

Generative Modelling

Rina BUOY



AMERICAN UNIVERSITY
OF PHNOM PENH

STUDY LOCALLY. LIVE GLOBALLY.

Which one is real face ?



A



B



C

Supervised vs. Unsupervised

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn function to map
 $x \rightarrow y$

Examples: Classification, regression, object detection, semantic segmentation, etc.

Unsupervised Learning

Data: x

x is data, no labels!

Goal: Learn some *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

Supervised vs. Unsupervised

Supervised Learning

Data: (x, y)

x is data, y is label

Goal: Learn function to map
 $x \rightarrow y$

Examples: Classification, regression, object detection, semantic segmentation, etc.

Unsupervised Learning

Data: x

x is data, no labels!

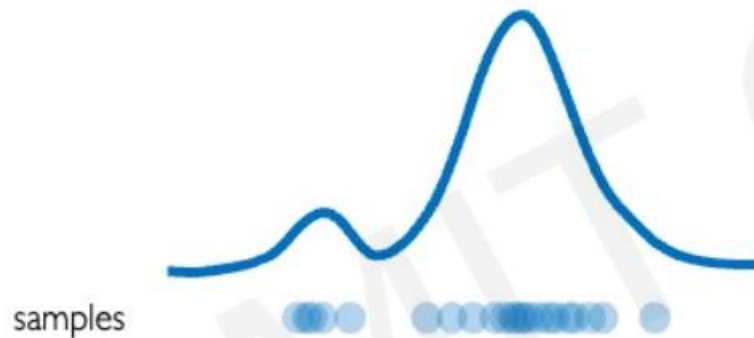
Goal: Learn the *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

Generative Modelling

Goal: Take as input training samples from some distribution and learn a model that represents that distribution

Density Estimation



Sample Generation



Input samples

Training data $\sim P_{data}(x)$



Generated samples

Generated $\sim P_{model}(x)$

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

Why Generative Modelling ?

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



Homogeneous skin color, pose

VS



Diverse skin color, pose, illumination

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



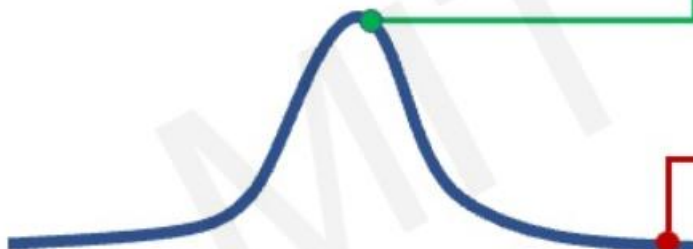
Why Generative Modelling ?

- **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



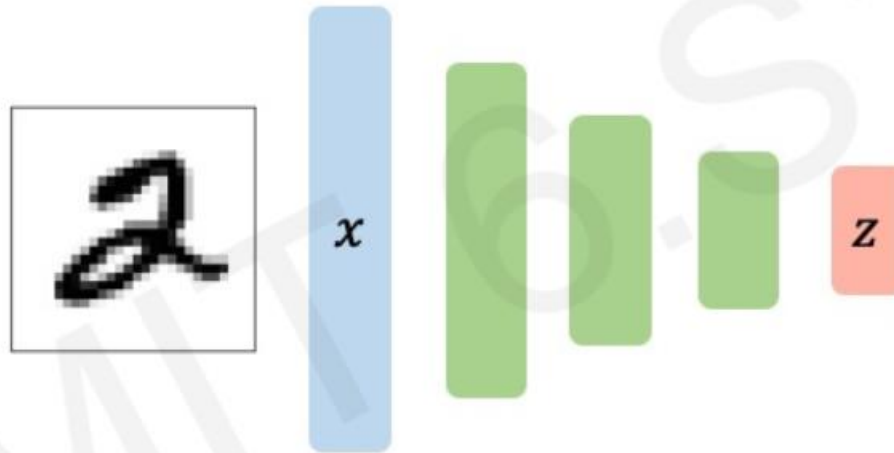
Harsh Weather



Pedestrians

Autoencoders

Unsupervised approach for learning a **lower-dimensional** feature representation from unlabeled training data

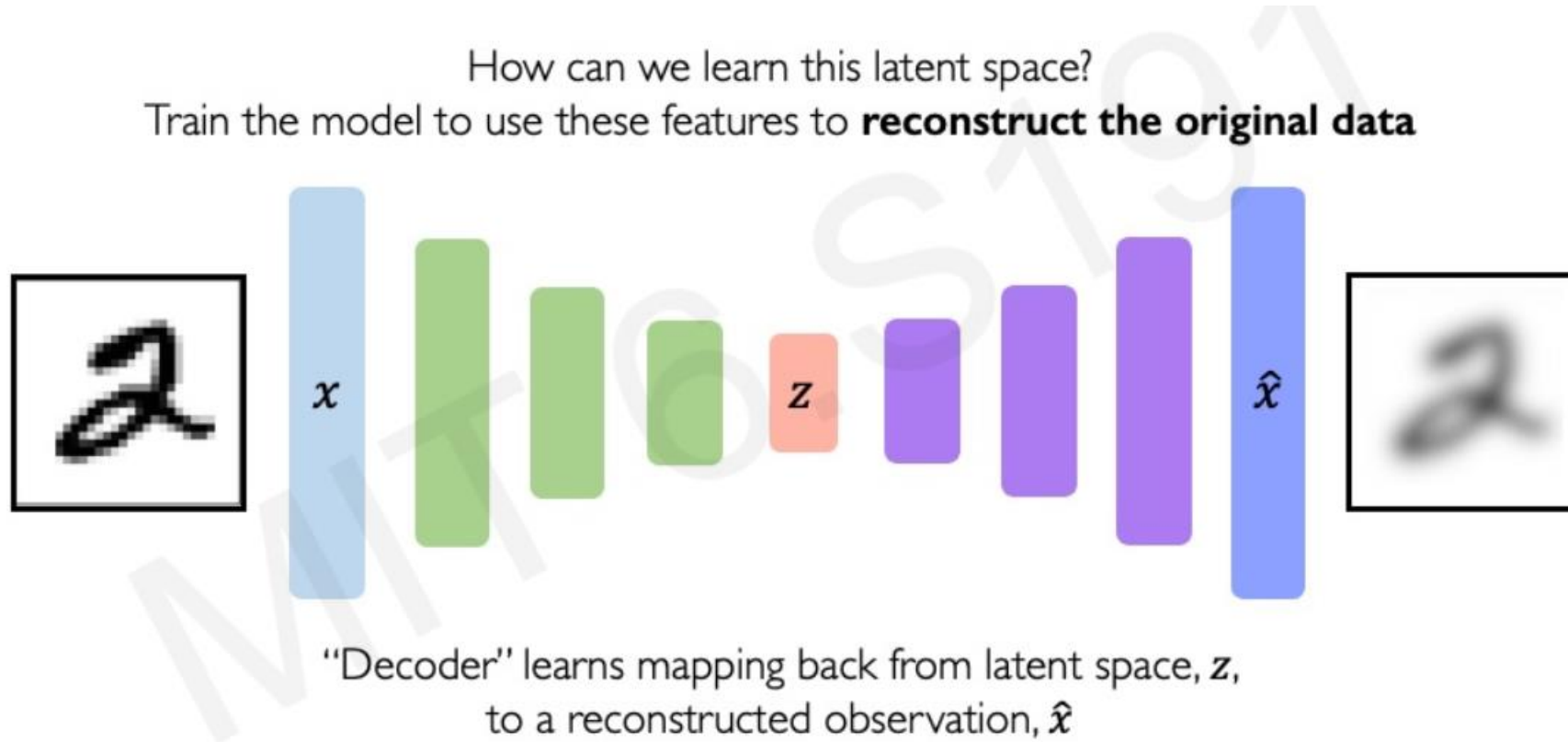


Why do we care about a low-dimensional z ?



“Encoder” learns mapping from the data, x , to a low-dimensional latent space, z

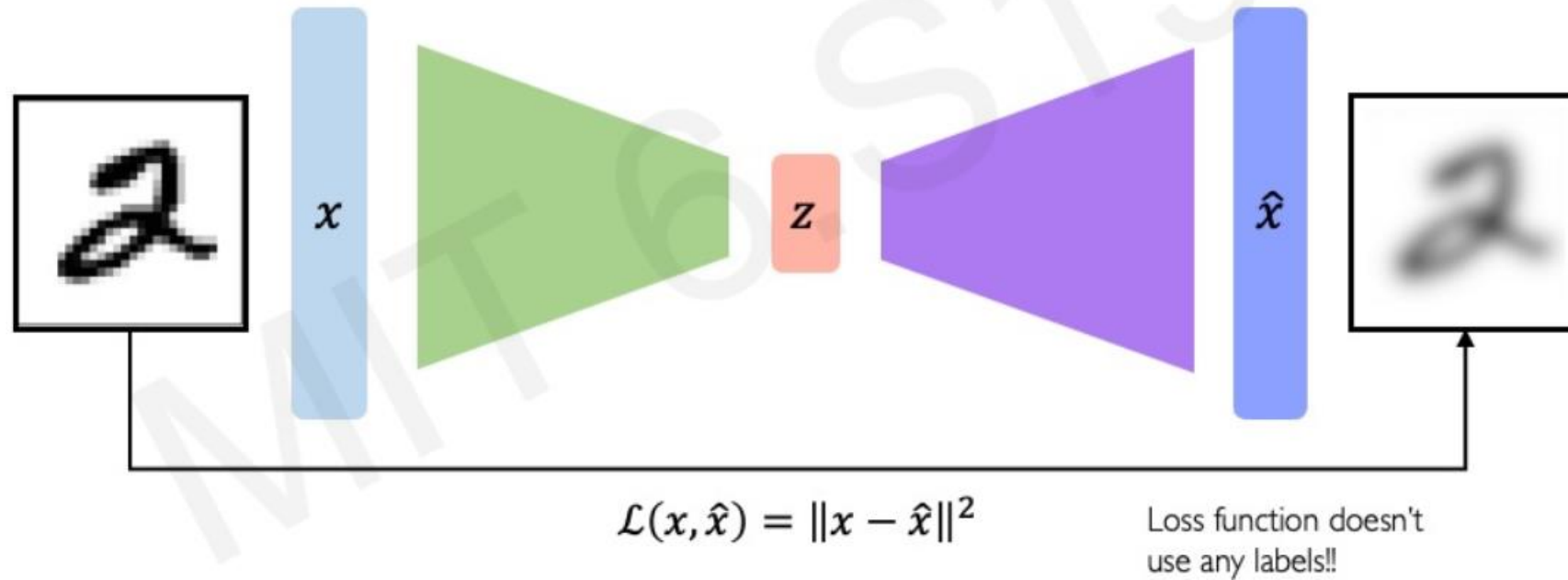
Autoencoders



Autoencoders

How can we learn this latent space?

Train the model to use these features to **reconstruct the original data**



Dimensionality of Latent Space

Autoencoding is a form of compression!
Smaller latent space will force a larger training bottleneck

2D latent space



5D latent space



Ground Truth



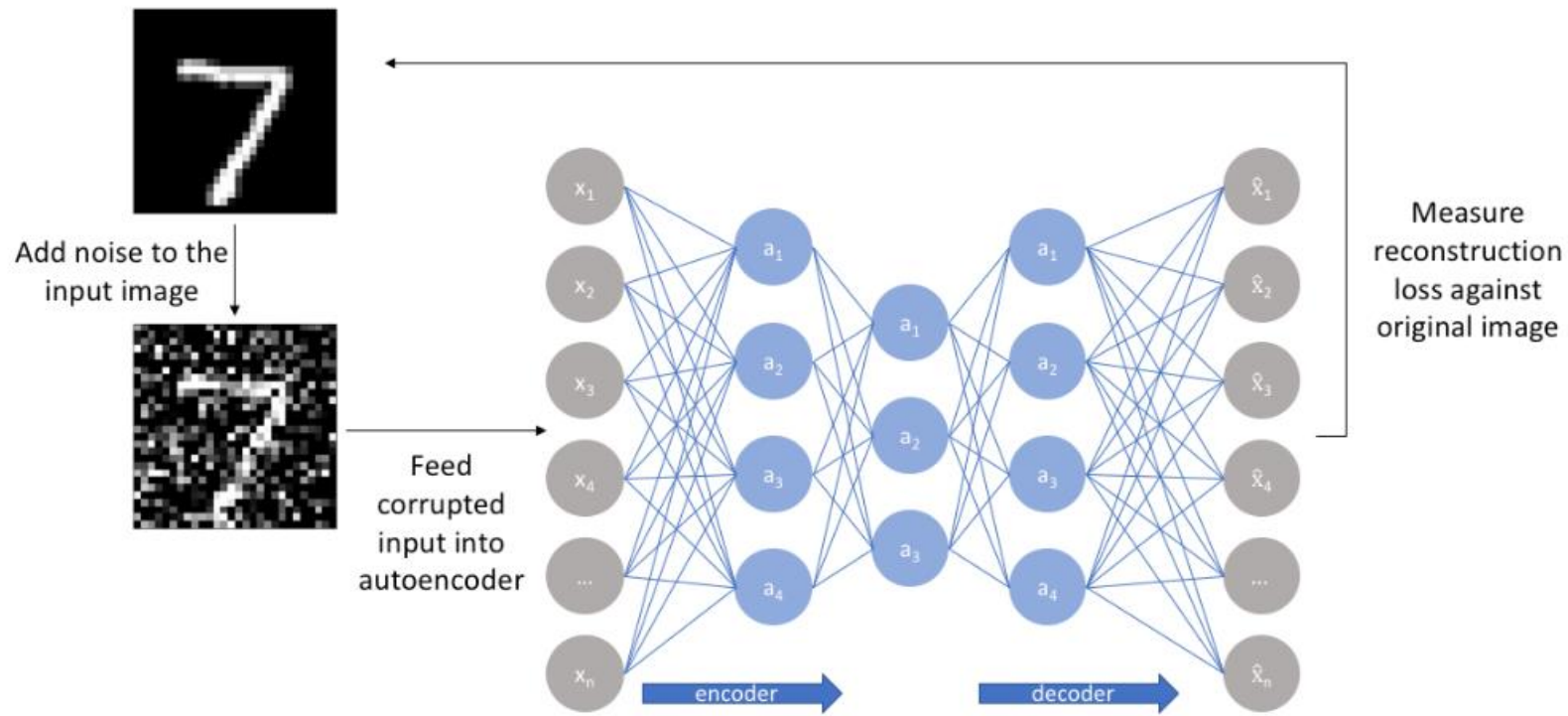
Autoencoders for Learning Representation

Bottleneck hidden layer forces network to learn a compressed latent representation

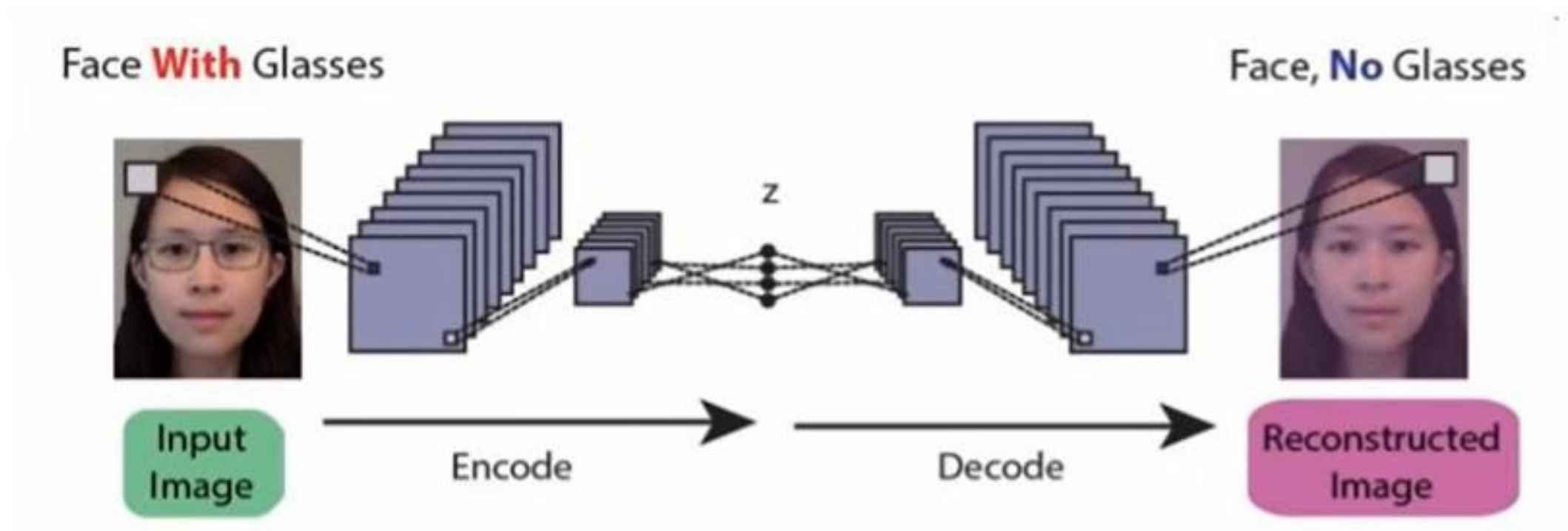
Reconstruction loss forces the latent representation to capture (or encode) as much “information” about the data as possible

Autoencoding = **Automatically encoding** data; “Auto” = **self**-encoding

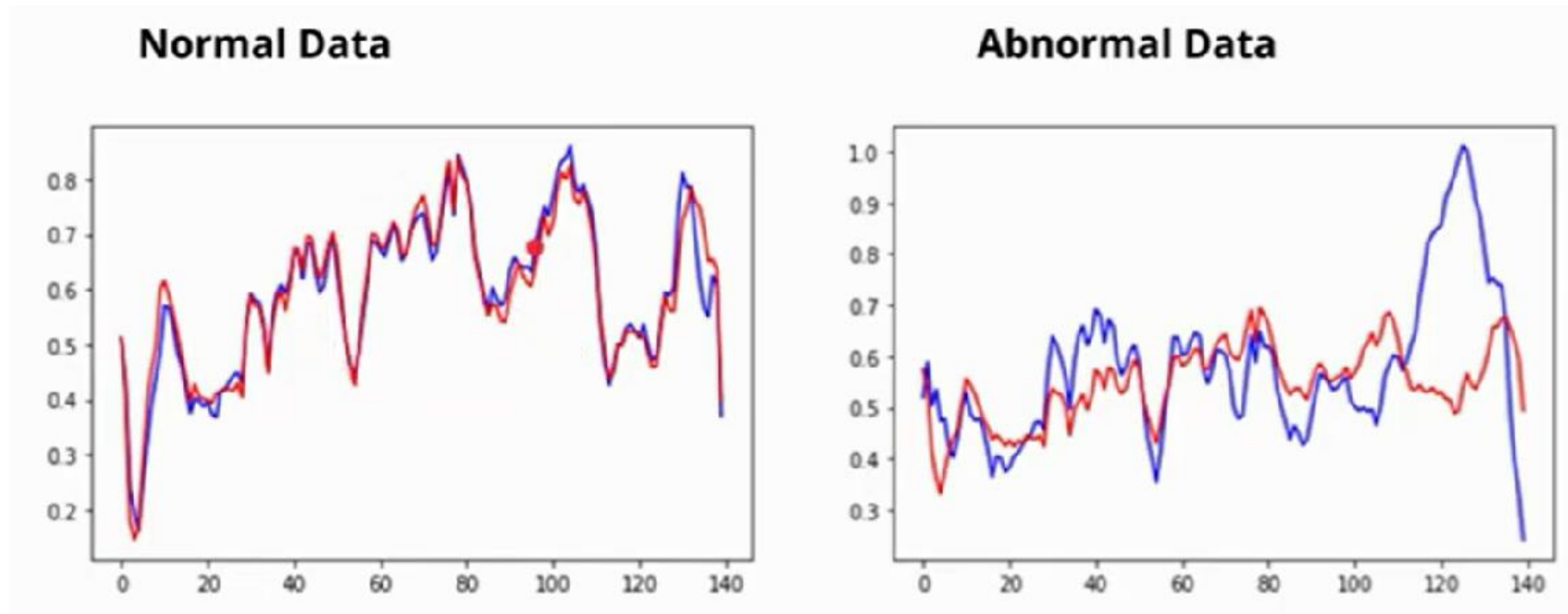
Examples – Denoising



Examples – Denoising



Examples – Outlier Detection



Examples – Image Colorization

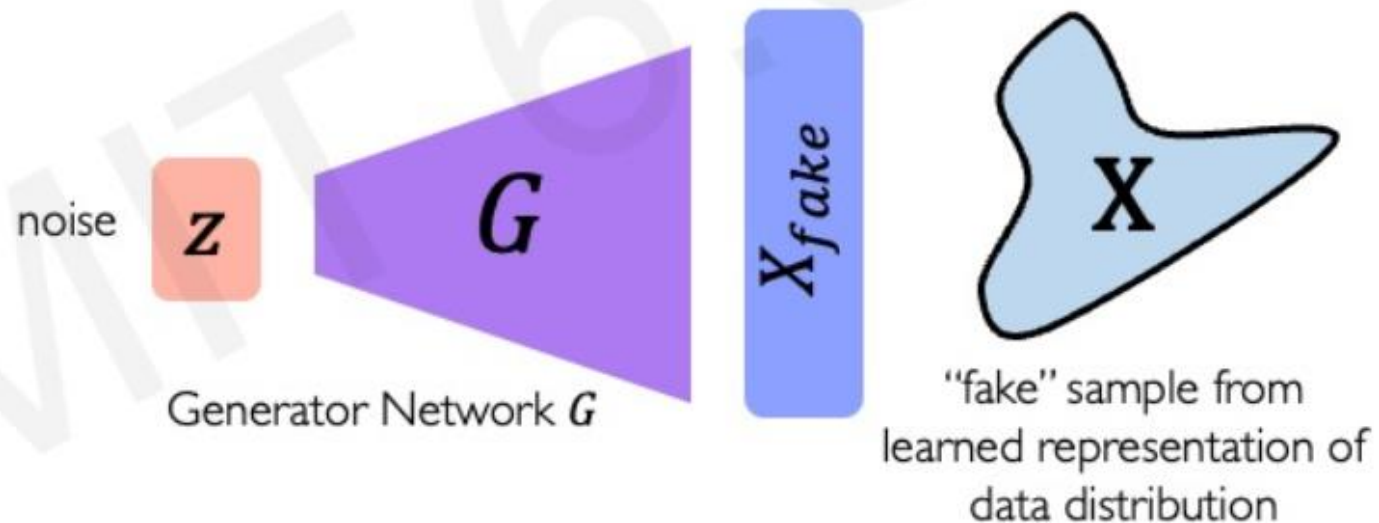


Generative Adversarial Networks

Idea: don't explicitly model density, and instead just sample to generate new instances.

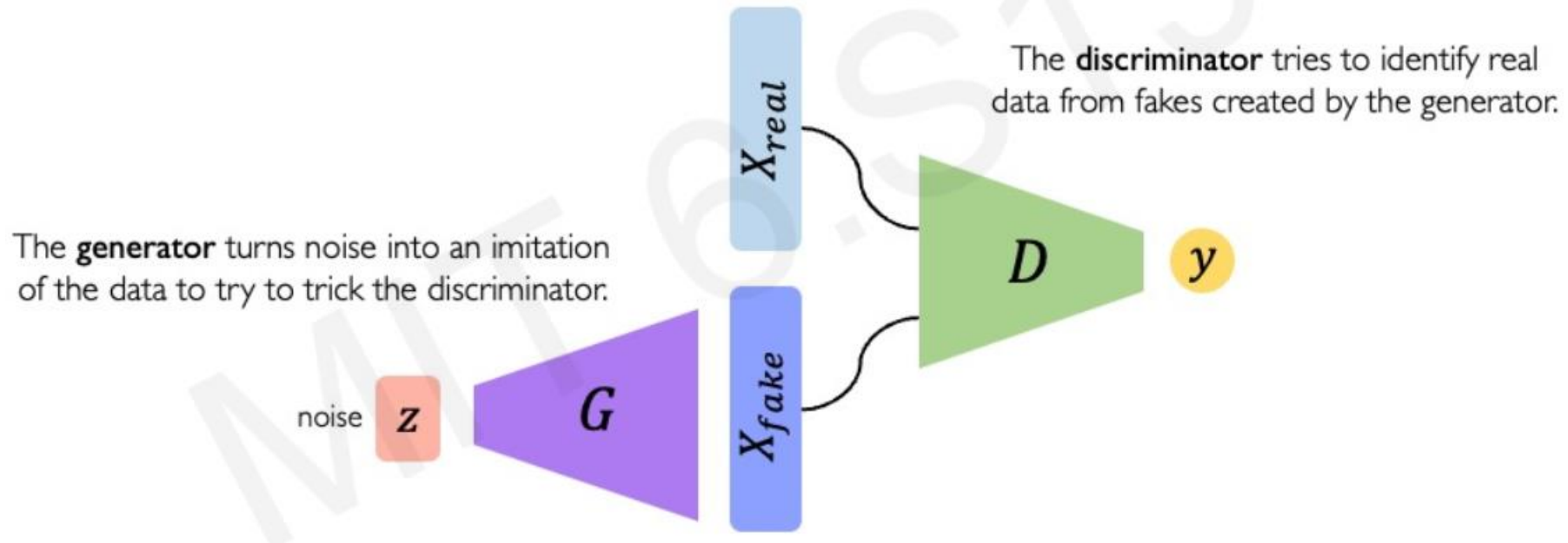
Problem: want to sample from complex distribution – can't do this directly!

Solution: sample from something simple (e.g., noise), learn a transformation to the data distribution.



Generative Adversarial Networks

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



Generative Adversarial Networks

Generator starts from noise to try to create an imitation of the data.

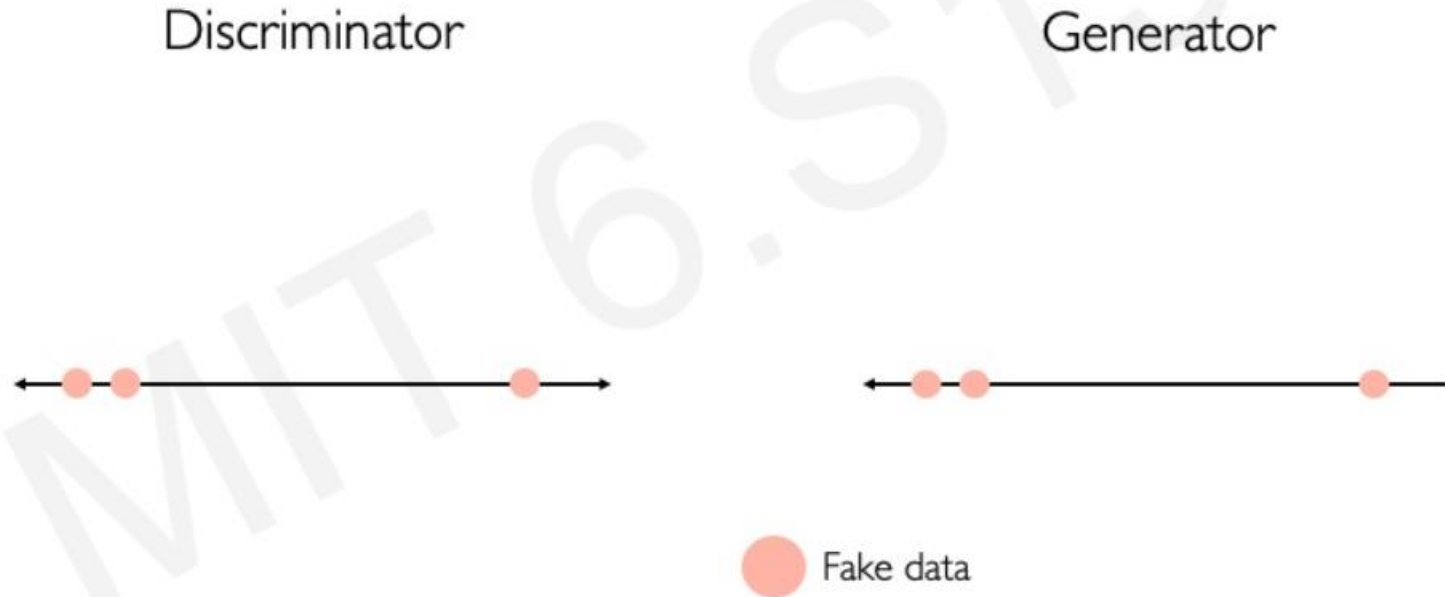
Generator



 Fake data

Generative Adversarial Networks

Discriminator looks at both real data and fake data created by the generator.



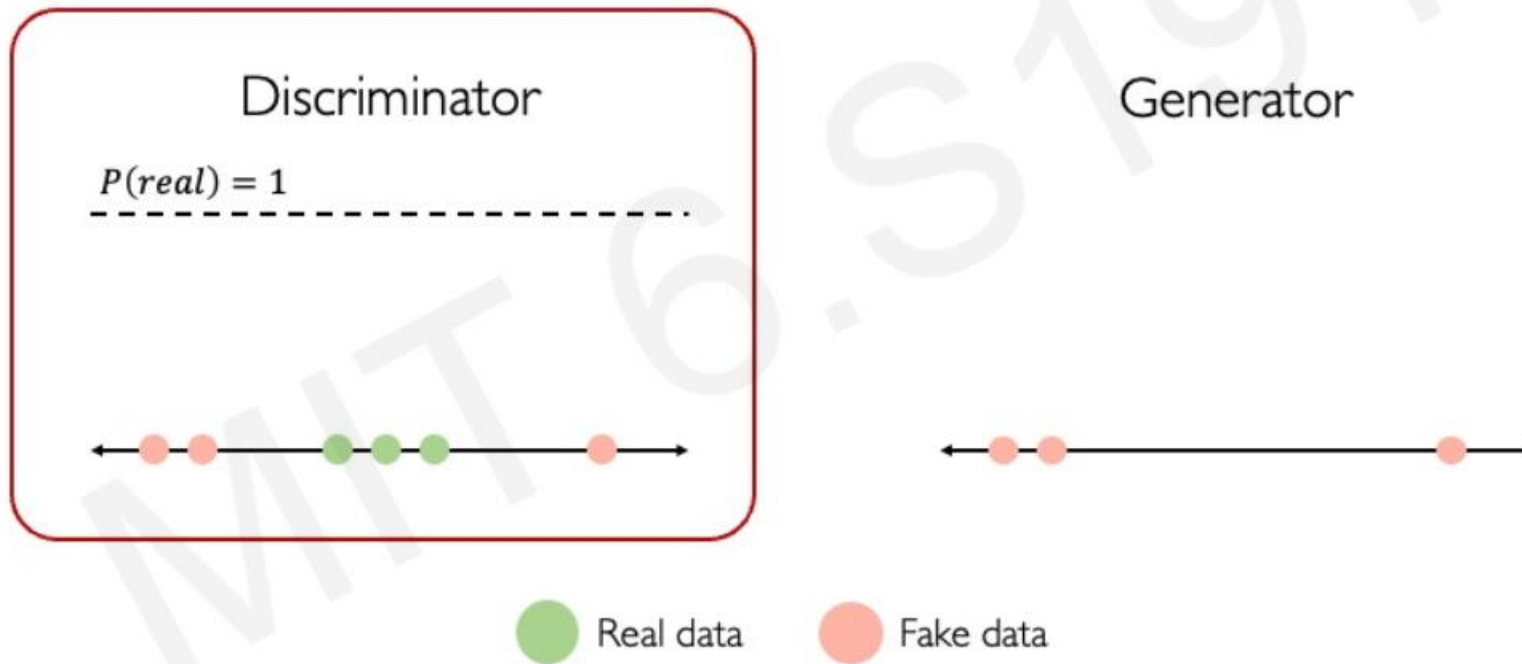
Generative Adversarial Networks

Discriminator looks at both real data and fake data created by the generator.



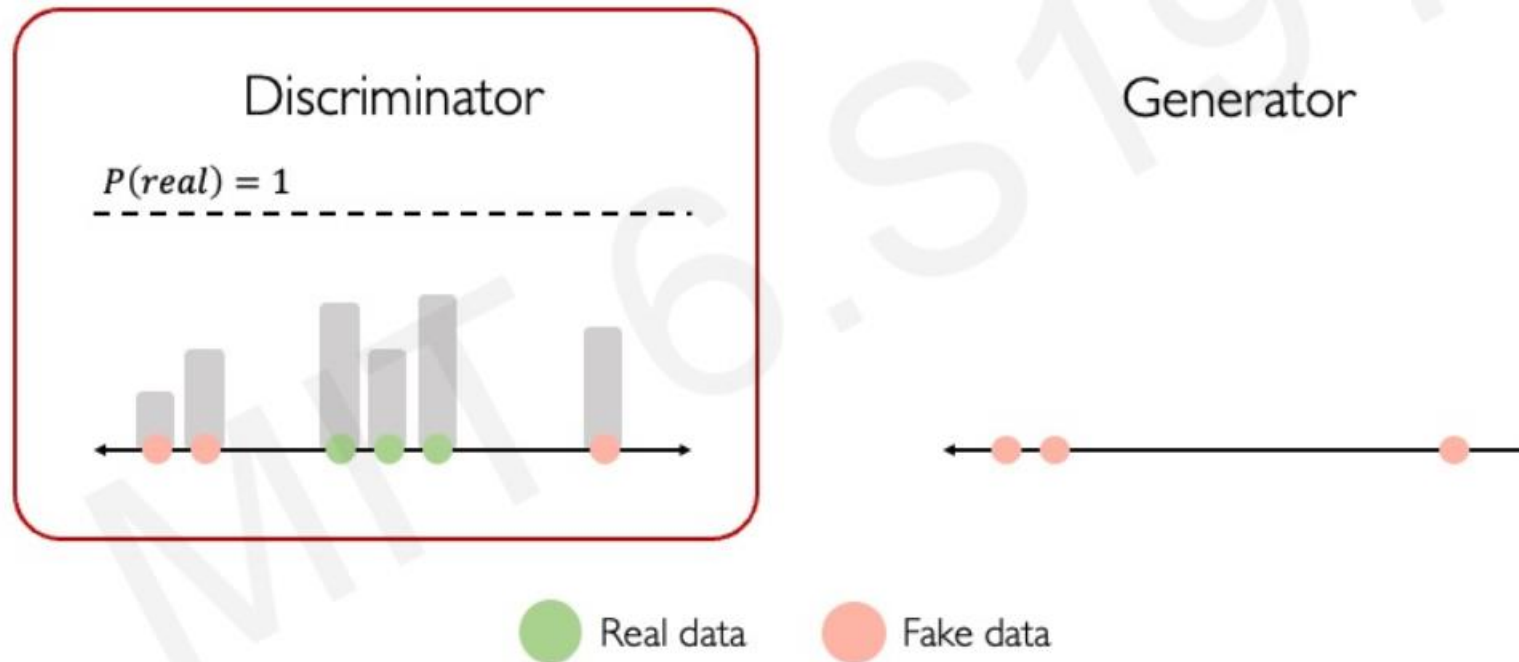
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



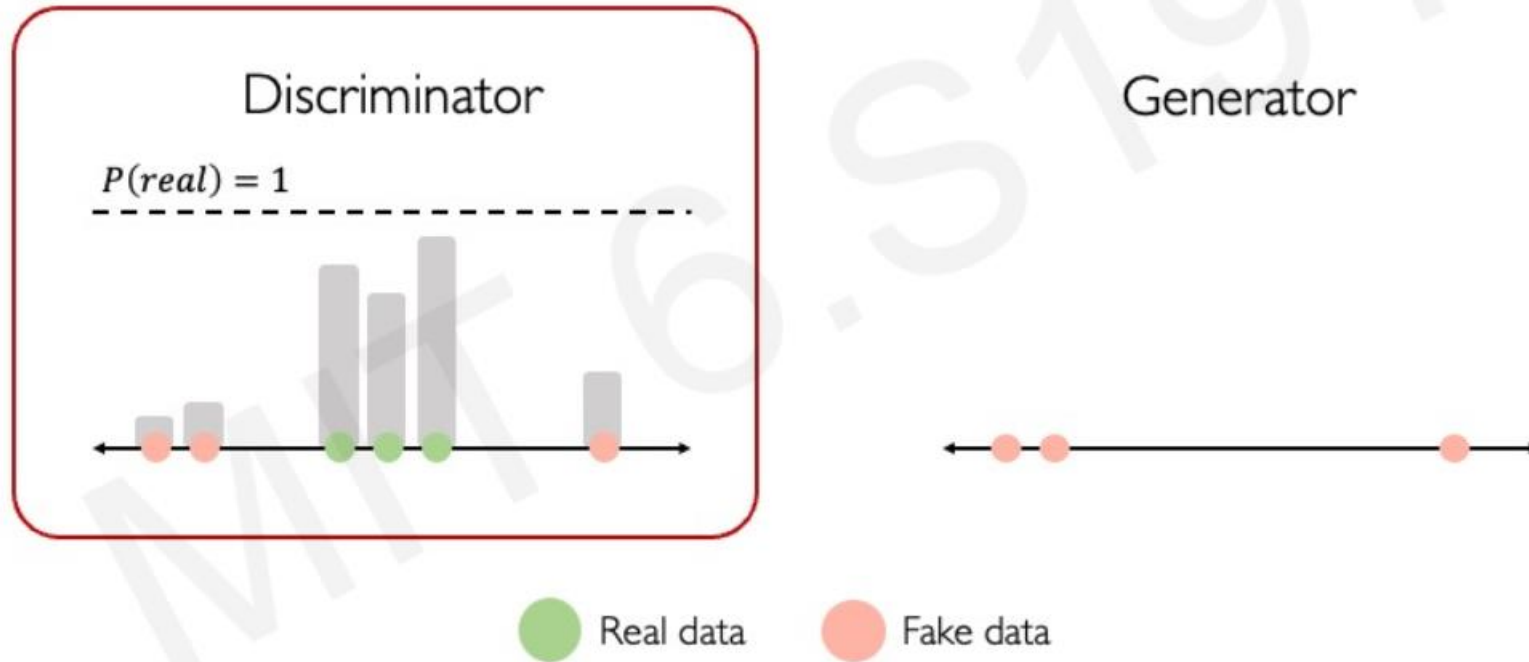
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



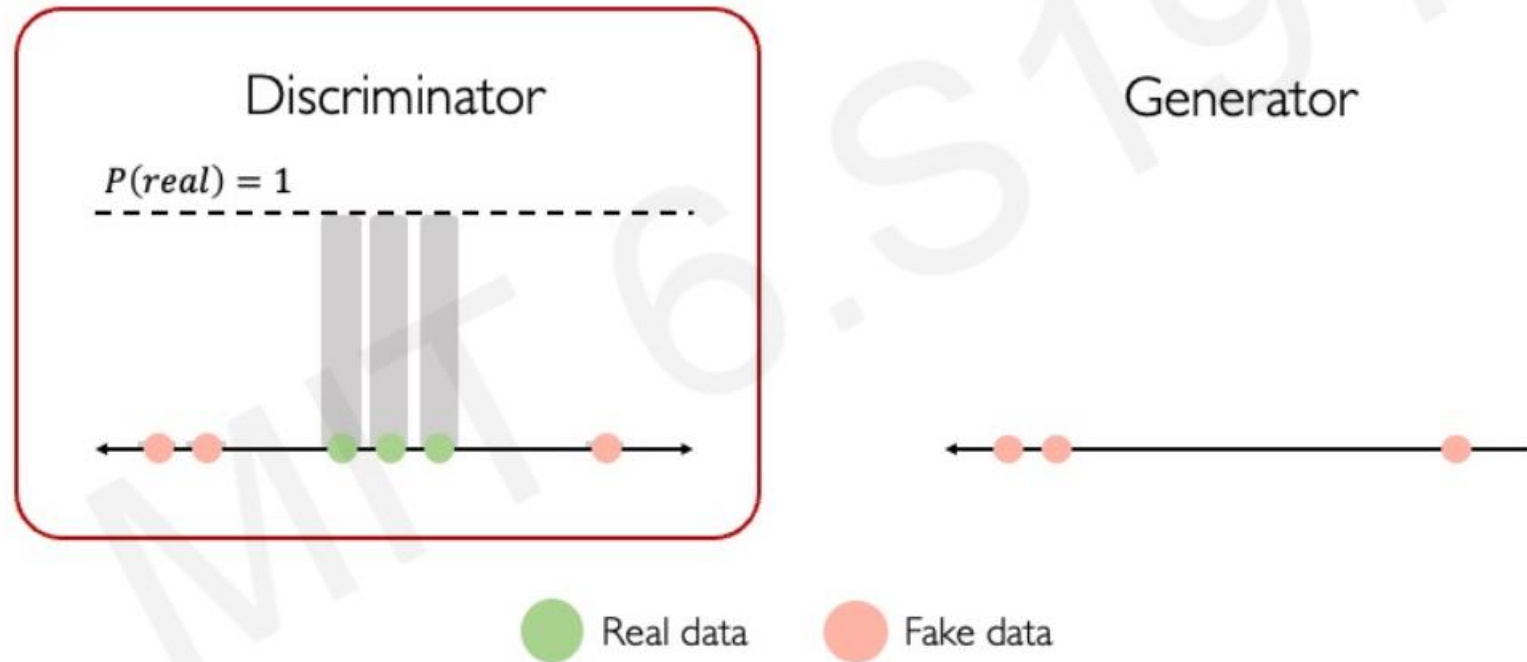
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



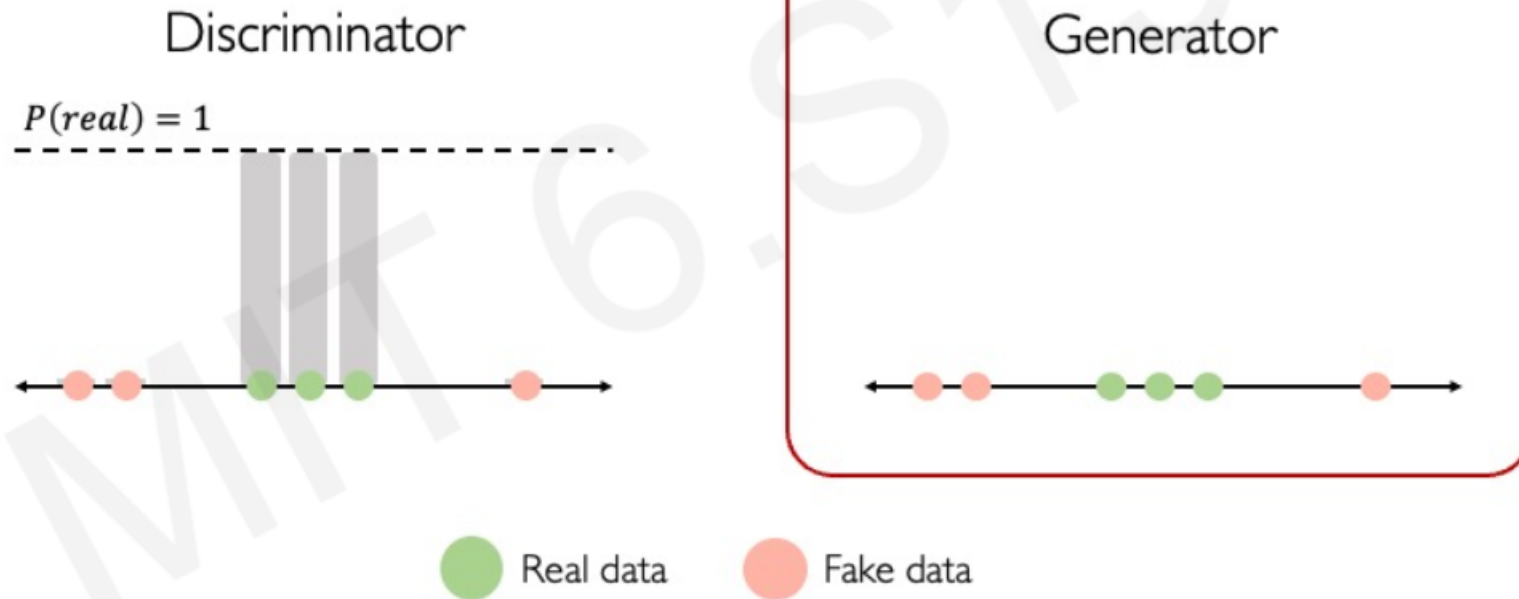
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



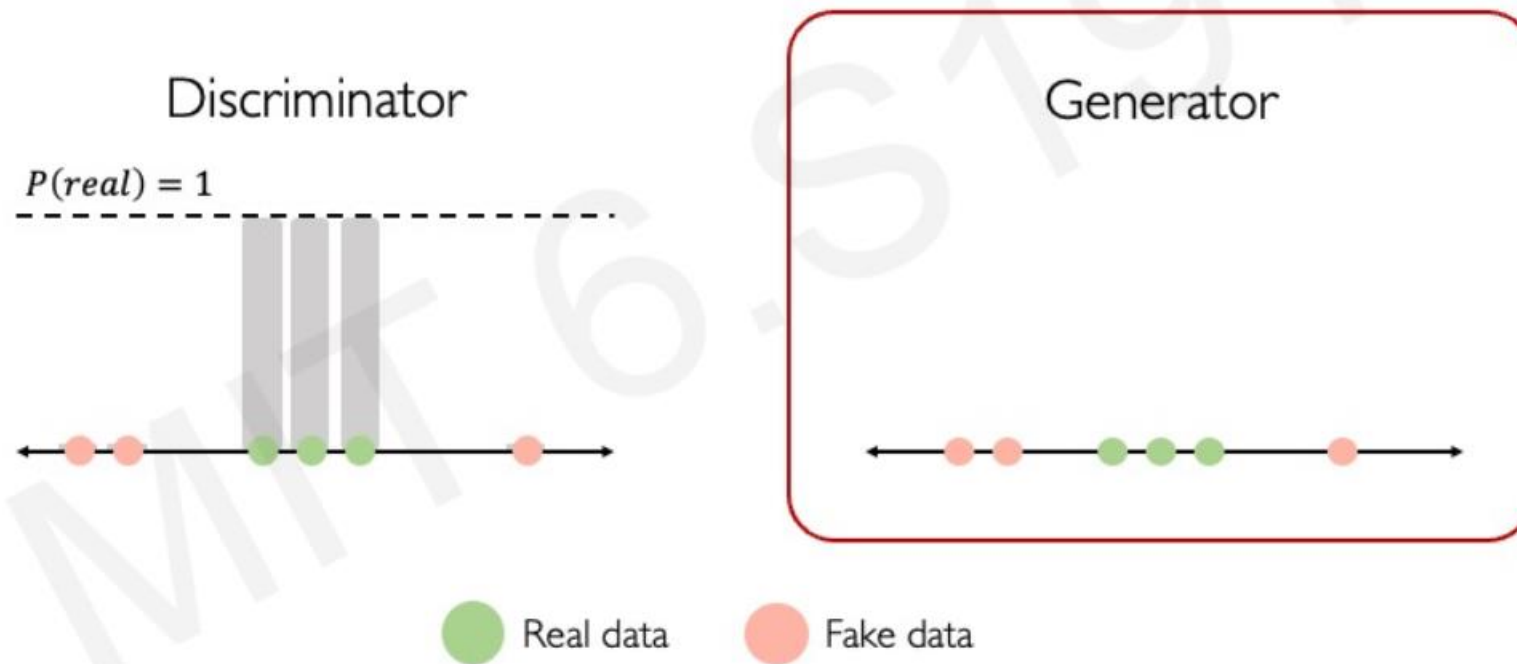
Generative Adversarial Networks

Generator tries to improve its imitation of the data.



Generative Adversarial Networks

Generator tries to improve its imitation of the data.



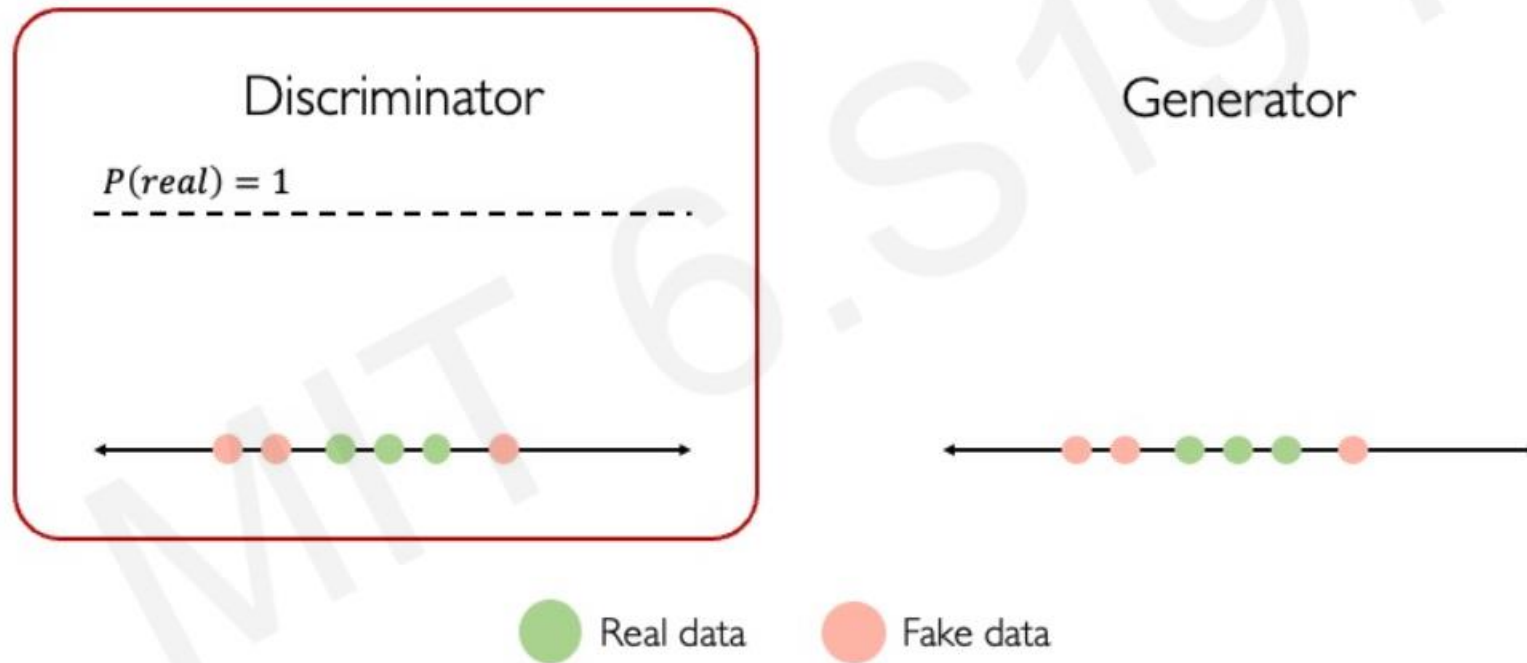
Generative Adversarial Networks

Generator tries to improve its imitation of the data.



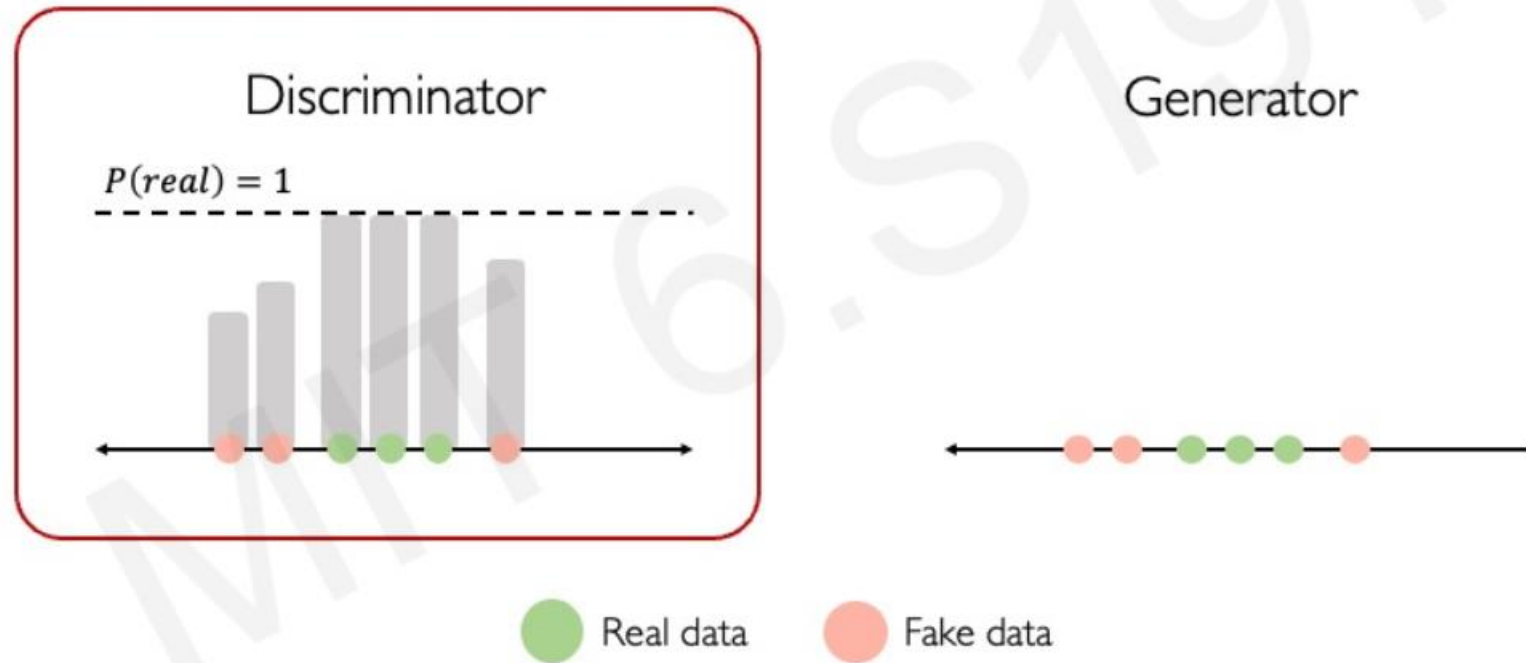
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



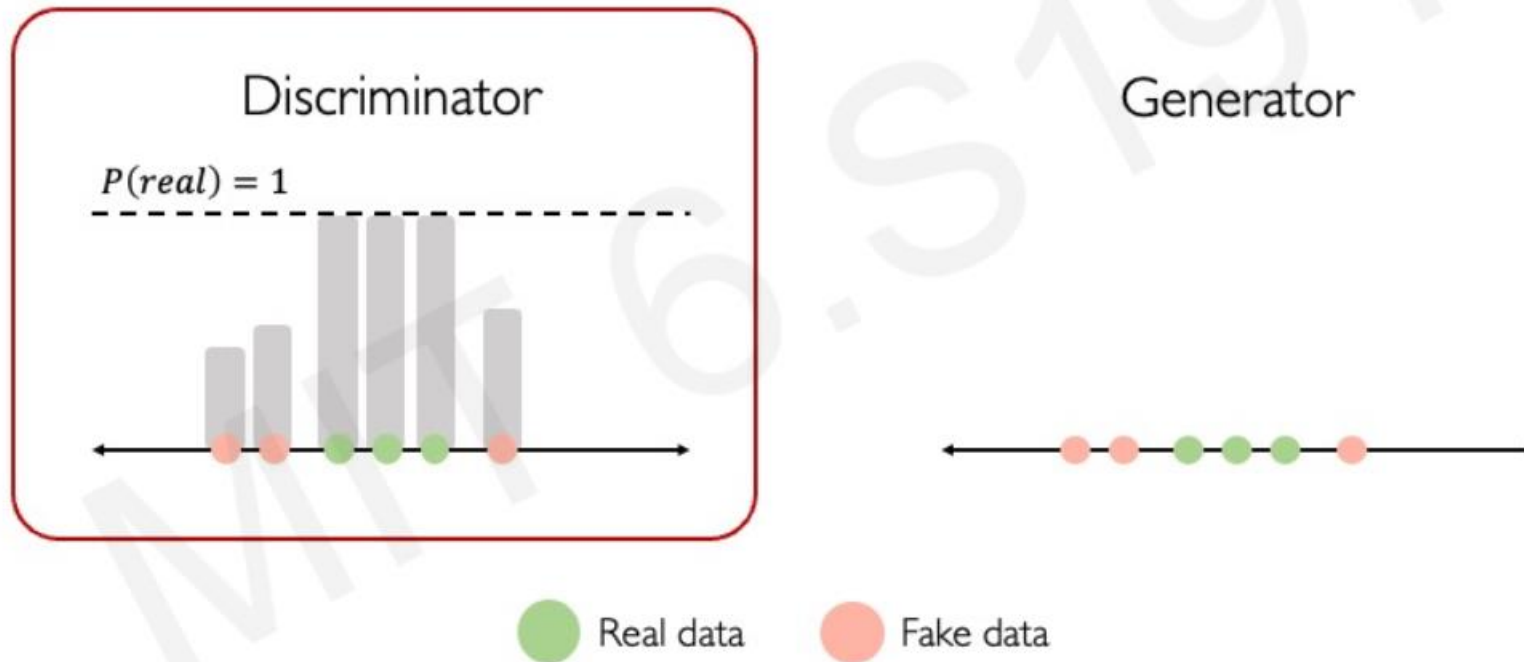
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.



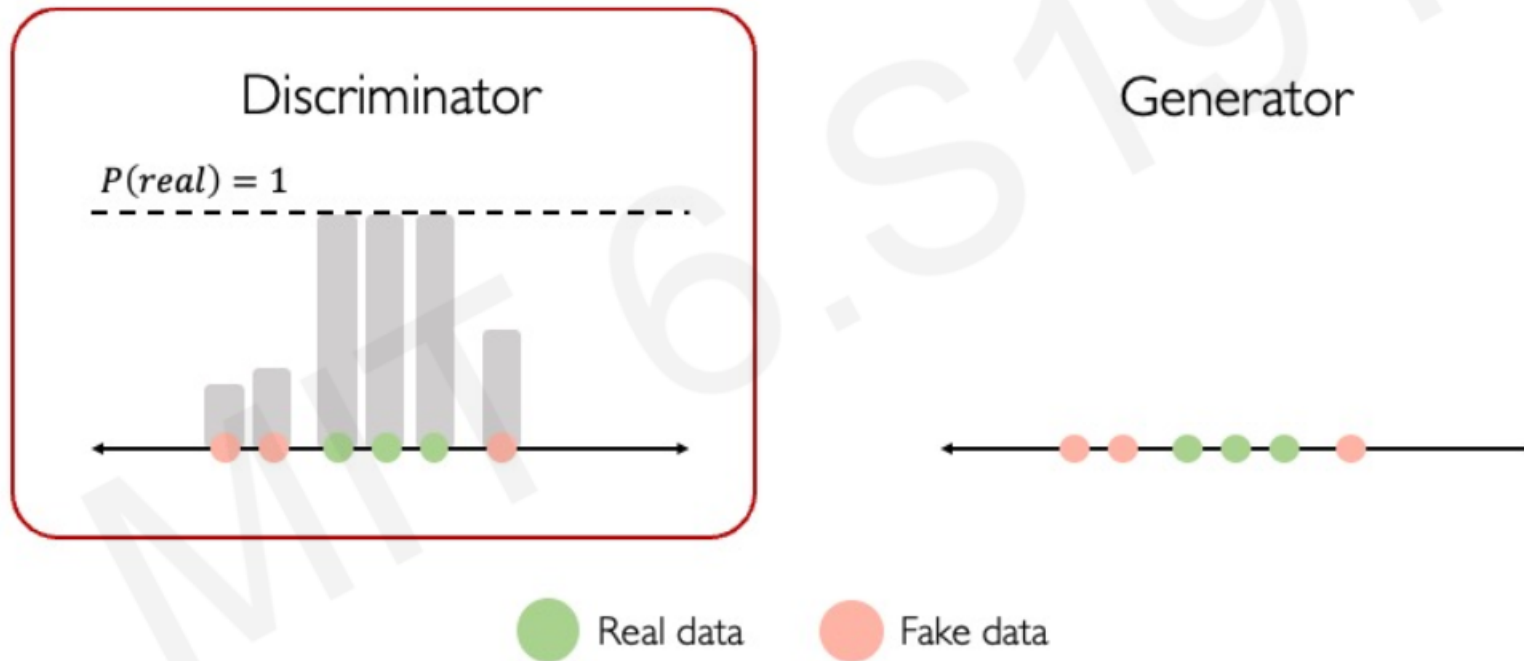
Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.

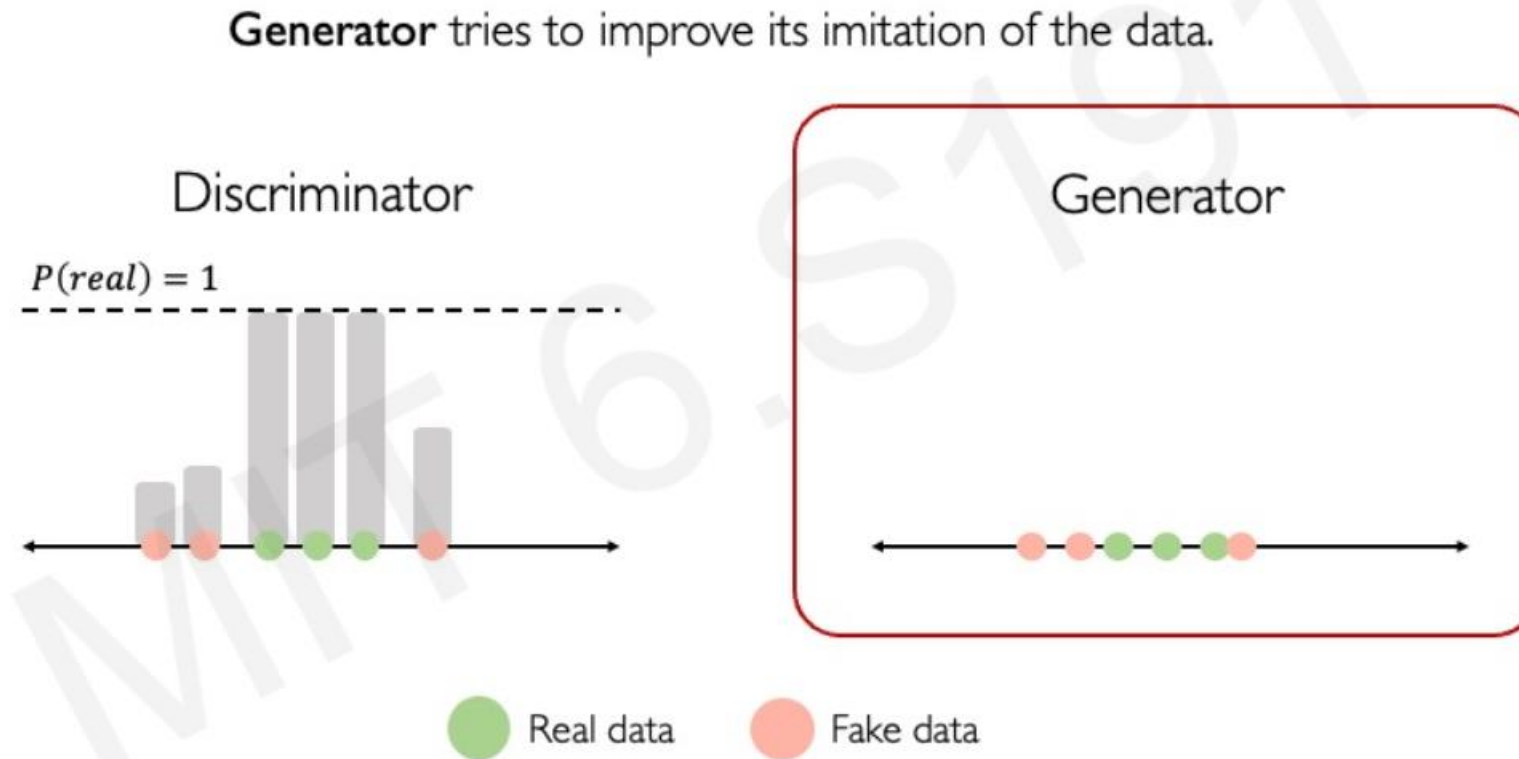


Generative Adversarial Networks

Discriminator tries to predict what's real and what's fake.

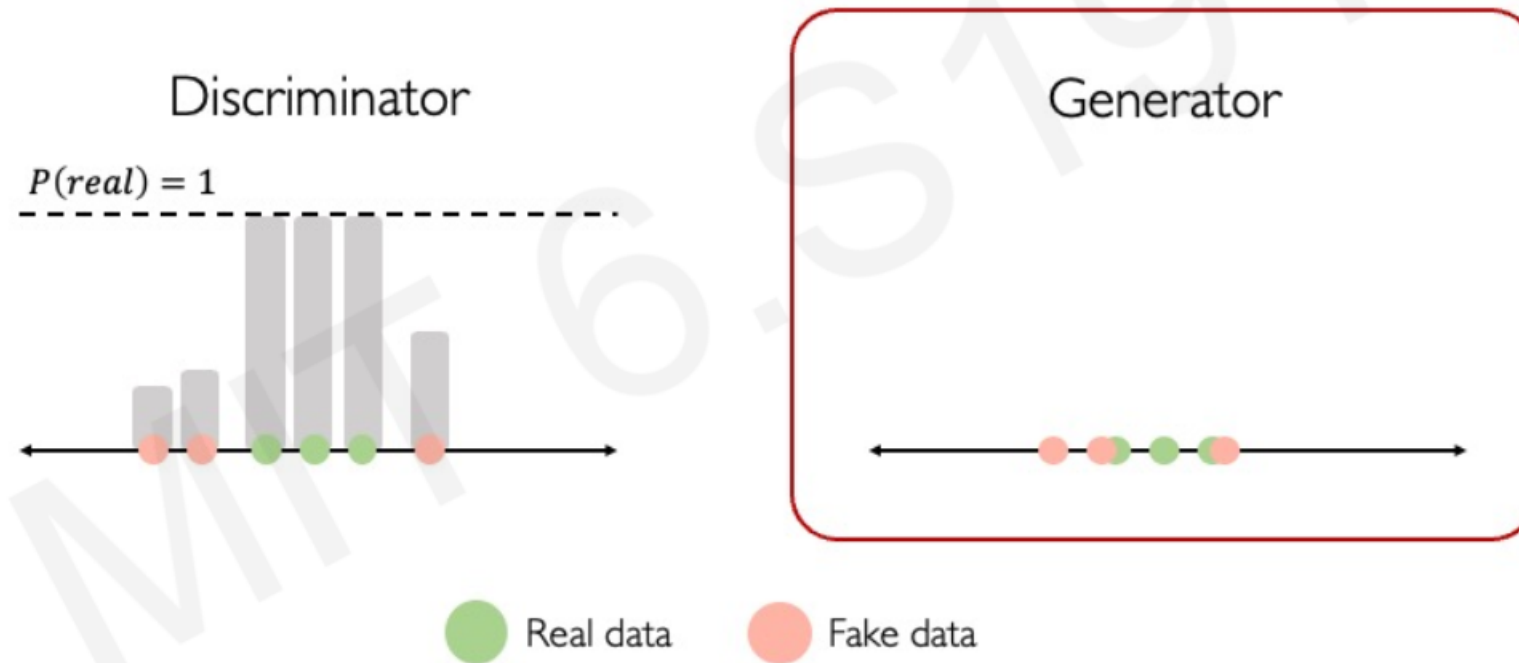


Generative Adversarial Networks



Generative Adversarial Networks

Generator tries to improve its imitation of the data.



Generative Adversarial Networks

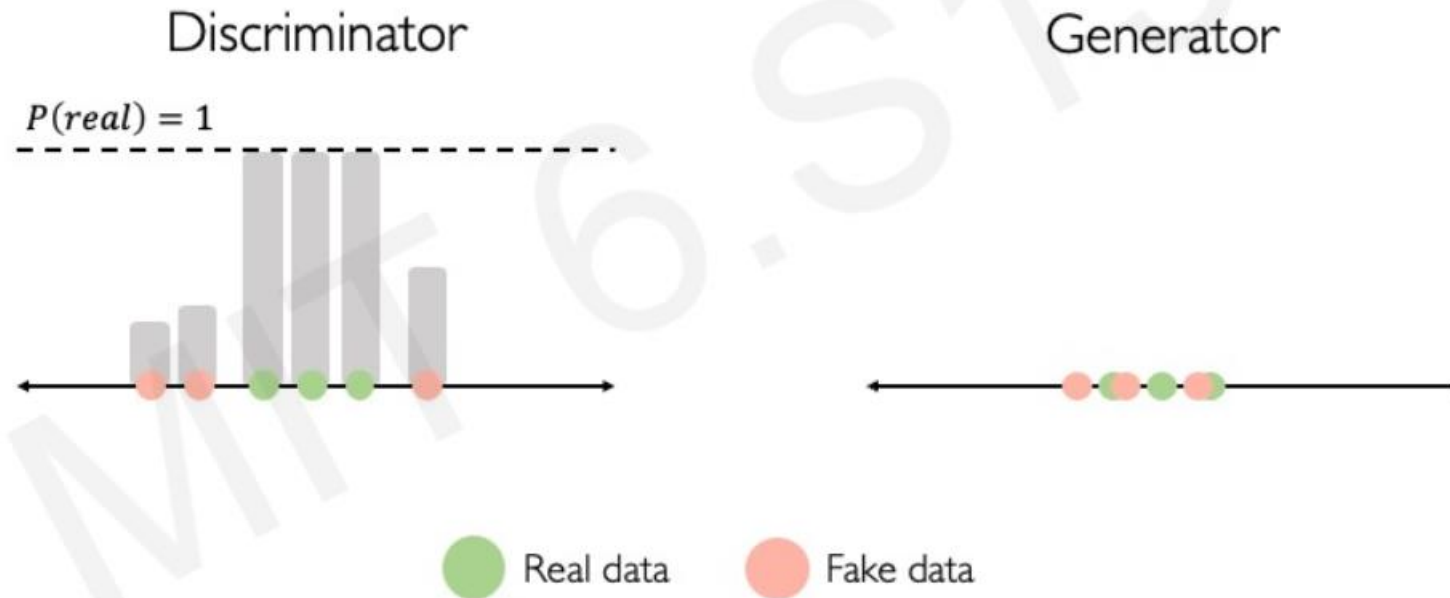
Generator tries to improve its imitation of the data.



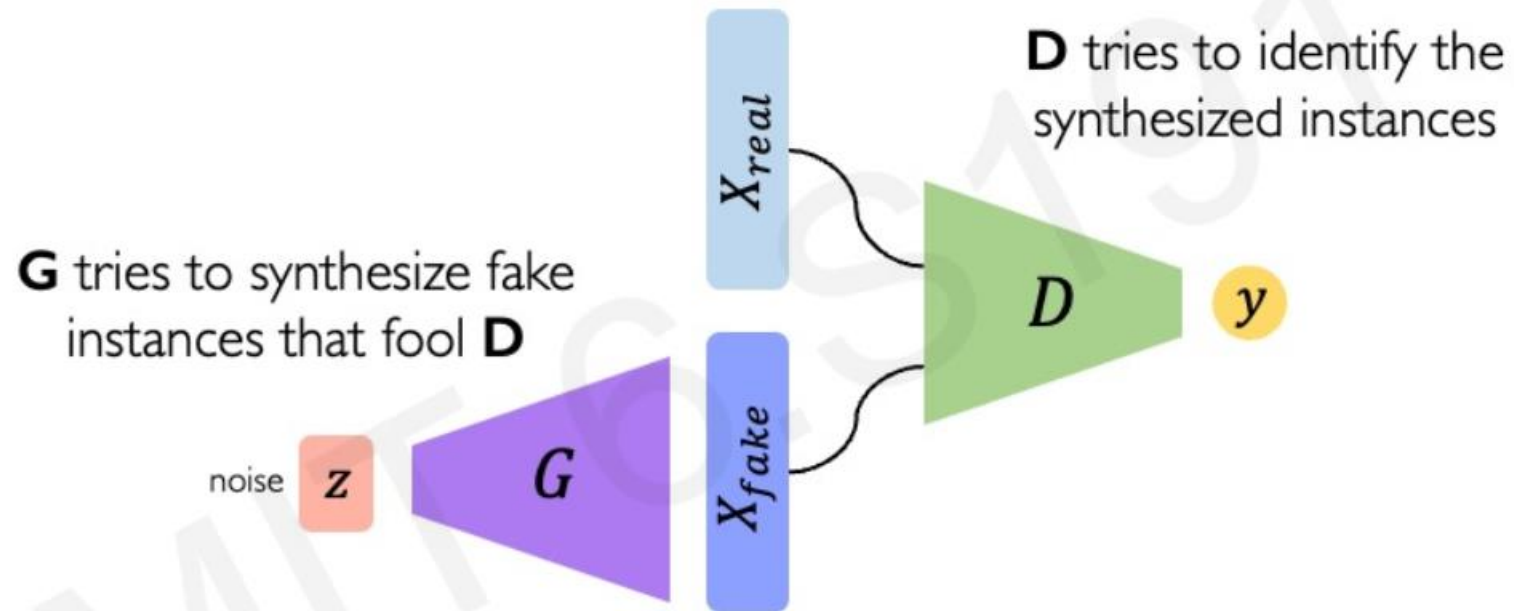
Generative Adversarial Networks

Discriminator tries to identify real data from fakes created by the generator.

Generator tries to create imitations of data to trick the discriminator.



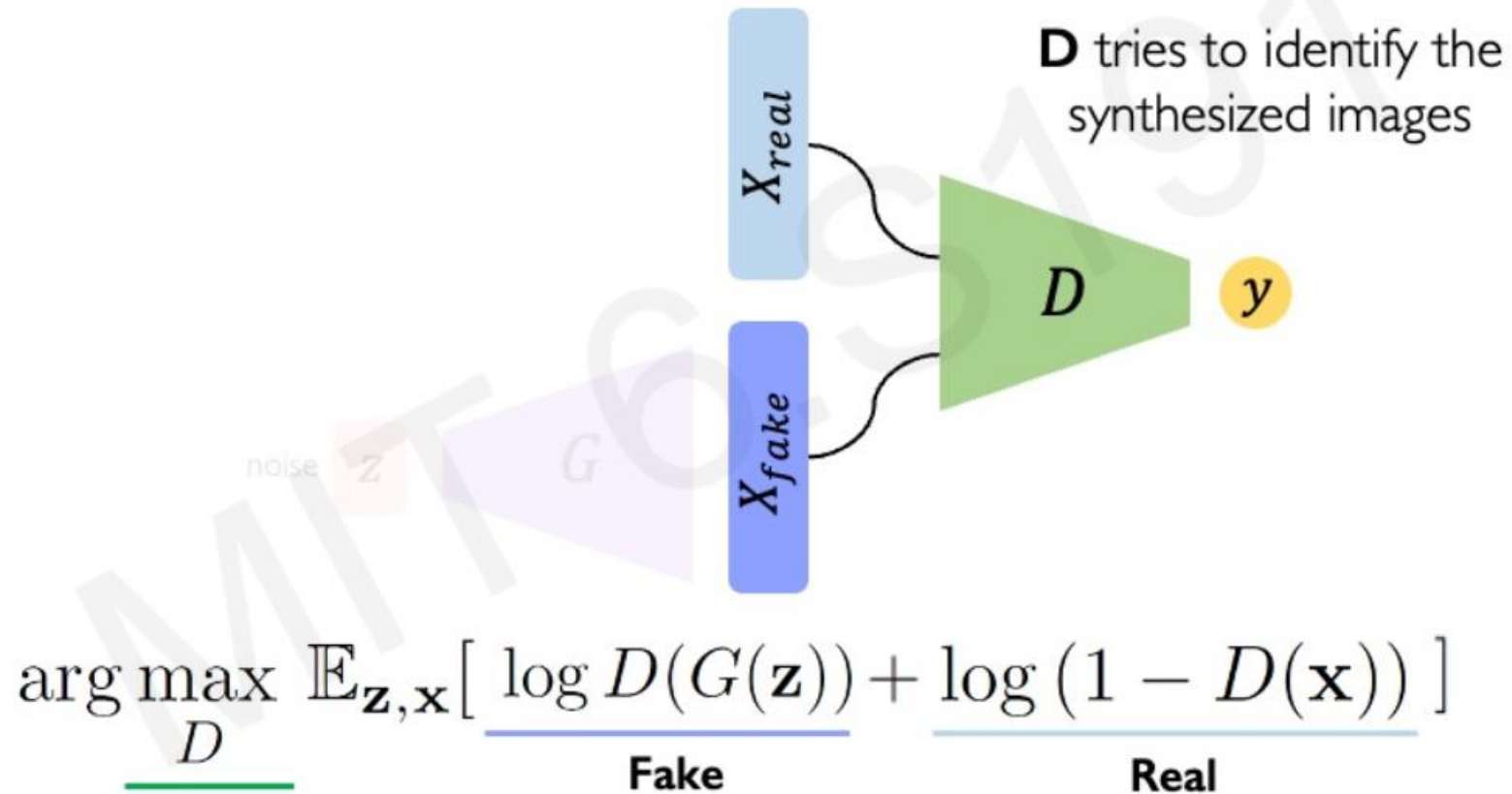
Generative Adversarial Networks



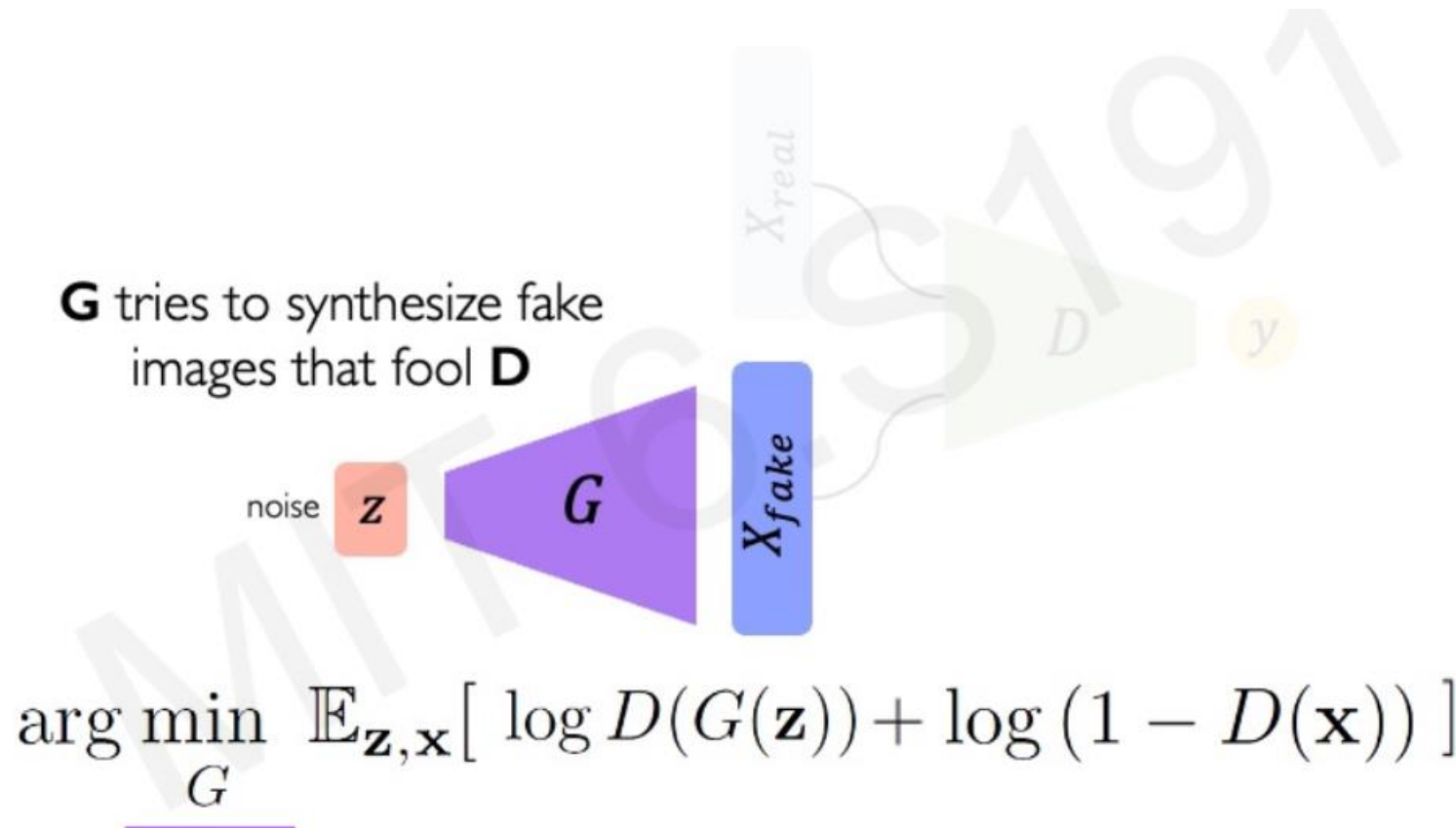
Training: adversarial objectives for D and G

Global optimum: G reproduces the true data distribution

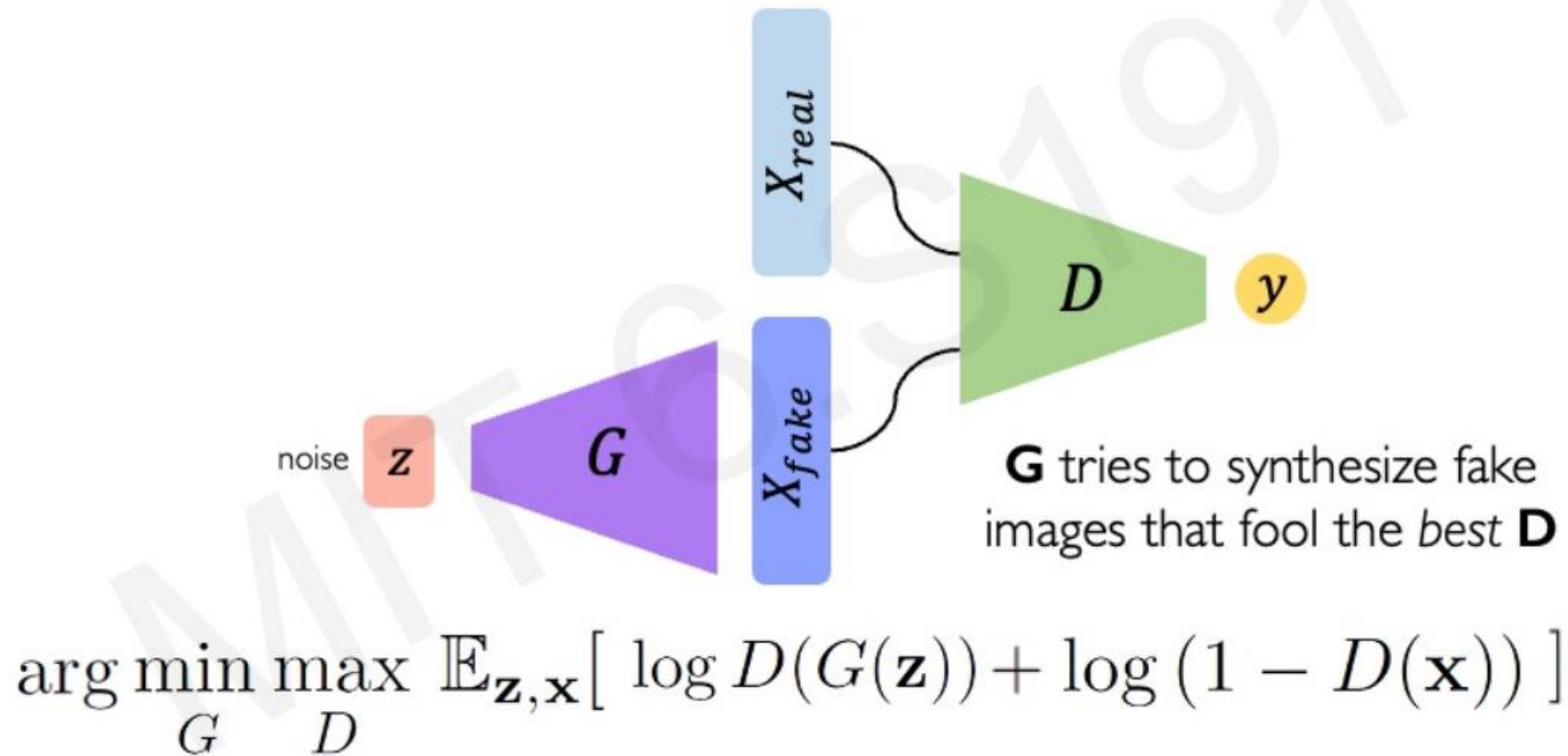
Generative Adversarial Networks



Generative Adversarial Networks

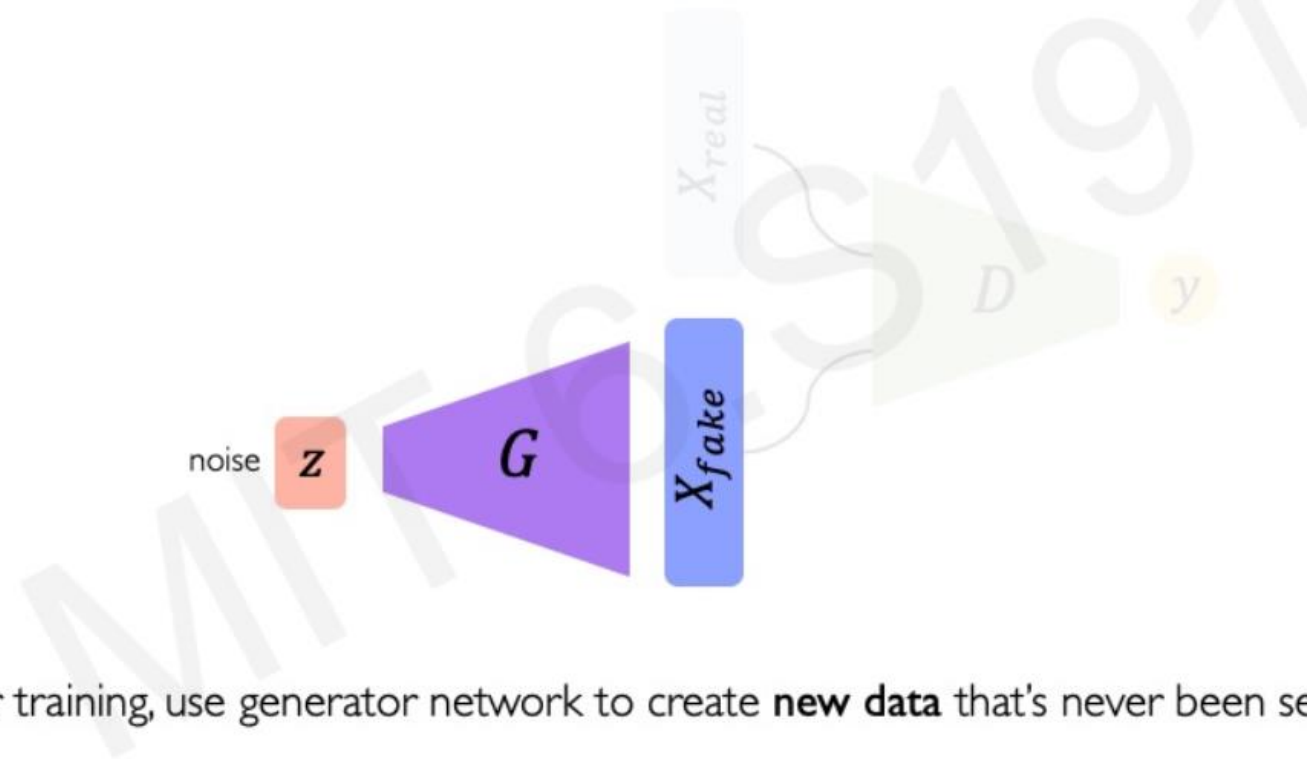


Generative Adversarial Networks



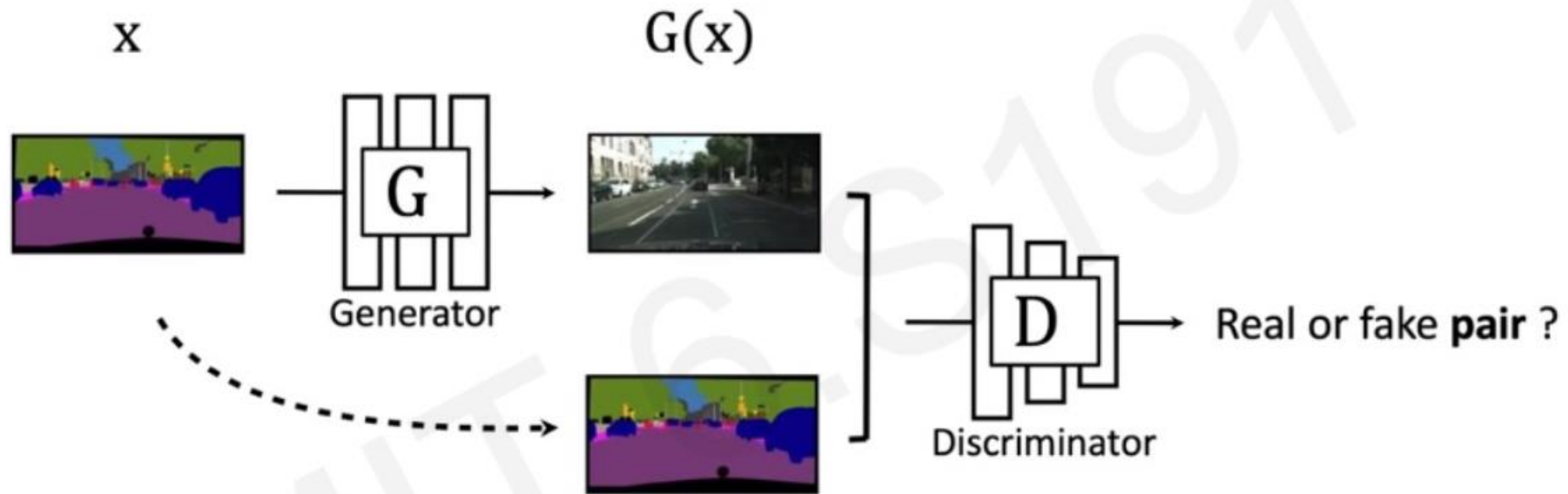
Generation

Generating new data with GANs



After training, use generator network to create **new data** that's never been seen before.

Applications



The discriminator, D , classifies between fake and real **pairs**.
The generator, G , learns to fool the discriminator.

Applications

Map → Aerial View

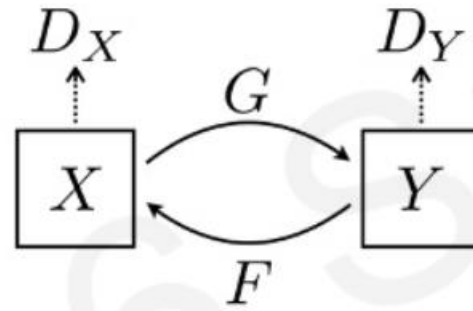


Aerial View → Map



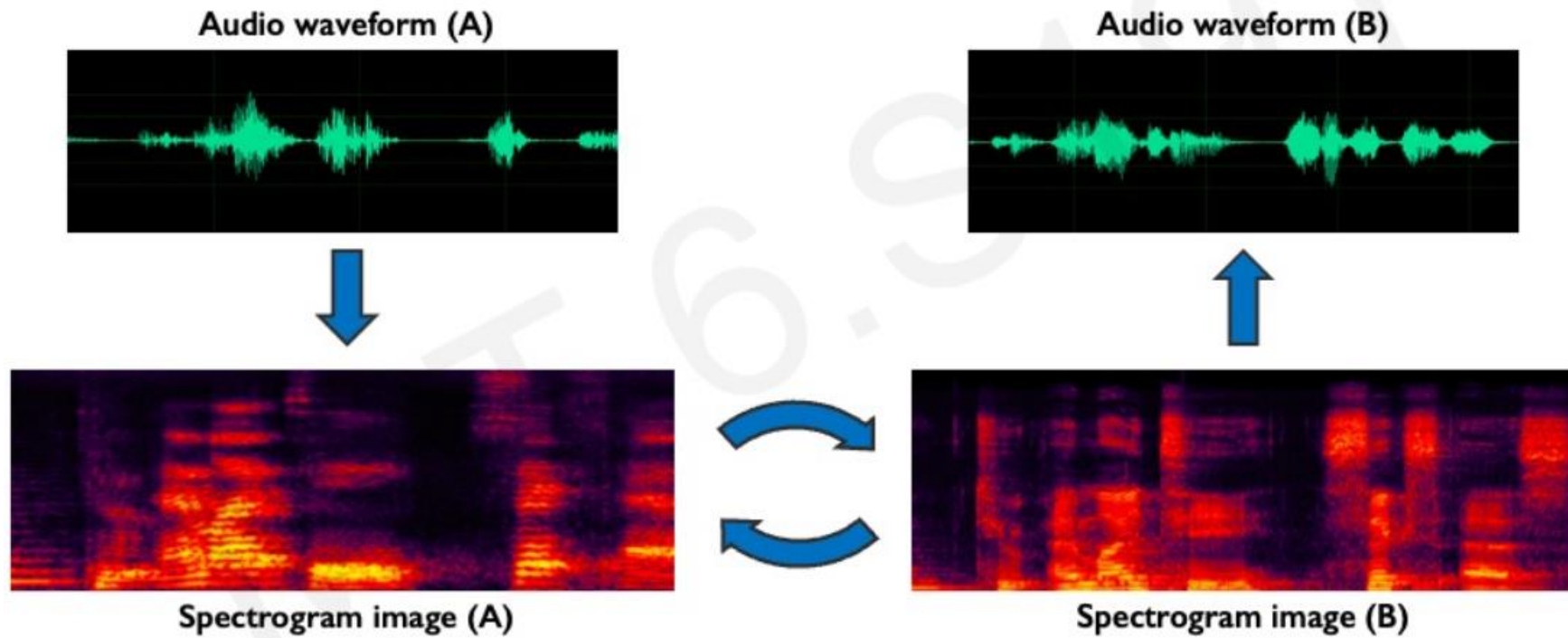
Applications

CycleGAN learns transformations across domains with unpaired data.

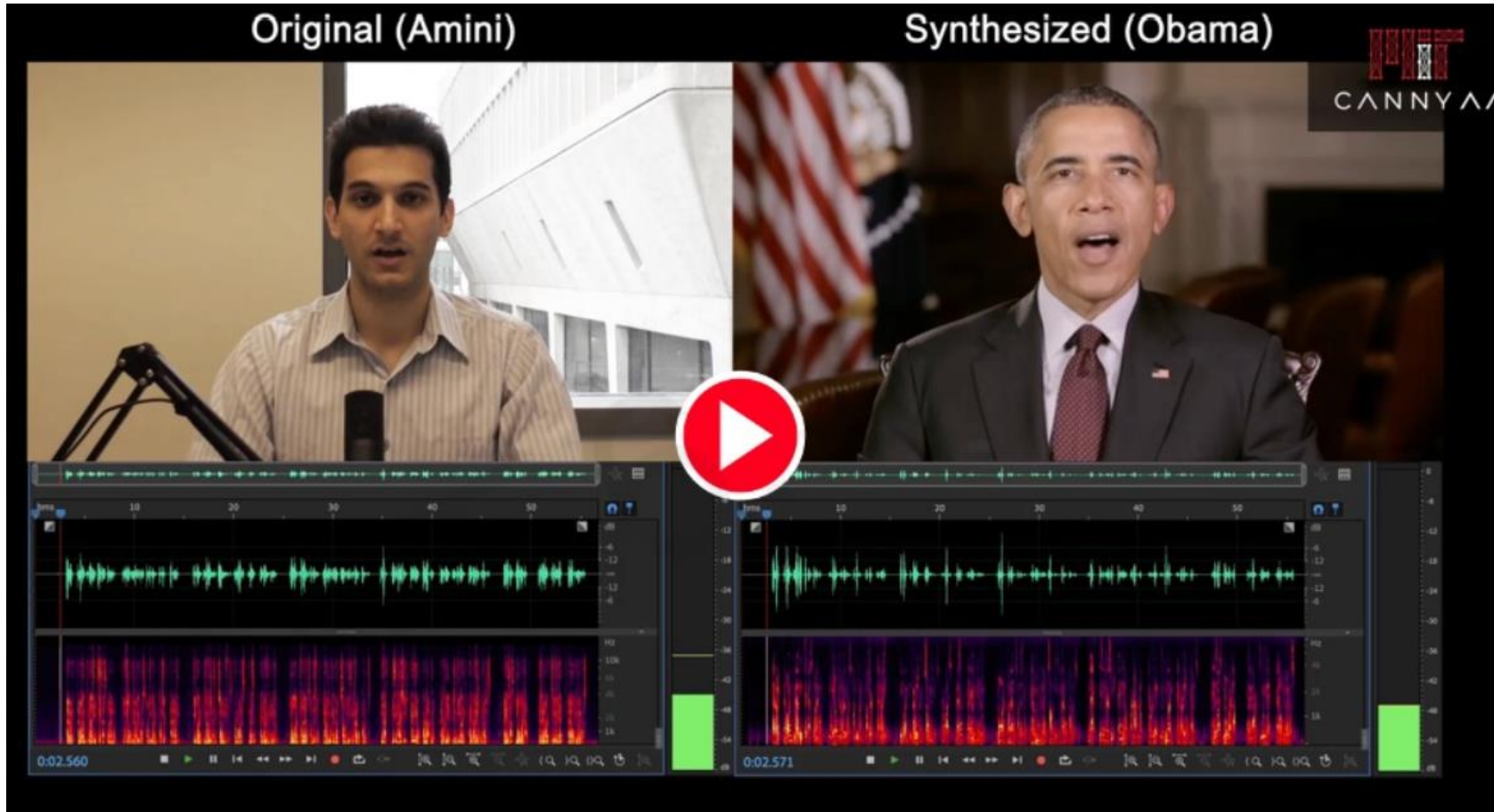


<https://www.youtube.com/watch?v=9reHvktowLY>

Applications



Applications



<https://www.youtube.com/watch?v=l82PxsKHxYc>

Thank you!!!