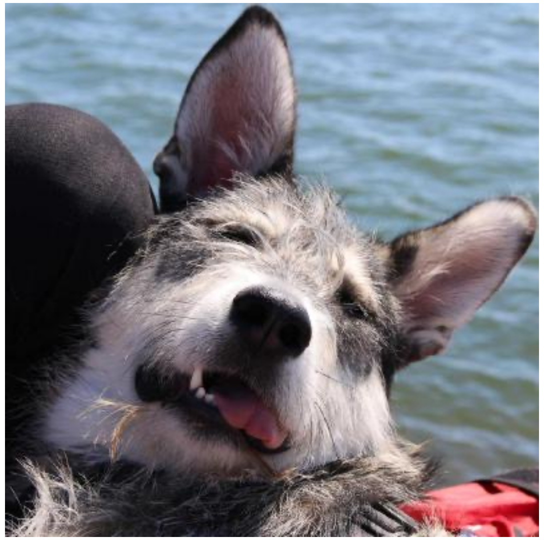
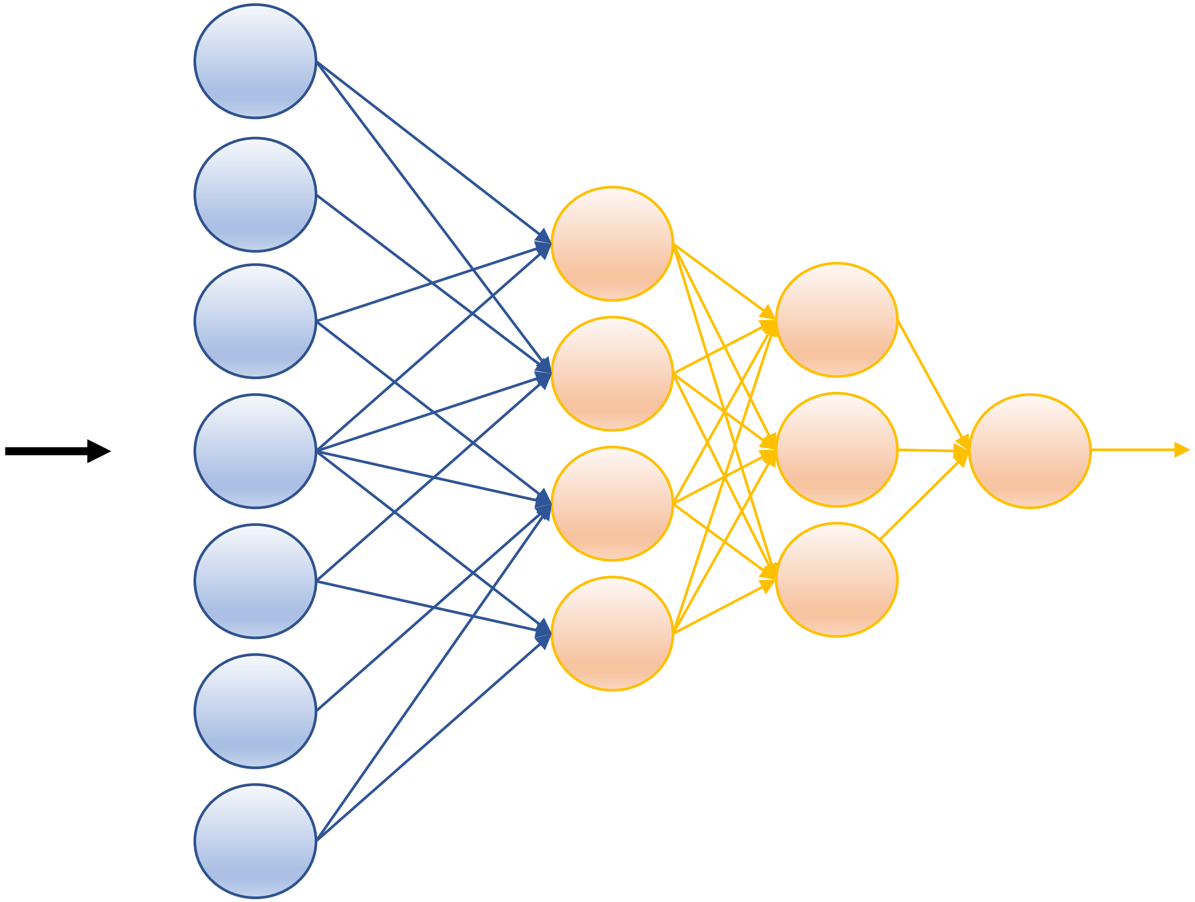
Generative Adversarial Network

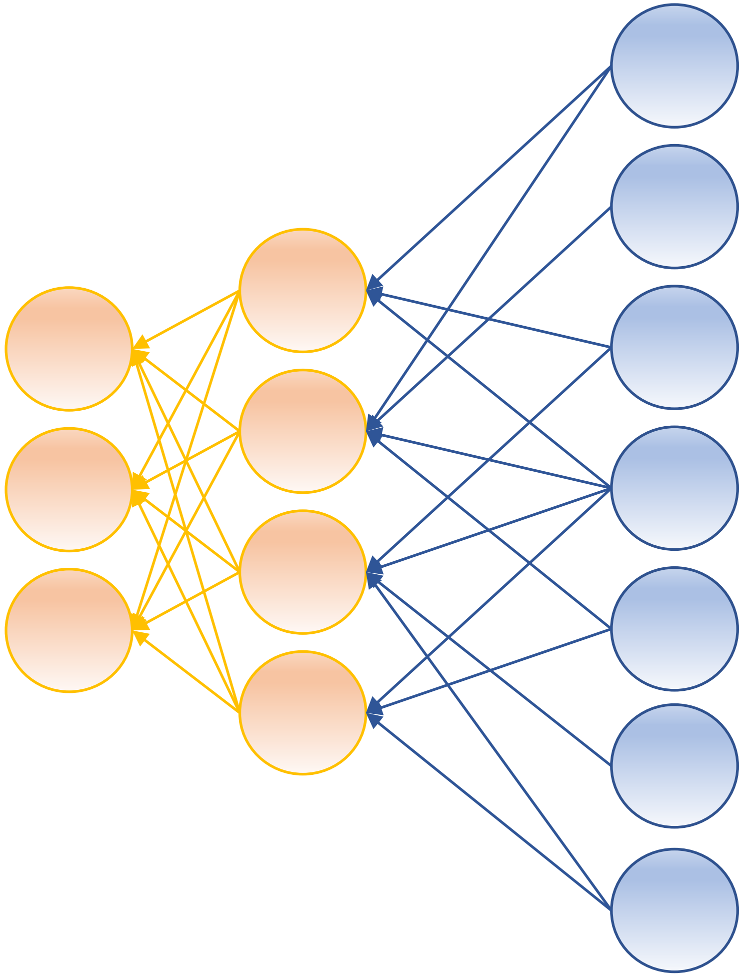
Image classifying ANN

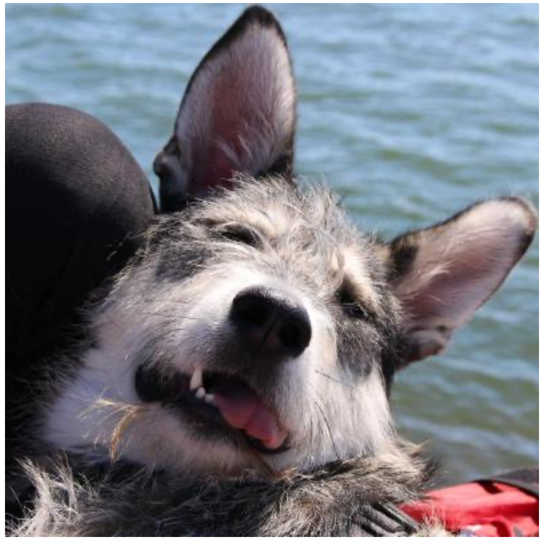




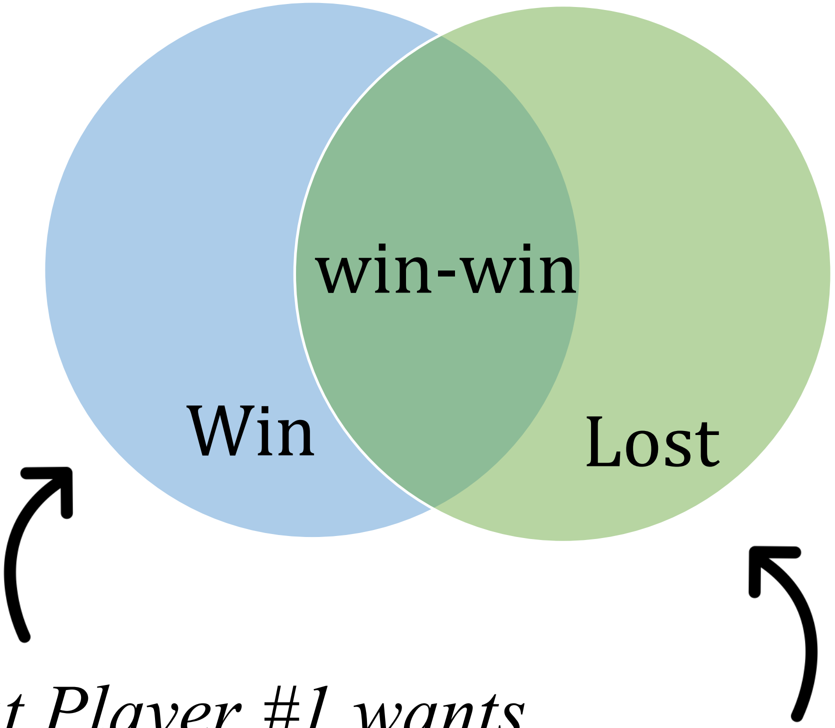
Pr(y=”Dog”)

Data (image) generating ANN





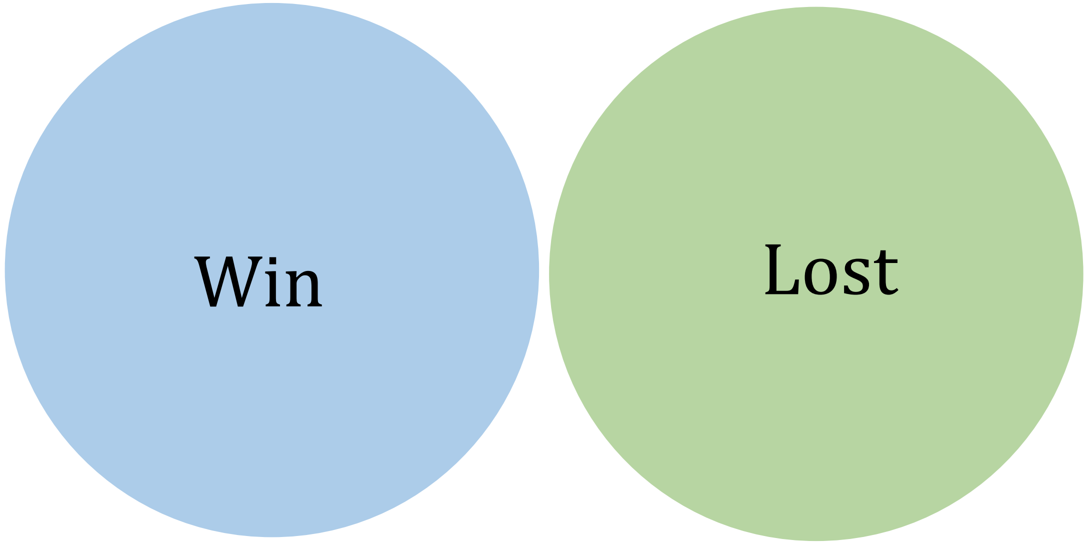
Non-zero sum game

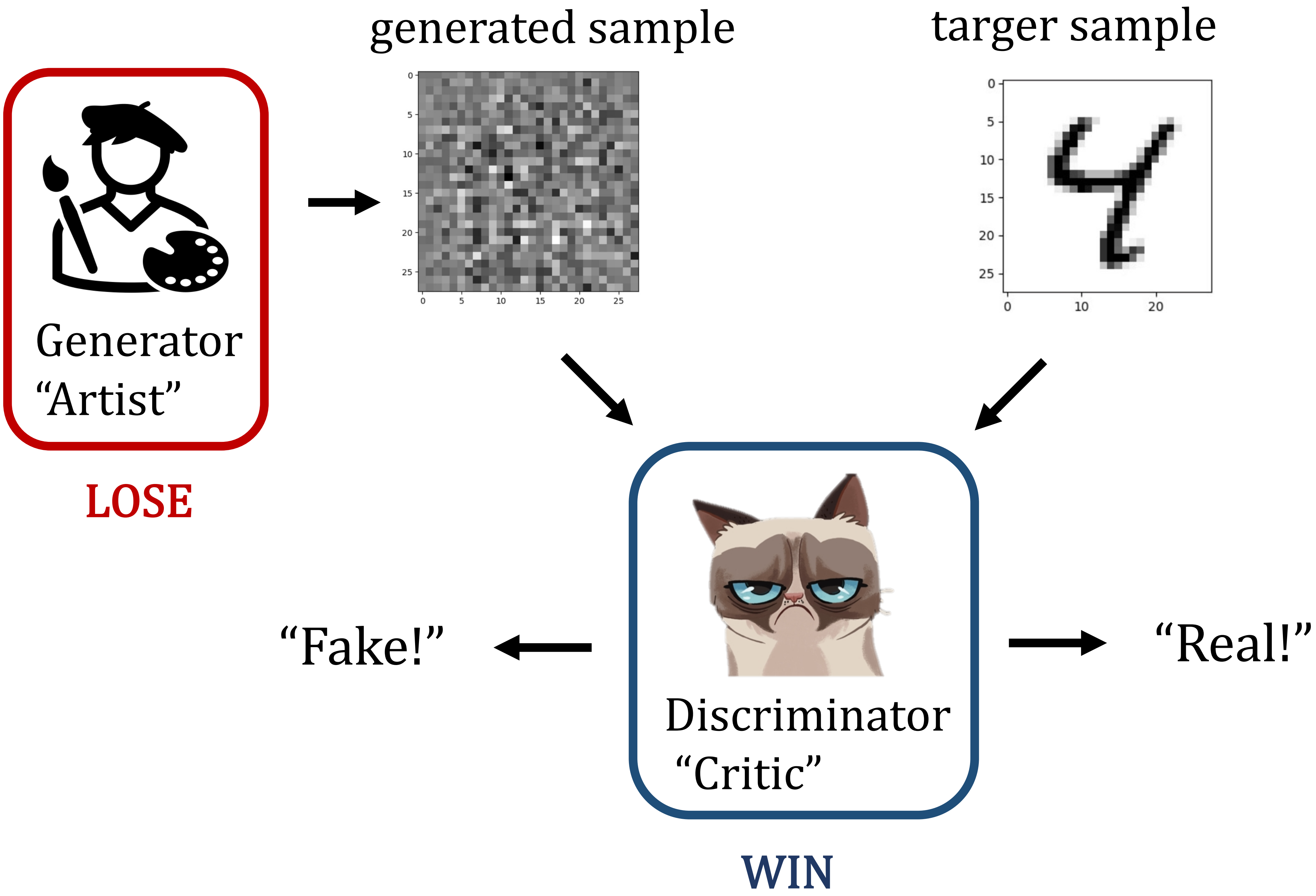


what Player #1 wants

what Player #2 wants

Zero sum game





generated sample

targer sample

Generator
“Artist”

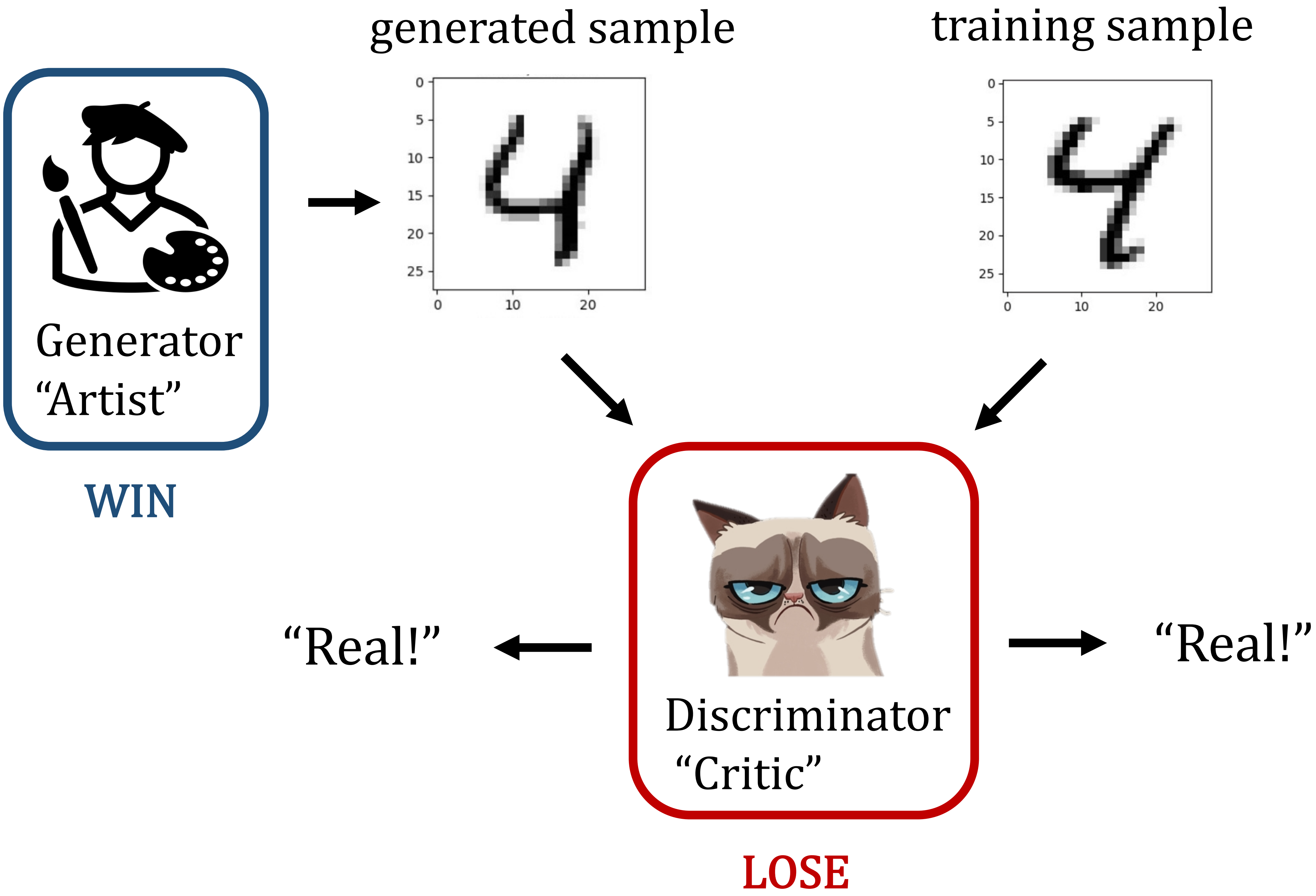
LOSE

Discriminator
“Critic”

WIN

“Fake!”

“Real!”



generated sample

training sample

Generator
"Artist"

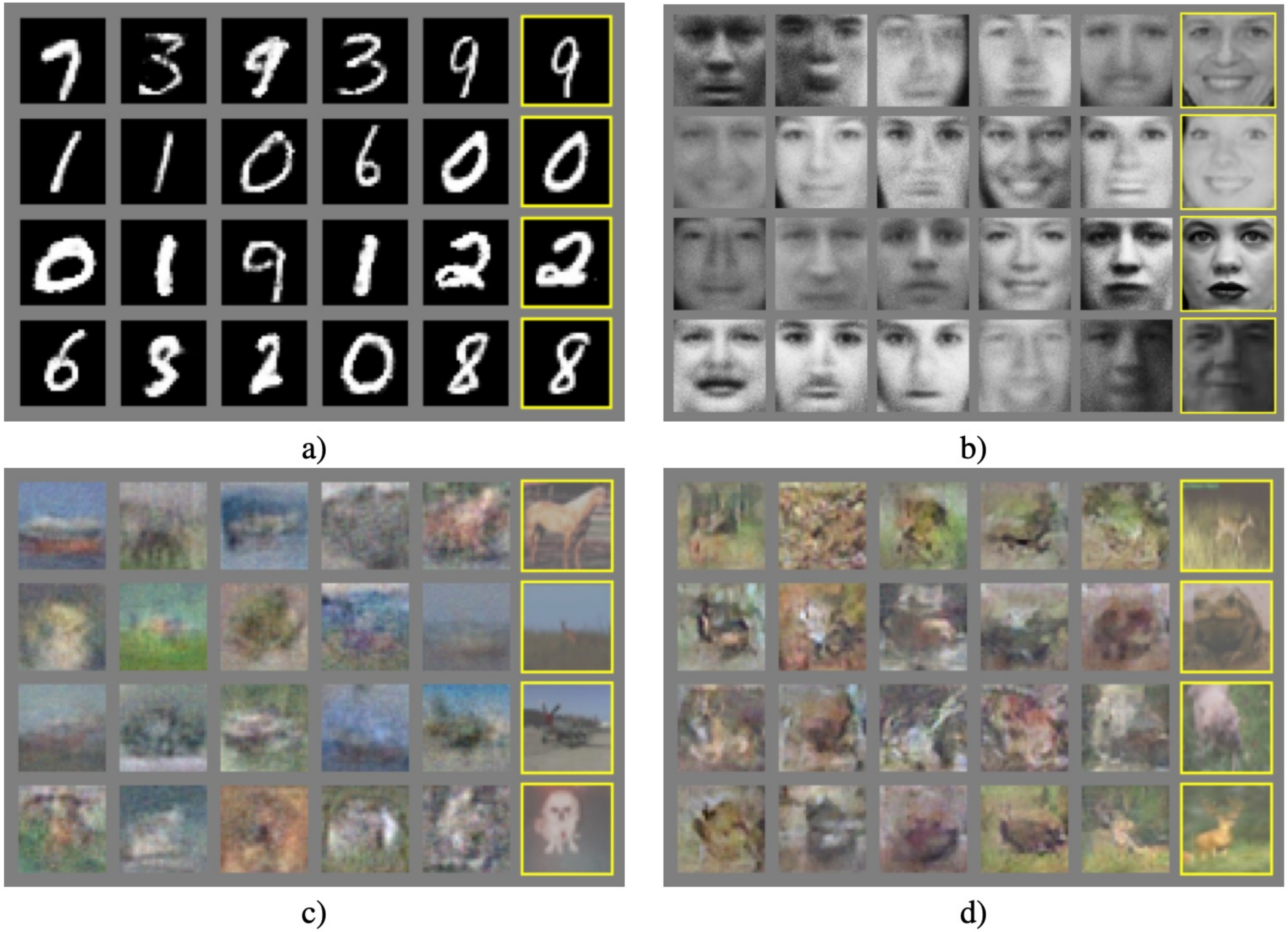
WIN

"Real!"

Discriminator
"Critic"

"Real!"

LOSE



a)

b)

c)

d)

“Generative Adversarial Nets” Ian Goodfellow 2014

GAN PROGRESS ON FACE GENERATION

Source: Goodfellow et al., 2014; Radford et al., 2016; Liu & Tuzel, 2016; Karras et al., 2018; Karras et al., 2019; Goodfellow, 2019; Karras et al., 2020; AI Index, 2021

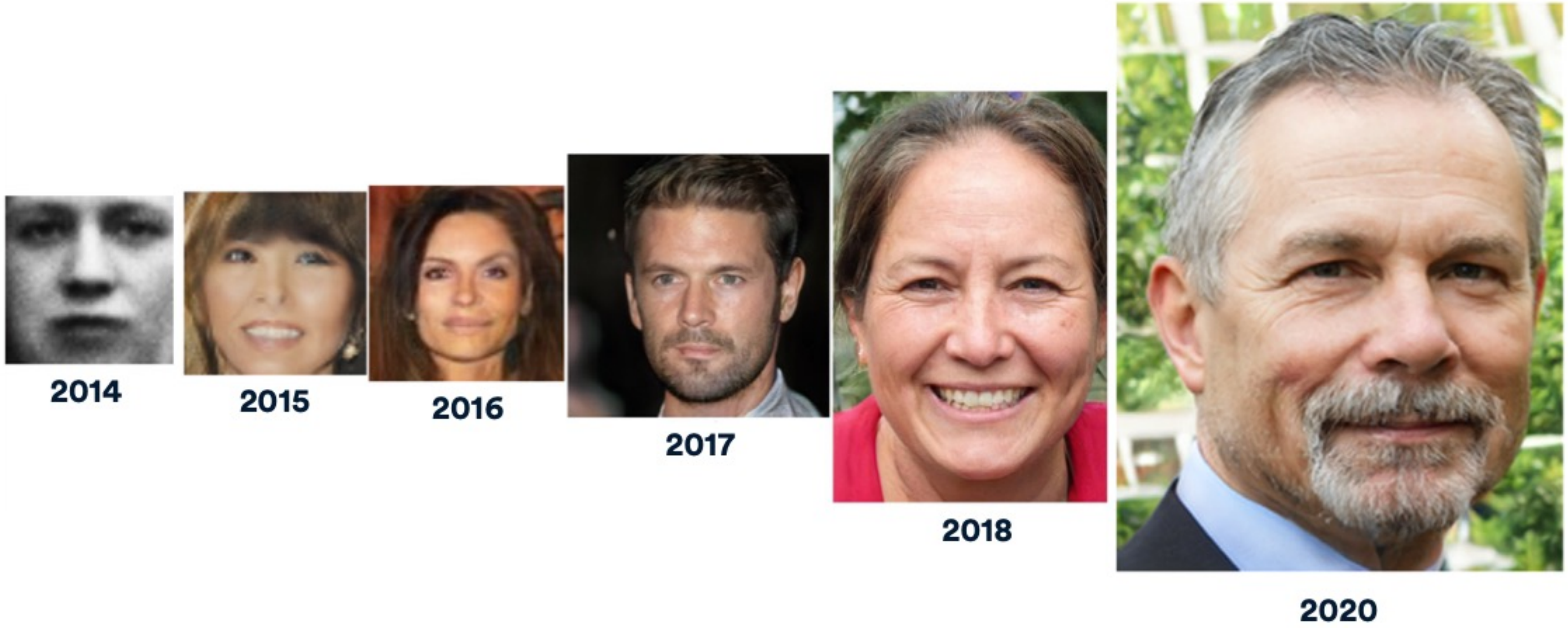
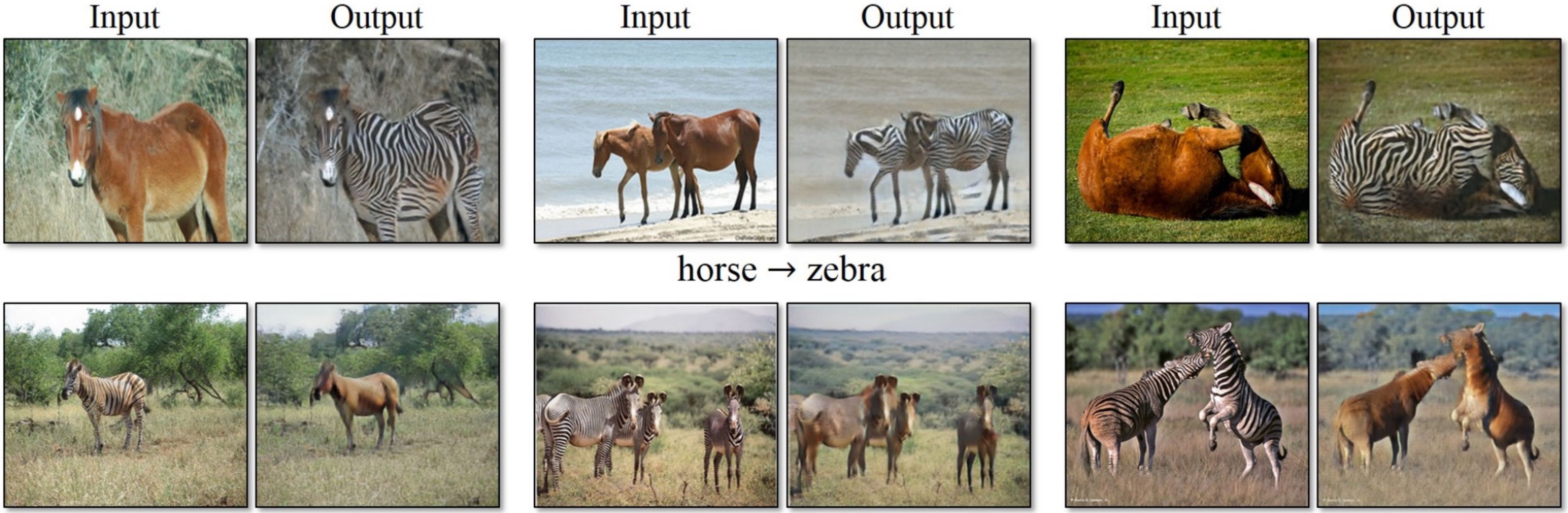


Figure 2.1.7

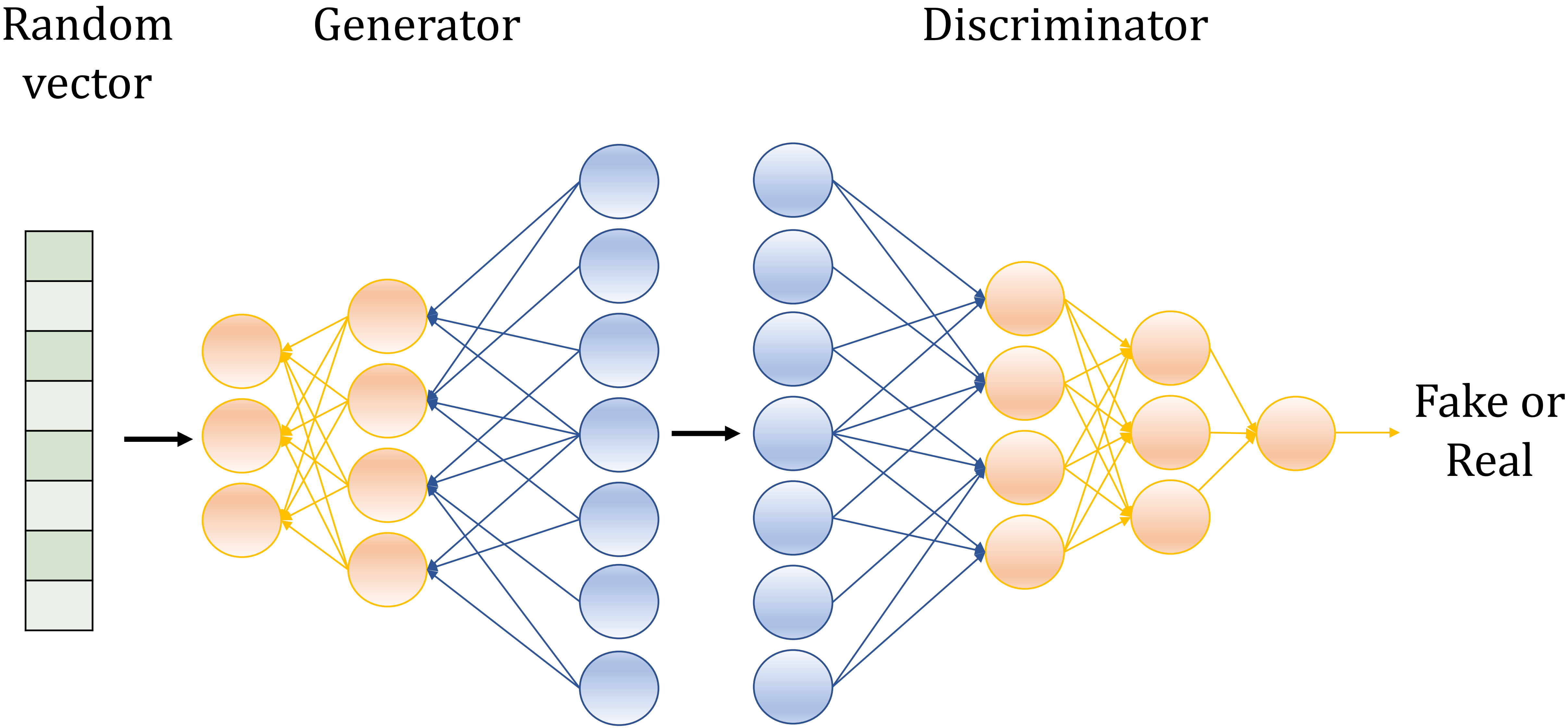
https://aiindex.stanford.edu/wp-content/uploads/2021/11/2021-AI-Index-Report_Master.pdf https://thispersondoesnotexist.com



horse → zebra

zebra → horse

https://junyanz.github.io/CycleGAN/



Random vector

Generator

Discriminator

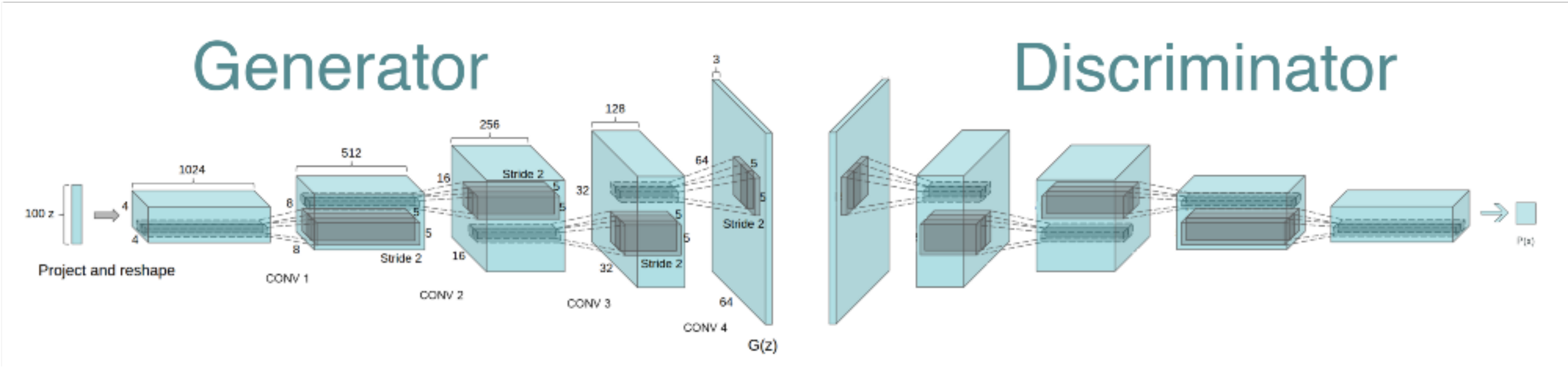
Fake or Real

```
codings_size = 100  # random vector
gen_out_size = 2    # shape of the generated sample

generator = tf.keras.models.Sequential([
   tf.keras.layers.Dense(…, input_shape=[codings_size]),
   tf.keras.layers.Dense(…),
   tf.keras.layers.Dense(gen_out_size)
])

discriminator = tf.keras.models.Sequential([
   tf.keras.layers.Dense(…, input_shape=[gen_out_size ]),
   tf.keras.layers.Dense(…),
   tf.keras.layers.Dense(1, activation=‘sigmoid’)
])
```

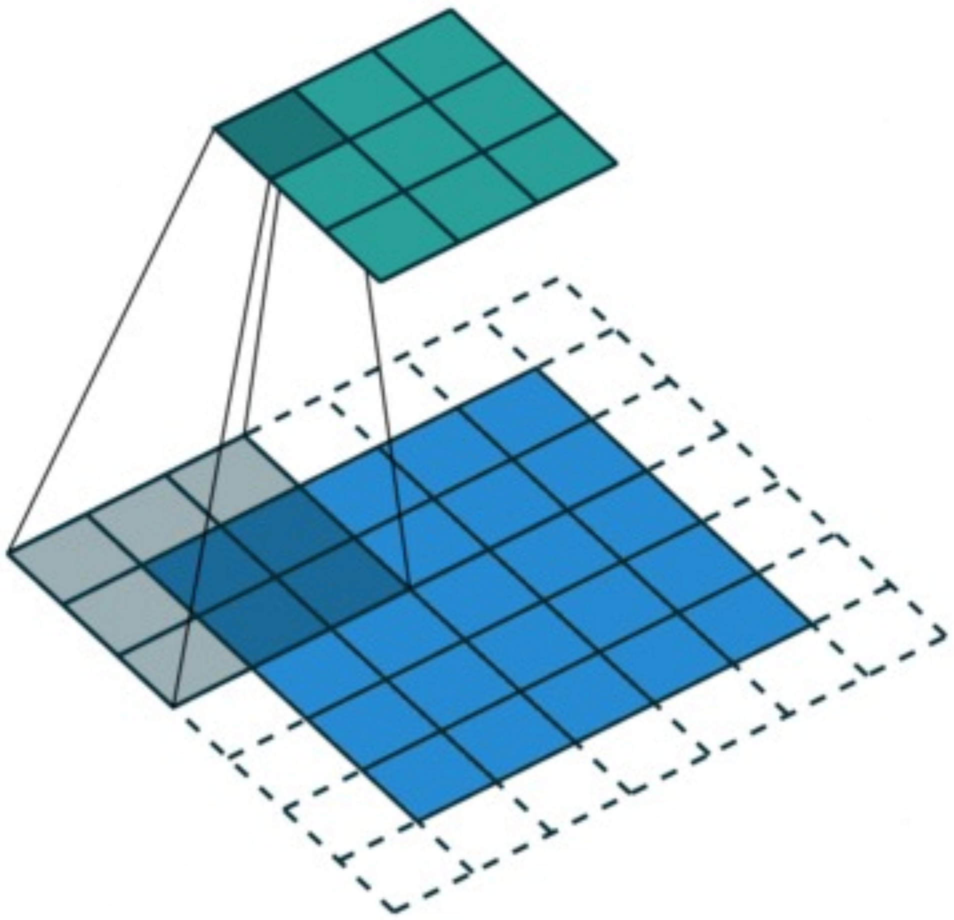
DCGAN – Deep Convolutional GAN



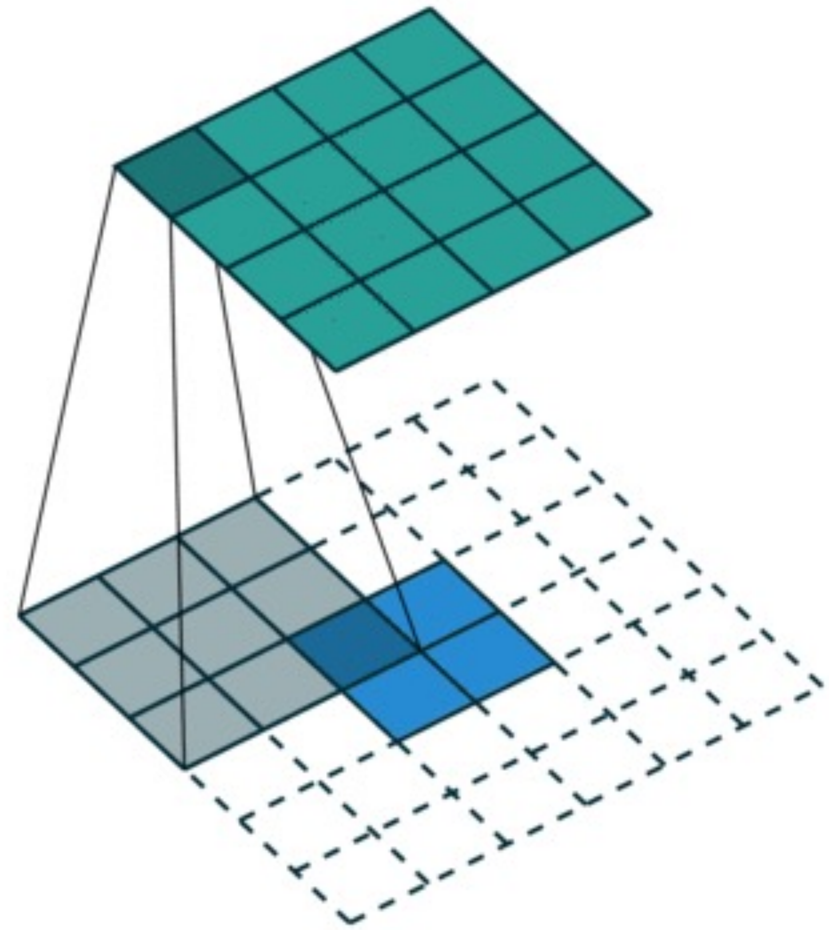
DCGAN – Convolution

output

input



DCGAN – Transposed convolution



output

input

Hands on: Generation MNIST digits images