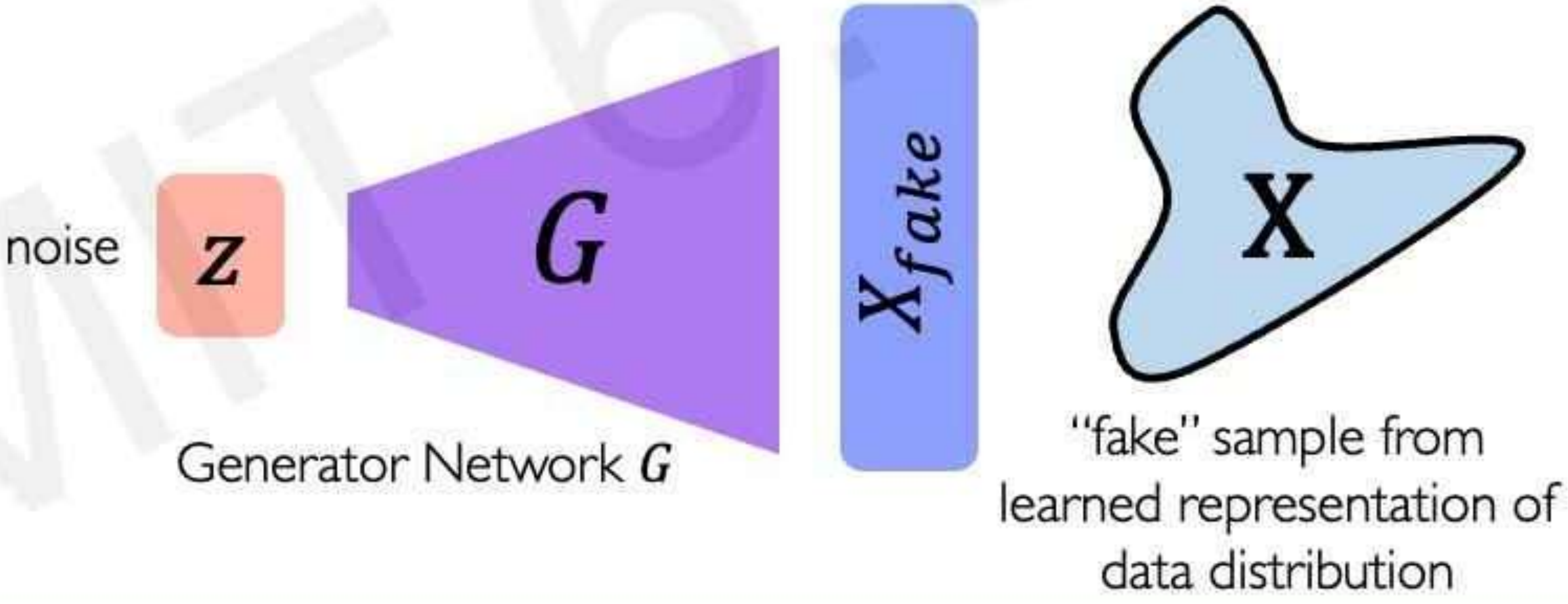
Generative Adversarial Networks (GANs)

What if we just want to sample?

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.

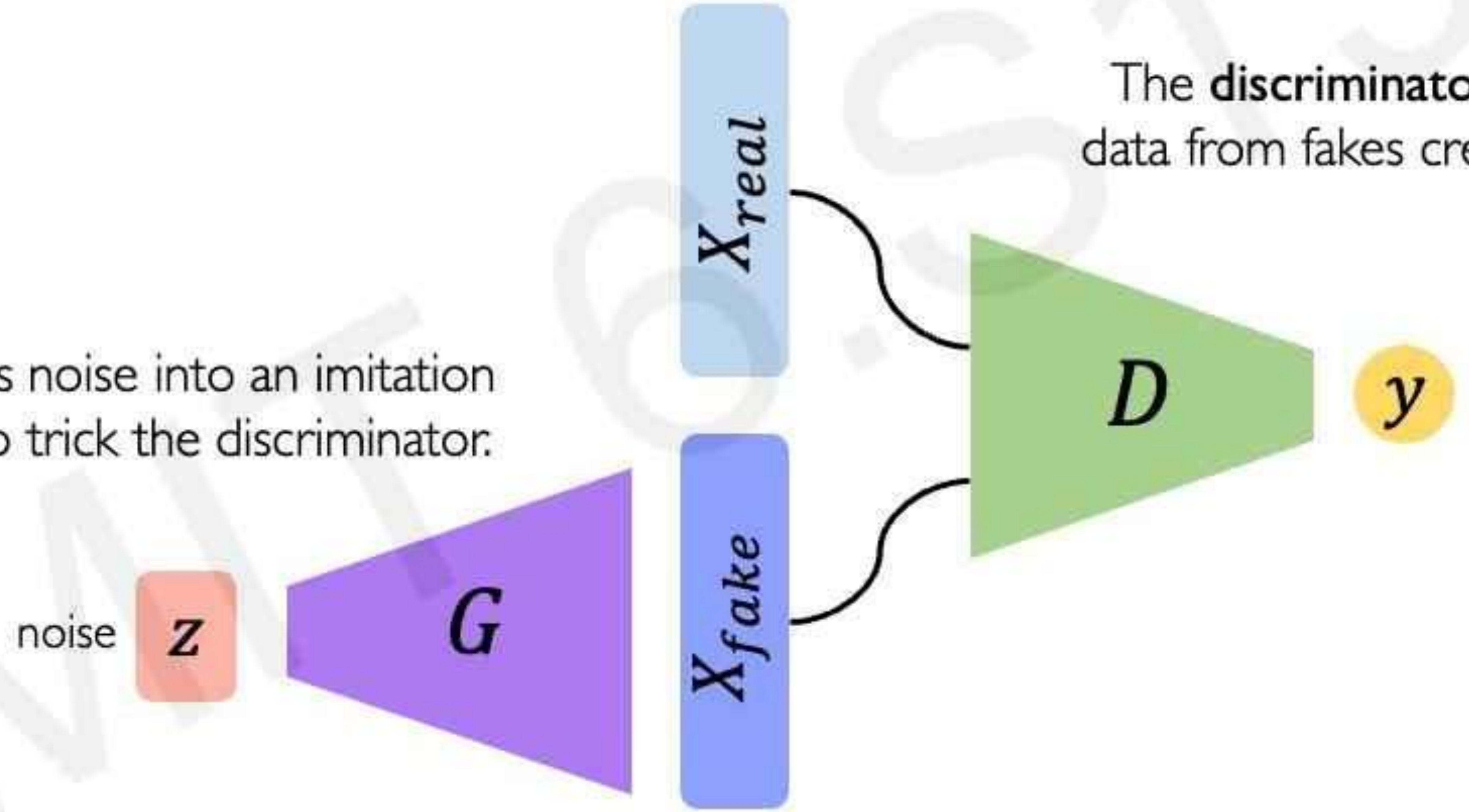


Generative Adversarial Networks (GANs)

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

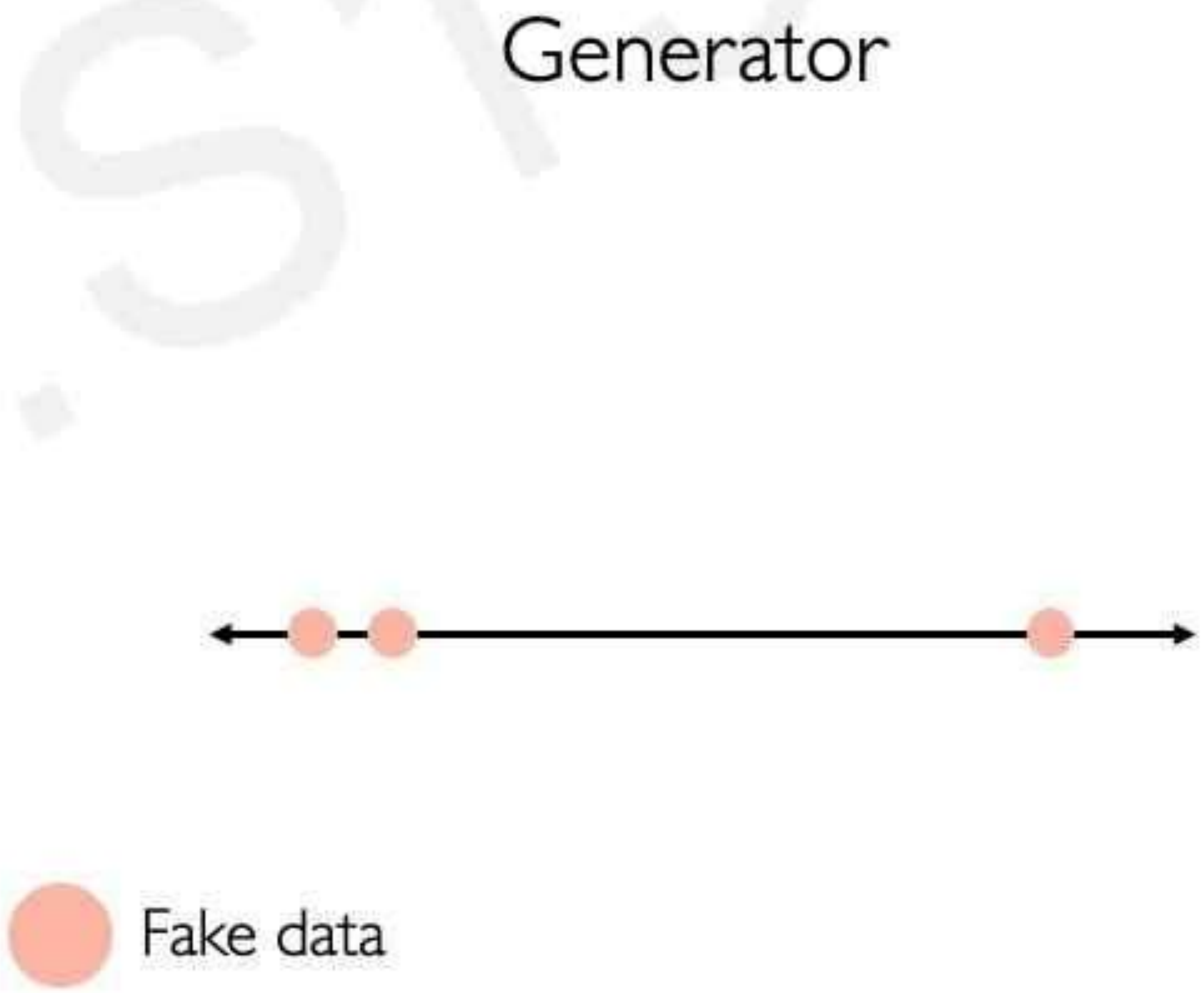
The **discriminator** tries to identify real data from fakes created by the generator.

The **generator** turns noise into an imitation of the data to try to trick the discriminator.



Intuition behind GANs

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



Intuition behind GANs

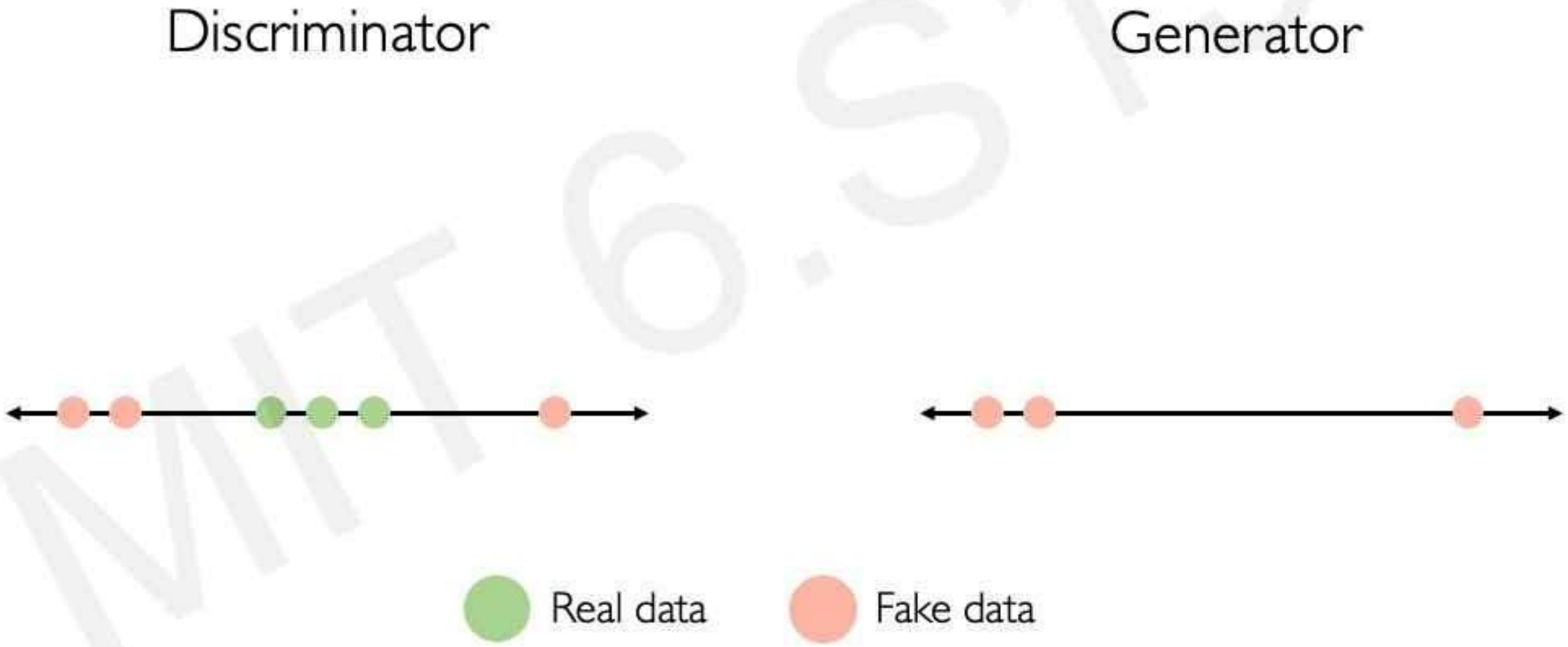
Discriminator looks at both real data and fake data created by the generator.

Discriminator

Generator

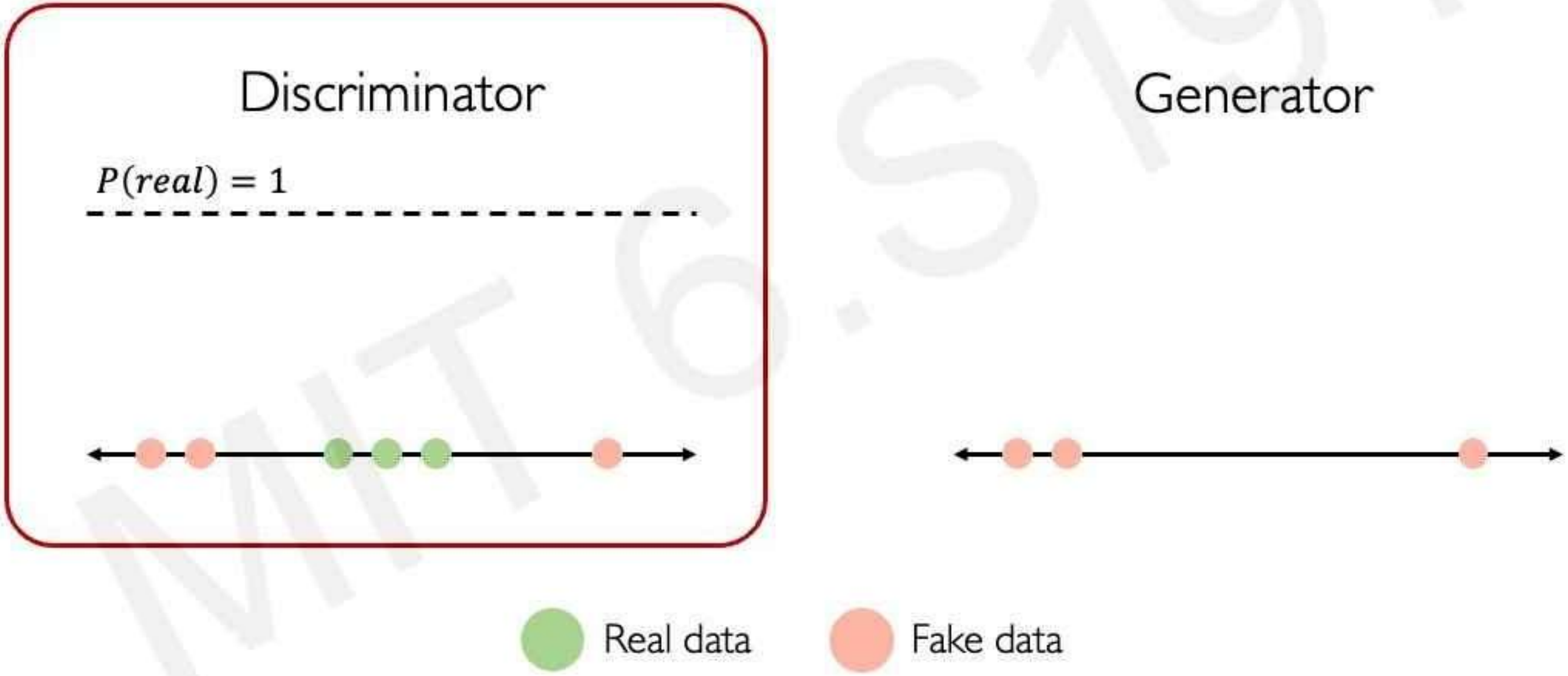
Intuition behind GANs

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



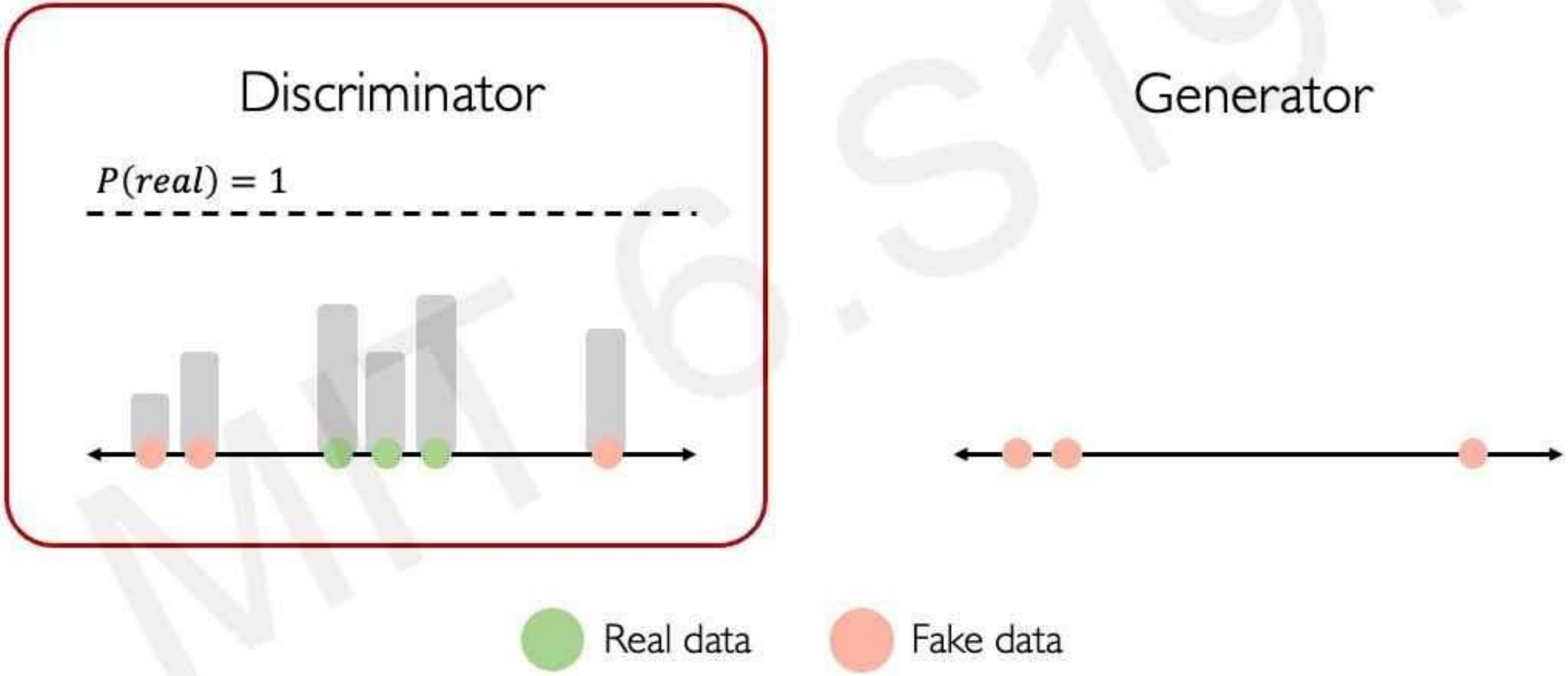
Intuition behind GANs

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



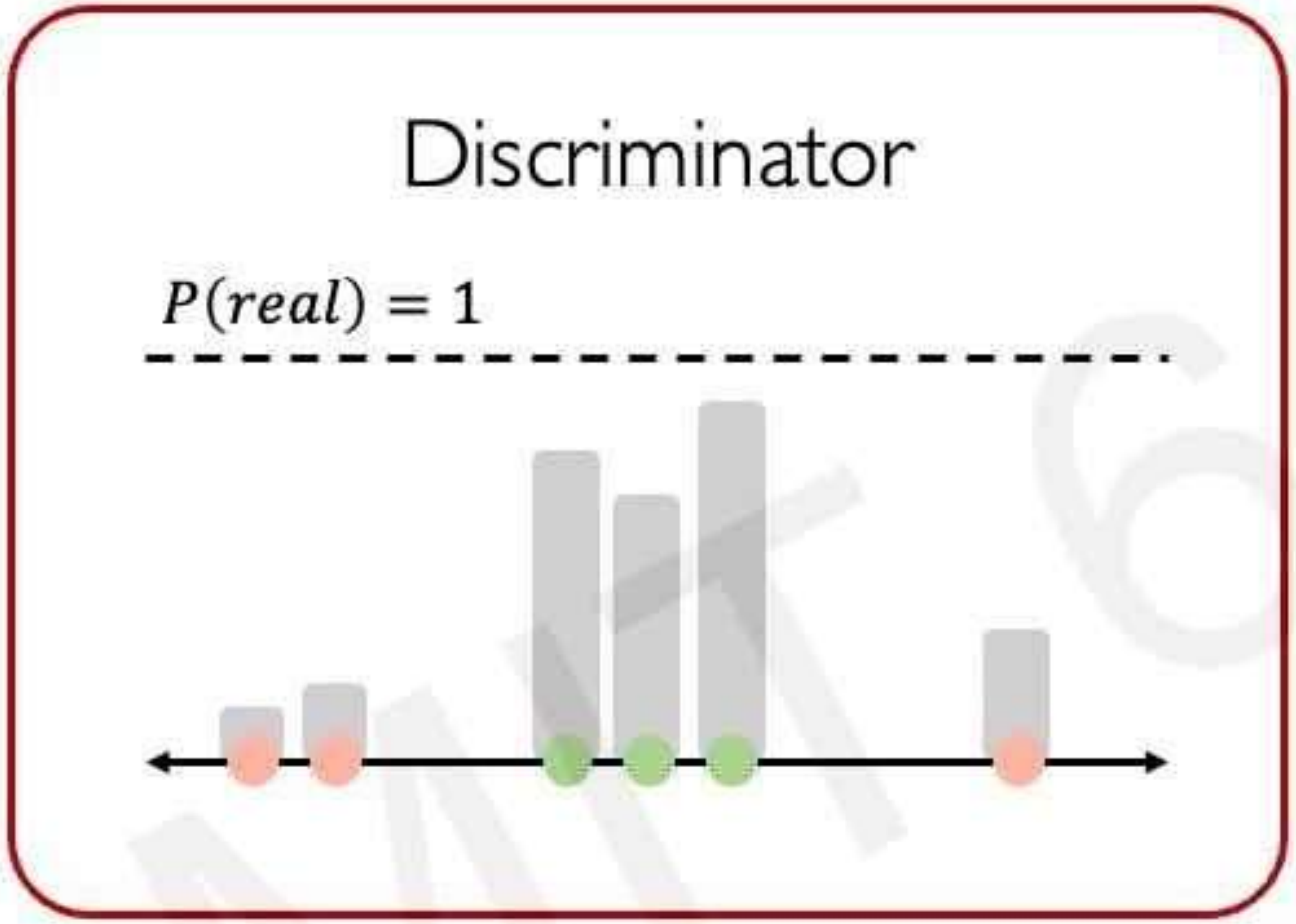
Intuition behind GANs

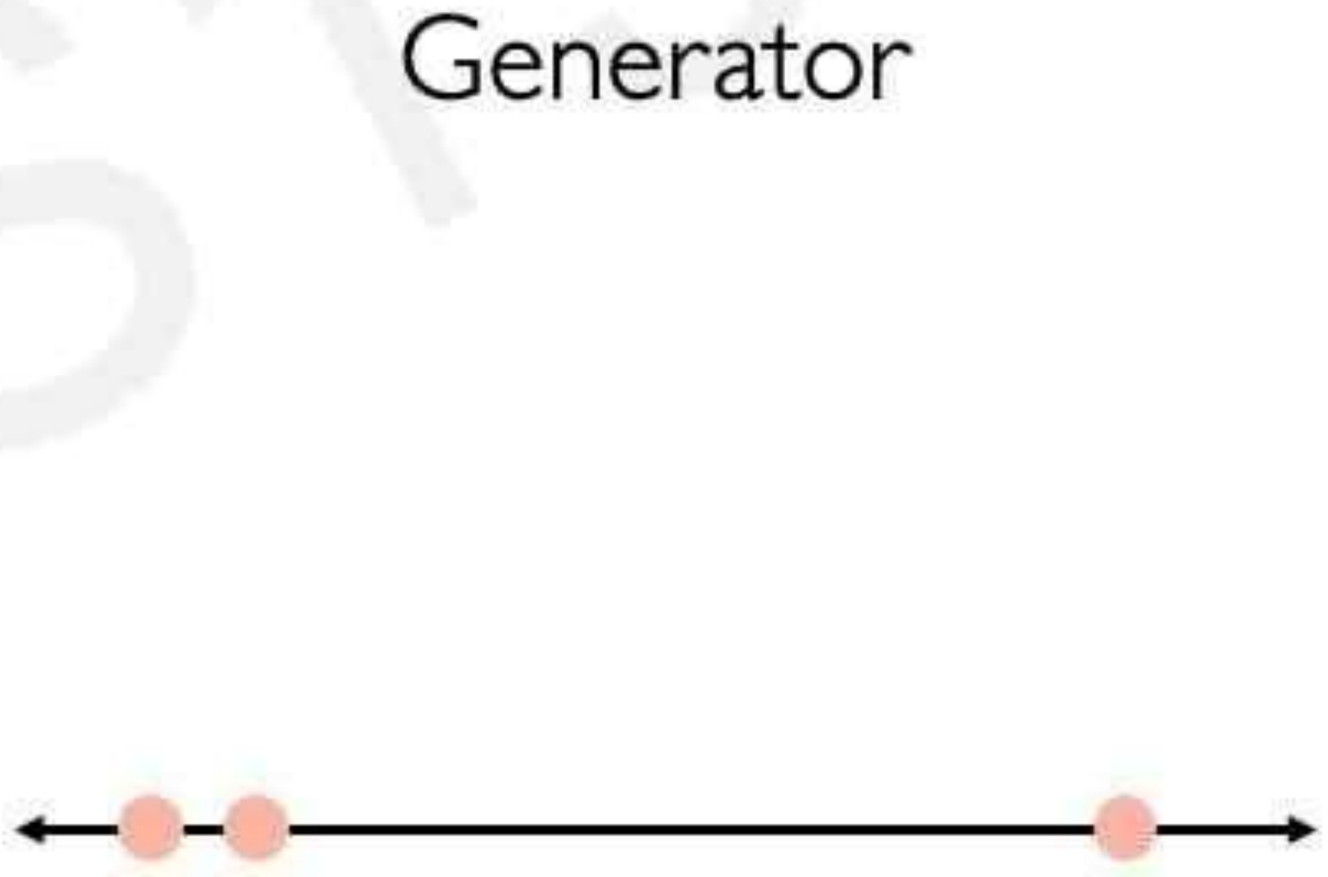
Discriminator tries to predict what's real and what's fake.



Intuition behind GANs

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

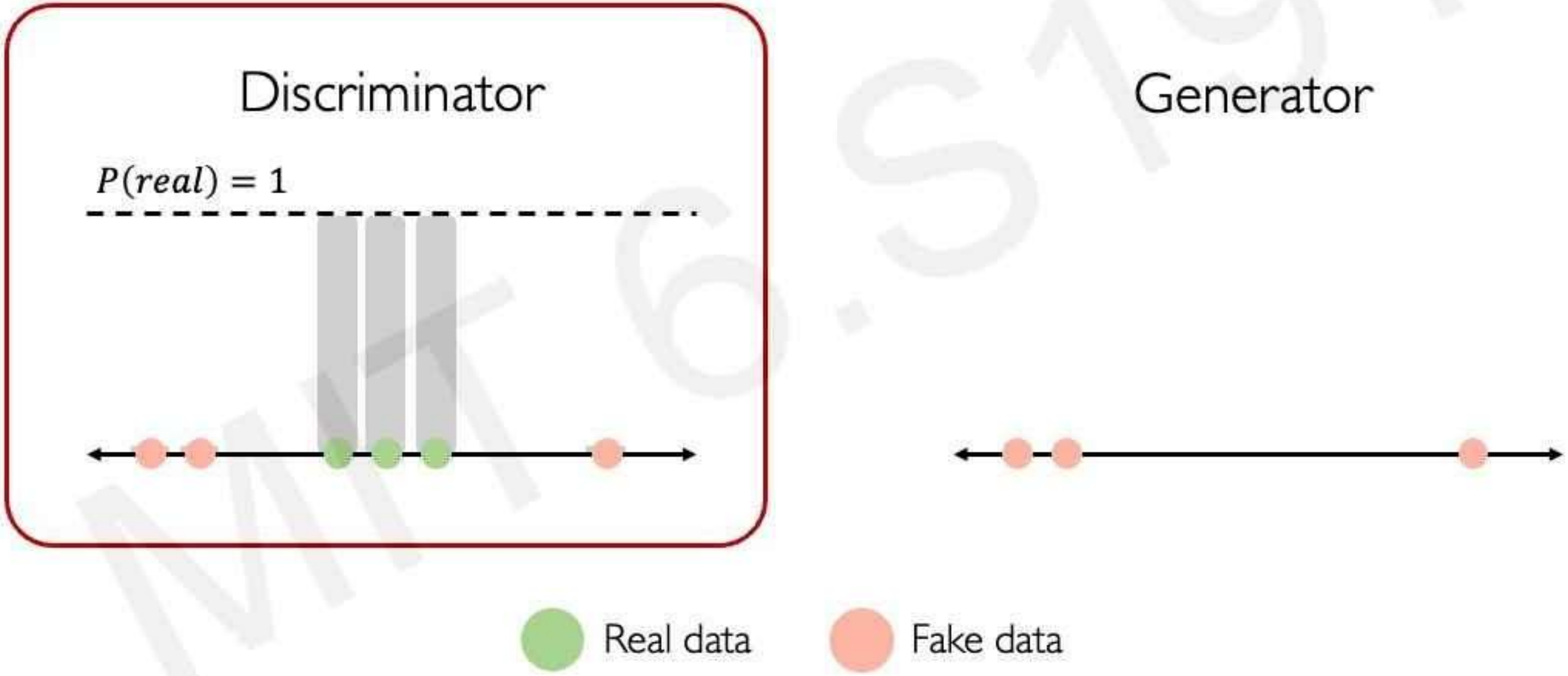




Real data   Fake data

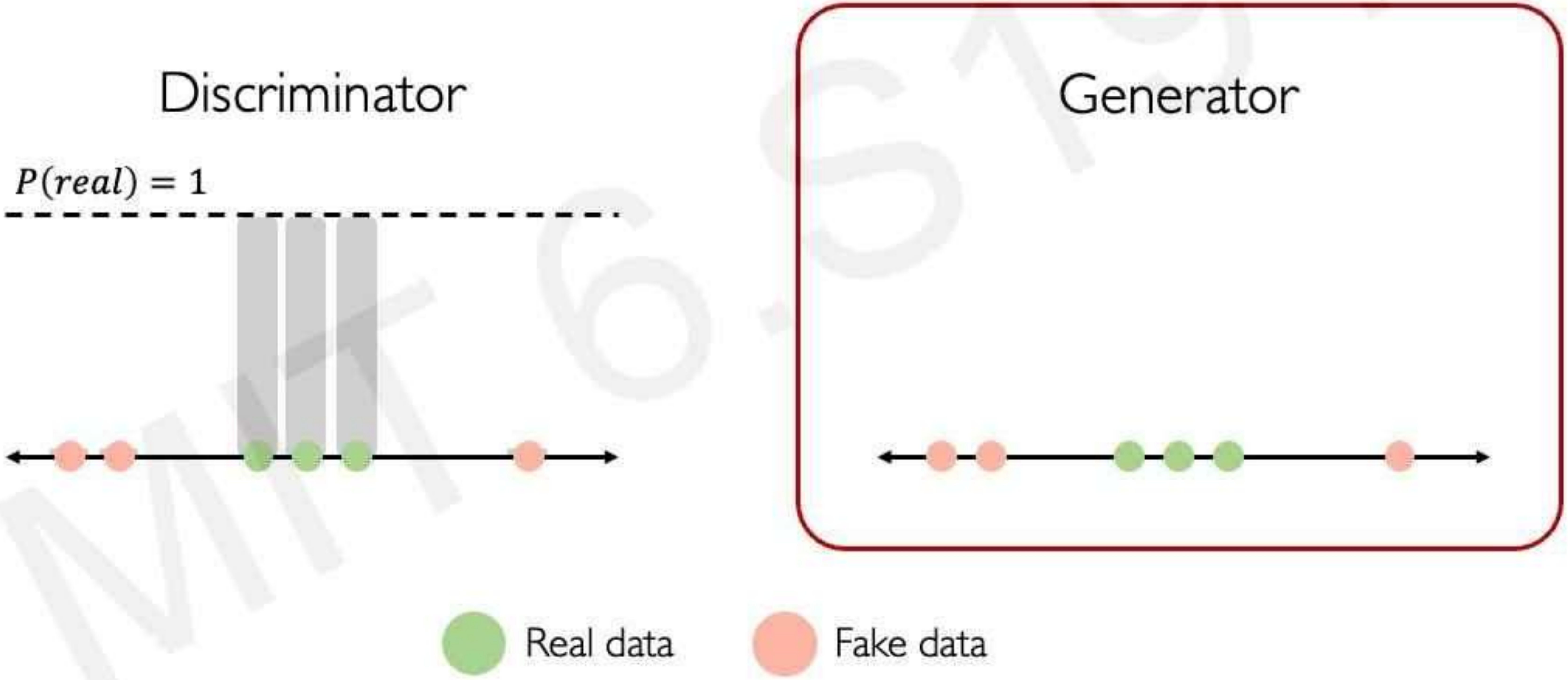
Intuition behind GANs

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



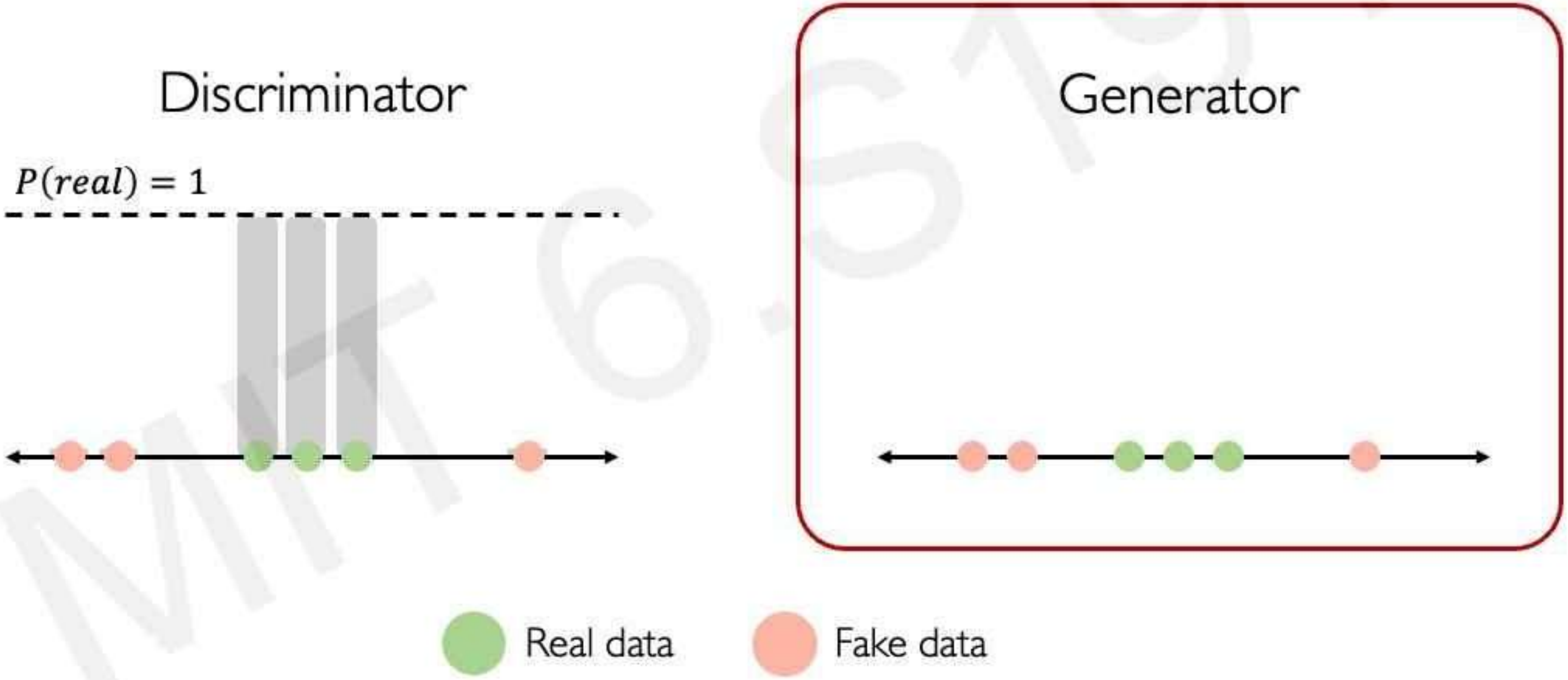
Intuition behind GANs

Generator tries to improve its imitation of the data.



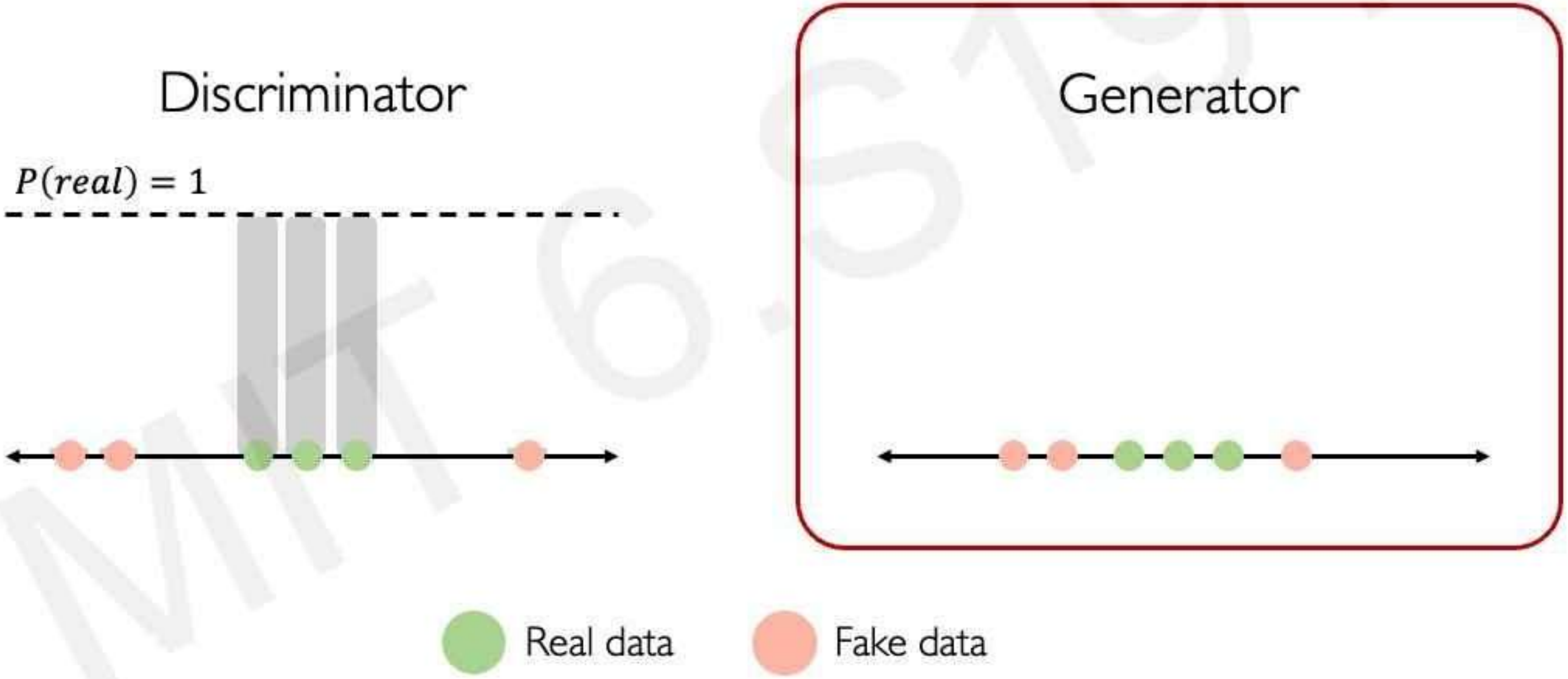
Intuition behind GANs

Generator tries to improve its imitation of the data.



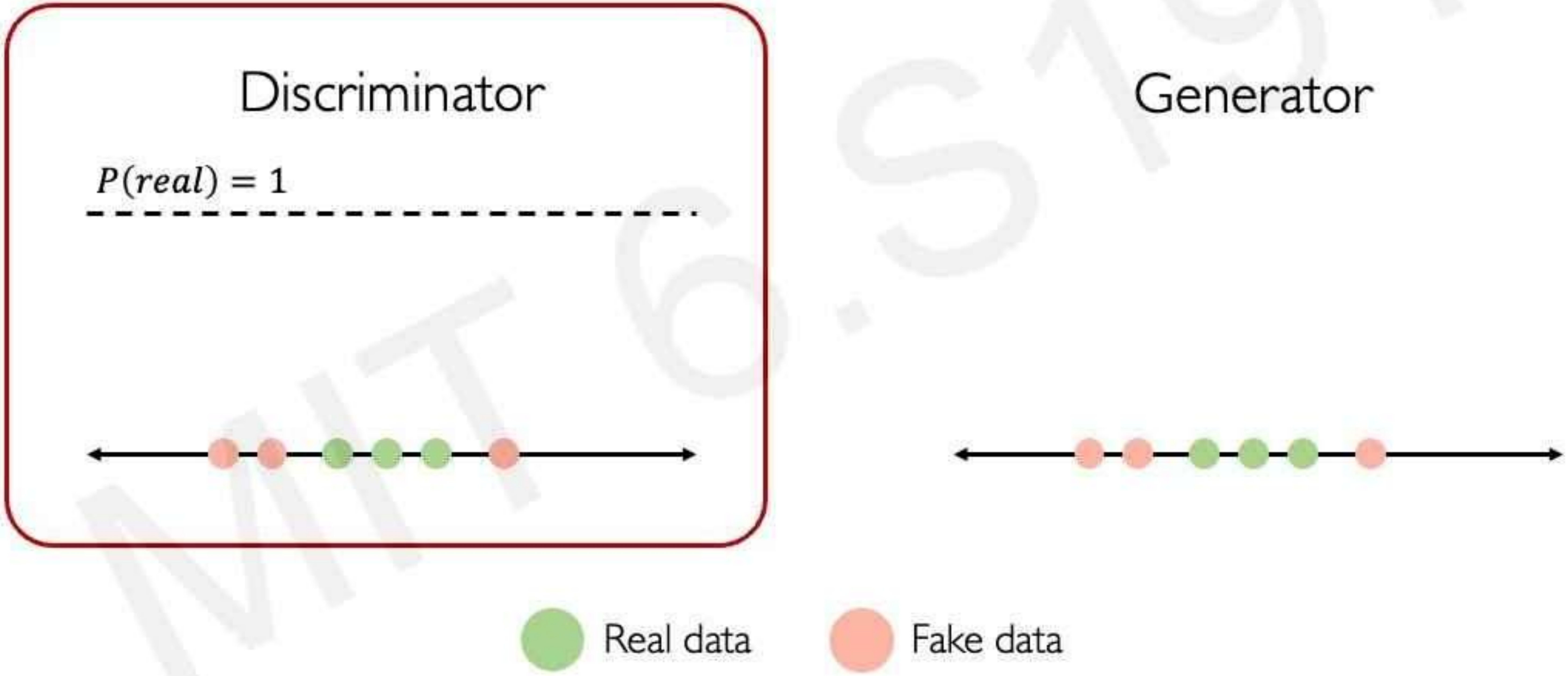
Intuition behind GANs

Generator tries to improve its imitation of the data.



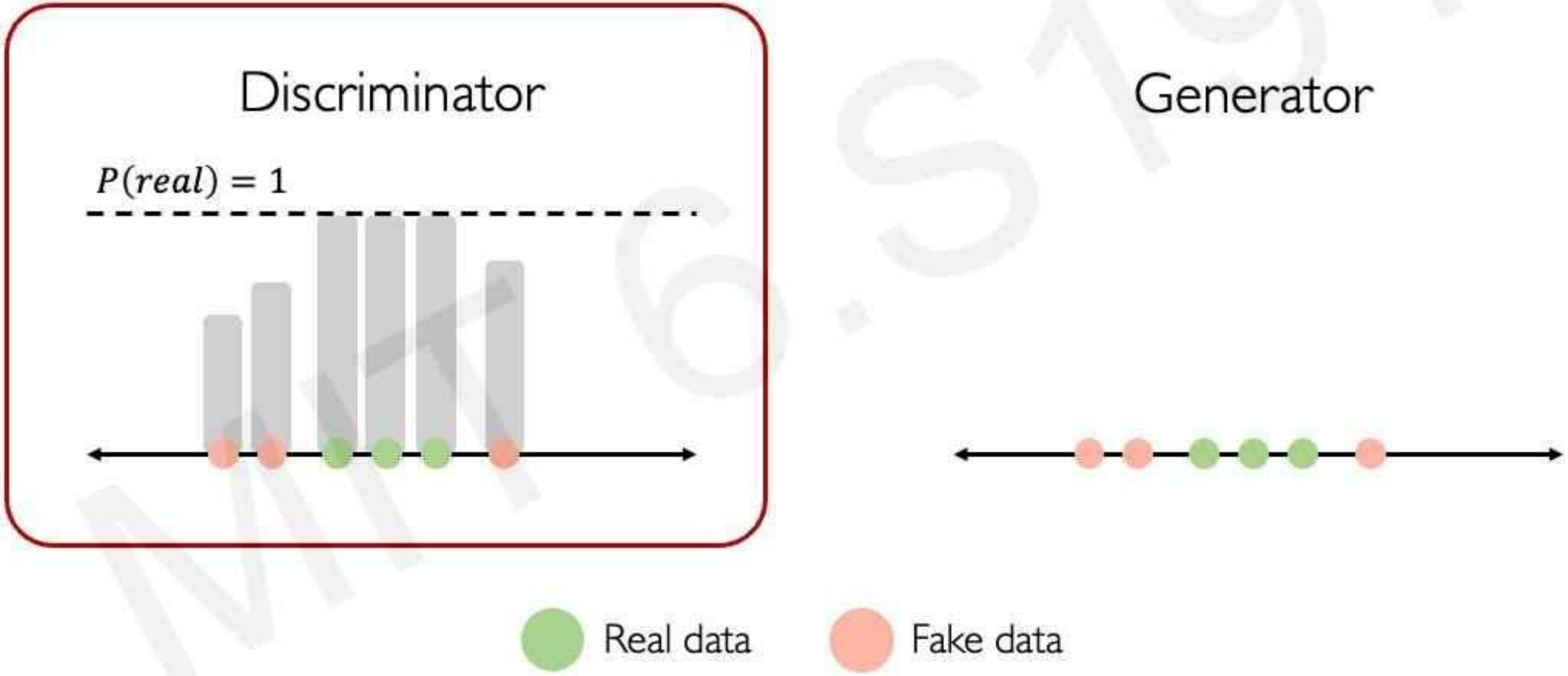
Intuition behind GANs

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



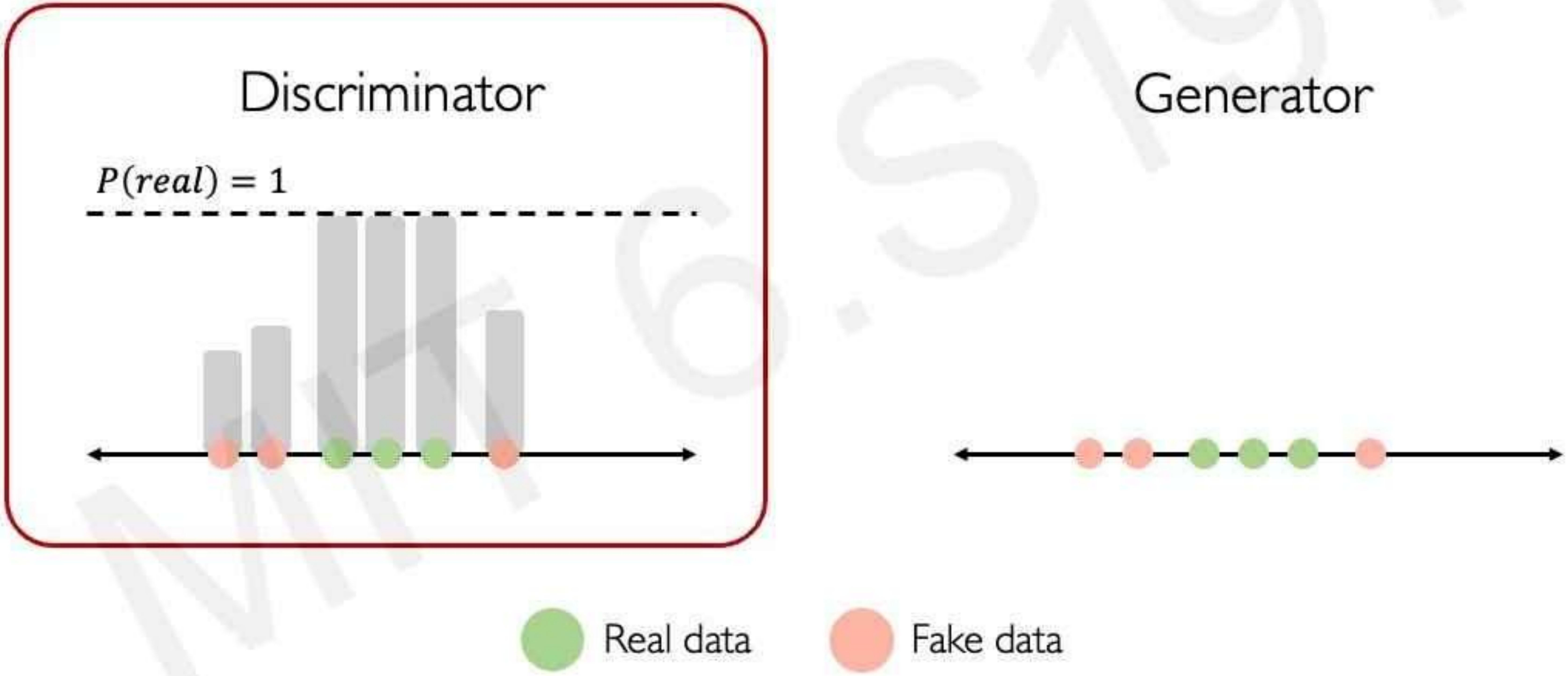
Intuition behind GANs

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



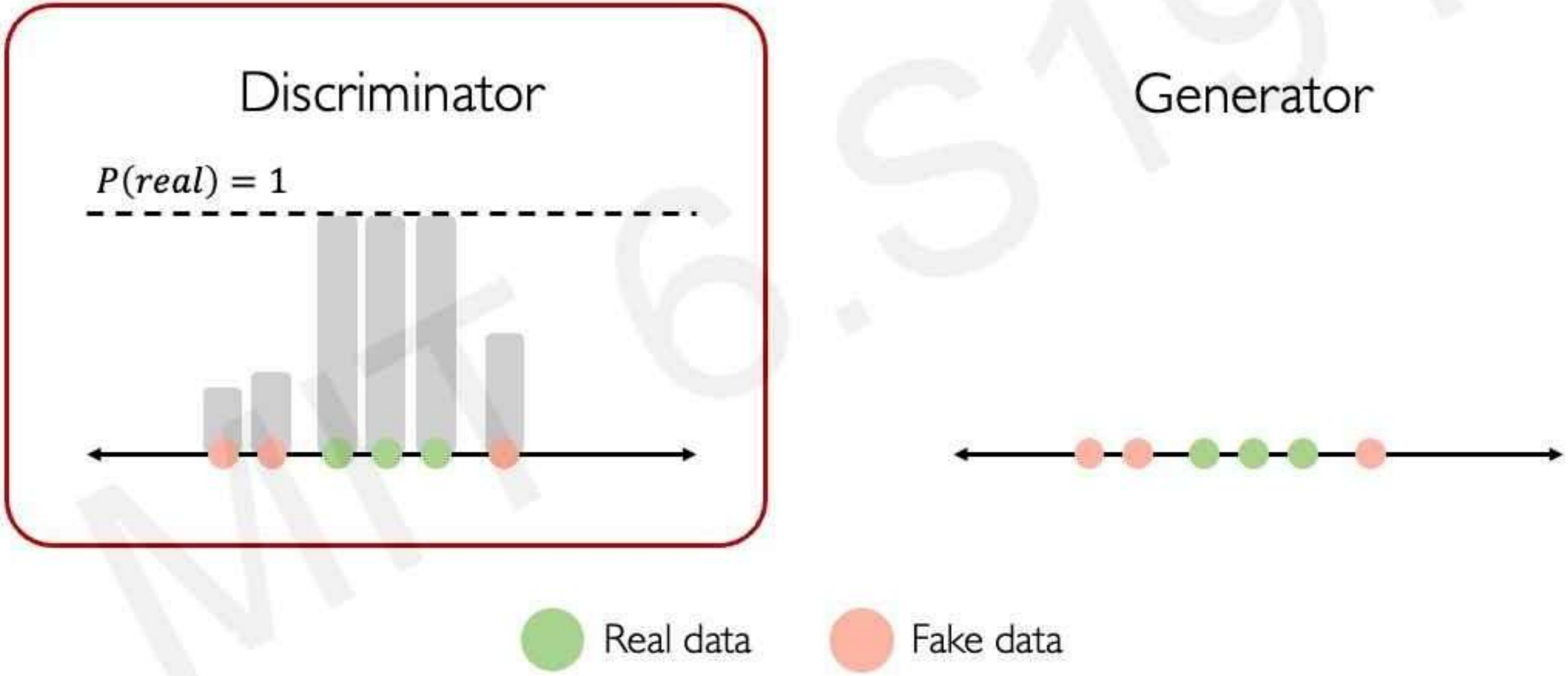
Intuition behind GANs

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



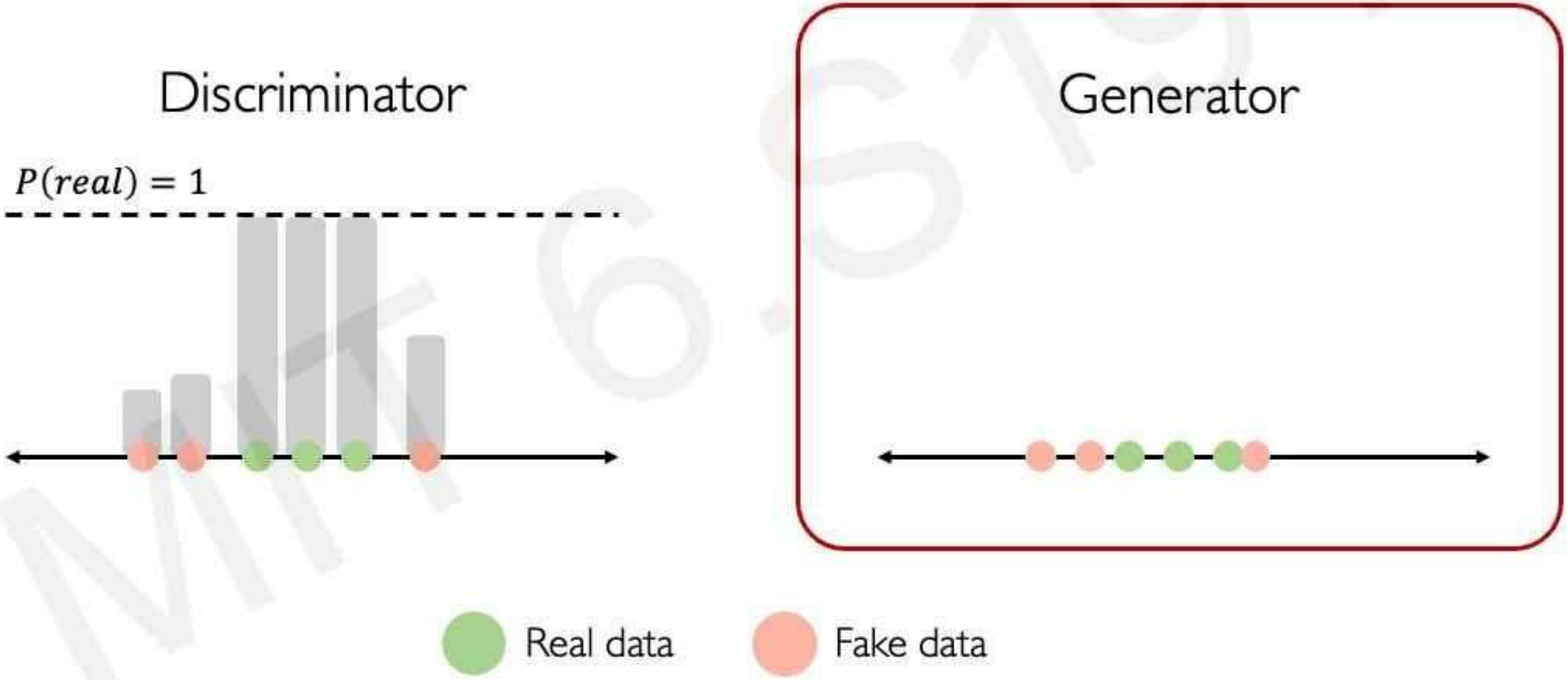
Intuition behind GANs

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



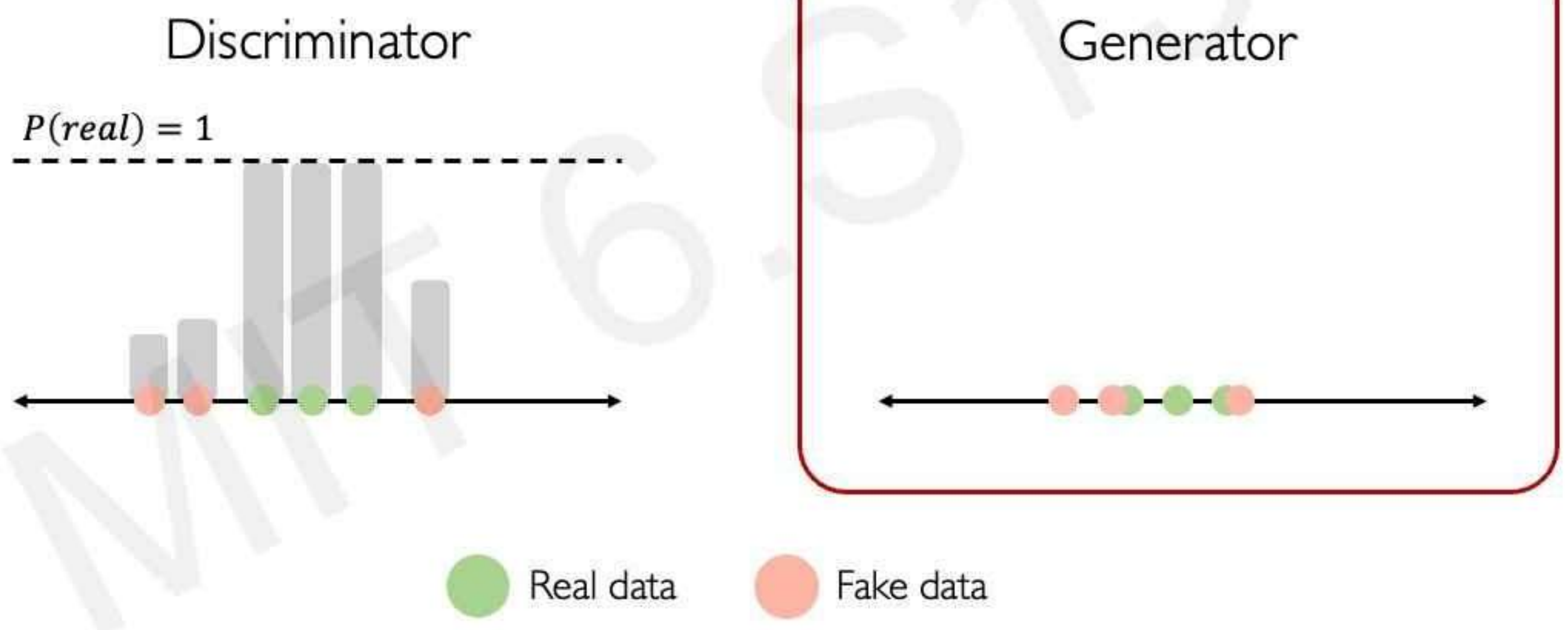
Intuition behind GANs

Generator tries to improve its imitation of the data.



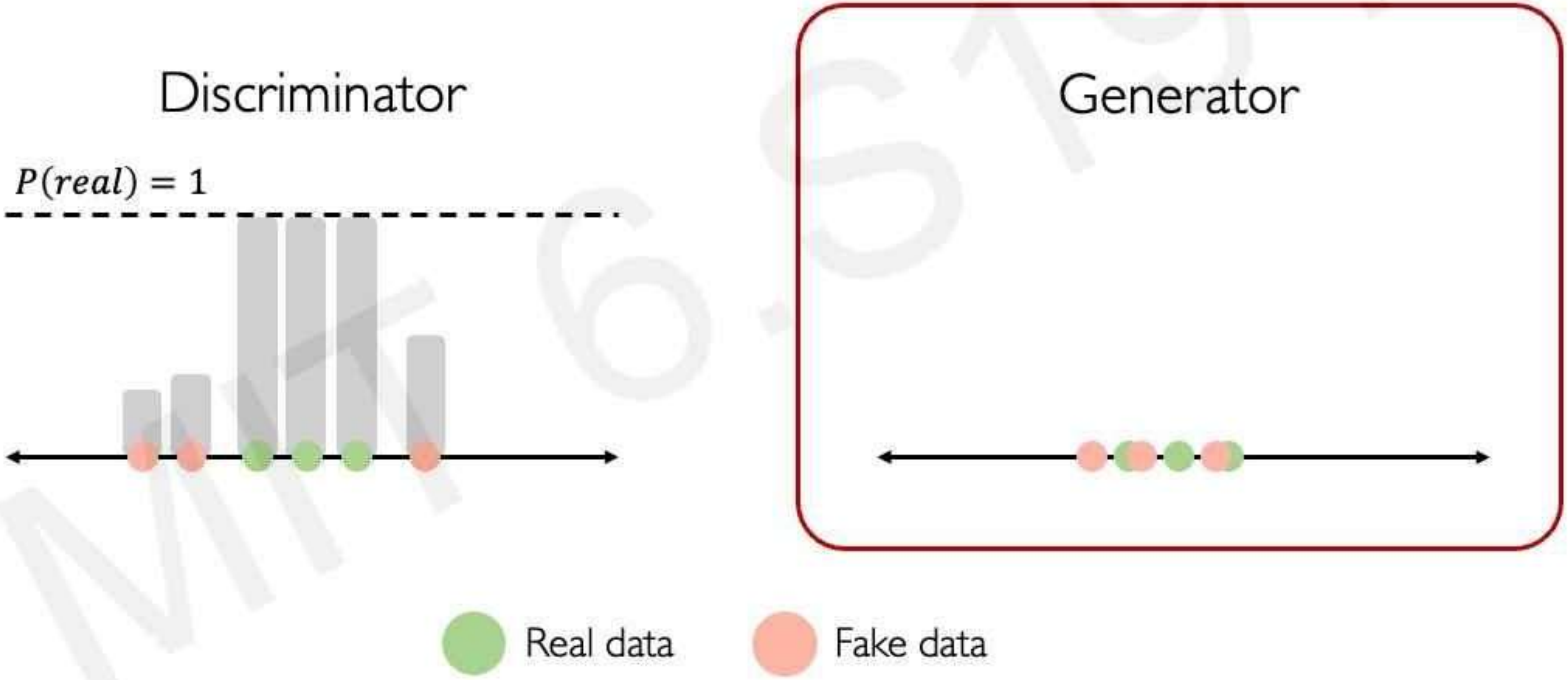
Intuition behind GANs

Generator tries to improve its imitation of the data.



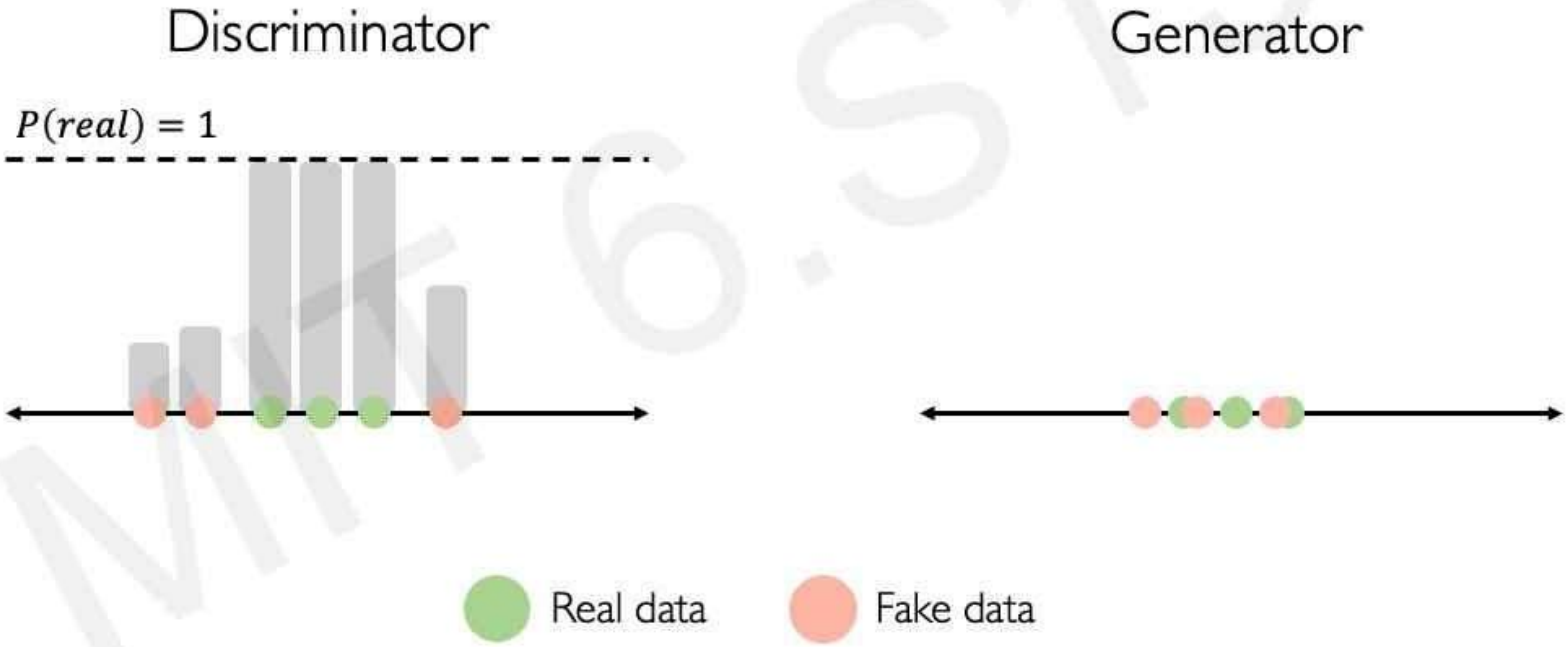
Intuition behind GANs

Generator tries to improve its imitation of the data.

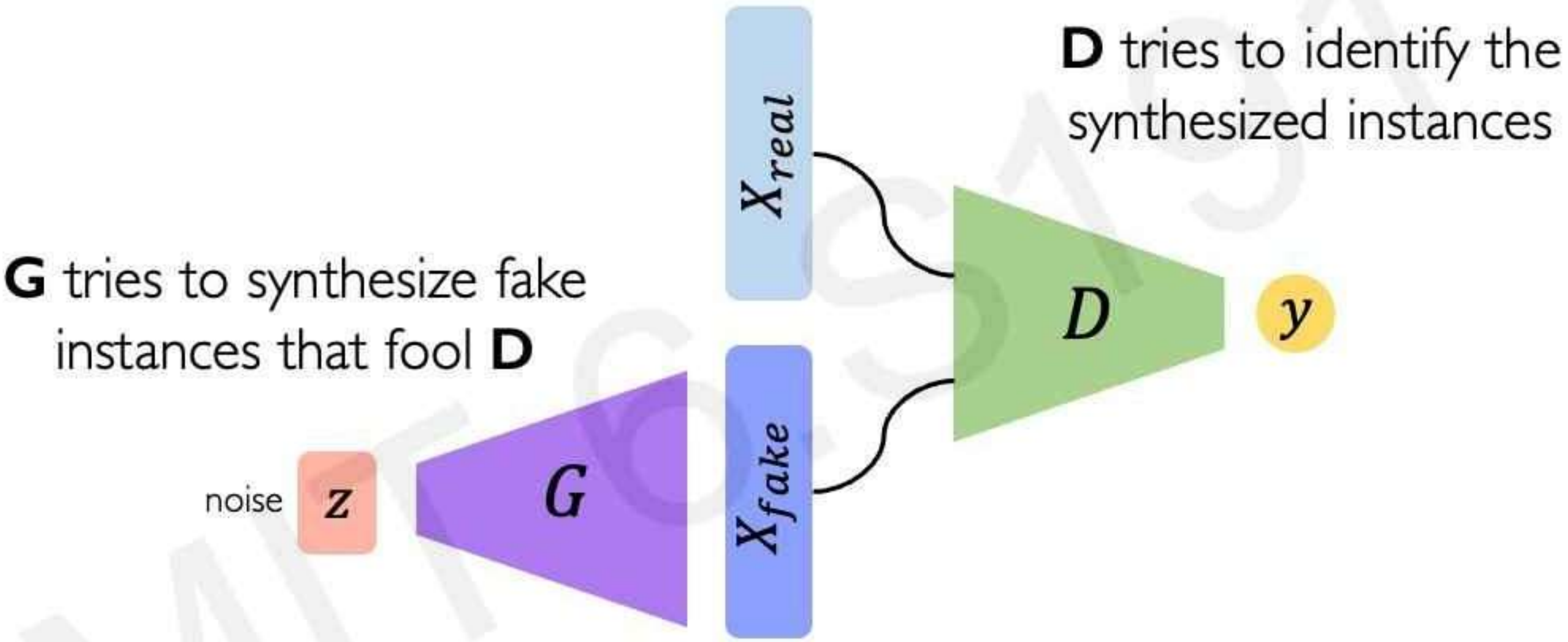


Intuition behind GANs

Discriminator tries to identify real data from fakes created by the generator.
Generator tries to create imitations of data to trick the discriminator.



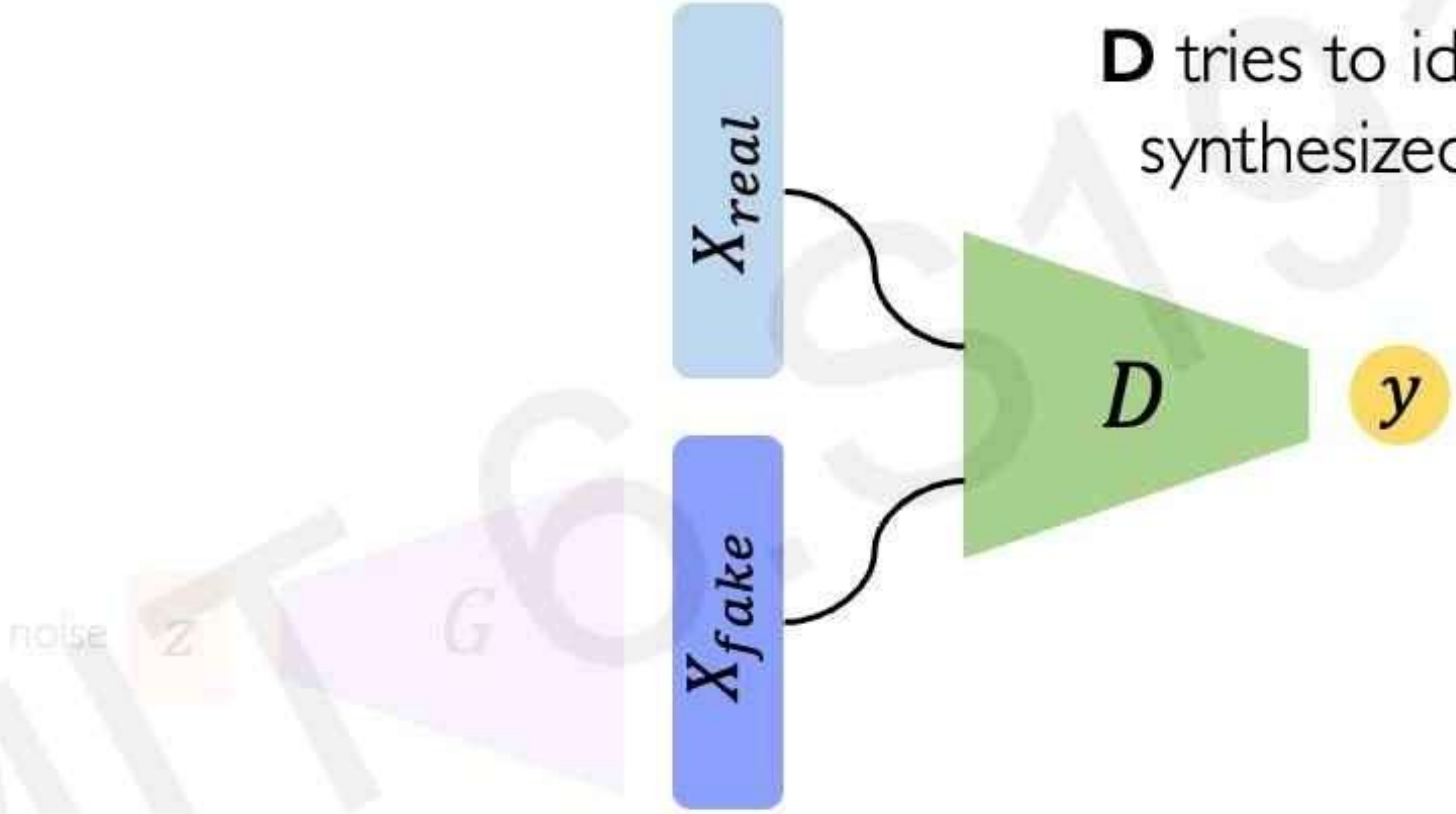
Training GANs



Training: adversarial objectives for **D** and **G**
Global optimum: G reproduces the true data distribution

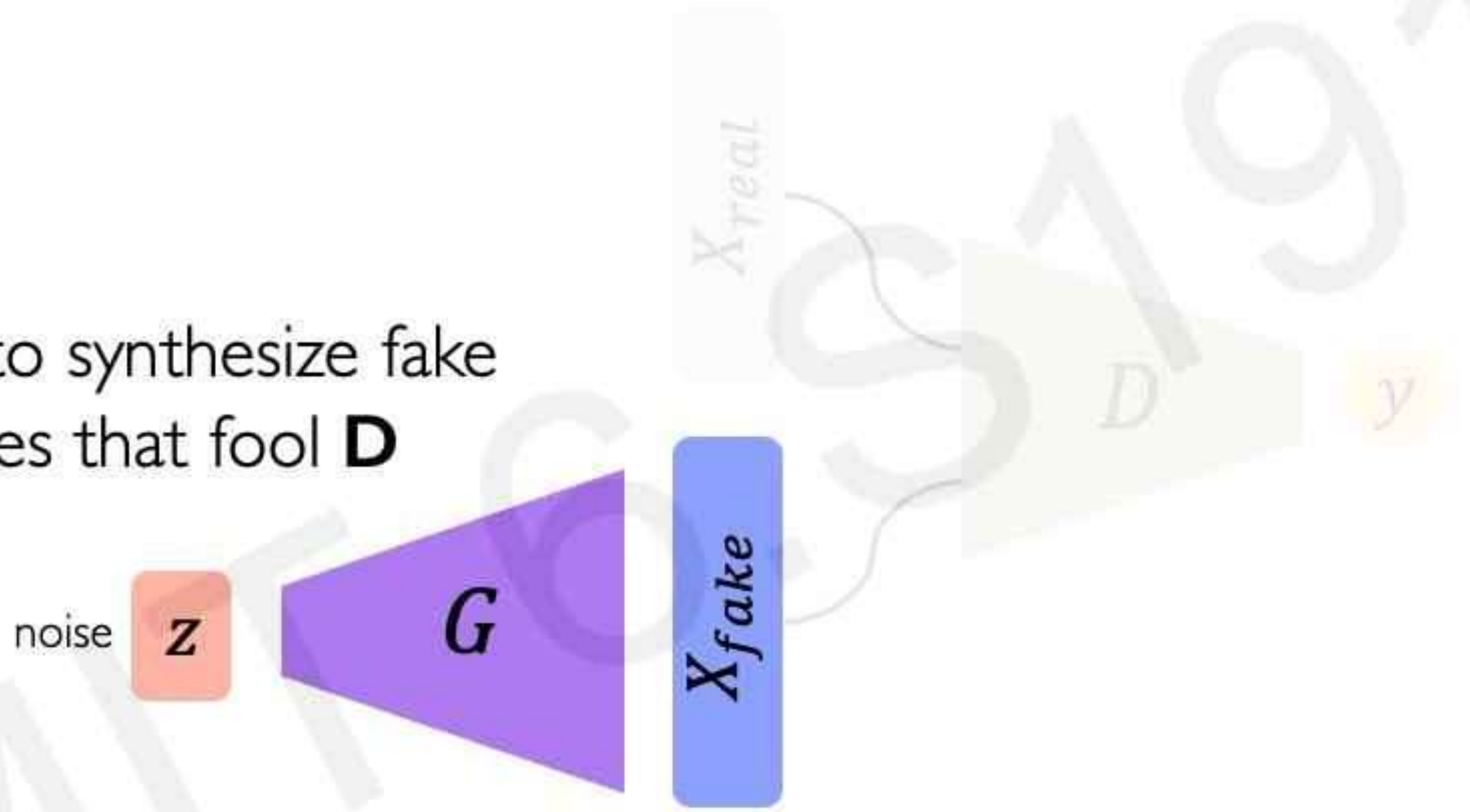
Training GANs: loss function

D tries to identify the synthesized images



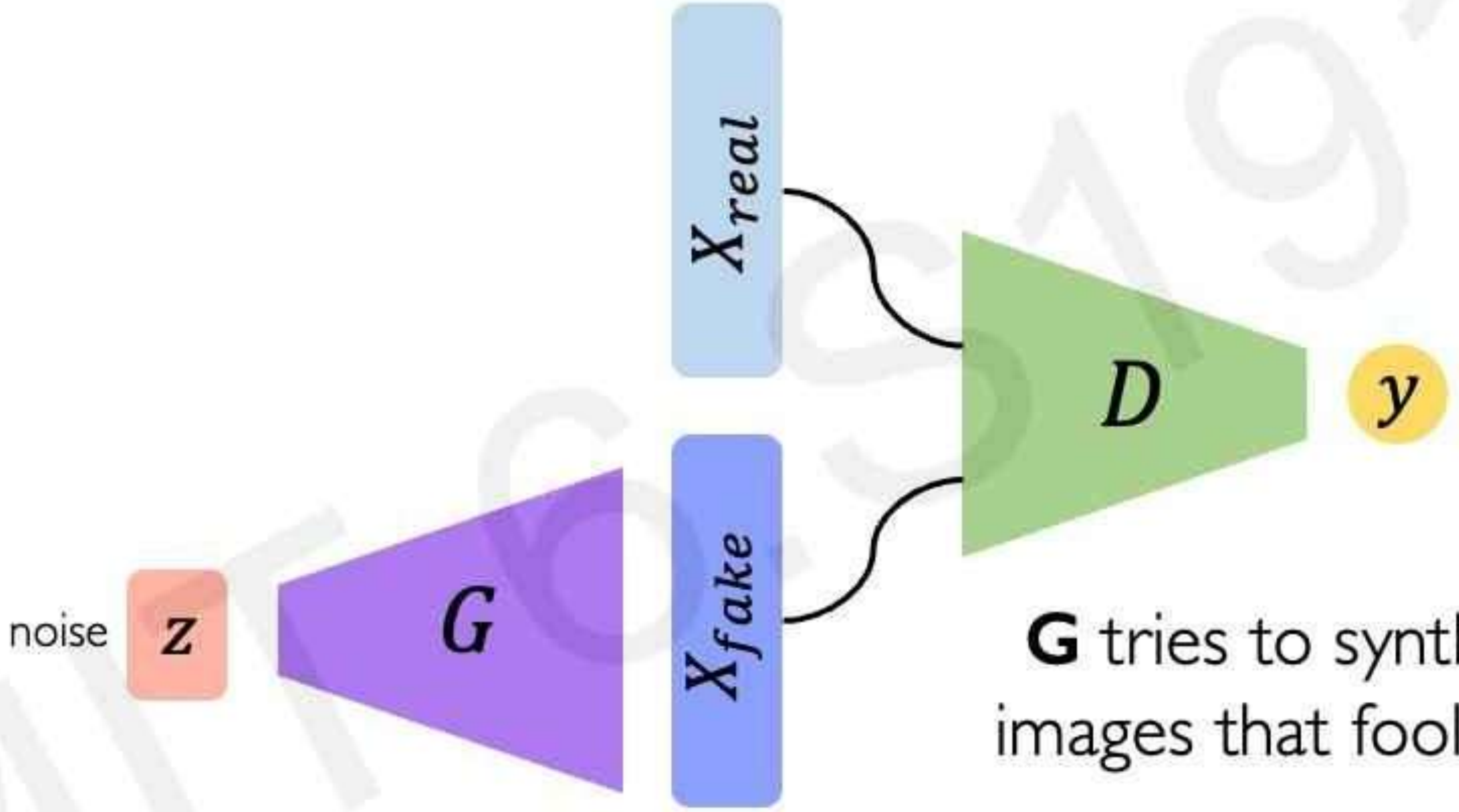
$$\underbrace{\arg\max_{D}}\ \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \underbrace{\log D(G(\mathbf{z}))}_{\textbf{Fake}} + \underbrace{\log\left(1 - D(\mathbf{x})\right)}_{\textbf{Real}}\ \right]$$

Training GANs: loss function



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

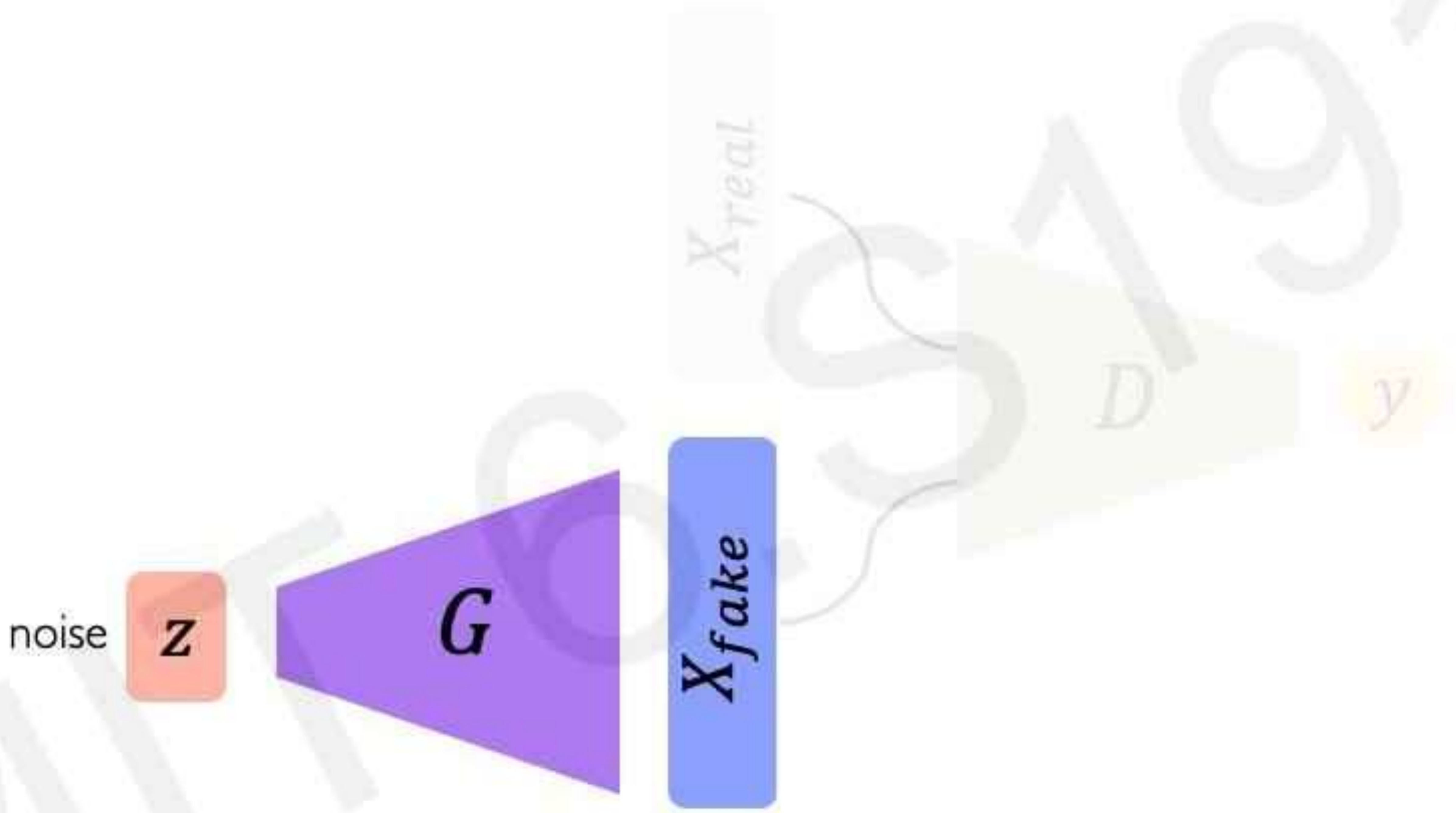
Training GANs: loss function



G tries to synthesize fake images that fool the *best* **D**

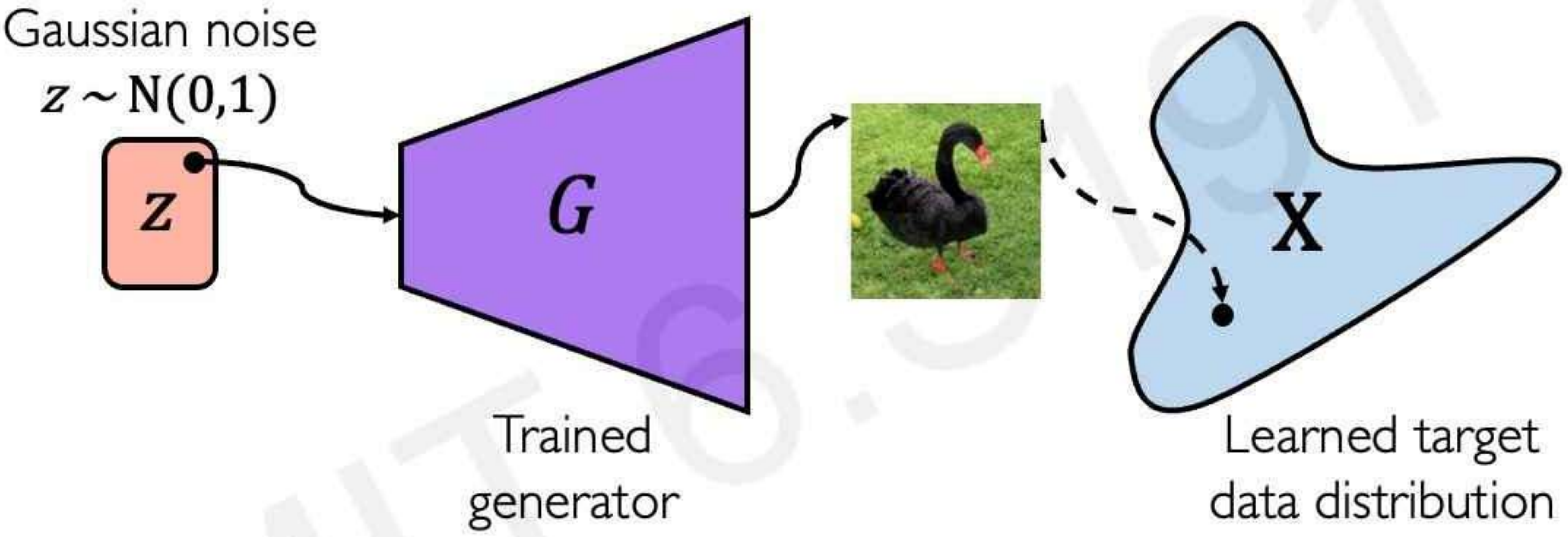
$$\arg\min_{G}\max_{D}\ \mathbb{E}_{\mathbf{z},\mathbf{x}}\left[\ \log D(G(\mathbf{z})) + \log\left(1 - D(\mathbf{x})\right)\ \right]$$

Generating new data with GANs

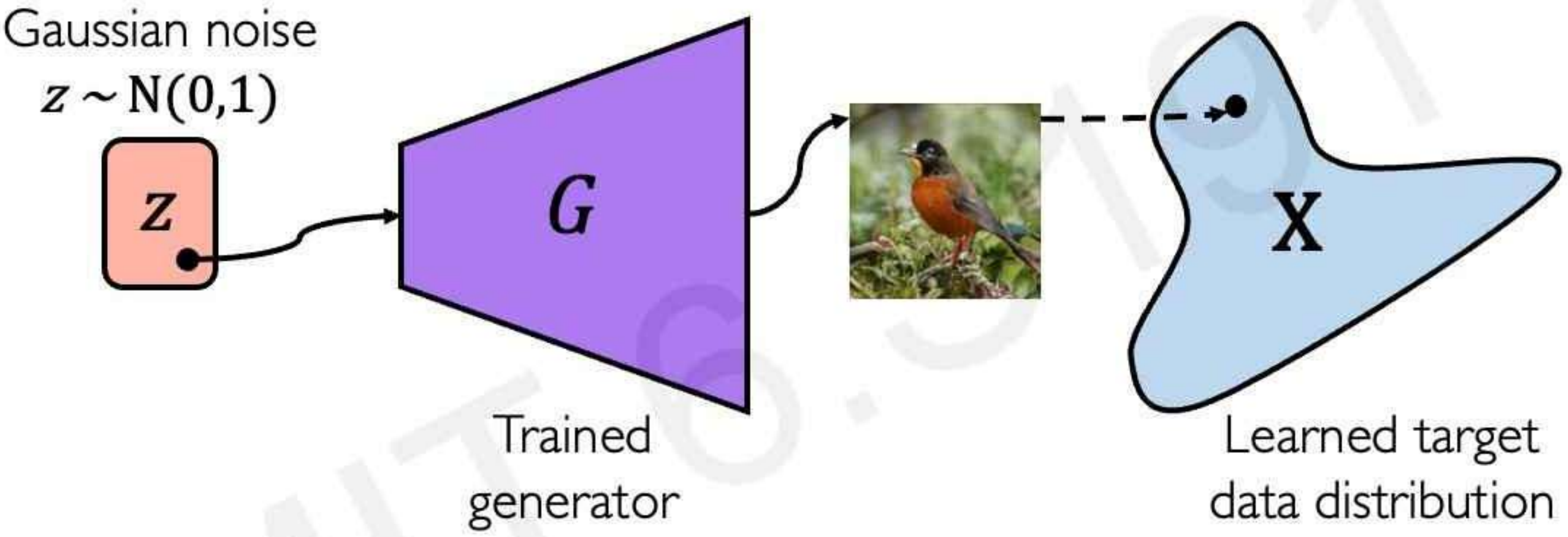


After training, use generator network to create **new data** that's never been seen before.

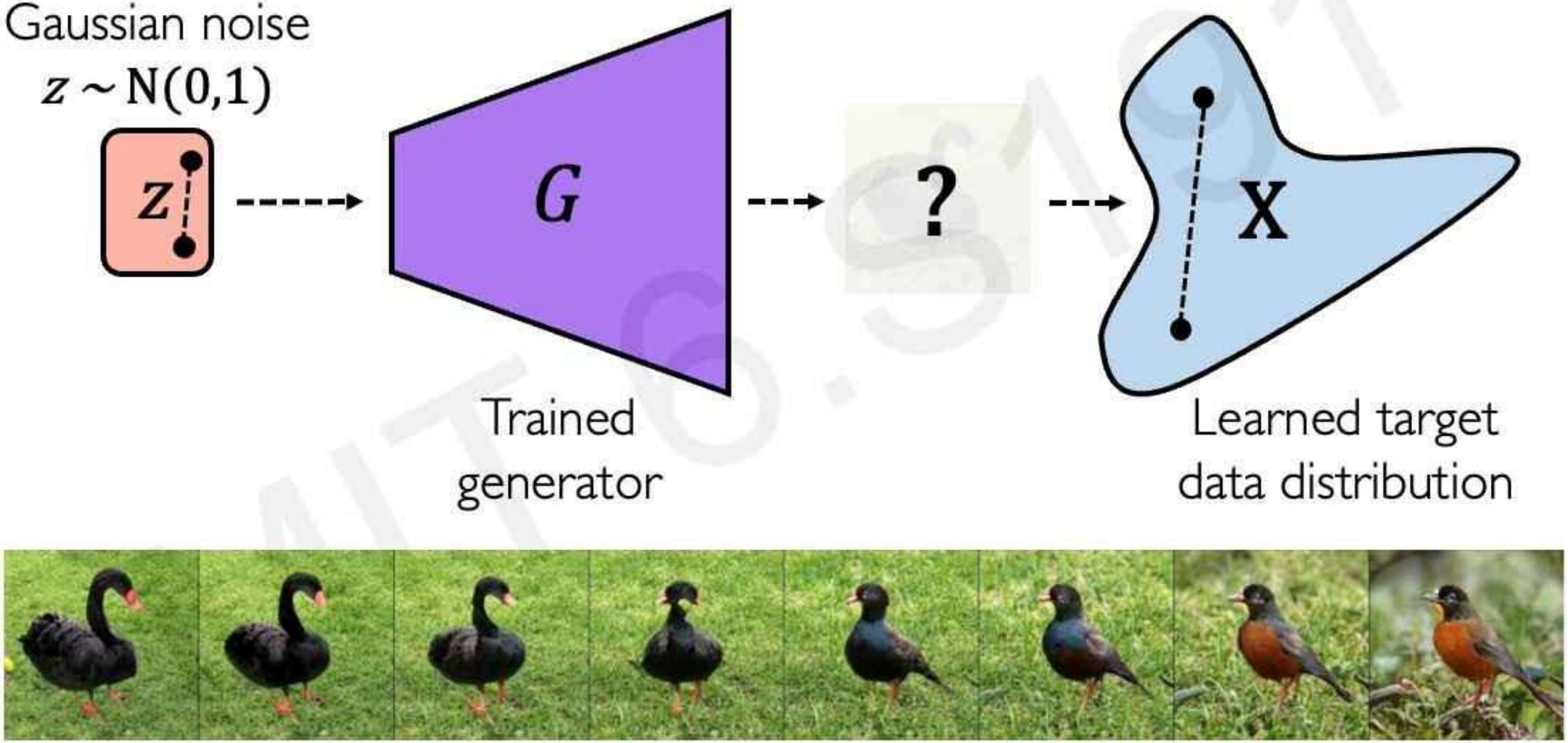
GANs are distribution transformers



GANs are distribution transformers



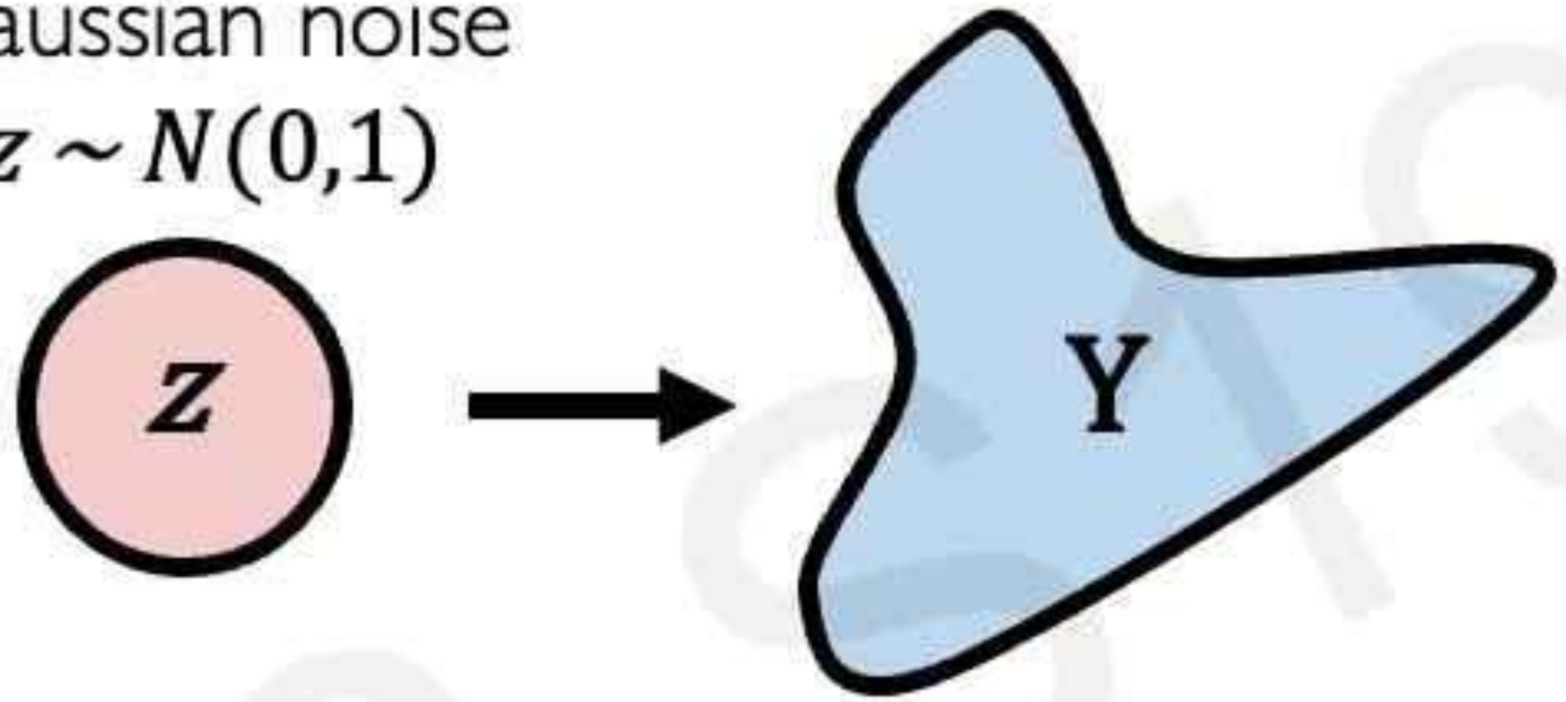
GANs are distribution transformers



Distribution transformations

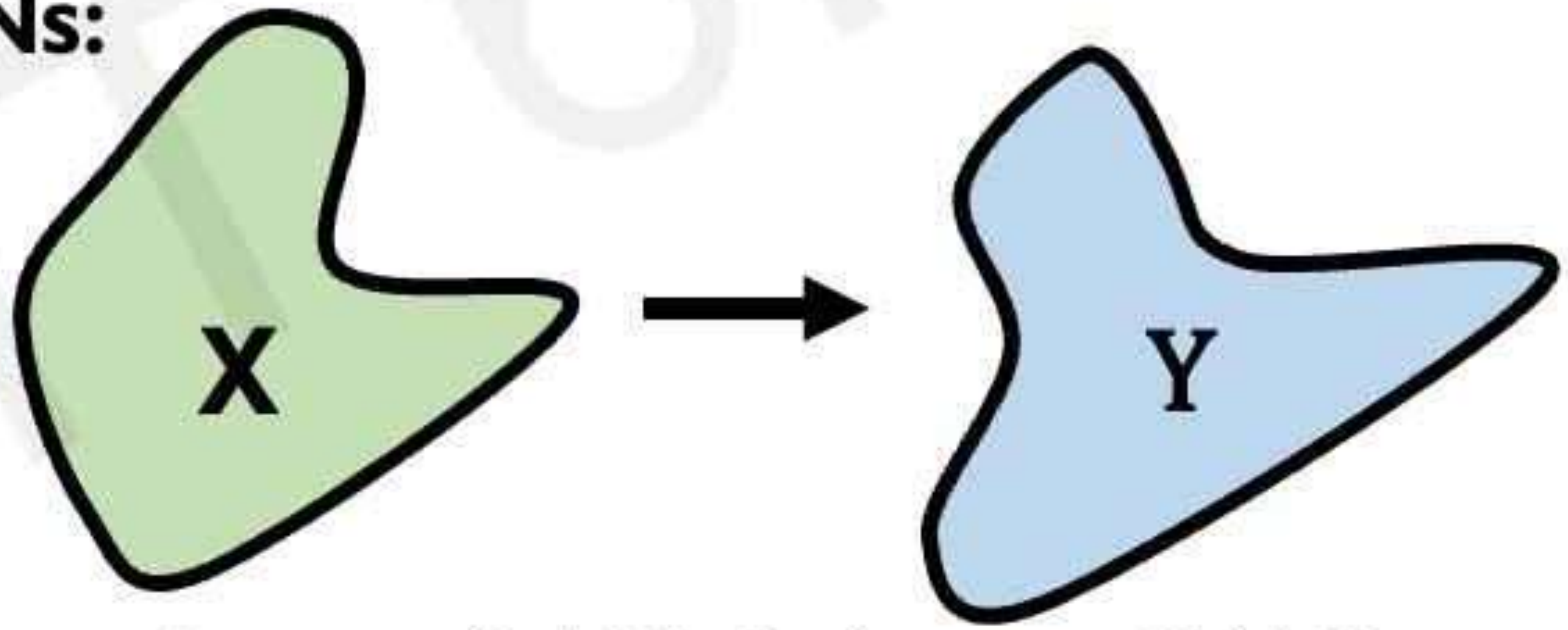
GANs:

Gaussian noise
$z \sim N(0,1)$



Gaussian noise → target data manifold

CycleGANs:



data manifold X → data manifold Y