Generative models

Cian M. Scannell, PhD
Assistant Professor
Medical Image Analysis
c.m.scannell@tue.nl

Topics:

- Introduction to generative modelling
- Latent variable models
- PCA
- Recap: U-Net
- Autoencoders
- Variational autoencoders

https://github.com/tueimage/8DM20

References

Main source: The book and blog posts of Jakub Tomczak
https://jmtomczak.github.io/

Learning objectives

The student can:

- Motivate the use of generative machine learning models
- Compare the benefits of generative and discriminative models
- Understand the term latent variable model
- Recall the PCA algorithm
- Relate autoencoder models to PCA
- Extend the idea of autoencoders to variational autoencoders
- Describe the VAE loss function
- Explain its derivation
- Discuss the reparameterization trick of VAEs
- Identify the limitations of VAEs

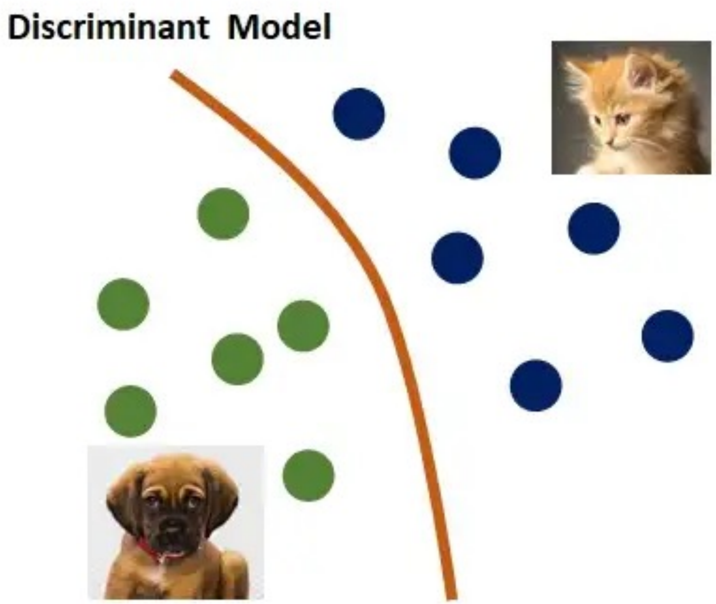
What is a generative model?

1. Generative modelling involves automatically discovering and learning the regularities or patterns in input data in such a way that the model can be used to generate or output new examples that plausibly could have been drawn from the original dataset.

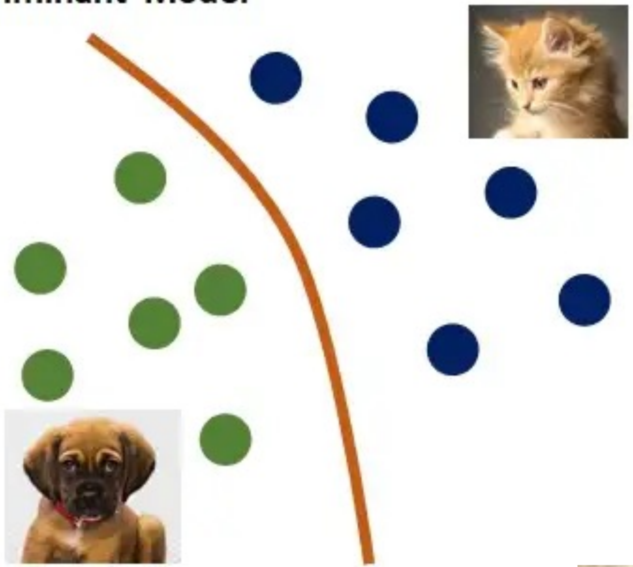
2. Statistically: it is a model of the joint probability distribution $P(X, Y)$ on a given observable variable X and target variable Y.
(Discriminative models capture only the conditional probability $P(Y|X)$)

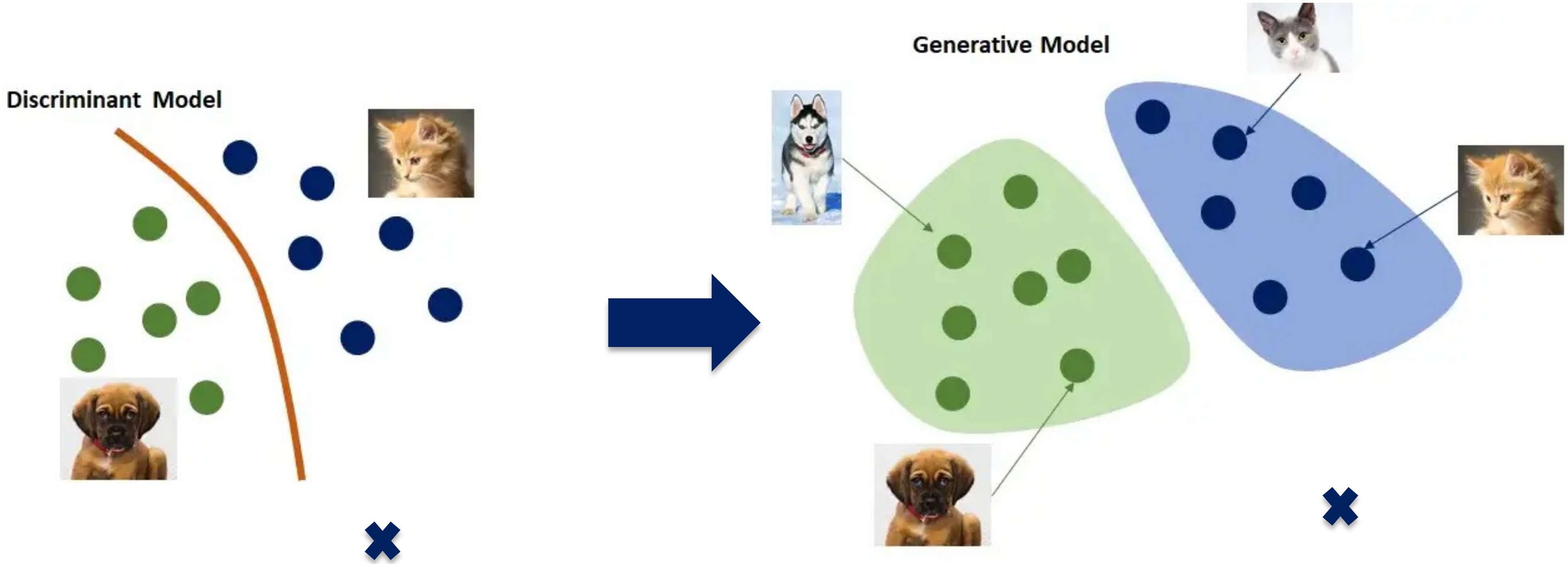
Why use generative models?

Motivation 1: better decision making


Discriminant Model

Discriminant Model





$P(y|x)$

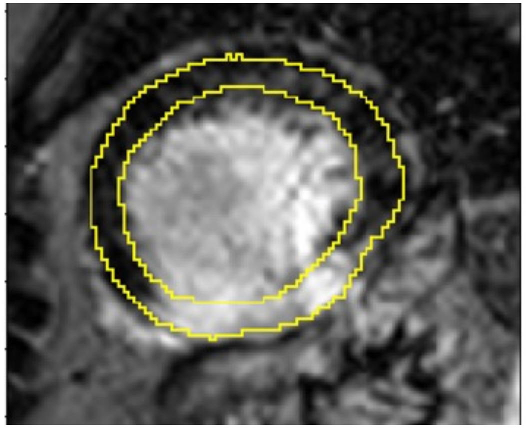
$P(y,x) = P(y|x) * P(x)$

$P(cat|x)$ is high so the decision is certain

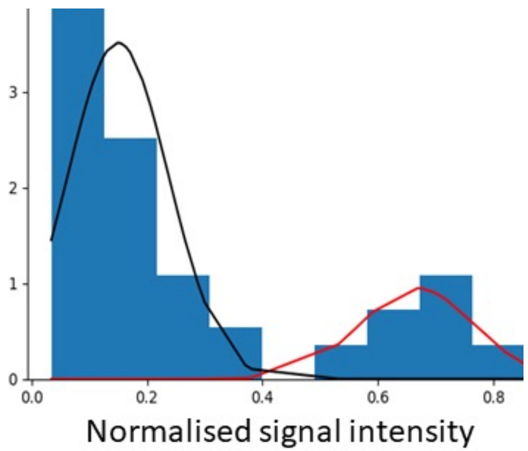
$P(cat|x)$ is high but $P(x)$ is low so the decision is uncertain

E.g. a Gaussian mixture model

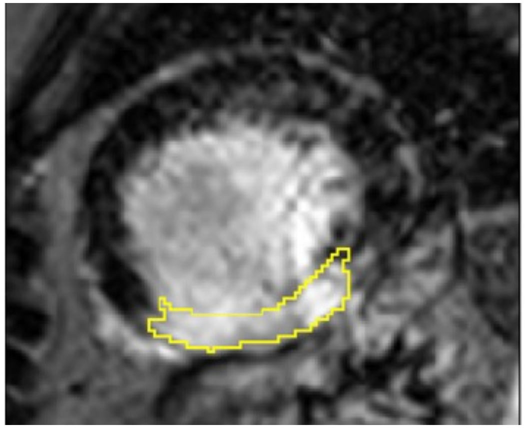
Cardiac MR scar imaging



Intensity histogram



Scar segmentation



Benefits of learning $P(x)$

- It could be used to assess whether a given object has been observed in the past or not.
- It could help to properly weight the decision.
- It could be used to assess uncertainty about the environment.
- It could be used to actively learn by interacting with the environment (e.g., by asking for labelling objects with low $P(x)$ - think of cat v dog example).

- Anomaly detection (normative learning)

- Sampling $\mathrm{x} \sim \boldsymbol{P}(\boldsymbol{x})$

Motivation 2: generating data

If you have learned the data distribution then you can sample from this distribution.

If you know what the data looks like then you can generate a new point.

$$x \sim \boldsymbol{P}(\boldsymbol{x})$$

T

write a biblical verse in the style of the king james bible explaining how to remove a peanut butter sandwich from a VCR

And it came to pass that a man was troubled by a peanut butter sandwich, for it had been placed within his VCR, and he knew not how to remove it.

And he cried out to the Lord, saying, "Oh, Lord, how can I remove this sandwich from my VCR, for it is stuck fast and will not budge?"

And the Lord spoke unto him, saying, "Fear not, my child, for I shall guide thy hand and show thee the way. Take thy butter knife, and carefully insert it between the sandwich and the VCR, and gently pry them apart. And with patience and perseverance, the sandwich shall be removed, and thy VCR shall be saved."

And the man did as the Lord commanded, and lo and behold, the sandwich was removed from the VCR, and the man was saved.

And the Lord said, "Verily I say unto thee, seek not to put thy peanut butter sandwiches in thy VCR, for it is not a suitable place for such things. Rather, keep thy sandwiches in thy refrigerator or on thy plate, where they belong."

And the man heeded the Lord's words, and from that day forth, he kept his sandwiches in their proper place, and was saved from trouble and woe. Amen.

https://thispersondoesnotexist.com/

Benefits of synthetic data (important for medical image analysis)

- It could increase the amount of available data.
- It could be used to modify the features of available data.
- It could be used to fill in missing points in the dataset.
- It could add uncertainty quantification and explainability to analysis.

Project aim to use synthetic data to supplement the training of a segmentation model

Do we already know any generative models?

Almost

PCA is not a generative model but it is an example of a latent variable model – a key idea for use going-forward

It can reconstruct approximations to datapoints from the lower dimensional latent space but does not generate new independent samples

Latent variable model

- A statistical model that includes variables that are not directly observed but are inferred from other variables that are observed

Latent variables

- Variables that are not directly measured but are estimated from observed data. They represent underlying factors that influence the observed variables.

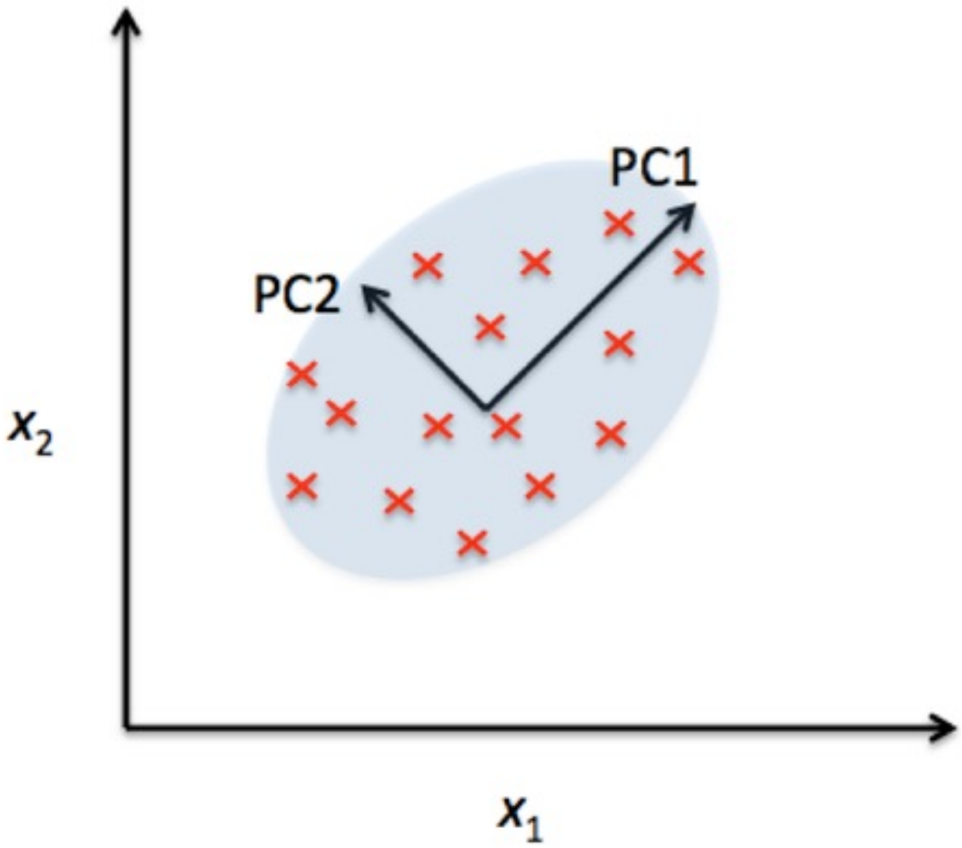
(e.g hair/ eye colour)

PCA recap

- Goal is to find a new representation to express the data set in with the constraint that the basis of the new representation is a linear combination of the original basis.

$$\boldsymbol{Y} = \boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{X}$$

$$Y = P^T X$$

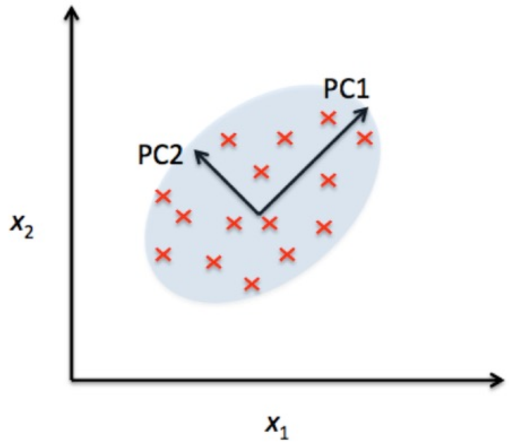


PCA recap

- We could show that the basis vectors of $\boldsymbol{P}$ are the eigenvectors of the covariance matrix of $\boldsymbol{X}$.
  - » $\widehat{\boldsymbol{X}}$ = normalise($\boldsymbol{X}$)
  - » Compute $\boldsymbol{UDV}^{\boldsymbol{T}}$ = SVD($\boldsymbol{X}$)
  - » Return $\boldsymbol{U}$: principal components and $\boldsymbol{D}^2$: amount of variance explained.

- Keep M principal components and discard others.
  - » Low-rank representation: $\boldsymbol{U}_{\boldsymbol{M}}^{\boldsymbol{T}}\boldsymbol{X}$.
  - » Reconstruction: $\boldsymbol{U}_{\boldsymbol{M}}\boldsymbol{U}_{\boldsymbol{M}}^{\boldsymbol{T}}\boldsymbol{X}$

(almost like generating)



PCA recap

- Keep M principal components and discard others.
  - » Low-rank representation: $\boldsymbol{U_M^T X}$.
  - » Reconstruction: $\boldsymbol{U_M U_M^T X}$

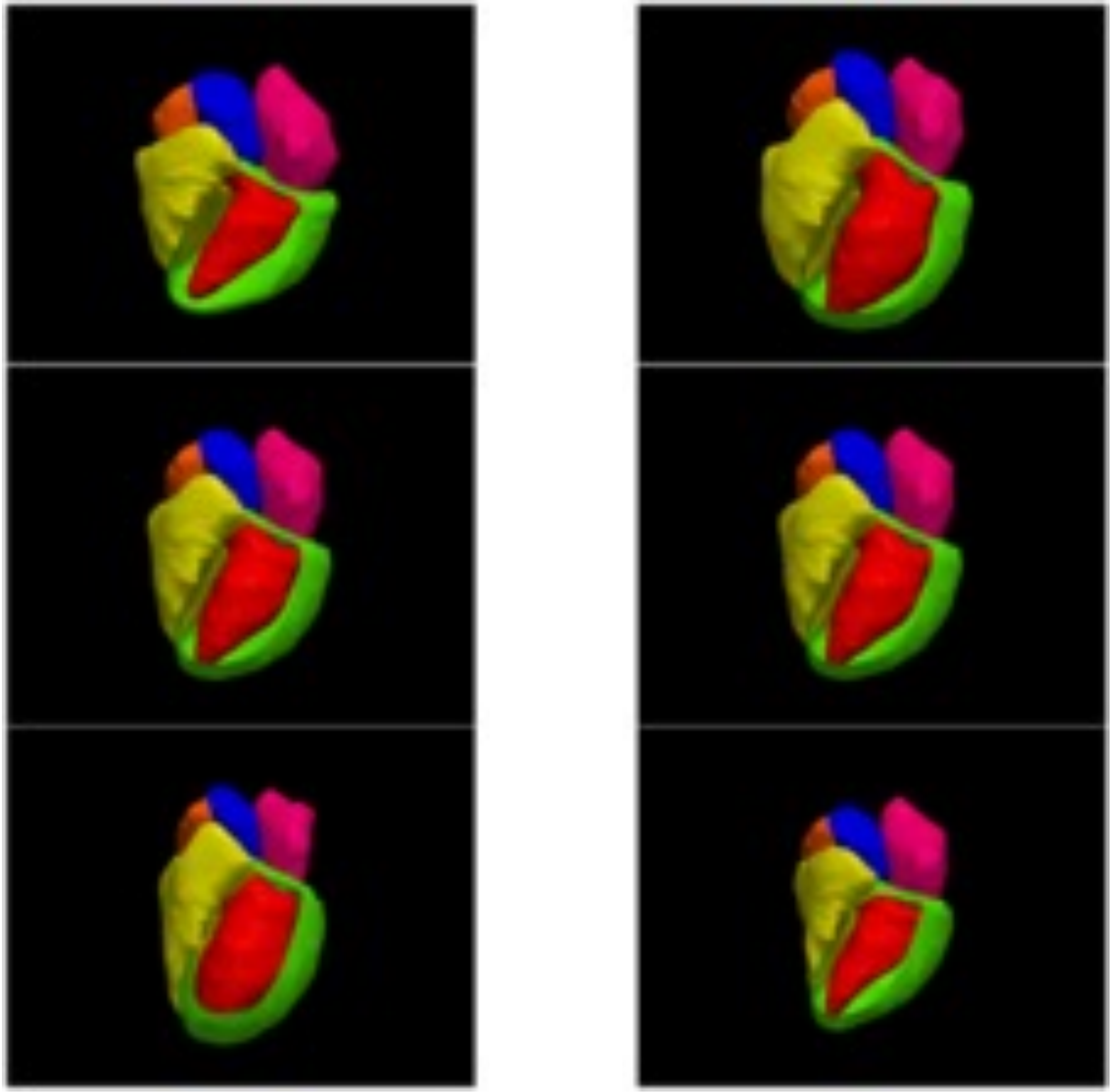
    (almost like generating)

PC1 PC2

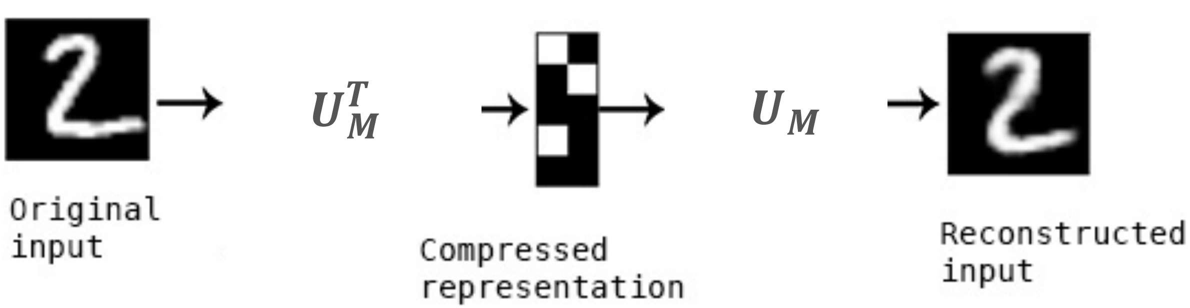
Mean shape + PC

Mean shape

Mean shape - PC



PCA



Latent variable: z

Latent variable generative model

z → Generator → 

Generated sample

Probabilistic PCA

Assumes that each latent variable is normally distributed:

$$z_n \sim \mathcal{N}(0, \mathbf{I})$$

Then a corresponding data point is generated via a projection:

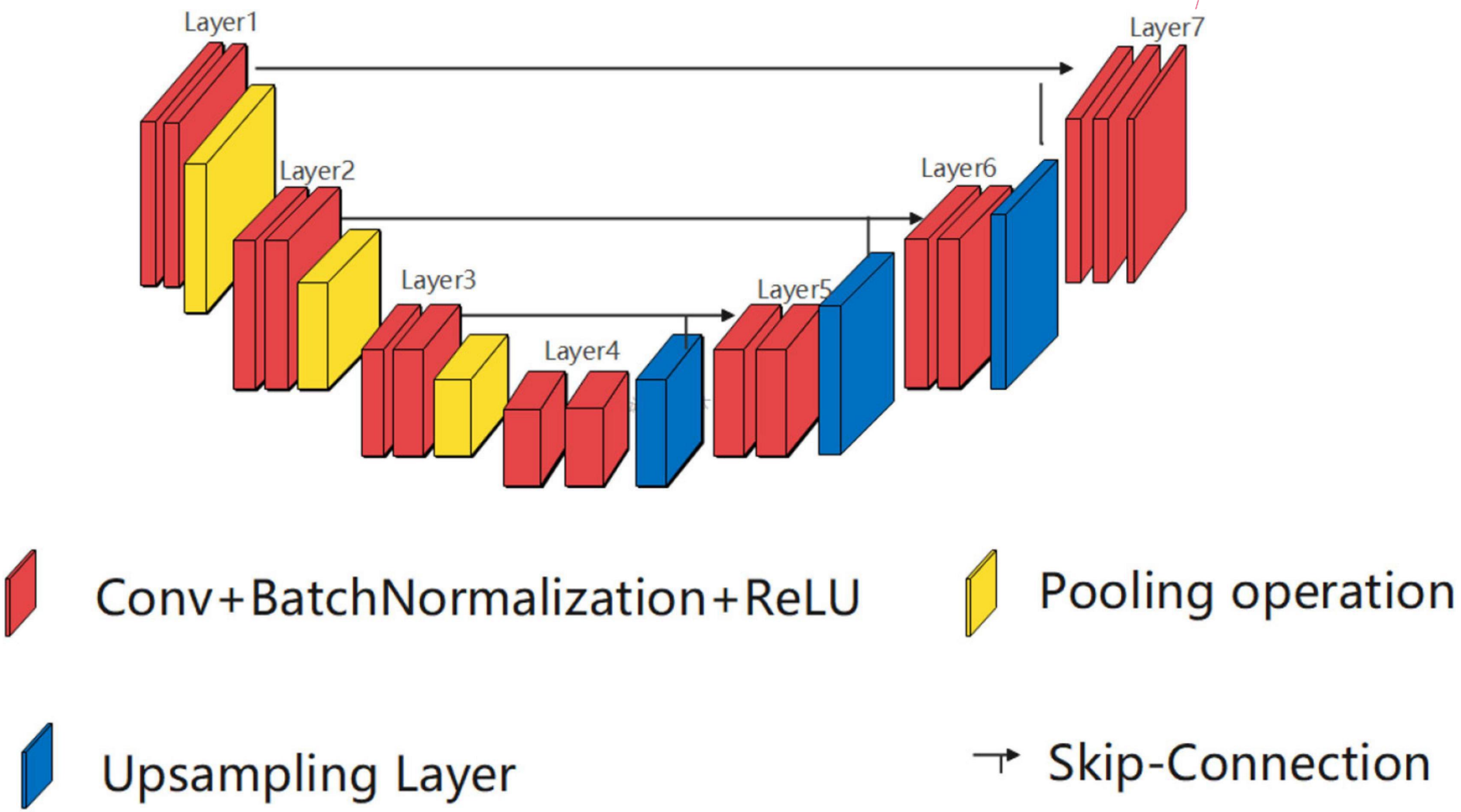
$$\mathrm{p}(x_n | z_n) \sim \mathcal{N}(\boldsymbol{U} z_n + b, \sigma^2 \mathbf{I})$$

IMAG/e

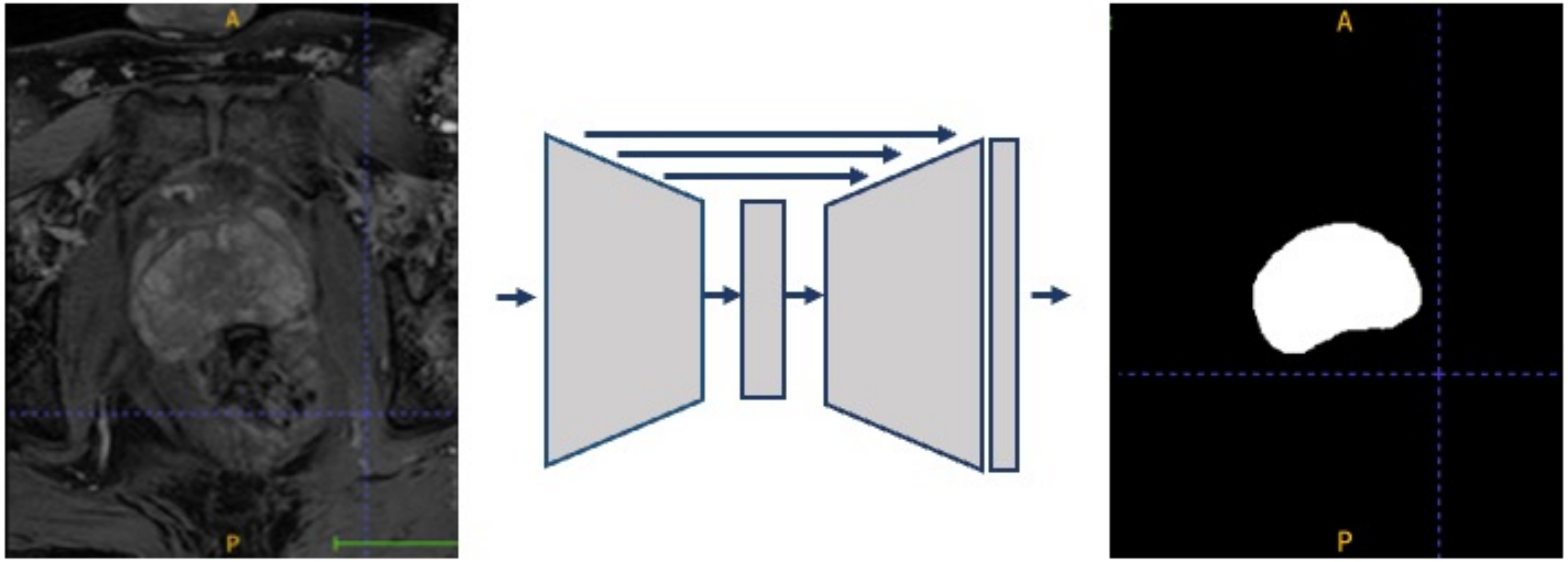
TU/e Technische Universiteit Eindhoven University of Technology

Deep Generative models

Recap: U-Net

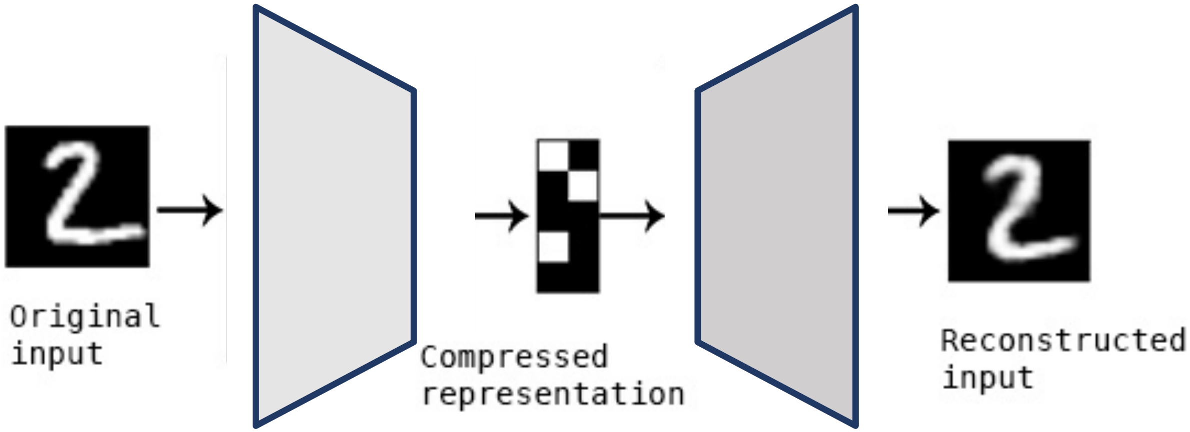

Layer1
Layer2
Layer3
Layer4
Layer5
Layer6
Layer7
Conv+BatchNormalization+ReLU
Pooling operation
Upsampling Layer
Skip-Connection

Recap: U-Net



Autoencoder (non-linear extension of PCA)

$f_\theta : x \mapsto z$ $\quad$ $g_\theta : z \mapsto \hat{x}$

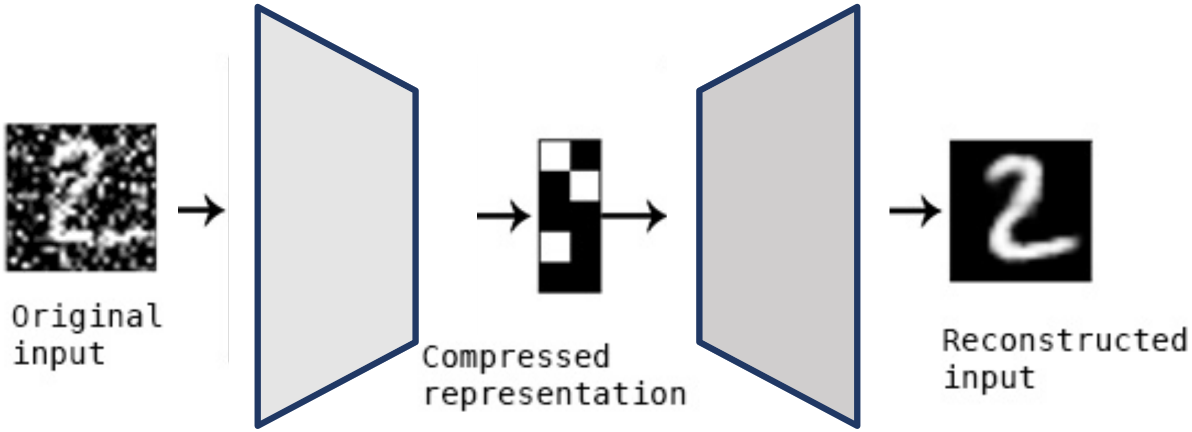


Loss function: $\|\hat{x} - x\|_2^2$

Latent variable: z

Denoising Autoencoder

$f_\theta : x + \epsilon \mapsto z$ $g_\theta : z \mapsto \hat{x}$



Loss function: $\|\hat{x} - x\|_2^2$

Latent variable: z

Applications:

- Representation learning
- Anomaly detection
- Normative learning
- Dimensionality reduction

"The information is in z"

…. But these are not generative

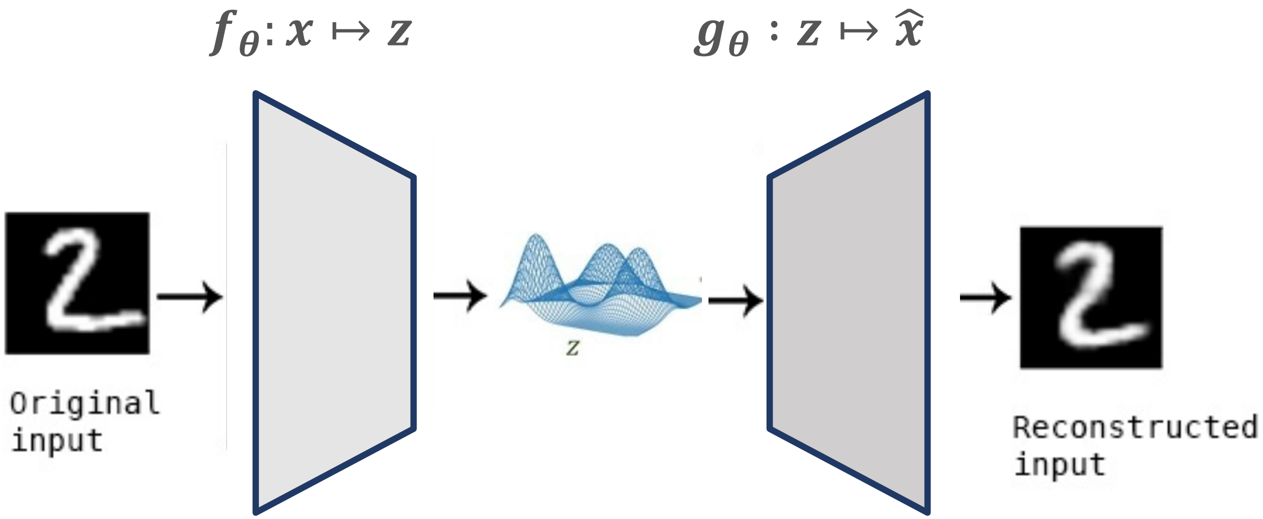
IMAG/e

TU/e Technische Universiteit Eindhoven University of Technology

Deep Generative models (Finally)

Where innovation starts

Variational Autoencoder - VAE

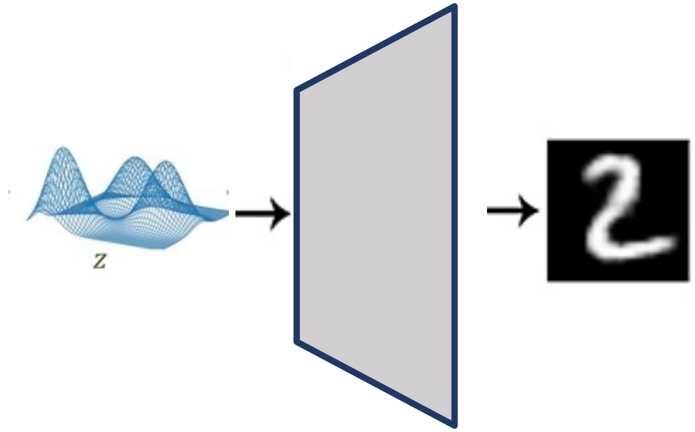


Latent variable: z

The loss is a combination of a difference between the distribution on z and an assumed prior distribution and a reconstrunction loss ($l1$ or $l2$).

Variational Autoencoder - VAE

$g_\theta : z \mapsto \hat{x}$

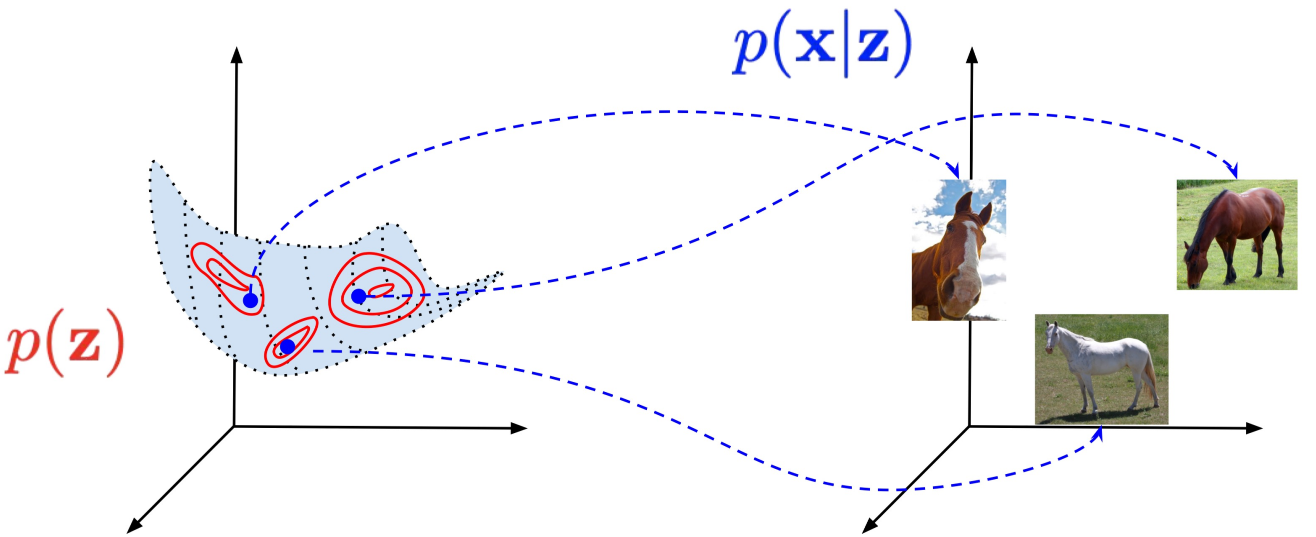


Latent variable: z

Generative latent variable models

- Introduce a probability distribution of the latent variables
- Sample $\mathrm{z} \sim p\,(\mathrm{z})$
- Generate $\mathrm{x} \sim \boxed{p\,(\mathrm{x}|\mathrm{z})}$

Generative process: Assume $p\,(\mathrm{x}|\mathrm{z})=\mathrm{f}(\mathrm{z})$
with f=neural network



How can we train such a model:

- The joint distribution describes the generative process:

$$p(\mathrm{x}, \mathrm{z}) = p(\mathrm{x}|\mathrm{z})\, p(\mathrm{z})$$

- But the latent variable is not available for training.

- Therefore, we marginalise it out:

$$p(\mathrm{x}) = \int_z p(\mathrm{x}|\mathrm{z})\, p(\mathrm{z})\, dz$$

$$= \mathbb{E}_{\mathrm{z} \sim p(\mathrm{z})} [p(\mathrm{x}|\mathrm{z})]$$

- Training goal is to maximise $p(\mathrm{x})$ for the dataset X by learning $p(\mathrm{z})$ and $p(\mathrm{x}|\mathrm{z})$ such that the latent variable z best captures the structure of the data

How can we train such a model:

$$p\,(\mathrm{x}) = \int_z p\,(\mathrm{x}|\mathrm{z})\; p\,(\mathrm{z})\; dz$$

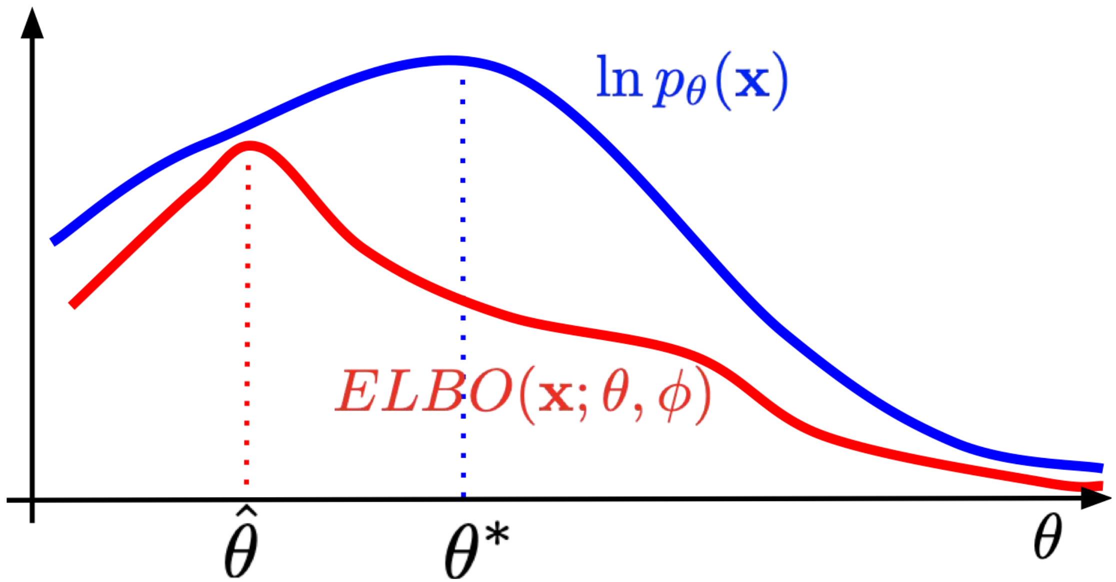
$$= \mathbb{E}_{\mathrm{z} \sim p\,(\mathrm{z})}\left[p\,(\mathrm{x}|\mathrm{z})\right]$$

- p (z) is the prior over latent space.
- p (x) is the marginal of the joint distribution (data likelihood).
- p (x|z) is the conditional likelihood

How can we train such a model:

- But for large datasets and multidimensional z, it is hard to approximate $p(x)$ (curse of dimensionality).
- We approximate the posterior $p(z|x)$ with a simpler distribution $q(z|x)$ (called a variational approximation).

$$
\begin{aligned}
log\, p(x) &= \mathbb{E}_{z \sim q(z|x)} log\left[\frac{p(x)\, p(z|x)}{p(z|x)}\right] \\
&= \mathbb{E}_{z \sim q(z|x)} log\left[\frac{p(x,z)}{q(z|x)}\right] + log\left[\frac{q(z|x)}{p(z|x)}\right] \\
&= \mathbb{E}_{z \sim q(z|x)} log\left[\frac{p(x,z)}{q(z|x)}\right] + KL(q(z|x), p(z|x)) \\
&\geq \mathbb{E}_{z \sim q(z|x)} log\left[\frac{p(x,z)}{q(z|x)}\right] \qquad \text{(ELBO)}
\end{aligned}
$$



$\ln p_\theta(\mathbf{x})$
$ELBO(\mathbf{x};\theta,\phi)$
$\hat{\theta}$
θ^*
θ

How can we train such a model:

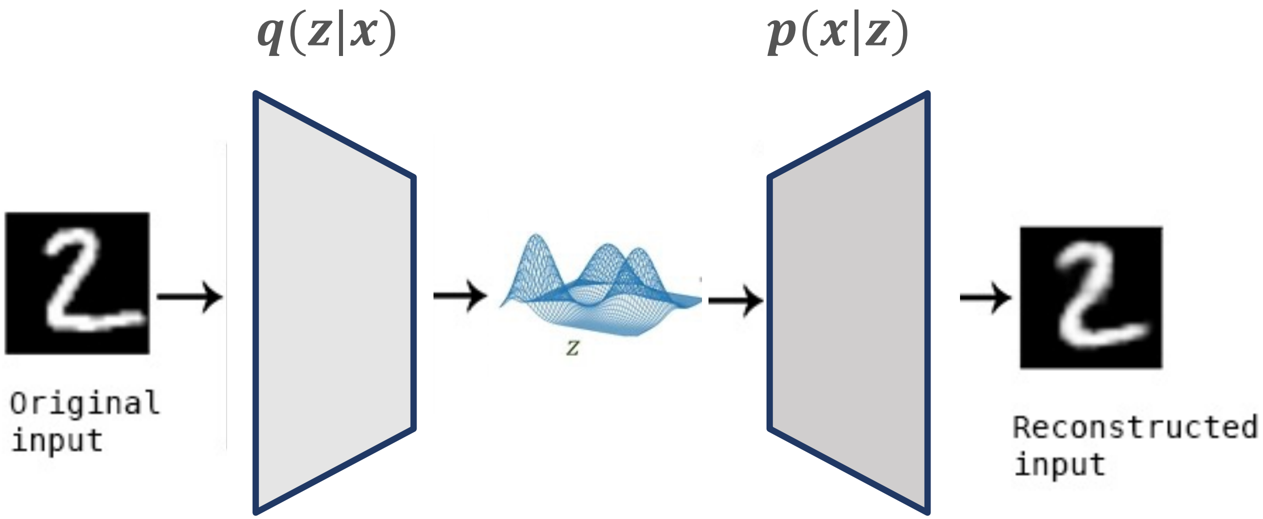
$$-log\, p\,(x) \;\leq \mathbb{E}_{z\sim\, q(z|x)}\, log\left[\frac{q(z|x)}{p(x,z)}\right]$$

$$= \mathbb{E}_{z\sim\, q(z|x)}\; log\left[\frac{q(z|x)}{p(x|z)p(z)}\right]$$

$$= KL(q(z|x), p(z)) + \mathbb{E}_{z\sim\, q(z|x)} - log[p(x|z)]$$

- Looks like an autoencoder: $\boldsymbol{q}: \boldsymbol{x} \mapsto \boldsymbol{z}$ and $\boldsymbol{p}: \boldsymbol{z} \mapsto \boldsymbol{x}$

Variational Autoencoder



Latent variable: z

Variational Autoencoder

- Minimise this bound to the negative log-likelihood assuming $q(z|x)$ is a Gaussian
- This is parameterised by a neural network – the encoder predicts parameters (mean, standard deviation) of the distribution.

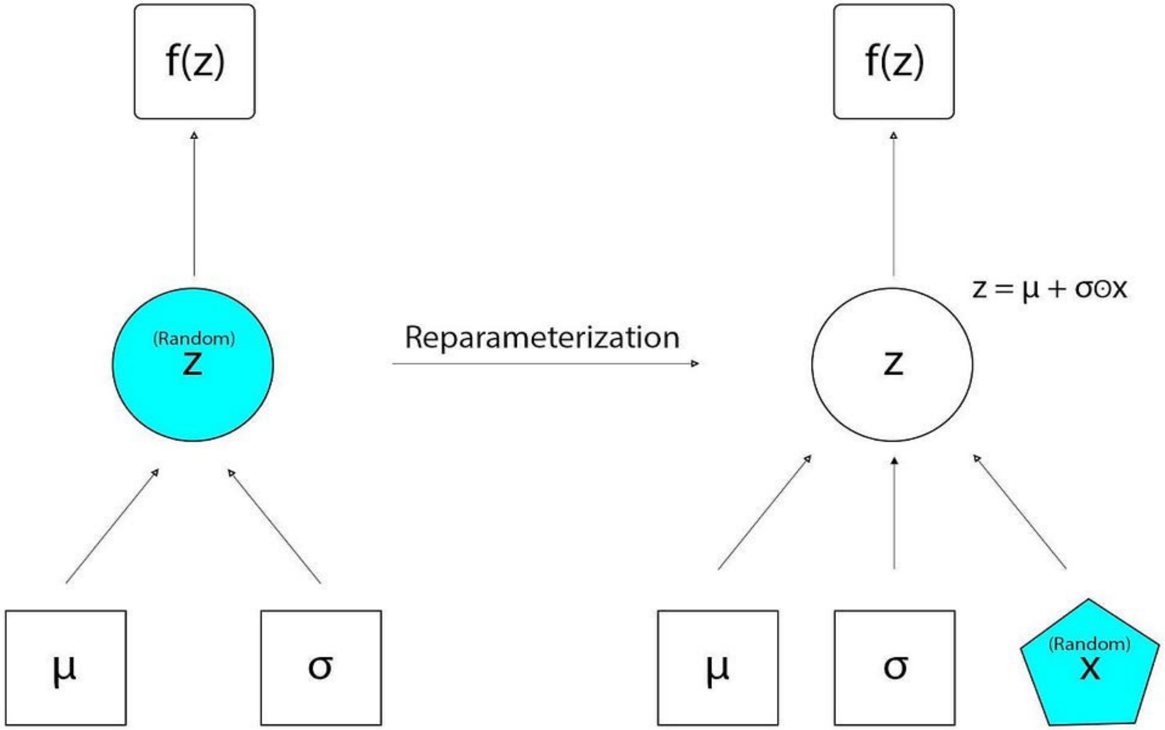
- Training finds the parameters of the neural network:

$$\theta^* = \operatorname{argmin}_\theta KL(q_\theta(z|x), p(z)) + \mathbb{E}_{z \sim q_\theta(Z|X)} - log[p_\theta(x|z)]$$

Aside: Reparameterisation trick:

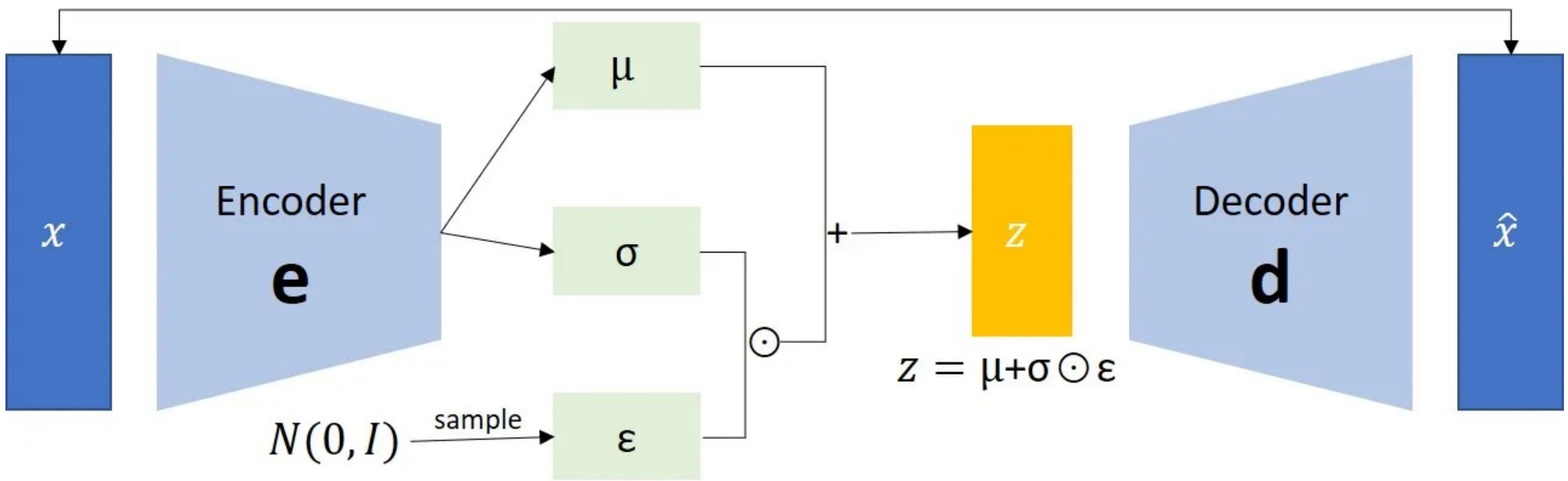
- Problem: need to backprop through random sampling which cause high variance of the gradients
- Solution: do not sample z directly.
- Instead of random sampling within the model, move the randomness outside the model and use it as input.
- Mathematically, instead of:
  - Push $\boldsymbol{x}$ through the encoder: $\boldsymbol{\mu_x}, \boldsymbol{\sigma_x} = \boldsymbol{q}(\boldsymbol{x})$
  - And sampling $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu_x}, \boldsymbol{\sigma_x^2})$
- Do:
  - Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
  - Push $\boldsymbol{x}$ through the encoder: $\boldsymbol{\mu_x}, \boldsymbol{\sigma_x} = \boldsymbol{q}(\boldsymbol{x})$
  - $\boldsymbol{z} = \boldsymbol{\epsilon\sigma_x} + \boldsymbol{\mu_x} \sim \mathcal{N}(\boldsymbol{\mu_x}, \boldsymbol{\sigma_x^2})$

Reparameterisation trick:



VAE – training procedure:

Minimize 1: $(x - \hat{x})^2$

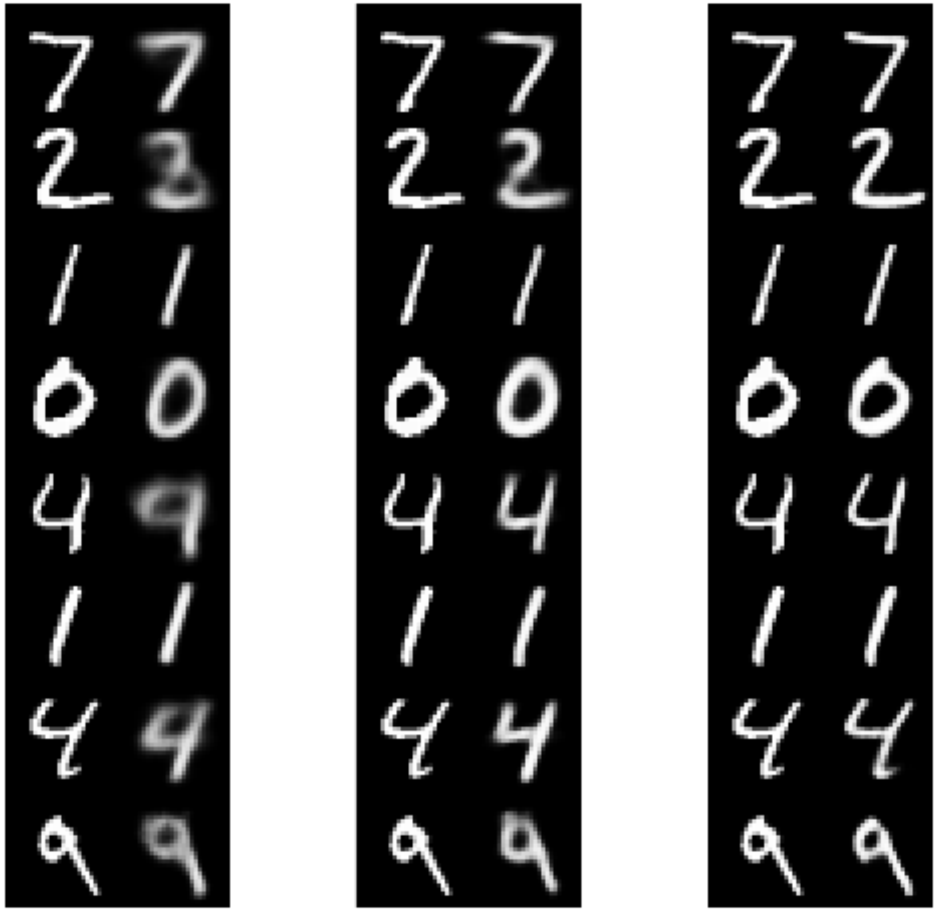


Minimize 2: $\frac{1}{2}\sum_{i=1}^{N}(\exp(\sigma_i) - (1+\sigma_i) + {\mu_i}^2)$

Assuming $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$

MNIST example:

- Reconstructions:



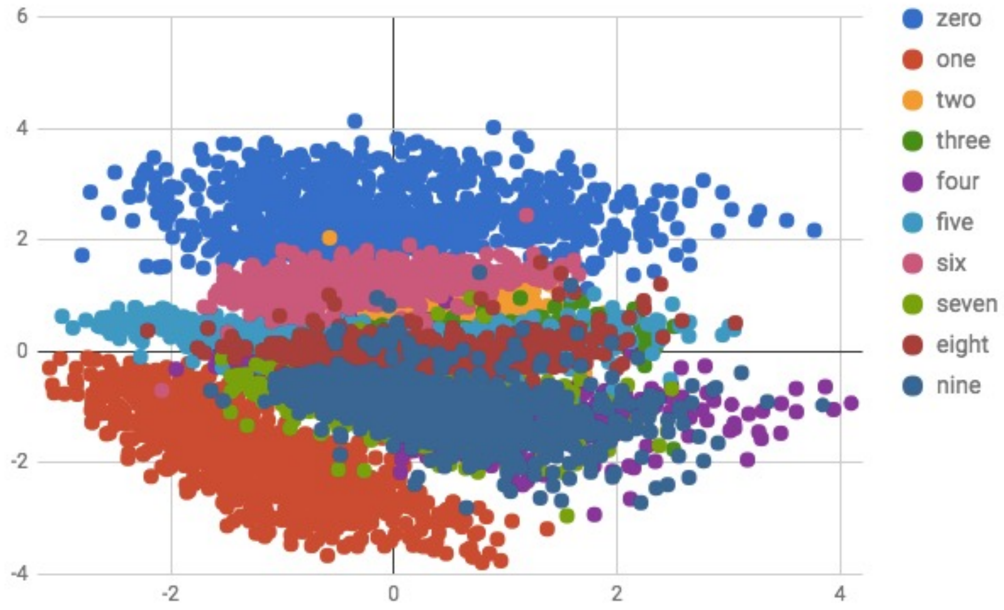
2-D 5-D 20-D

Limitations:

- VAEs may not generate samples that are as high-quality or as realistic as those generated by other models (next lecture), especially for images.
- VAEs may struggle to generate samples from high-dimensional distributions or distributions with complex structures.
- Classes may overlap in latent space. (Latent space is entangled).
- Some points in latent space may generate samples that do not make sense (e.g. a combination of the shapes for two different numbers in the MNIST example)
- VAEs can be sensitive to hyperparameter choice and may require careful tuning to achieve good results.

MNIST example:

- Latent space:



$\beta - \mathbf{VAE}$

- Loss $= \mathbb{E}_{z \sim q(z|x)} - log[p(x|z)] + \beta\, KL(q(z|x), p(z))$
- $\beta > 1$ gives stronger constraints on latent representation.

Disentanglement

- A representation is disentangled if each variable in the inferred latent representation z is only sensitive to one single generative factor and relatively invariant to other factors.
- For example, a model trained on photos of human faces might capture the gentle, skin colour, hair colour, hair length, whether wearing a pair of glasses and many other relatively independent factors in separate dimensions. Such a disentangled representation is very beneficial to facial image generation.

$\beta - \mathbf{VAE}$

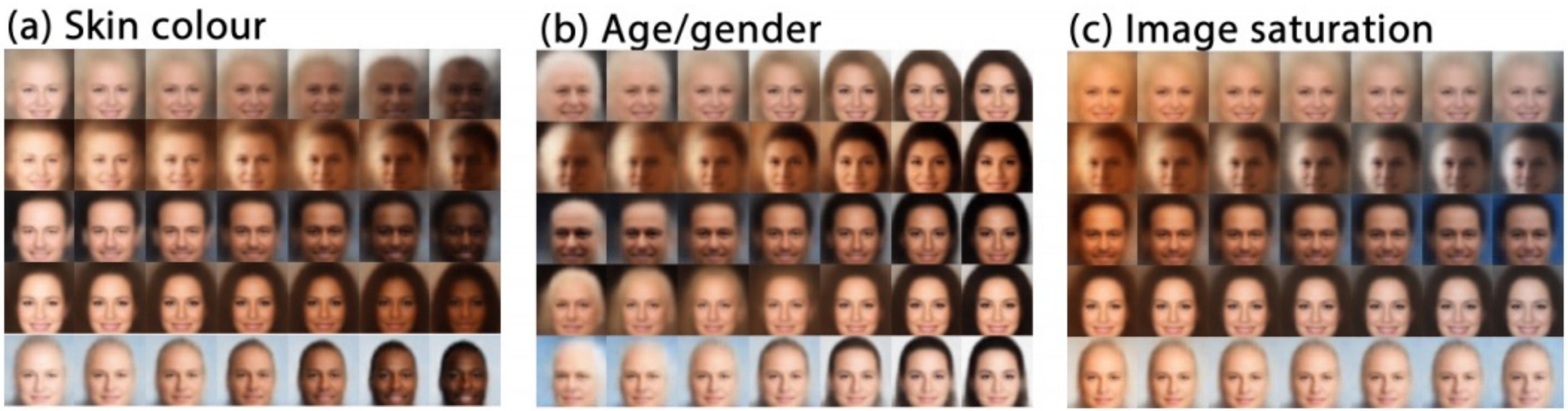


Figure 4: **Latent factors learnt by β-VAE on celebA:** traversal of individual latents demonstrates that β-VAE discovered in an unsupervised manner factors that encode skin colour, transition from an elderly male to younger female, and image saturation.

Generative Models part II

Cian M. Scannell, PhD
Assistant Professor
Medical Image Analysis
c.m.scannell@tue.nl

Topics:
- Addressing VAE limitations
- GANs
- Conditional generation
- Image-to-image translation
- Limitations and extensions to GANs
- Fréchet inception distance
- Diffusion models

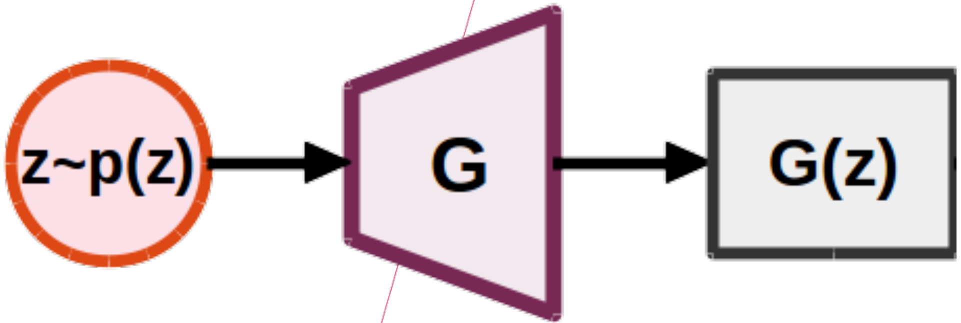
Intended learning outcomes

The student can:

- Discuss the intuition behind generative adversarial networks (GANs)
- Understand the benefits of GANs in the context of the limitations of VAEs
- Formulate the GAN training process and loss function
- Classify applications of GANs
- Give examples of issues with GANs
- Motivate solutions to these issues
- Relate diffusion models to VAEs

Recap VAEs

- A VAE is a latent variable model
  - Sample $z \sim p(z)$
  - Generate $x \sim p(x|z)$



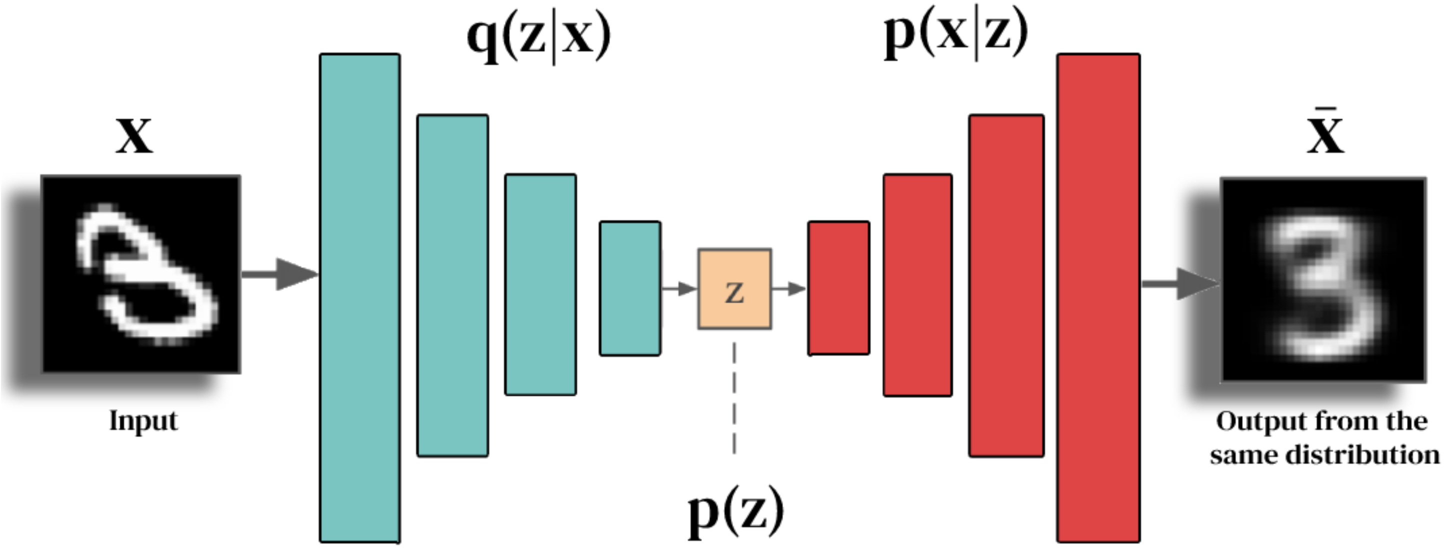
- A VAE is an *explicit* likelihood model
  - It tries to explicitly compute (approximate) the density $p(x)$
  - But we need to do a lot of work to get around the lack of an analytical solution and the curse of dimensionality

Limitation:

- Intractable likelihood - > Maximise ELBO rather than likelihood directly

Limitations:

- VAEs may not generate samples that are as high-quality or realistic, especially for images.



- Classes may overlap in latent space. (Latent space is entangled).
- Some points in latent space may generate samples that do not make sense (e.g. a combination of the shapes for two different numbers in the MNIST example)

Generative adversarial networks (GANs)

Previously:

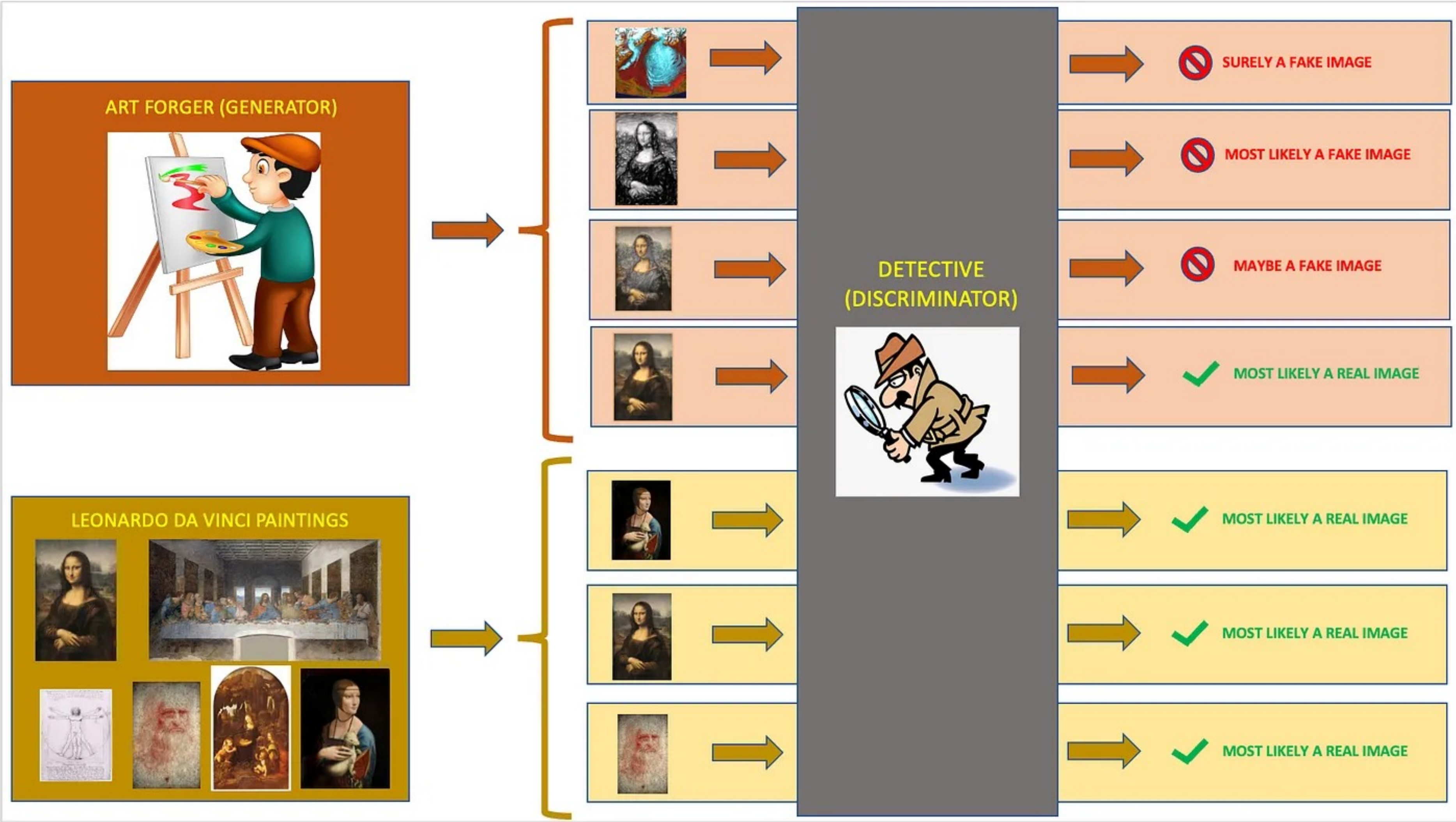
- We used the log-likelihood to describe the differences between generative samples and the training distribution during training.
- As well as the practical difficulties, it is unclear if better likelihood numbers correspond to better samples.

Generative adversarial networks (GANs)

Now:

- Is there another way?
- I.e. instead of using the log-likelihood, is there another way to calculated the differences between real and generate data?
- Can we learn it?

GAN intuition:

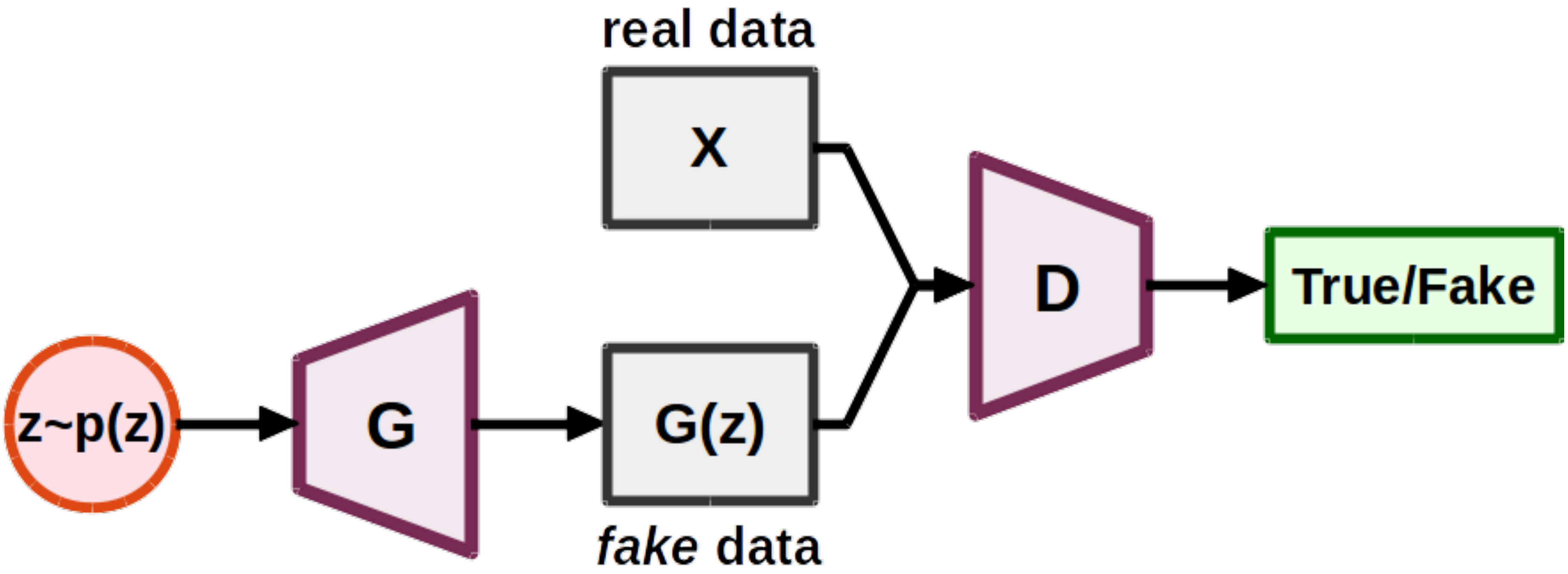


Generative adversarial networks (GANs)

Main idea:

- Train a classifier to distinguish samples as either real or generated
- Train the generator to create realistic samples that fool the classifier

GAN:



Algorithm:

- Generator - $G_\beta : Z \ \to X$.
- Discriminator - $D_\alpha : X \ \to [0, 1]$.
- Discriminator tries to maximize $\log D_\alpha(x) \ + \log 1 - D_\alpha\big(G_\beta(z)\big)$ with respect to α.
- Generator tries to minimise it with respect to β.

$$\min_\beta \max_\alpha \mathbb{E}_{x \sim p_{real}} [\log D_\alpha(x)] + \mathbb{E}_{z \sim p(z)} [\log 1 - D_\alpha\big(G_\beta(z)\big)\ ]$$

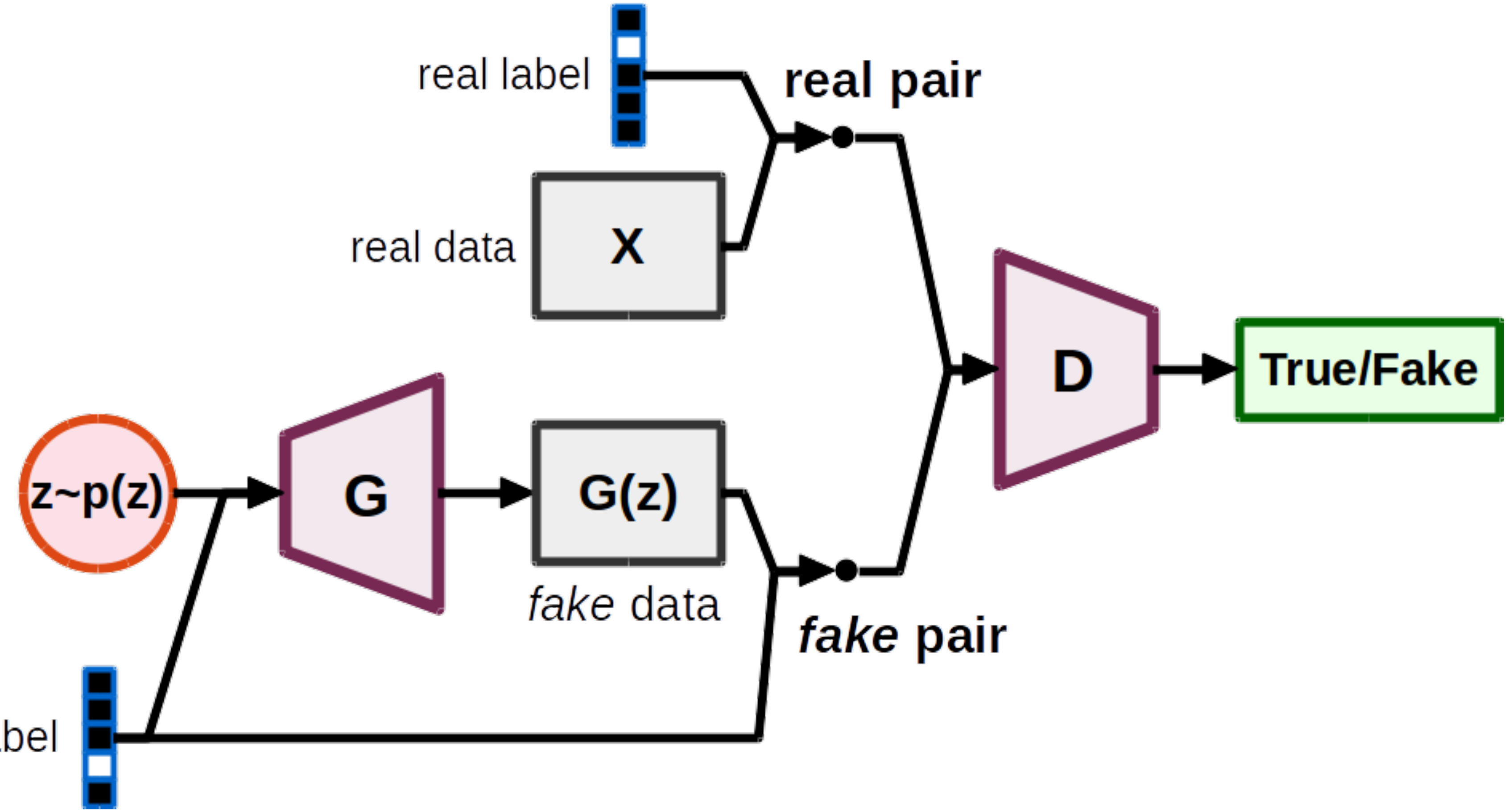
- This is known as the adversarial (or GAN) loss as there are two parts trying to achieve two opposite goals.

Challenges:

- **Unstable** optimization procedure
  - Generator and discriminator loss often oscillate without converging.
- Potential for **mode collapse**
  - Generator beats discriminator but only predicts same (or limited set of) image(s) every time
- Depends heavily on choice of hyperparameters
- Difficult to evaluate
  - When is an image realistic?

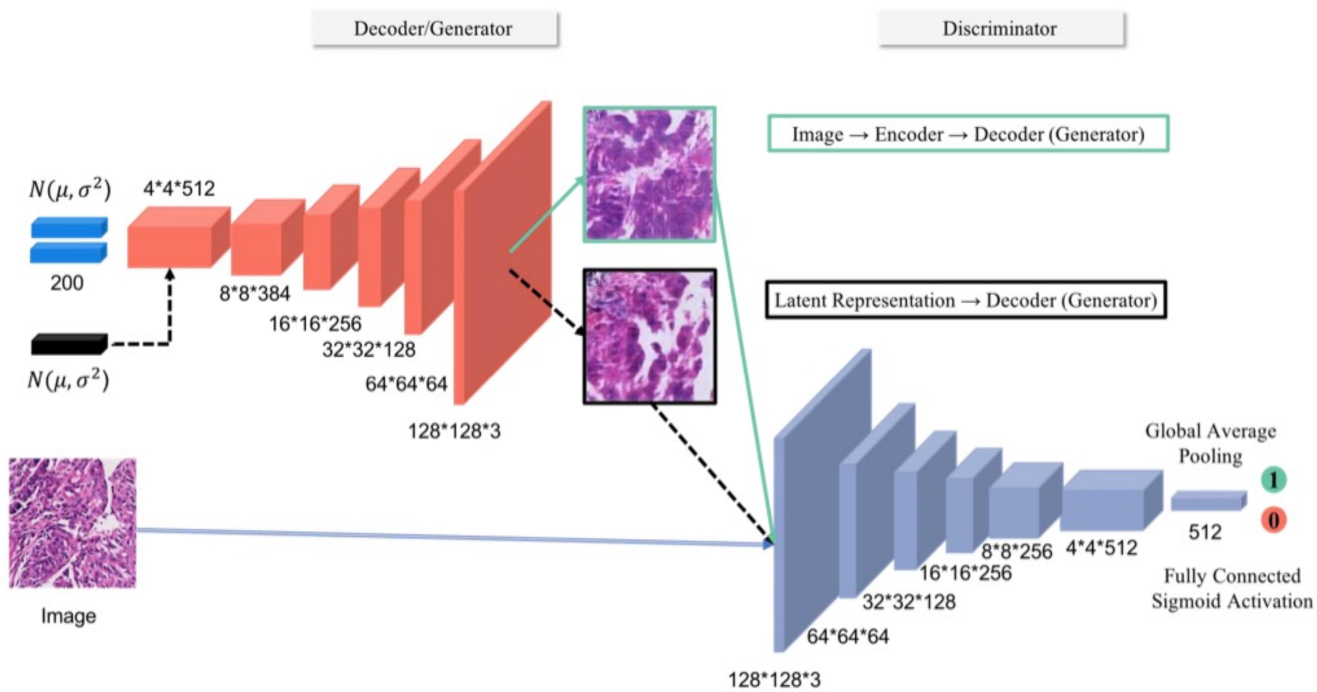
Conditional generation

Conditional GAN (cGAN):

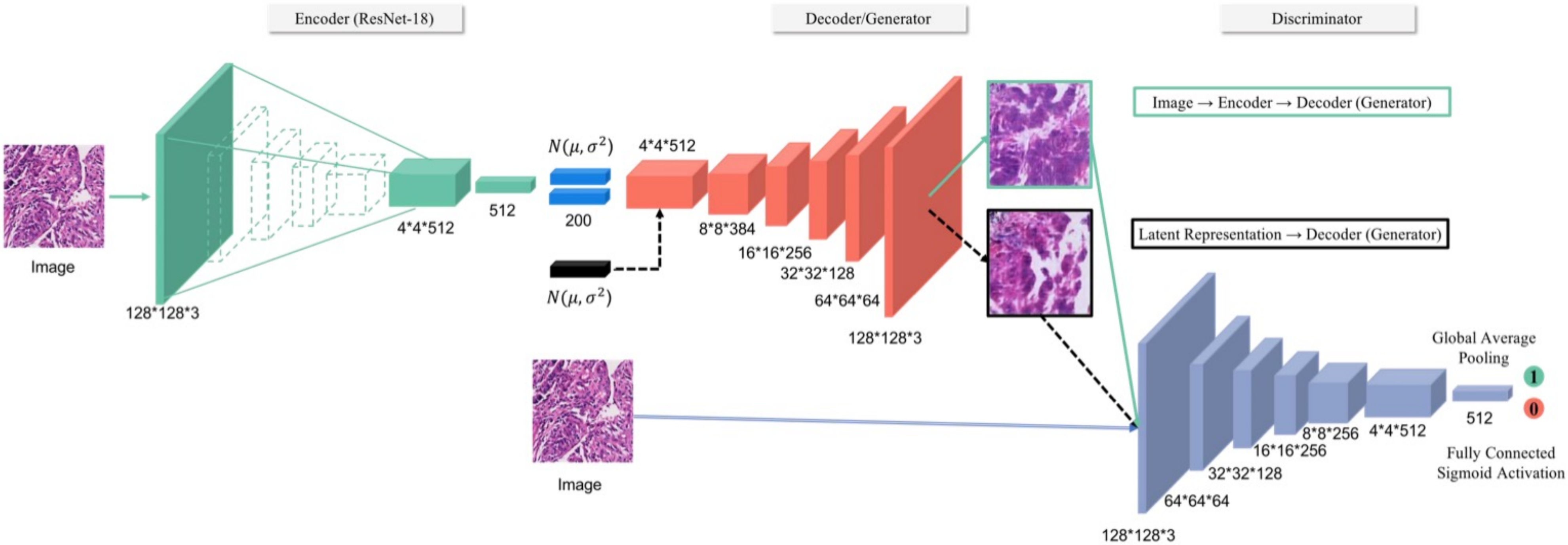


$D_\alpha(x|y)$ and $G_\beta(z|y)$ instead of $D_\alpha(x)$ and $G_\beta(z)$.

Hybrid models – VAE-GAN:



Hybrid models – VAE-GAN:



Discussion points

1. What is a latent variable model?

Discussion points

2. What will the latent variables be?

Discussion points

3. How do we enforce this?

Learning objectives

The student can:

- Motivate the uses of generative machine learning models
- Compare the benefits of generative and discriminative models
- Understand the term latent variable model
- Recall the PCA algorithm
- Relate autoencoder models to PCA
- Extend the idea of autoencoders to variational autoencoders
- Describe the VAE loss function
- Explain its derivation
- Remember the reparameterization trick of VAEs
- Identify the limitations of VAEs

Learning objectives

The student can:

- Discuss the intuition behind generative adversarial networks (GANs)
- Formulate the GAN training process and loss function
- Classify applications of GANs
- Give examples of issues with GANs
- Motivate solutions to these issues
- Understand the benefits of GANs in the context of the limitations of VAEs

IMAG/e

TU/e Technische Universiteit Eindhoven University of Technology

Questions?