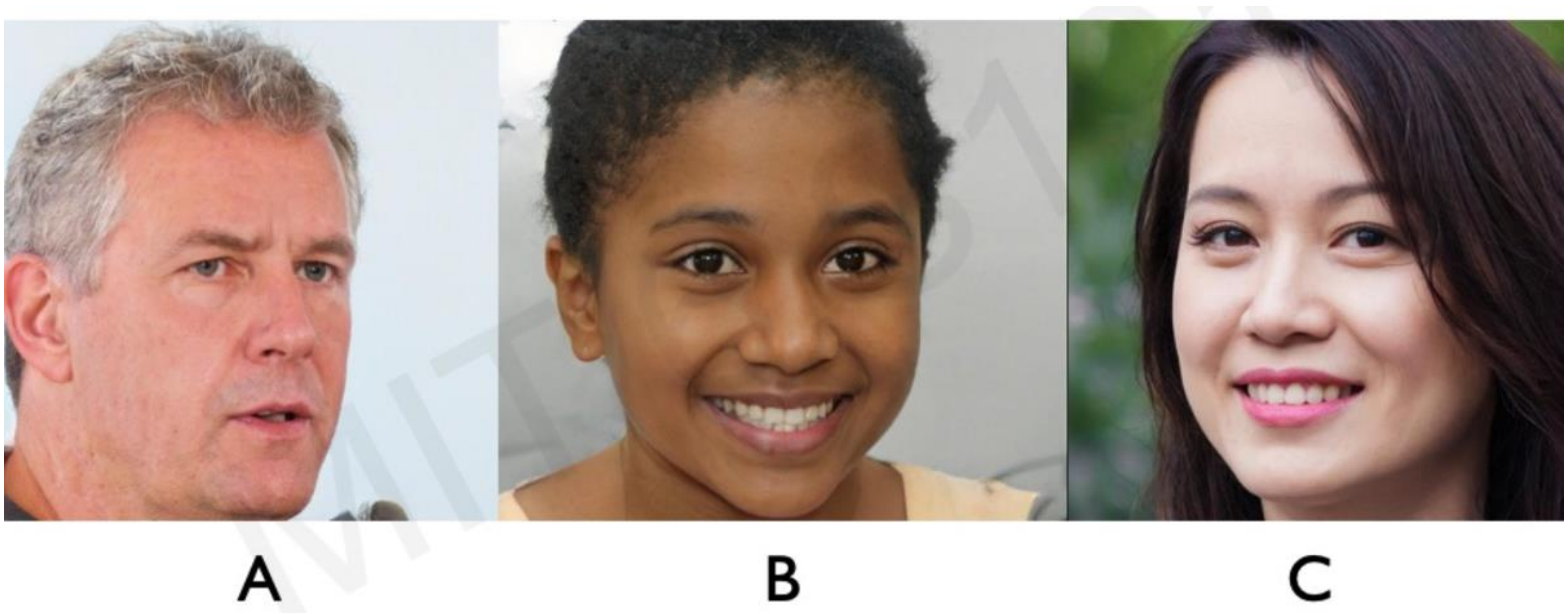
Generative Modelling

Rina BUOY

AMERICAN UNIVERSITY
OF PHNOM PENH

STUDY LOCALLY. LIVE GLOBALLY.

Which one is real face ?



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.

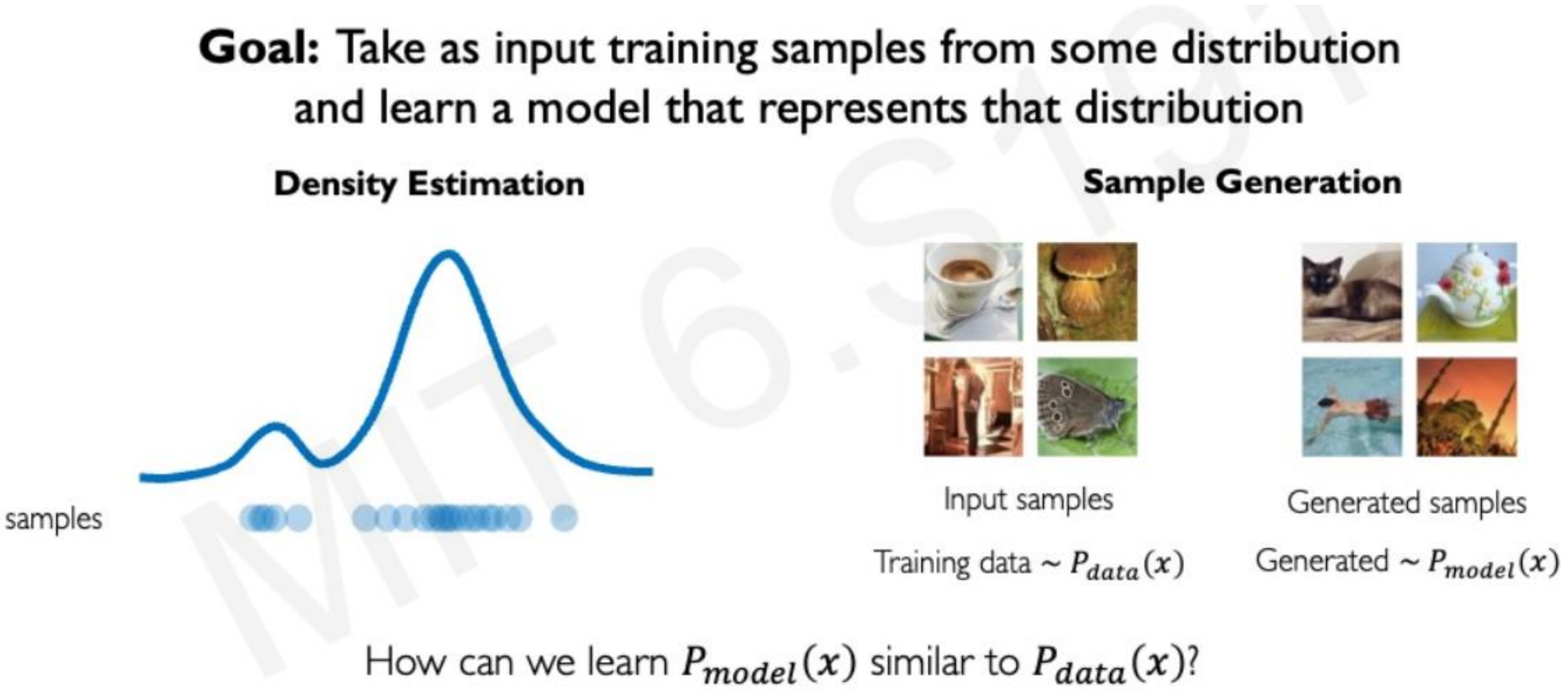
Unsupervised Learning

Data: x
x is data, no labels!

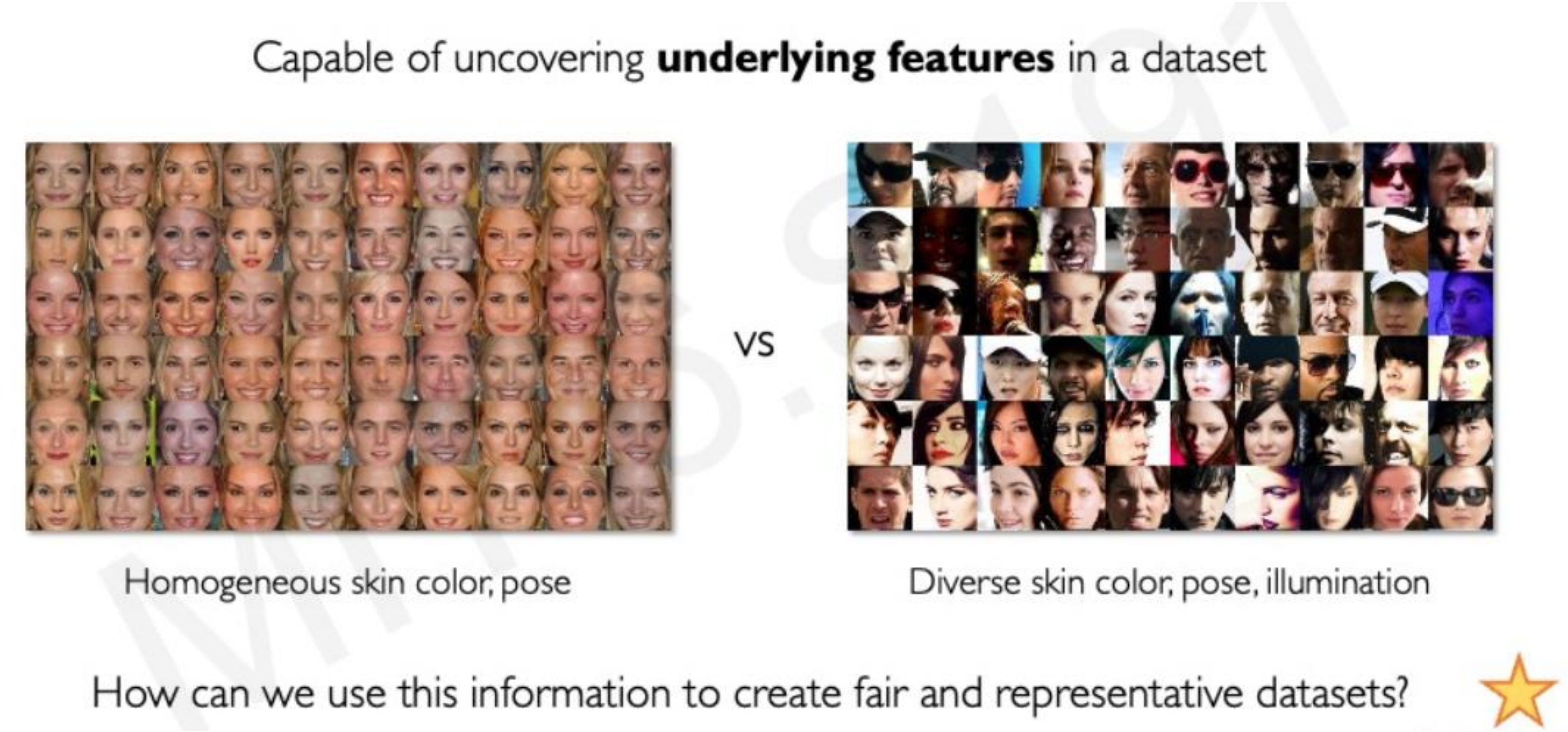
Goal: Learn the *hidden* or *underlying structure* of the data

Examples: Clustering, feature or dimensionality reduction, etc.

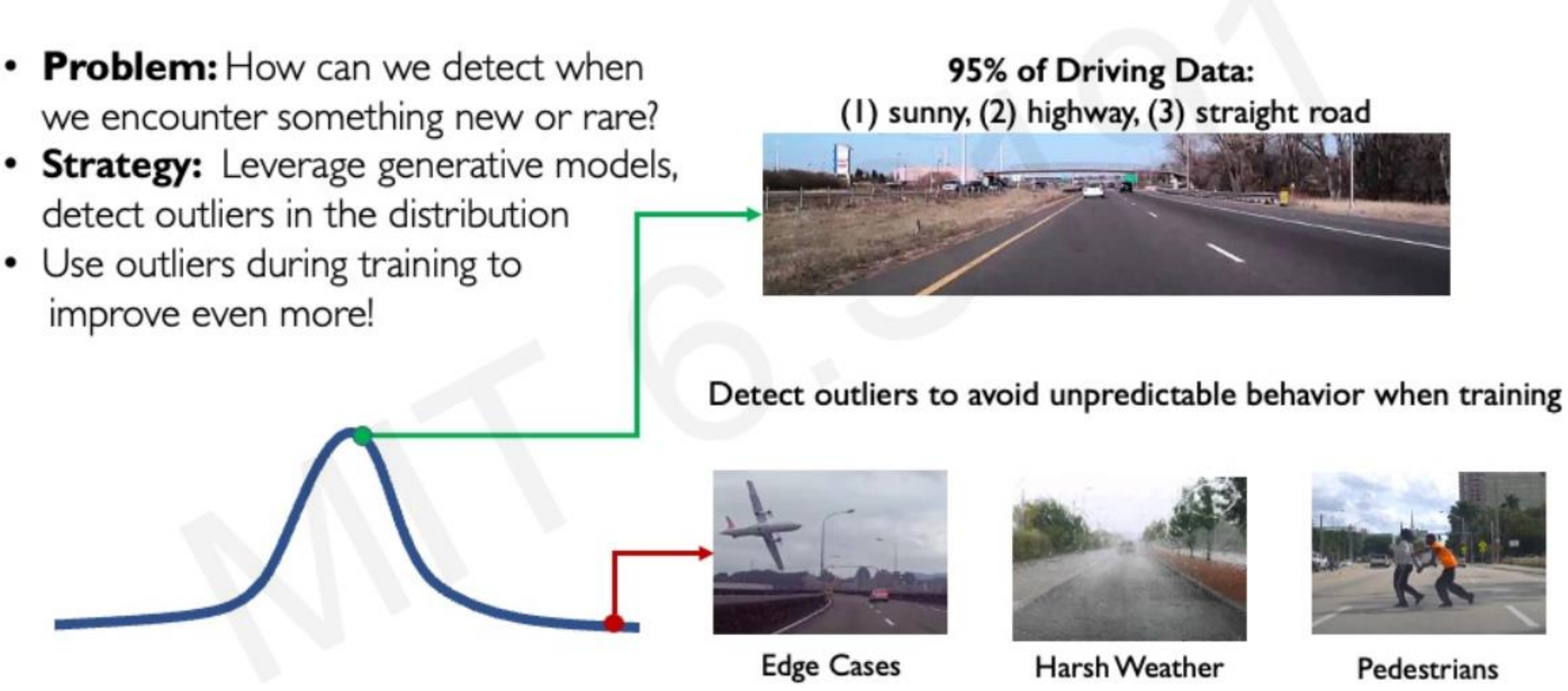
Generative Modelling



Why Generative Modelling ?



Why Generative Modelling ?



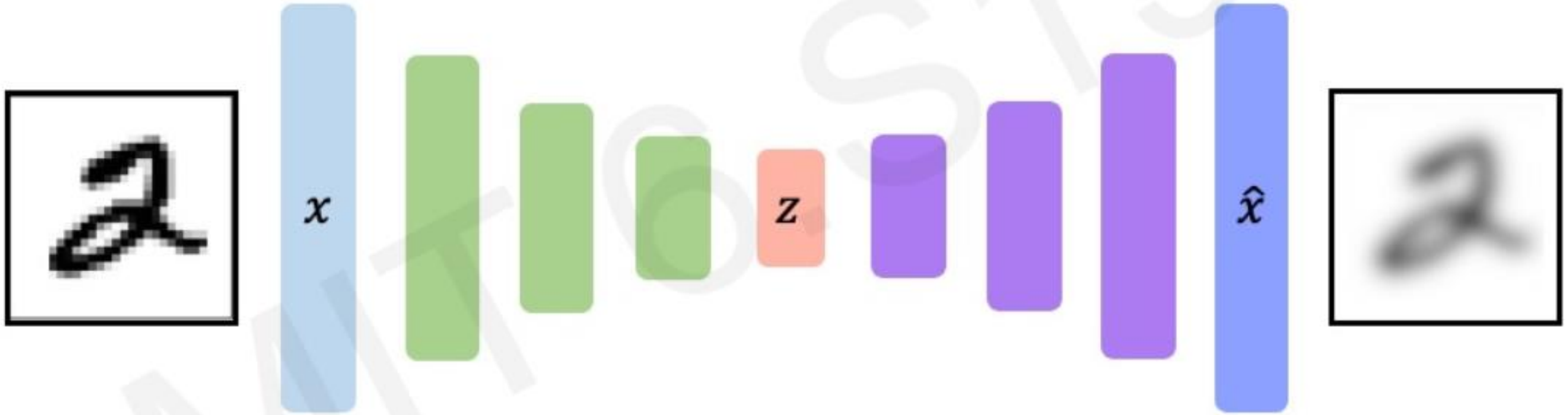
Autoencoders

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

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

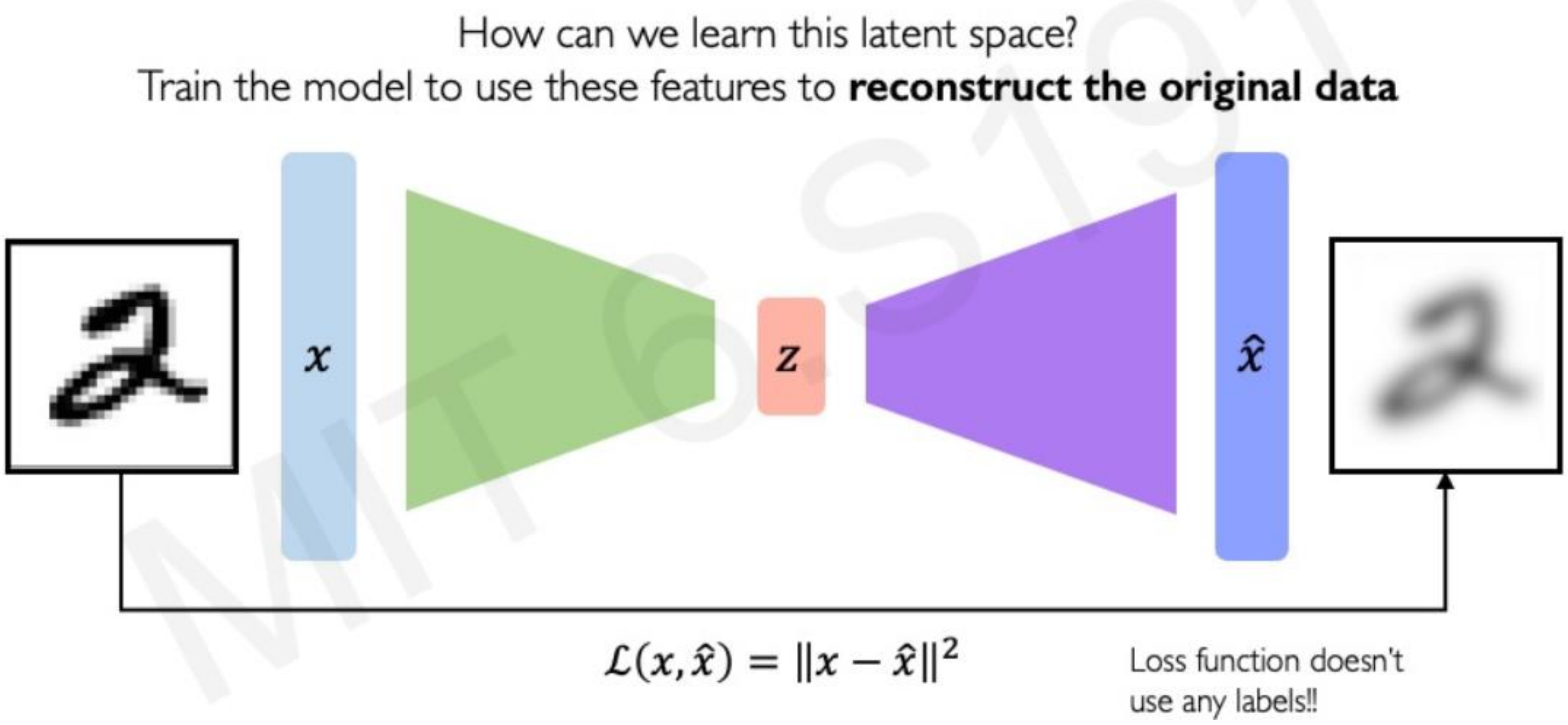
Autoencoders

How can we learn this latent space?
Train the model to use these features to **reconstruct the original data**



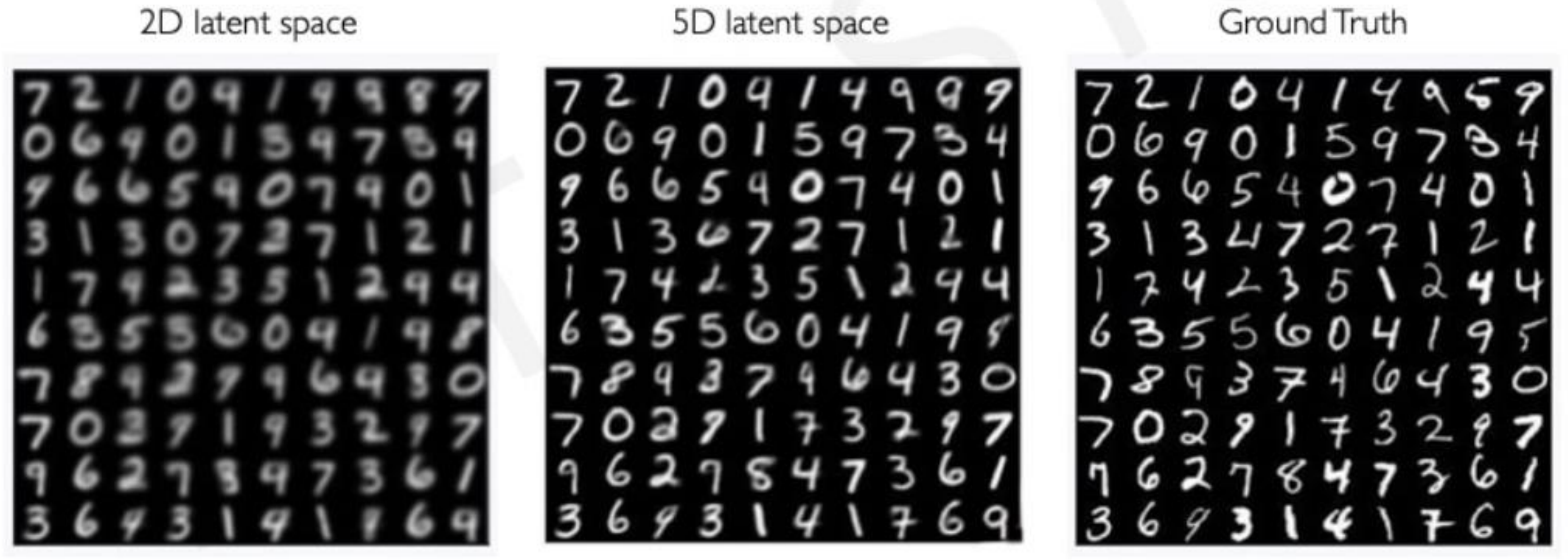
"Decoder" learns mapping back from latent space, z,
to a reconstructed observation, $\hat{x}$

Autoencoders



Dimensionality of Latent Space

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



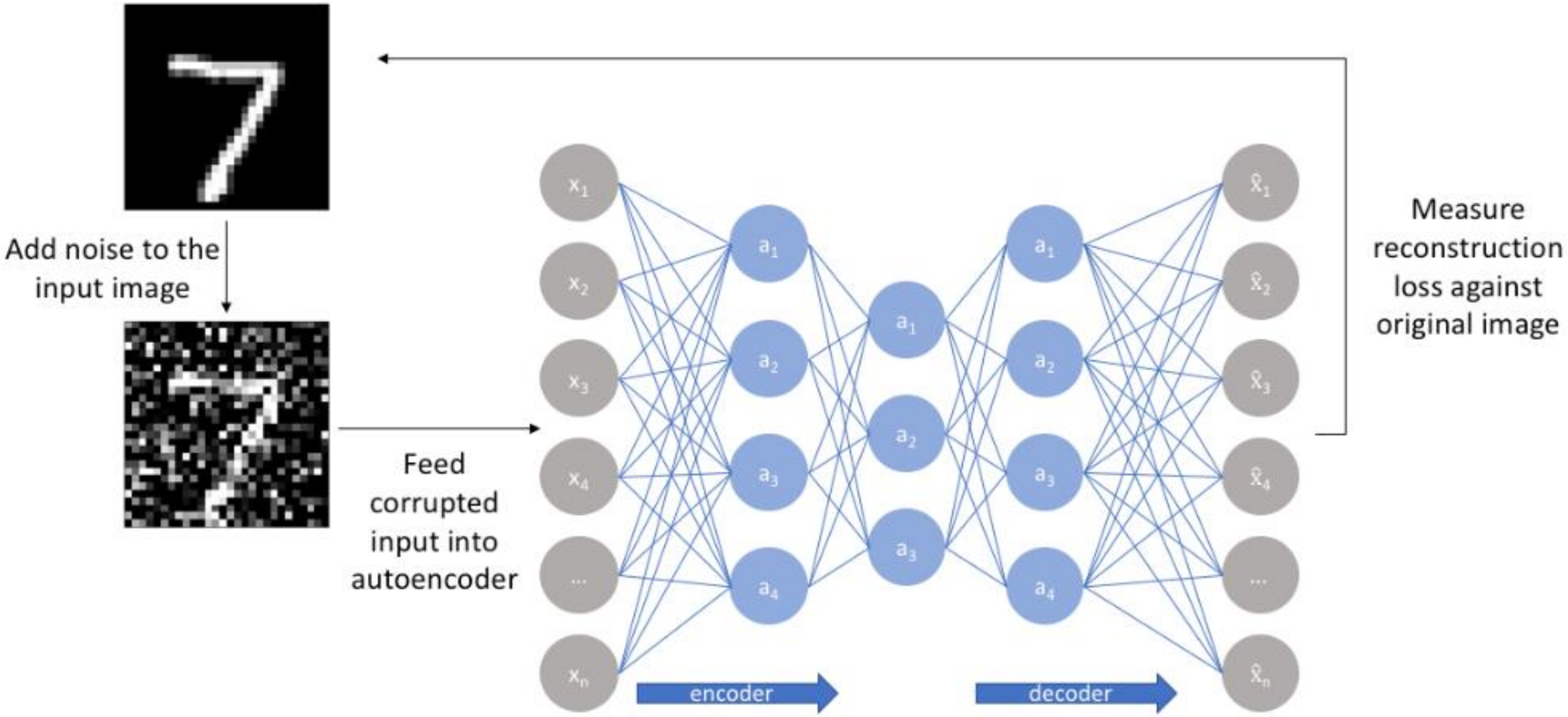
Autoencoders for Learning Representation

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

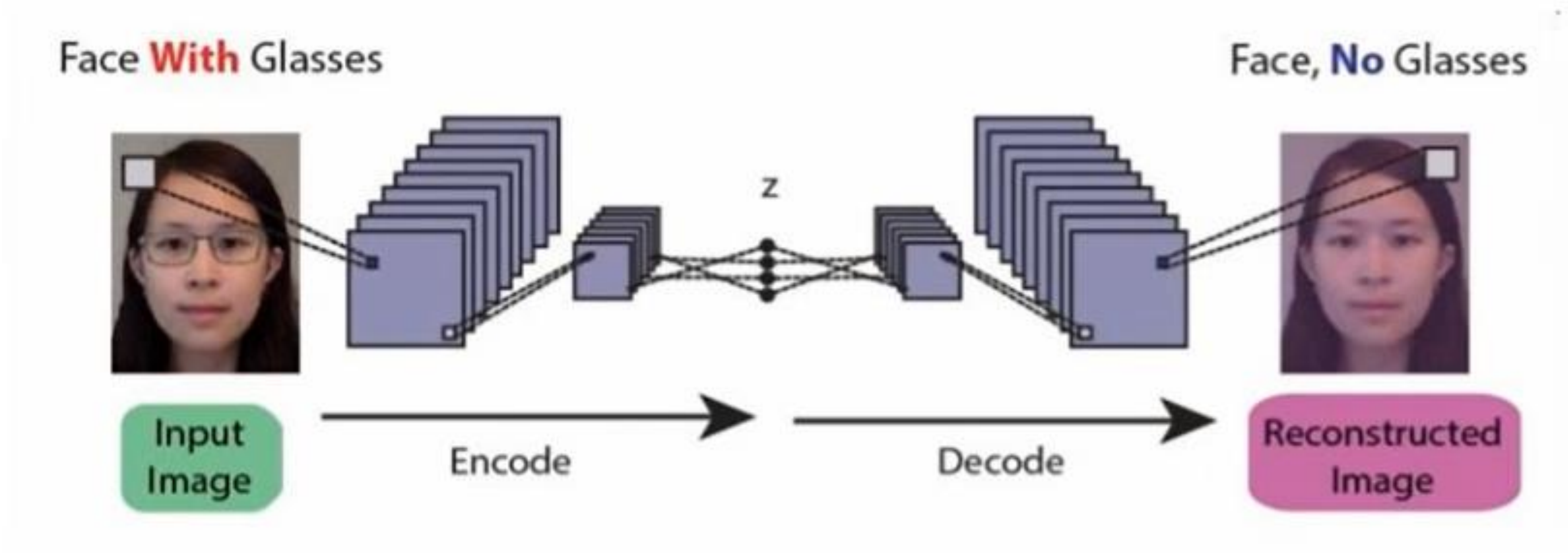
Autoencoding = **Auto**matically **encoding** data; "Auto" = **self**-encoding

Examples – Denoising

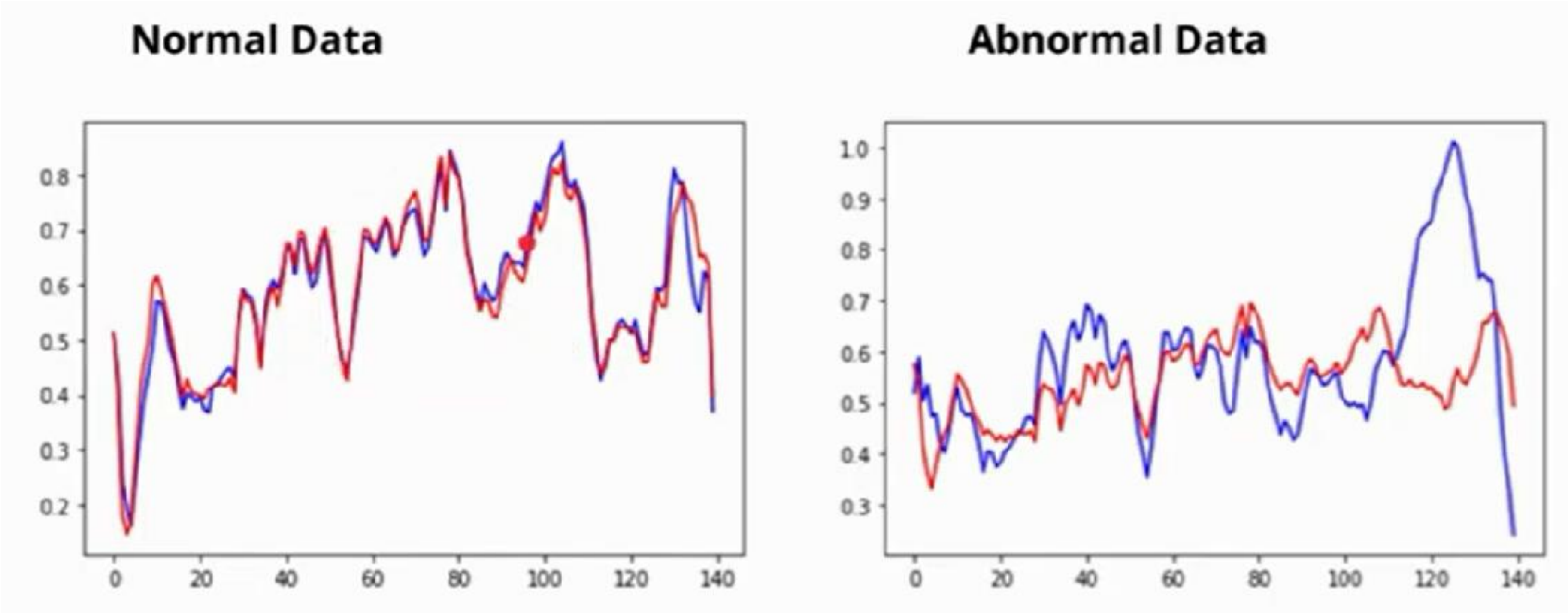


https://www.jeremyjordan.me/autoencoders/

Examples – Denoising



Examples – Outlier Detection



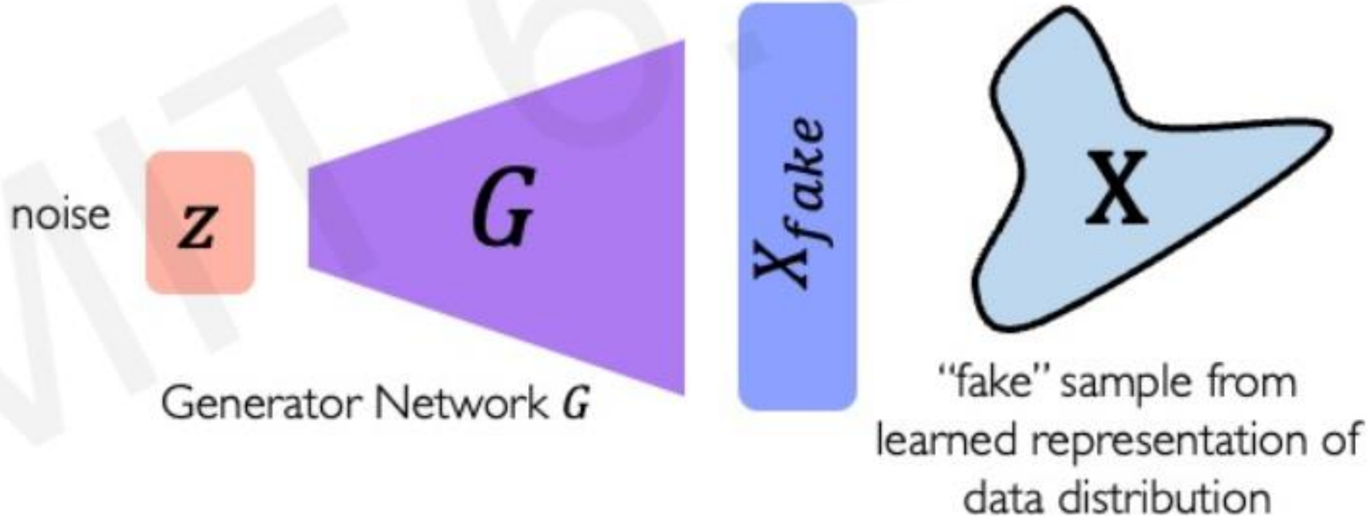
Examples – Image Colorization

Generative Adversarial Networks

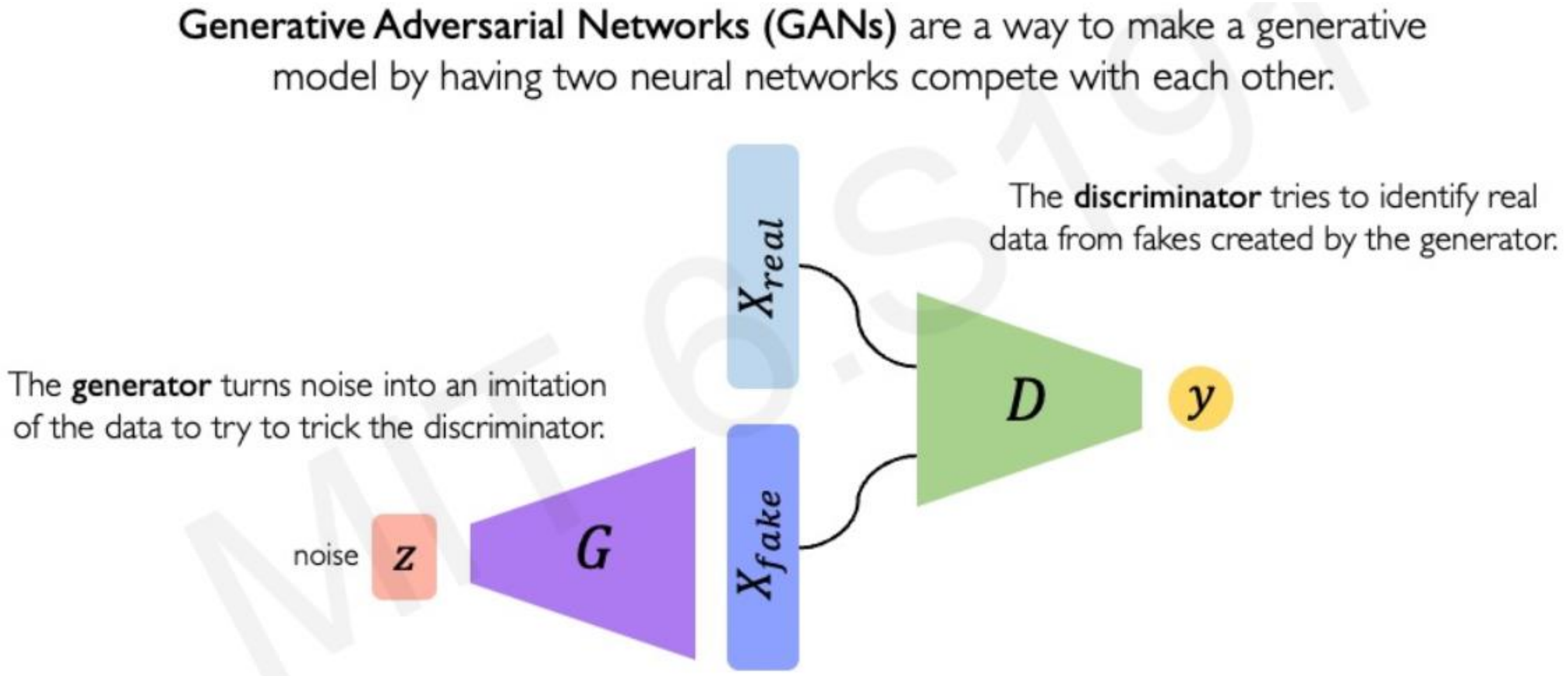
Idea: don't explicitly model density, and instead just sample to generate new instances.

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.

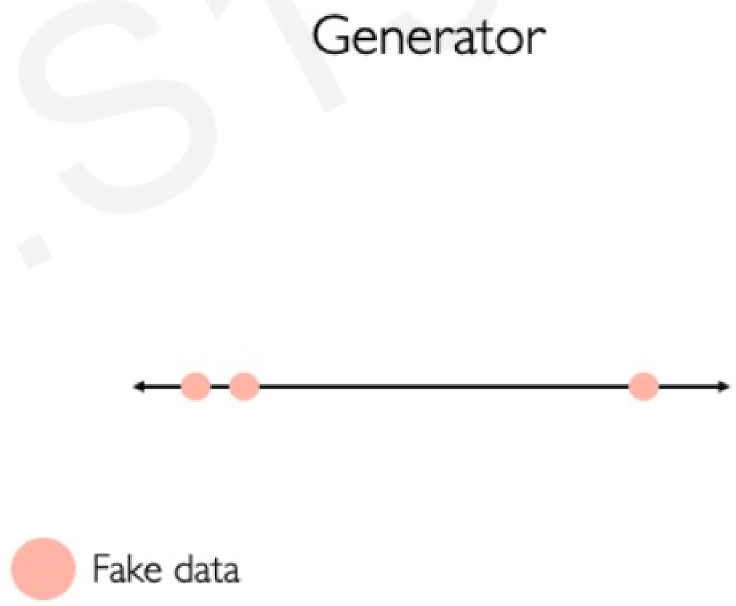


Generative Adversarial Networks

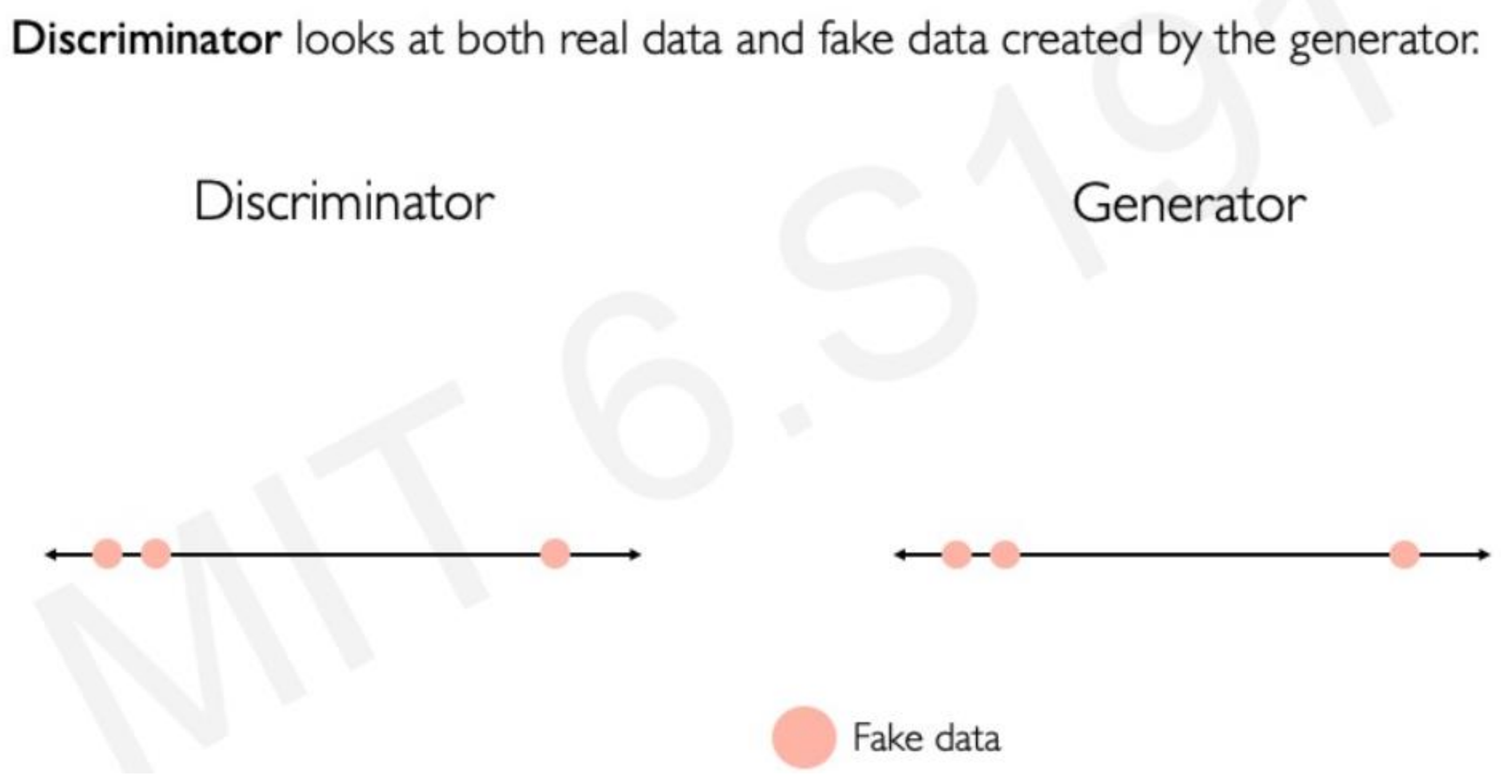


Generative Adversarial Networks

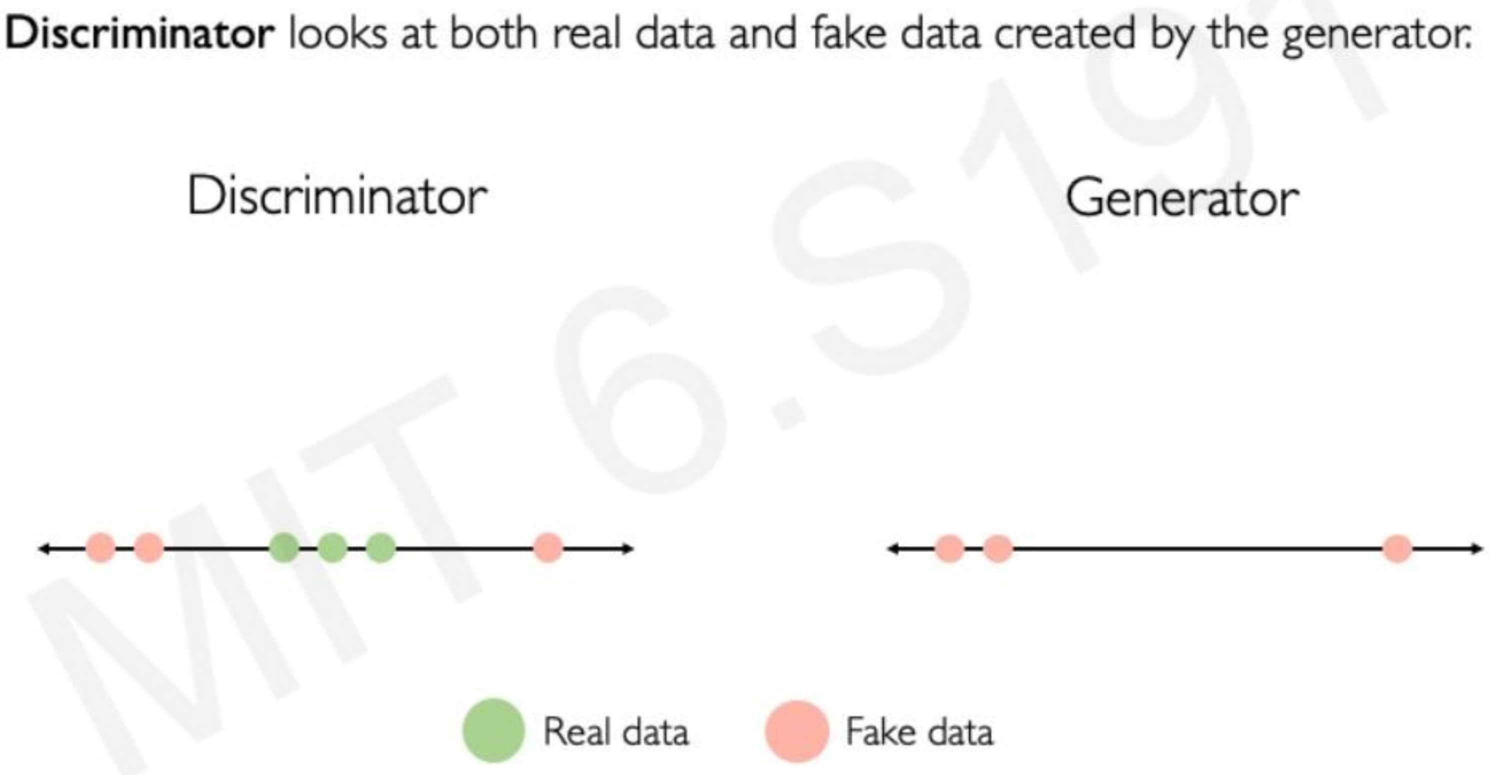
Generator starts from noise to try to create an imitation of the data.



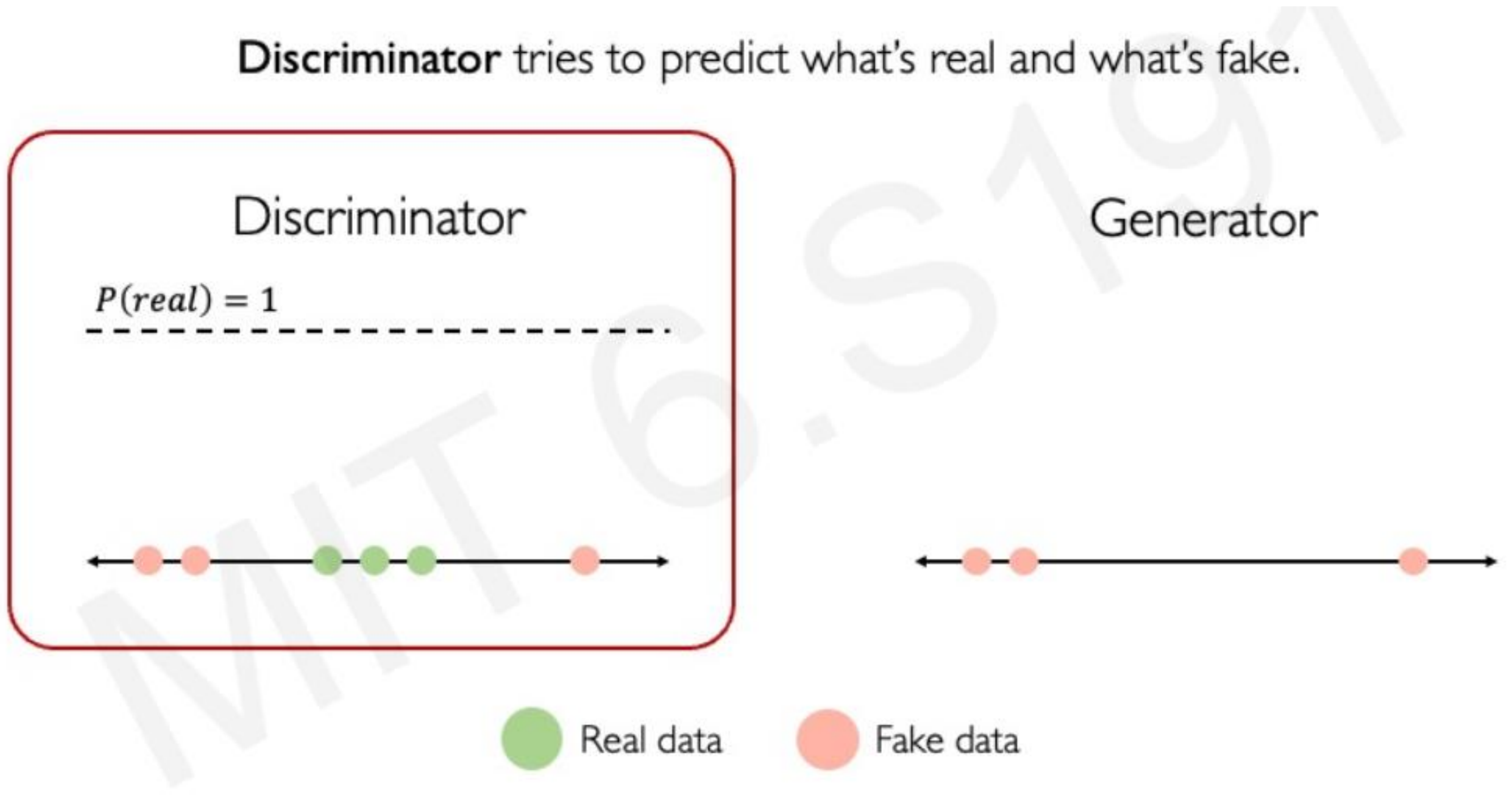
Generative Adversarial Networks



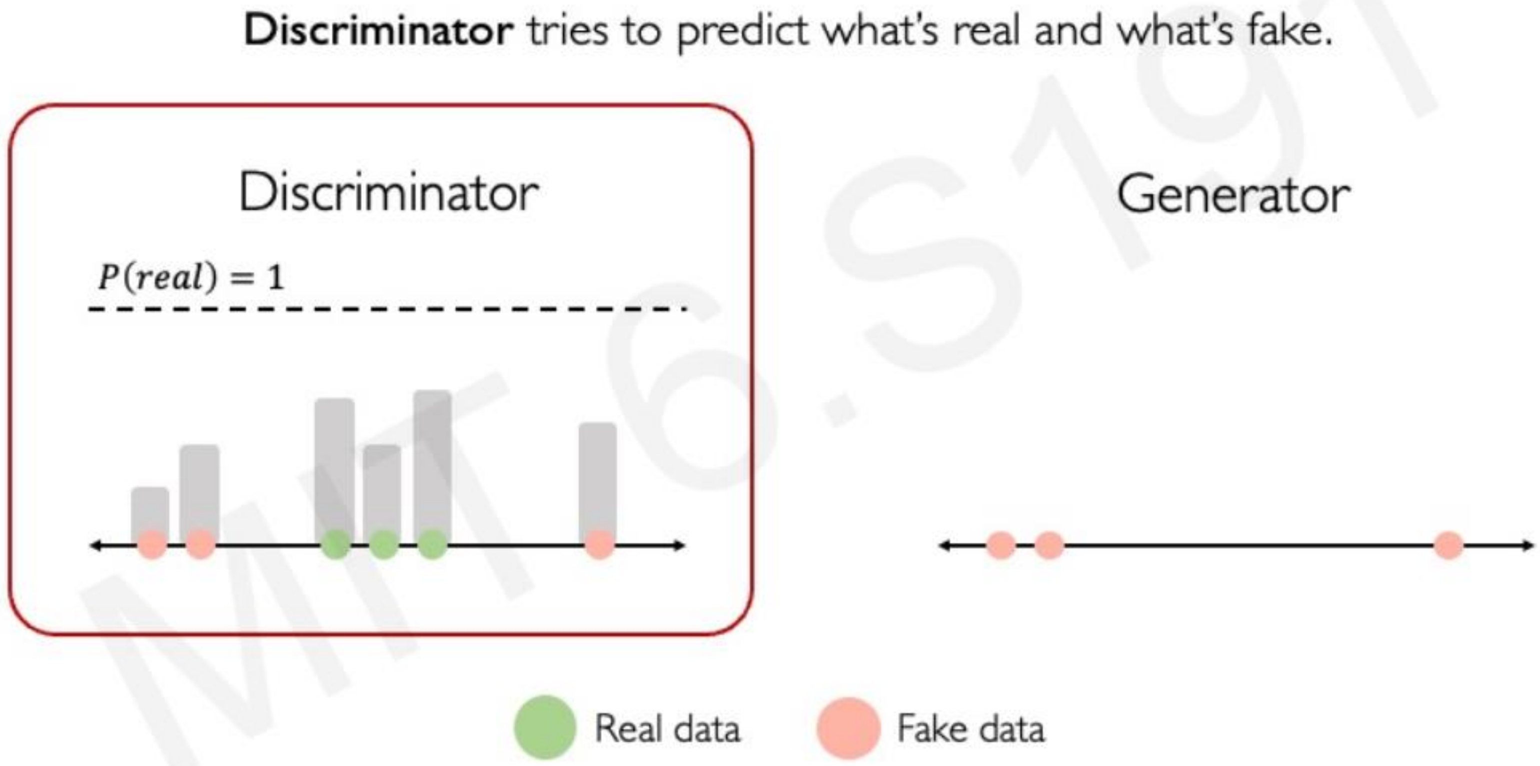
Generative Adversarial Networks



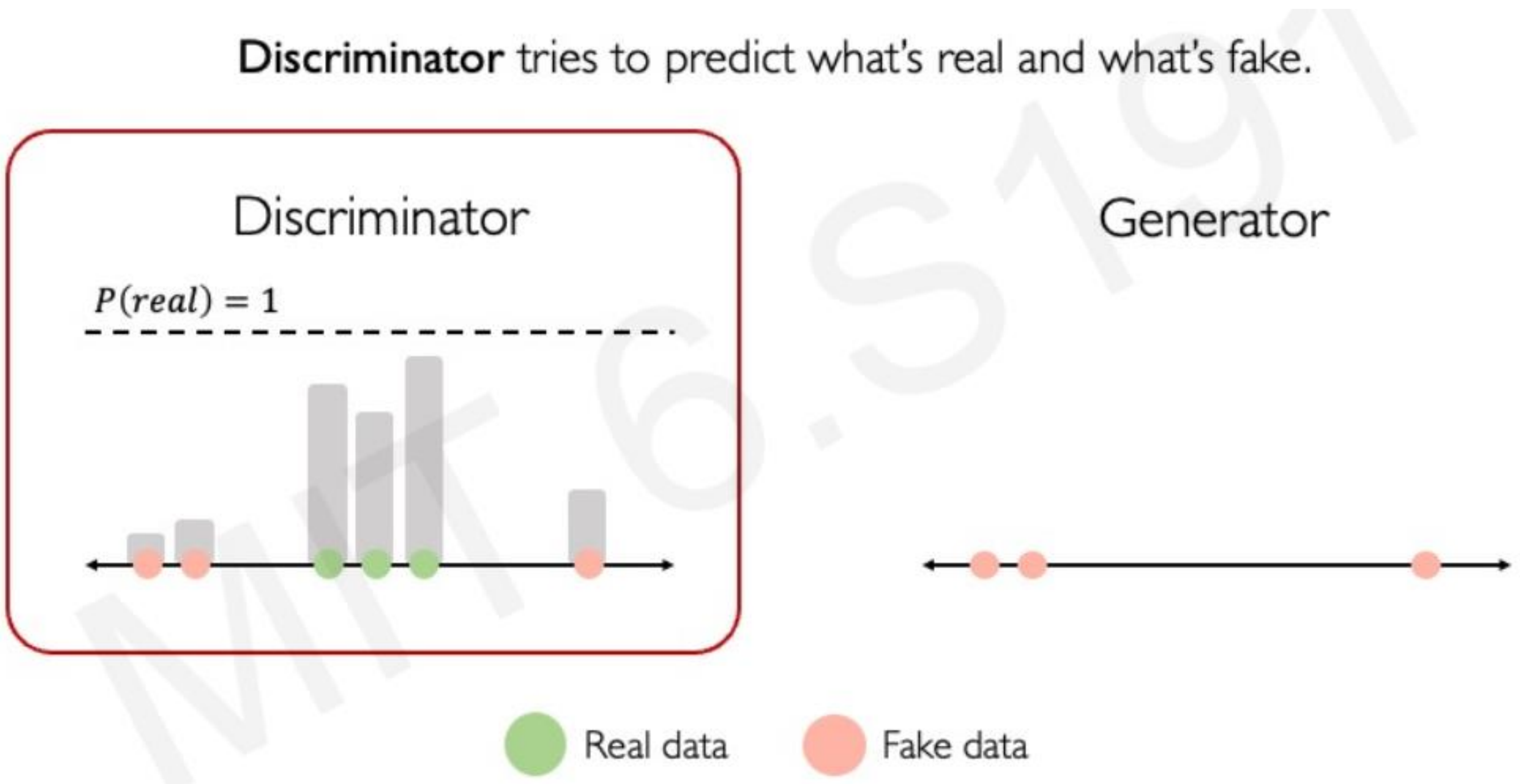
Generative Adversarial Networks



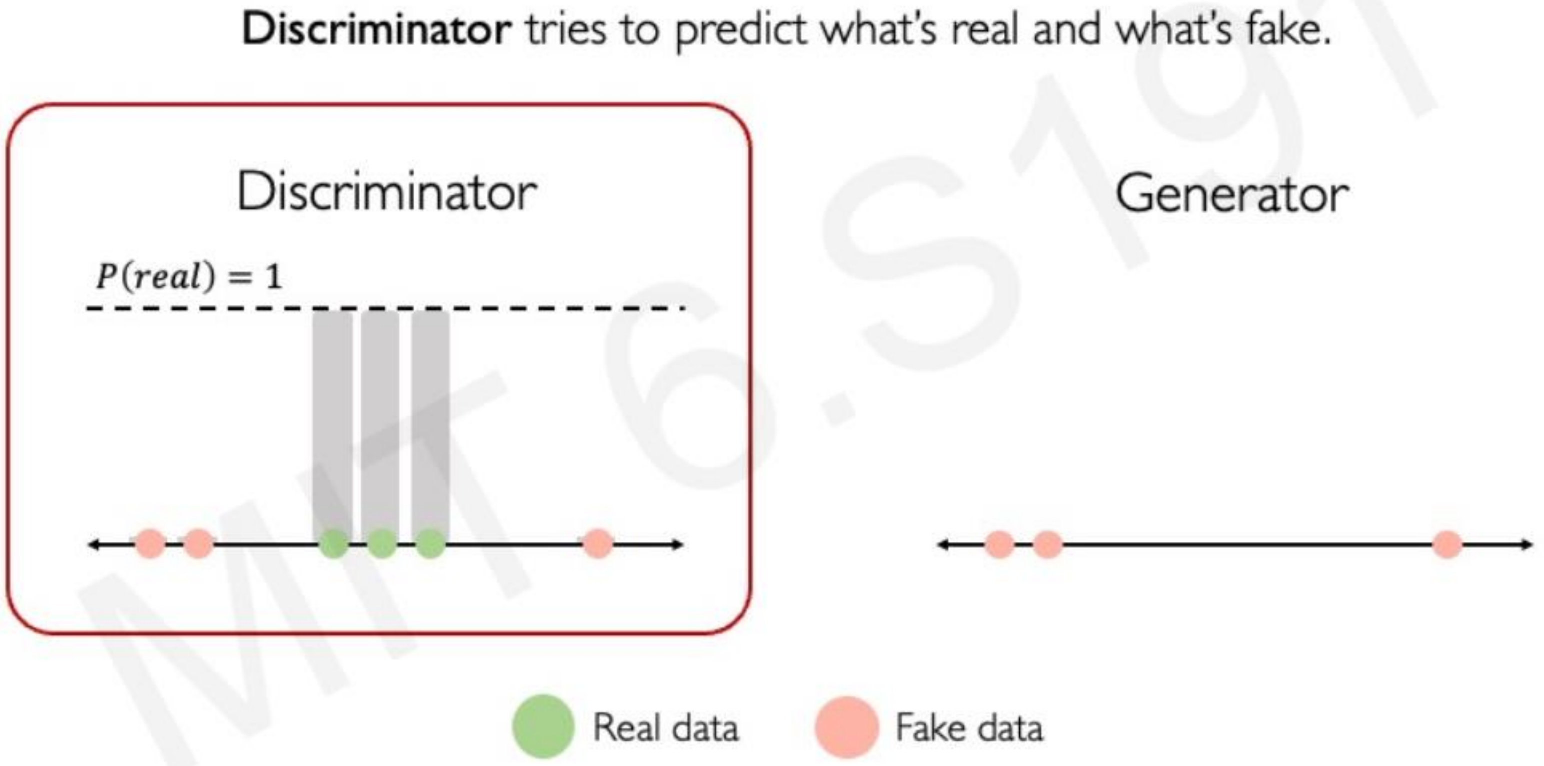
Generative Adversarial Networks



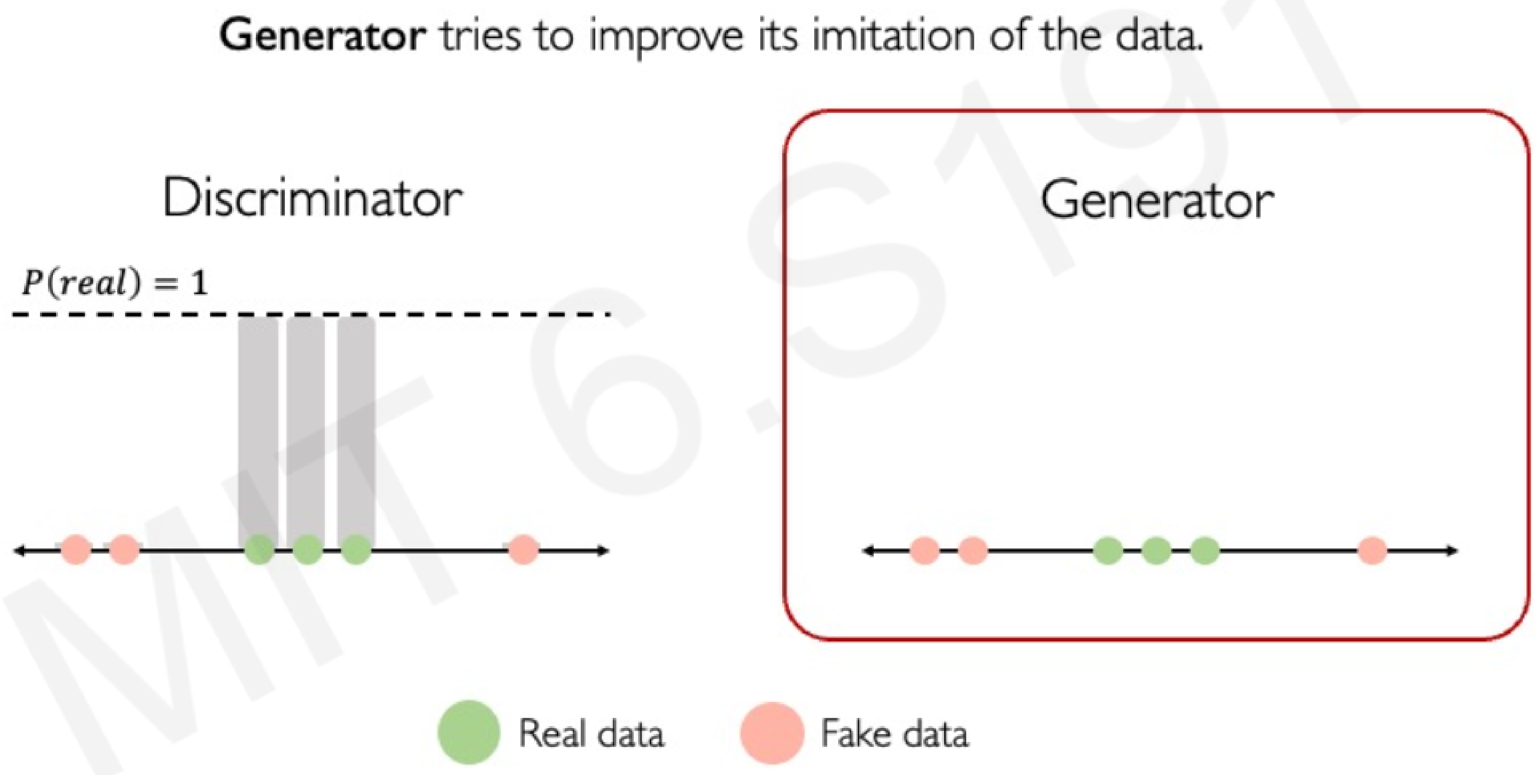
Generative Adversarial Networks



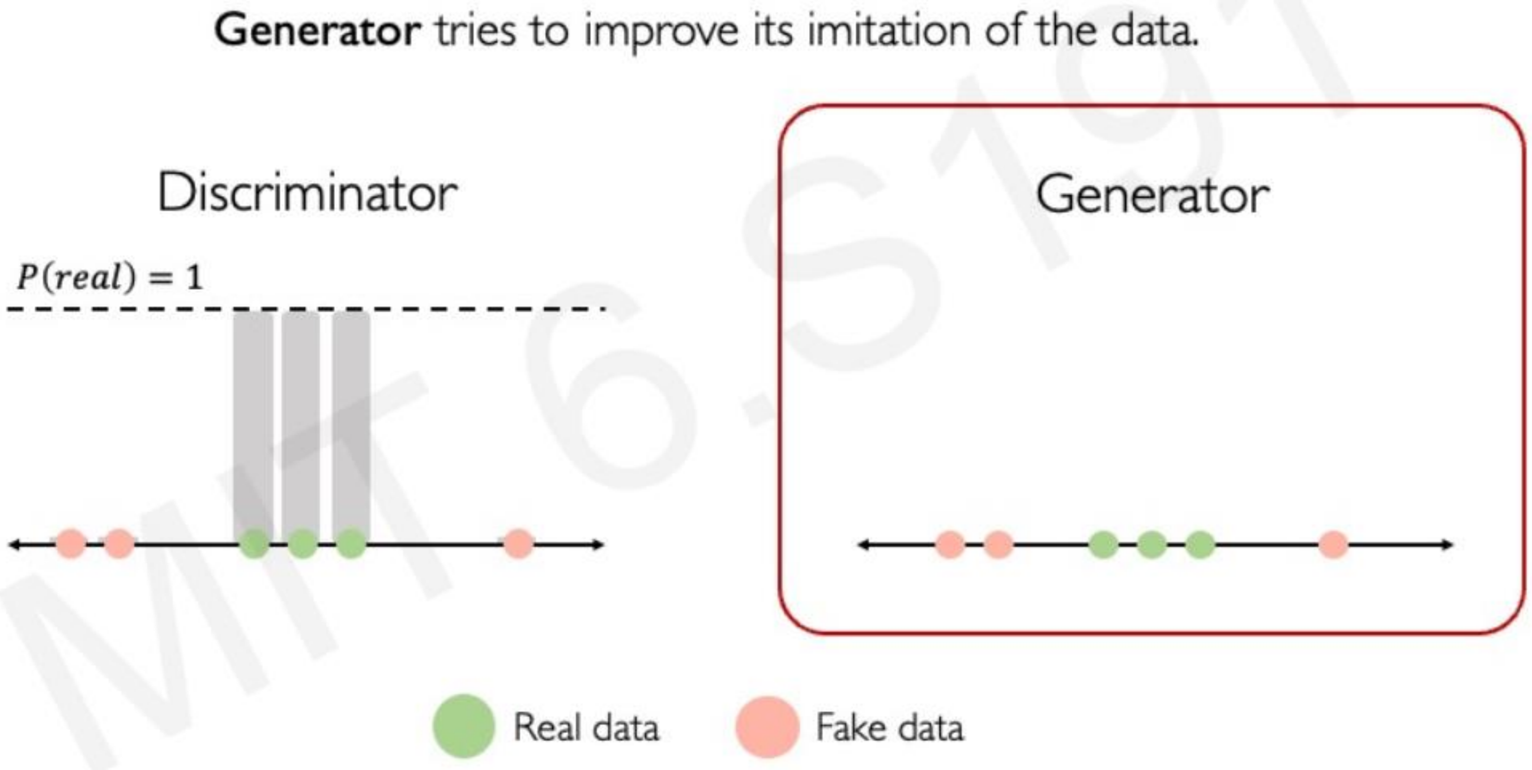
Generative Adversarial Networks



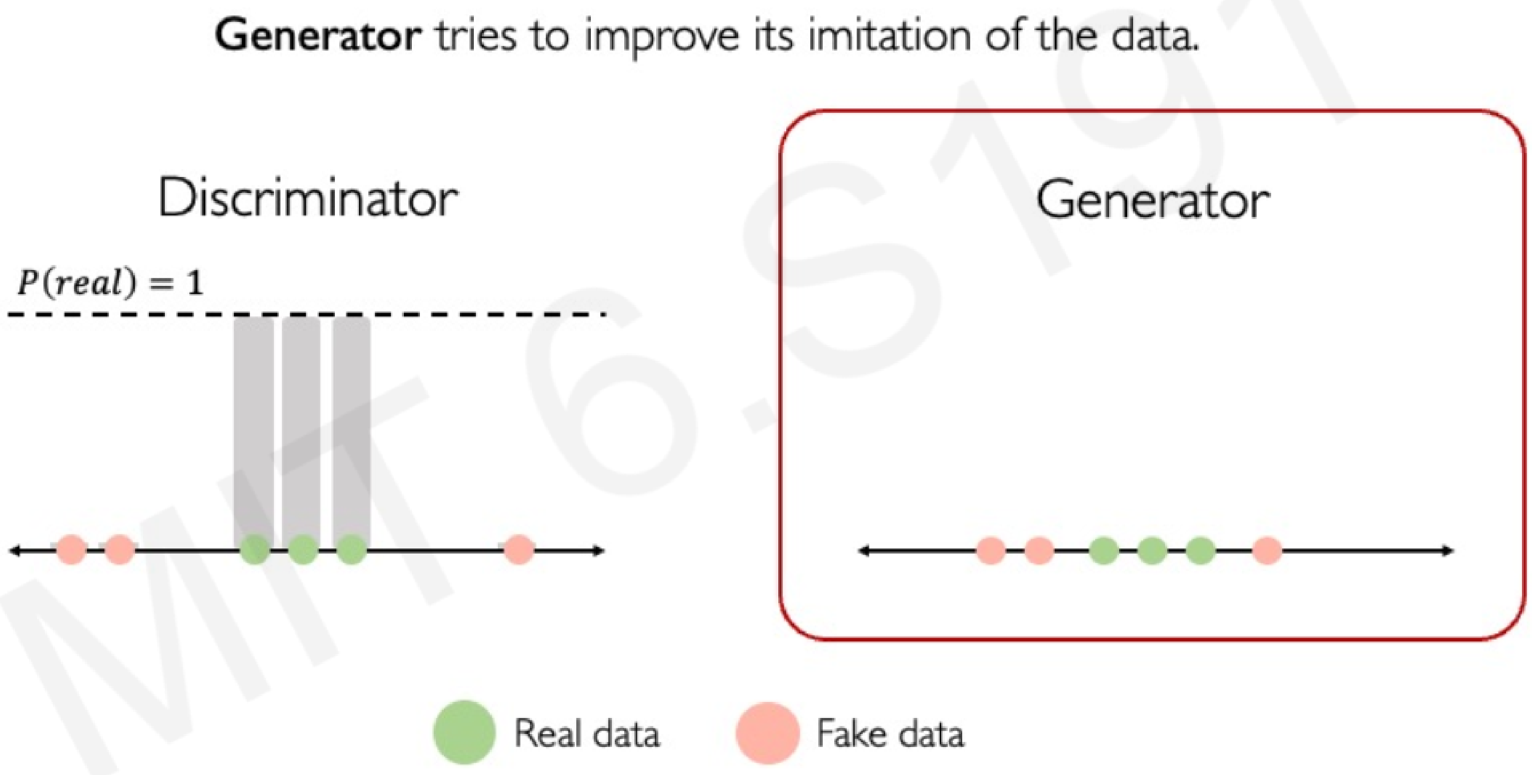
Generative Adversarial Networks



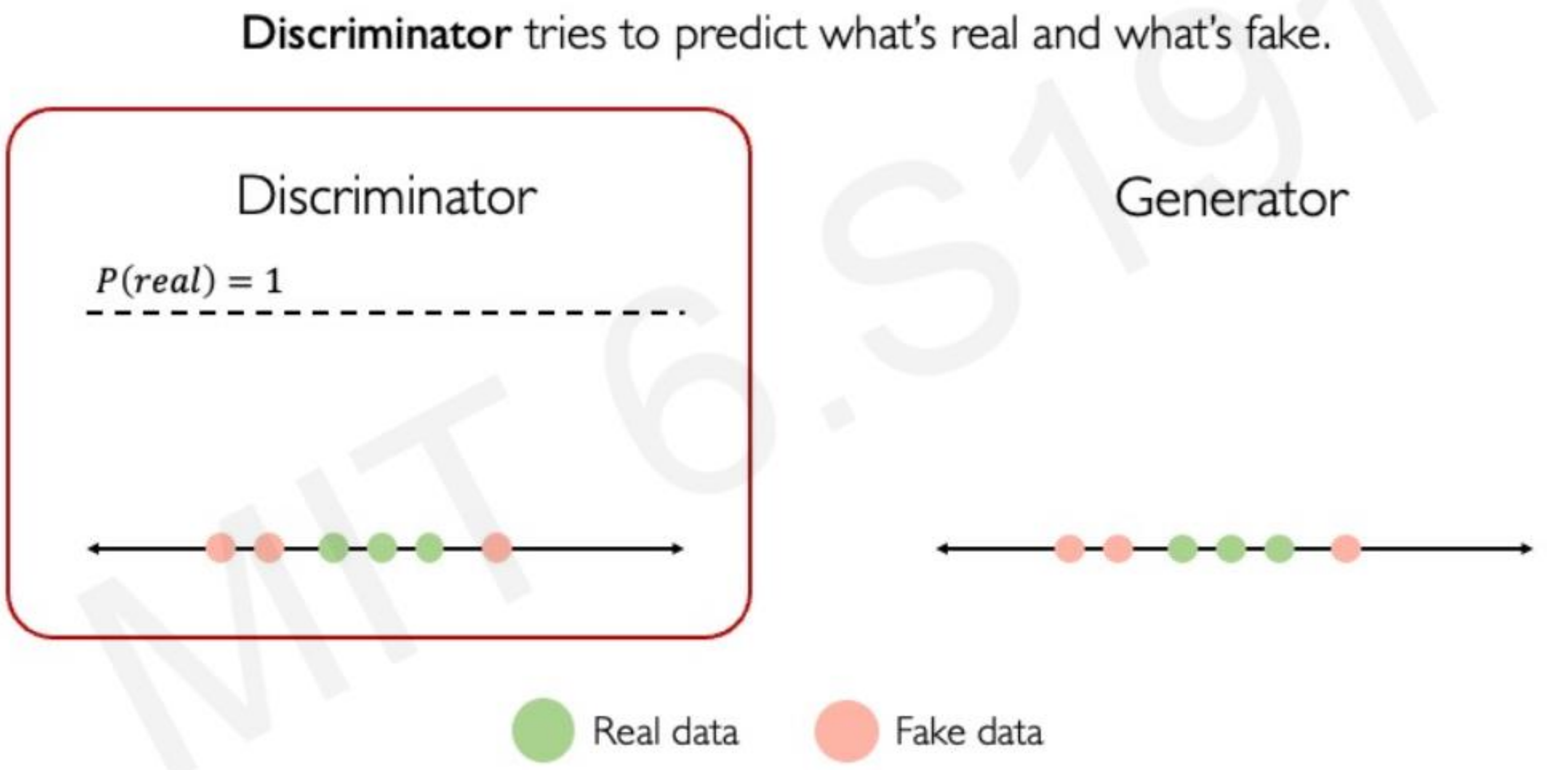
Generative Adversarial Networks



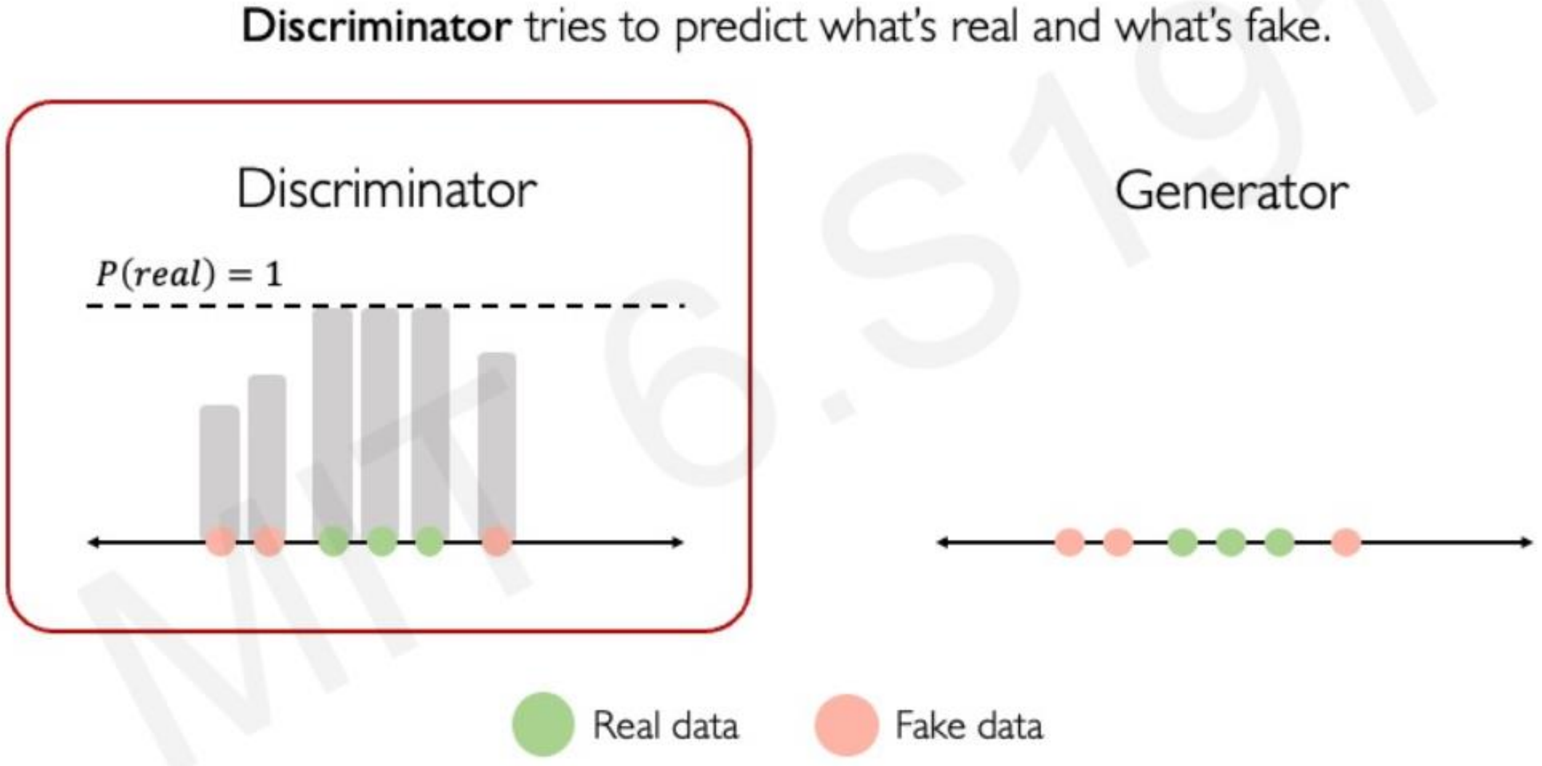
Generative Adversarial Networks



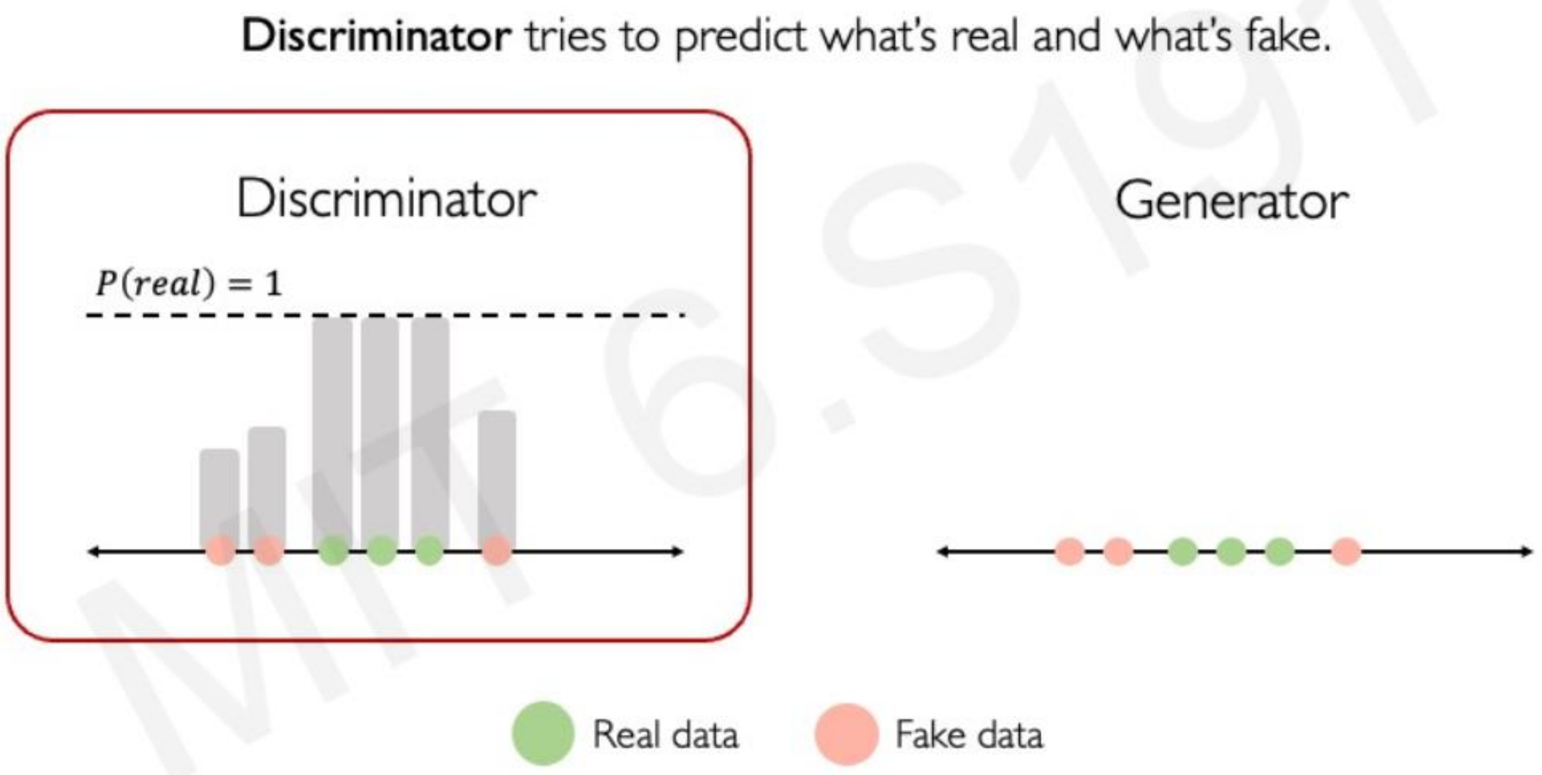
Generative Adversarial Networks



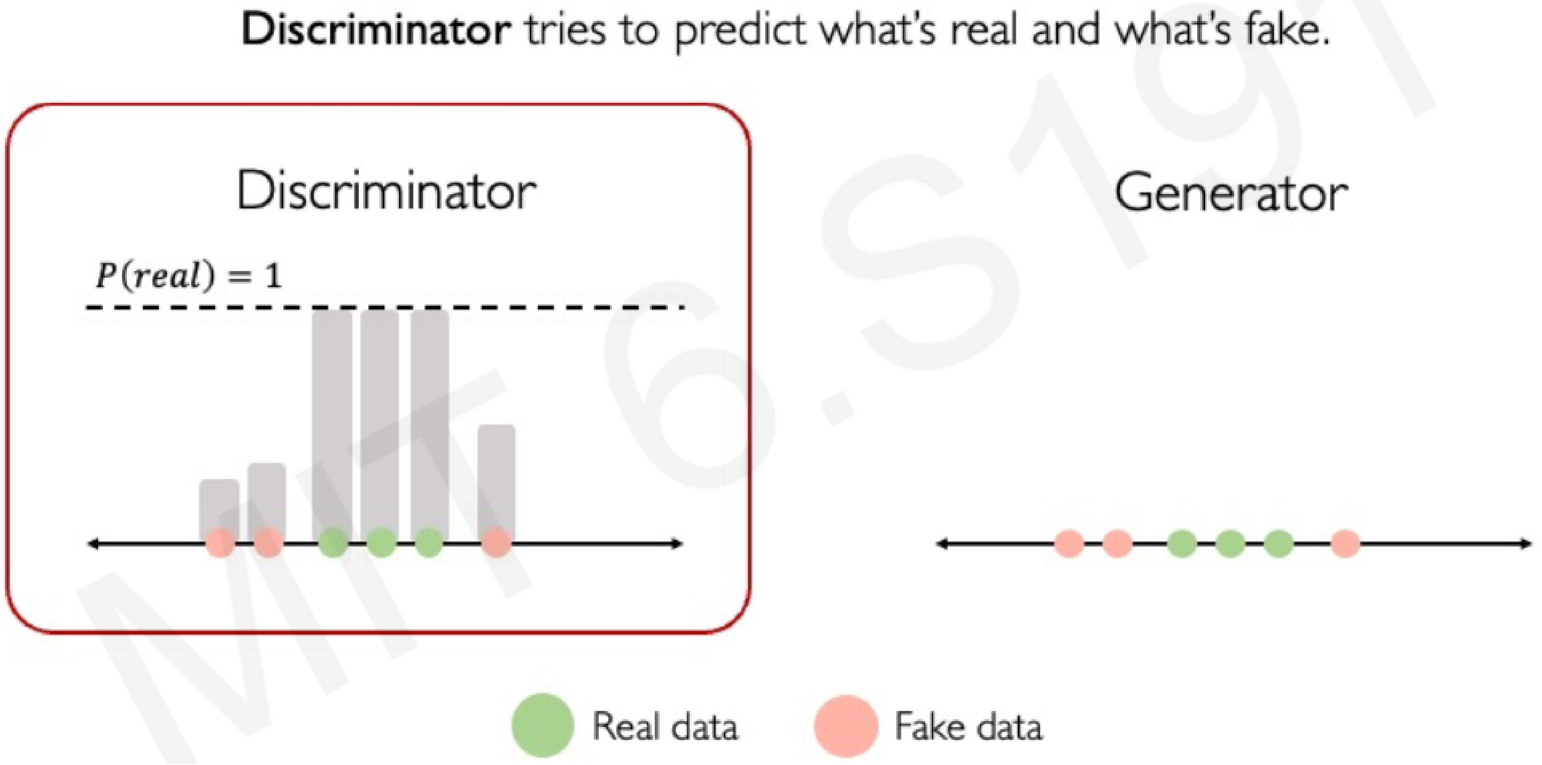
Generative Adversarial Networks



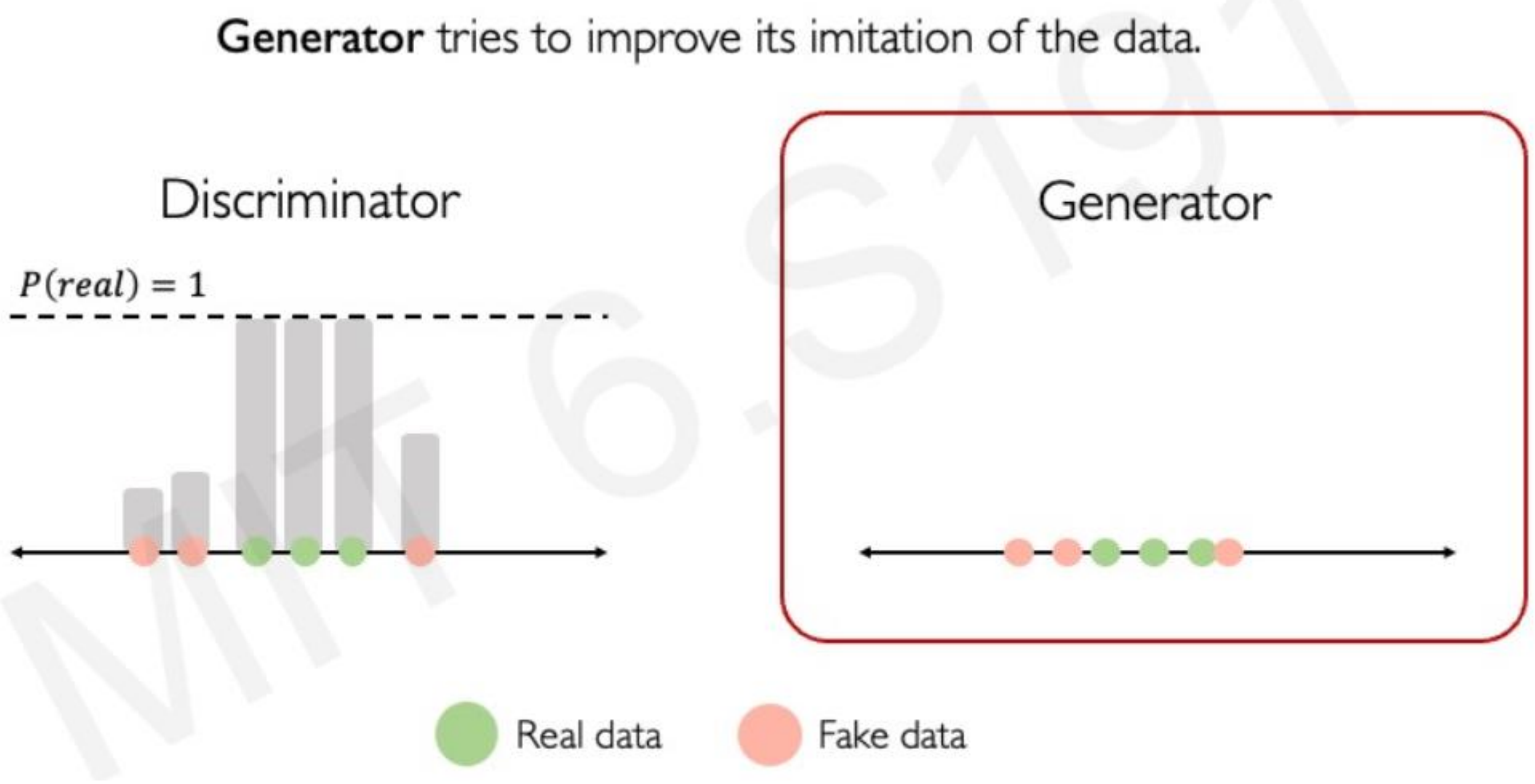
Generative Adversarial Networks



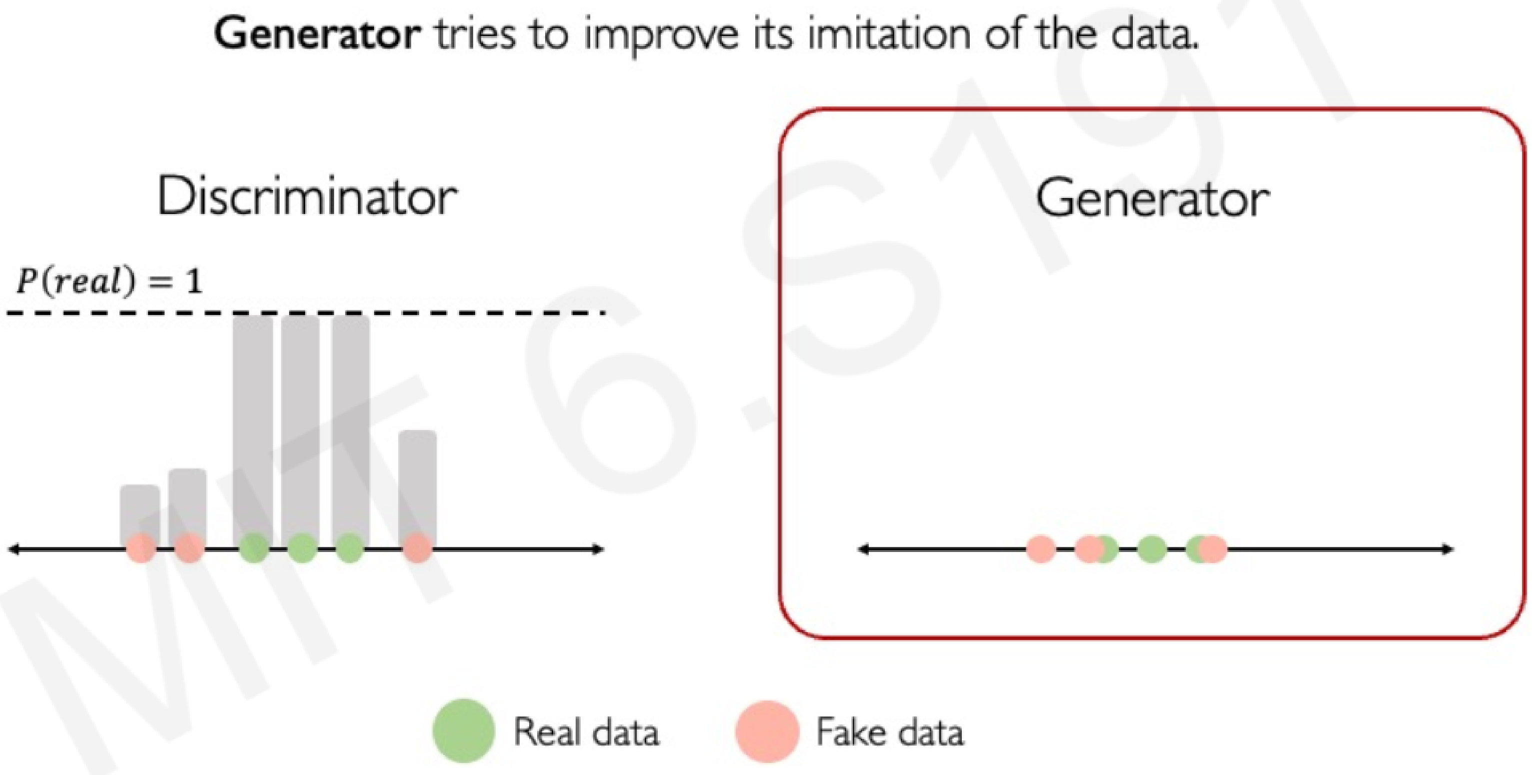
Generative Adversarial Networks



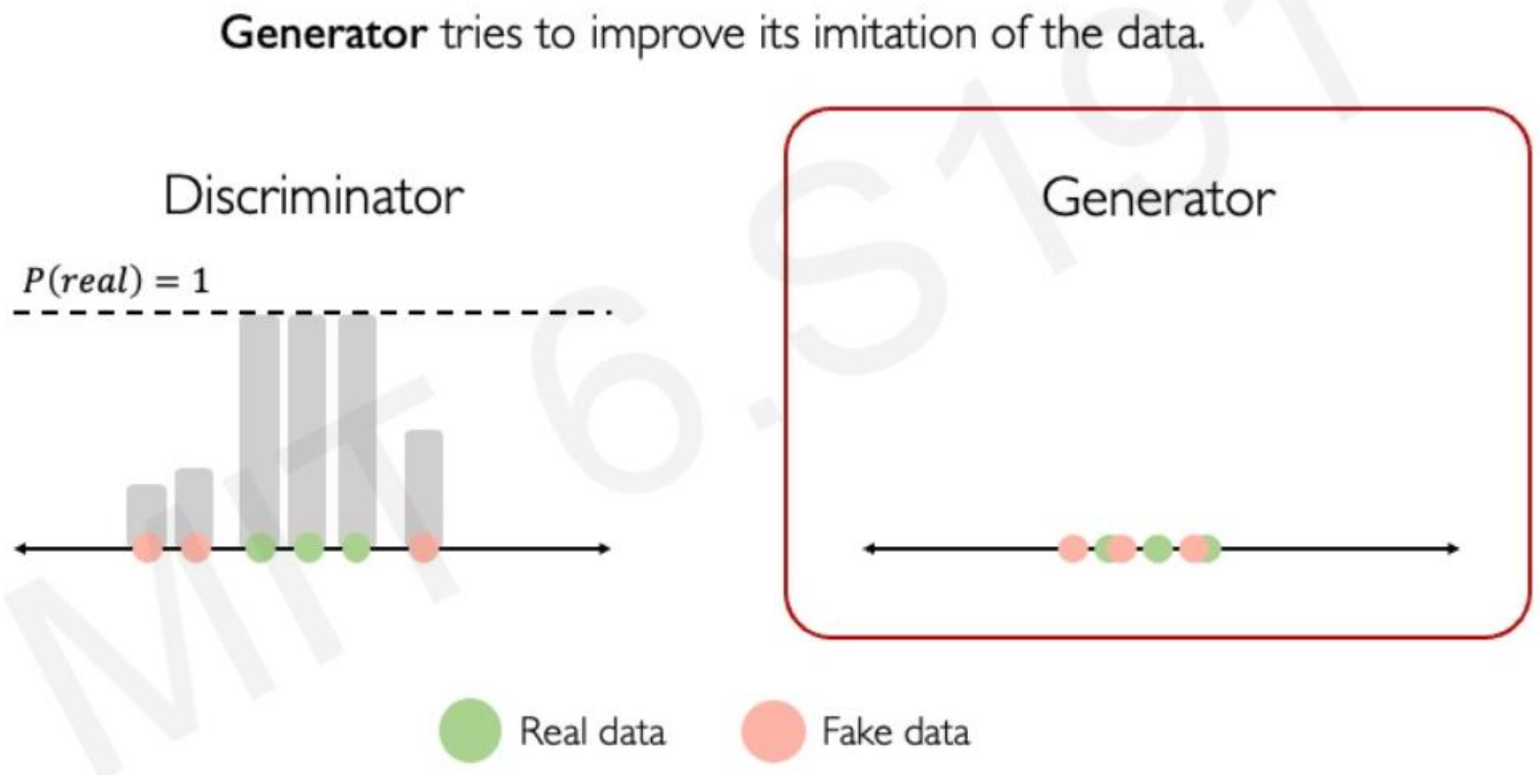
Generative Adversarial Networks



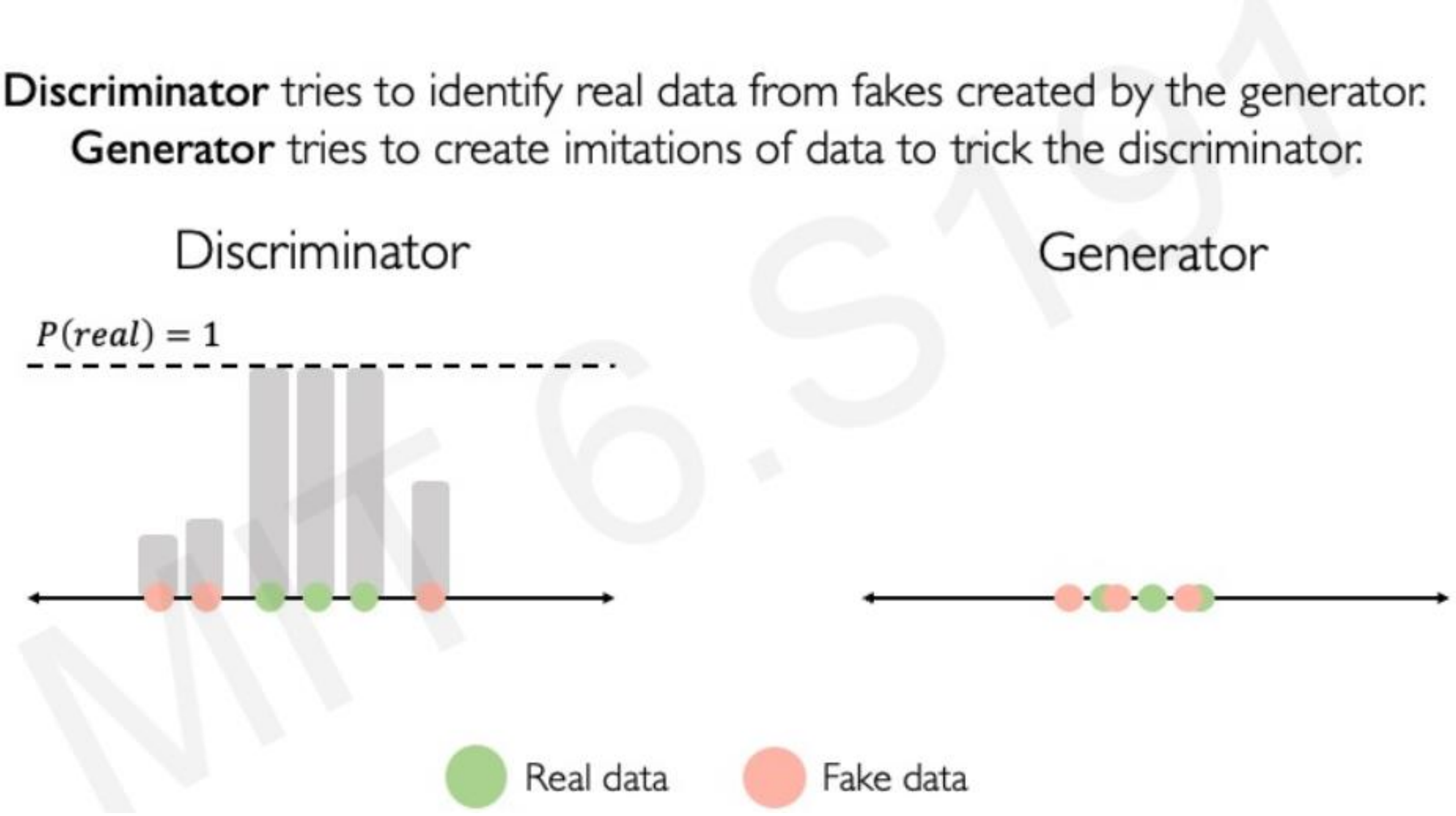
Generative Adversarial Networks



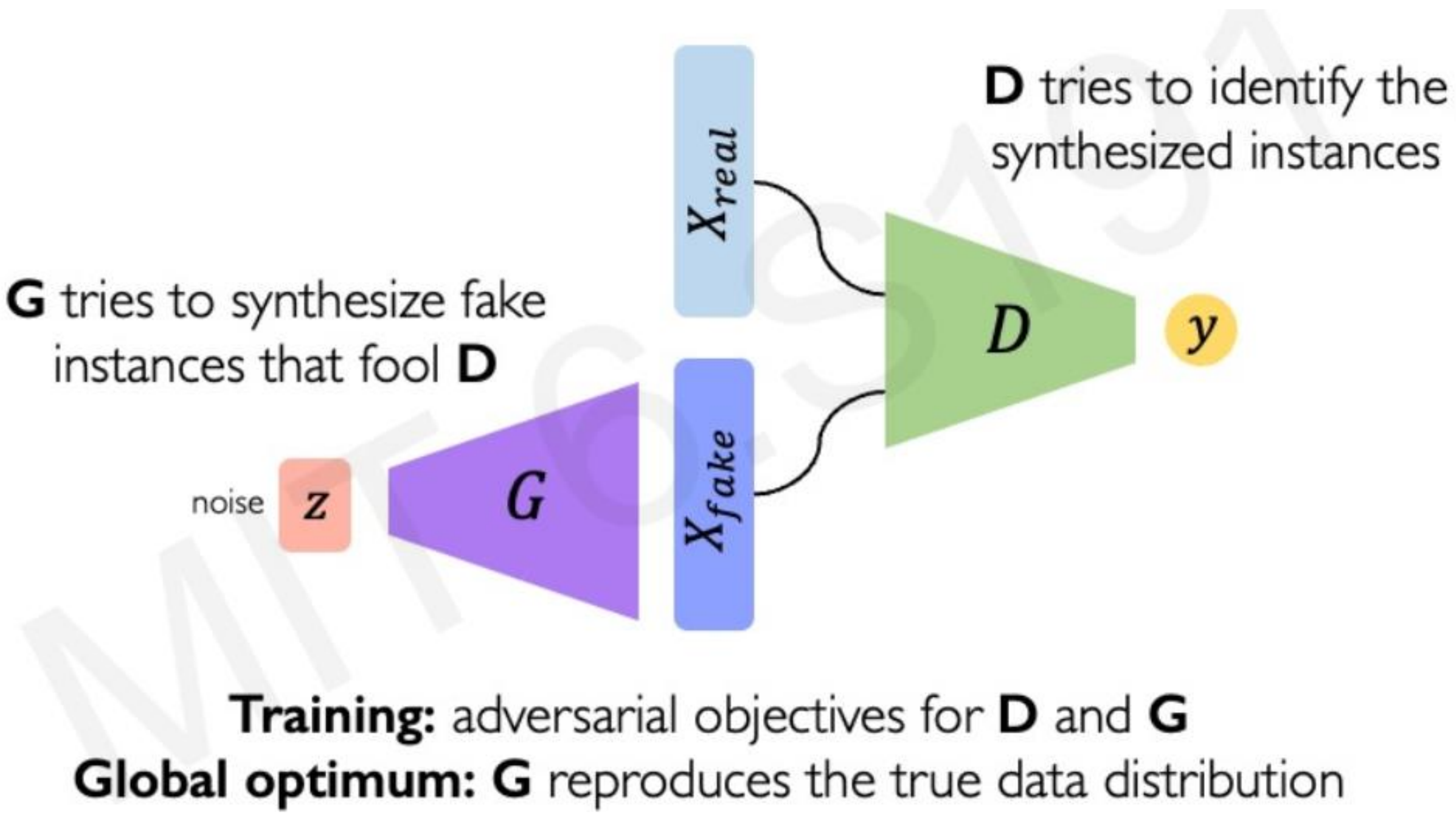
Generative Adversarial Networks



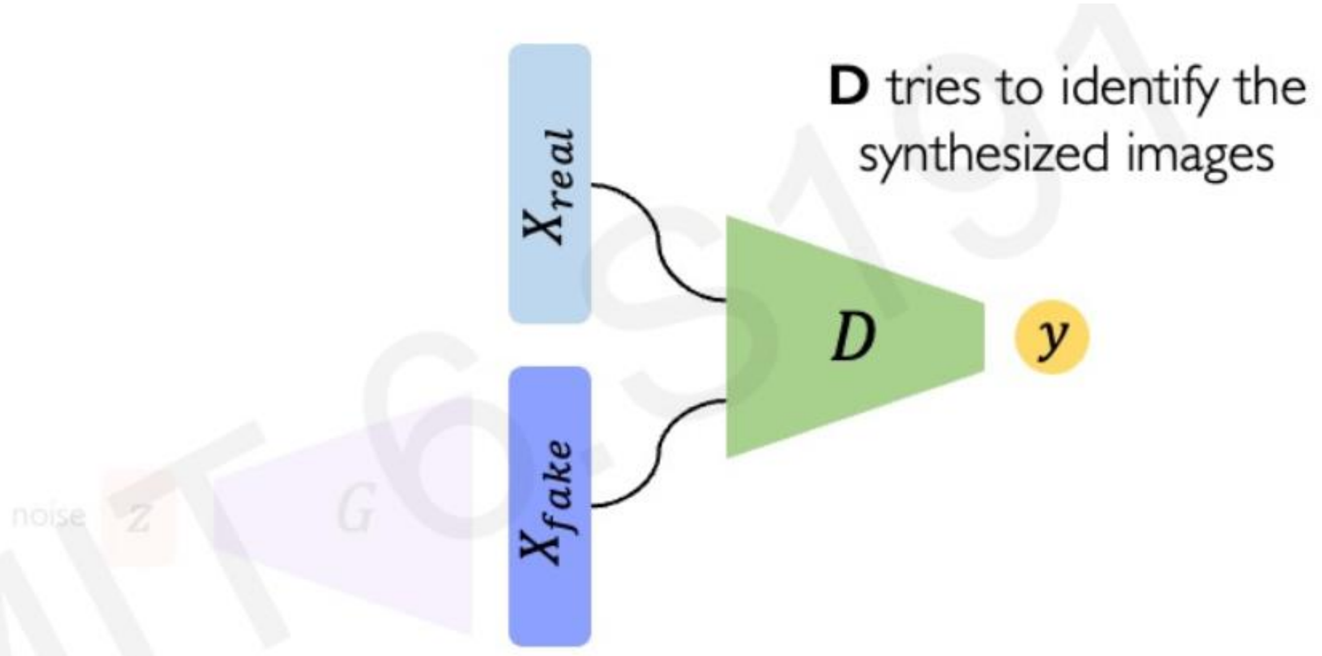
Generative Adversarial Networks



Generative Adversarial Networks

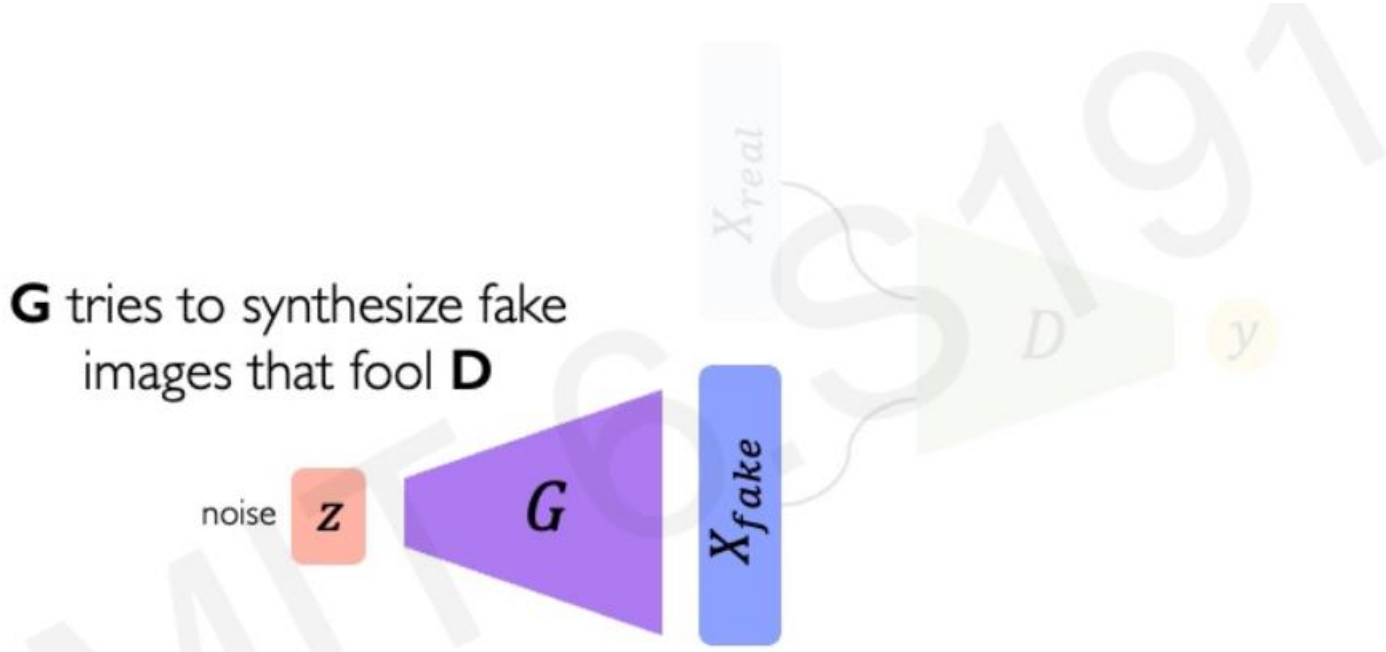


Generative Adversarial Networks



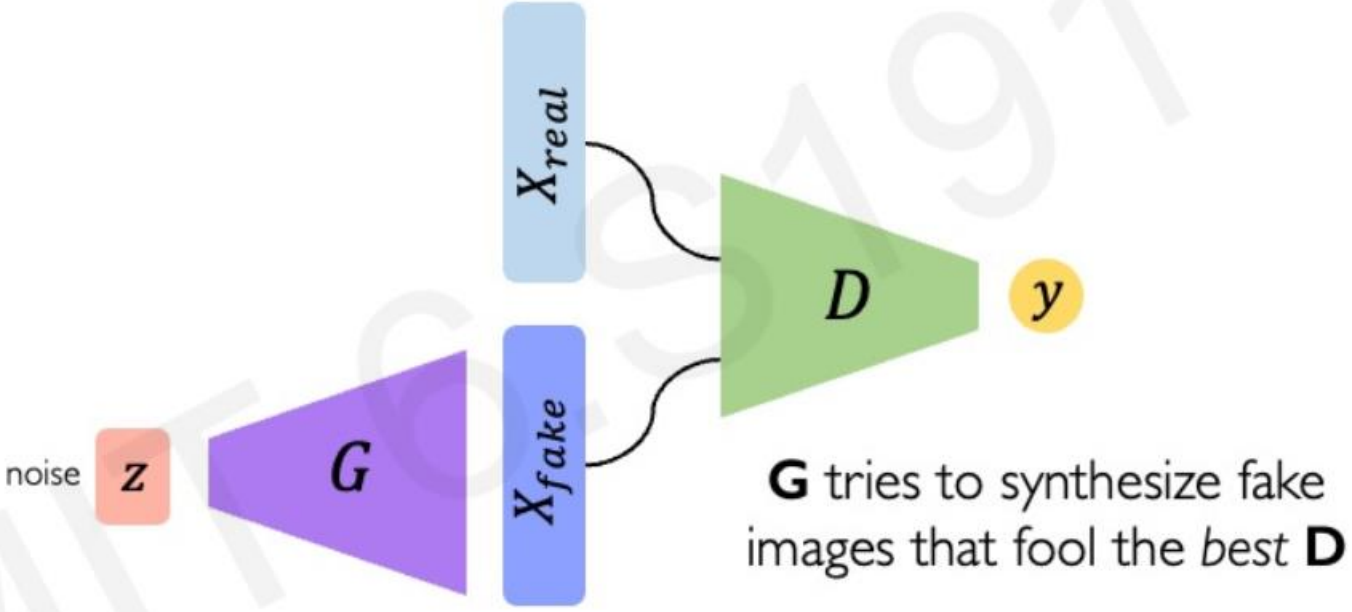
$$\underbrace{\arg\max_{D}}\ \mathbb{E}_{\mathbf{z},\mathbf{x}}\Big[\ \underbrace{\log D(G(\mathbf{z}))}_{\textbf{Fake}} + \underbrace{\log(1 - D(\mathbf{x}))}_{\textbf{Real}}\ \Big]$$

Generative Adversarial Networks



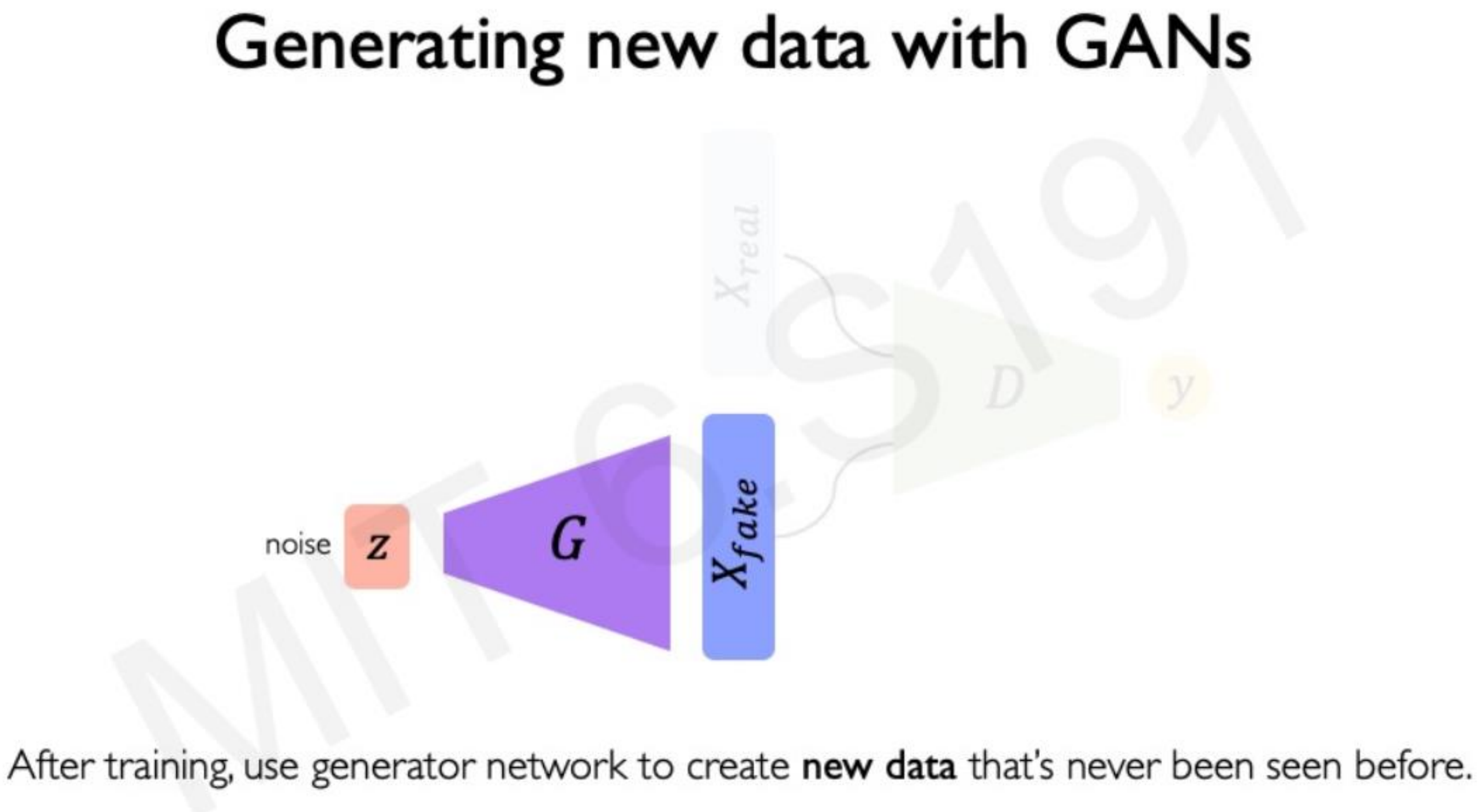
$$\arg\min_{G} \; \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\, \log D(G(\mathbf{z})) + \log\left(1 - D(\mathbf{x})\right) \,\right]$$

Generative Adversarial Networks

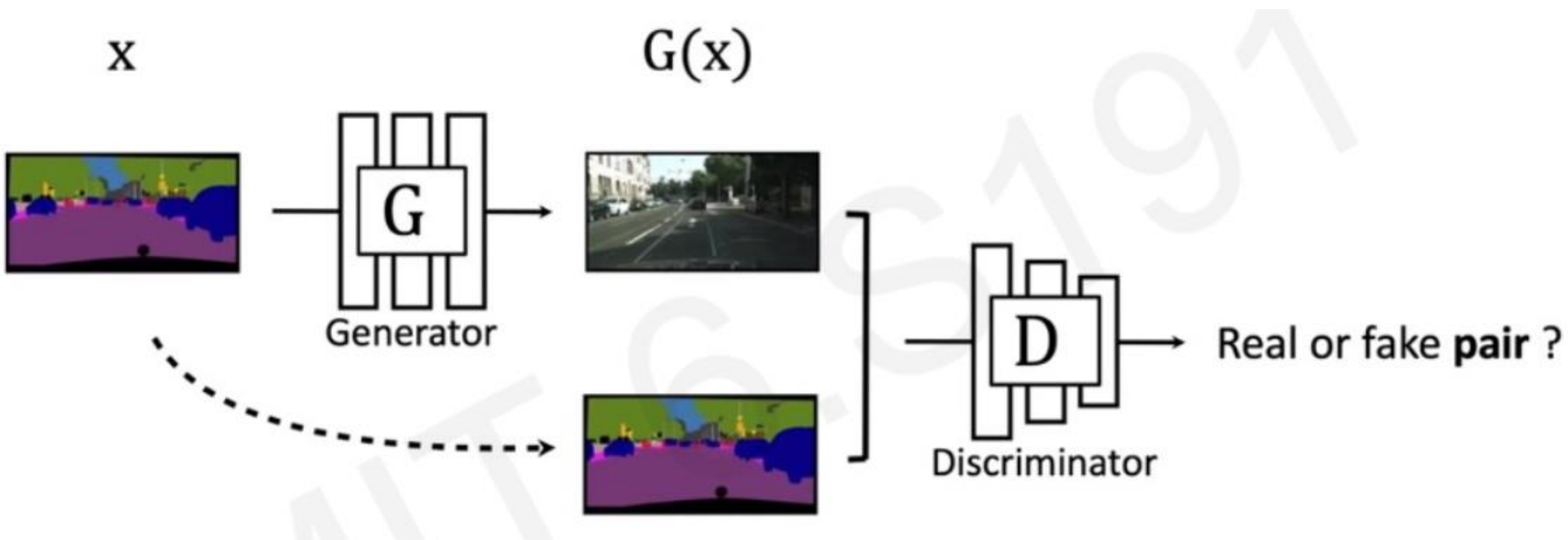


$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{z},\mathbf{x}}[\ \log D(G(\mathbf{z})) + \log(1 - D(\mathbf{x}))\ ]$$

Generation

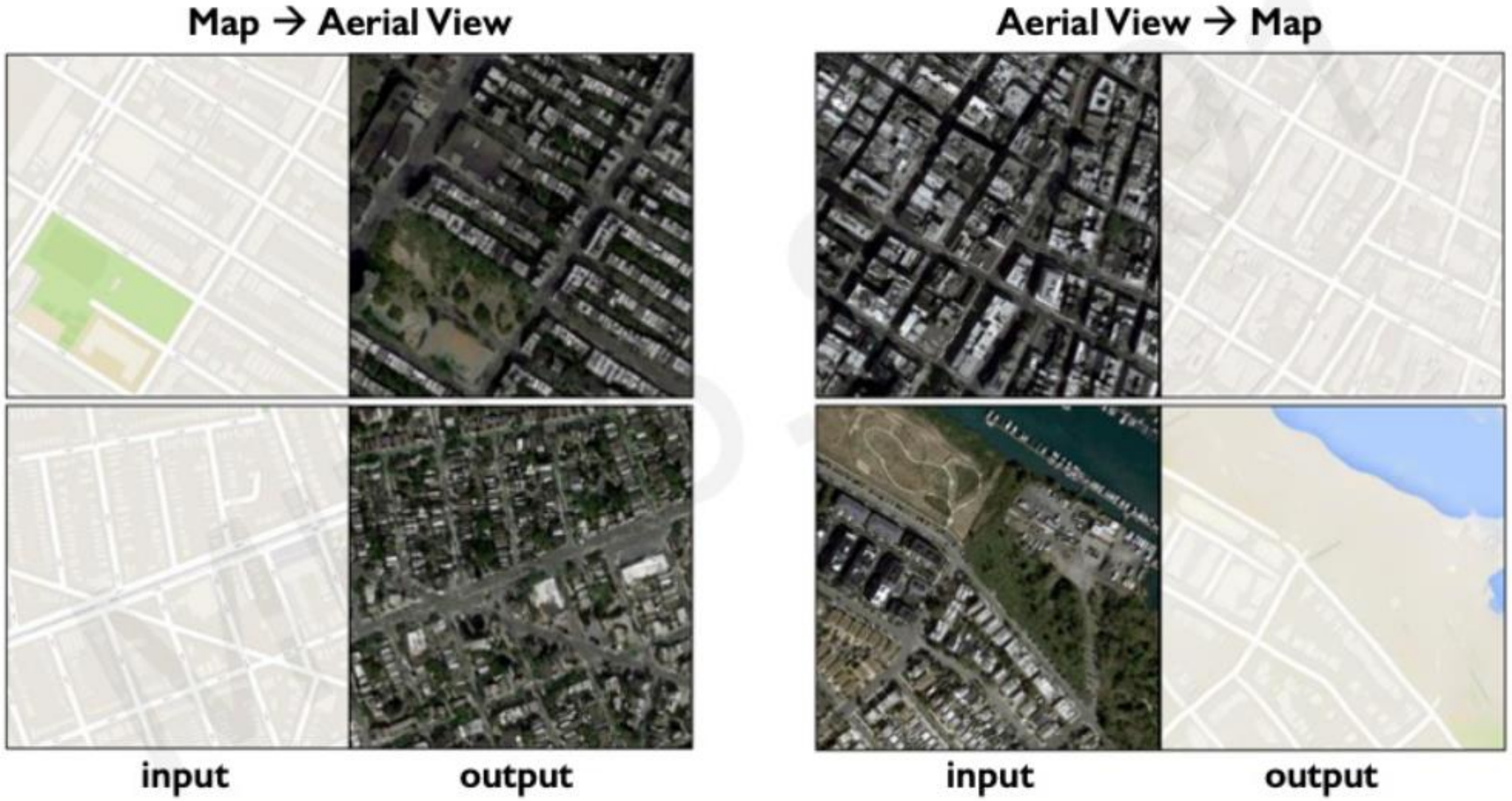

Generating new data with GANs
noise
z
G
X_{fake}
X_{real}
D
y
After training, use generator network to create **new data** that's never been seen before.

Applications

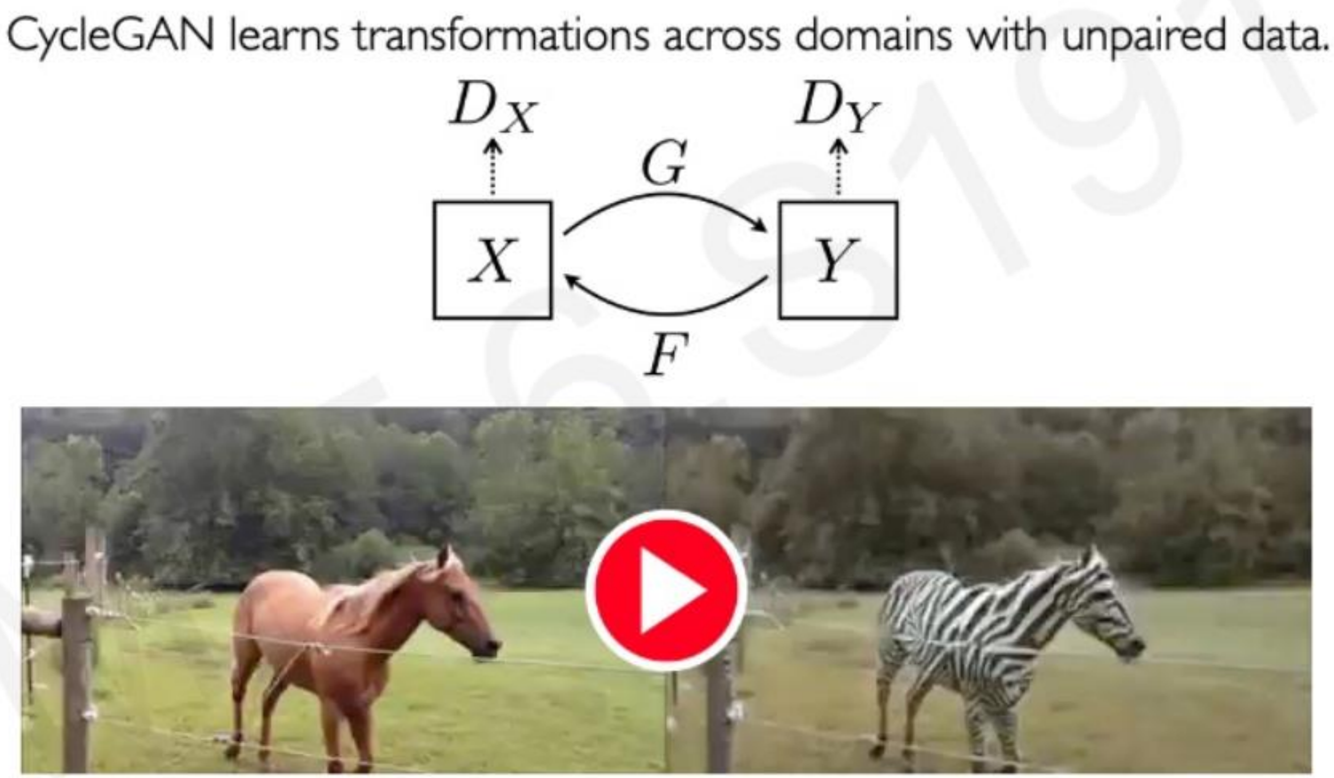


The discriminator, D, classifies between fake and real **pairs**.
The generator, G, learns to fool the discriminator.

Applications

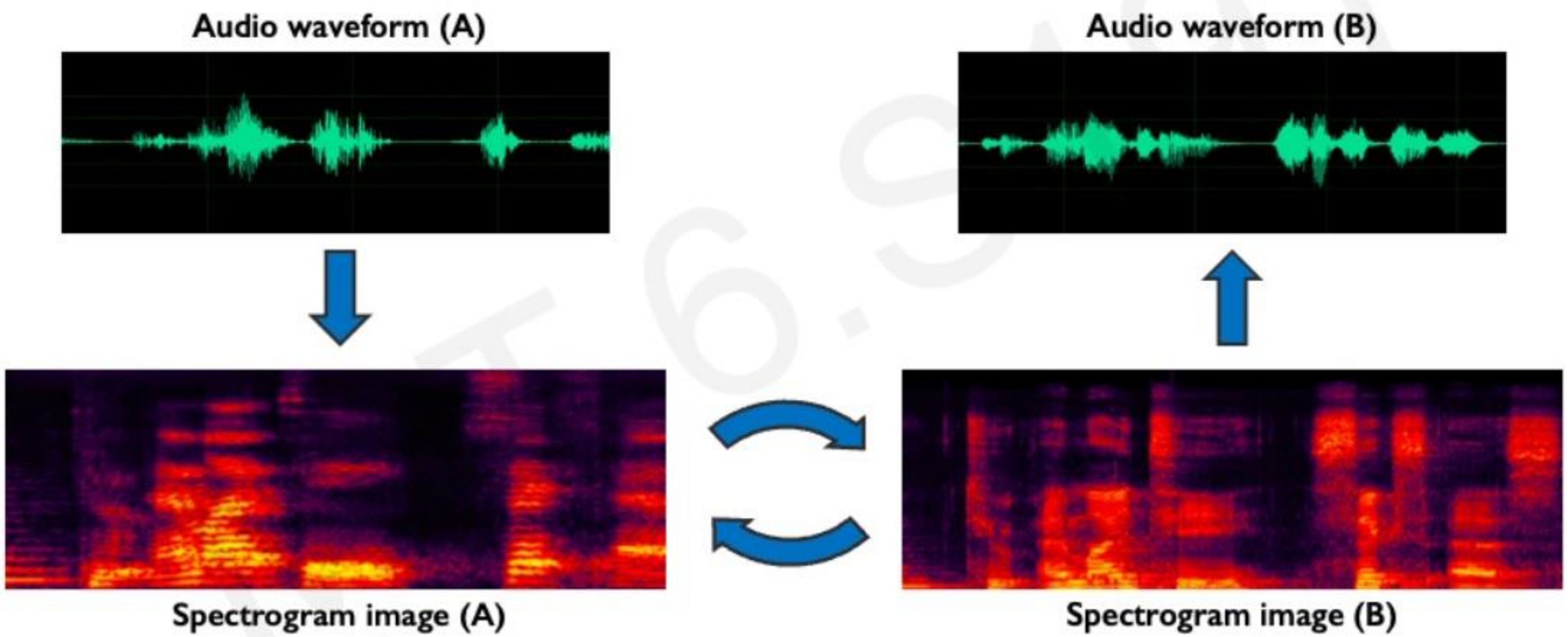


Applications

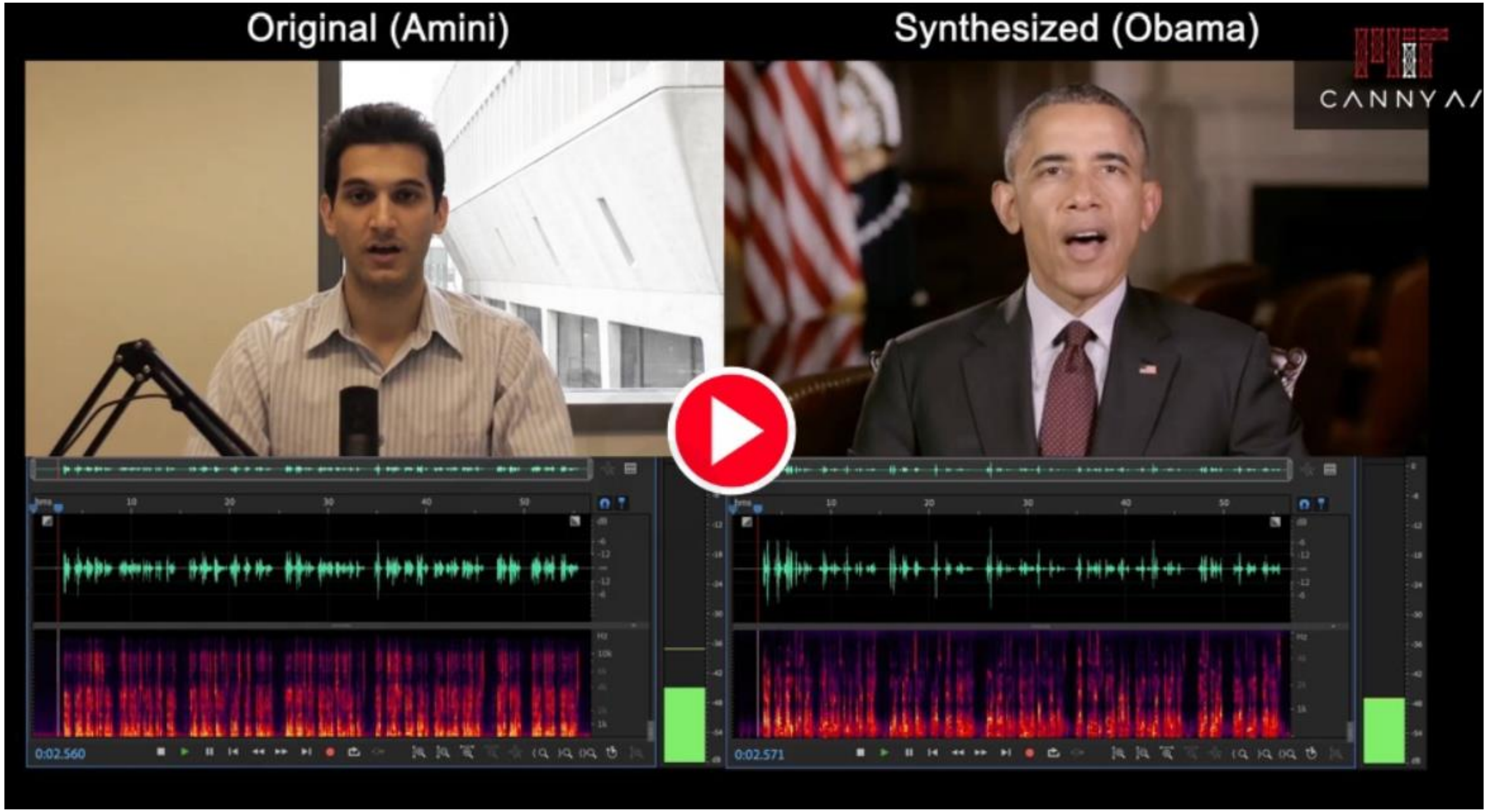


https://www.youtube.com/watch?v=9reHvktowLY

Applications



Applications



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

Thank you!!!