Deep Learning
Generative Deep Learning

Manuel Piñar Molina

GANs (Generative Adversarial Networks)

Applications of GANs

Adversary training is the most interesting idea in the last 10 years of machine learning,

- **Image Synthesis (Fake images)**
- **Video Synthesis (Fake videos)**
- **Image-to-Image Translation (Artistic creations)**
- **Super-resolution**
- **Data Augmentation (++++ Dataset)**
- **Style Transfer**
- **Security**
- **3D object generation**

Applications of GANs

Fake images

From Progressive Growing of GANs[5](5):

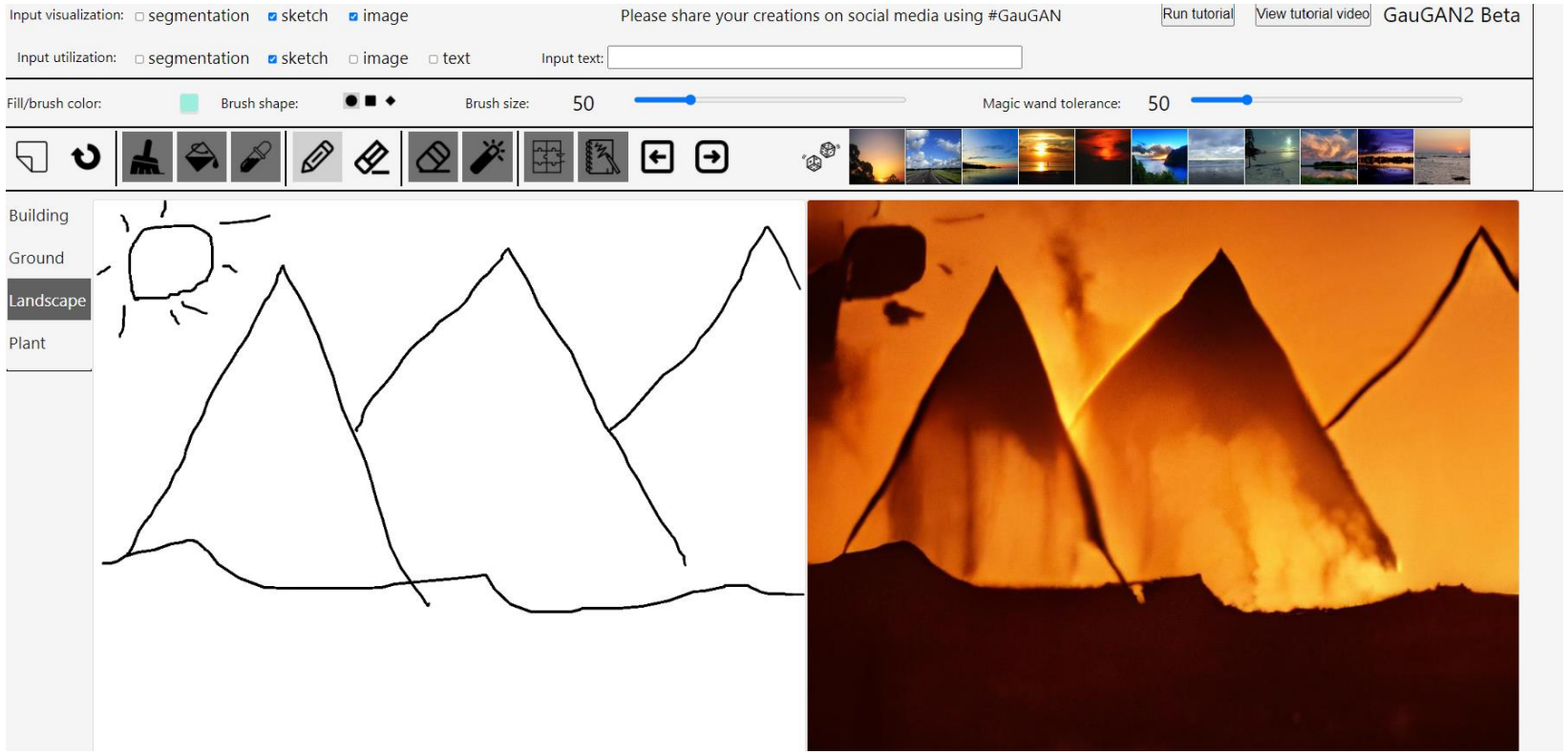
Nvidia

Applications of GANs

Fake videos

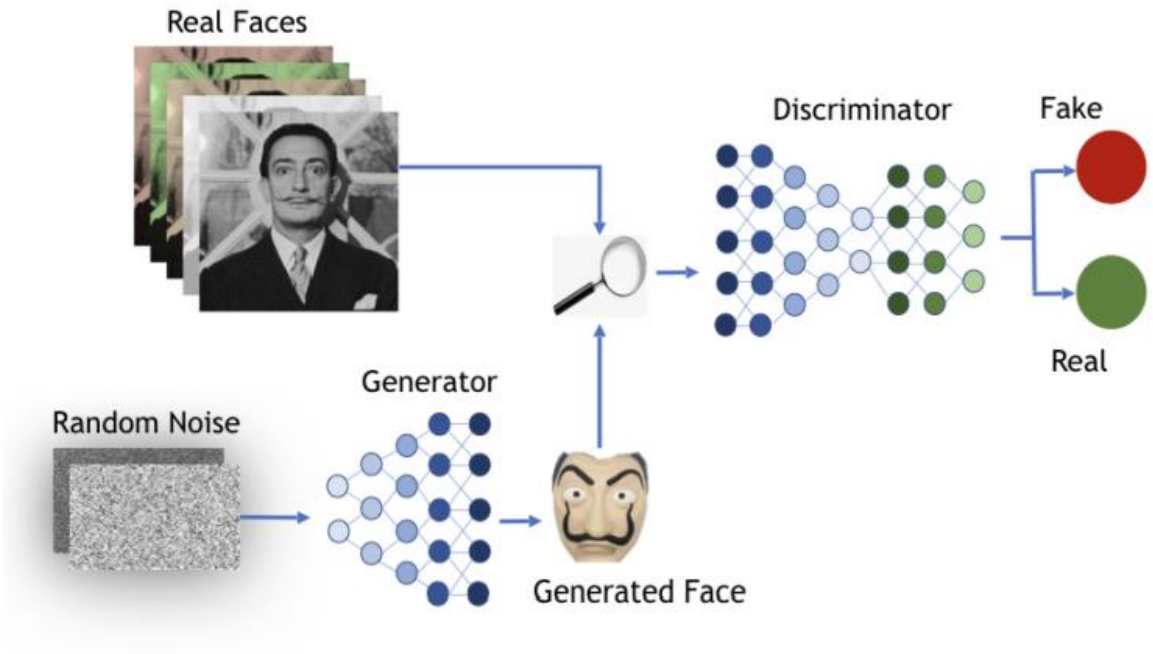
Applications of GANs

Artistic creations (Gaugan 2)



Applications of GANs

Security



- Cybersecurity
- Fraud detection
- Biometric authentication
- Video surveillance
- Malware analysis

Generative Adversarial Networks in Security: A Survey [Kalyan 2020]

Applications of GANs

3D OBJECT GENERATION

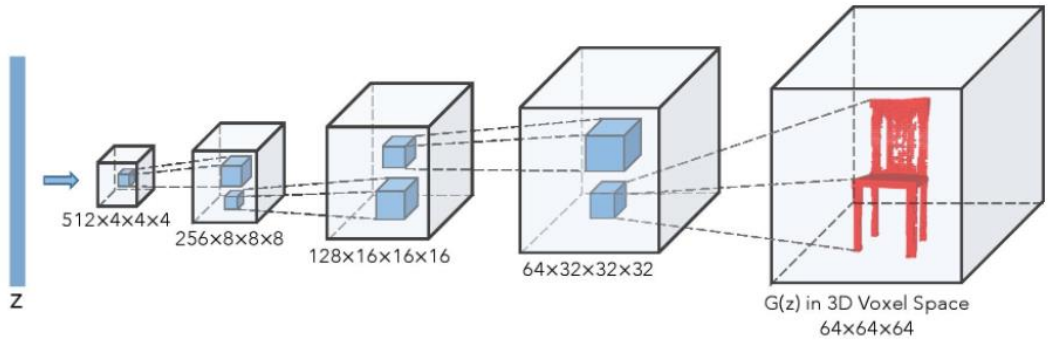
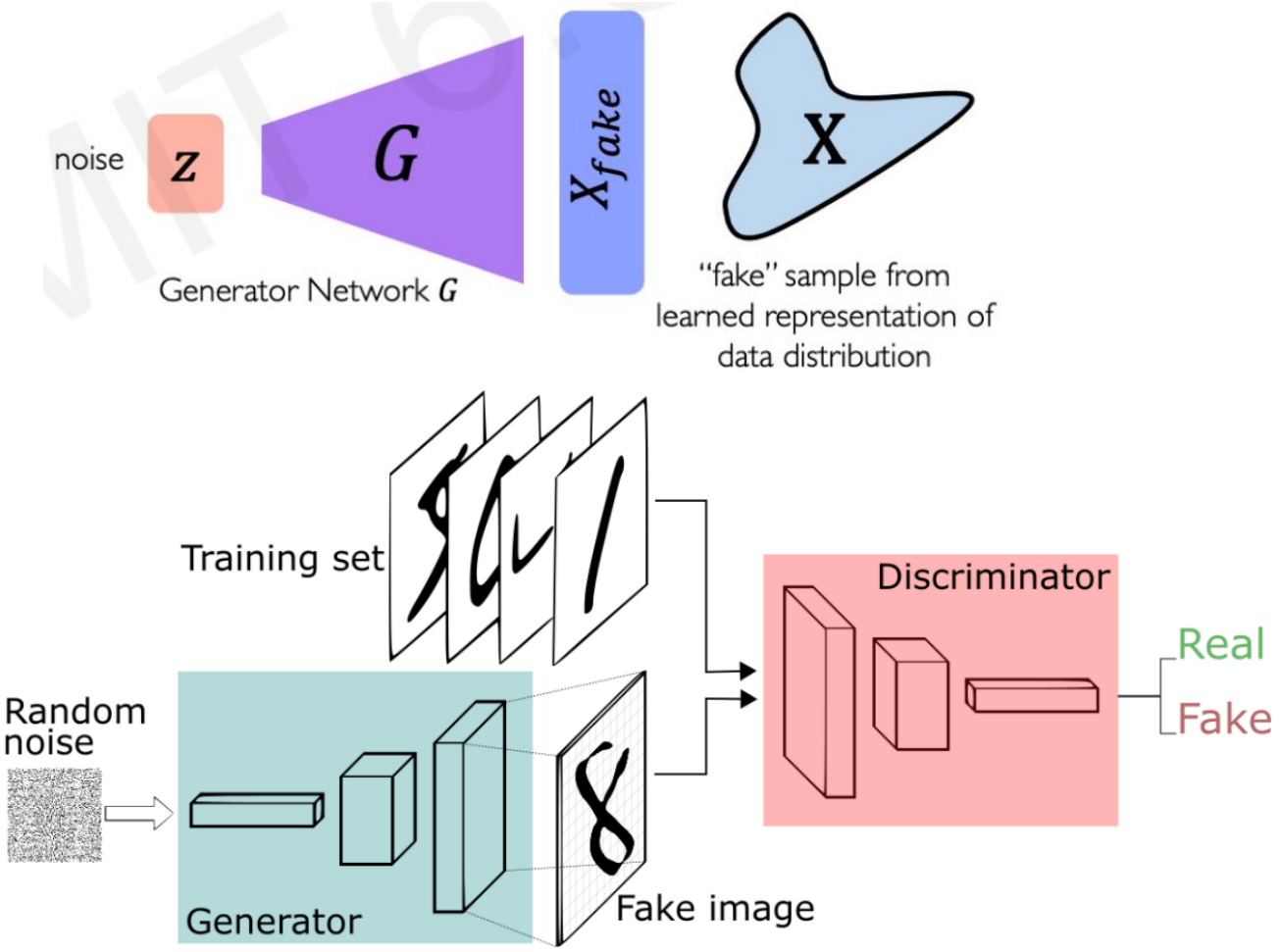


Figure 1: The generator of 3D Generative Adversarial Networks (3D-GAN)

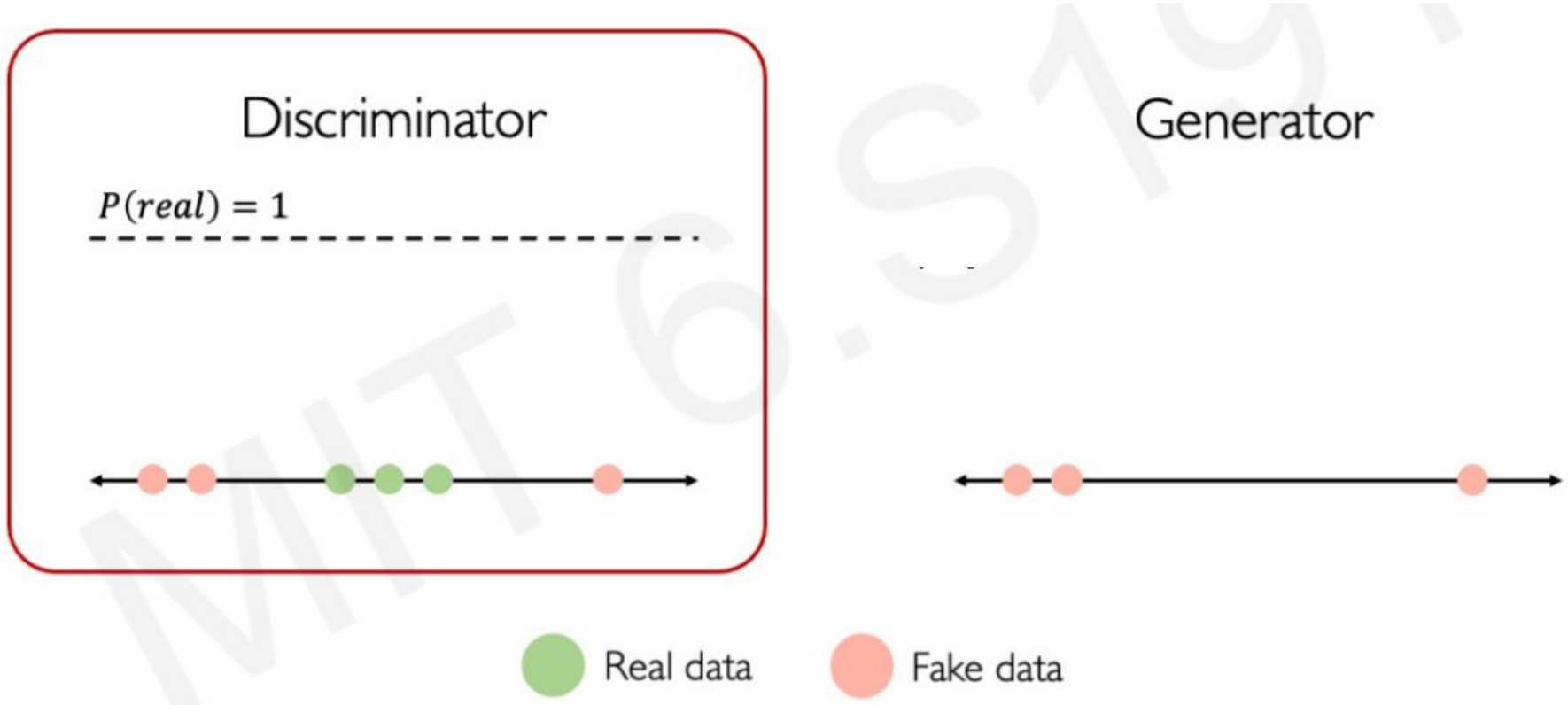
Figure 2: Shapes synthesized by 3D-GAN

MIT

Structure of GANs

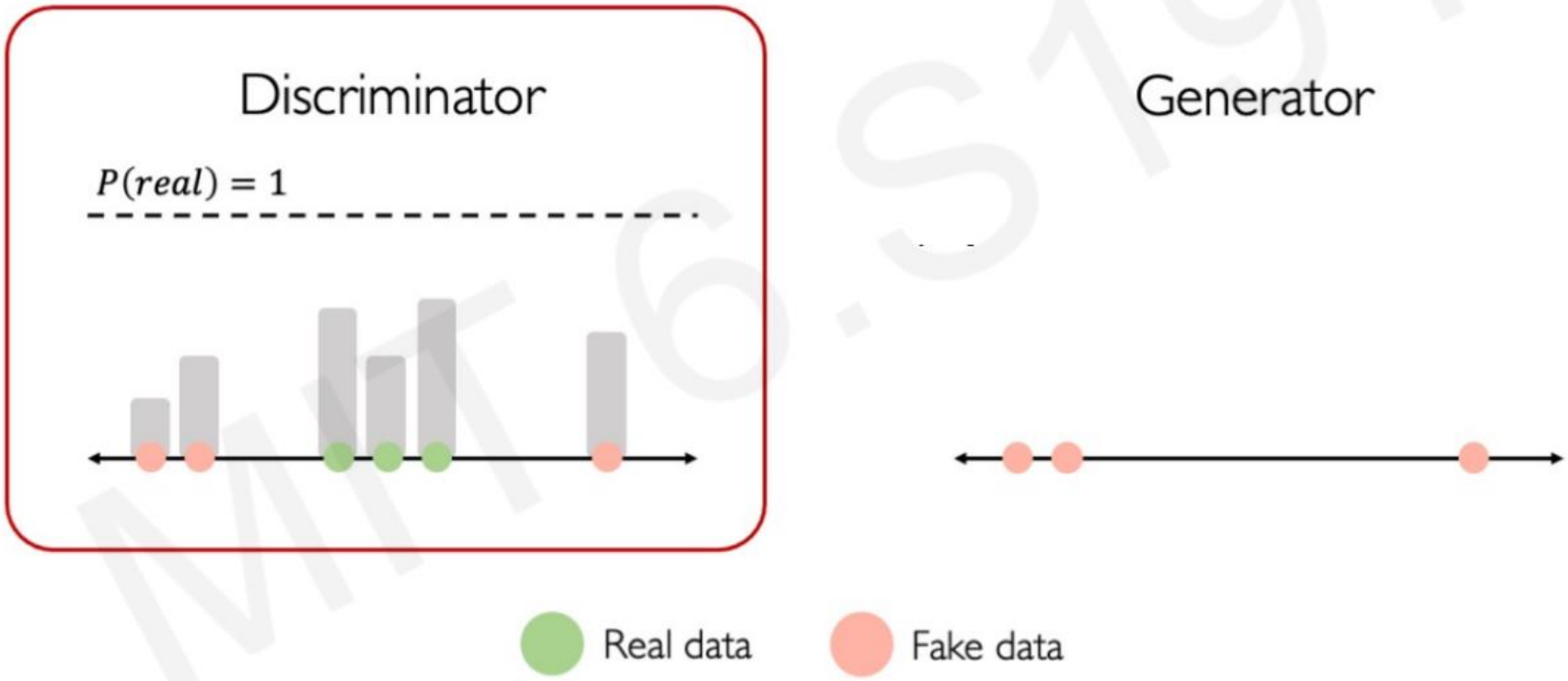


How does it work?



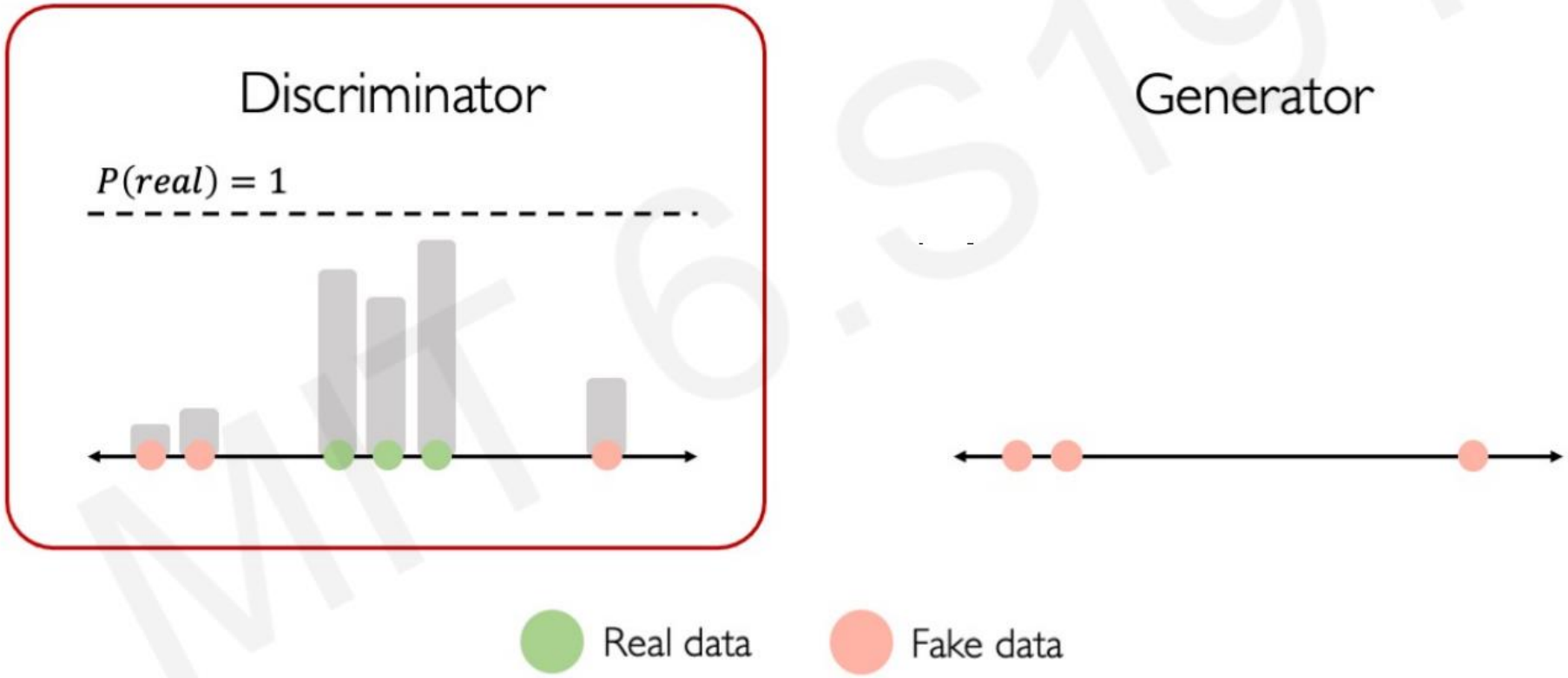
MIT 6.S191: Introduction to Deep Learning

How does it work?



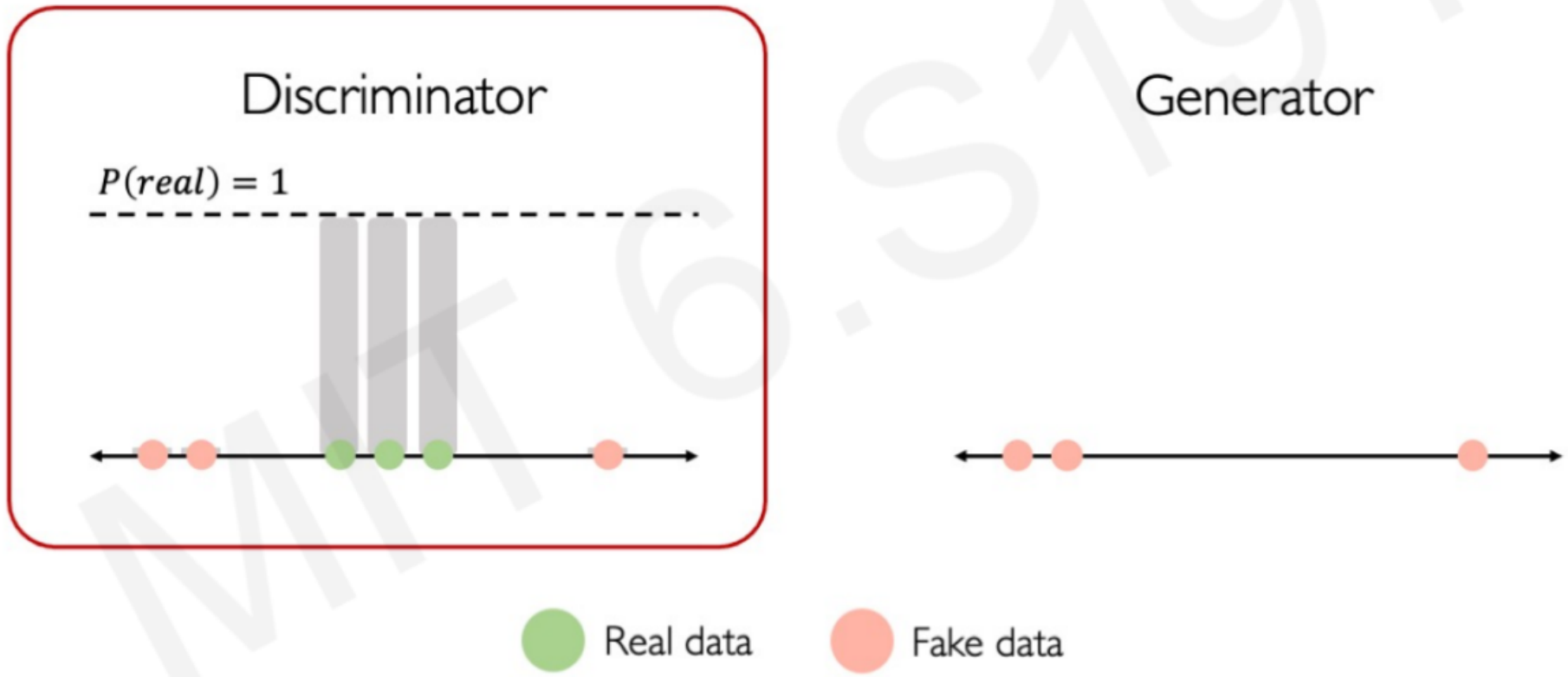
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



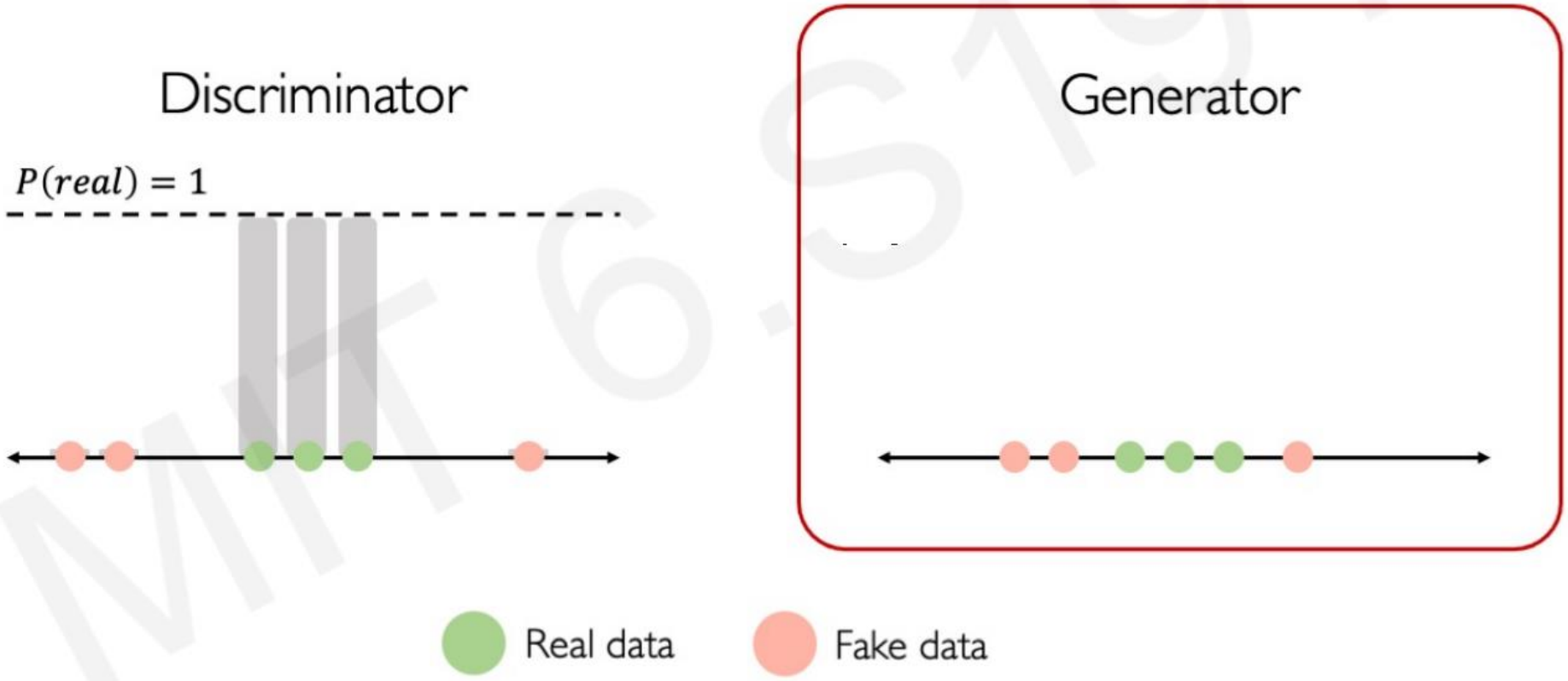
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



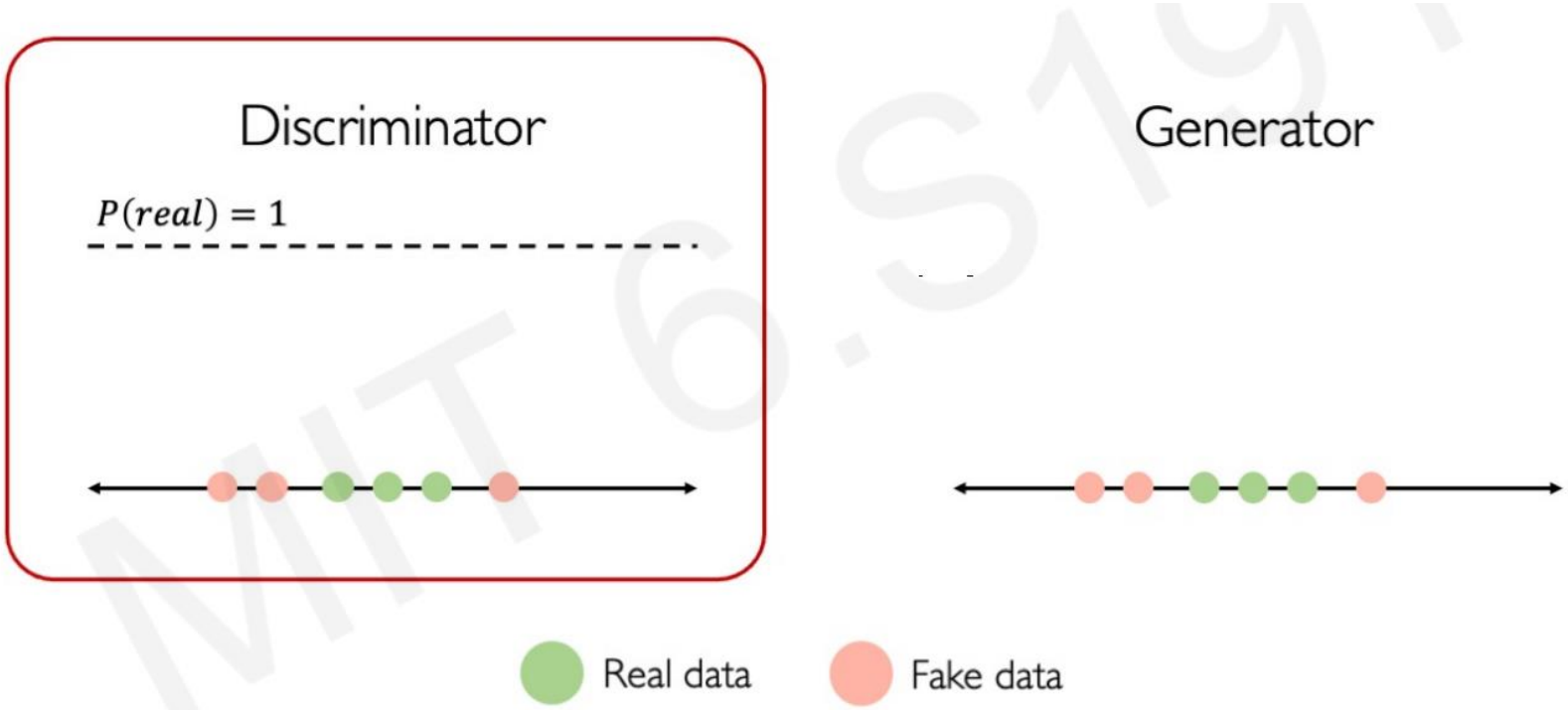
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



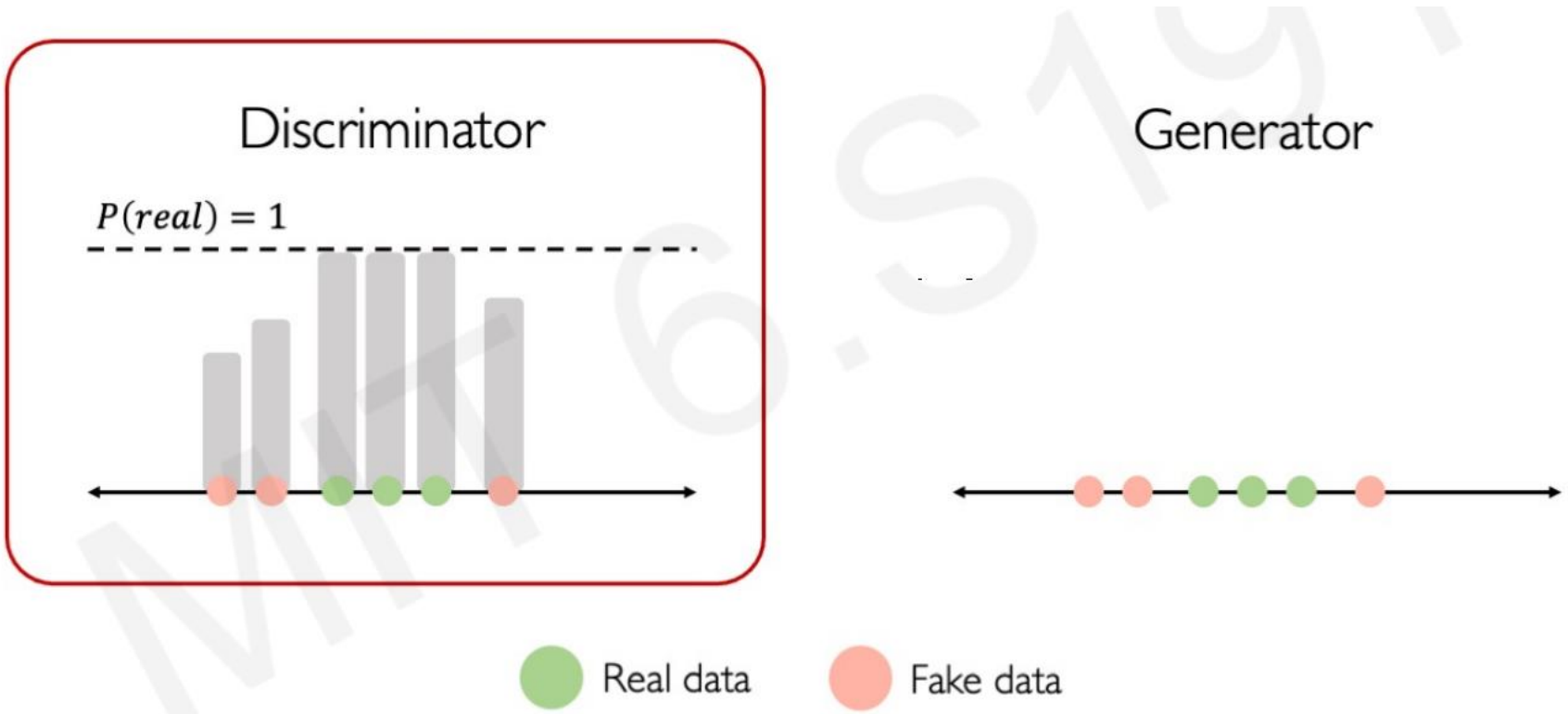
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



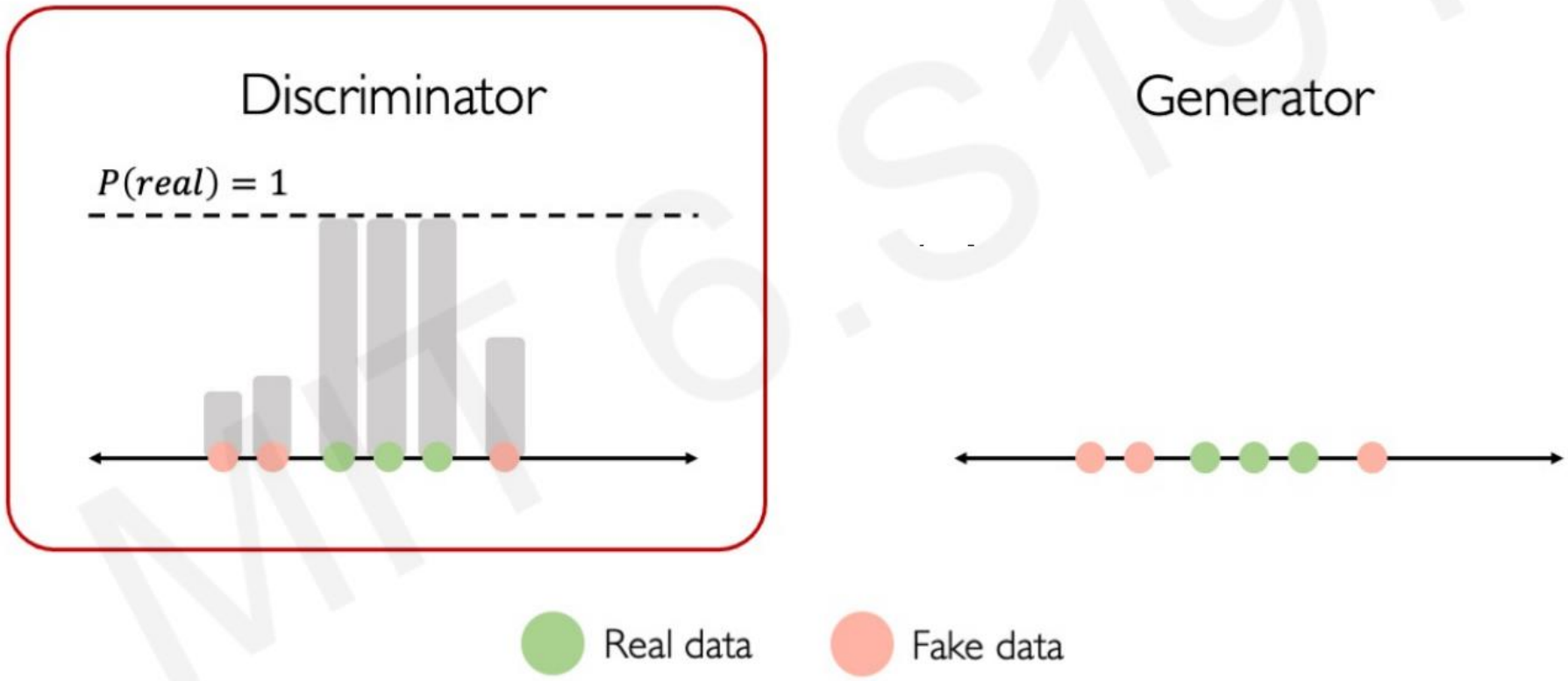
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



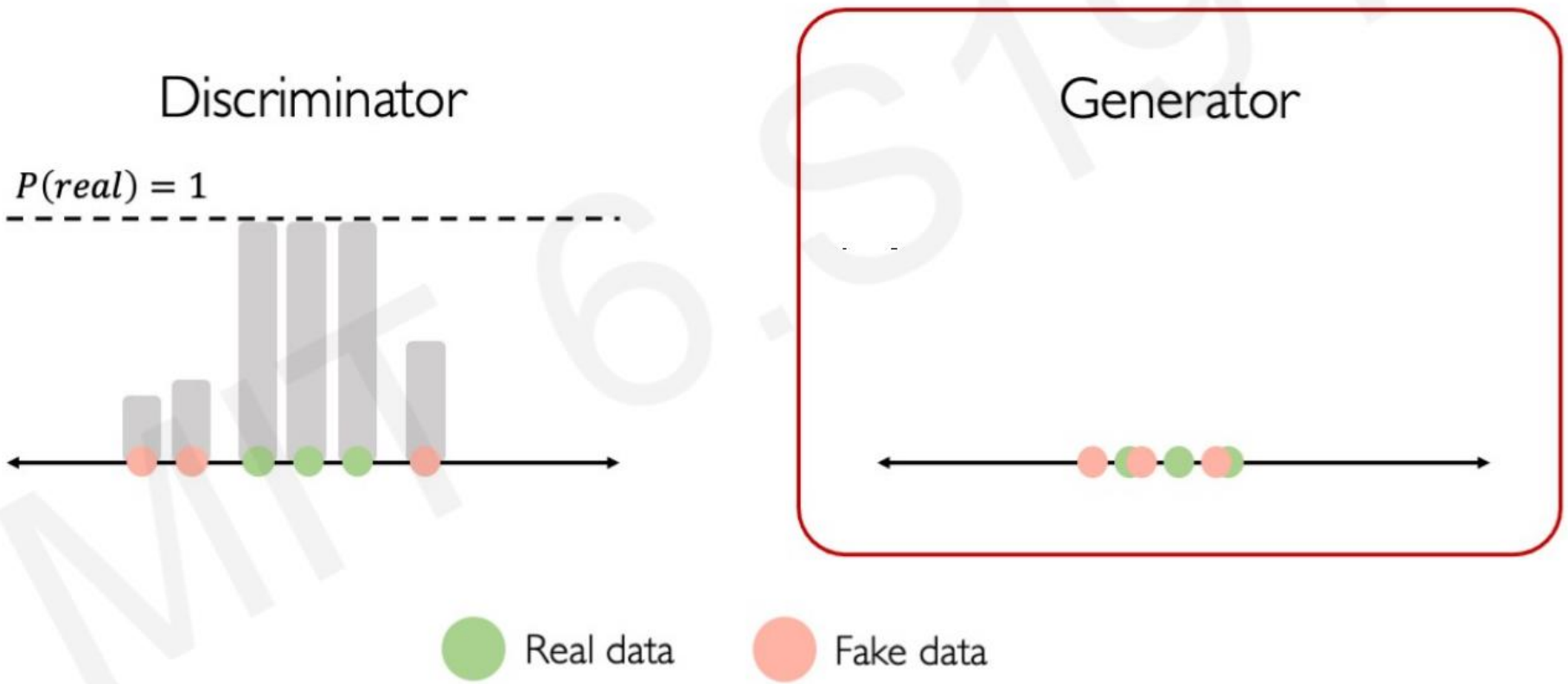
[MIT 6.S191: Introduction to Deep Learning]

How does it work?



[MIT 6.S191: Introduction to Deep Learning]

How does it work?



[MIT 6.S191: Introduction to Deep Learning]

How does it train?

Minimax game theory:

- Two players
- One wins only if one loses
- Widely use in two players games (chess, *Backgammon…*)
- Two players are called the maximizer and the minimizer
- Maximizer tries to get highest score
- Minimizer tries to get lowest score

How is the error function calculated?

CROSS-ENTROPY LOSS:

From discriminator,

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

How is the error function calculated?

CROSS-ENTROPY LOSS:

From generator,

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

How is the error function calculated?

CROSS-ENTROPY LOSS:

ALL NET,

$$\min_{G}\max_{D} V(D,G) = \underbrace{E_{x\sim p_{data}}[\log D(x)]}_{\text{Error from the discriminator model training}} + \underbrace{E_{z\sim Pz(z)}[\log(1 - D(G(z)))]}_{\text{Error from the combined model training}}$$

How can GANs be evaluated?

- Loss function?
- Is it objective or subjective?
- Warning→ does not converge!!!

Last trends

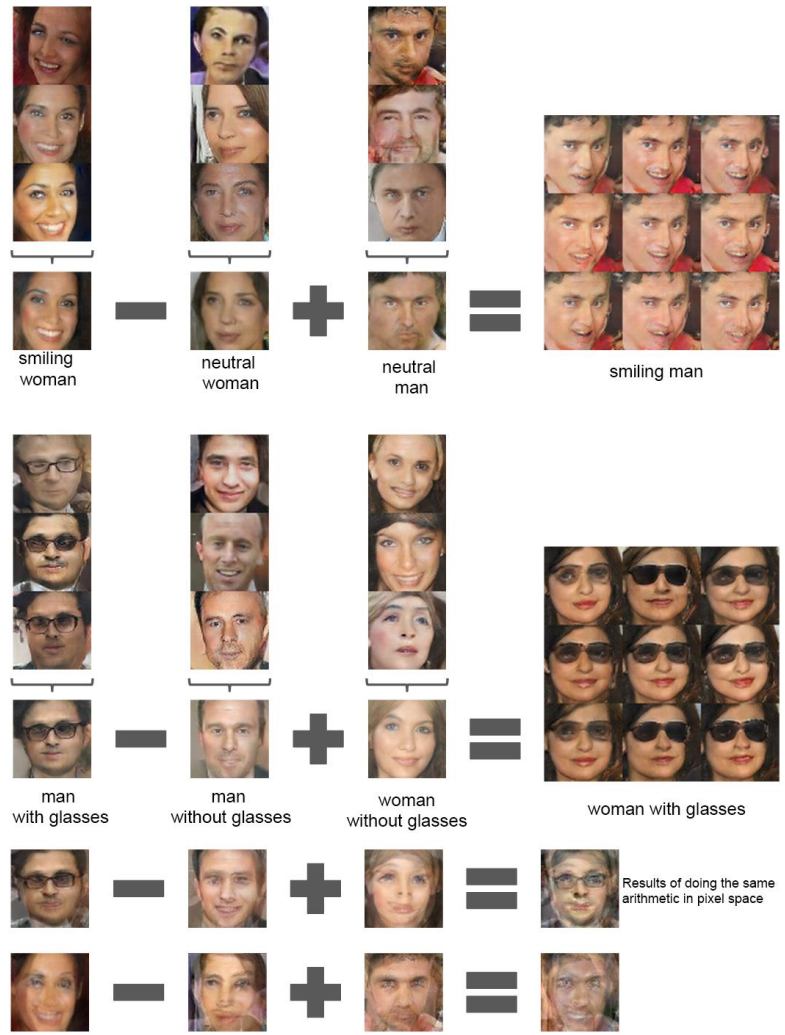
DCGAN deep convolutional generative adversarial networks [Radford]

- A series of restrictions on the architecture of the network are proposed and tested to make it more stable for any configuration in the trainings.
- The trained discriminators will be used for image classification tasks, testing the proper functioning for other unsupervised algorithms.
- The filters learned by the GANs are displayed and empirically show which filters have been learned to create specific objects.
- The generators have important arithmetic properties that allow easy manipulation of semantic qualities of the generated samples.

DCGAN

DCGAN

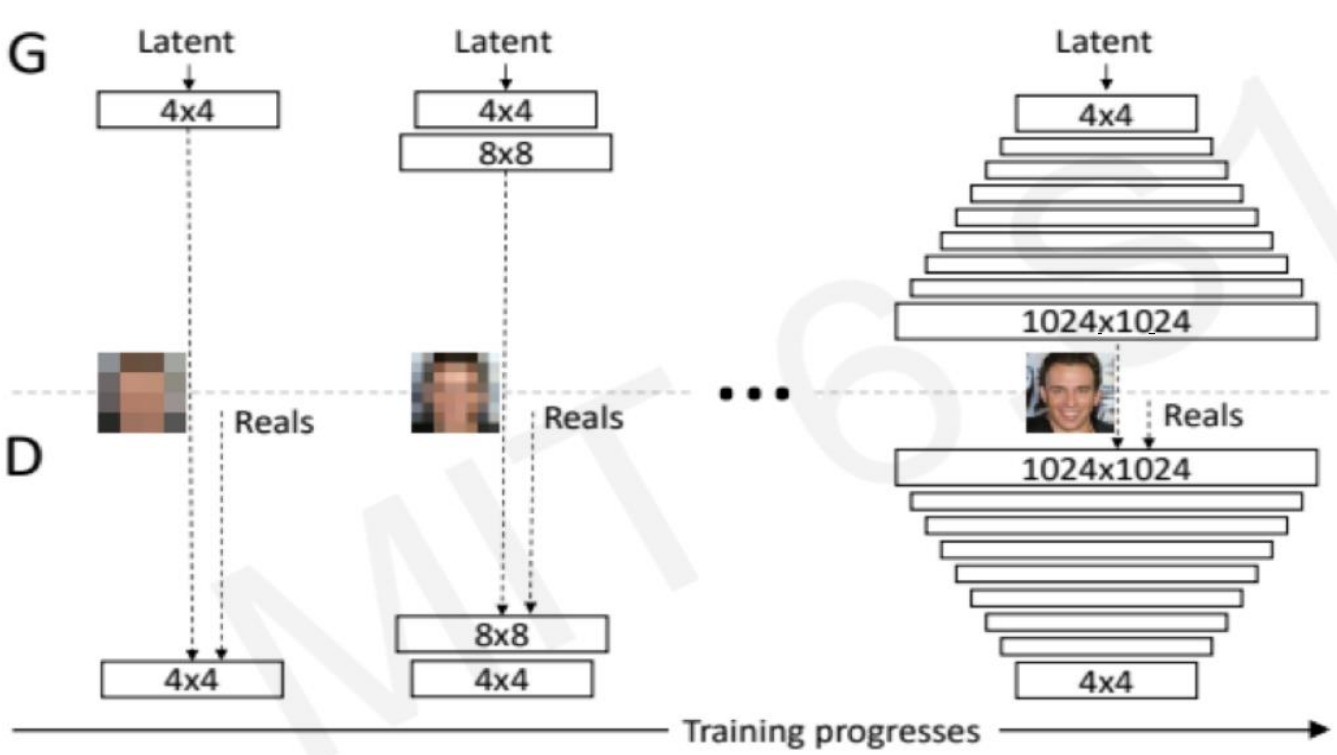
Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems



Progressive Growing Technique (Nvidia)

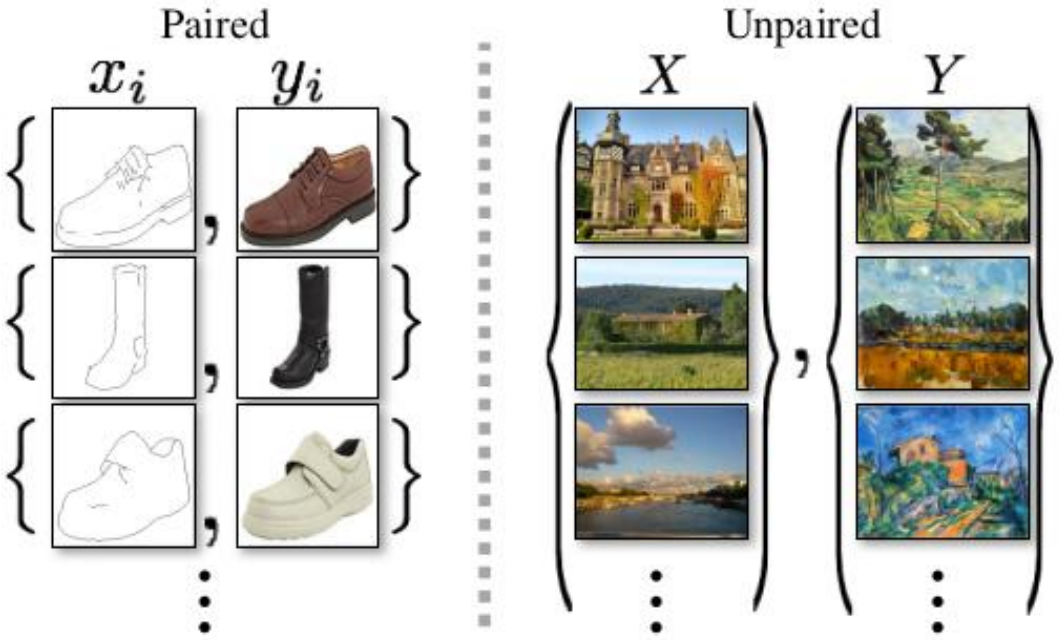
From Progressive Growing of GANs[5](5):

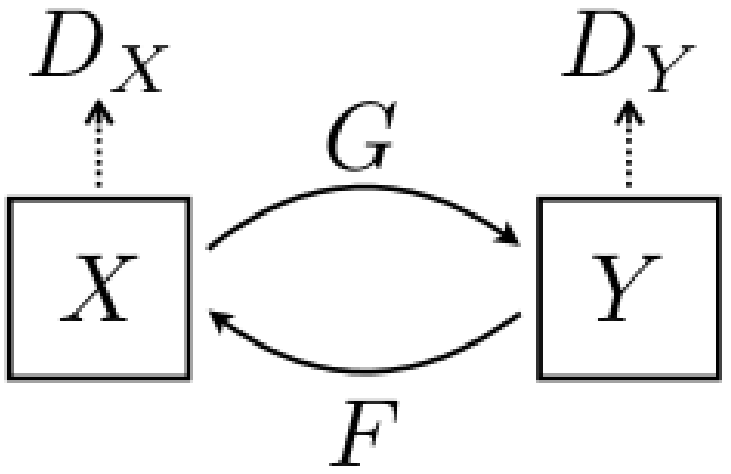
Progressive Growing Technique (Nvidia)



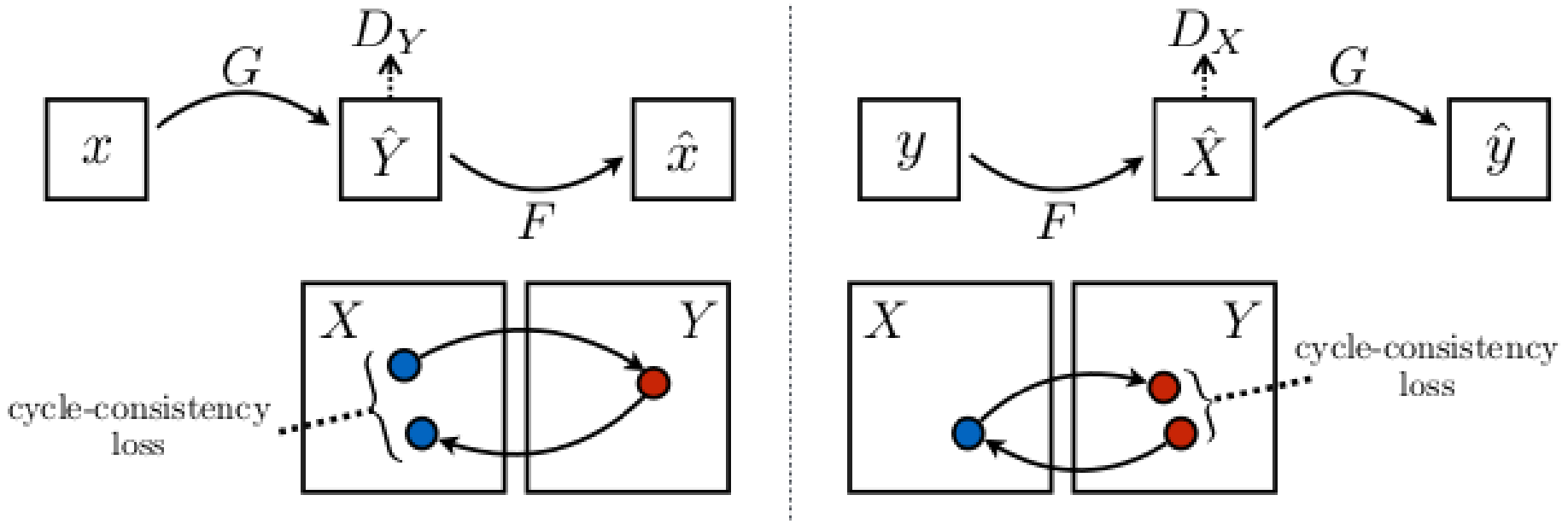
https://www.youtube.com/watch?v=G06dEcZ-QTg&feature=youtu.be

CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]



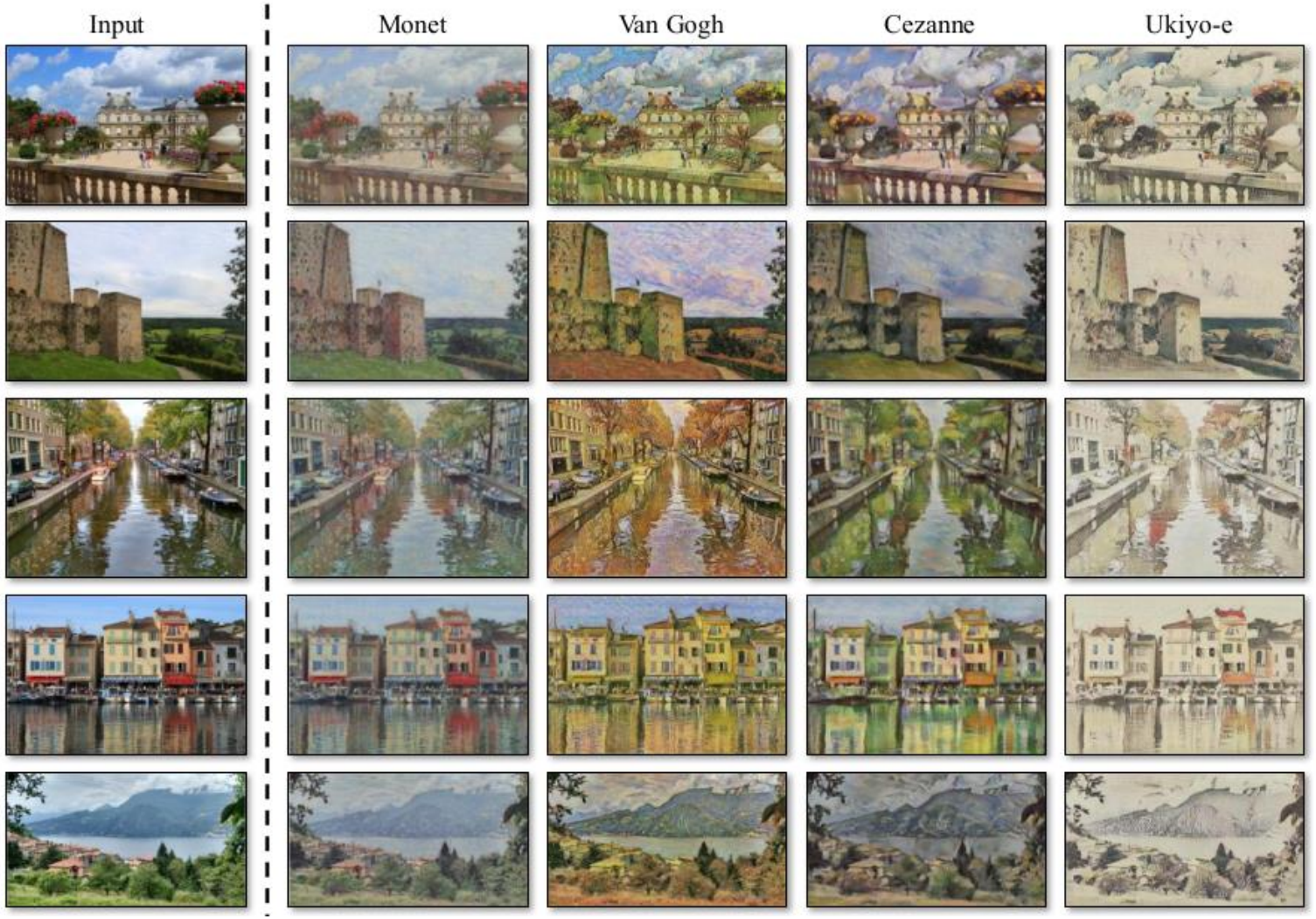


CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]



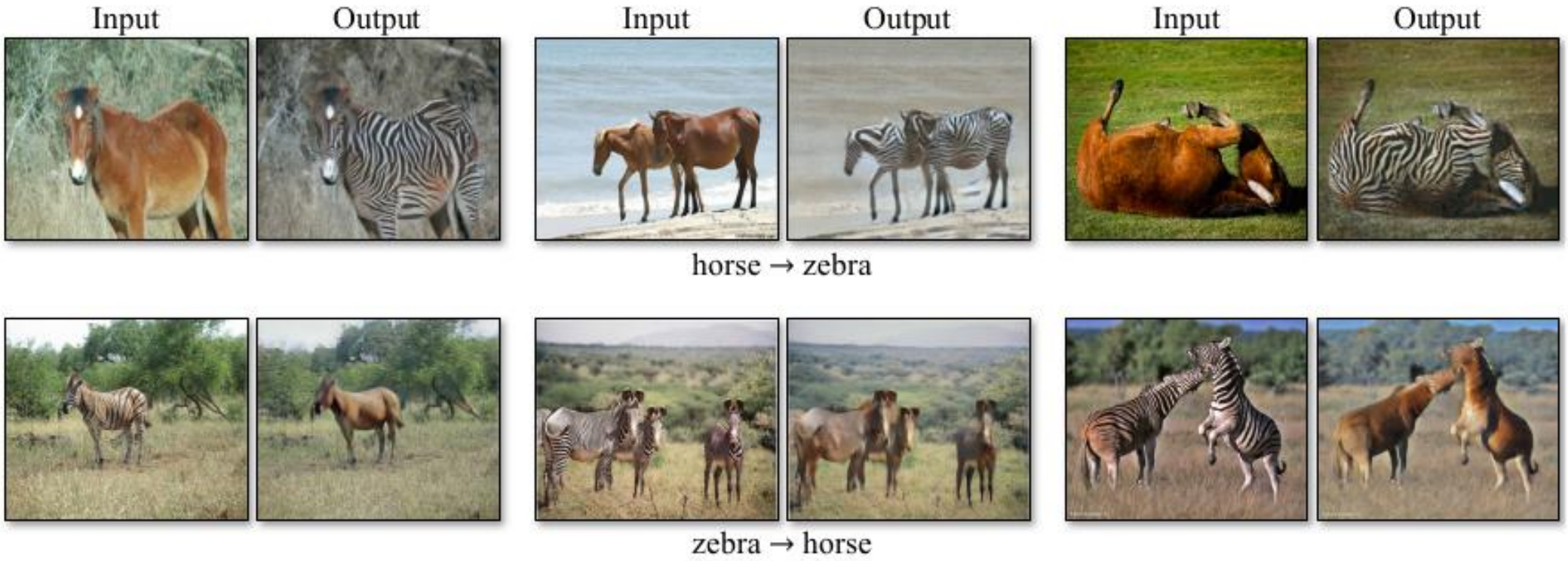
CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]

Style Transfer



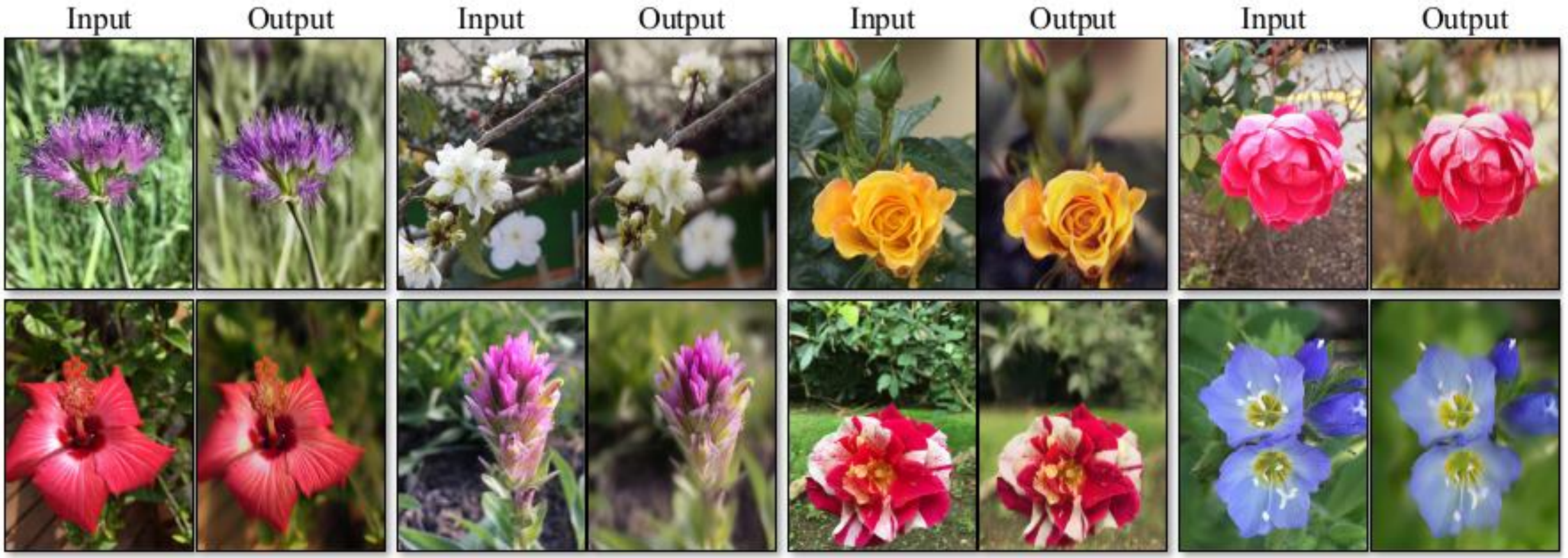
CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]

Object Transfiguration



CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]

Image enhancement



CYCLEGANS Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Zu-Park]

Season transfer

winter Yosemite → summer Yosemite

summer Yosemite → winter Yosemite

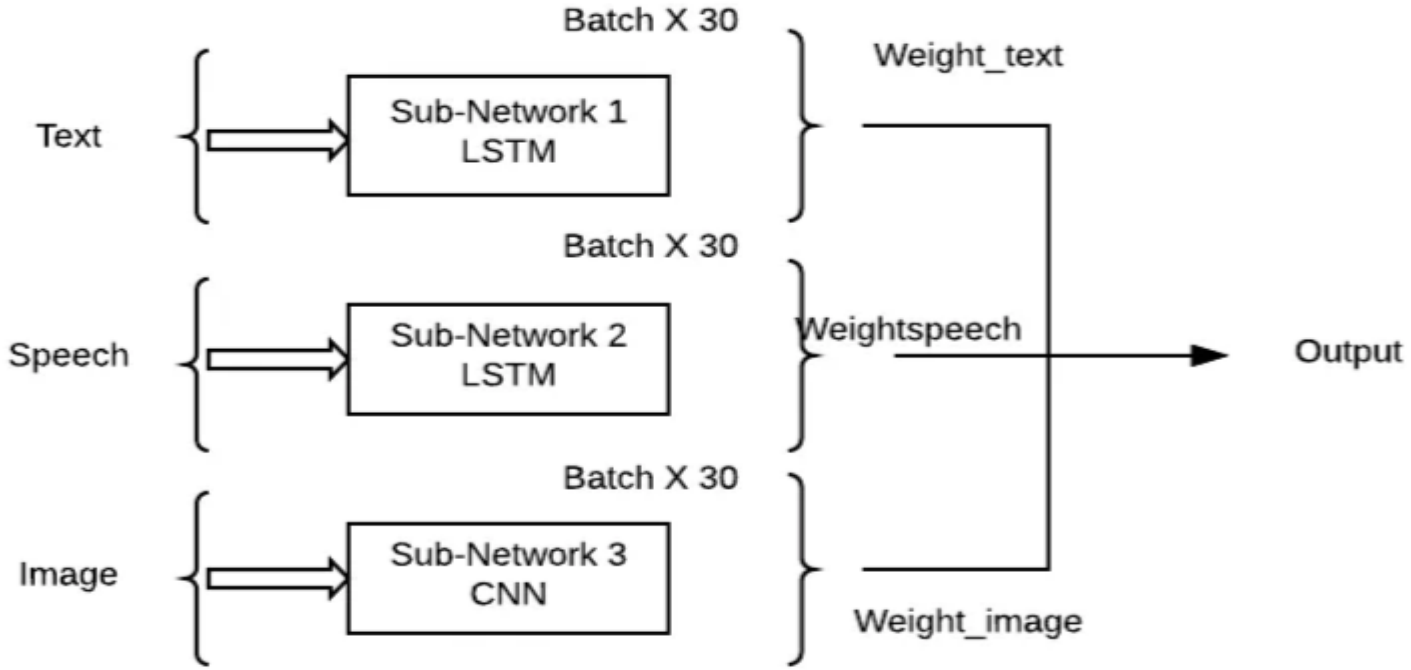
ChatGPT (Geerative pre-trained Transformers)

What is ChatGPT???

- Large Language Model
- Architecture based on → TRANSFORMERS
- Learn from the context
- Wide range aplication
- GPT3/GPT4…
- Multimodal Deep Learning

MULTIMODAL DEEP LEARNING

Why MULTIMODAL???



Multimodal Deep Learning. I recently submitted my thesis on… | by P Purvanshi Mehta | Towards Data Science

ChatGPT

Structure:

1. Token layer
2. Encoding layer
3. Transformer layer
4. Decoding Layer
5. Outputs generation layer

ChatGPT

Tokenization layer

HOW DOES IT WORKS?

Input: “The cat is in the garden."

Tokenization: [“The", “cat", “is", “in", “the", “garden."]

ChatGPT

Encoding layer

HOW DOES IT WORKS?

Input: “The cat is in the garden."

Tokenization: [“The", “cat", “is", “in", “the", “garden."]

Encoding: [vector_1, vector_2, vector_3, vector_4, vector_5, vector_6]

Token: "cat" Embedding vector: [0.1, 0.3, -0.2, ..., 0.6]

ChatGPT

Transformer layer

- ATTENTION LAYER (*“attention is all you need”* [Vaswani])
- FEED-FORWARD LAYER
- BATCH NORMALIZATION

ChatGPT

Transformer layer

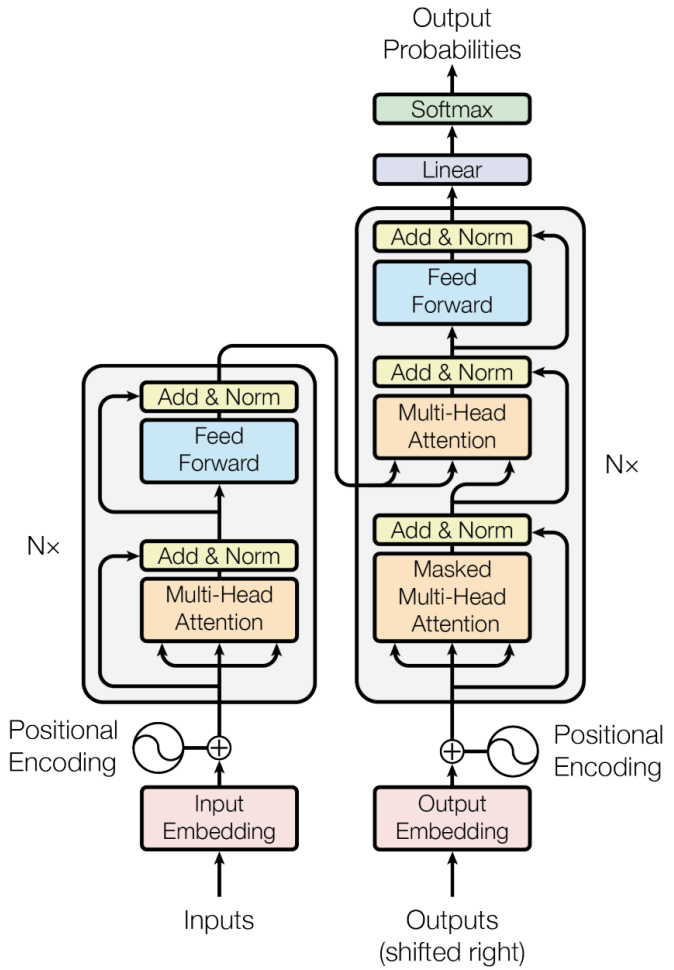


Figure 1: The Transformer - model architecture.

ChatGPT

Decoding layer

HOW DOES IT WORKS?

Input: [vector_1, vector_2, vector_3, vector_4, vector_5, vector_6]

Decoding: [“The", “cat", “is", “in", “the", “garden."]

ChatGPT

Outputs generation layer

It is an output layer that uses an autoregressive language model to generate the sequence of output tokens. This layer receives as input the semantic and syntactic context vector generated by the encoding layers, and uses a text generation strategy called "autoregressive decoding".

The output generation layer in ChatGPT uses an autoregressive language model to generate the sequence of output tokens, based on the semantic and syntactic context generated by the encoding layers and using an autoregressive decoding and sampling strategy.

DALL-E

AI system that can create realistic images and art from a description in natural language.