INF721

2024/2

Deep Learning

L21: Generative Adversarial Network

Logistics

Last Lecture

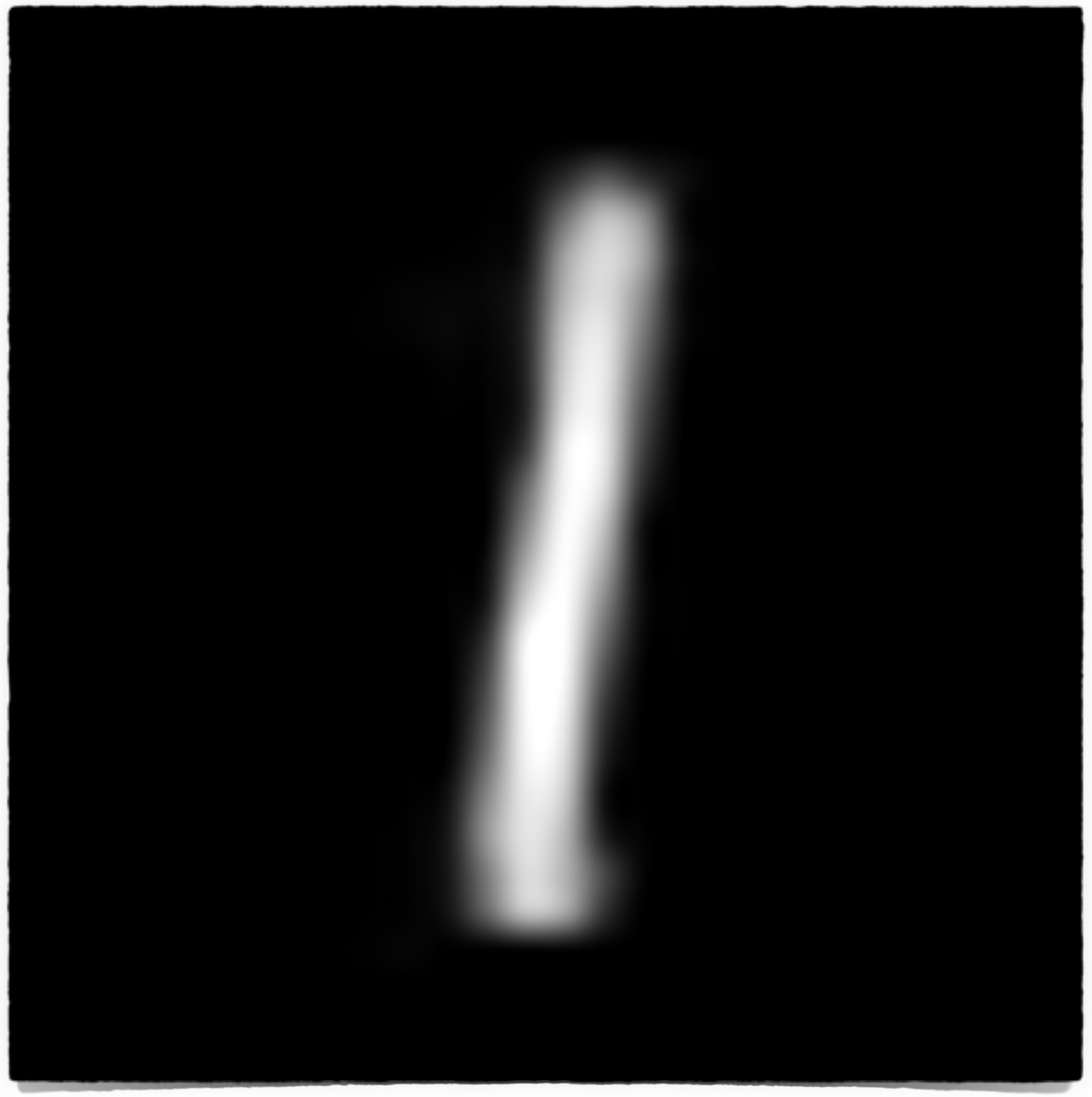
- Generative Models
- Autoencoders
  - Denoising Aucoders
  - Masked Autoencoders
- Variational Autoencoders
  - KL-Divergence

UFV

Lecture Outline

- Generative Adversarial Networks
- Problems with VAEs
- Generator as a CNN
- Discriminator as a CNN
- Adversarial Training
- GANs over the years
- Results

Problems with Variational Autoencoders



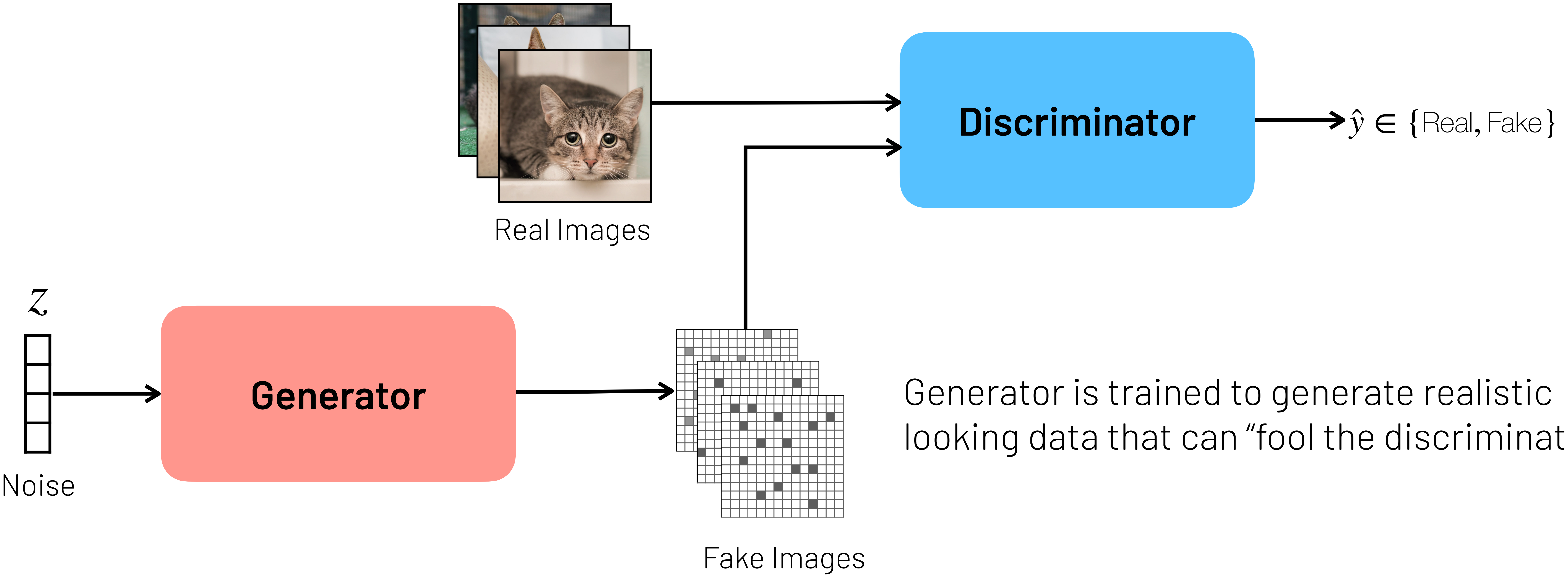
In VAEs, we model an explicit probability distribution $N(\mu, \sigma)$ and evaluate it with MSE + KLD:

- The MSE + KLD loss tend to produce blurry images
- We do not explicitly evaluate in our loss "the quality" of the generated samples.
- What makes an image a good image?

UFV

Generative Adversarial Networks (GANs)

GANs use a separated network, called a Discriminator, to evaluate the quality of images produced by the Generator Network:

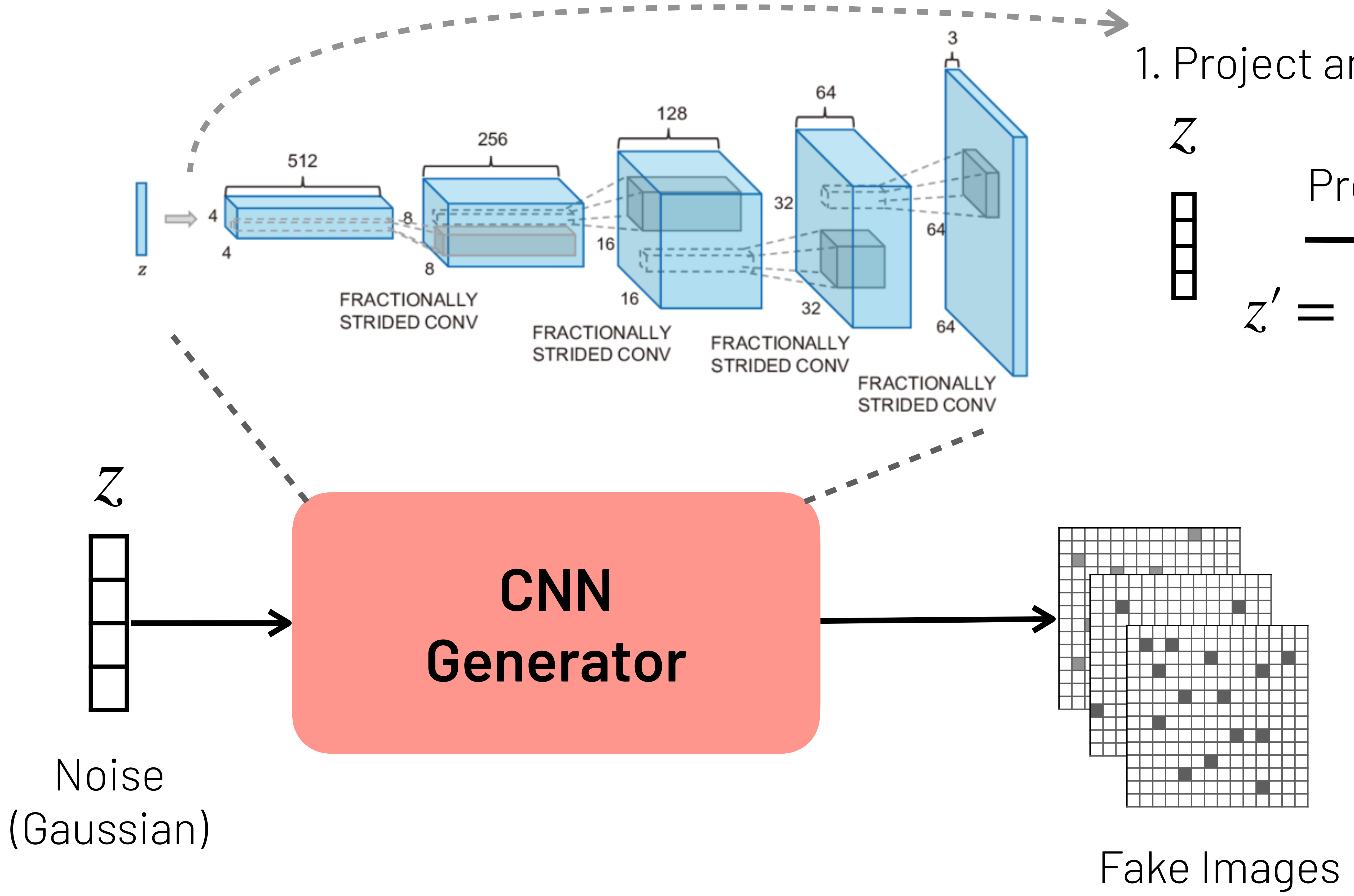


Generator is trained to generate realistic looking data that can "fool the discriminator"

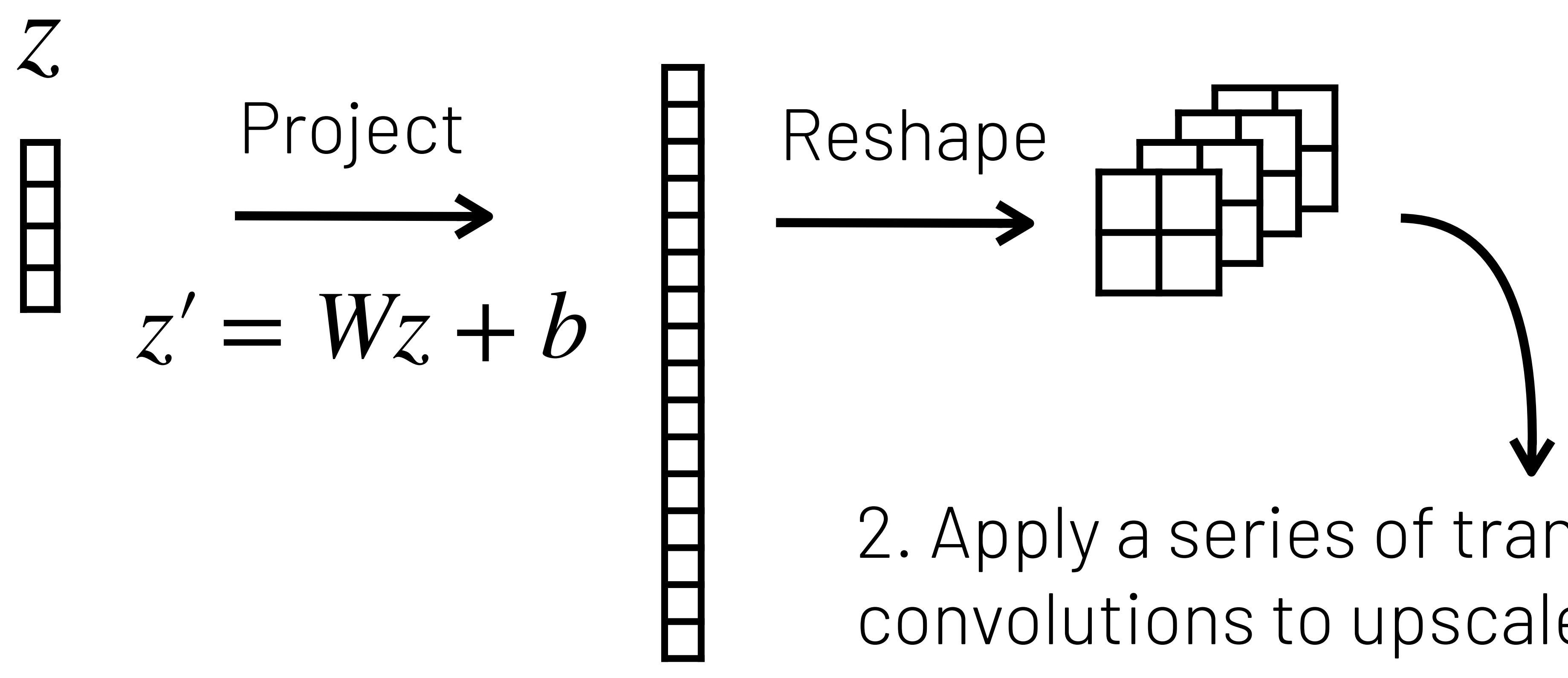
Goodfellow et al. 2014, Generative Adversarial Networks

A Typical Generator (DCGAN)

The Generator takes a vector of noise z and outputs an image. This is typically implemented as a Convolutional Neural Network with Transposed Convolutions and ReLUs:



1. Project and reshape the z vector

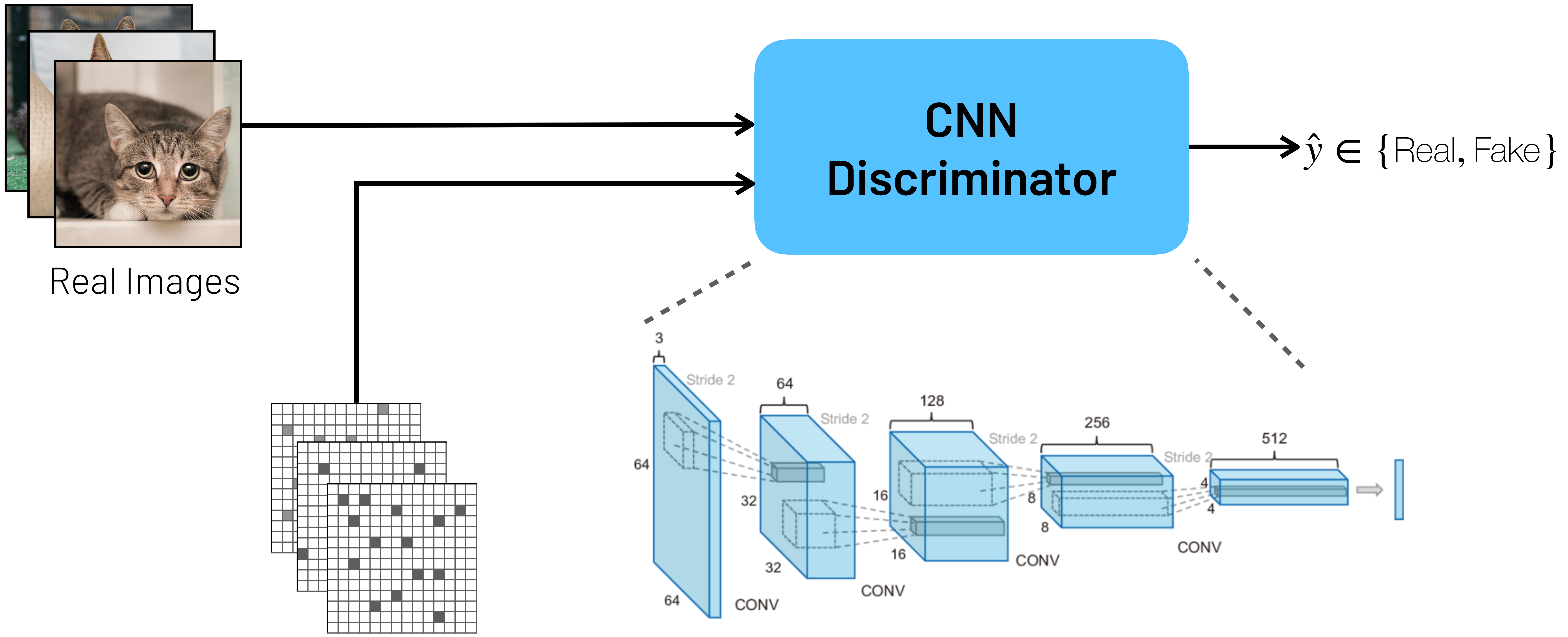


$$z' = Wz + b$$

2. Apply a series of transposed convolutions to upscale the feature map up to $w * h * 3$

A Typical Discriminator (DCGAN)

The Discriminator takes as input x af k real images from the dataset and k fake images produced by the generator, and produces an output $\hat{y} \in \{\text{Real, Fake}\}$ (binary classification)

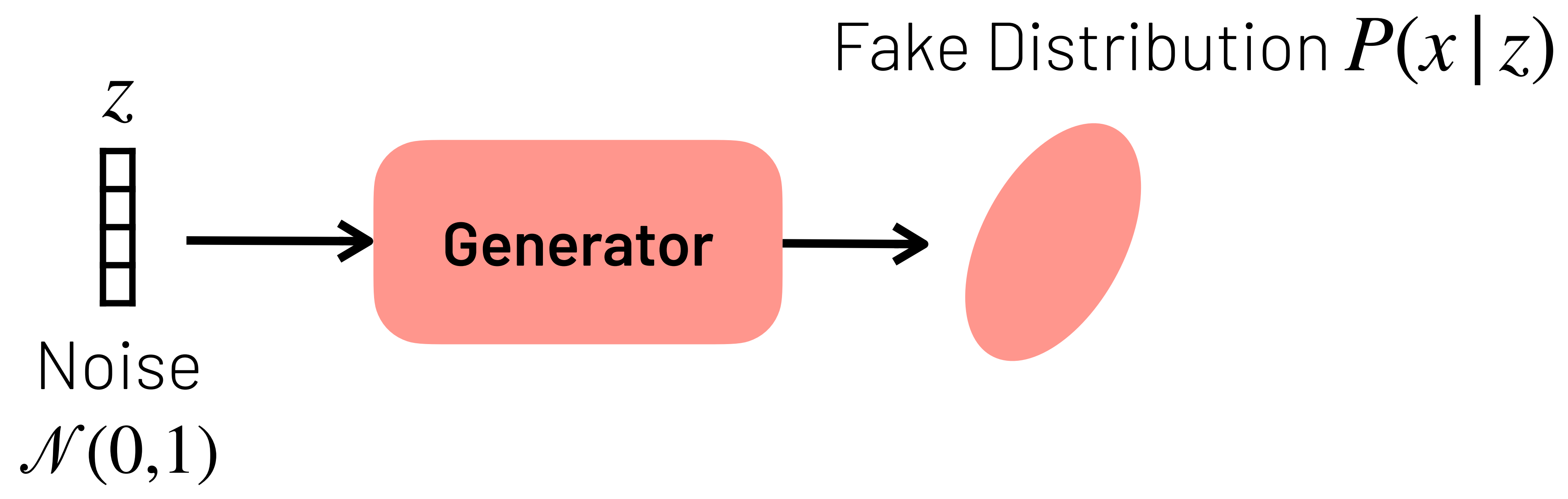


Fake Images

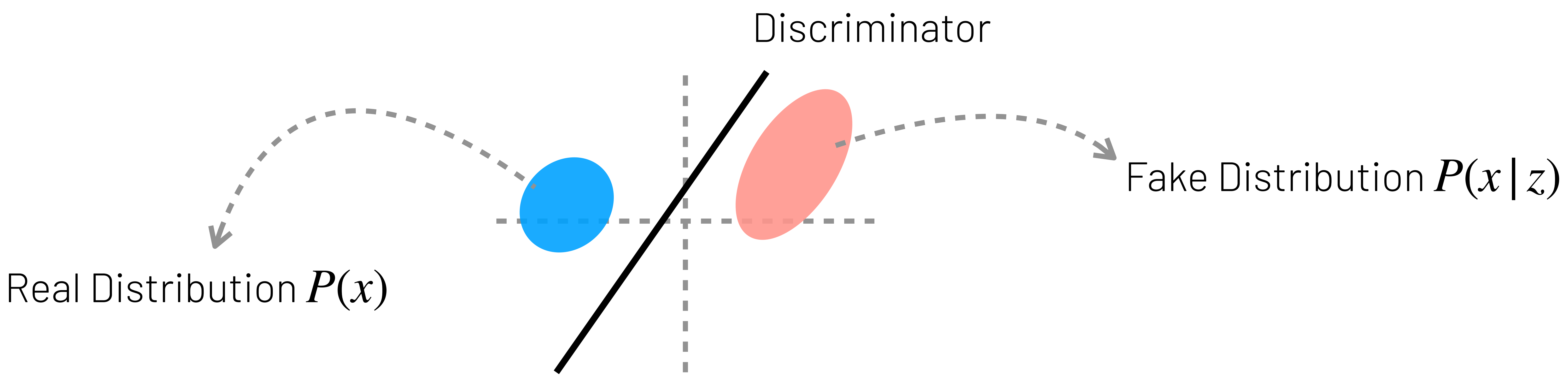
The Discriminator typically has a symmetric architecture to the generator (with LeakyReLUs instead of ReLUs)

Geometric Intuition

The generator is implicitly modeling a conditional distribution $P(x \mid z)$, assuming the existence of latent features z that represent the real data:



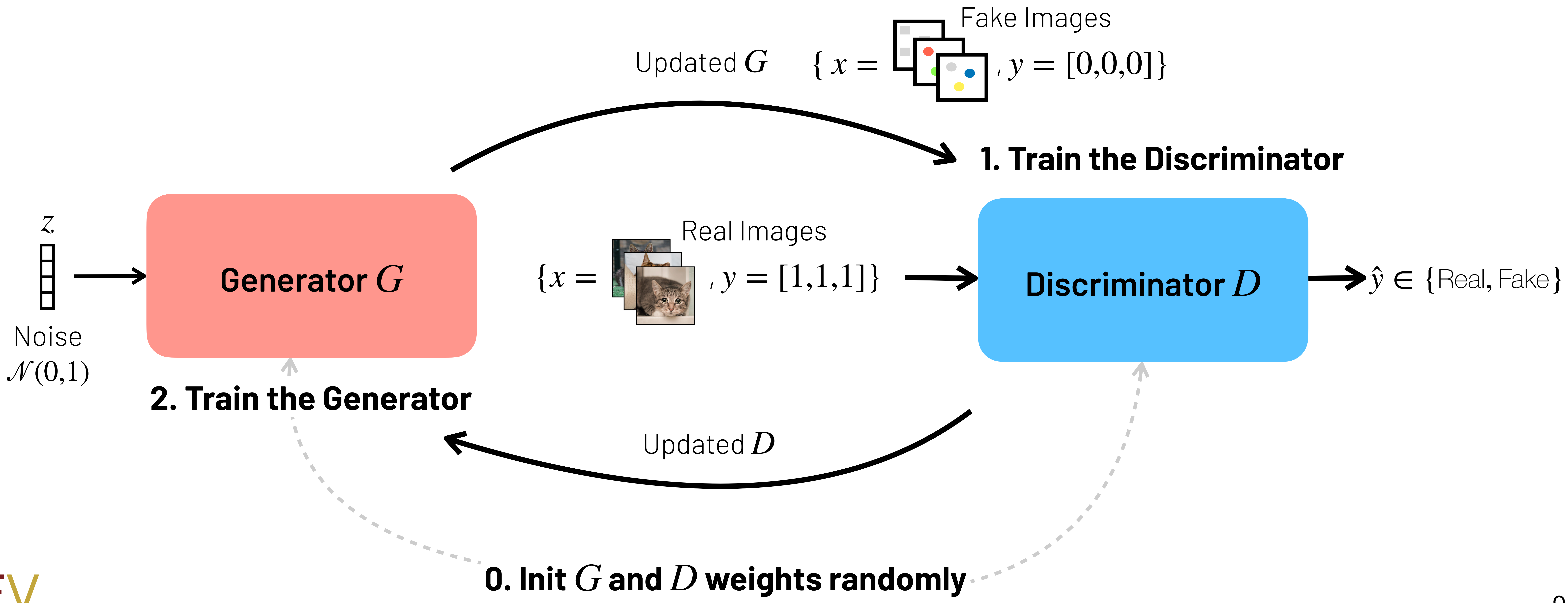
The Discriminator is trying to separate the the fake distribution from the real distribution:



UFV

The Adversarial Training

A GAN is trained as a zero-sum game where both networks compete against each other, and the only way for the generator to “win” is by generating realistic data:

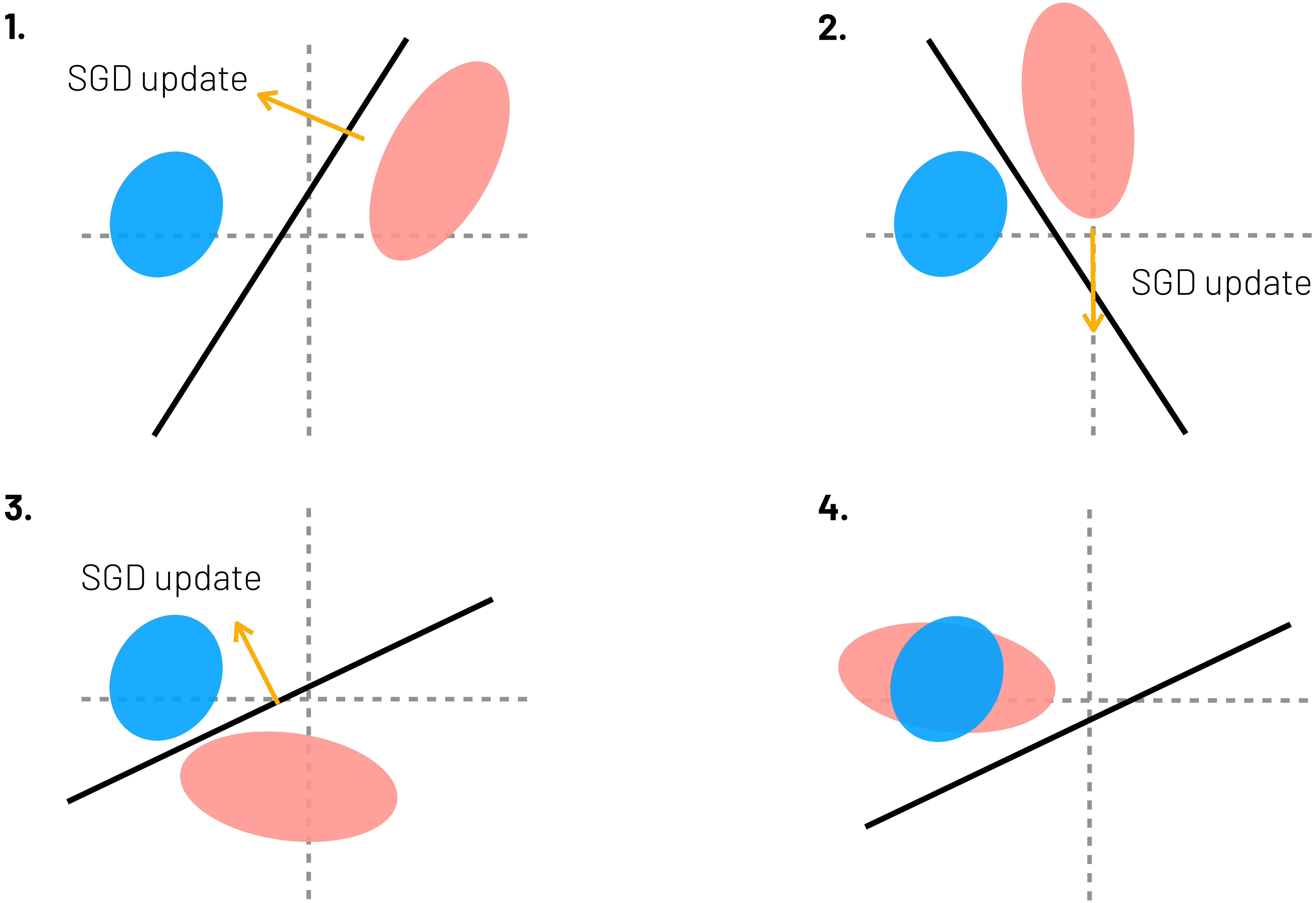


The Adversarial Training

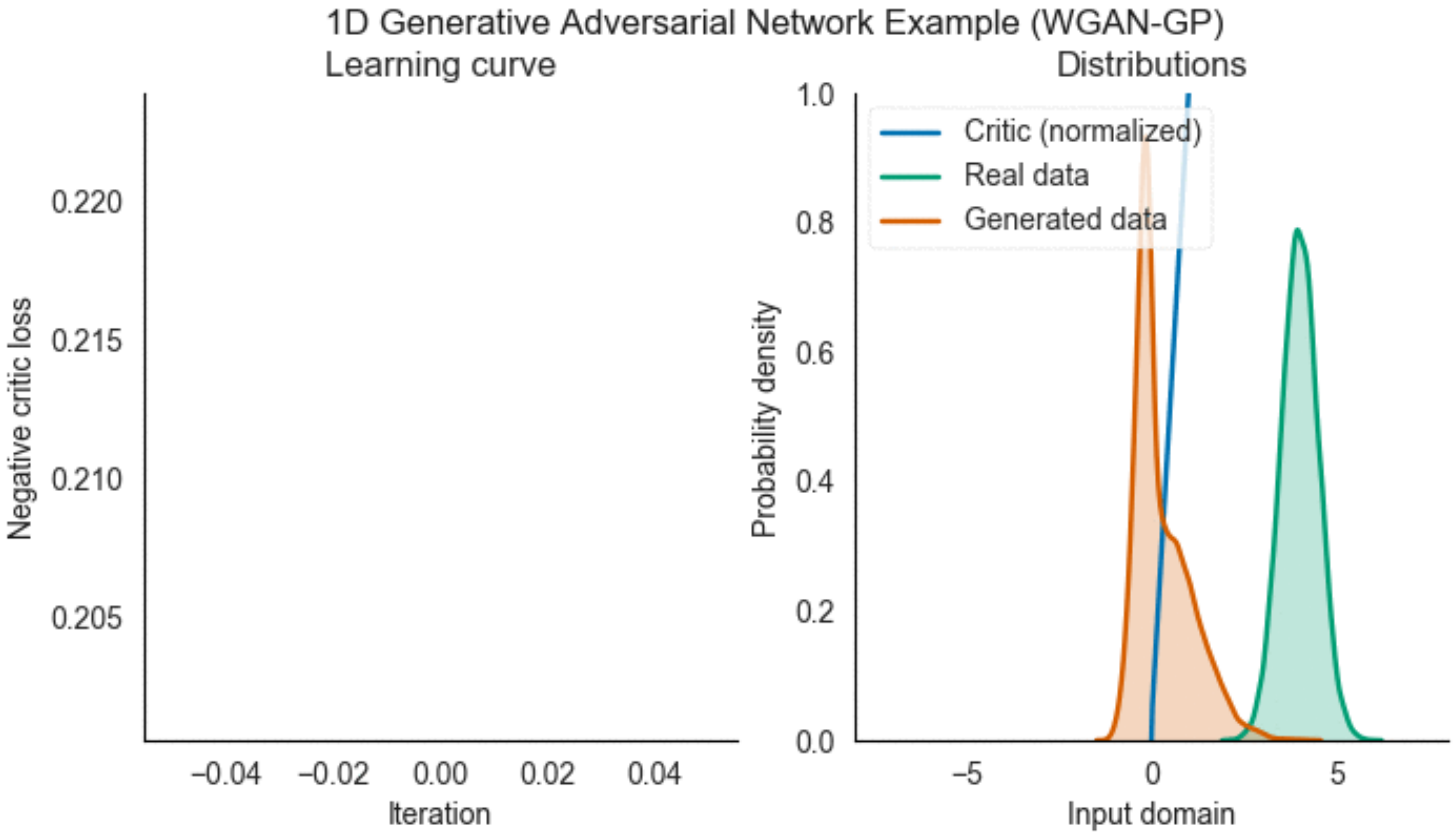
A GAN is trained as a zero-sum game where both networks compete against each other, and the only way for the generator to “win” is by generating realistic data:

```
for each epoch:
    for each mini batch:
        for i from 1 to k:   # Train Discriminator k times
            Take m real examples from the training dataset
            Generate m fake examples using the generator
            Update ONLY D weights by maximizing log(D(x)) + log(1 - D(G(z)))
        Generate m new fake examples
        Update ONLY the G weights by maximing log(D(G(z)))
```

Geometric Intuition: Training



Geometric Intuition: Training



Results: DCGAN (2015)

Real Images

Fake Images

The Evolution of GANs

Year	Model	Key Inovations
2014	**GAN**	Original adversarial framework
2015	**DCGAN**	Stable CNN architecture
2016	**Conditional GAN**	Controlled generation
2017	**WGAN**	Improved training stability
2017	**Progressive GAN**	High resolution images
2018	**StyleGAN**	Style-based generation
2019	**BigGAN**	Large scale training
2020	**StyleGAN2**	Image quality improvements
2021	**StyleGAN3**	Motion coherence

Results over the years

2014 2015 2016 2017

2018 2019 2020 2021

UFV

Next Lecture

Project Presentations!