Generative Adversarial Networks

—

Eklavya Sarkar, Biometrics Security and Privacy

Generative Adversarial Networks (GANs)

2014

2015

2016

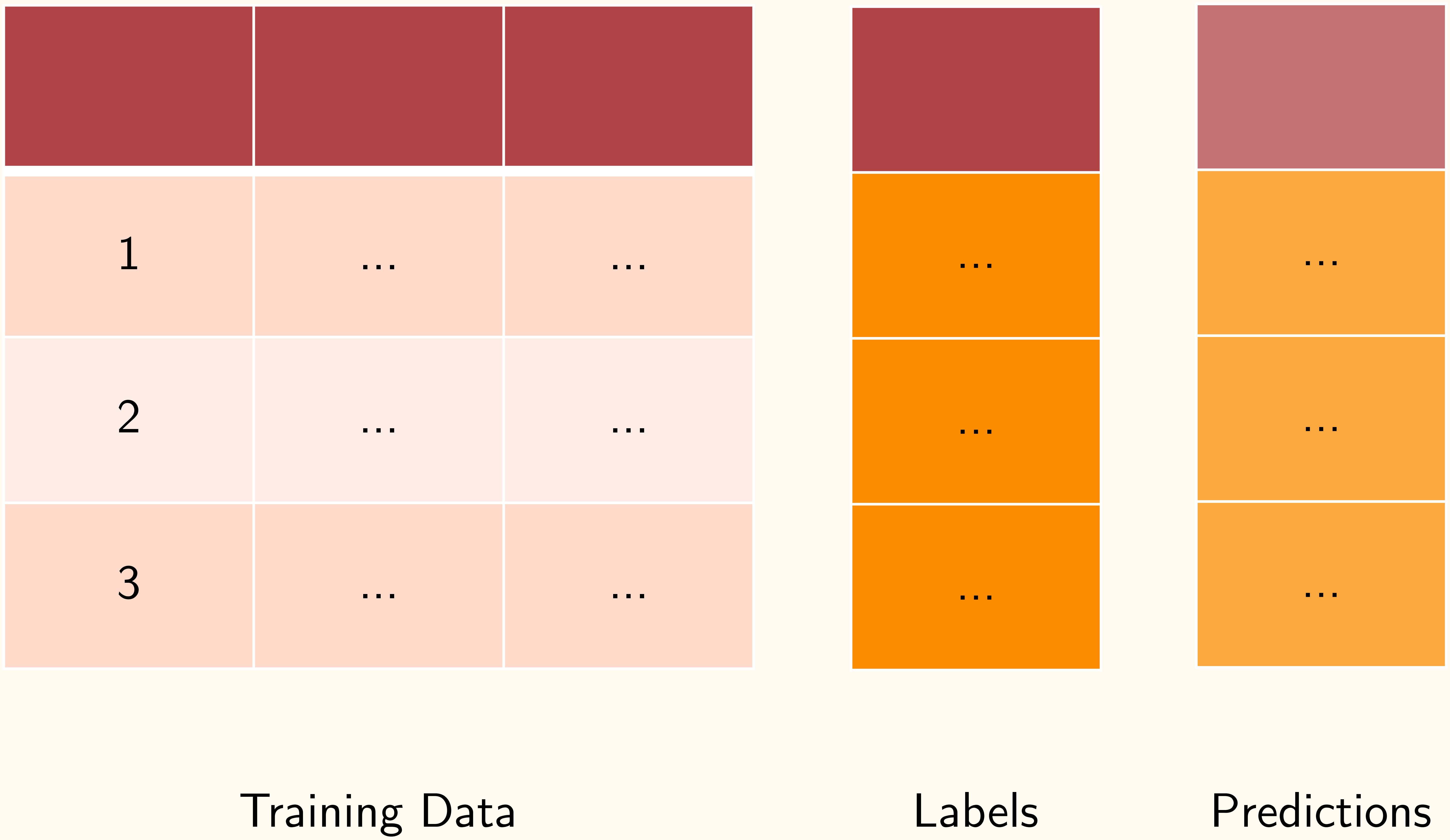
2017

2018

2019

- Generative modelling → progress in facial image generation
- Newer and more sophisticated methods every year

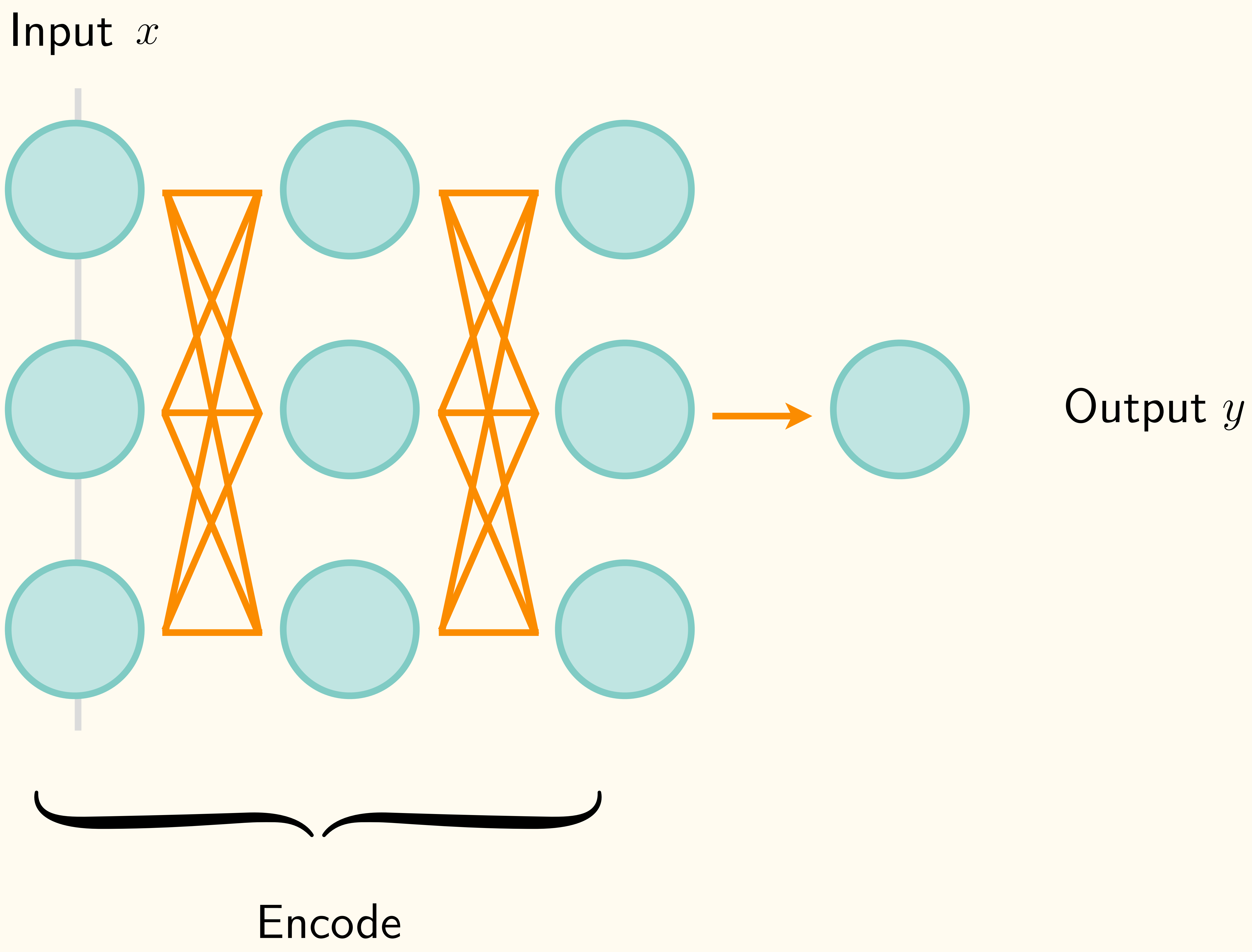
Supervised Learning



Supervised Artificial Neural Networks (ANNs):

- Take an *input*
- Produce an *output*
- Optimized with a *loss-function*

Supervised Learning



Supervised Artificial Neural Networks (ANNs):

- Take an *input*
- Produce an *output*
- Optimized with a *loss-function*

Unsupervised Learning

- Generative models are different.

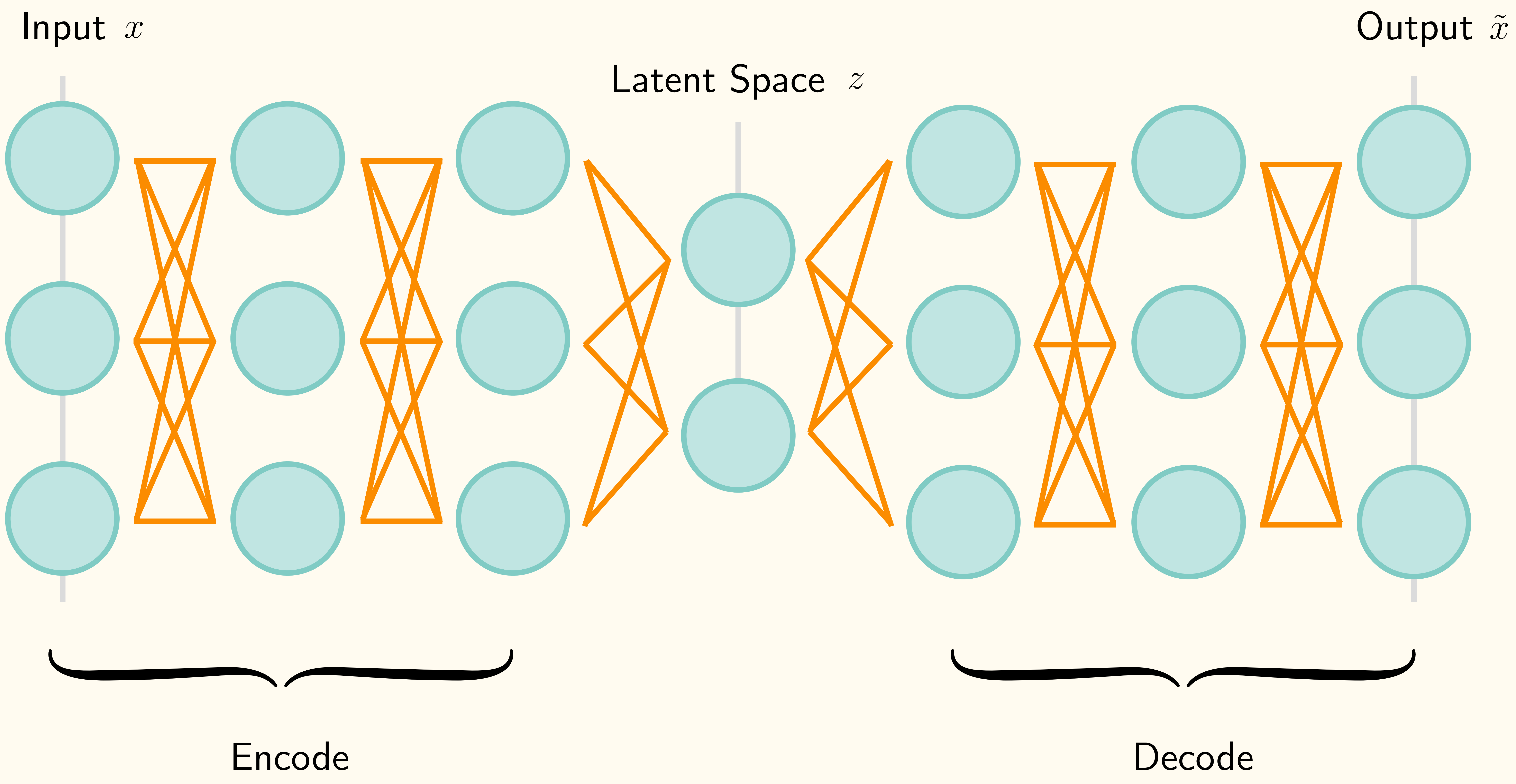
- Instead of taking a sample as *input*, they produce a sample as *output*.

Unsupervised Learning

- Train model on photographs of cats
  - Learn to produce new cat-like images
- Train it on known drug molecules
  - Generate new ‘drug-like’ molecules to use as candidates in a virtual screen.
- A generative model is trained on samples which are drawn from a:

  probability distribution
- Its job is to produce new samples from that same probability distribution.

Auto-Encoder



Auto-Encoder

- **Encoder**: samples → compressed representations
- **Decoder**: compressed representations in latent space → original samples

- What if we take *random vectors* in the latent space (picking a random value for each component of the vector) and pass them through the decoder ?

  - If everything goes well, the decoder should produce a completely *new sample* that *resembles* the ones it was trained on.

Auto-Encoder

- Problem ?
  - Doesn't work very well.

- Encoder may only produce vectors in a *small* region of the latent space.
- But if we pick a vector from *outside* that region ?
  - Output could look nothing like the training samples

- Decoder has only learned to work for the *particular* latent vectors from encoder
  - Not for *arbitrary* ones.

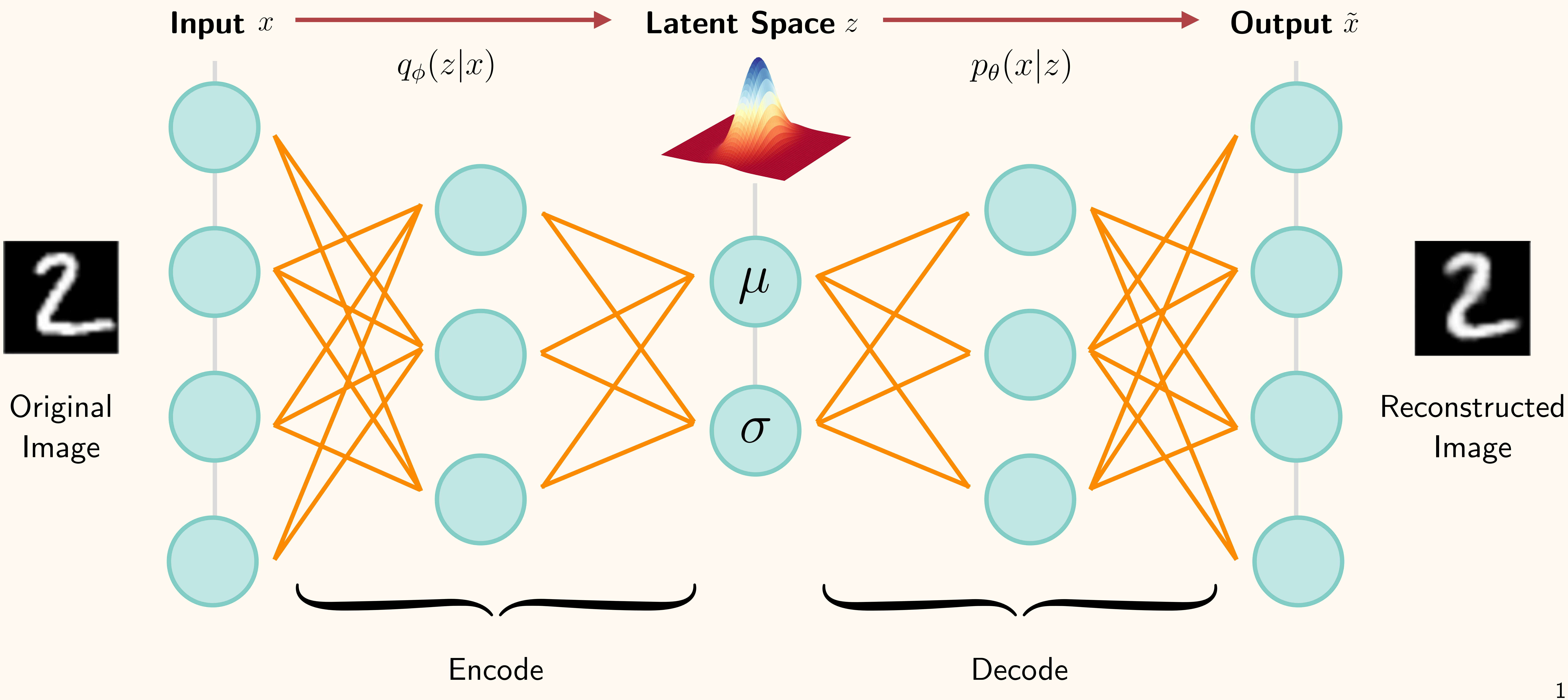
Variational Auto-Encoder (VAE)

- Solution ?
  - Variational Auto-Encoder

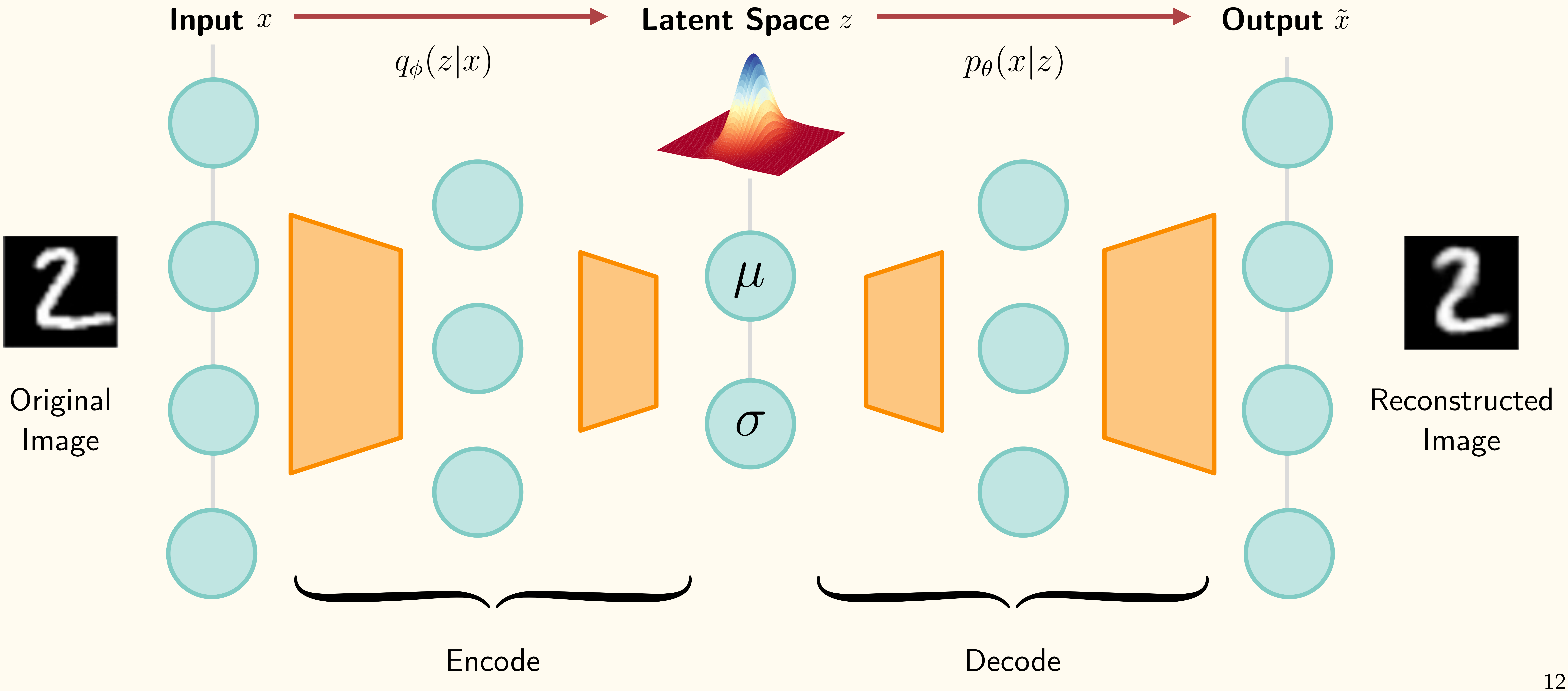
- Add a term to the loss function that *forces* the latent vectors to follow a specified distribution, e.g. $\mathcal{N} \sim (0,1)$
  - We can *expect* the decoder to work well on them

- We add random noise to the latent vector
  - Prevents decoder from being too sensitive to details

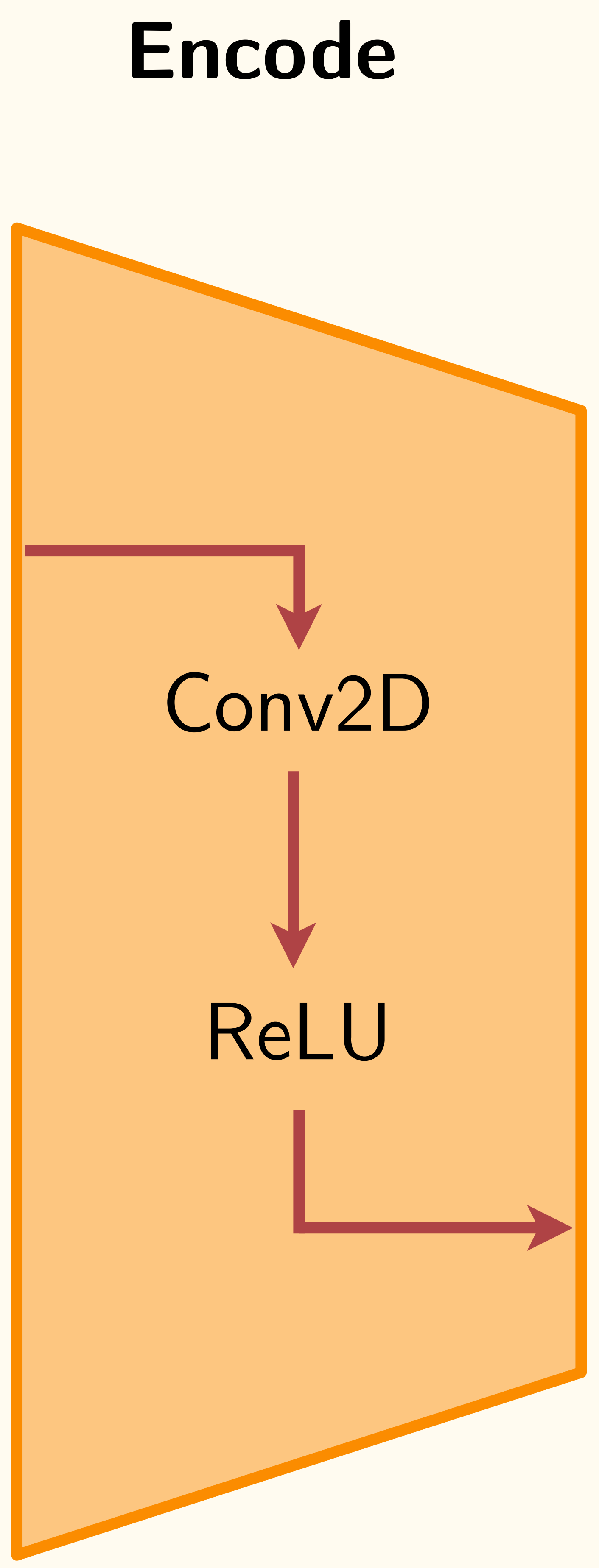
Variational Auto-Encoder (VAE)



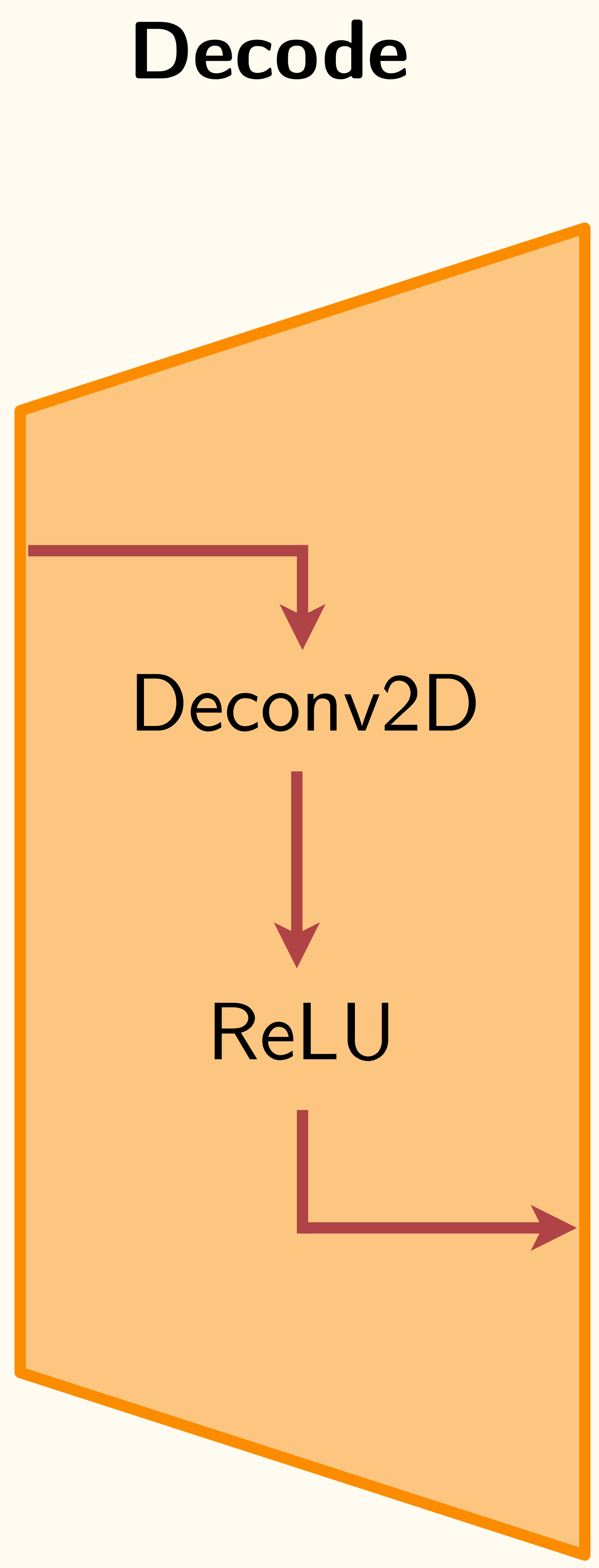
Simplified VAE Representation



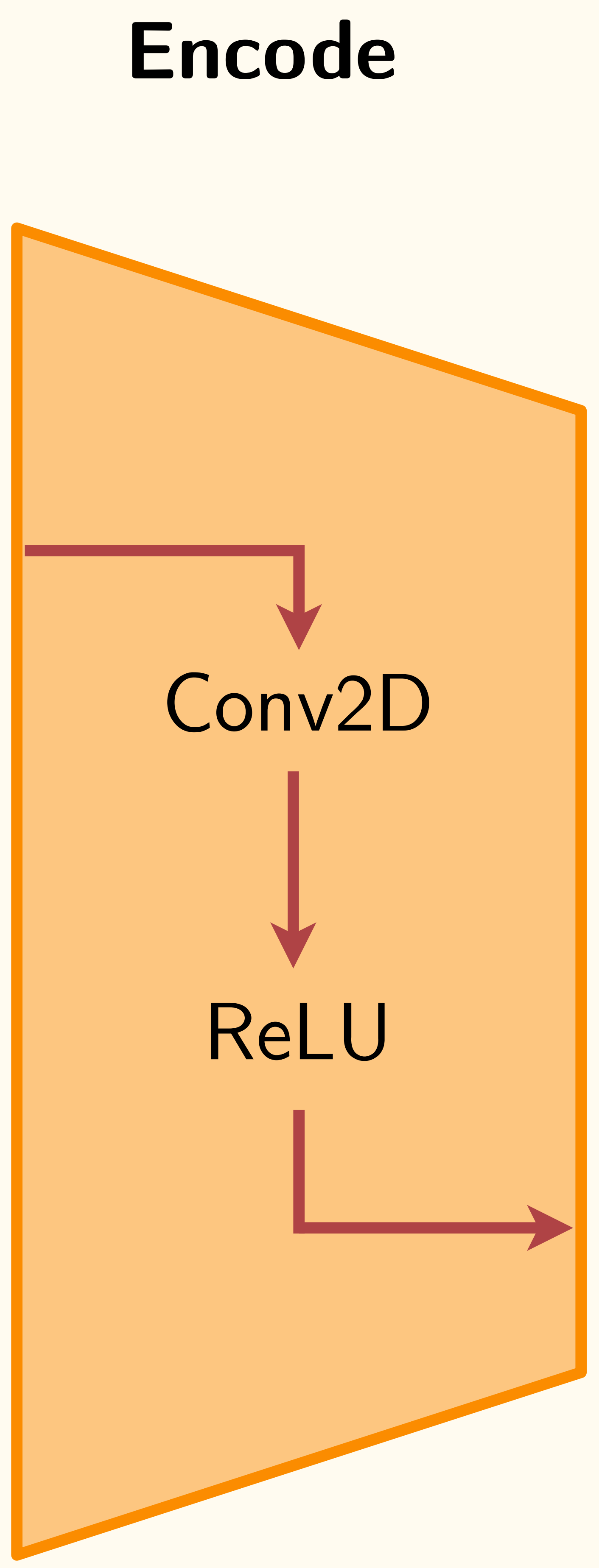
Encoder-Decoder



- An **encoder** is just a convolutional operation coupled with an activation function
- A **decoder** is de-convolutional operation and activation function
- De-conv2D is just a **transposed** Conv2D

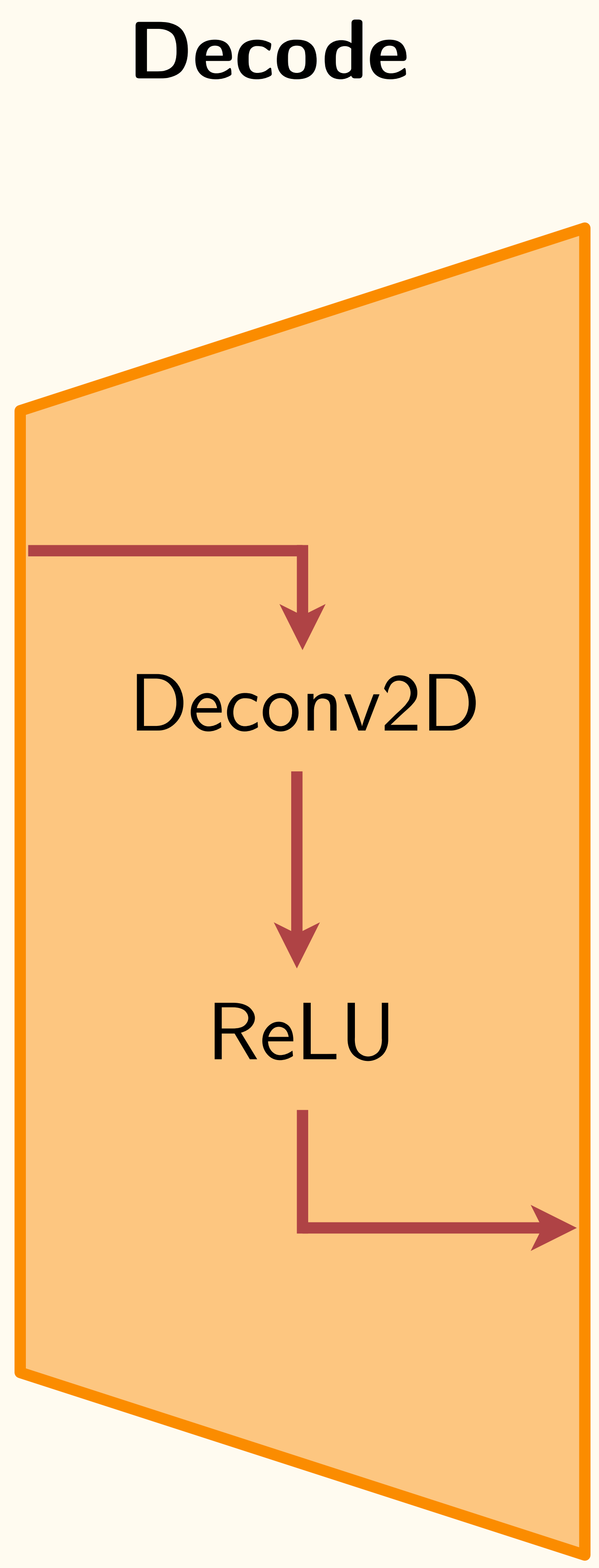


Encoder-Decoder

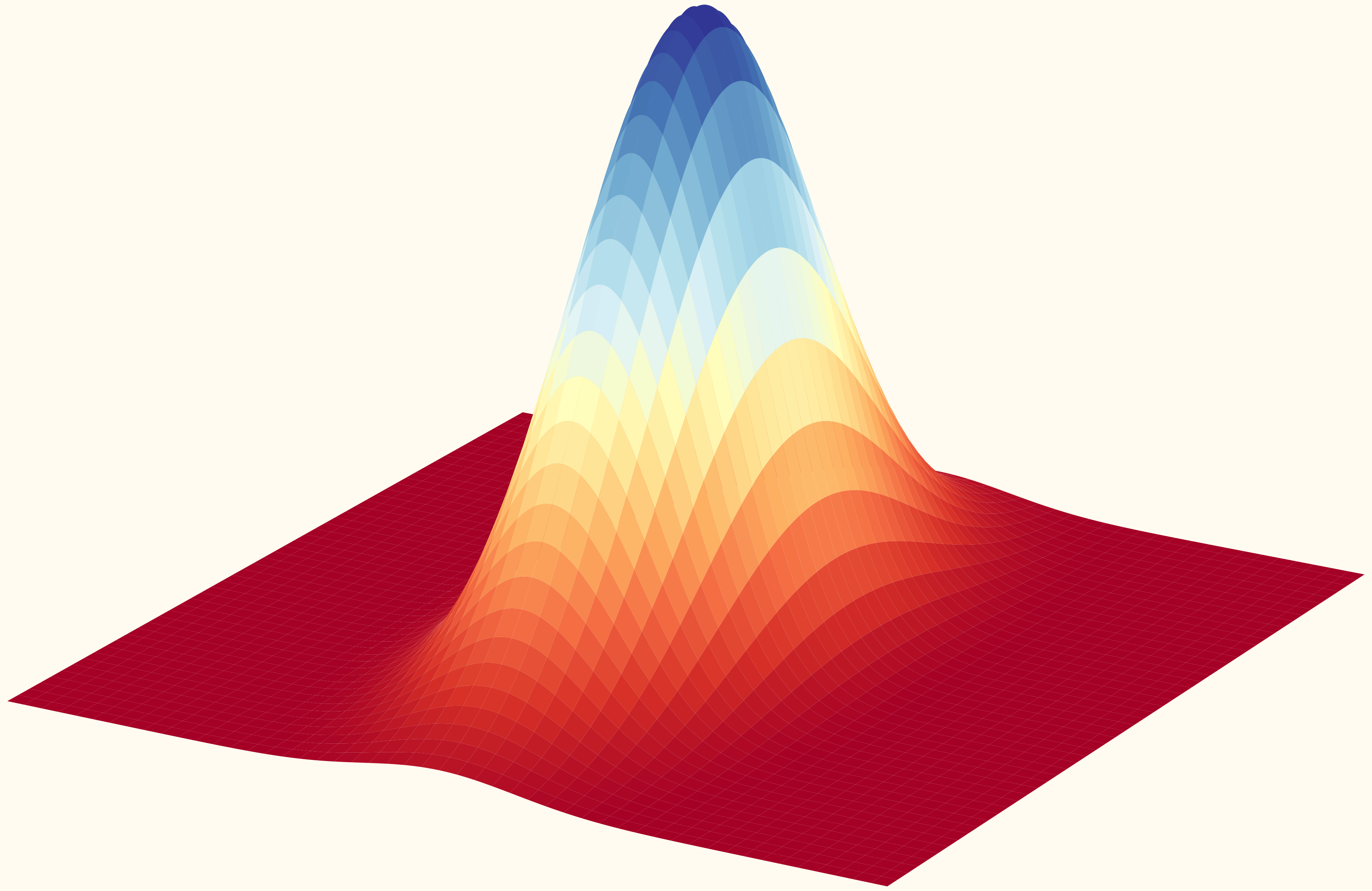



```
# Encoder
tf.nn.conv2d()
tf.nn.relu()

# Decoder
tf.nn.conv2d_transpose()
tf.nn.relu()
```



Latent Distribution



- The latent vector $z \sim (\mu, \sigma)$
- Compressed representation of input x

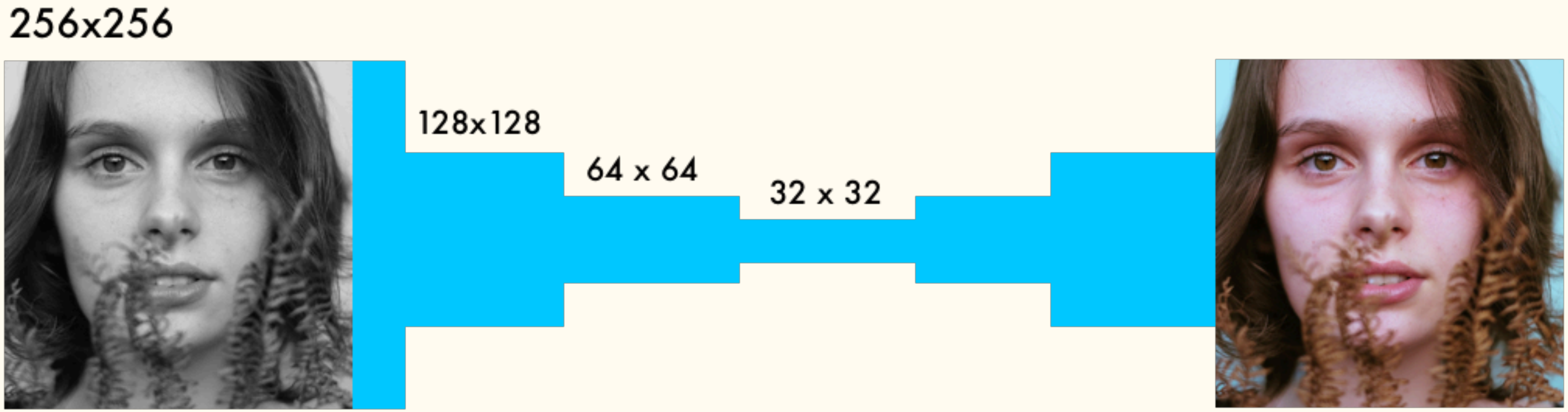
- $q_\Phi(z|x)$ encodes x to z
- $p_\theta(x|z)$ decodes z to x
- θ and Φ represent weights and biases

- Remember: we **sample** from the distribution $z \rightarrow$ lossy reconstruction

VAE Applications

Colorize B&W images

- Encoder-Decoder architecture
- Inception ResNet V2 Classifier



Emil Wallner, 2017. *Colorizing B&W Photos with Neural Networks*. FloydHub.

Generative Adversarial Network

- Works similarly to VAEs
  - Uses encoder-decoder system
  - Converts latent vectors into samples
  - ‘Generator’ instead of decoder

- But trains in a different way:
  - Passes random vectors into the generator
  - Directly evaluates the outputs on how well they follow the expected distribution

Generative Adversarial Network

- Essentially create a loss function to measure:
  - How well generated samples match the training samples
  - Use that loss function to optimise the model

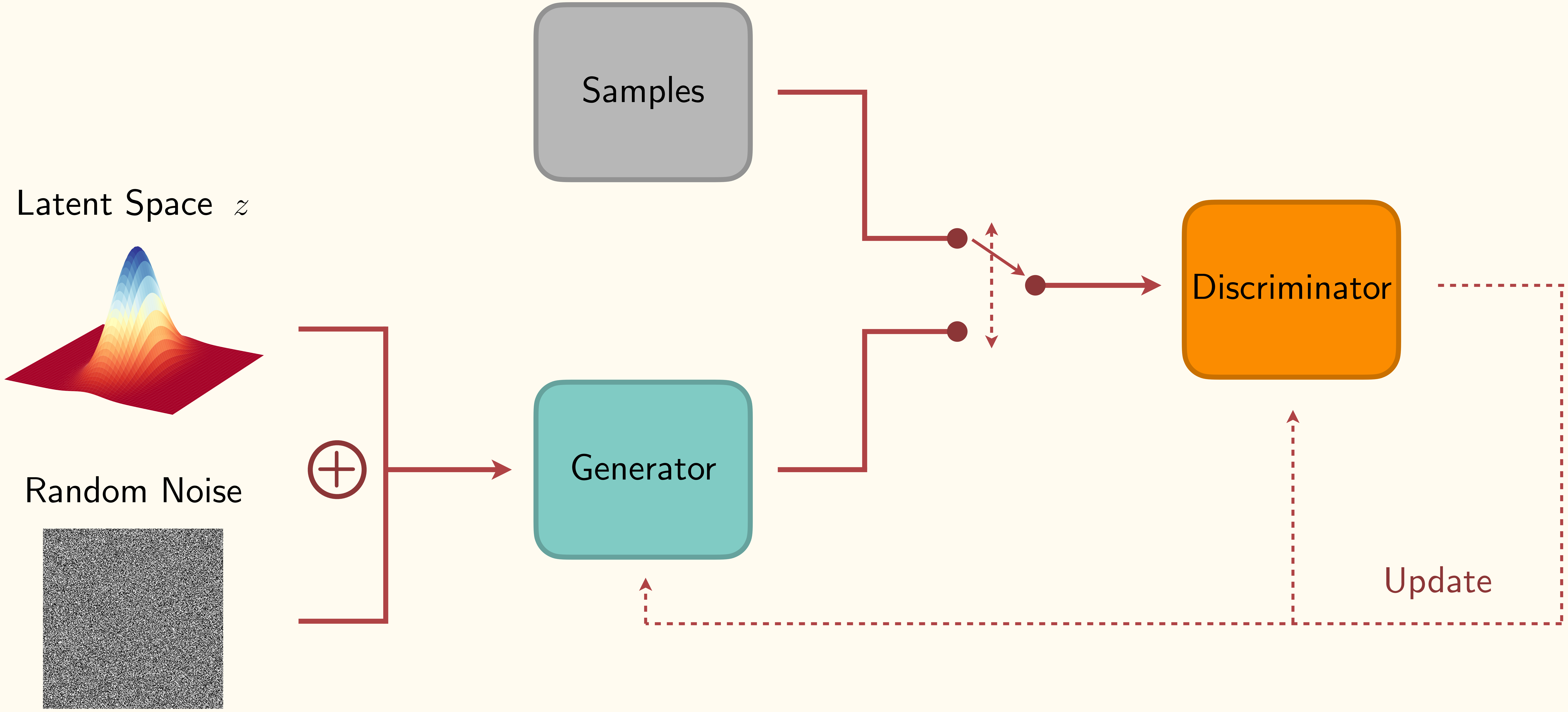
- How to make such a loss-function ?
  - You don't.
  - GAN learns the loss function from the data.

Generative Adversarial Network

- **Generator**: takes random vectors → generates synthetic samples

- **Discriminator**: tries to *distinguish* between 
  - The generated samples from real training samples
  - It takes a sample as input and outputs a probability of being a real sample
    - This acts as a loss function for the generator.

Generative Adversarial Network



Generative Adversarial Network

- **Generator** and **discriminator** are run *simultaneously*.

- Generator tries to ‘fool’ discriminator by presenting realistic samples
- Discriminator tries to get better at distinguishing real from fake samples

- Min-max game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Generative *Adversarial* Network**s**

GAN Applications

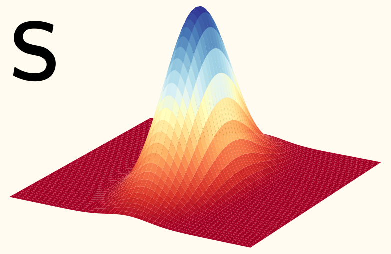
Colorize B&W images

- Even more advanced and realistic
- GAN based architecture

Jason Antic, 2018. *DeOldify: Colorizing and Restoring Old Images and Videos with Deep Learning*. FloydHub.

VAE vs GANs

Roughly:

- GANs tend to produce higher-quality samples 
- VAEs tend to produce higher-quality distributions 

- *Individual* samples generated by GANs more closely resemble training samples
- *Range* of samples generated by VAE more closely match range of training samples

Very active field of research → Things changing constantly !

Progression of GANs

- 2014: Original GAN
- 2015: Deep Convolutional GAN
- 2016: Coupled GAN
- 2017: Progressively Growing GAN
- 2018: Style-based GAN
- 2019: StyleGAN 2
- 2021: ?

2014

2015

2016

2017

2018

2019

StyleGAN 2

Source input image

Projection into FFHQ-StyleGAN2's W Latent Space

Projection into FFHQ-StyleGAN2's $W+$ Latent Space

Projection into Toon-StyleGAN2's Latent Space

Thank you !

Room 207-2, Idiap Research Institute

www.idiap.ch/~esarkar/

+41 78 82 50 754

eklavya.sarkar@idiap.ch

Generative Adversarial Network