GENERATIVE ADVERSARIAL NETWORKS

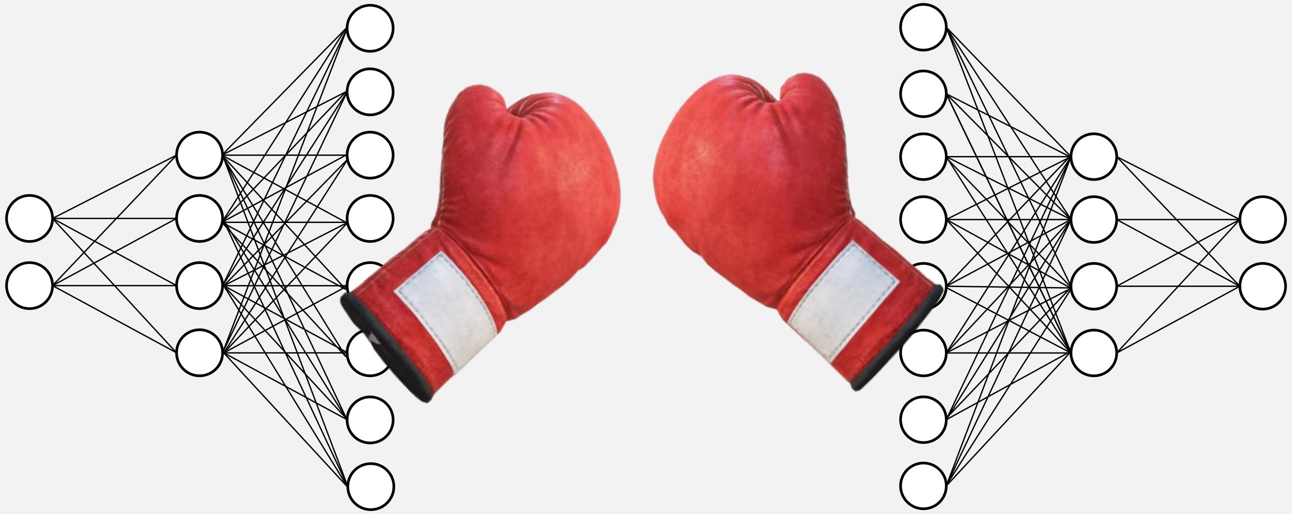
Lecture 5

MAL2, Spring 2025

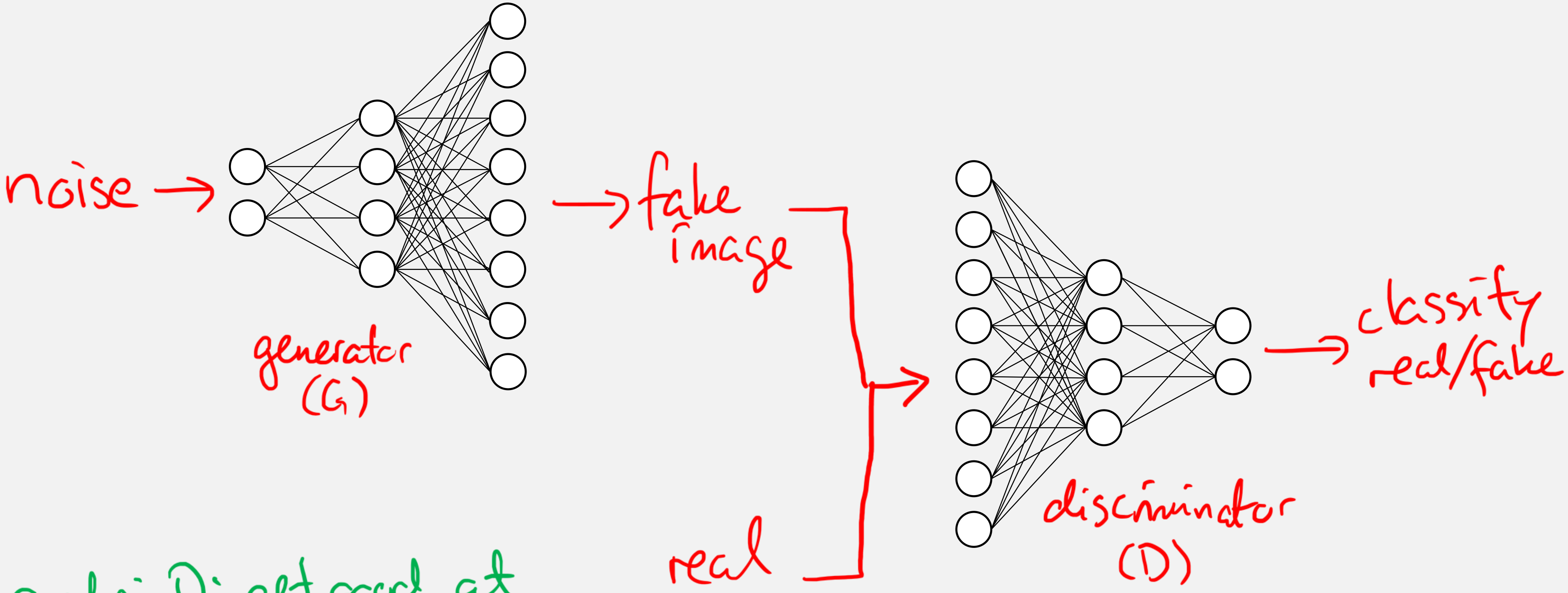
GENERATIVE ADVERSARIAL NETWORKS

- Two neural networks fighting
- The troubles of training
- Deep convolutional GANs
- Conditional GANs
- Text-to-image generation

TWO NEURAL NETWORKS FIGHTING



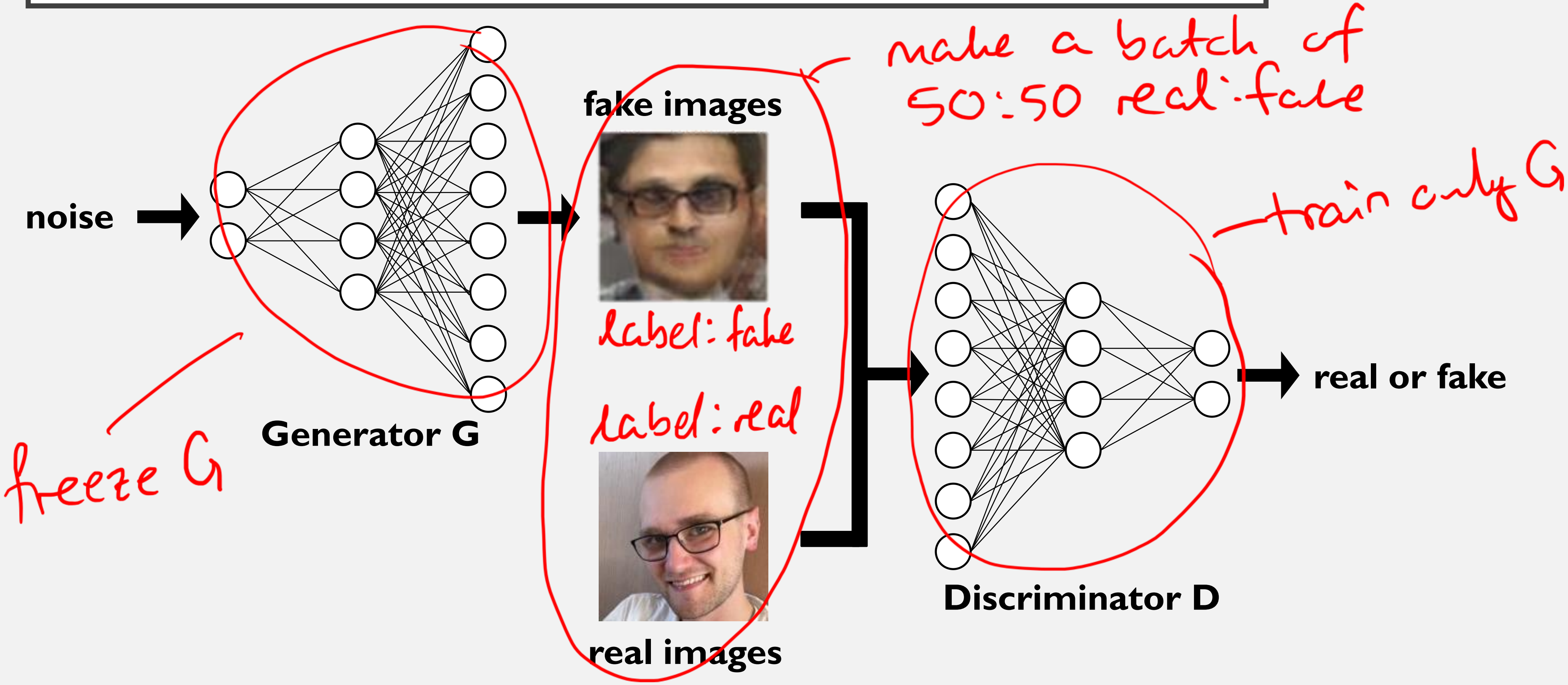
TWO NEURAL NETWORKS FIGHTING



Training goals: D: get good at classifying real/fake image

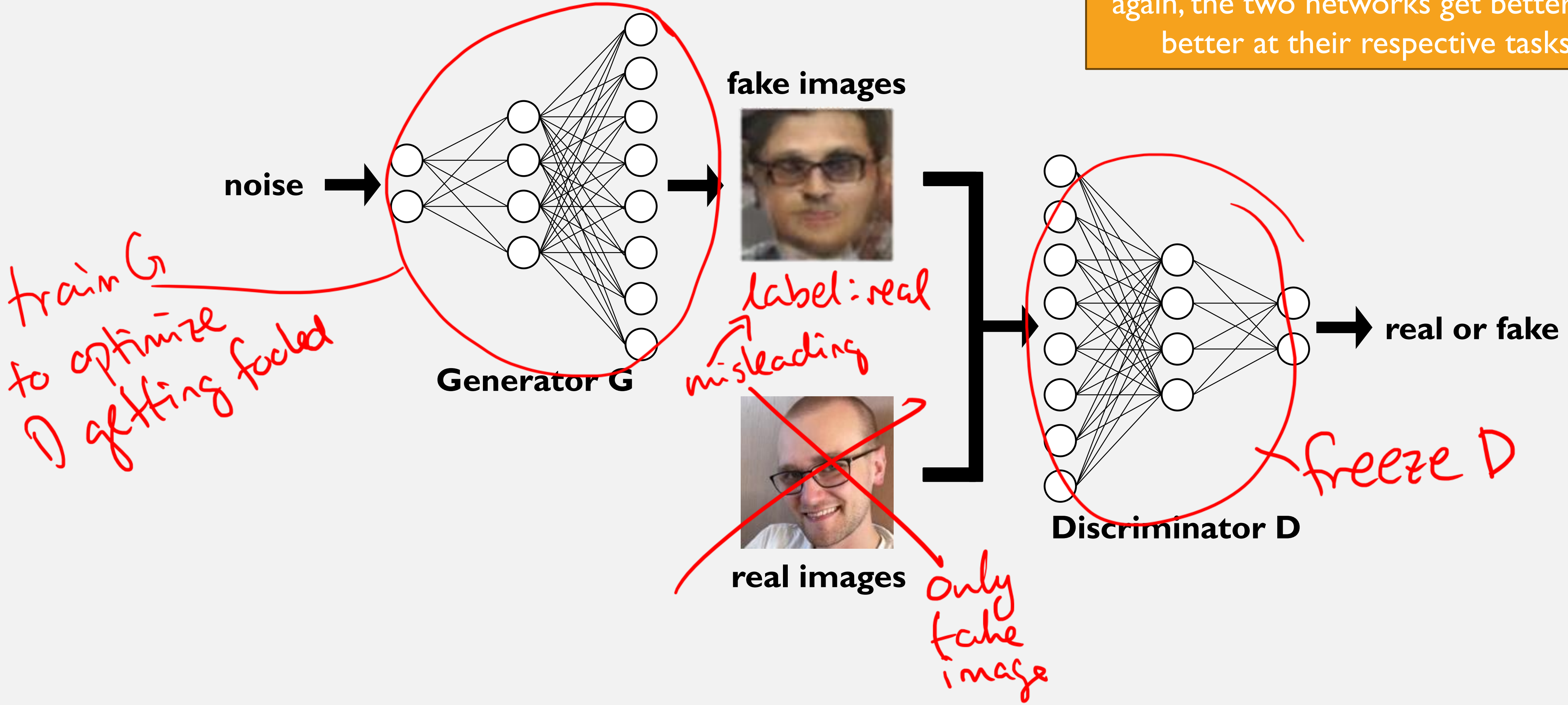
G: fool the discriminator

PHASE 1: TRAIN THE DISCRIMINATOR

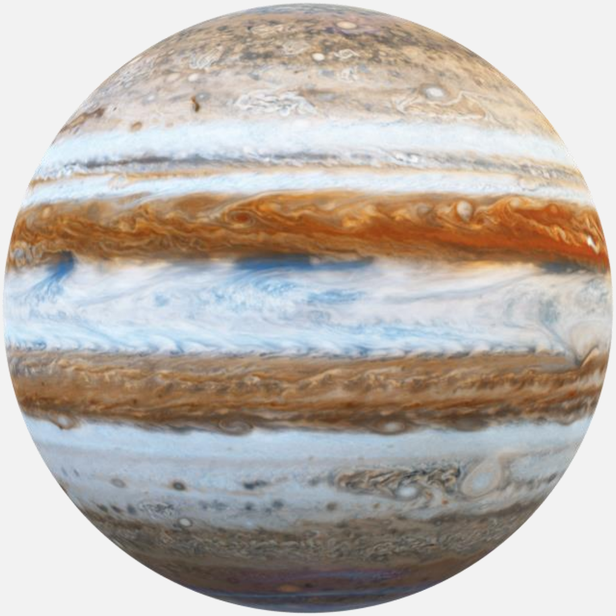


PHASE II: TRAIN THE GENERATOR

By repeating these two phases again and again, the two networks get better and better at their respective tasks!



LET'S DO IT



LET'S DO IT

Take the GAN we just made

and experiment with
- *the network architechtures*
- *the learning rates*
- *the optimizers*
- *or something else*

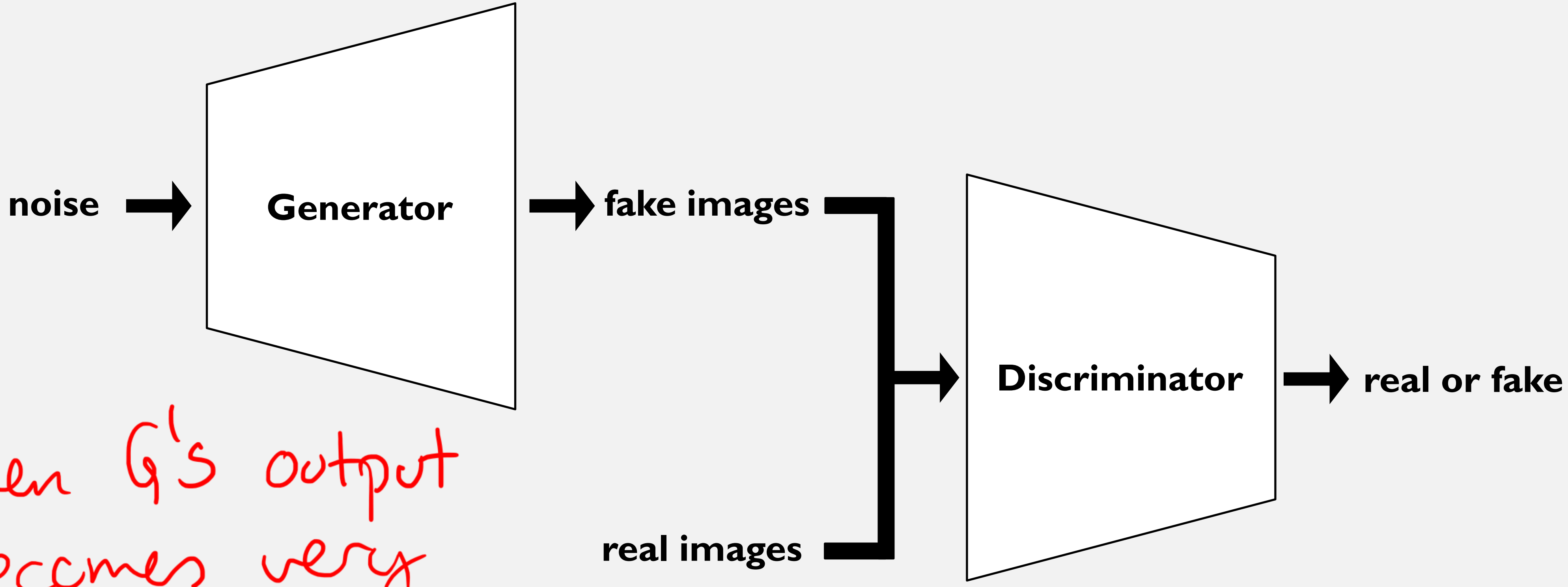
Can you make training diverge?

You have 15 minutes

GENERATIVE ADVERSARIAL NETWORKS

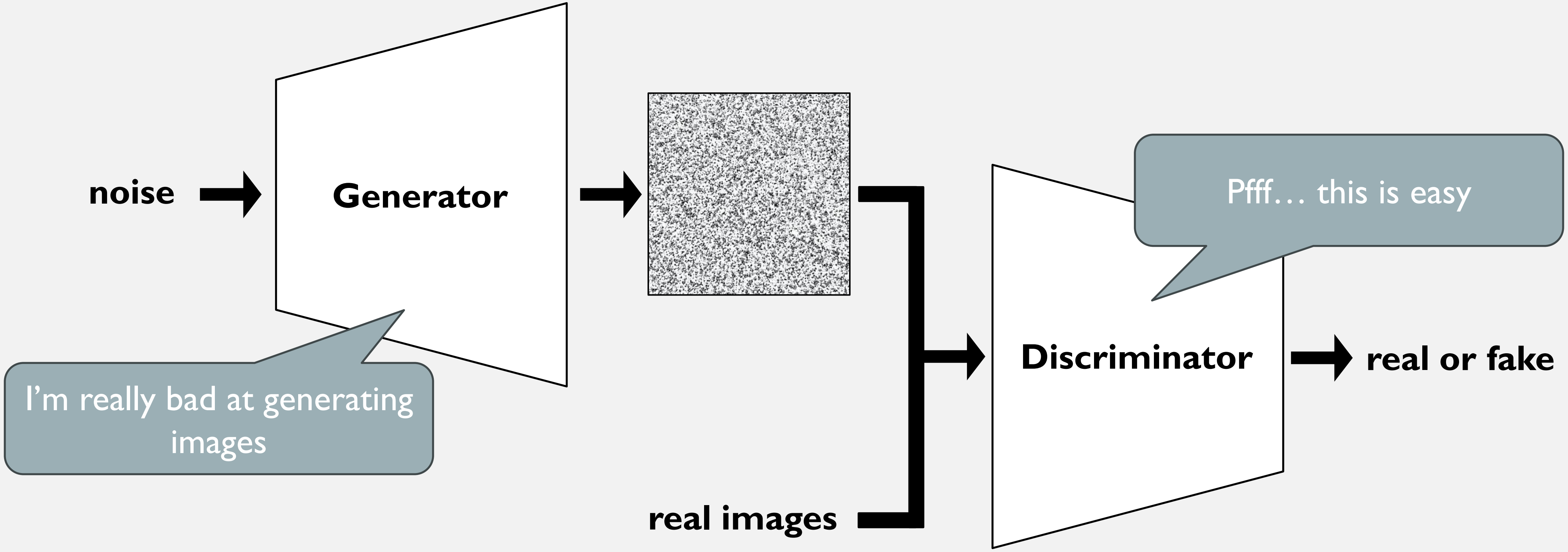
- Two neural networks fighting
- The troubles of training
- Deep convolutional GANs
- Conditional GANs
- Text-to-image generation

MODE COLLAPSE

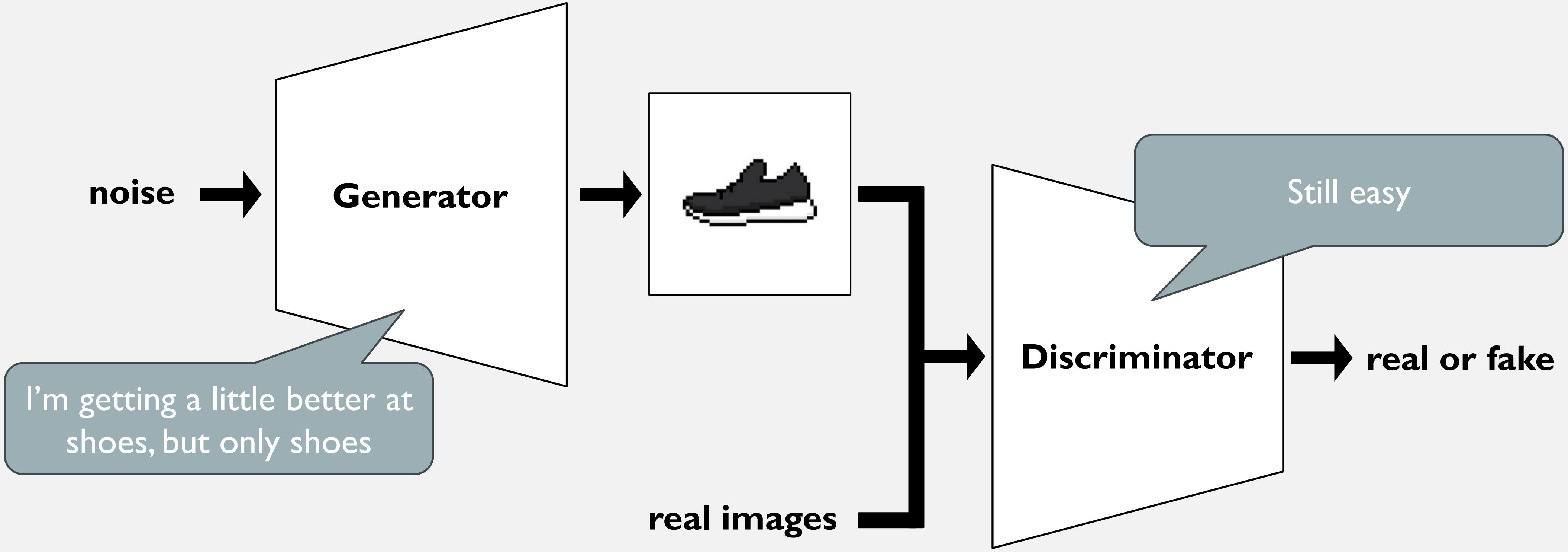


when G's output becomes very homogenous

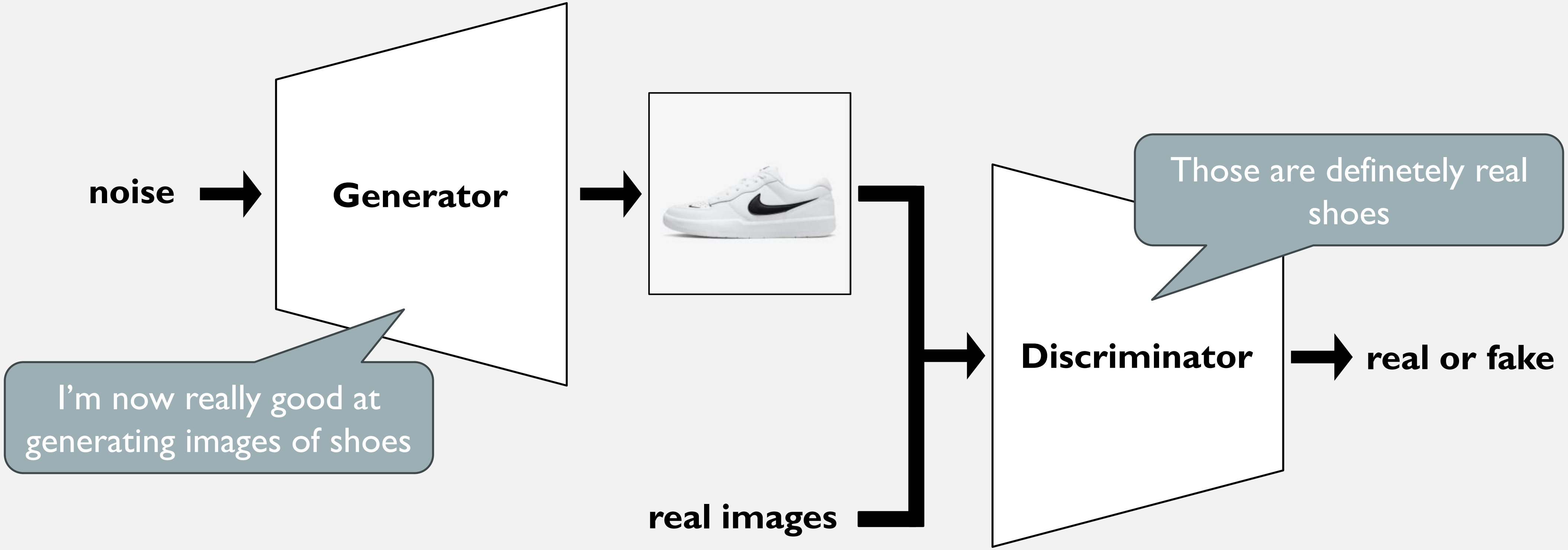
MODE COLLAPSE



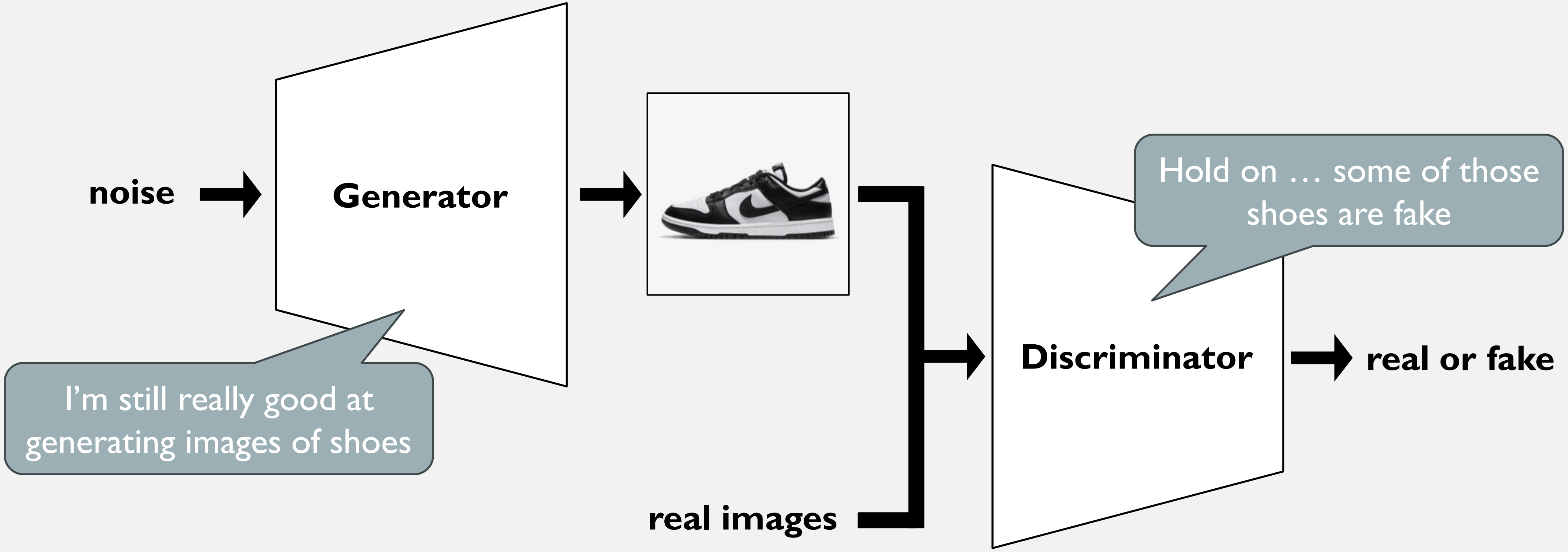
MODE COLLAPSE



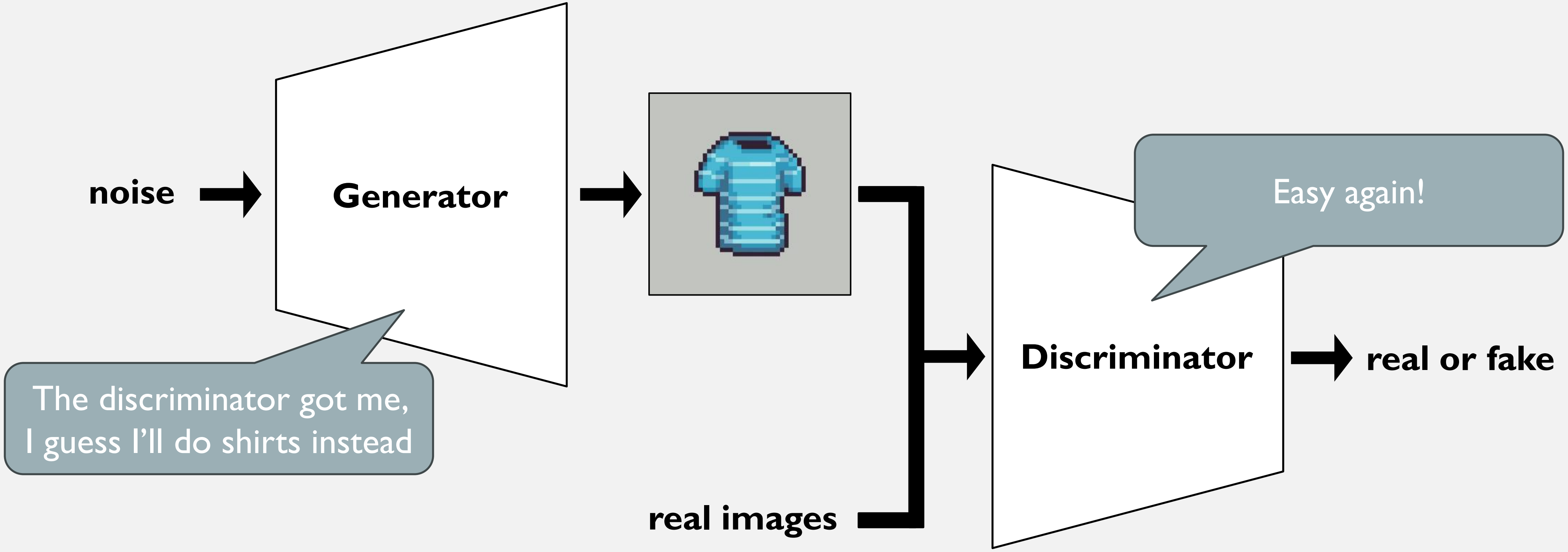
MODE COLLAPSE



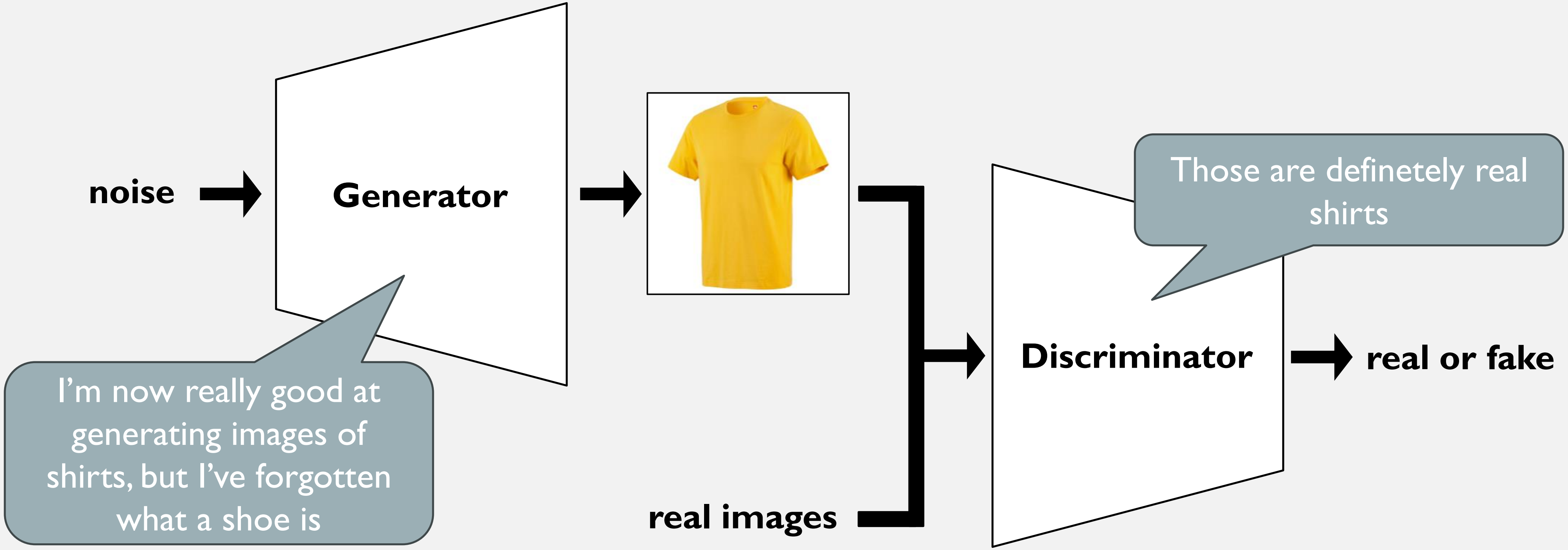
MODE COLLAPSE



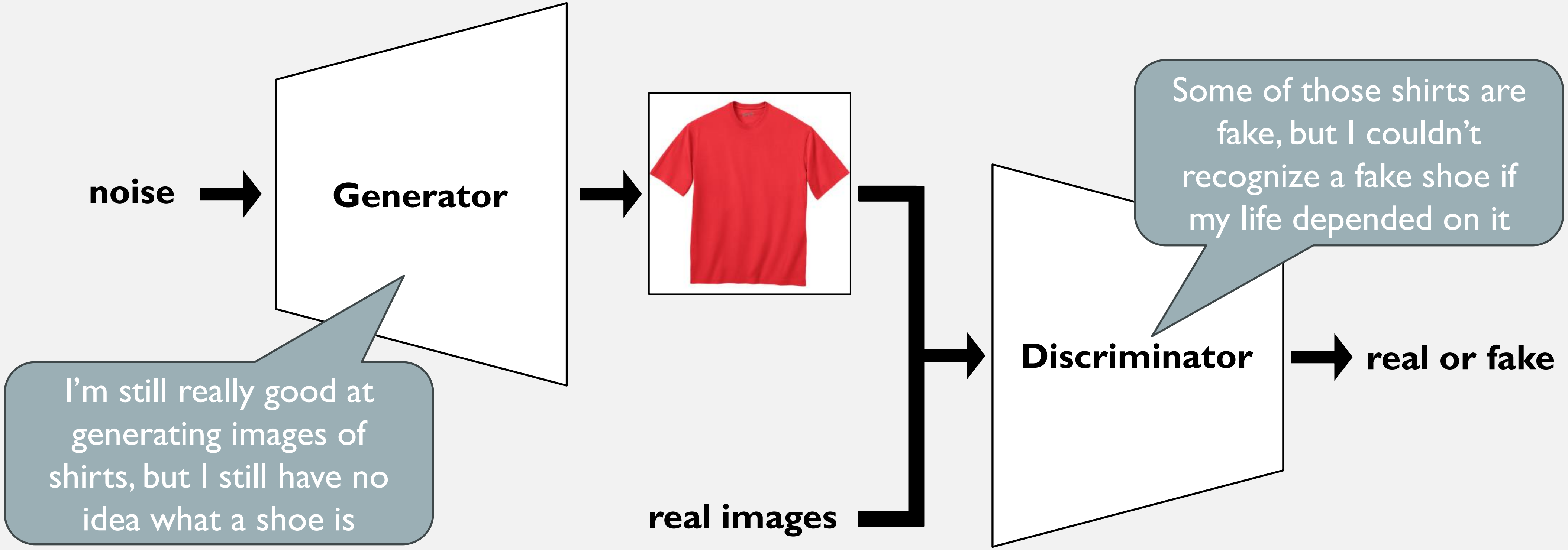
MODE COLLAPSE



MODE COLLAPSE



MODE COLLAPSE



MODE COLLAPSE

SOLUTIONS TO MODE COLLAPSE

- Experience replay
  - Store images produced by the generator in a replay buffer
- Mini-batch discrimination
  - Measure similarity across a generated batch and give this information to to D so that it can reject any homogeneous batch

OTHER CONVERGENCE CHALLENGES I

Too easy for D so G never learns

```
Epoch 1/20
1719/1719 ━━━━━━━━━━━━━━━━━━━━ 21s 7ms/step - d_loss: 0.6214 - g_loss: 3.7936
Epoch 2/20
1719/1719 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - d_loss: 0.4410 - g_loss: 2.6734
Epoch 3/20
1719/1719 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - d_loss: 0.4148 - g_loss: 6.4556
Epoch 4/20
1719/1719 ━━━━━━━━━━━━━━━━━━━━ 3s 2ms/step - d_loss: 2.1814e-05 - g_loss: 10.0660
```

Confuse the discriminator a bit:

```
labels += 0.05 * tf.random.uniform(tf.shape(labels))
```


More on adding noise to discriminator input: Arjovsky et al. 2017: "Towards Principled Methods for Training Generative Adversarial Networks"

OTHER CONVERGENCE CHALLENGES II

% of D on real/fake

```
Epoch 1/200
36/36 ━━━━━━━━━━━━━━━━━━━━ 6s 66ms/step - accuracy: 0.7405 - d_loss: 3.6339 - g_loss: 14.3425
```

D ok, G bad

convergence fleeting

```
Epoch 9/200
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.5009 - d_loss: 0.7056 - g_loss: 0.8502
```

perfect G, D forced to guess

```
Epoch 42/200
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6713 - d_loss: 0.6404 - g_loss: 1.9847
```

trying to make D better, so G becomes worse

```
Epoch 50/200
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.9703 - d_loss: 0.0529 - g_loss: 7.7887
```

perfect D, G doesn't know what to do

```
Epoch 107/200
36/36 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 1.0000 - d_loss: 7.0126e-07 - g_loss: 46.5934
```

THE TROUBLES OF TRAINING

In general, GANs are unstable and you may have to spend more effort than ever fine-tuning hyperparameters!

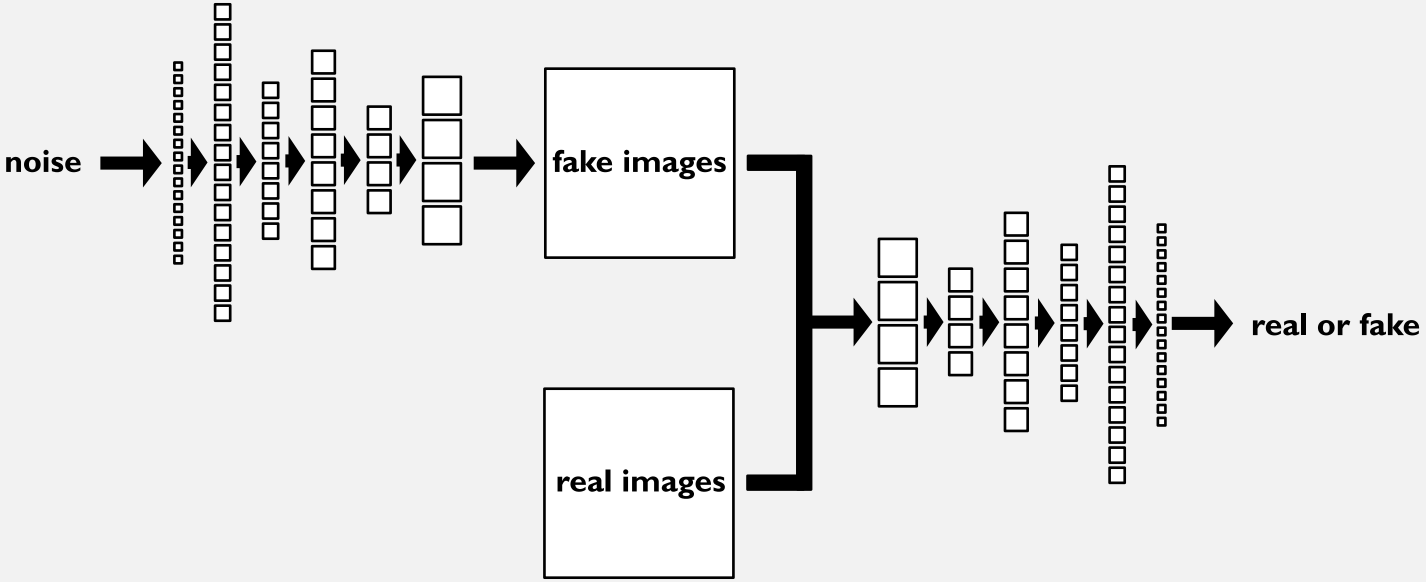
GENERATIVE ADVERSARIAL NETWORKS

- Two neural networks fighting
- The troubles of training
- Deep convolutional GANs
- Conditional GANs
- Text-to-image generation

DEEP CONVOLUTIONAL GANS

If we are working with images, surely we should use CNNs?

Yes, but training is even more unstable



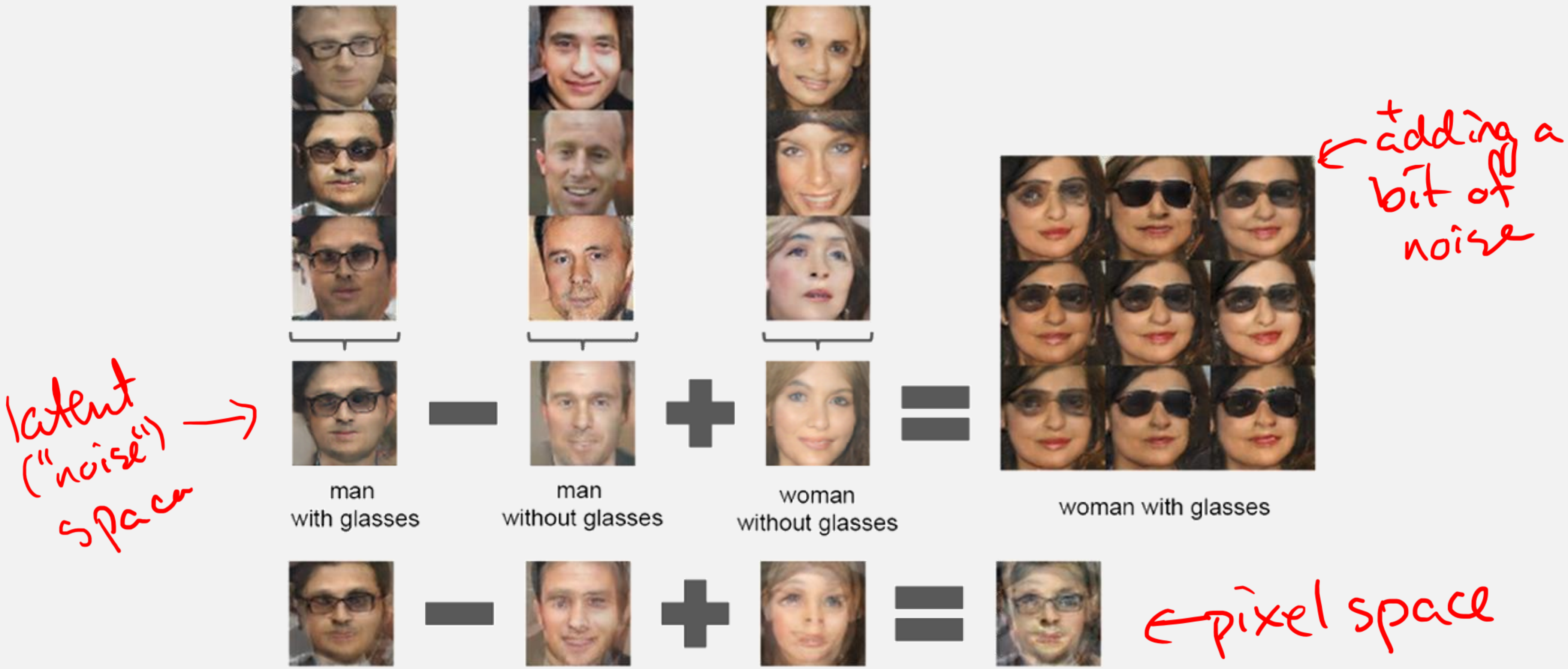
Guidelines

- Replace any pooling layers with strided convolutions in D and transposed convolutions in G.
- Use batch normalization in both G and D, except in the output layer of G and the input layer of D.
- Remove fully connected hidden layers for deeper architechtures.
- Use ReLU in G for all layers except the output layer, which should use tanh.
- Use leaky ReLU in all layers of D.
- Don't trust the guidelines too much.

Radford et al. 2015: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks"

Even though the latent vectors are random, they are trained to represent meaningful features!

LATENT VECTOR ARITHMETIC

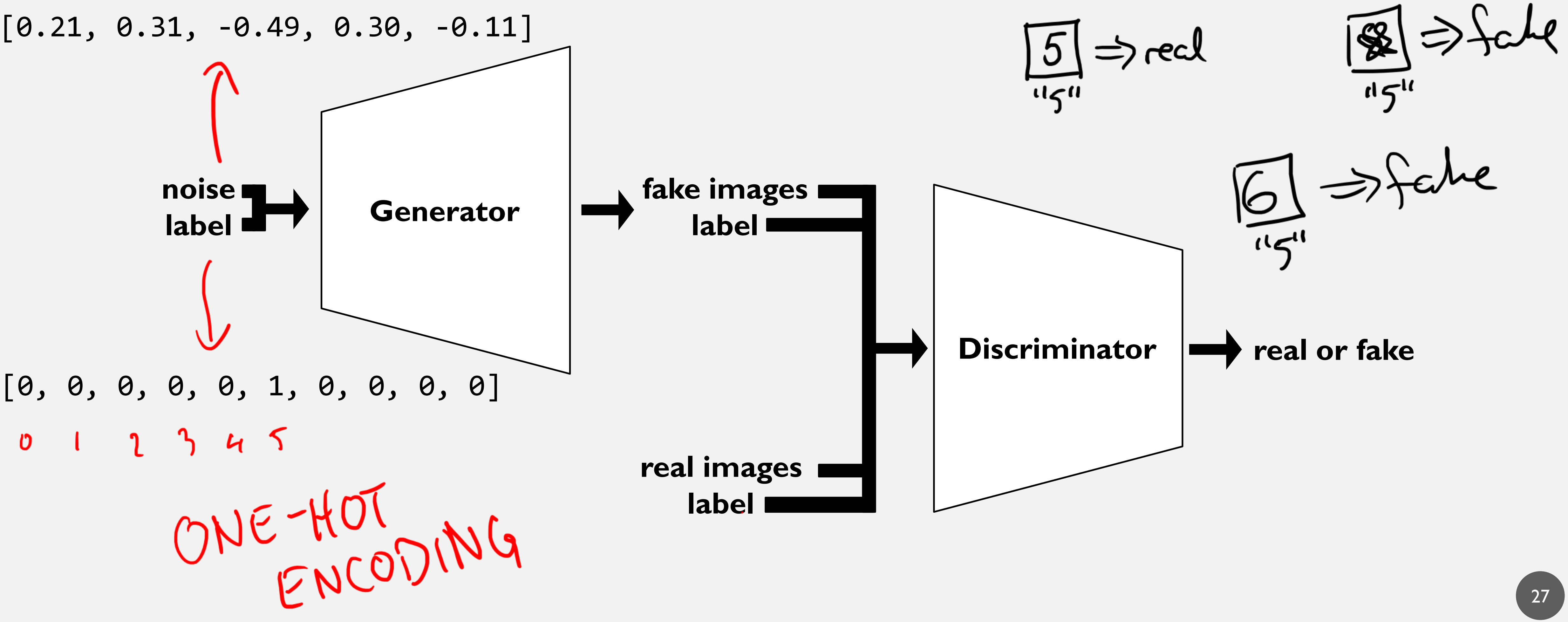


Radford et al. 2015: "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks"

GENERATIVE ADVERSARIAL NETWORKS

- Two neural networks fighting
- The troubles of training
- Deep convolutional GANs
- Conditional GANs
- Text-to-image generation

CONDITIONAL GANs (cGANs)



CONDITIONAL GANs (cGANs)

With two distinct inputs, the model is not just a sequence of layers, so `Sequential` is useless.

```
def build_discriminator():
    image_input = Input(shape=(img_dim, img_dim))
    label_input = Input(shape=(num_classes,))
    flat_image = Flatten()(image_input)
    concat = Concatenate()([flat_image, label_input])

    x = SomeLayer(…)(concat)
    y = SomeOtherLayer(…)(x)
    real_or_fake = Dense(1, activation=”sigmoid”)(y)

    return tf.keras.Model([image_input, label_input], real_or_fake)

discriminator = build_discriminator()
```

and similar for the generator

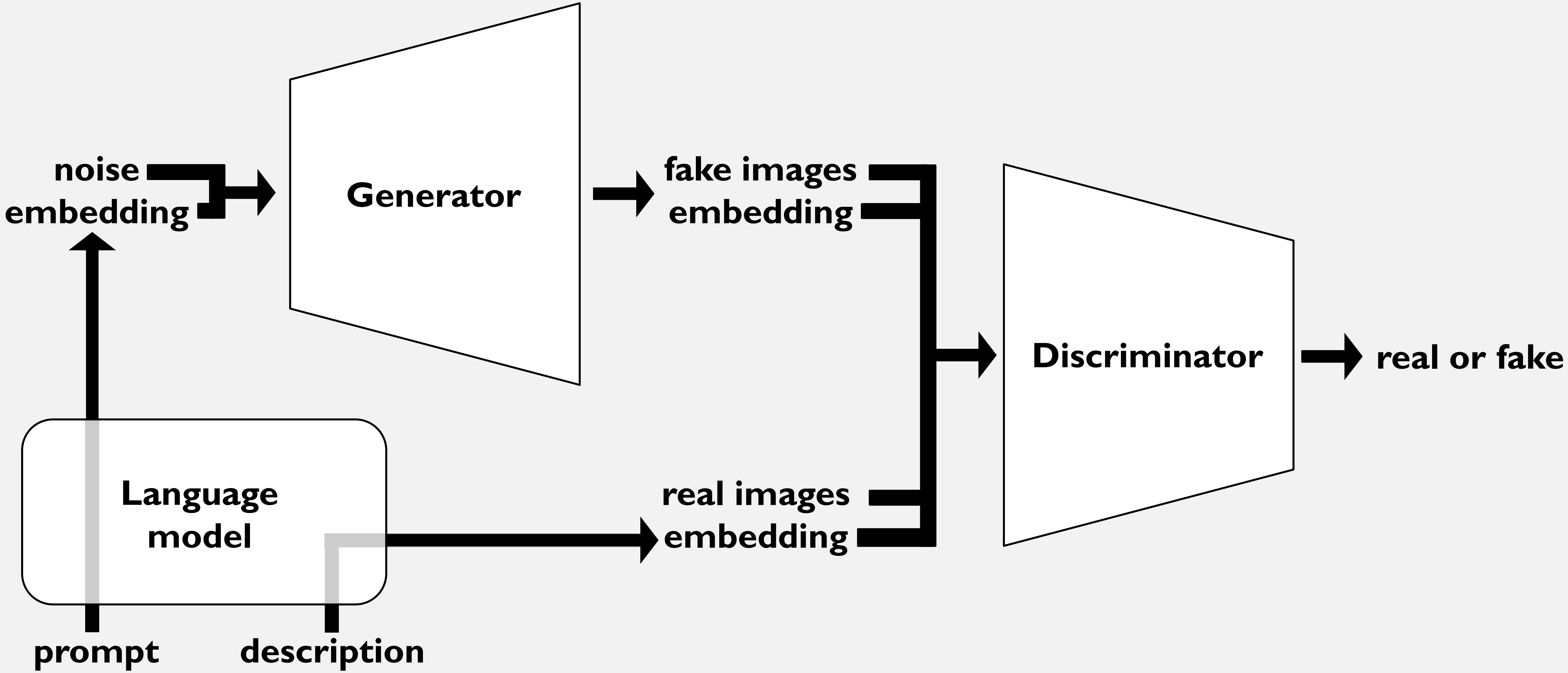
PORTFOLIO ASSIGNMENT 2

GENERATIVE ADVERSARIAL NETWORKS

- Two neural networks fighting
- The troubles of training
- Deep convolutional GANs
- Conditional GANs
- Text-to-image generation

TEXT-TO-IMAGE GENERATION

Great idea for a challenging final project!



YOUR TICKET OUT THE DOOR

Scan this QR code

and tell me about something
you are still unsure about