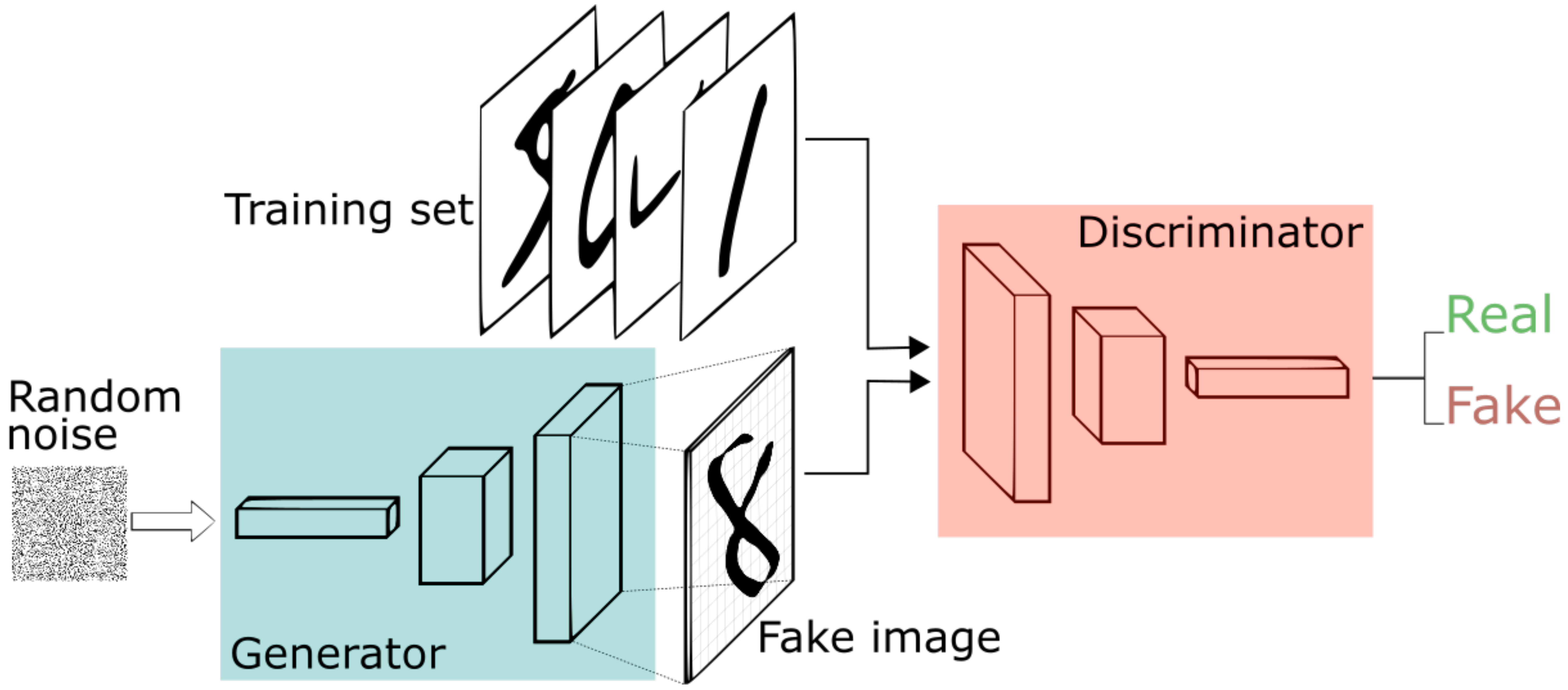
Generative Adversarial Networks (GANs)

Analogy

- Generative: team of counterfeiters, trying to fool police with counterfeit money
- Discriminative: police trying to detect the counterfeit money
- Competition drives to improve both, until counterfeits are virtually indistinguishable from genuine currency.
- Now counterfeiters have, as a side effect, learned something about real currency.

Big Idea

- Train a generative model G(z) to generate data with random noise, z, as input
- Adversary is discriminator D(x) which is trained to distinguish synthetic and true data
- Represent G(z) & D(z) as multilayer perceptrons for differentiability



```python
def make_generator_model():
    model = tf.keras.Sequential()
    model.add(layers.Dense(7*7*256, use_bias=False, input_shape=(100,)))
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())

    model.add(layers.Reshape((7, 7, 256)))
    assert model.output_shape == (None, 7, 7, 256)  # Note: None is the batch size

    model.add(layers.Conv2DTranspose(128, (5, 5), strides=(1, 1), padding='same', use_bias=False))
    assert model.output_shape == (None, 7, 7, 128)
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())

    model.add(layers.Conv2DTranspose(64, (5, 5), strides=(2, 2), padding='same', use_bias=False))
    assert model.output_shape == (None, 14, 14, 64)
    model.add(layers.BatchNormalization())
    model.add(layers.LeakyReLU())

    model.add(layers.Conv2DTranspose(1, (5, 5), strides=(2, 2), padding='same', use_bias=False,
              activation='tanh'))
    assert model.output_shape == (None, 28, 28, 1)

    return model
```

```python
def make_discriminator_model():
    model = tf.keras.Sequential()
    model.add(layers.Conv2D(64, (5, 5), strides=(2, 2), padding='same',
                                     input_shape=[28, 28, 1]))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))

    model.add(layers.Conv2D(128, (5, 5), strides=(2, 2), padding='same'))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))

    model.add(layers.Flatten())
    model.add(layers.Dense(1))

    return model
```

Probability that discriminator identifies real data as real

$$\min_{G} \max_{D} E_{z,x} \left[\log D(G(z)) + \log(1 - D(x)) \right]$$

Probability that discriminator identifies generated image as fake

Global Optimum: A global minimum of this minimax function

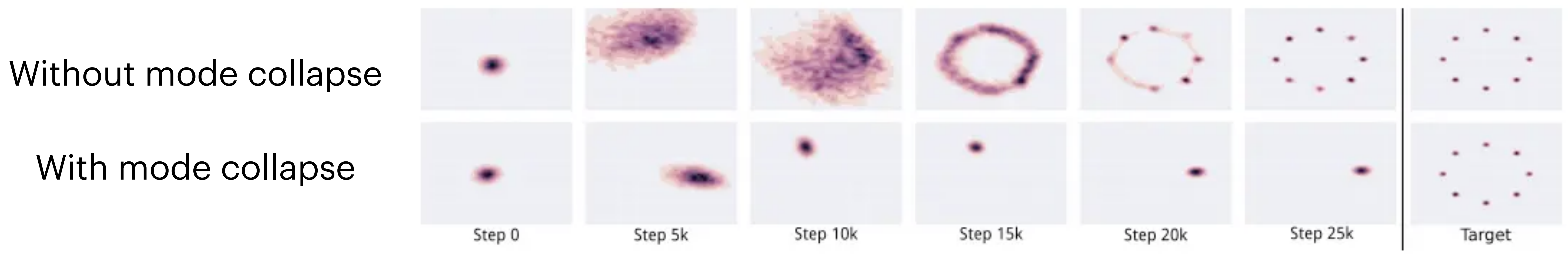
Training GANs

Forward propagation (generation & classification)
Backward propagation (adversarial training)
Fake Data
Real Data
Use generator to create fake data from random input
Use discriminator to separate true data from real data
Gaussian Noise
Generative Network
Discriminative Network
Update the generator's weights to increase classification error
Update the discriminator's weights to decrease classification error
Input random variable (Drawn from a simple distribution, e.g gaussian noise)
The generative network is trained to **maximise** the final classification error
The **generated distribution** and **true distribution** are not compared directly
The discriminative network is trained to **minimise** the final classification error
The classification error is the reference metric for the training of both networks

Drawbacks of using GANs

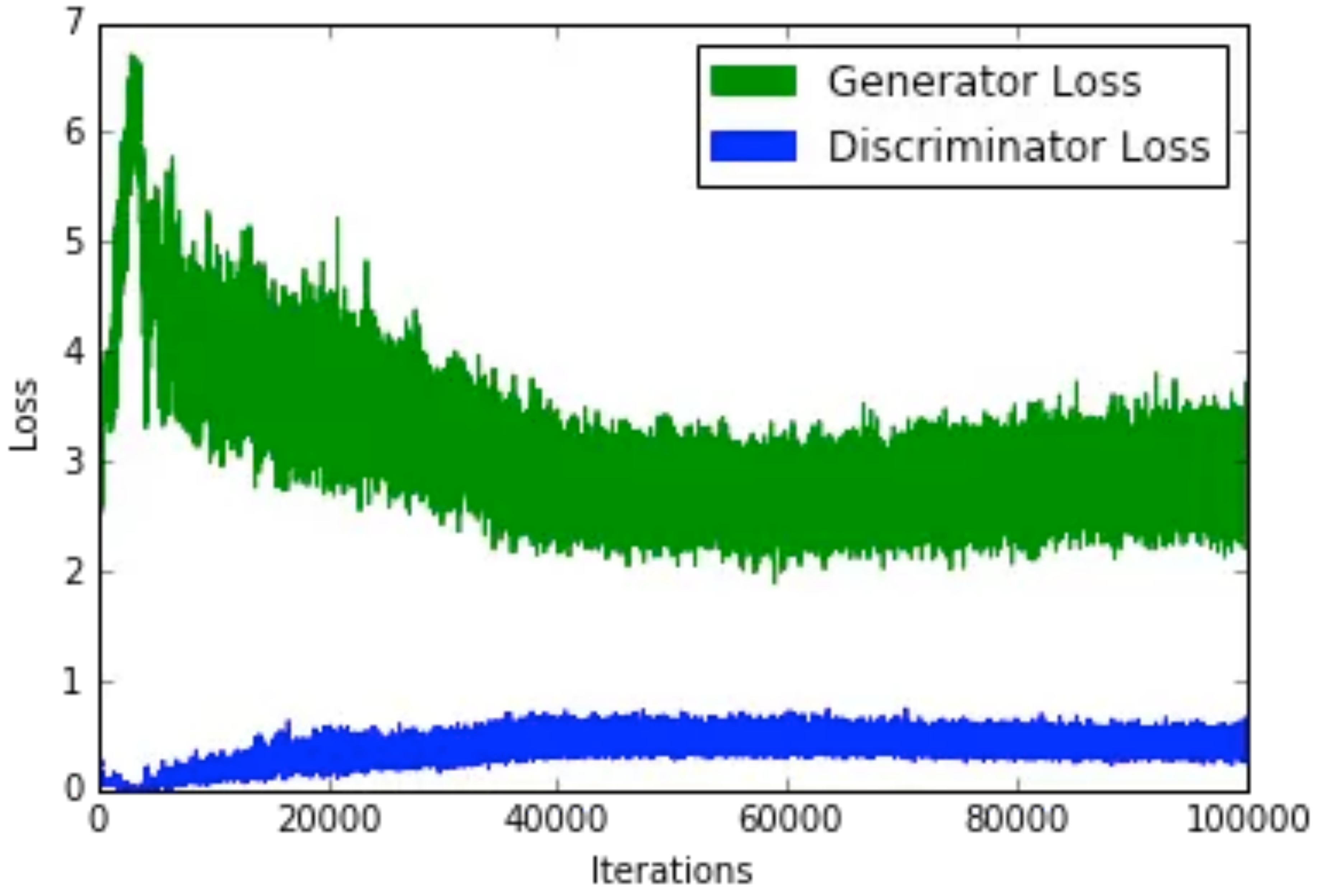
Mode Collapse

- Natural data distributions are highly complex and multimodal.
- During mode collapse, the generator produces samples that belong to a limited set of modes. The generator believes that it can fool the discriminator by locking on to a single mode, so it produces only samples from this exclusive mode.
- The discriminator eventually figures this out. Generator just locks on to another mode.
- Cycle repeats



Convergence
When do we stop training?

Since the generator loss improves when the discriminator loss degrades (and vice-versa), we cannot judge convergence based on the value of the loss function.



Typical plot of a GAN loss function. Note how convergence cannot be interpreted from this plot

Quality

- It is difficult to quantitatively tell when the generator is producing high quality samples.
- Additional perceptual regularisation added to the loss function can help mitigate the situation to some extent.

Metrics

- The GAN objective function explains how well the Generator or the Discriminator is performing with respect to its adversary.
- It does not represent the quality or diversity of its output.
- Hence we need distinct metrics that can measure these.

Techniques for improving Performance

- Alternative loss functions: Replace Jensen Shannon divergence of conventional GANs with Wasserstein Loss
- Two Timescale Update Rule (TTUR): Use a different learning rate for D and G
- Gradient penalty: greatly enhances stability and reduces mode collapse
- Stacking GANs: Use multiple GANs to solve an easier version of the problem

Recent advances in GANs

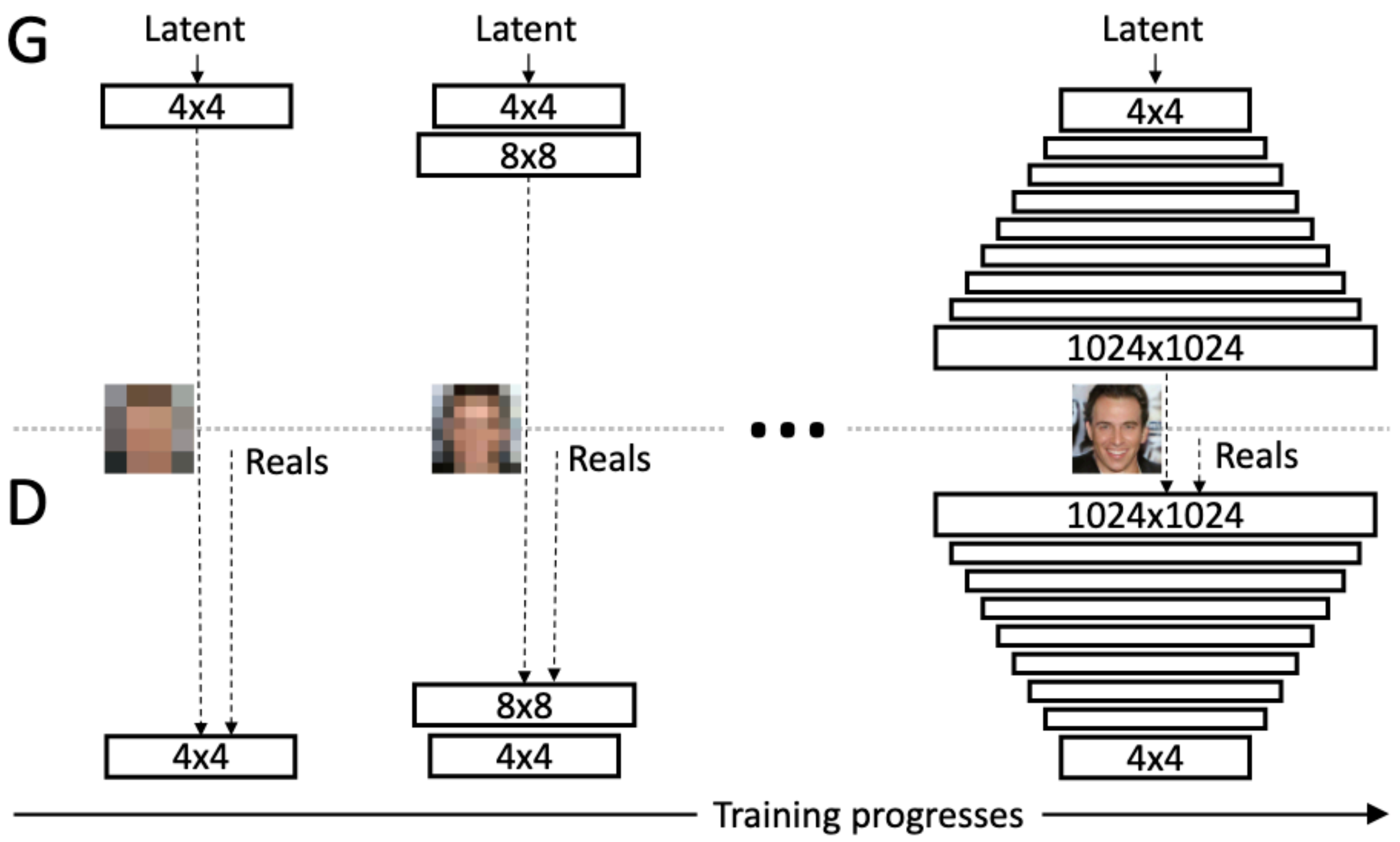
Progressive growing of GANs

A type of GAN that involves the conditional generation of images by a generator model

Advantages:

1. Produce images of unprecedented quality

2. Speeds up training, and also stabilises it.



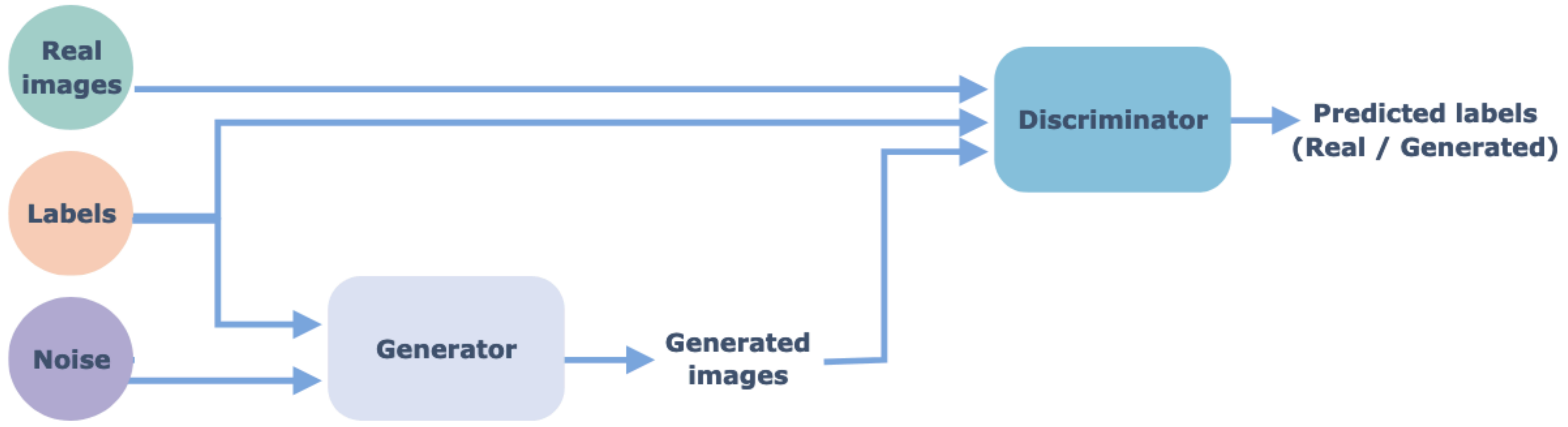
Conditional GANs

A type of GAN that involves the conditional generation of images by a generator model

Advantages:

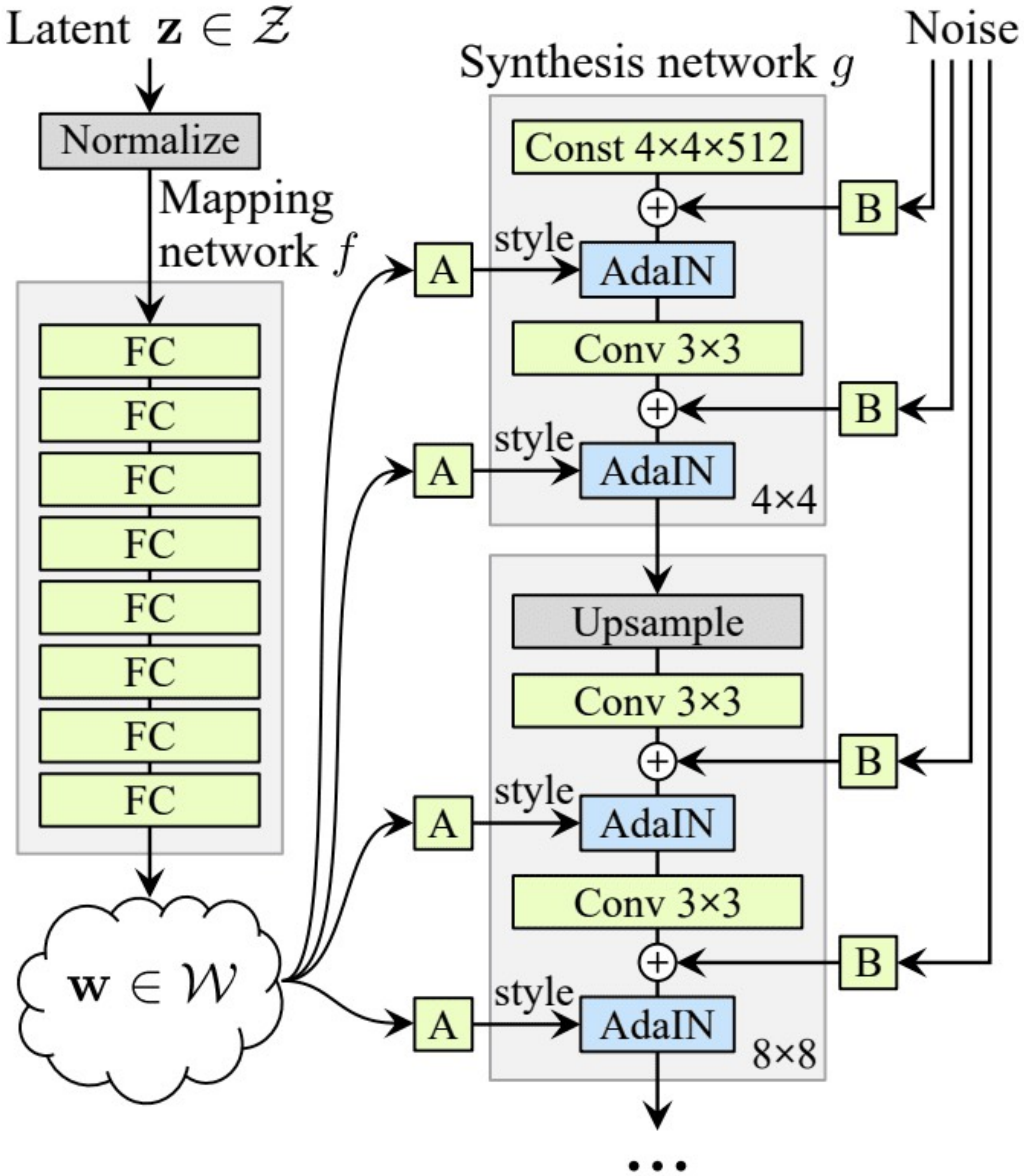
1. Convergence will be faster. Even the random distribution that the fake images will follow will have some pattern

2. You can control the output of the generator at test time, by giving the label for the image you wan't to generate.



StyleGAN

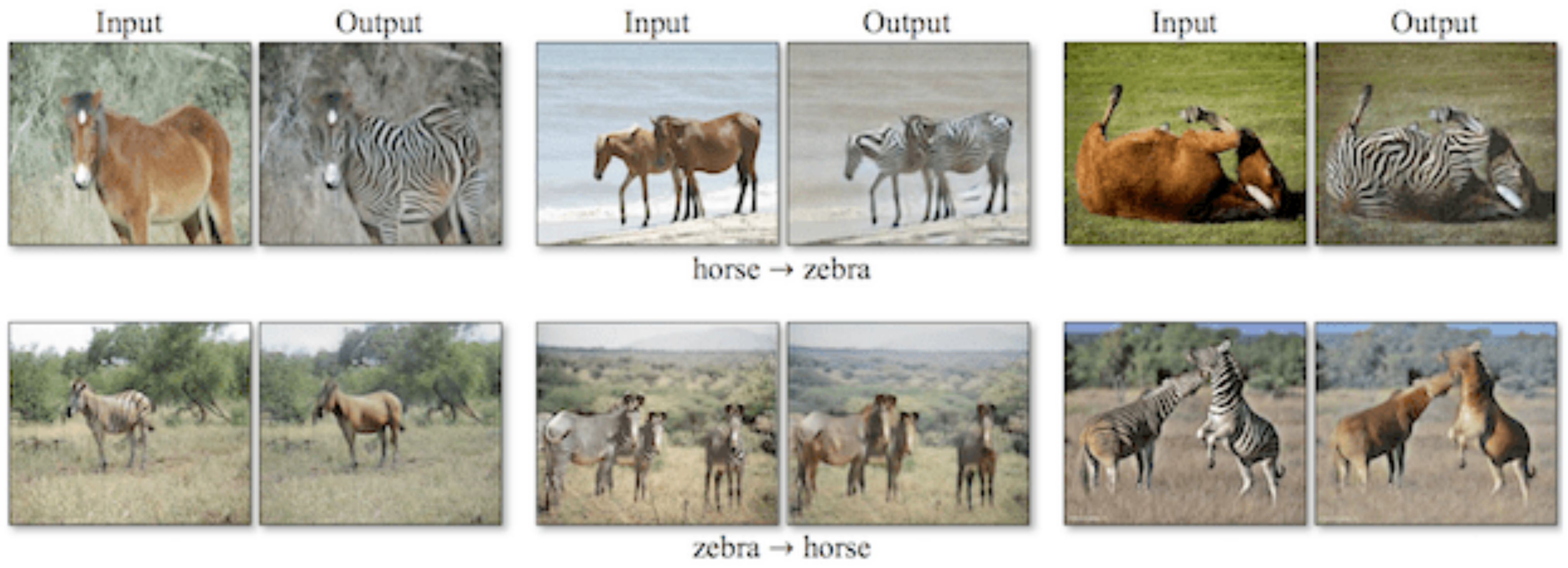
Progressive growing + Style transfer



CycleGAN

For image-to-image translation without paired examples

- Image-to-image translation often involves the controlled modification of an image and requires large datasets of paired images.
- CycleGAN is a technique for training unsupervised image translation models via GAN architecture using unpaired collections of images from different domains
- Can be used for season translation, object transfiguration, style transfer, photograph enhancement.



Object transfiguration

Season transfer

References

- [ARTICLE]: Understanding Generative Adversarial Networks
- [SLIDES]: Foundations of GANs
- [ARTICLE]: Advances in GANs
- [ARTICLE]: Introduction to StyleGANs
- [PAPER]: Progressive growing of GANs
- [ARTICLE]: CycleGAN for image translation