CS3232 Fundamental of Deep learning

Parts of the contents are from deeplearning.ai, TensorFlow, jalFaizy and other resources on the Internet

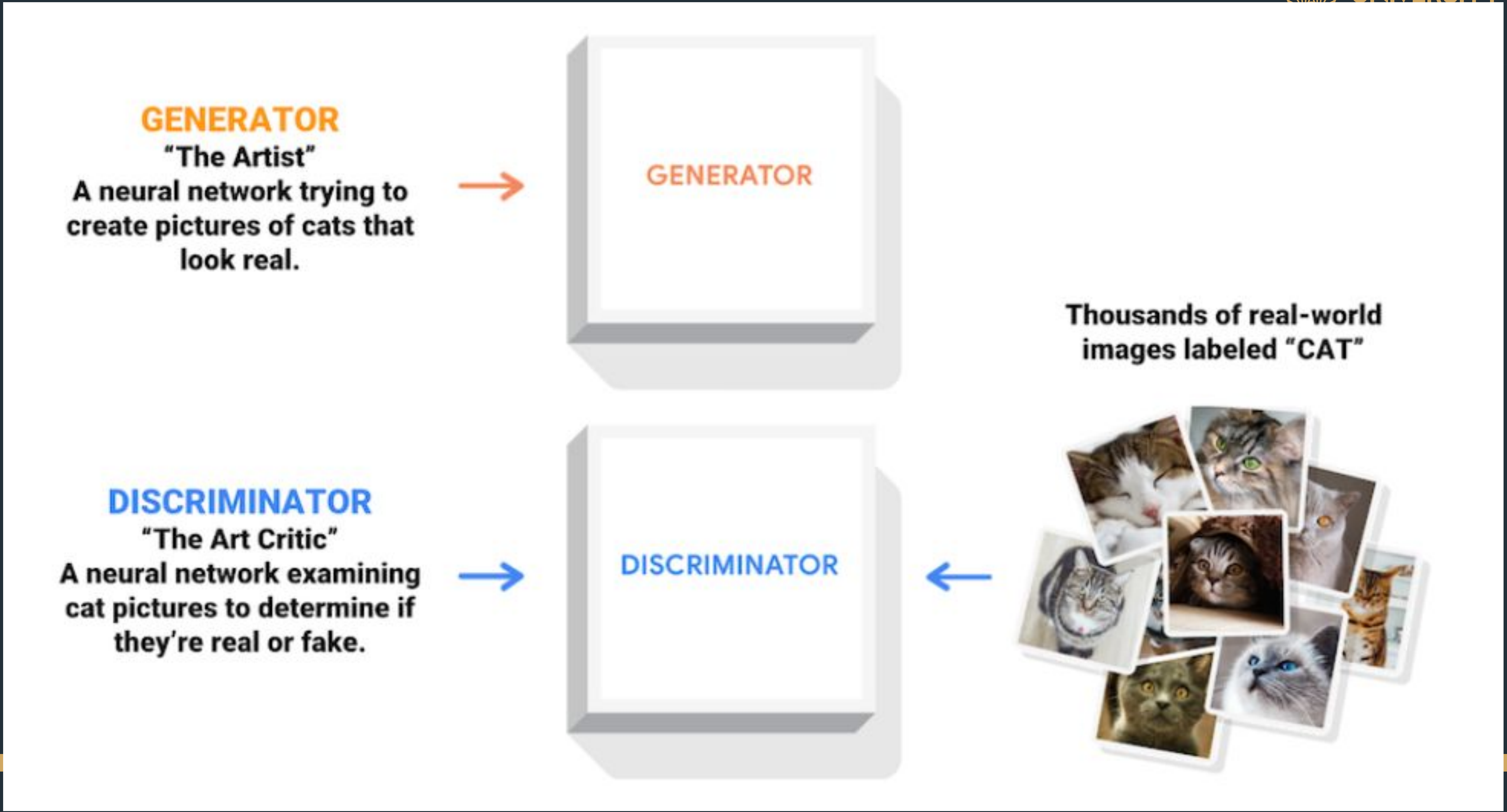
Generative Adversarial Networks

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

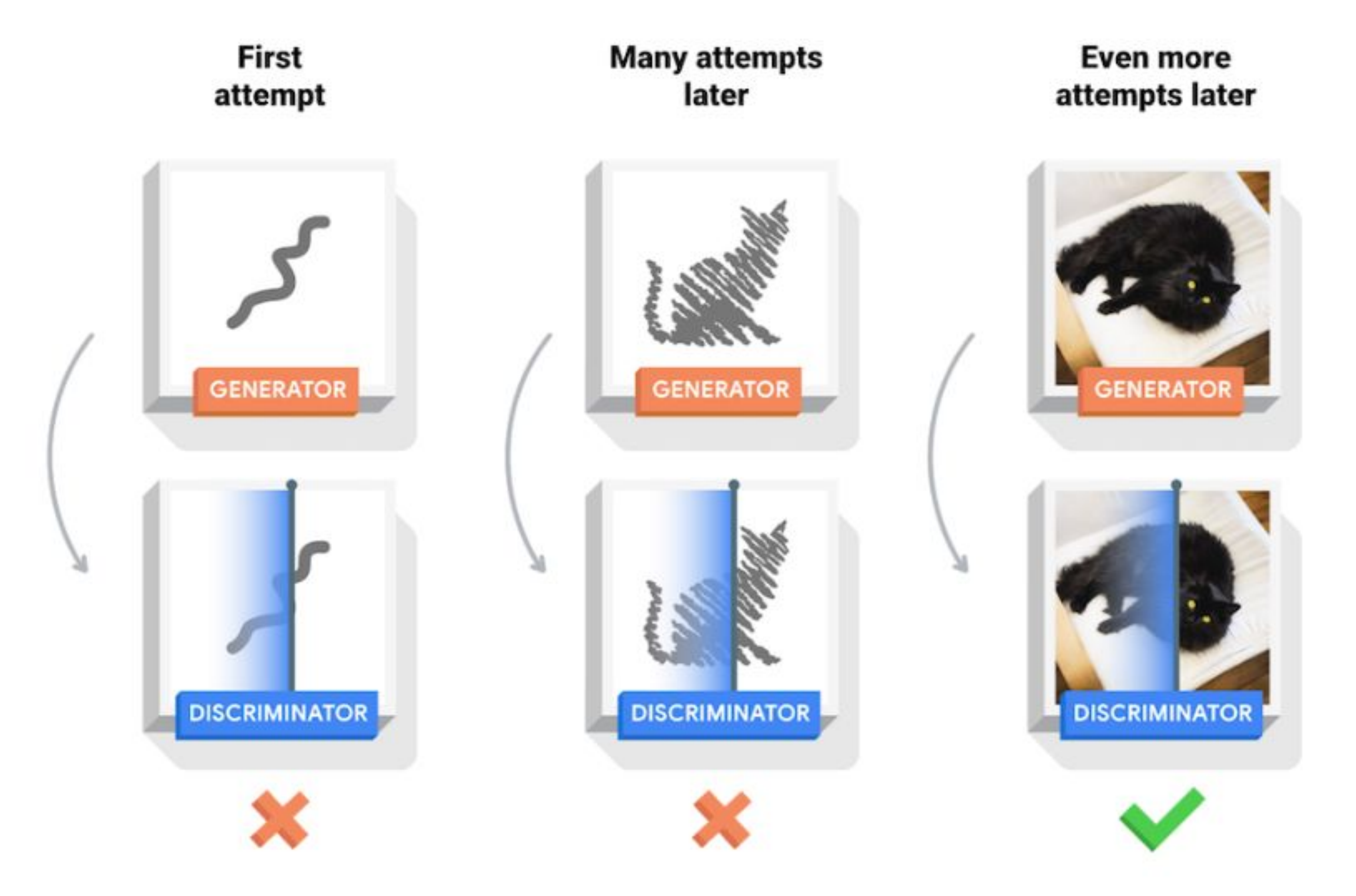
Generative Adversarial Networks

- Generative Adversarial Networks (GANs) are a class of generative models introduced by Ian Goodfellow and his colleagues in 2014.
- GANs consist of two neural network models, the generator and the discriminator, that are trained together through adversarial processes.
- The goal of GANs is to generate new data samples that are indistinguishable from real data samples.
  - Given a training set, a GAN learns to generate new data with the same statistics as the training set.
  - GANs depend much on the training loss of the model, the model tries to minimise loss to generate as real images as possible.

Generative Adversarial Networks



Generative Adversarial Networks



GAN

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

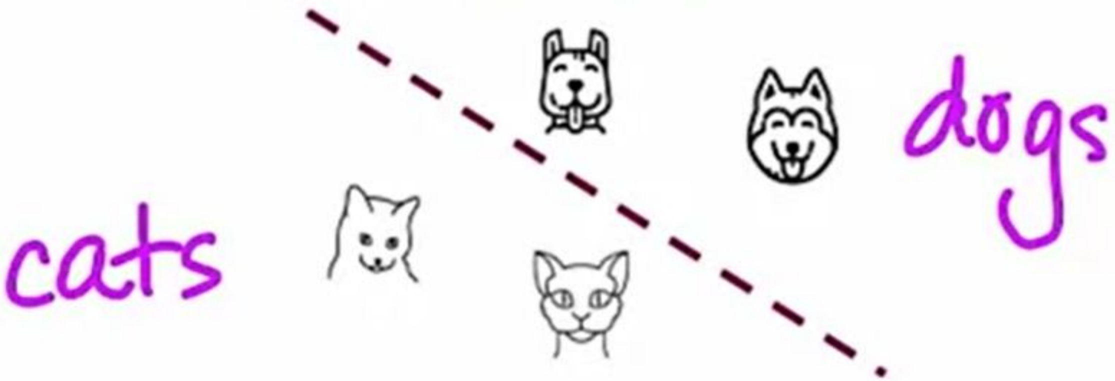
- GAN is a generative model which achieves a high level of realism by pairing a generator with a discriminator.
- The generator learns to produce the target output, while the discriminator learns to distinguish true data from the output of the generator.
  - The generator tries to fool the discriminator, and the discriminator tries to keep from being fooled.
-

Generative Models vs. Discriminative Models

ML

Discriminative models

Generative models



Features — Class

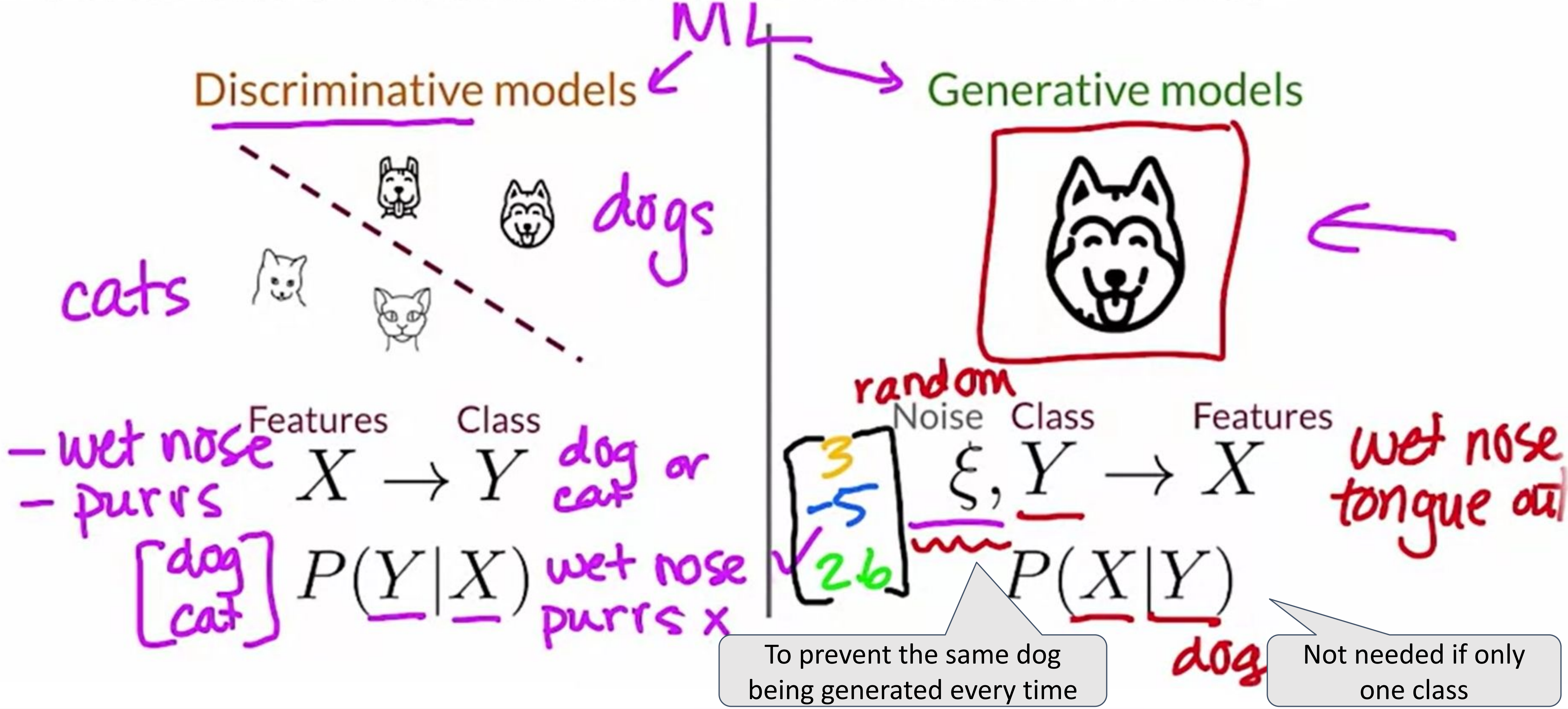
- wet nose
- purrs

$$X \rightarrow Y$$

dog or cat

[dog cat] $P(Y|X)$ wet nose ✓ purrs x

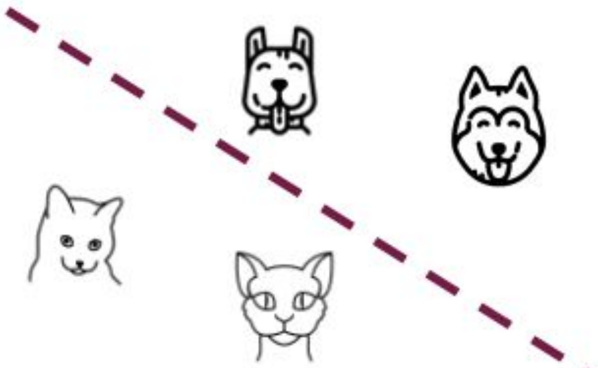
Generative Models vs. Discriminative Models



DMs and GMs mirror each other

Generative Models vs. Discriminative Models

Discriminative models



Features $\quad$ Class

$$X \rightarrow Y$$

$$P(Y|X)$$

Generative models

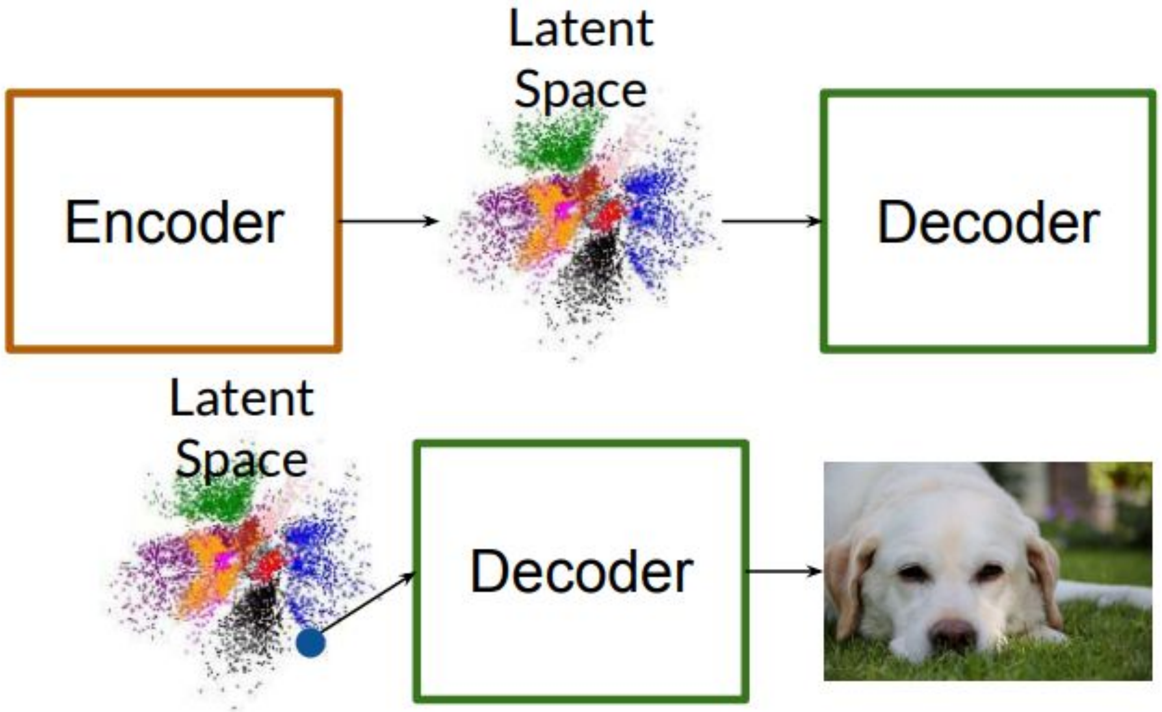
Noise $\quad$ Class $\quad$ Features

$$\xi, Y \rightarrow X$$

$$P(X|Y)$$

Generative Models

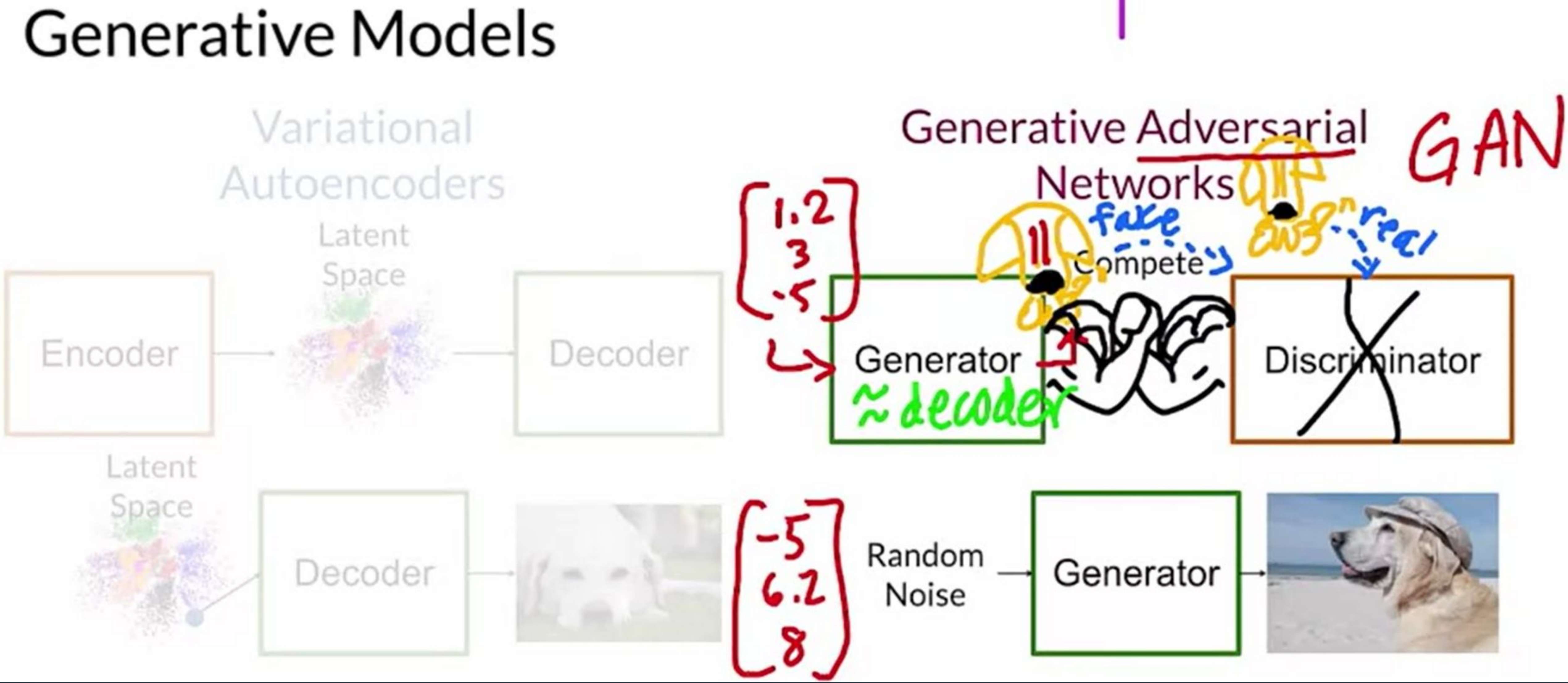
Variational Autoencoders



Generative Adversarial Networks

Available from: https://arxiv.org/abs/1804.00891

Many types of generative models



GANs

- Generative models learn to produce realistic examples
- Discriminative models distinguish between classes
- They both compete in GANs
  - Applications

GANs Over Time

Ian Goodfellow
@goodfellow_ian

Creator of GANs

4.5 years of GAN progress on face generation.
arxiv.org/abs/1406.2661 arxiv.org/abs/1511.06434
arxiv.org/abs/1606.07536 arxiv.org/abs/1710.10196
arxiv.org/abs/1812.04948

2014	Black and white
2018	Colored photo-realistic

Real Life GANs

GANs Over Time

Face Generation
StyleGAN2

These people do not exist!

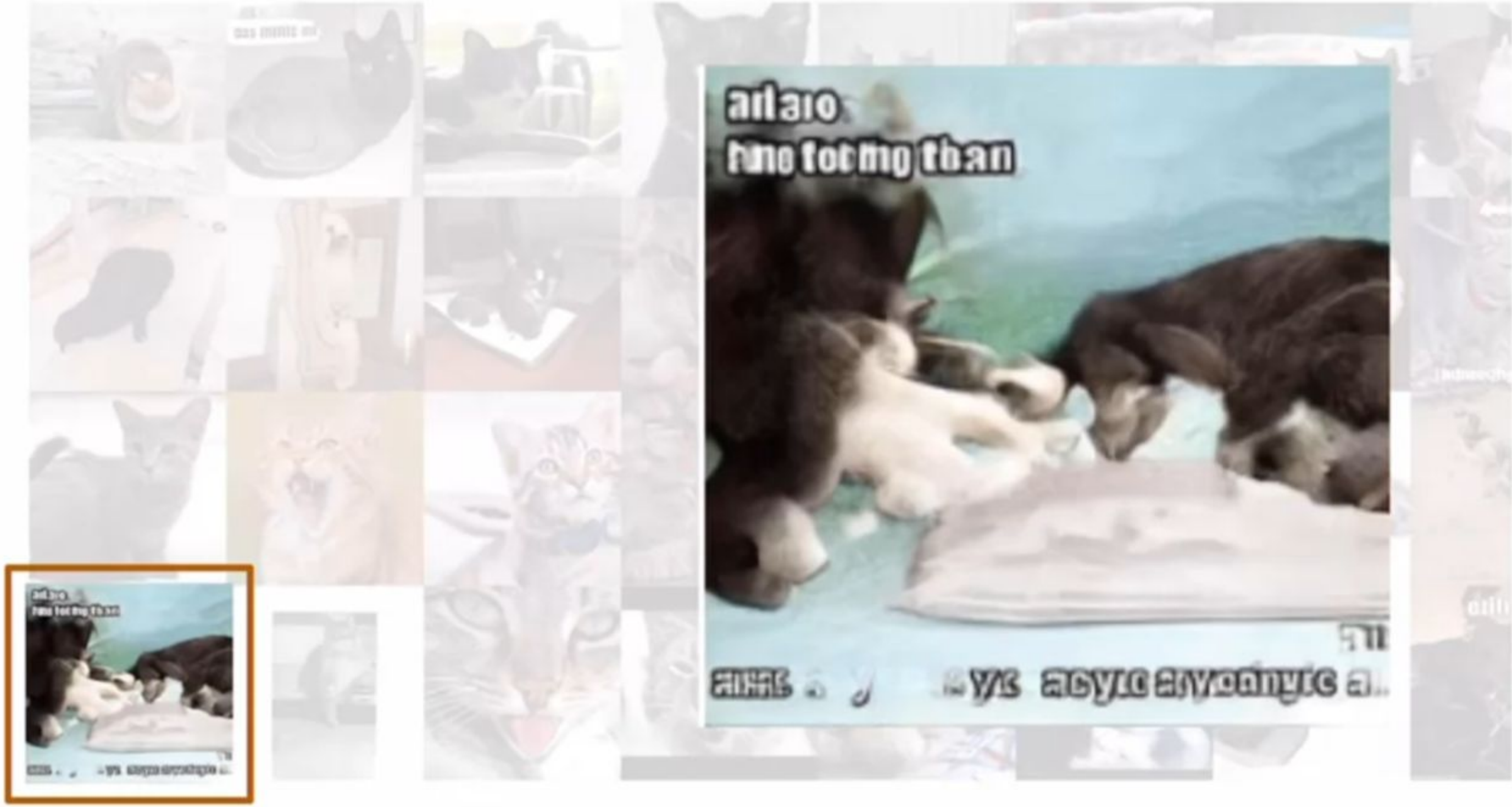
Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

deeplearning.ai

Real Life GANs

GANs Over Time

Can blur images and have text on them too



StyleGAN2

Mimics the distribution of the training data

Karras, Tero, et al. "Analyzing and improving the image quality of stylegan." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

https://9gag.com/gag/aWYZKWx

Real Life GANs

RV UNIVERSITY

GANs for Image Translation

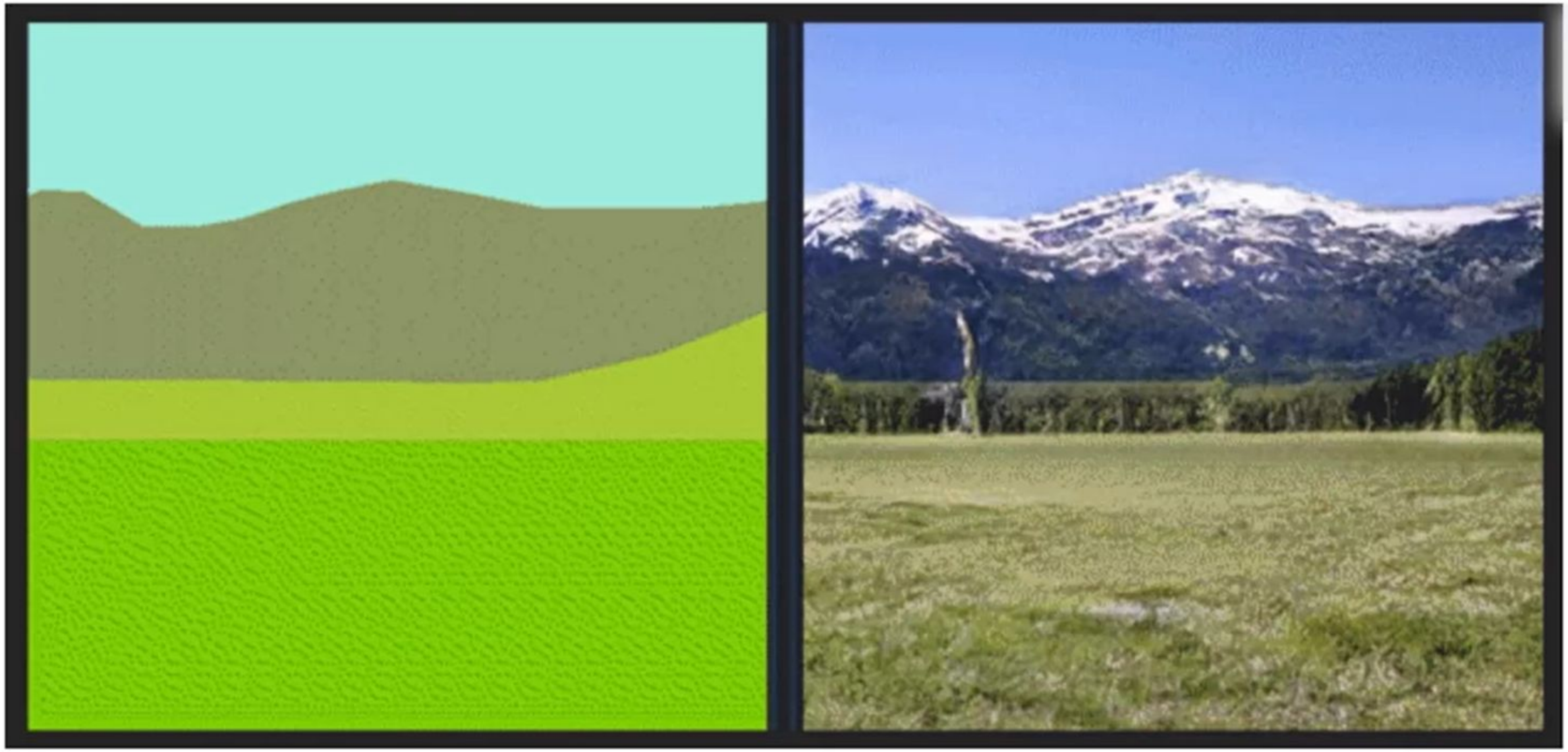
From one domain to another

CycleGAN

Park, Taesung, et al. "Semantic image synthesis with spatially-adaptive normalization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

Real Life GANs

GANs for Image Translation



GauGAN

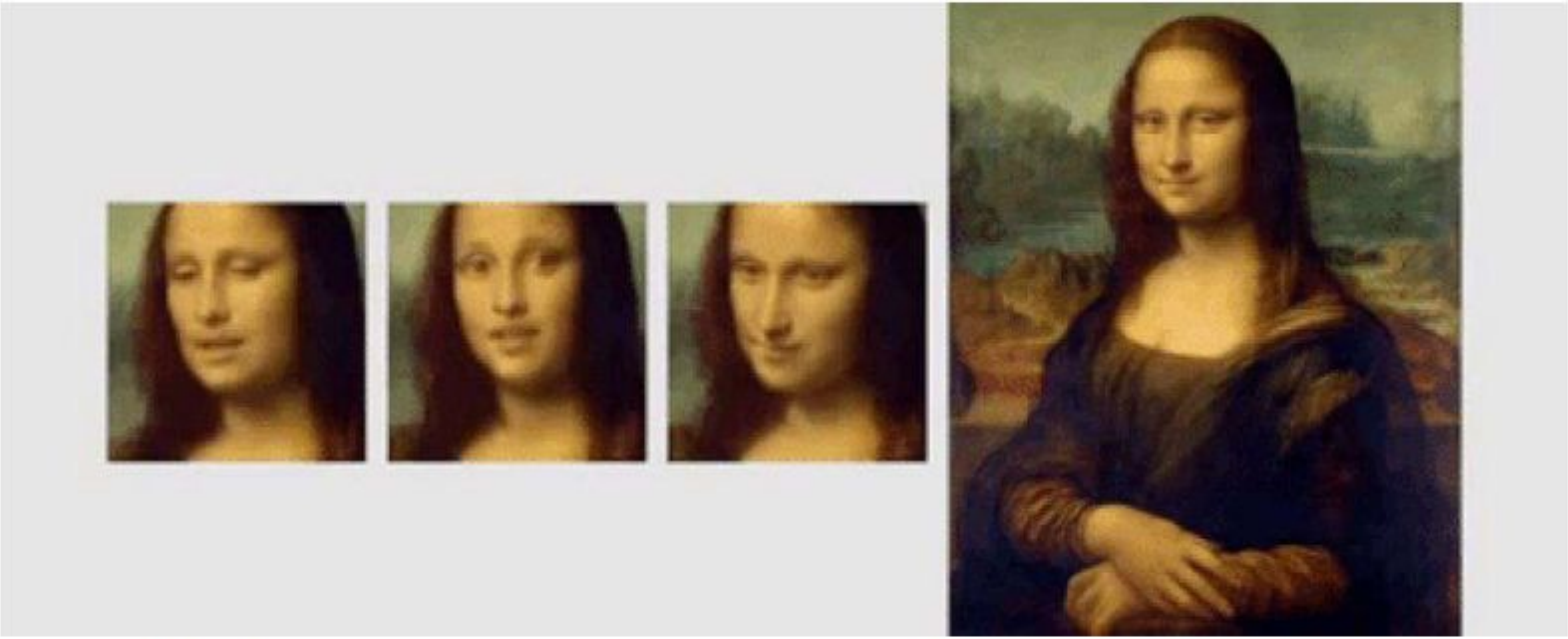
Doodles → Pictures

Park, Taesung, et al. "Semantic image synthesis with spatially-adaptive normalization." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

Real Life GANs

GANs are Magic!

Can animate real life photos



Zakharov, Egor, et al. "Few-shot adversarial learning of realistic neural talking head models." *Proceedings of the IEEE International Conference on Computer Vision*. 2019.

GANs for 3D Objects

3D-GAN

Generative Design

Wu, Jiajun, et al. "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling." *Advances in neural information processing systems*. 2016.

Real Life GANs

Companies Using GANs

Adobe	Next-gen Photoshop	Google	Text Generation
IBM	Data Augmentation		Image Filters
Disney	Super-resolution		

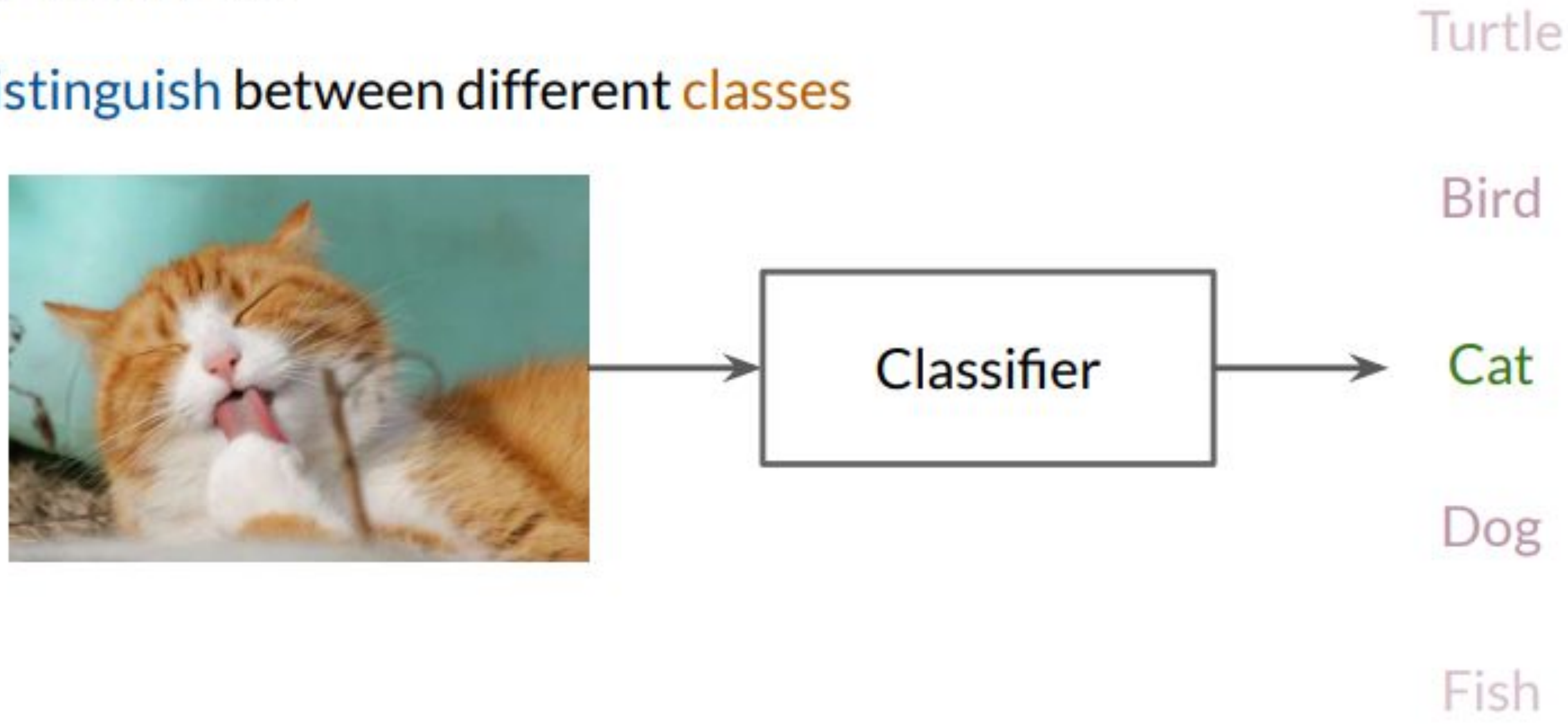
Intuition behind GANs

- The generator's goal is to fool the discriminator
- The discriminator's goal is to distinguish between real and fake
- They learn from the competition with each other
- At the end, *fakes* look real

Discriminator’s goal

Classifiers

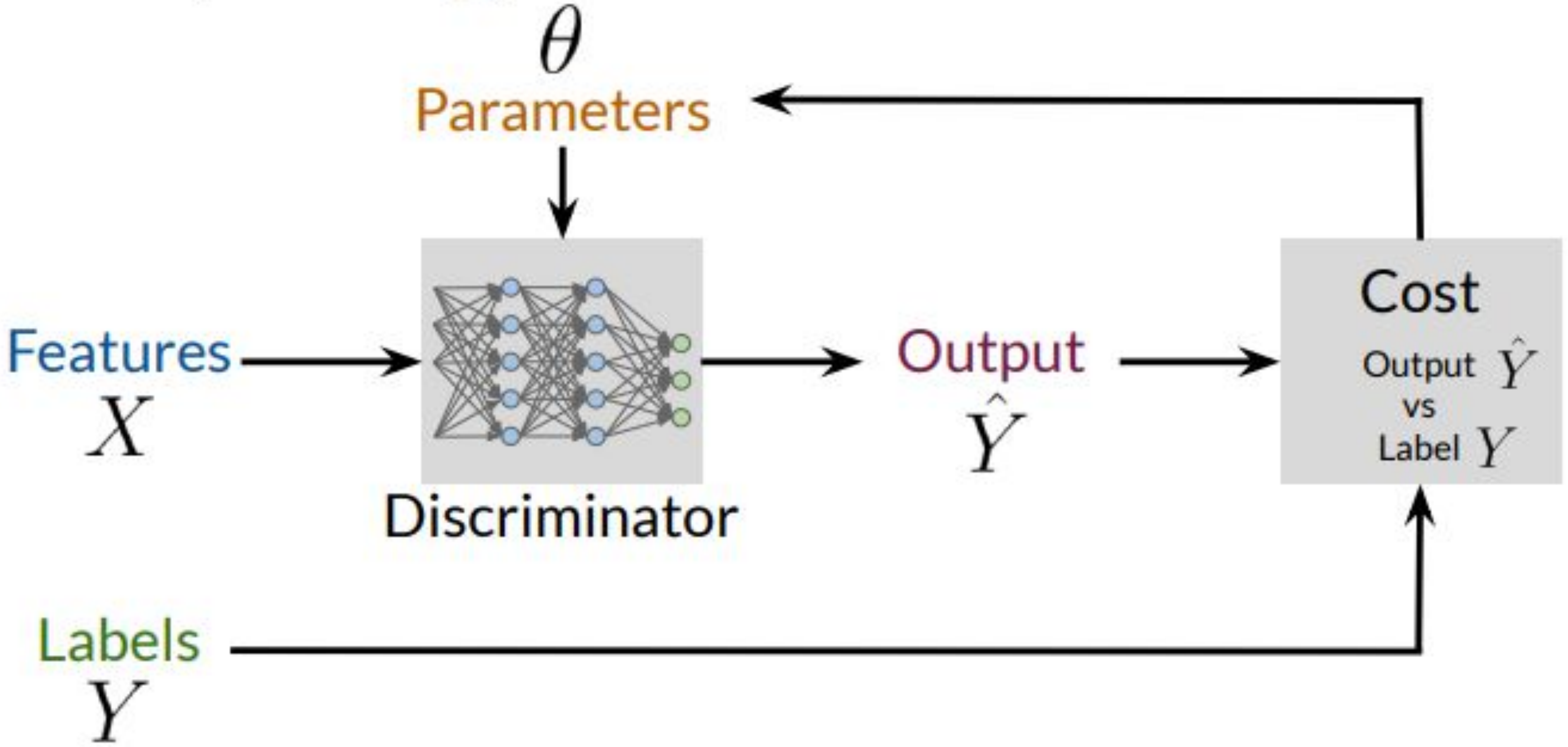
Distinguish between different classes



IT can classify text also

Discriminator

Classifiers (training)



Model probability of each class

Classifiers

Turtle

Bird

$P($ Cat $|$  $)$

Dog

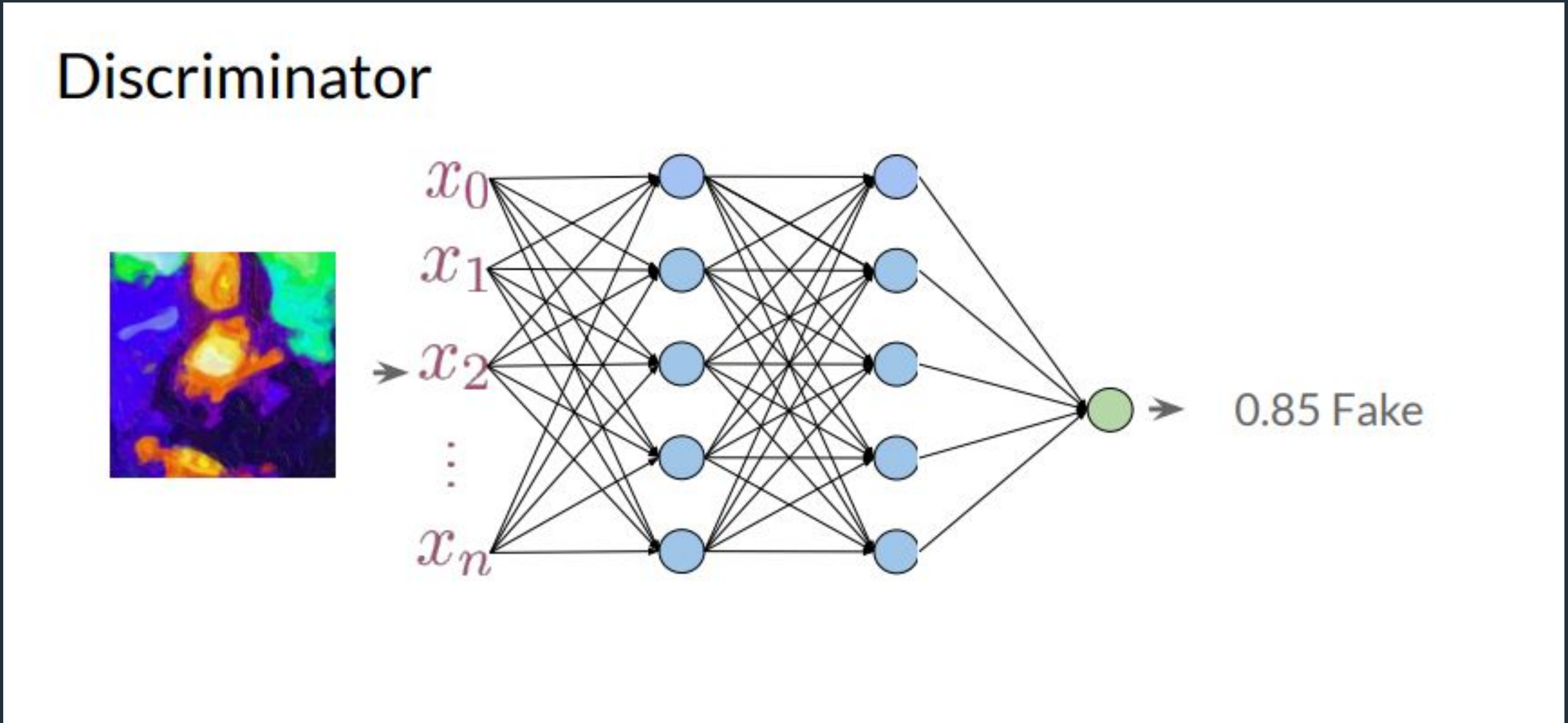
Fish

Classifiers

Conditional Probability

$$P(\underset{\text{Class}}{Y} \mid \underset{\text{Features}}{X})$$

Classify real vs fake (rather than cat, dog or bird)

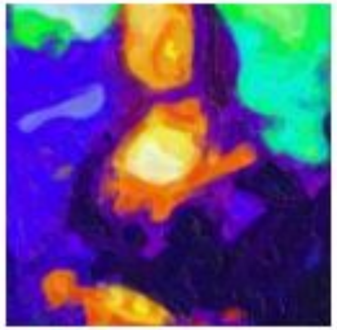


RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

Discriminator

$$P(\underset{\text{Class}}{\text{Fake}} \mid \underset{\text{Features}}{X})$$

Discriminator

$P($ Fake $|$  $) = 0.85 \longrightarrow \boxed{\text{Fake}}$

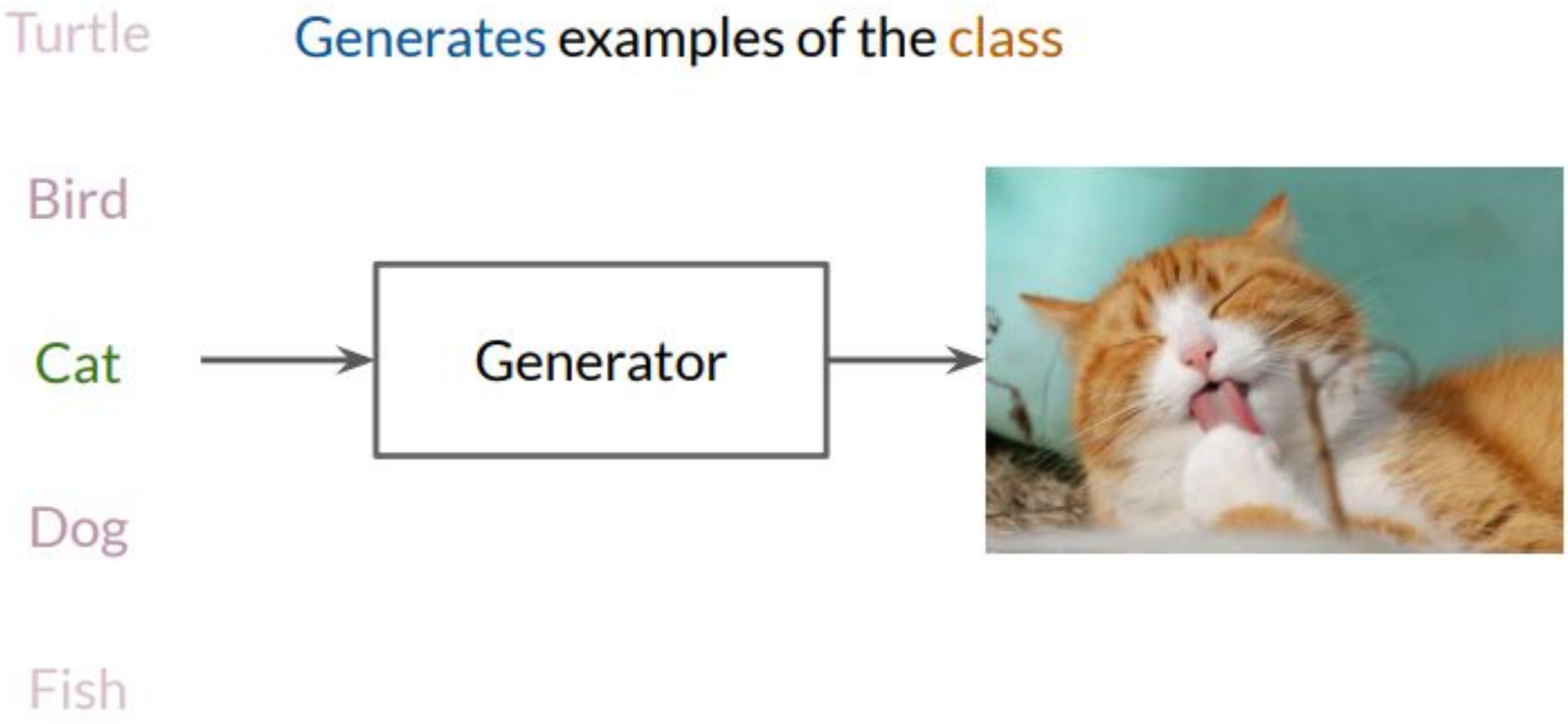
Class   Features

Summary

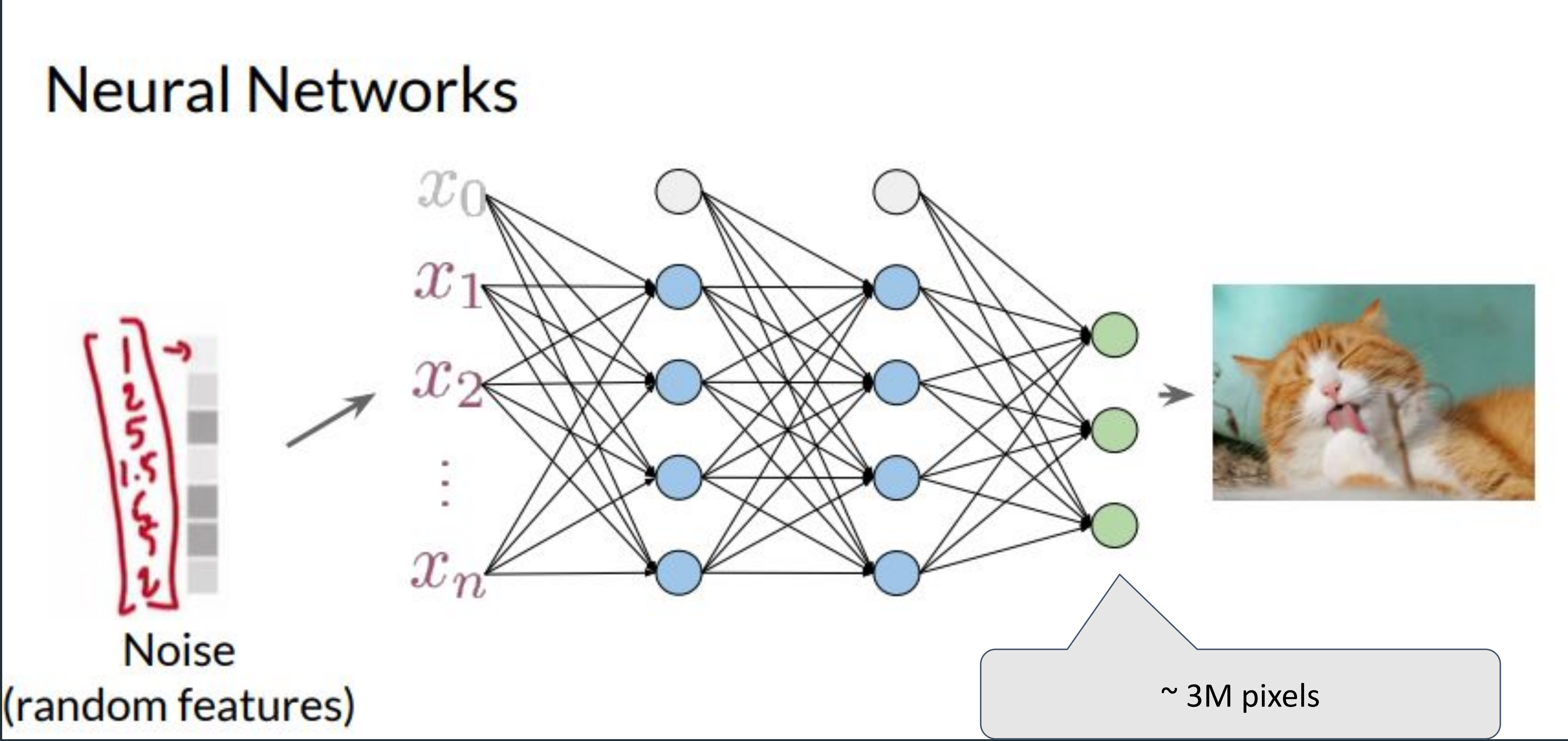
- The discriminator is a classifier
- It learns the probability of class Y (real or *fake*) given features X
- The probabilities are the feedback for the generator

Generator

Generates examples of the class

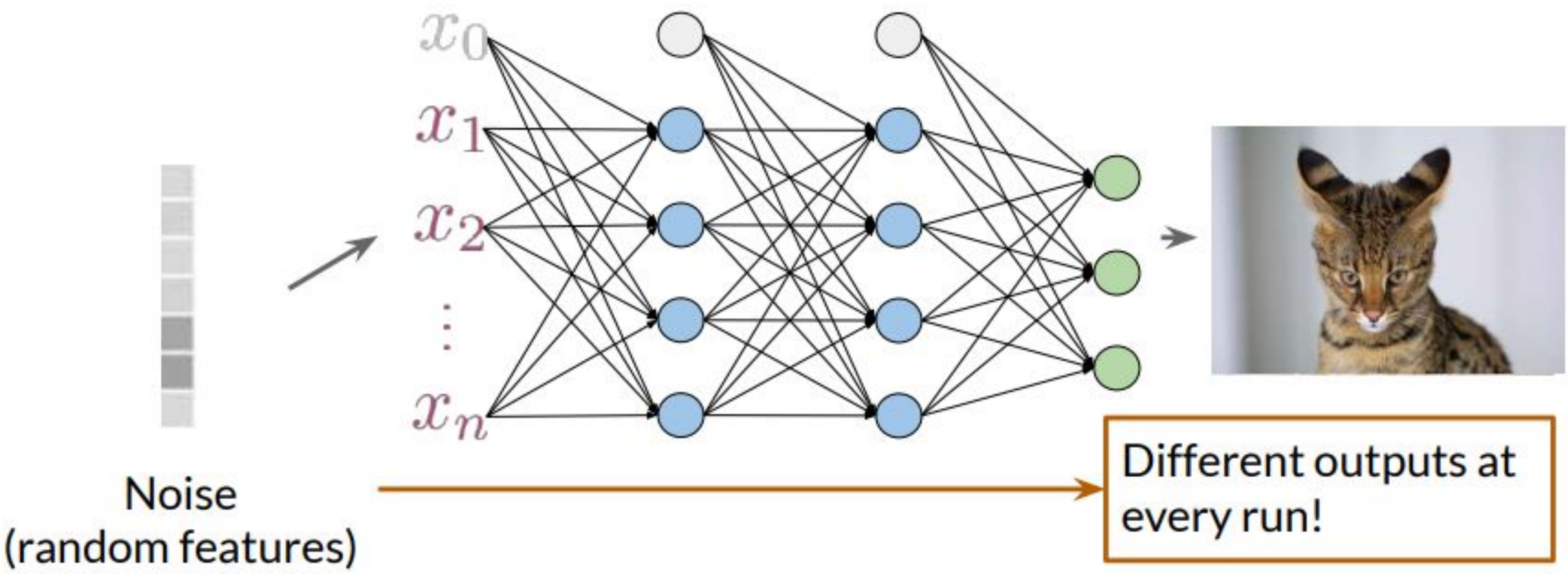


Generate different examples at different runs

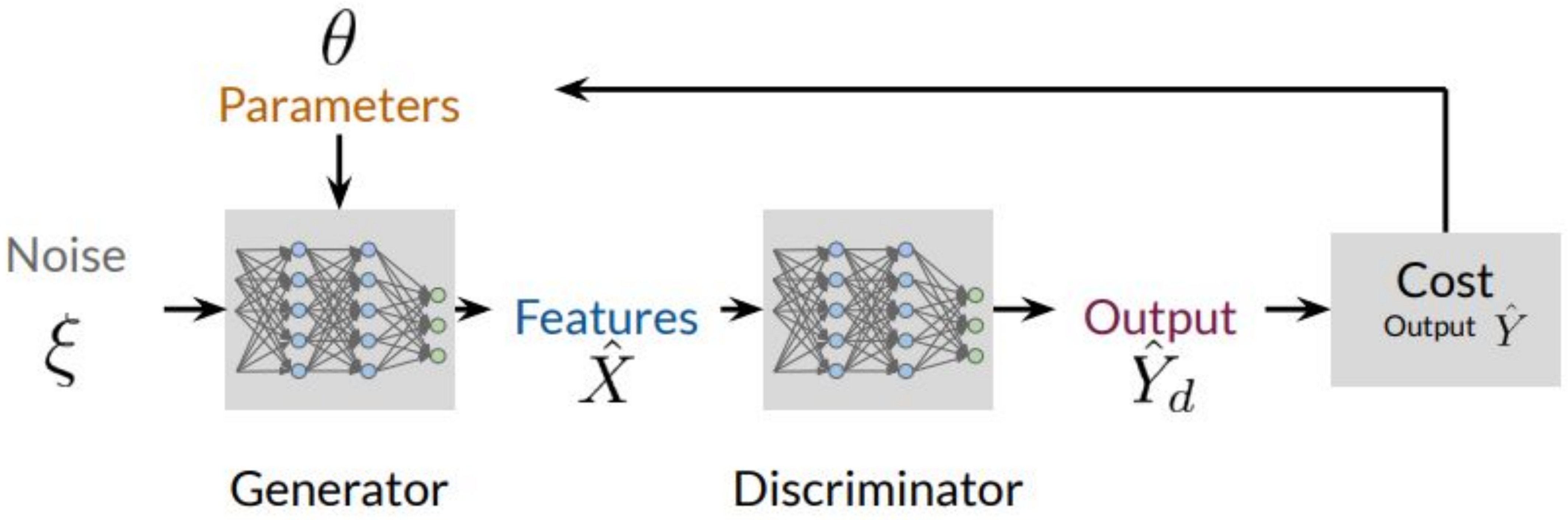


~ 3M pixels

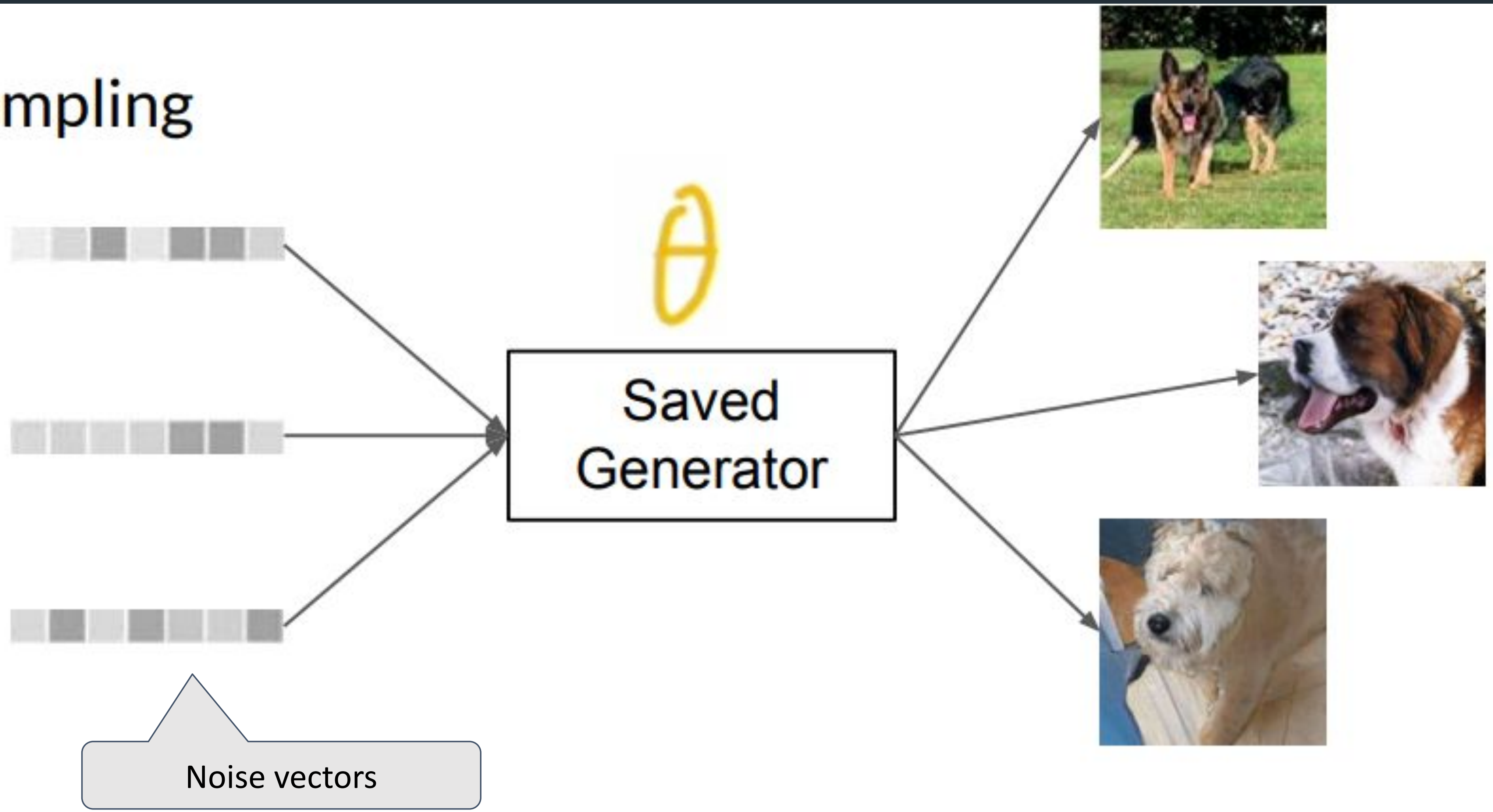
Neural Networks



Generator: Learning



Sampling



Generator

Turtle

Bird

$P($  $|$ Cat $)$

Dog

Fish

Generator

Conditional Probability

$$P(\underset{\text{Features}}{X} \mid \underset{\text{Class}}{Y})$$

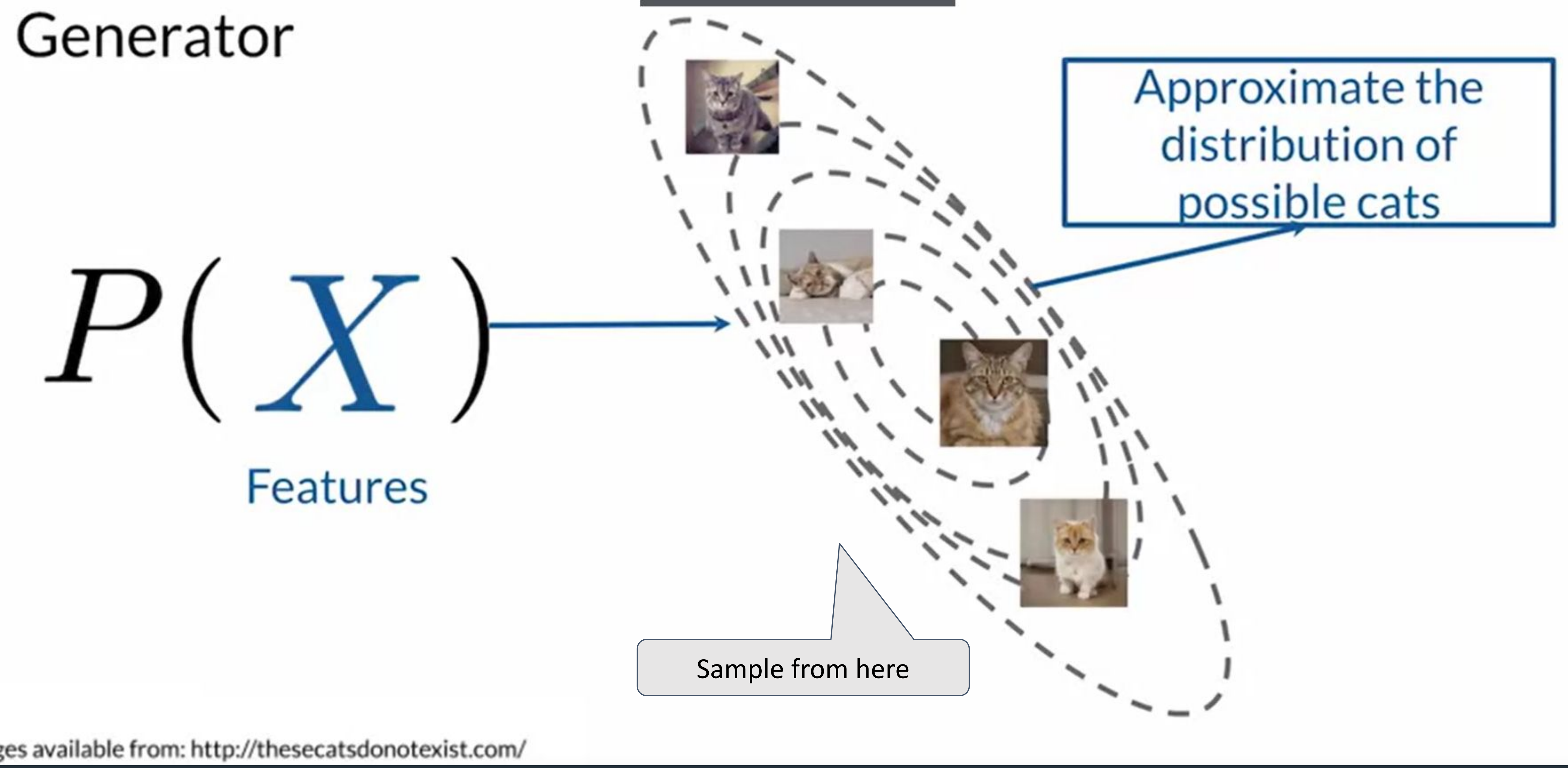
When only one class, no Y

Generator

Conditional Probability

$$P(\underbrace{X}_{\text{Features}} \mid \underbrace{\cancel{Y}}_{\text{Class}})$$

More common types are more likely to be generated



Summary

- The generator produces fake data
- It learns the probability of features X
- The generator takes as input noise (random features)

Binary cross entropy function

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h(x^{(i)},\theta) + (1-y^{(i)})\log(1-h(x^{(i)},\theta))]$$

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h(x^{(i)},\theta) + (1-y^{(i)})\log(1-h(x^{(i)},\theta))\right]$$

Prediction

Label

Features

Parameters

Average loss of the whole batch

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\boxed{y^{(i)}\log h(x^{(i)};\theta)} + \boxed{(1-y^{(i)})\log(1-h(x^{(i)},\theta))}\right]$$

When is this term relevant?
What happens when the prediction is close?
When the prediction is very bad?

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\boxed{y^{(i)}\log h(x^{(i)};\theta)} + \boxed{(1-y^{(i)})\log(1-h(x^{(i)},\theta))}\right]$$

	$y^{(i)}$	$h(x^{(i)},\theta)$	$y^{(i)}\log h(x^{(i)},\theta)$
Fake	0	any	0
real	1	0.99	~0
	1	~0	-inf

Relevant when the label is 1

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\boxed{y^{(i)}\log h(x^{(i)},\theta)} + \boxed{(1-y^{(i)})\log(1-h(x^{(i)},\theta))}\right]$$

How about this term?

BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\boxed{y^{(i)} \log h(x^{(i)}, \theta)} + \boxed{(1 - y^{(i)}) \log(1 - h(x^{(i)}, \theta))}\right]$$

$y^{(i)}$	$h(x^{(i)}, \theta)$	$(1 - y^{(i)}) \log(1 - h(x^{(i)}, \theta))$
1	any	0
0	0.01	~0
0	~1	-inf

Relevant when the label is 0

BCE Cost Function

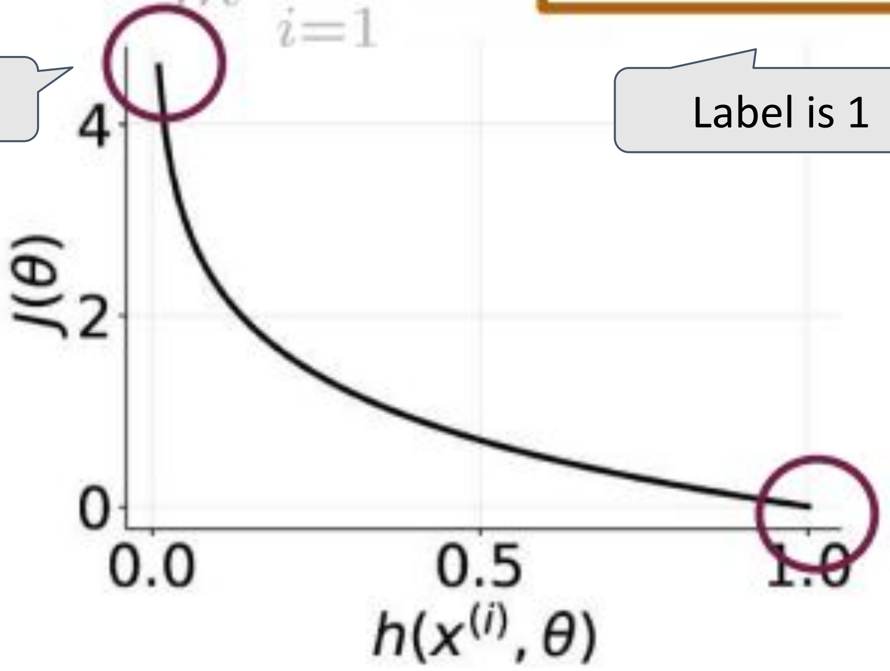
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[\boxed{y^{(i)} \log h(x^{(i)}, \theta)} + \boxed{(1 - y^{(i)}) \log(1 - h(x^{(i)}, \theta))}\right]$$

Both terms evaluate to -∞ when the predictions are bad which is how it needs to be as a cost function (so that it gets minimized during optimization)

Ensures that the cost is always greater or equal to 0

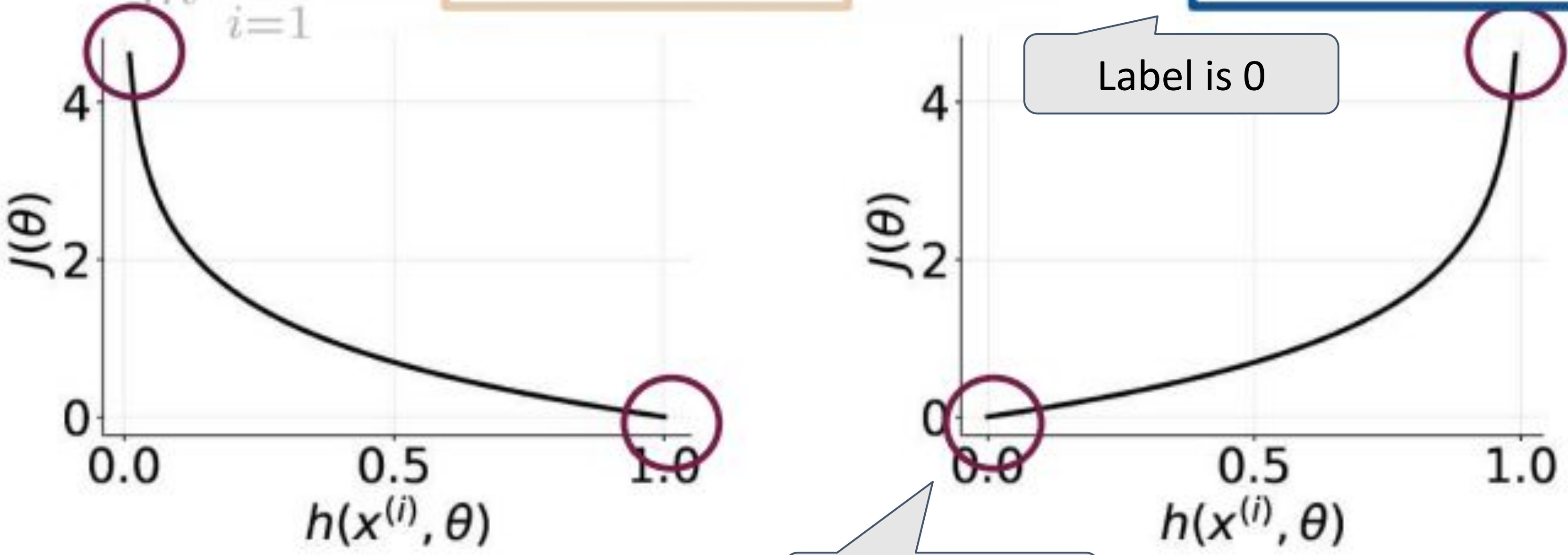
BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\boxed{\log h(x^{(i)},\theta)} + (1-y^{(i)})\boxed{\log(1-h(x^{(i)},\theta))}\right]$$



BCE Cost Function

$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\boxed{\log h(x^{(i)},\theta)} + (1-y^{(i)})\boxed{\log(1-h(x^{(i)},\theta))}\right]$$



Label is 0

Ground truth is 0 bu prediction is 1, loss is ∞

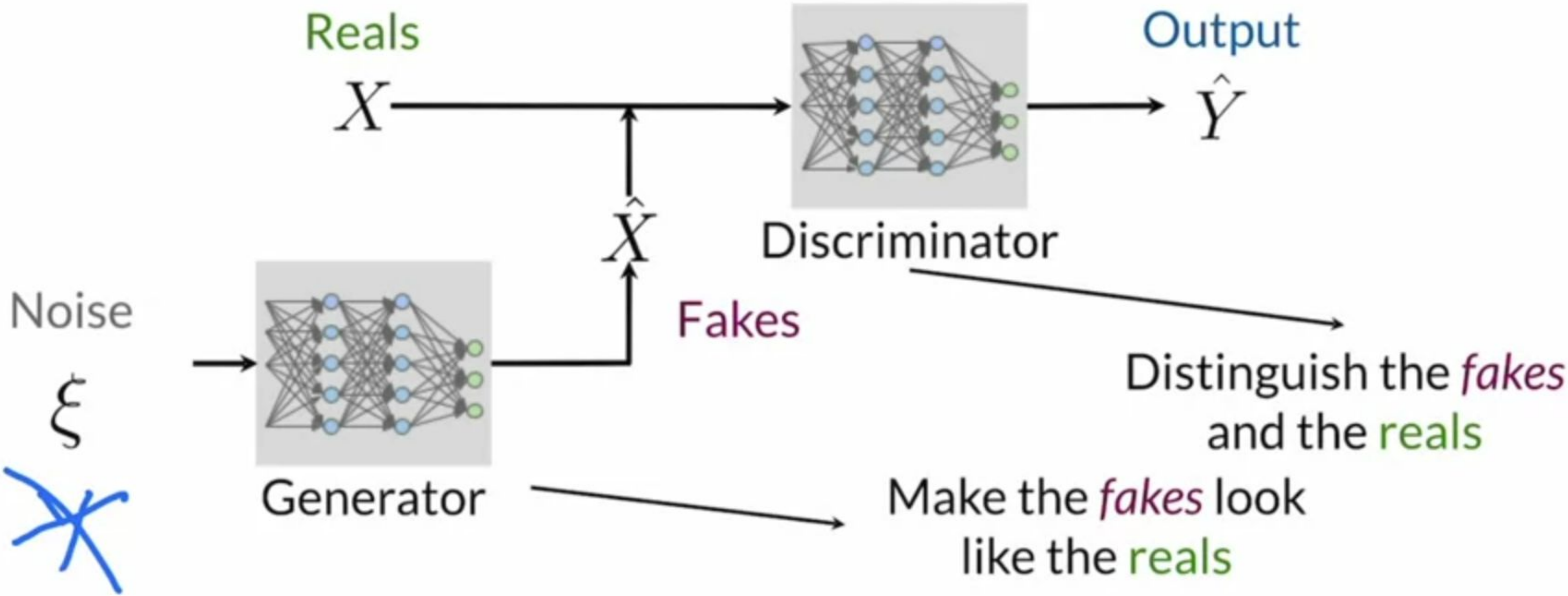
Loss is ~ 0

- In summary, the BCE cost function has two main terms that are relevant for each of the classes
  - When the prediction and the label are similar, the BCE loss is close to 0
  - When they are very different, BCE loss approaches infinity
- The BCE loss is performed across a mini-batch of several examples, say n examples
  - It takes the average of all the examples
  - Each of those examples can be different. One of them can be 1, the other four could be 0, for their different classes.

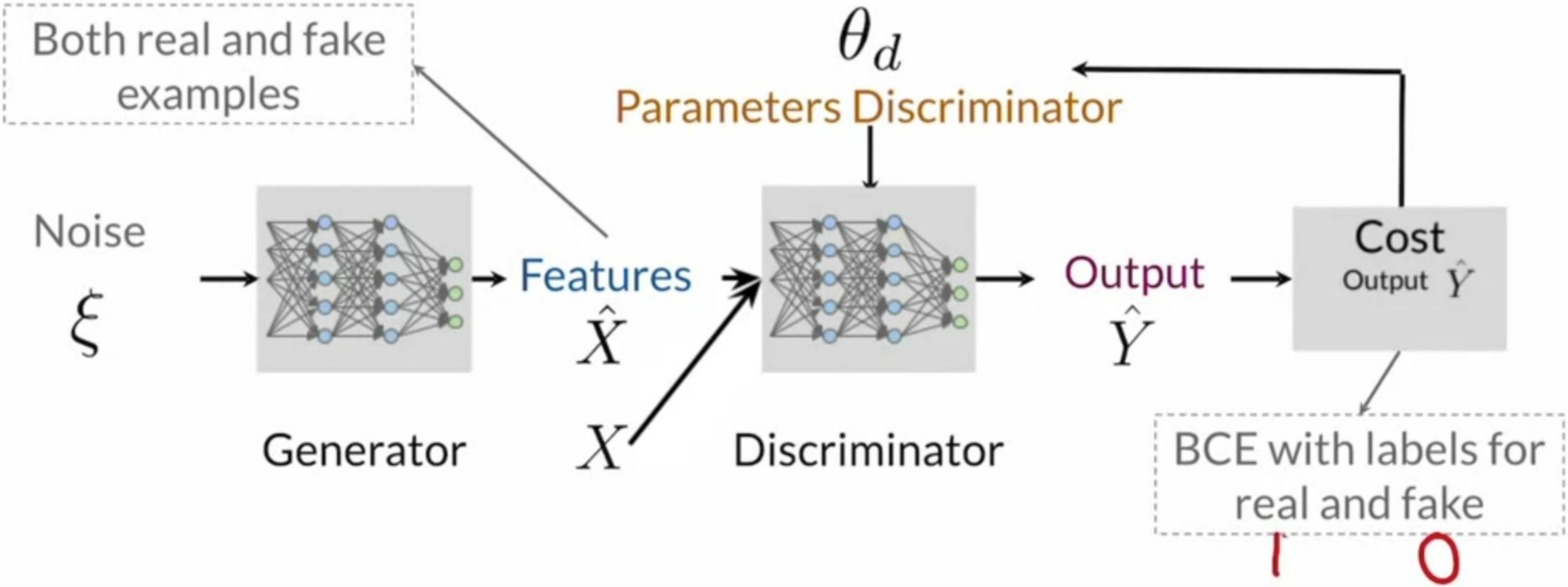
Putting all together

RV UNIVERSITY
Go, change the world
an initiative of RV EDUCATIONAL INSTITUTIONS

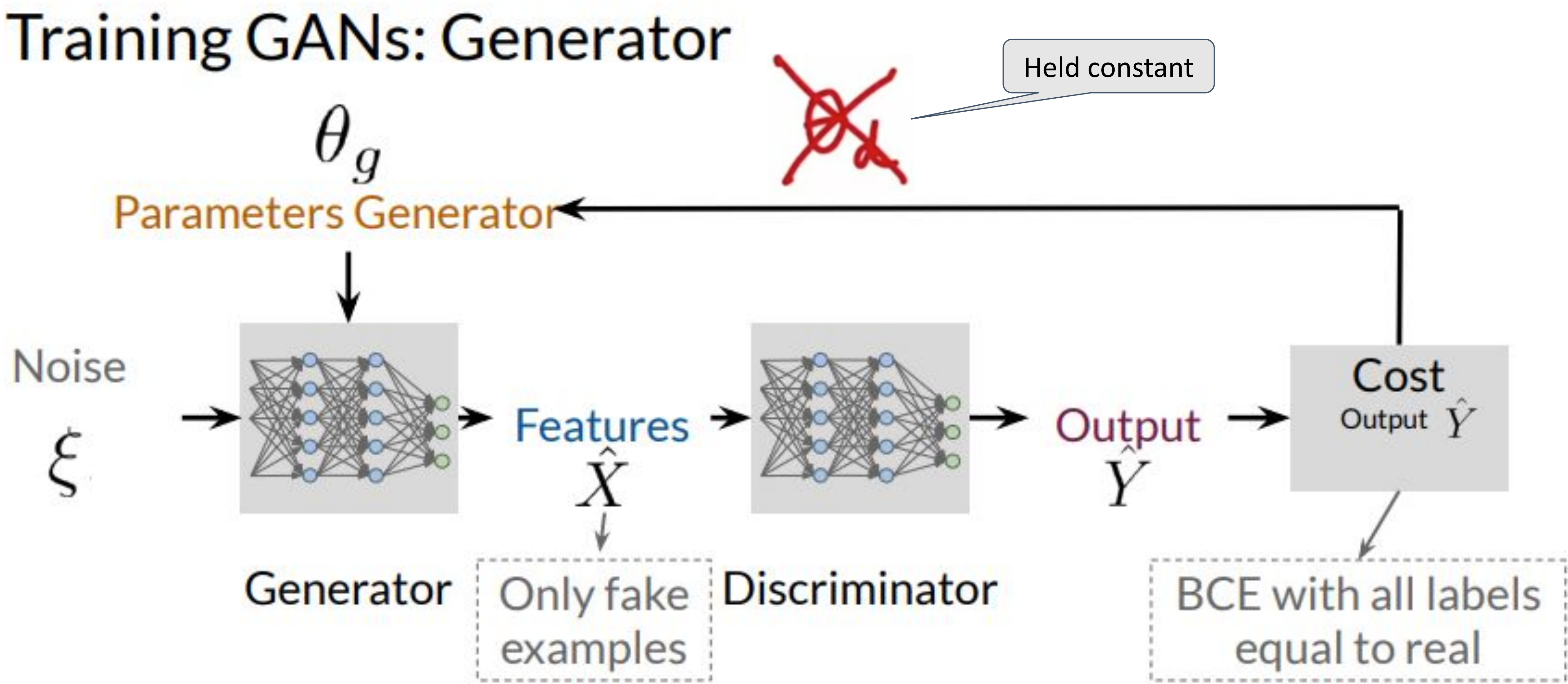
GANs Model



Training GANs: Discriminator

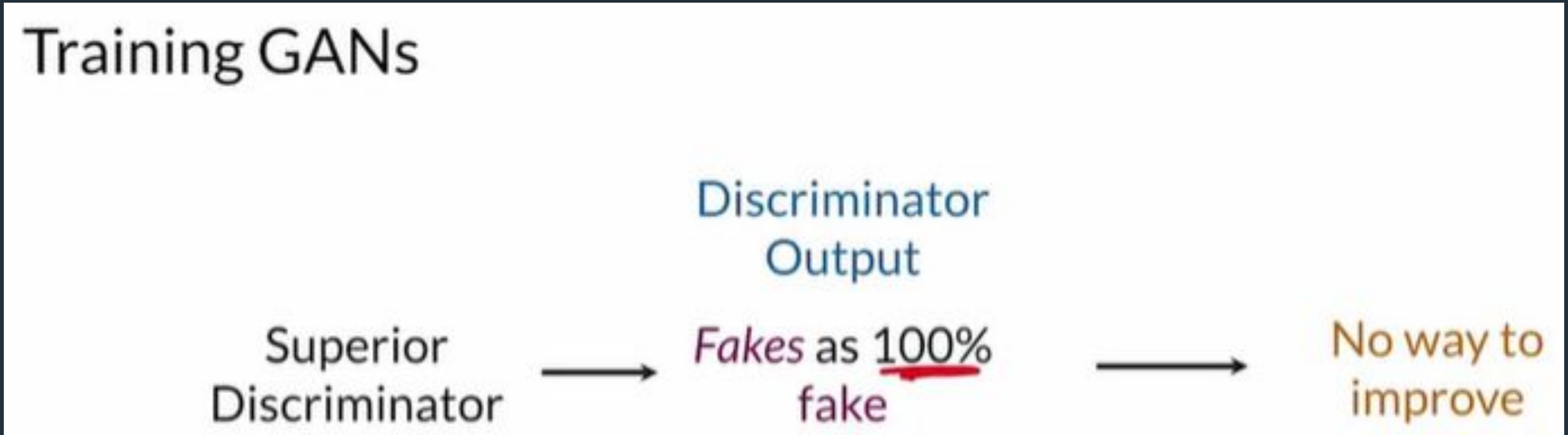


The two models are trained alternatively



Keep both models at the same level

- GANs train in alternating fashion
- The two models should always be at a similar "skill" level
  - If one model significantly better than the other, it doesn't help the other learn because the feedback is not useful.
    - Imagine if you were a beginning artist, and you showed your work to an art expert, asking whether your painting looked like a famous piece and all they said was 'no'. Because they have a very discerning eye, they know your image is not right, but won't be able to tell you how close you are.



Handson Session-1

GAN