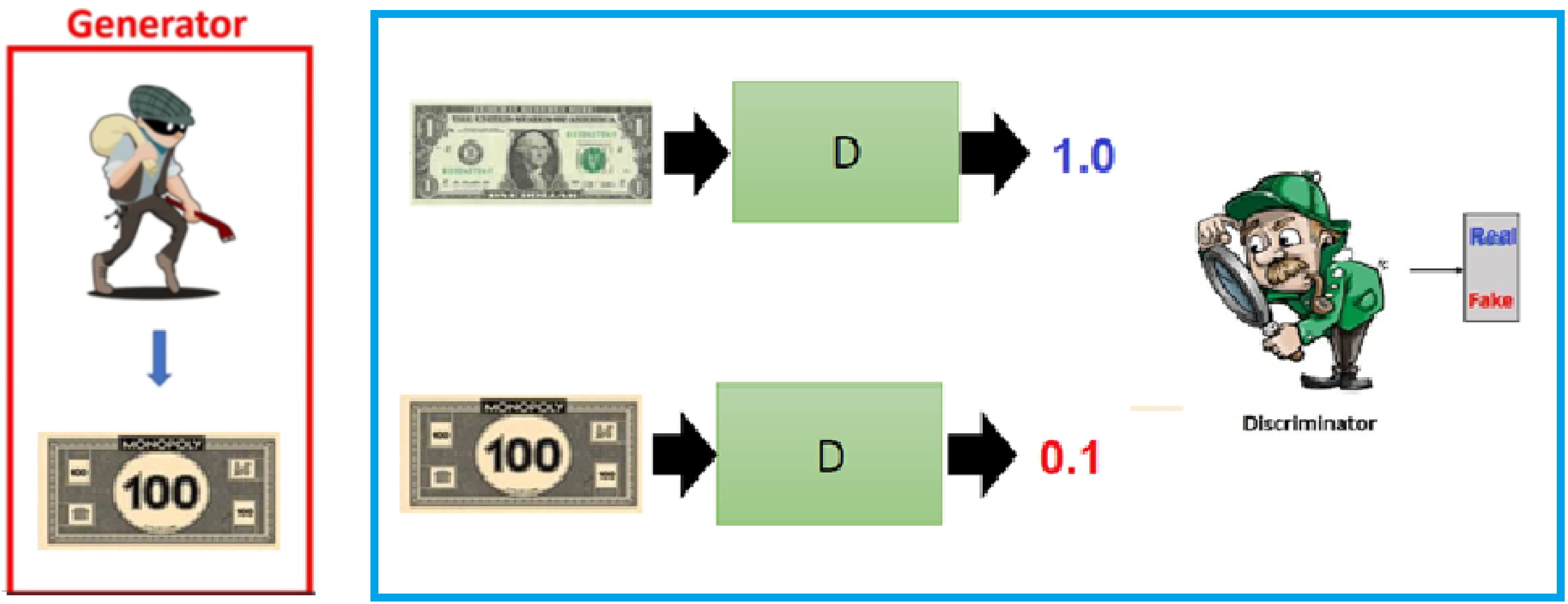


I: Input for Generator

Generative Adversarial Networks



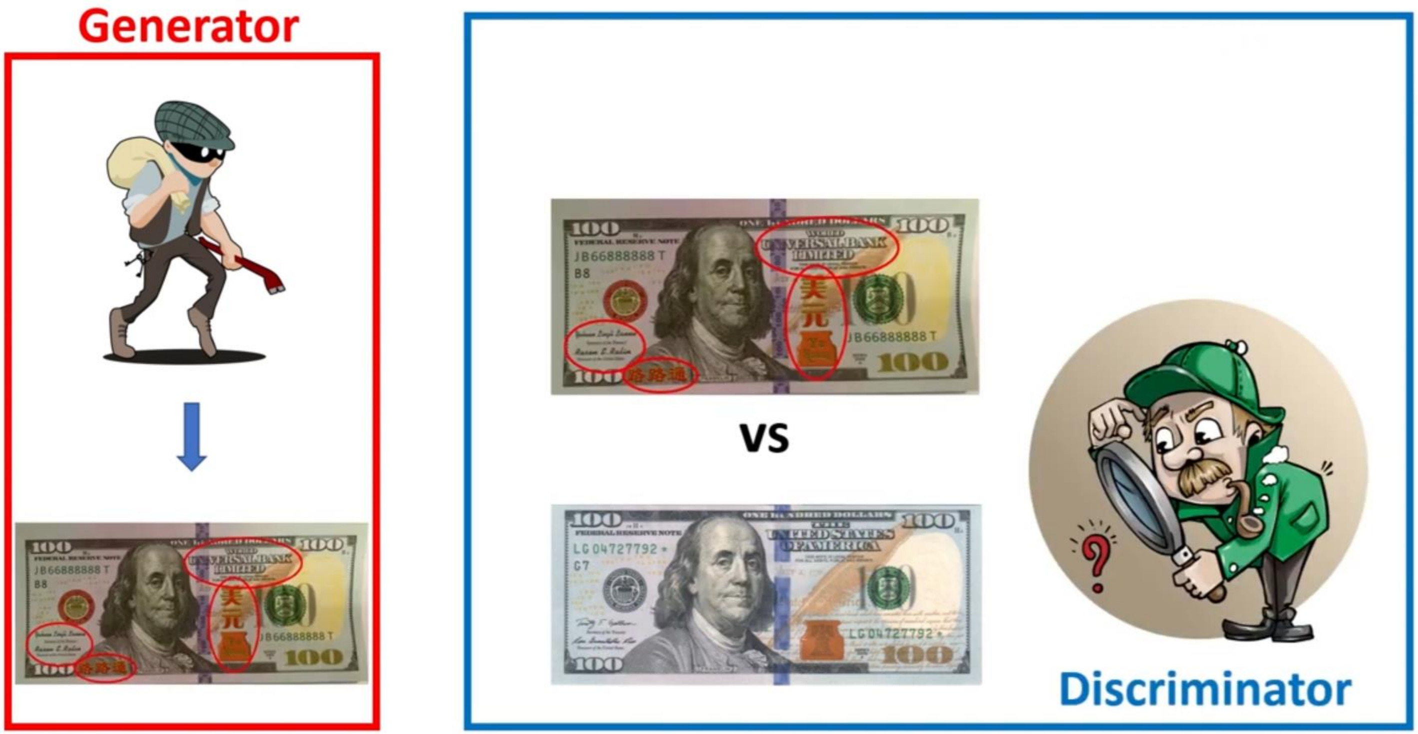
Generator & Discriminator



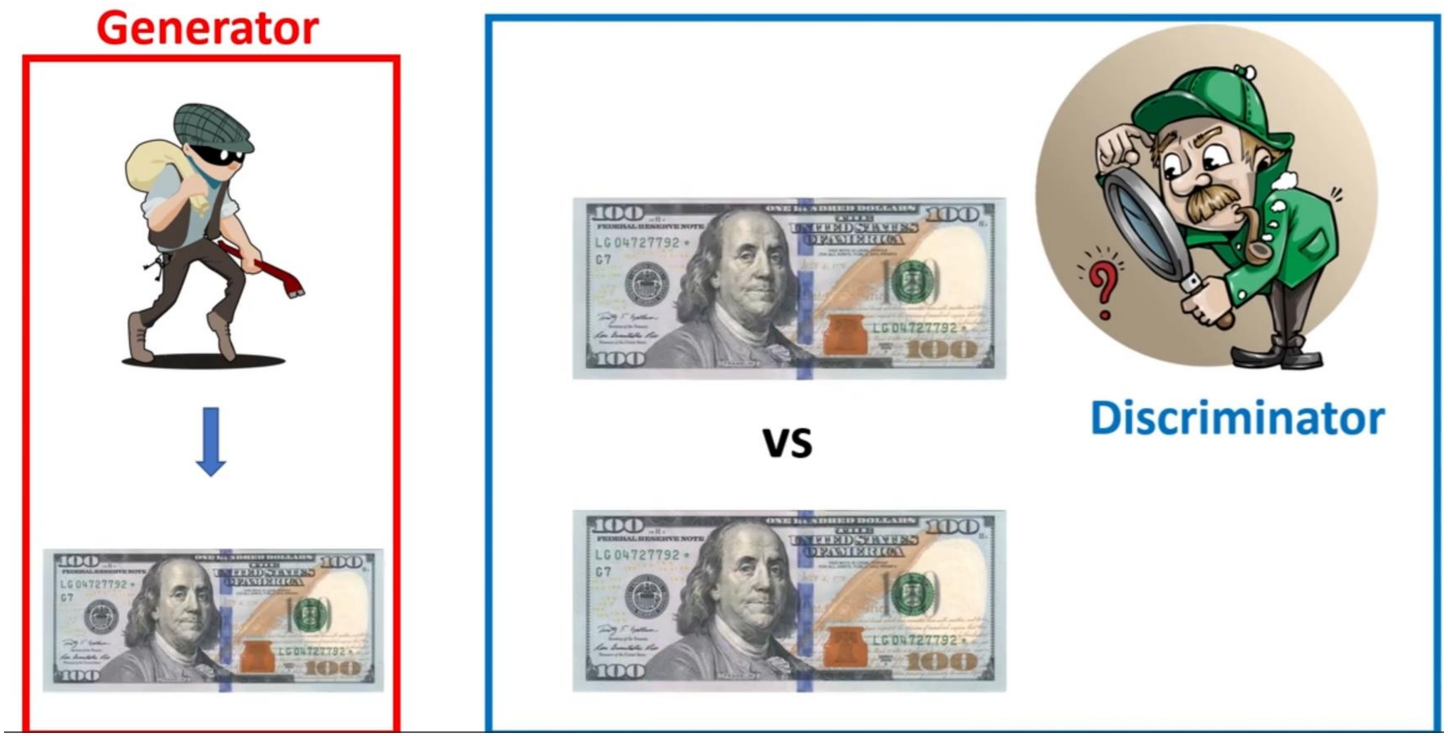
Generator

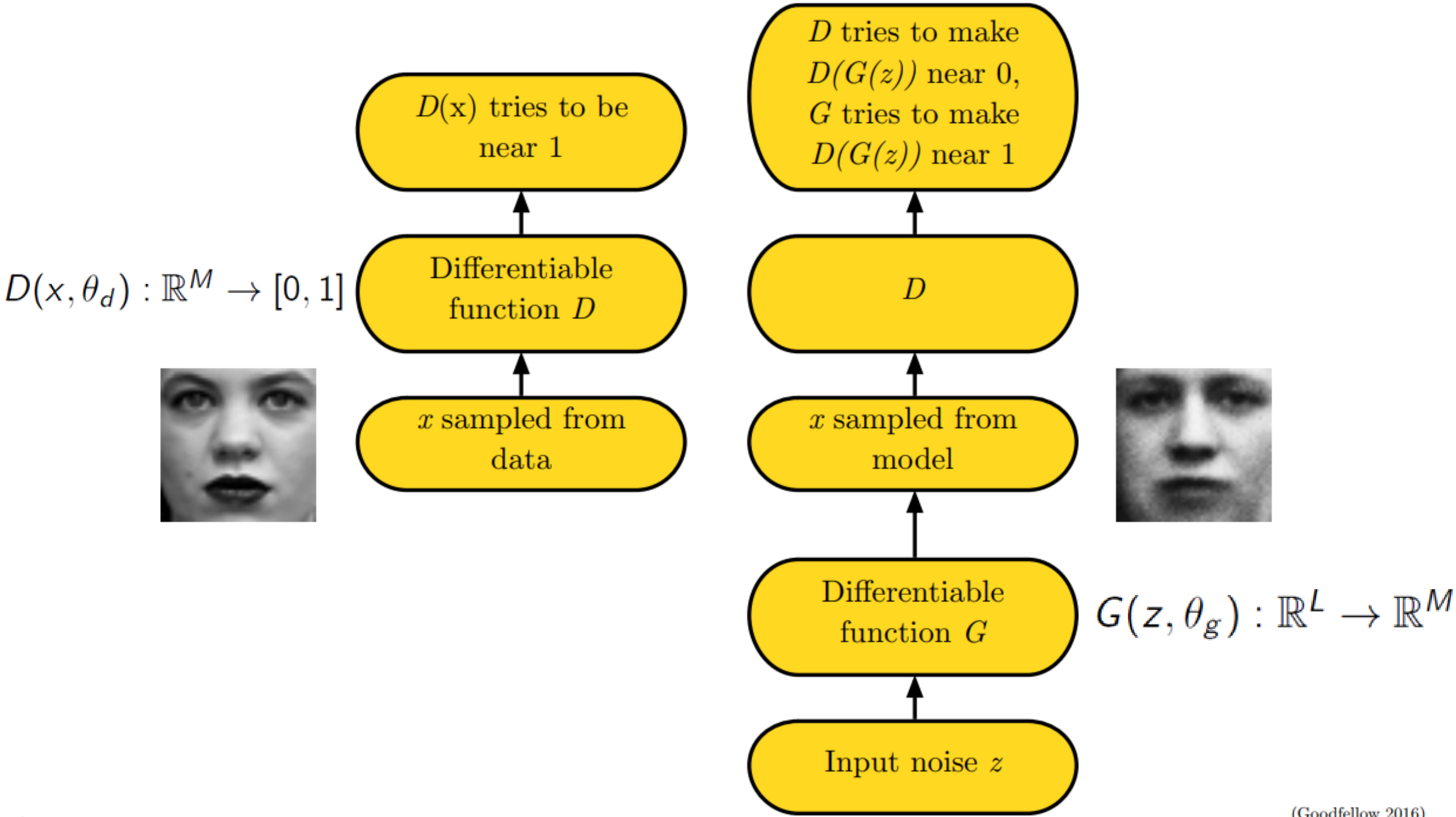
Discriminator

Generator & Discriminator



Generator & Discriminator

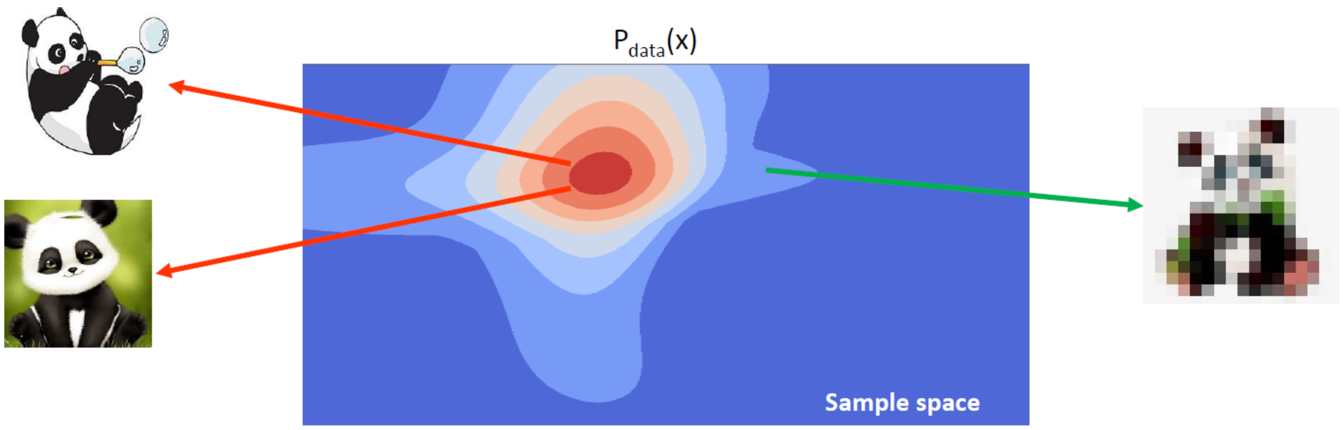


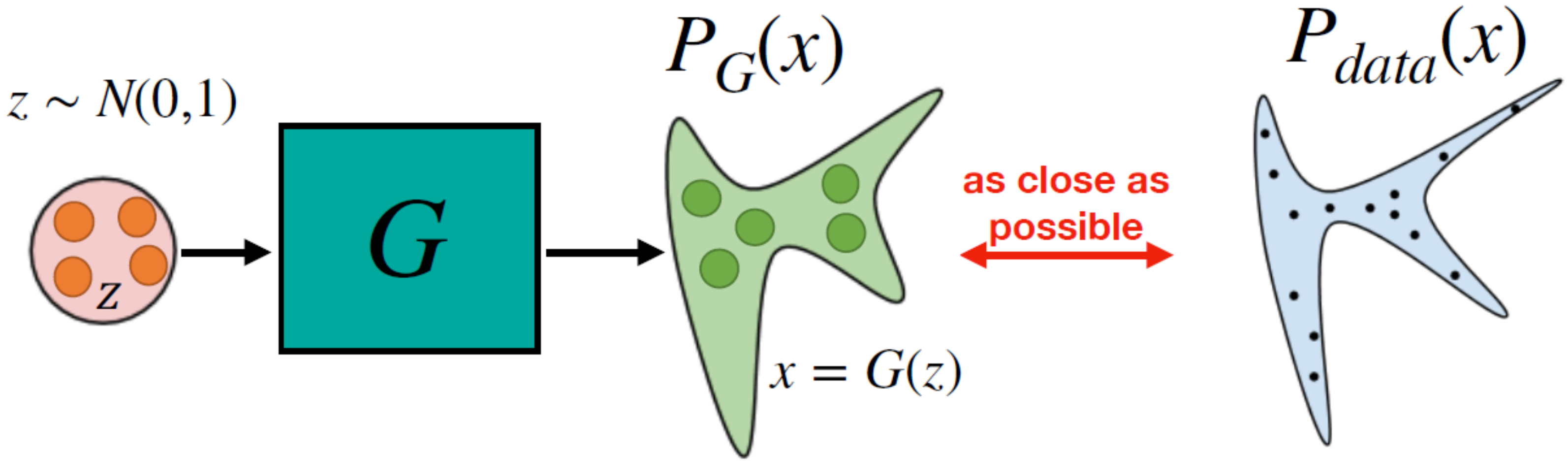


(Goodfellow 2016)

Probability distribution of real data

- $P_{data}(x)$
- x: samples





GAN - Problem

- **Generator**:
  - Needs to learn distribution p_g over data instances $x \in \mathbb{R}^M$
  - $G(z, \theta_g) : \mathbb{R}^L \to \mathbb{R}^M$ is a neural network
  - z is a noise fed into the generator following a prior on $z \sim p_z(z)$
- **Discriminator**:
  - $D(x, \theta_d) : \mathbb{R}^M \to [0, 1]$ is a neural network
  - $D(x)$ is the probability that x comes from real data rather than p_g
- GAN aims at learning θ_g and θ_d which optimize an objective V as $\min_G \max_D V(D, G)$, by:

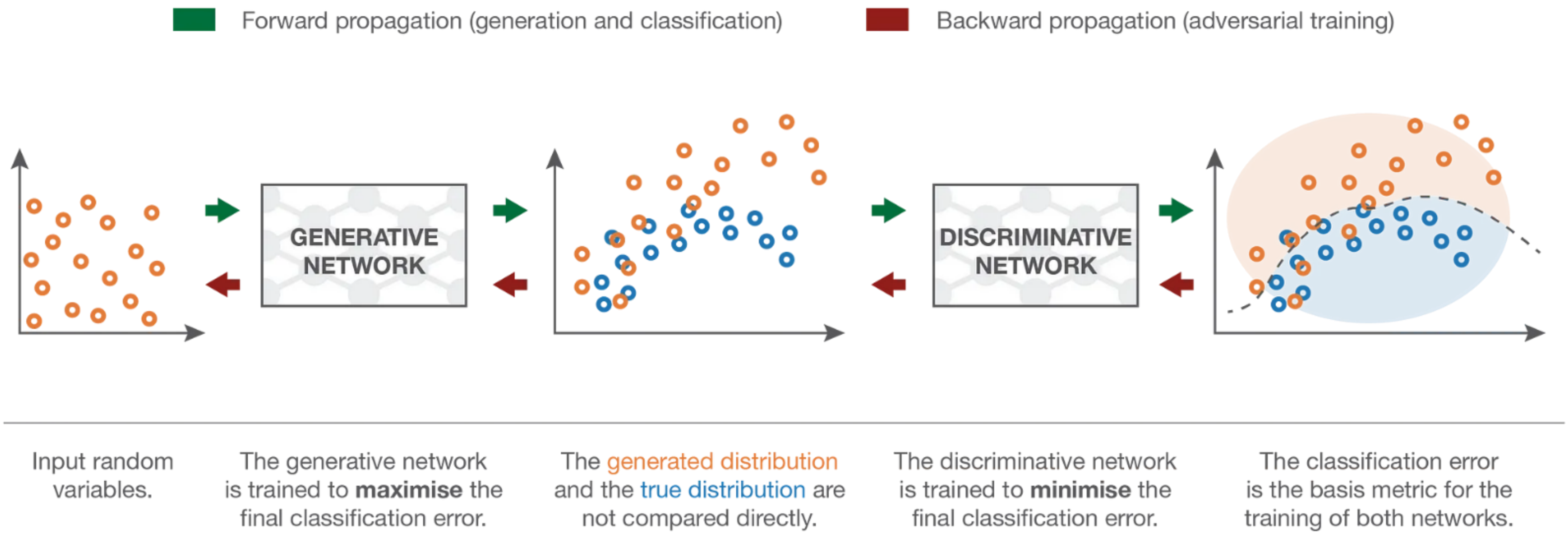
Minimax objective function

The discriminator aims to maximize its ability to distinguish real data from synthetic data produced by the generator, while the generator aims to minimize the discriminator's success by creating increasingly realistic samples (**fooling the discriminator**).

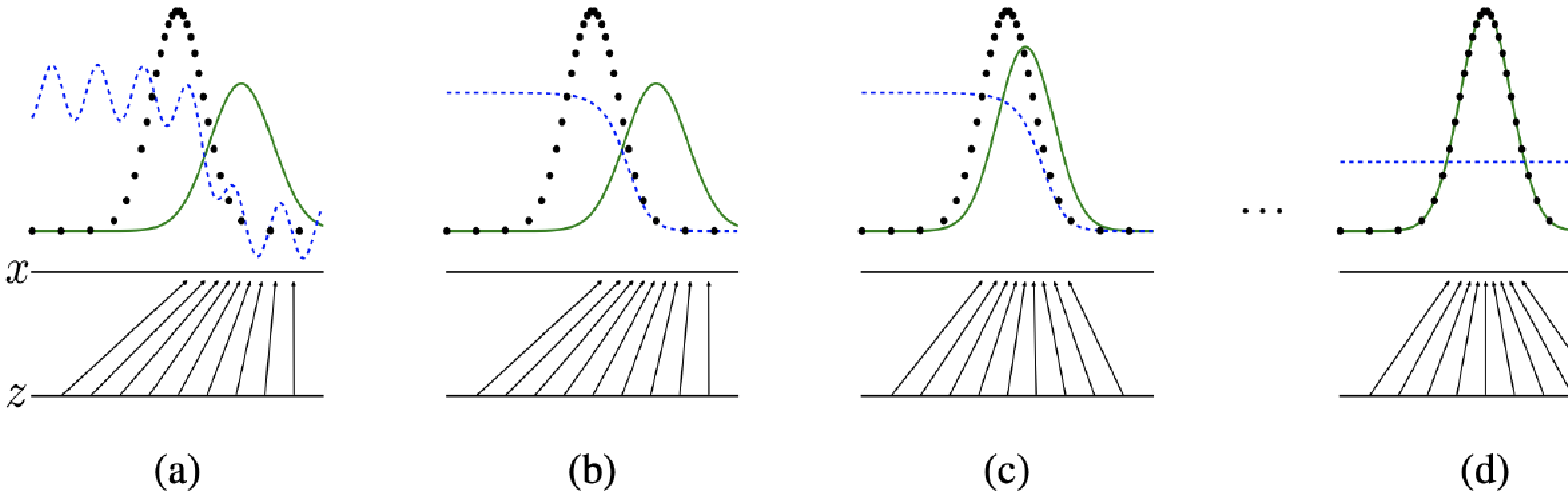
Minimax objective function:

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

GANs - Training Objective



Distributions during training



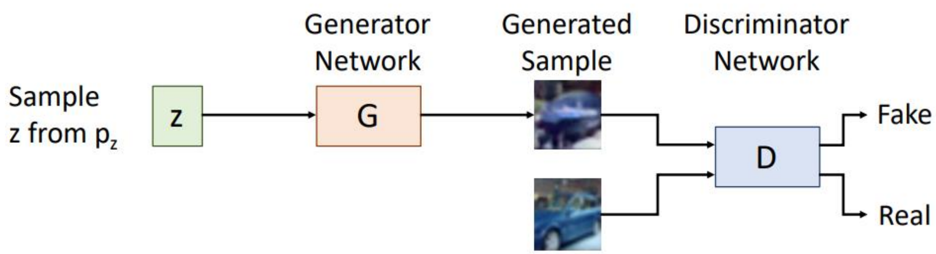
- Discriminator improves from (a) to (b)
- Generator improves from (b) to (c)
- Generator perfectly mimics the true data and the discriminator assigns probability1/2 everywhere in (d)

Images from the data set are **not labeled**

Mini-Max game

This game between generator and discriminator is th and can be described with the following equation.

$$\min_{G} \max_{D} \left(\overbrace{E_{x\sim p_{data}}[\log D(x)]}^{\text{Discriminator wants } D(x) = 1 \text{ for real data}} + E_{z\sim p(z)} \overbrace{\underbrace{\left[\log\left(1 - D(G(z))\right)\right]}_{\text{Generator wants } D(x) = 1 \text{ for fake data}}}^{\text{Discriminator wants } D(x) = 0 \text{ for fake data}} \right)$$

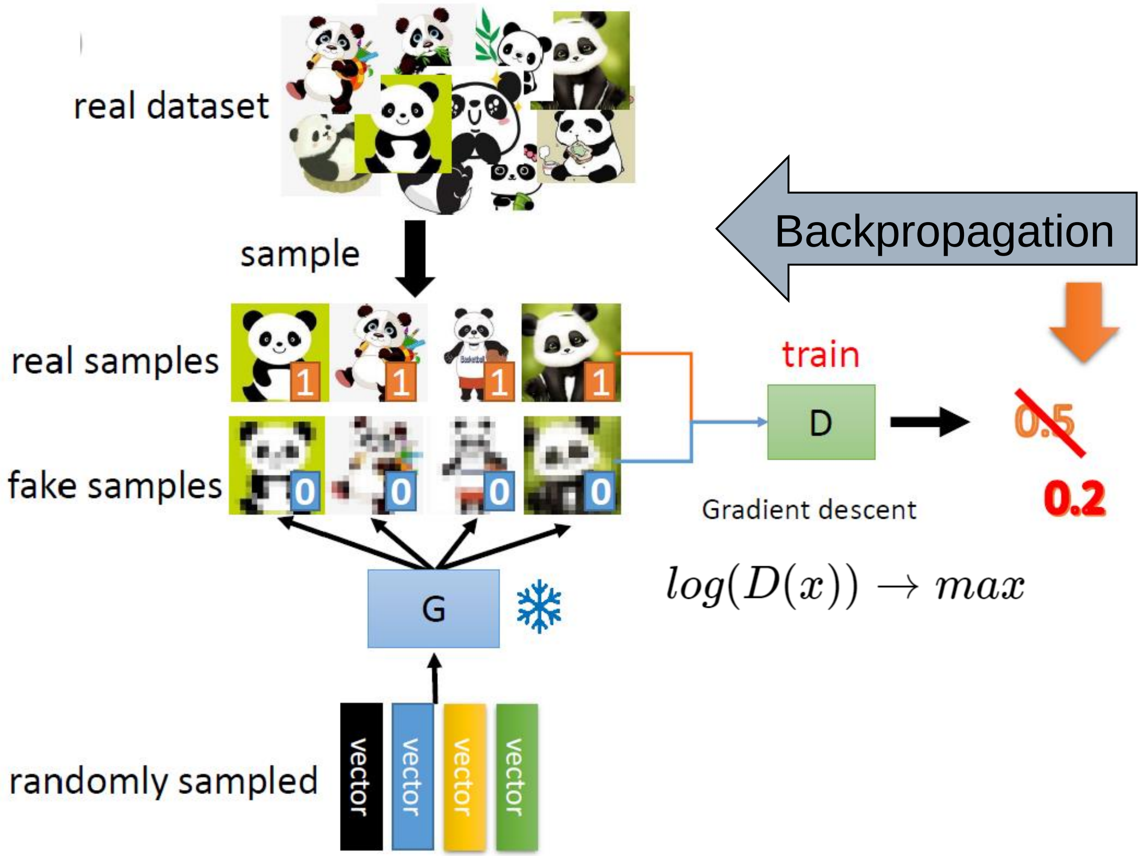


$$D(x) = \begin{cases} 1 & \text{real data} \\ 0 & \text{fake data} \end{cases} \qquad D(G(z)) = \begin{cases} 1 & \text{real data} \\ 0 & \text{fake data} \end{cases}$$

z is the noise vector

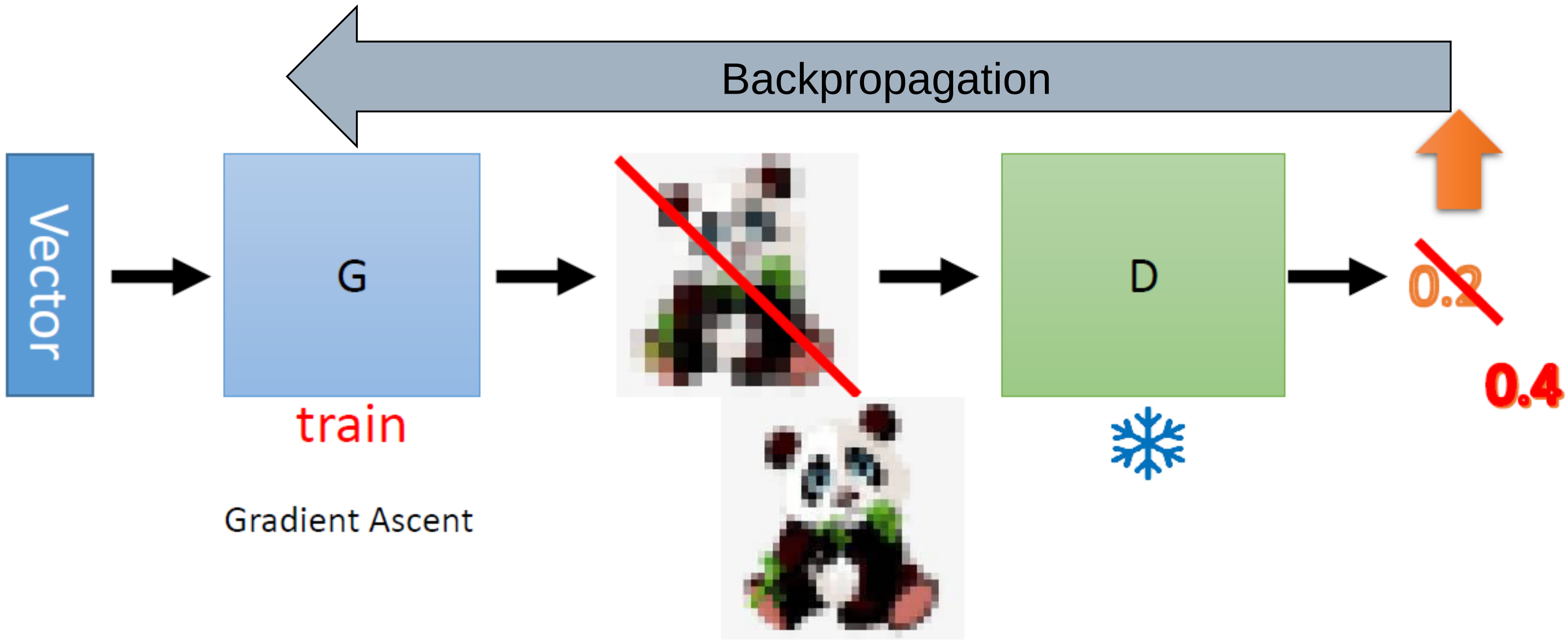
GAN Training Step 1: Train D (Fix G)

We sample a batch of random noise vectors and a batch of images from the data set. Then, we're going to **freeze the generator network** and use the objective function in order to update the parameters of the discriminator to calculate the loss.



GAN Training Step 2: Train G (Fix D)

Generator learns to "fool" the Discriminator. Hence, it will try to do the opposite which is to minimize the objective function. $\log(D(G(z))) \rightarrow \min$



GAN Optimization

Algorithm 1: GAN Optimization

for $1, \ldots, \text{numIters}$ **do**

  for $1, \ldots, K$ **do**

    Sample n noise samples: $\{z^{(1)}, \ldots, z^{(n)}\}$ from $p_z(z)$

    Sample n real samples: $\{x^{(1)}, \ldots, x^{(n)}\}$ from $p_{\text{data}}(x)$

    Update discriminator parameters θ_d using gradient **ascent**:

$$\nabla_{\theta_d} \frac{1}{n} \sum_{i=1}^{n} \log D(x^{(i)}, \theta_d) + \log\left(1 - D(G(z^{(i)}, \theta_g), \theta_d)\right)$$

  Sample n noise samples: $\{z^{(1)}, \ldots, z^{(n)}\}$ from $p_z(z)$

  Update generator parameters θ_g using gradient **descent**:

$$\nabla_{\theta_g} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 - D(G(z^{(i)}, \theta_g), \theta_d)\right)$$

GAN Variants

Conditional GAN (cGAN) : In a conditional GAN (cGAN), **both the generator and the discriminator are conditioned on some extra information, such as a class label or other external input.**

Deep Convolutional GAN (DCGAN): To generate high-quality images. In this model, both the generator and discriminator are built using convolutional neural networks (CNNs).

StyleGAN is a sophisticated variant of GANs designed to generate ultra-realistic, high-resolution images with precise control over style and appearance. By separating high-level attributes (like pose) from low-level details (like textures)

CycleGAN

CycleGAN **offers a unique capability in the field of image-to-image translation** without the need for paired training data.

More GANs

- DCGAN
- WGAN
- CGAN
- LAPGAN
- SRGAN
- CycleGAN
- WGAN-GP
- EBGAN
- VAE-GAN
- BiGAN
- SGAN
- SimGAN
- VGAN
- iGAN
- 3D-GAN
- CoGAN
- Cat-GAN
- MGAN
- S^2GAN
- LSGAN
- AffGAN
- TP-GAN
- IcGAN
- ID-CGAN
- AnoGAN
- LS-GAN
- Triple-GAN
- TGAN
- BS-GAN
- MalGAN
- RTTGAN
- GANCS
- SSL-GAN
- MAD-GAN
- PrGAN
- AL-CGAN
- ORGAN
- SD-GAN
- MedGAN
- SGAN
- SL-GAN
- SketchGAN
- GoGAN
- RWGAN
- MPM-GAN
- MV-GAN
- StyleGAN
- GANSynth
- ProGAN
- Context-RNN-GAN

Applications of GANs

- Image Generation
- Image-to-Image Translation
- Super-Resolution Imaging
- Video Generation
- Text-to-Image Generation
- Music Compositio
- Text Generation
- Machine Translation
- Text-to-Speech Conversion
- …

Image Inpainting

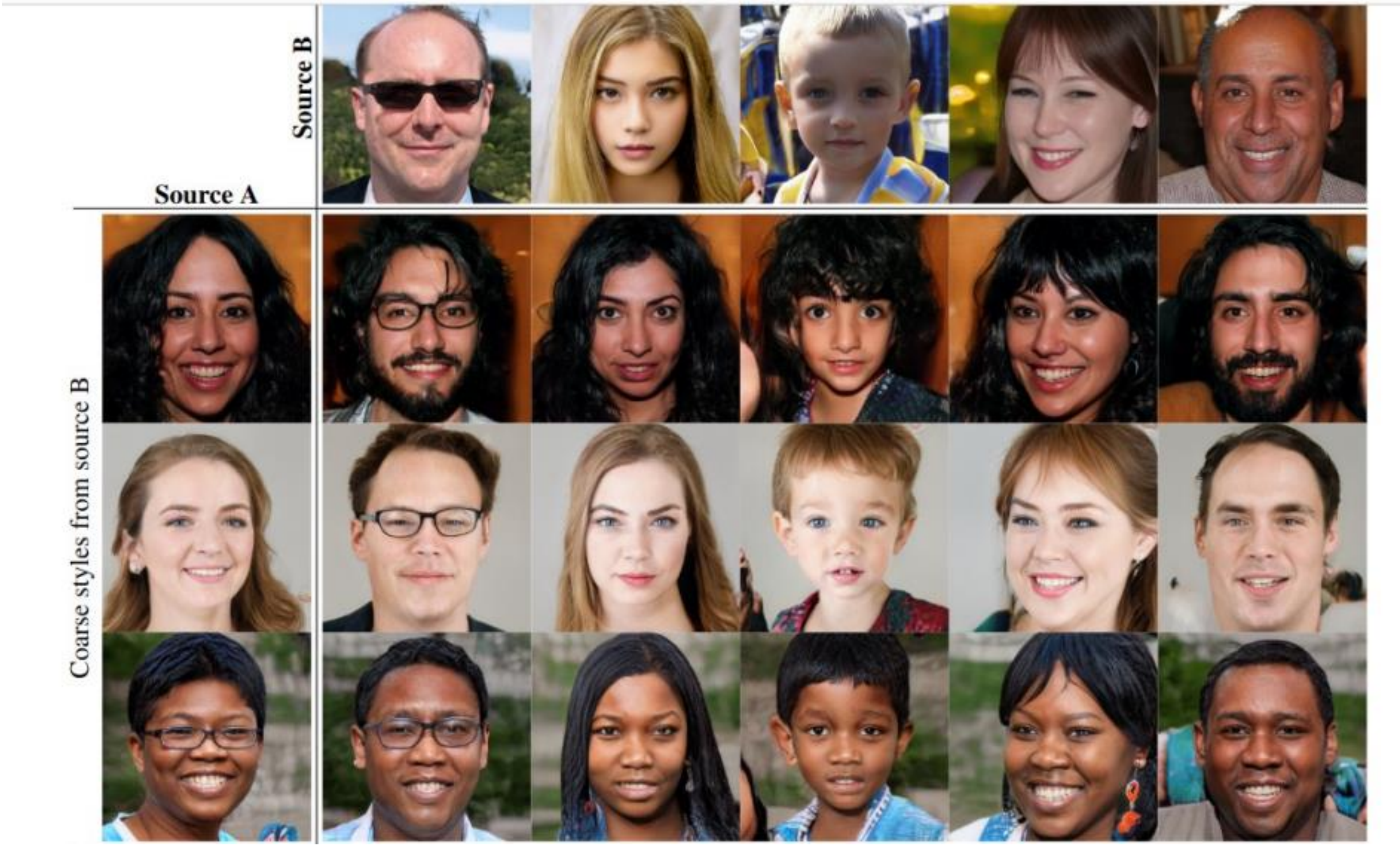
Conditional Image

Inpainting with L2 loss

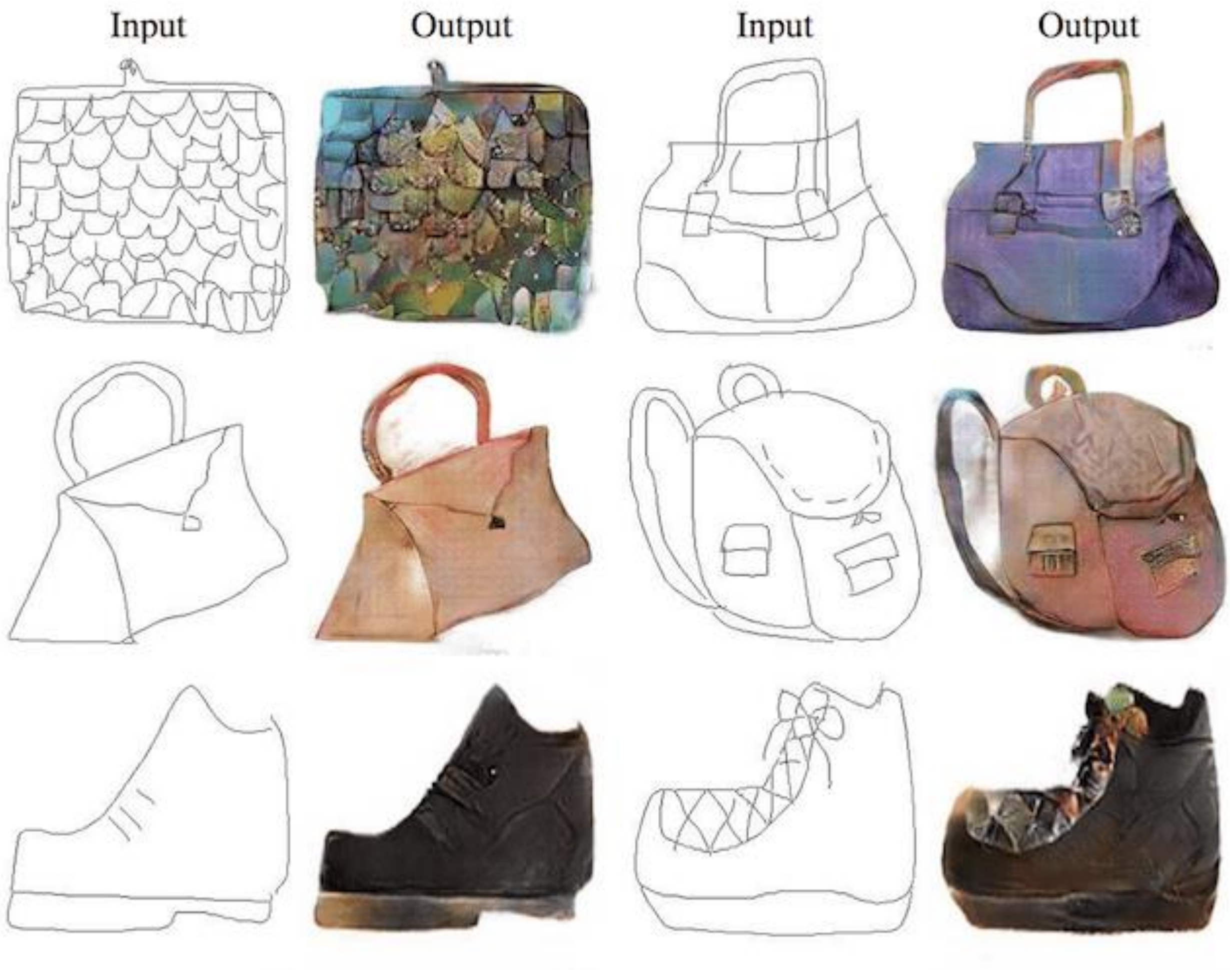
Inpainting with CGAN

Context Encoders: Feature Learning by Inpainting, D.Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A. Efros,2016

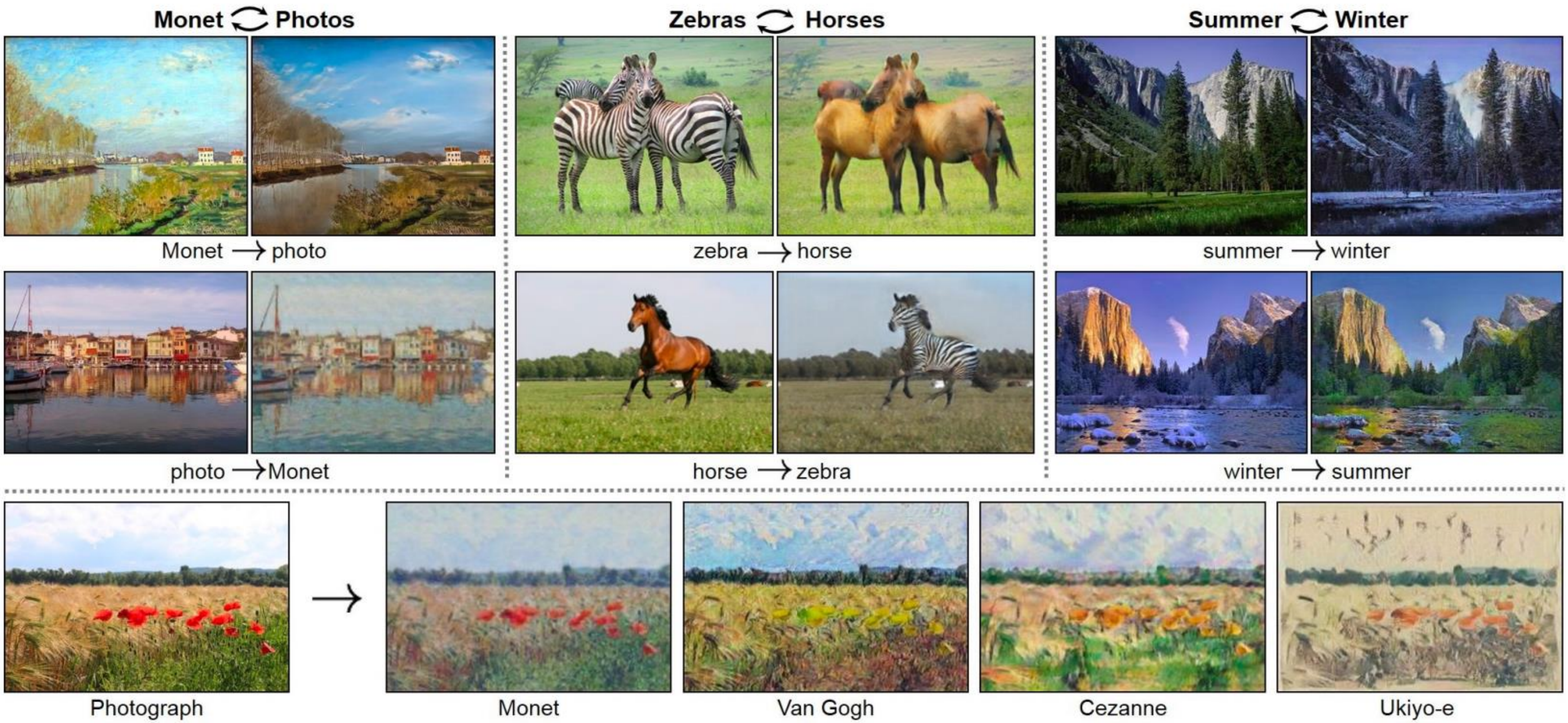
Mixing styles from two source images



CycleGAN (Image-to-Image Translation)



CycleGAN (Image-to-Image Translation)

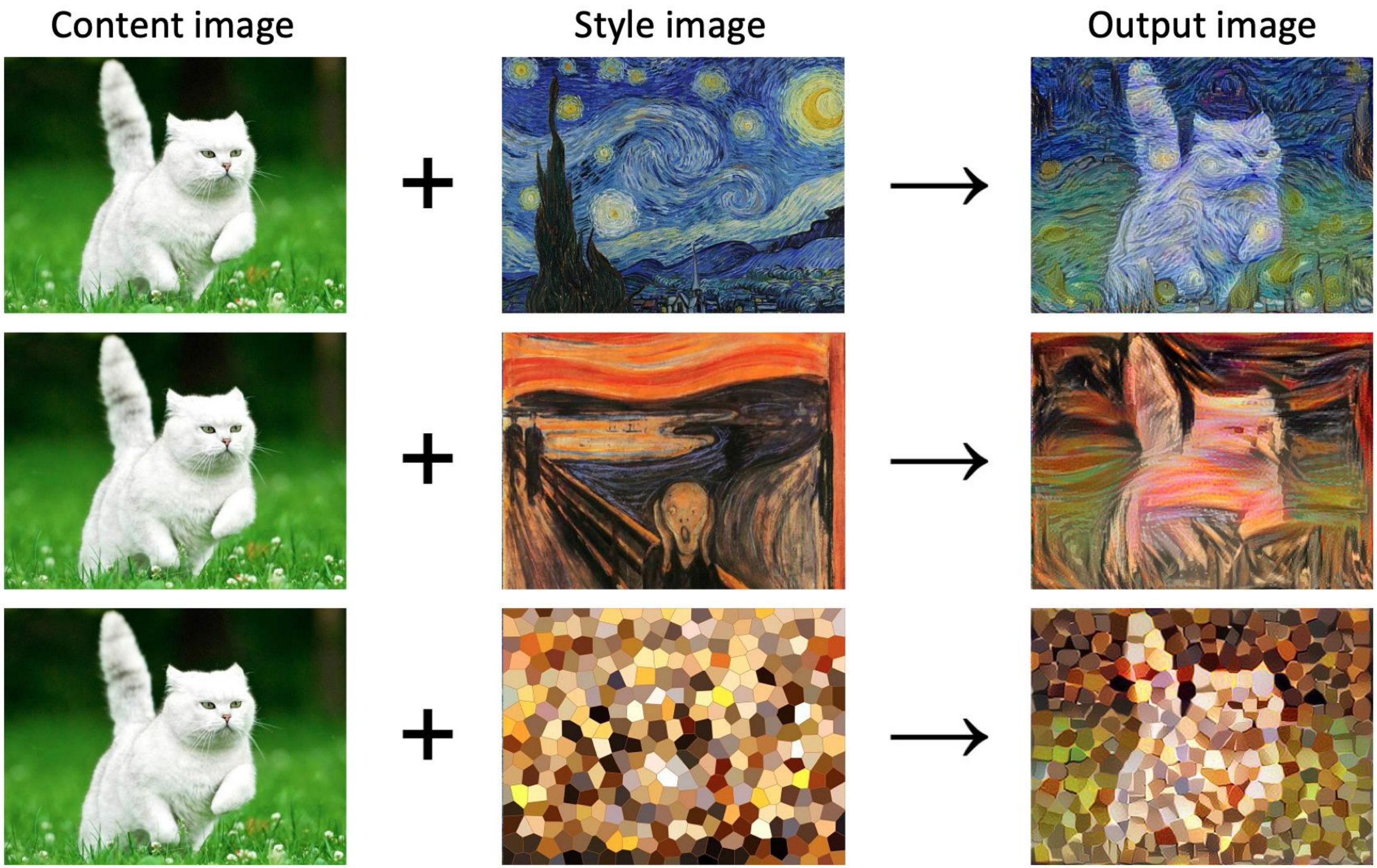


https://github.com/junyanz/CycleGAN

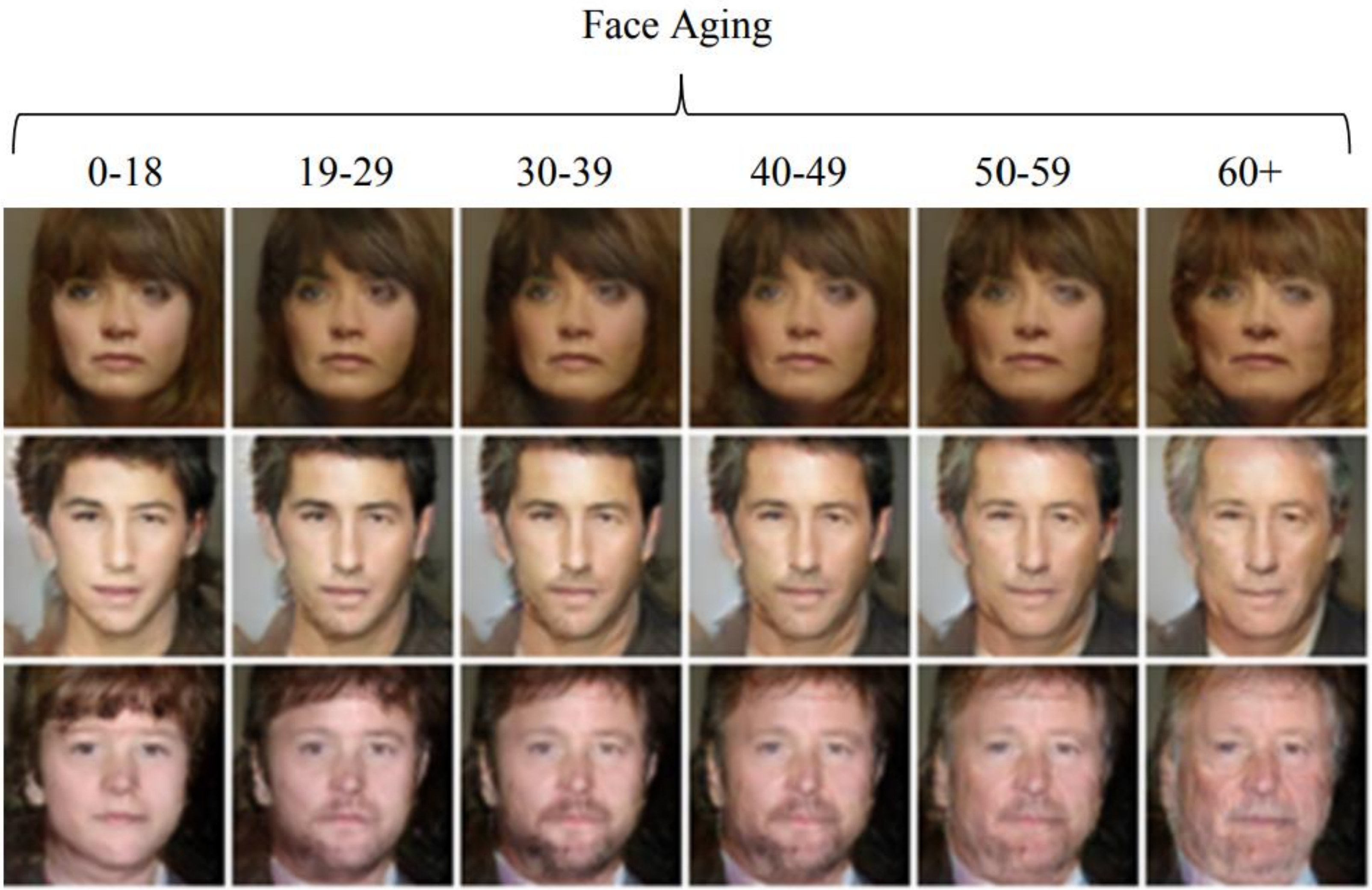
CycleGAN (Video-to-Video Translation)

https://github.com/junyanz/CycleGAN

Style transfer



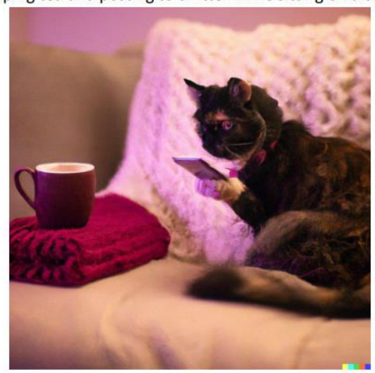
Face Aging



OpenAI-DALL E-2: Text-to-Image

A photo of a quaint flower shop storefront with a pastel green and clean white facade and open door and big window

Cat sipping tea and posting to twitter while sitting on a couch



A rabbit detective sitting on a park bench and reading a newspaper in a victorian setting

A lion in a hoodie hacking on a laptop

Teddy bears shopping for groceries in ancient Egypt

Teddy bears working on new AI research on the moon in the 1980s

(Diffusion Model)

OpenAI-Sora: Text-to-Video

Sora (2024)

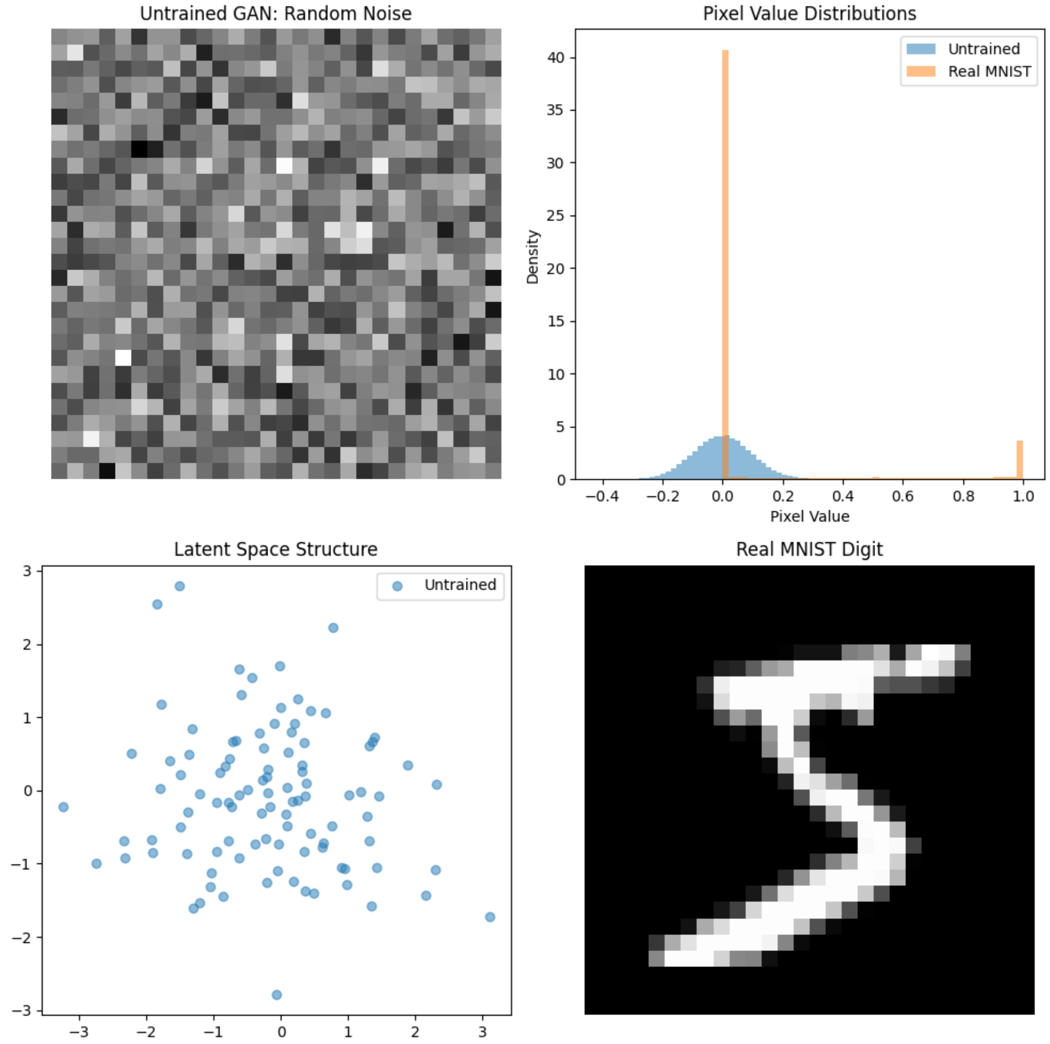
Create real high quality videos from a text description

https://openai.com/sora

Prompt: Several giant wooly mammoths approach treading through a snowy meadow, their long wooly fur lightly blows in the wind as they walk, snow covered trees and dramatic snow capped mountains in the distance, mid afternoon light with wispy clouds and a sun high in the distance creates a warm glow, the low camera view is stunning capturing the large furry mammal with beautiful photography, depth of field.

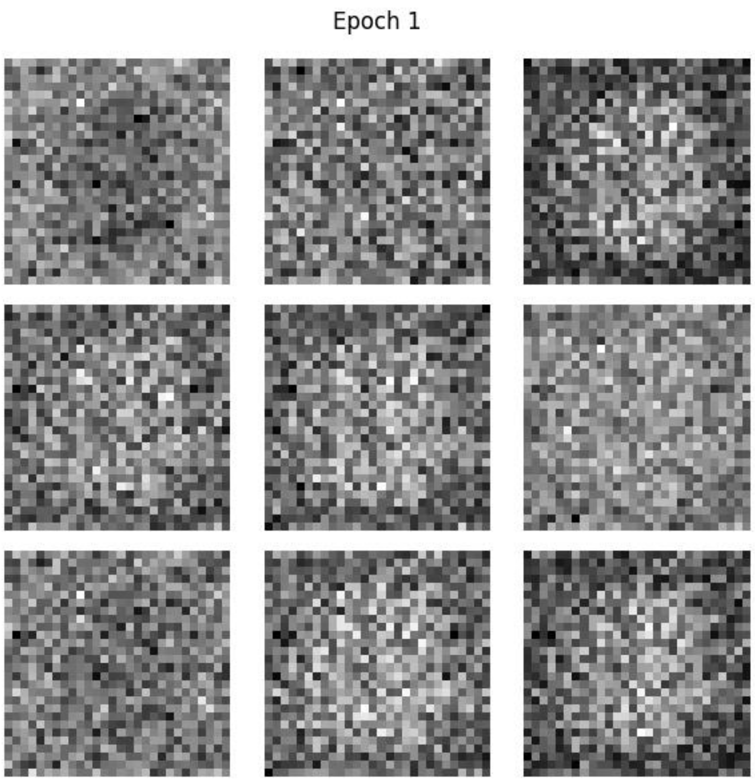
(Diffusion transformer model)

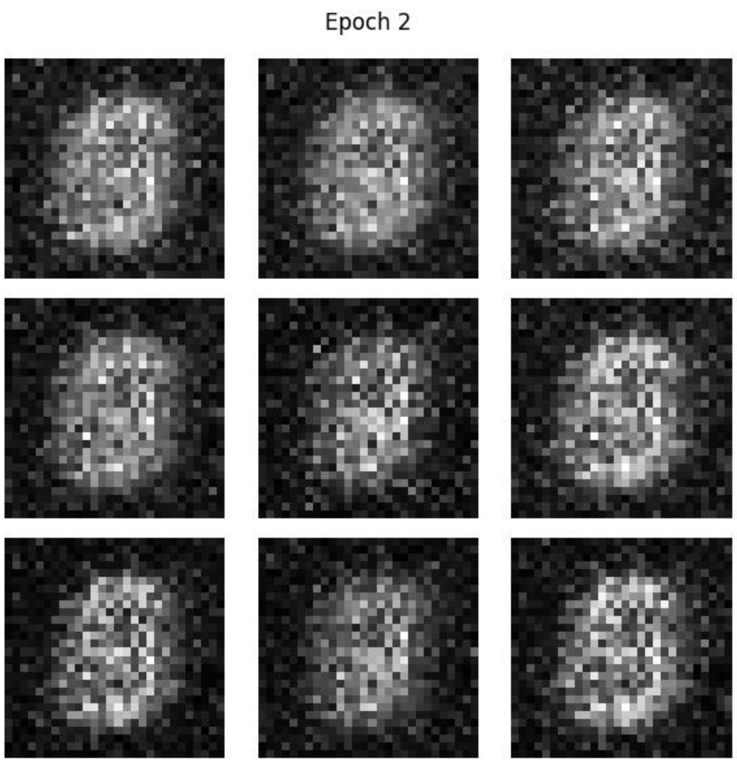
GAN for MNIST dataset



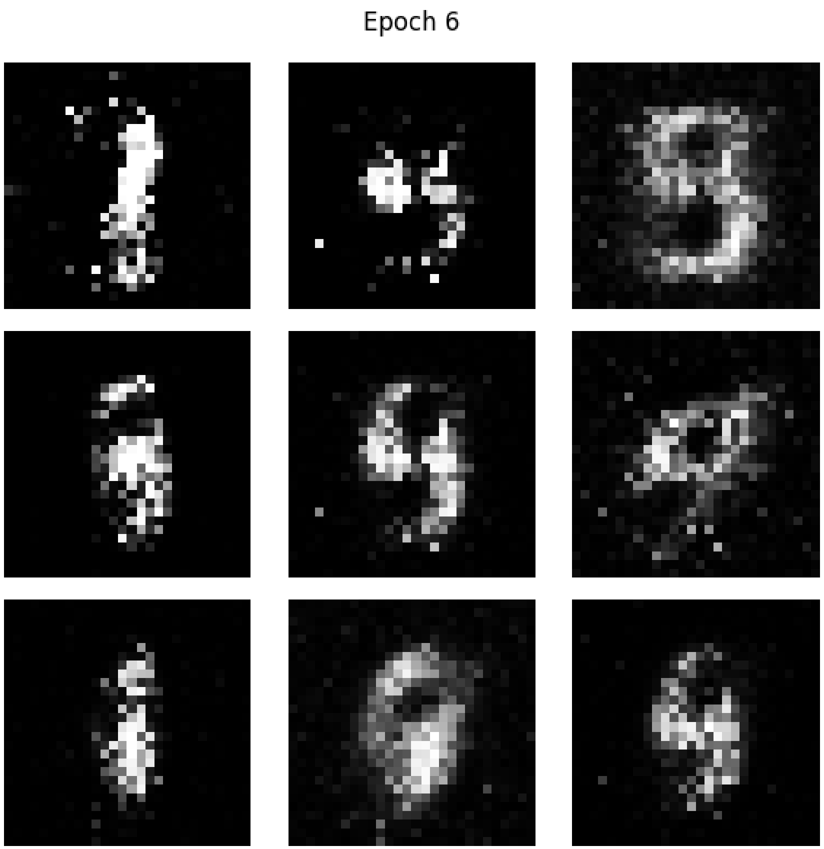
GAN for MNIST dataset

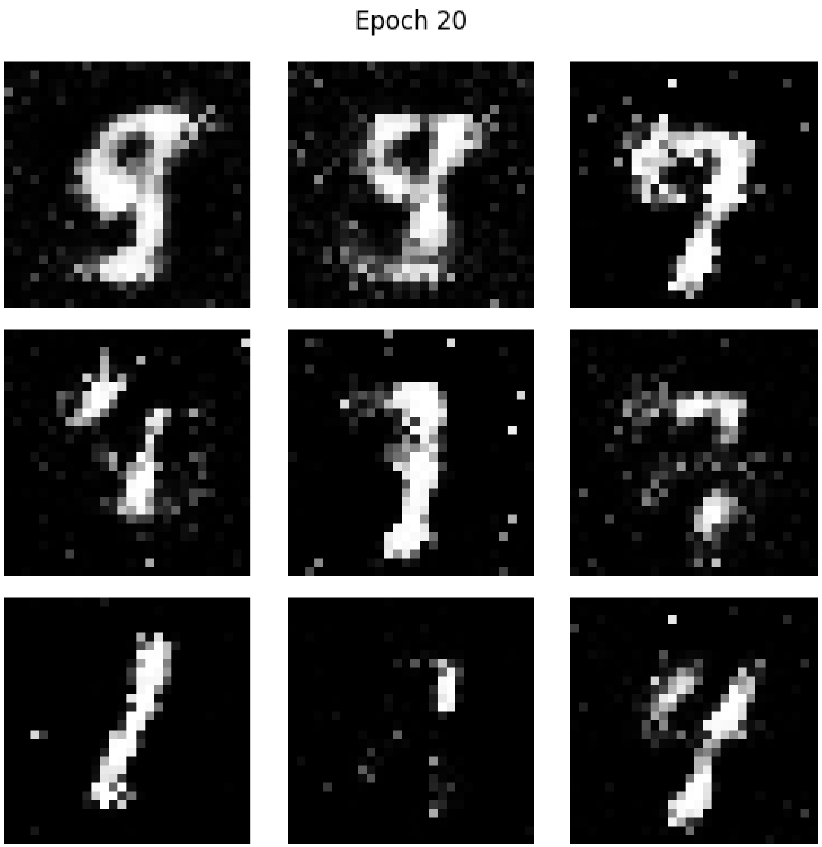
- Total possible 28x28 grayscale images: 256^784 ≈ 10^1888 possibilities
- Actual MNIST digits: maybe 10^6 distinct styles
- The GAN learns this tiny subspace within the huge space



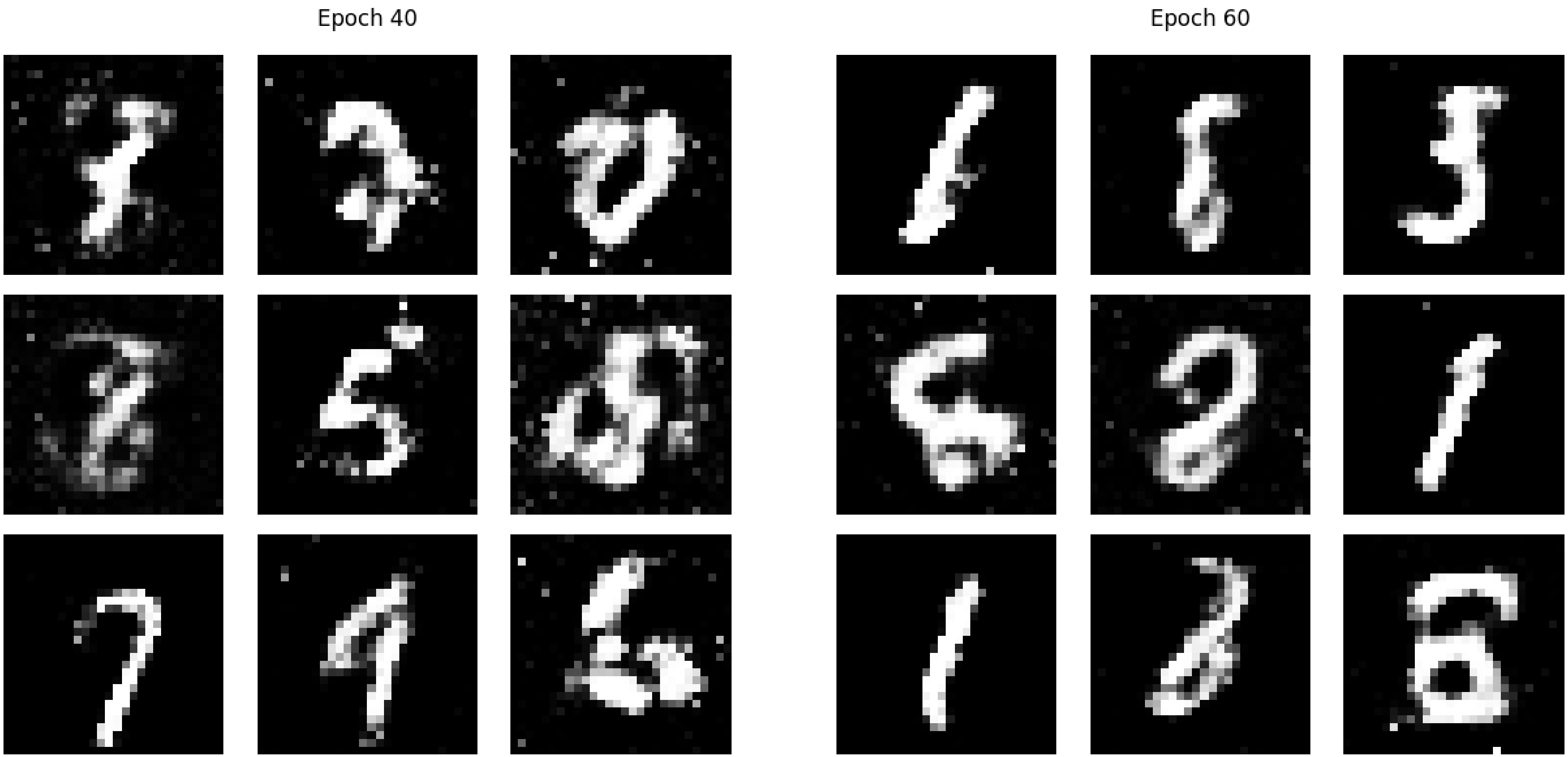


GAN for MNIST dataset

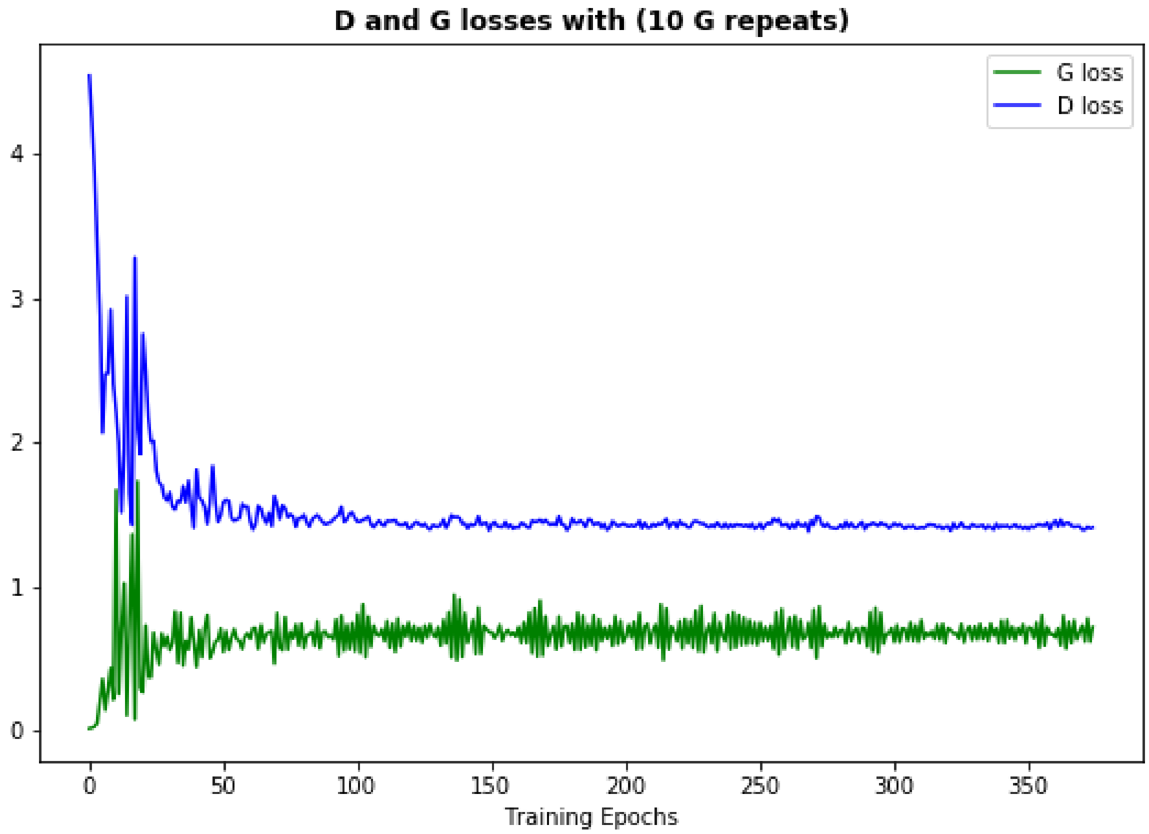




GAN for MNIST dataset



Pattern of losses



Unlike typical models where loss should just decrease, here the losses reflect an ongoing adversarial game.

Mode Collapse

Mode collapse = Generator produces limited variety

Mode collapse occurs when a GAN's generator learns to produce only a **limited subset** of the real data distribution, instead of the full variety.

For MNIST with 10 digits (0-9), mode collapse means:

•**Good training**: Generator produces all 10 digits

•**Mode collapse**: Generator produces only, say, digit "1" or "7"

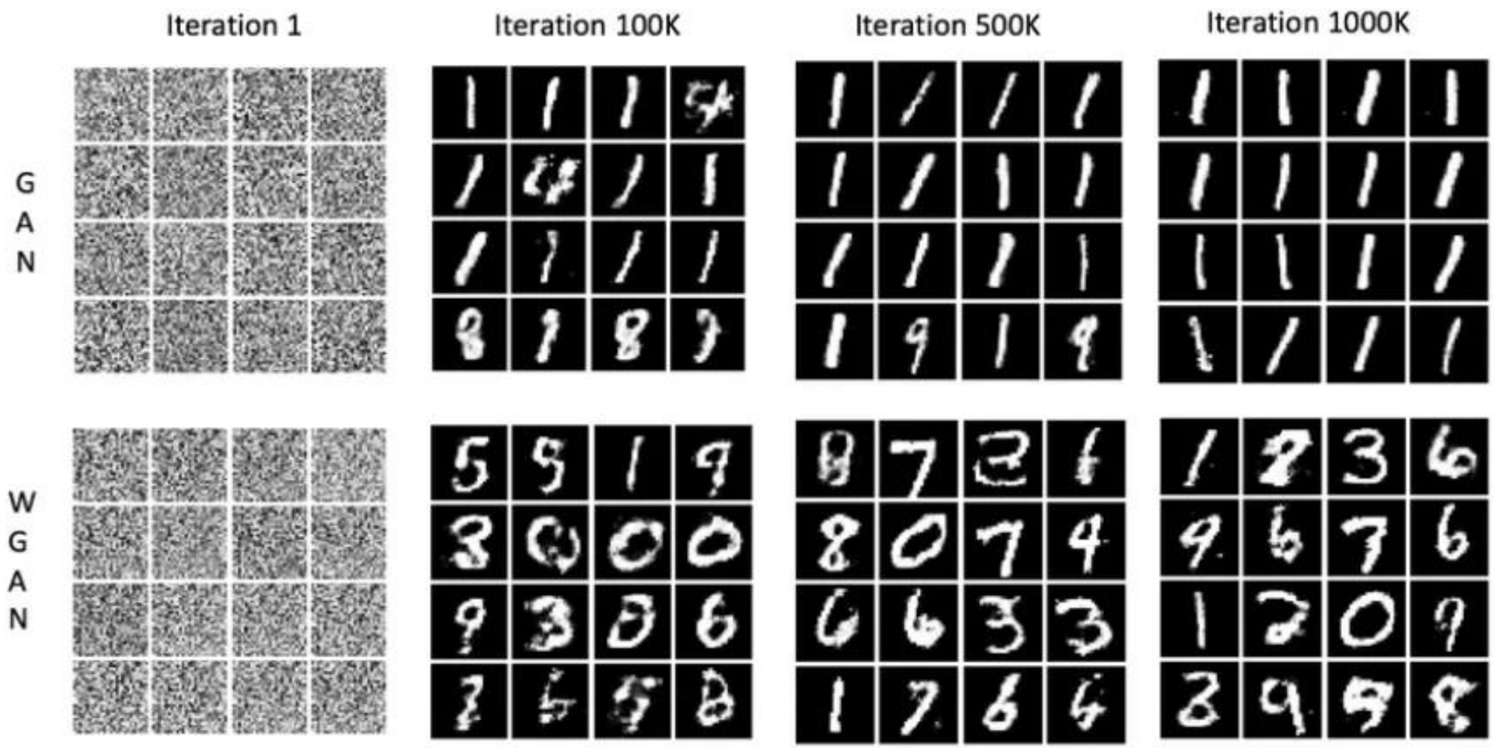
Mode Collapse

Real-world analogy:

A student who discovers they can get an "A" by memorizing one perfect essay and submitting it for every assignment, rather than learning to write different essays. The teacher (discriminator) keeps giving good grades for that one essay, so the student has no incentive to learn anything else!

WGAN

WGAN (Wasserstein Generative Adversarial Network) is a stabilized variant of GANs that replaces the traditional Jensen-Shannon divergence loss with the Wasserstein distance (Earth Mover's distance) to measure the dissimilarity between real and generated data distributions.



Why GAN are hard to train ?

- Generator keeps generating similar images (so nothing to learn)
- Trade-off of generating **more accurate** vs **high coverage** samples
- The two learning tasks need to have balance to achieve stability
- If Discriminator is not sufficiently trained, it can worse generator
- If Discriminator is over-trained, will produce no gradients


Questions