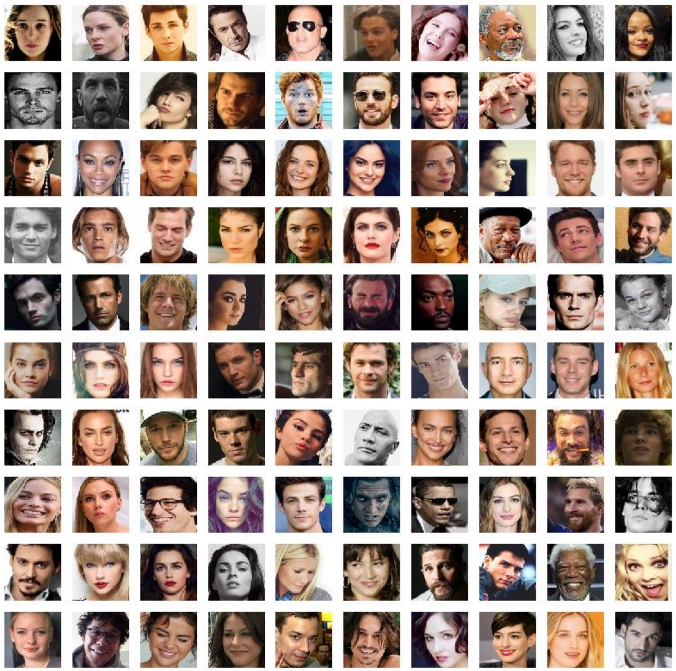
Image Processing and Analysis

Generative models
Variational Auto-Encoder

olivier.bernard@insa-lyon.fr

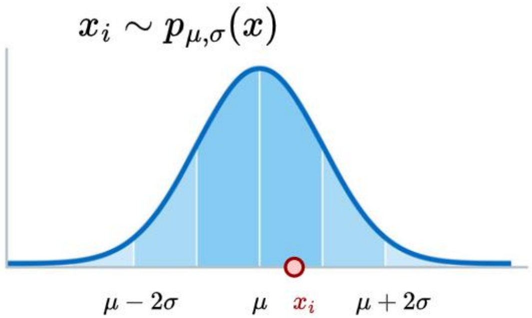
What is the purpose of generative models?

- How to generate synthetic faces?



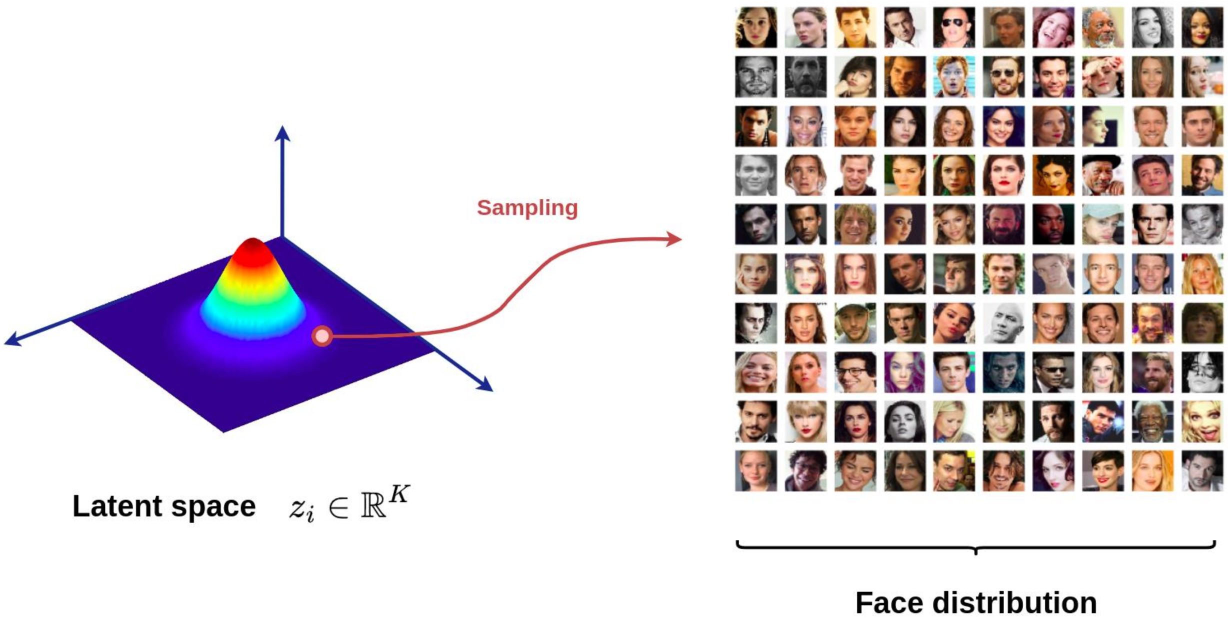
By modeling the corresponding distribution $x \sim p_\theta(x)$!

➔ Are classical distributions still relevant ?



- How to model complex distributions?

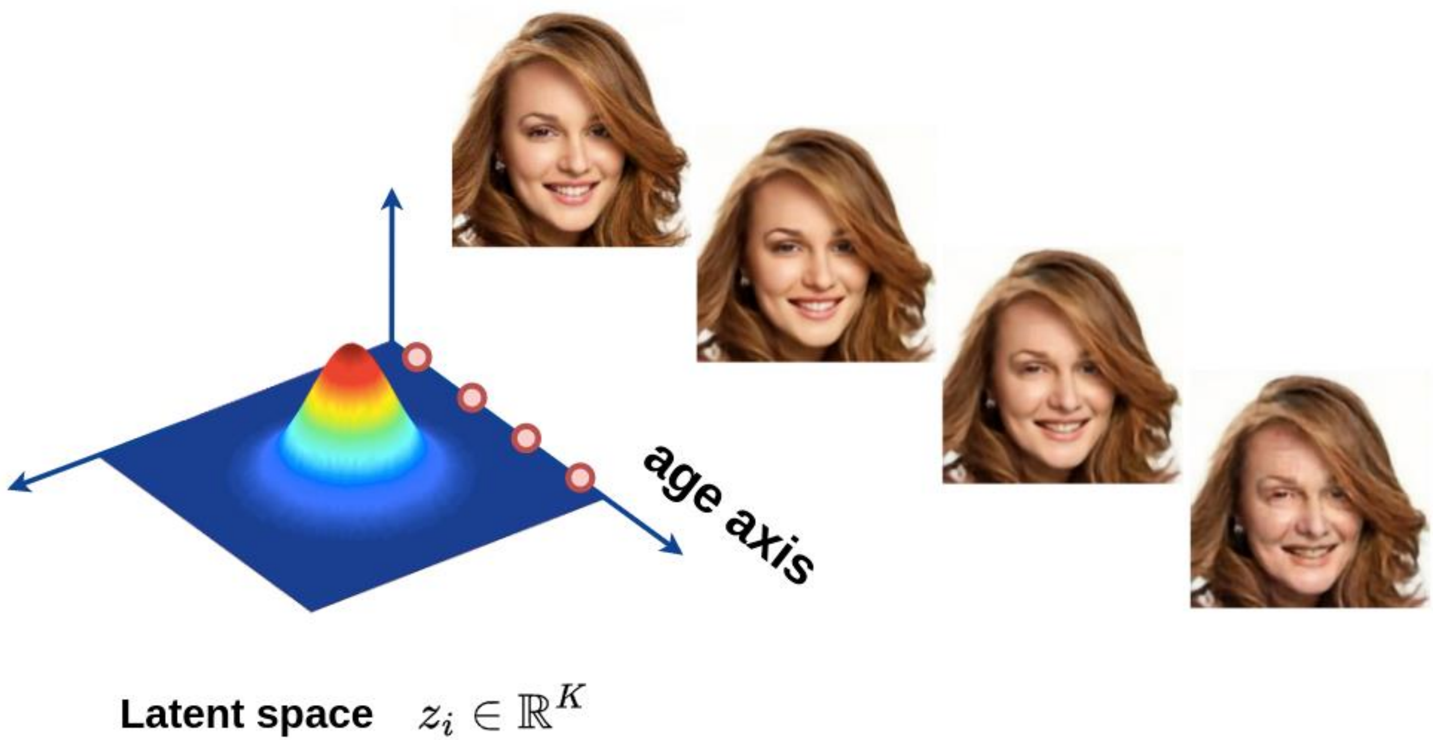
The image space (modeled by $p(x)$) is projected into a more efficient hidden representation space, called the latent space (modeled by $p(z)$)



What is the purpose of generative models?

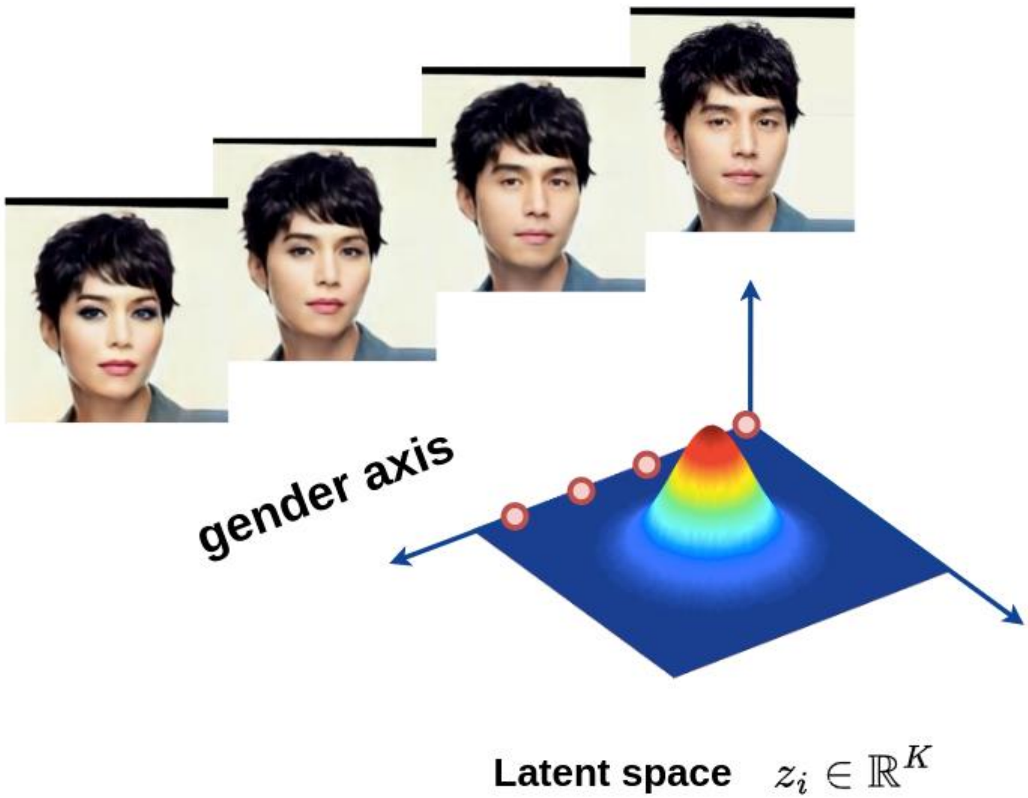
- For what purpose?

An obsession: mastering the latent space!



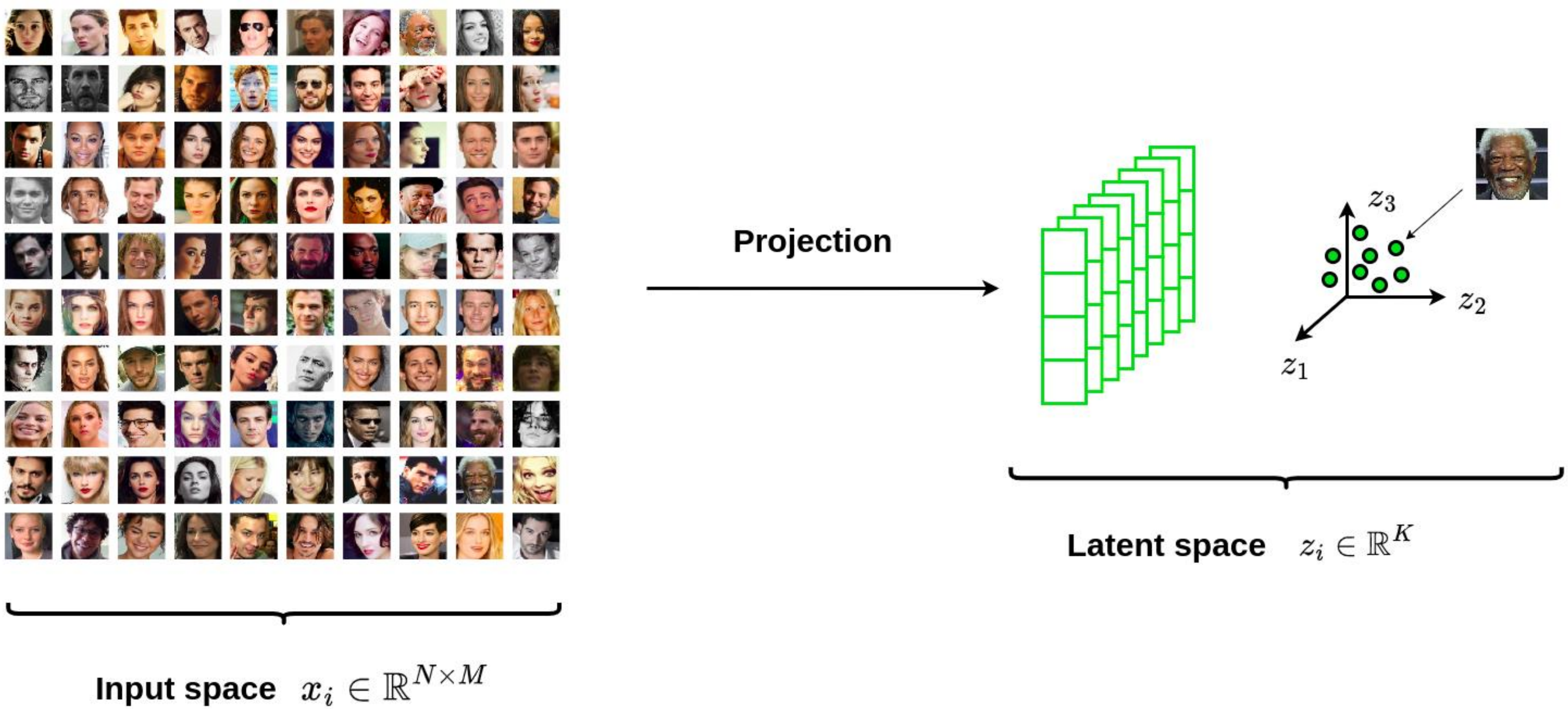
- For what purpose?

An obsession: mastering the latent space!

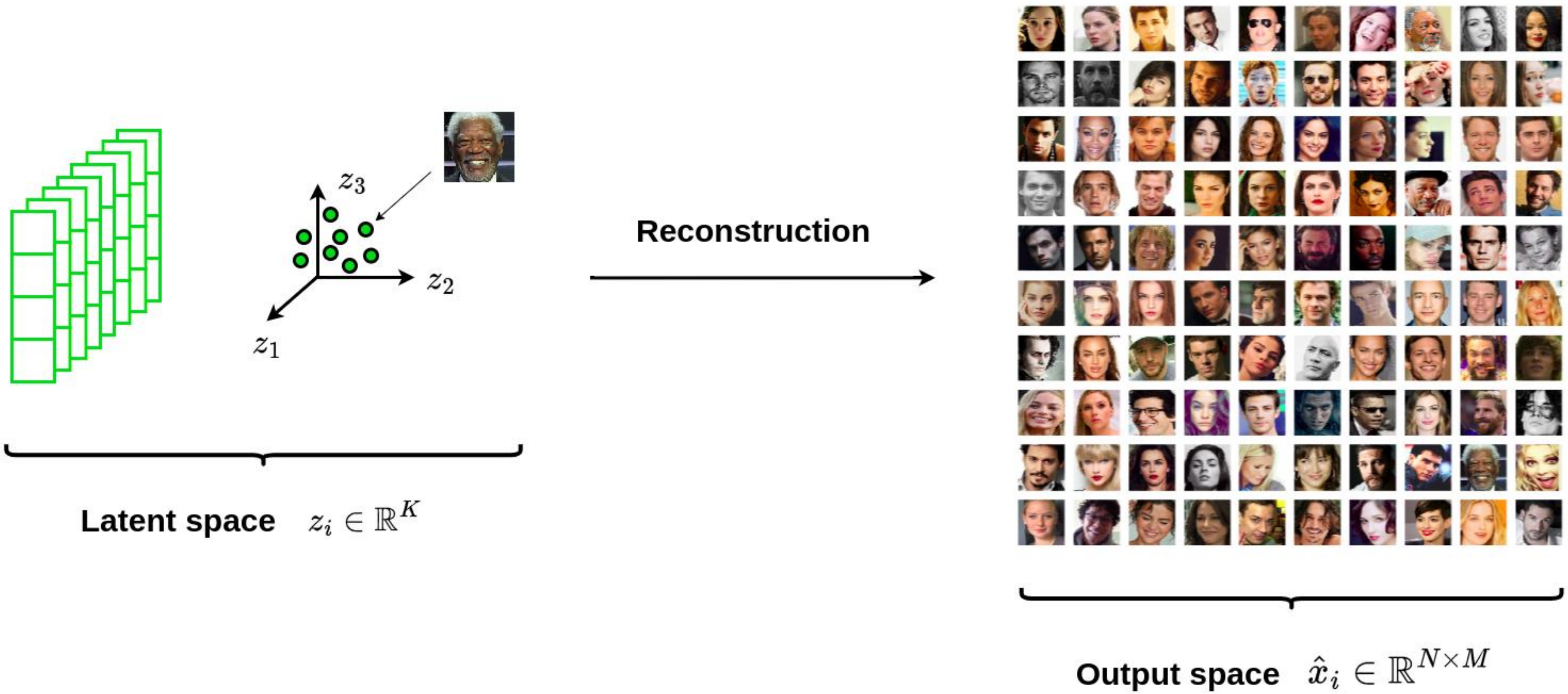


Auto-encoders

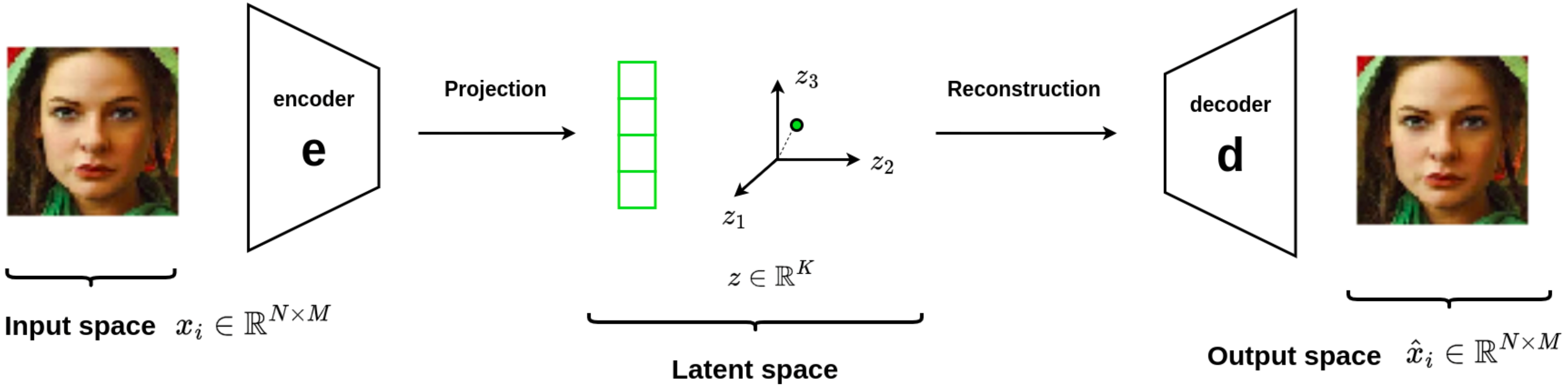
- Projection into a more efficient and lower-dimensional representation space



- How to have a relevant representation space?



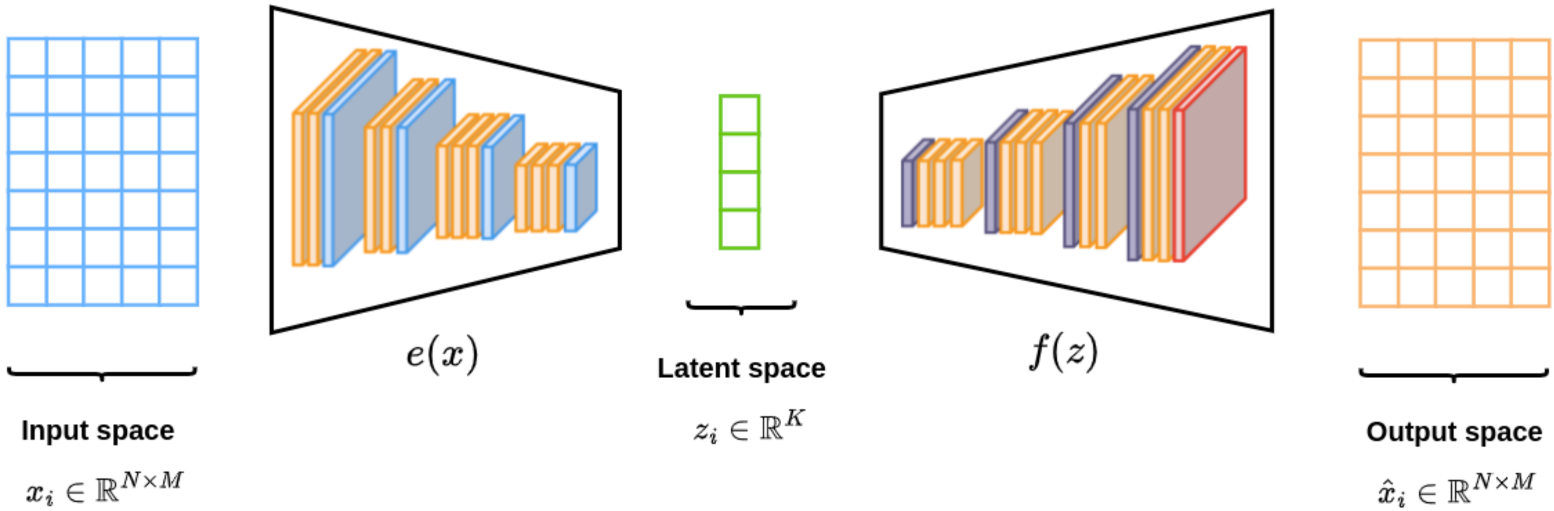
- Standard architectures



- Example of a cost function

$$\text{loss} = \|x - \hat{x}\|^2$$

- Encoder / Decoder modeled by neural networks (convolutional)

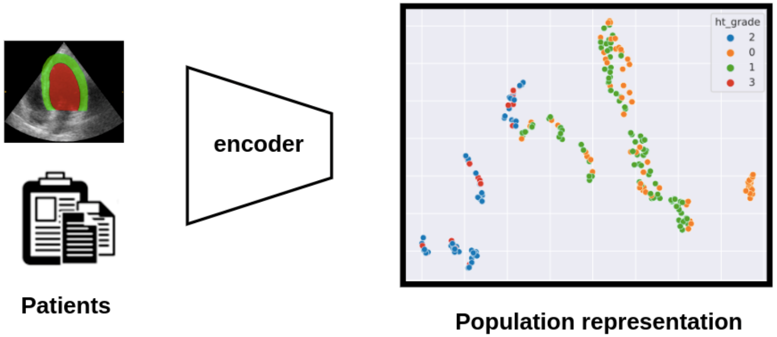


$$\text{loss} = \|x - f(e(x))\|^2$$

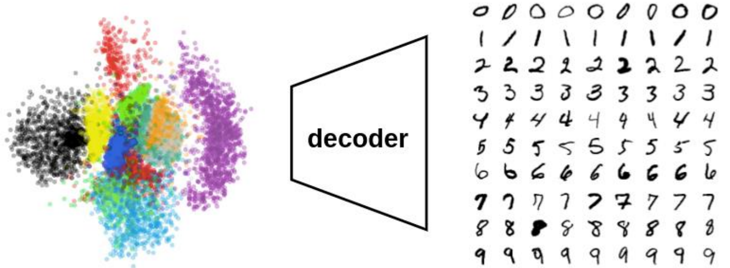
Interest of autoencoders

- Autoencoder? What for?

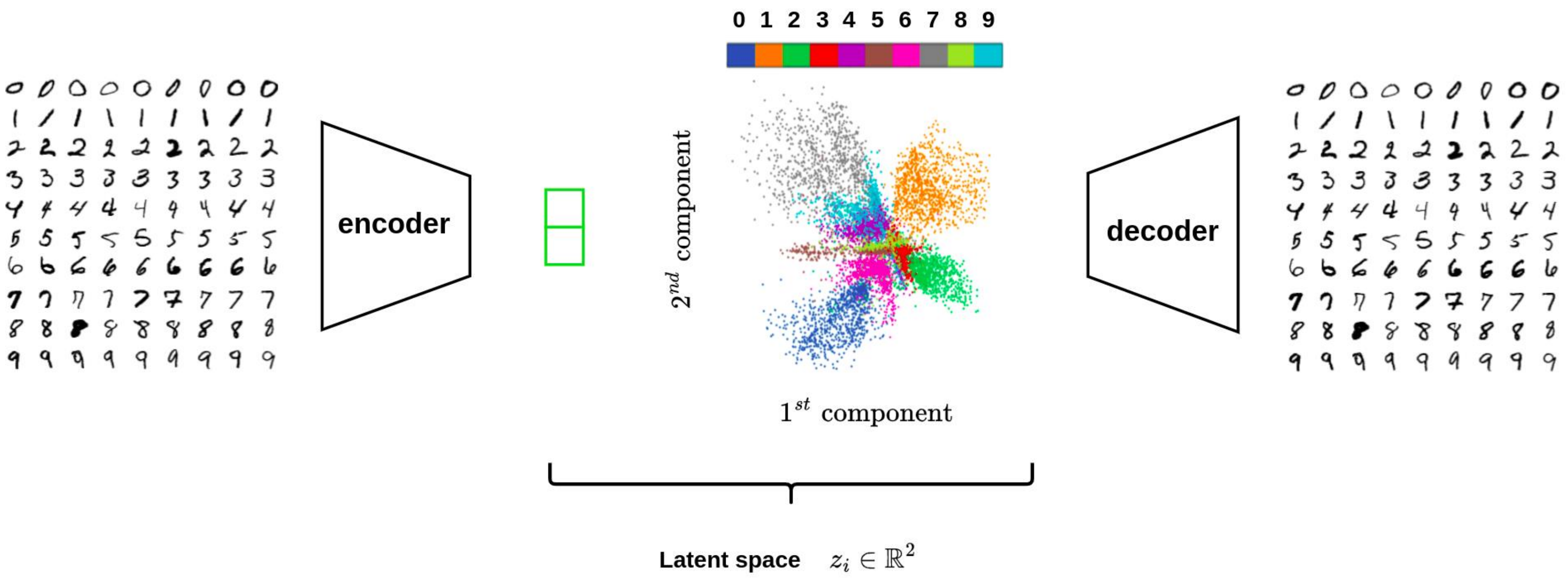
➔ Data representation



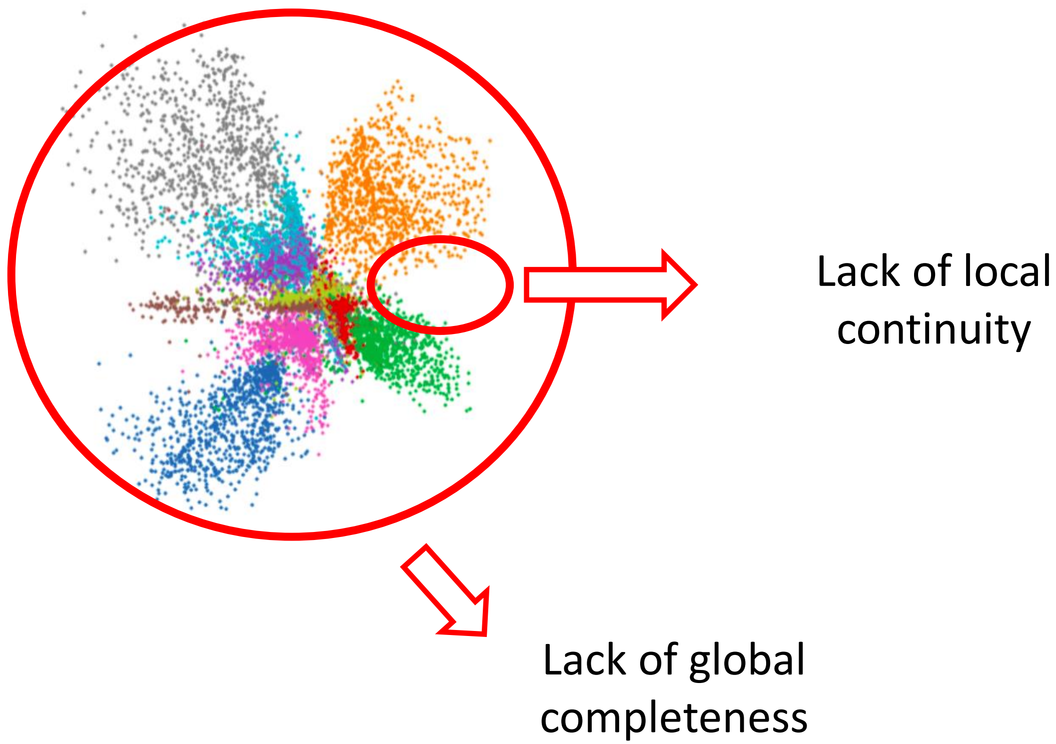
➔ Generative model



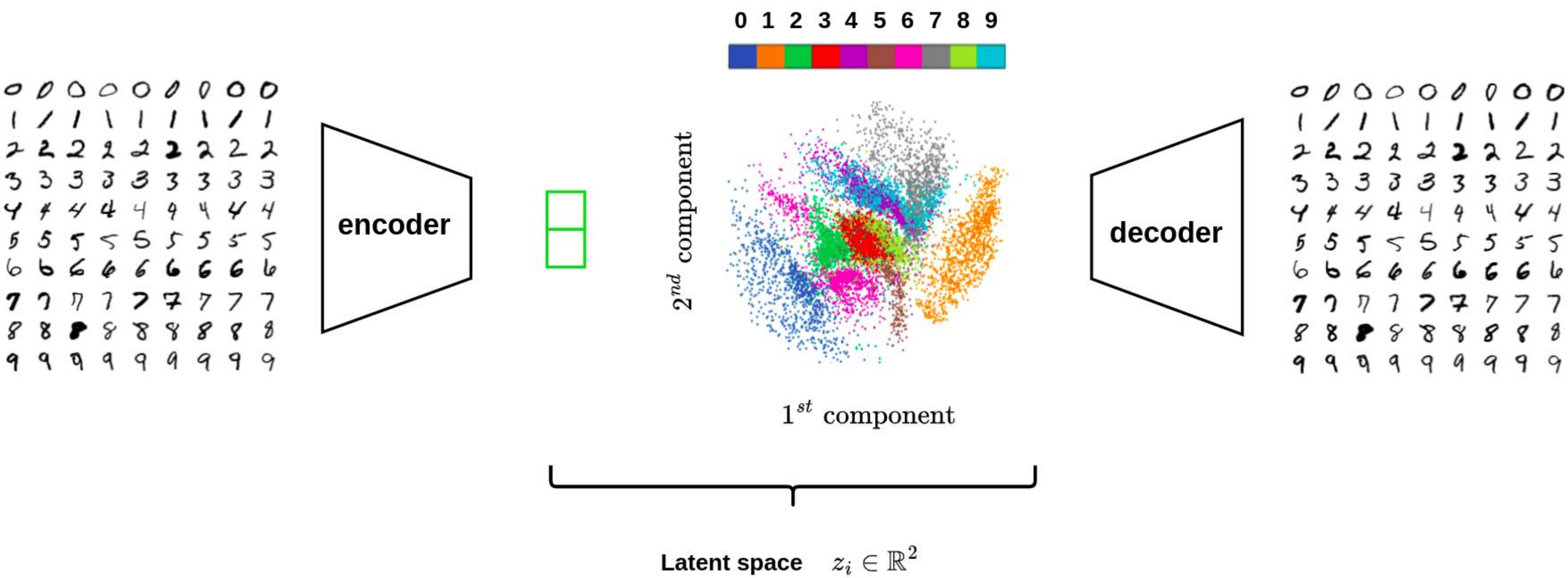
- Generative models



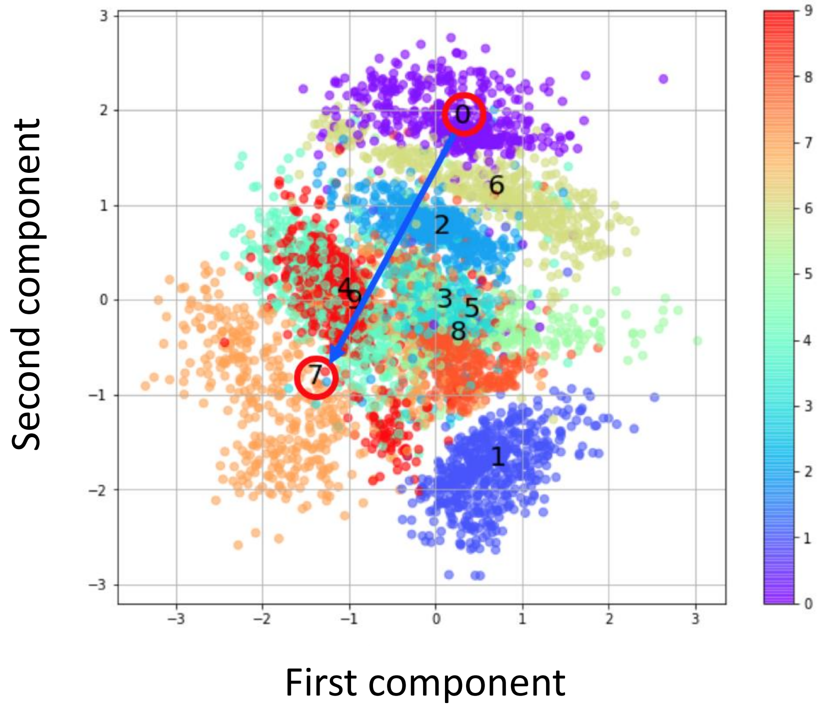
- Need to better control the structure of the latent space



- Generative model with better properties thanks to the *variational framework*

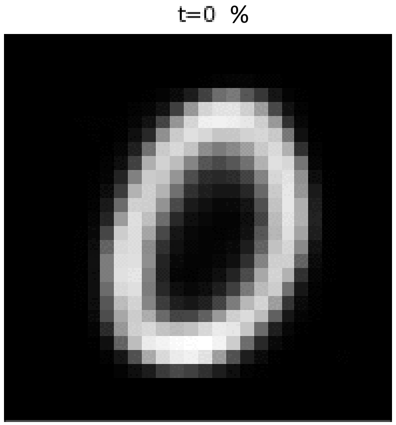


- Generative model with variational framework



Linear interpolation in the latent space

$$t \cdot z_0 + (1-t) \cdot z_7, \qquad 0 \leq t \leq 1$$



Variational Auto-Encoder

The entire mathematics are described in the following blog

https://creatis-myriad.github.io/tutorials/2022-09-12-tutorial-vae.html

Key concepts

- Reinforcement of a structured latent space

  - Through a probabilistic framework

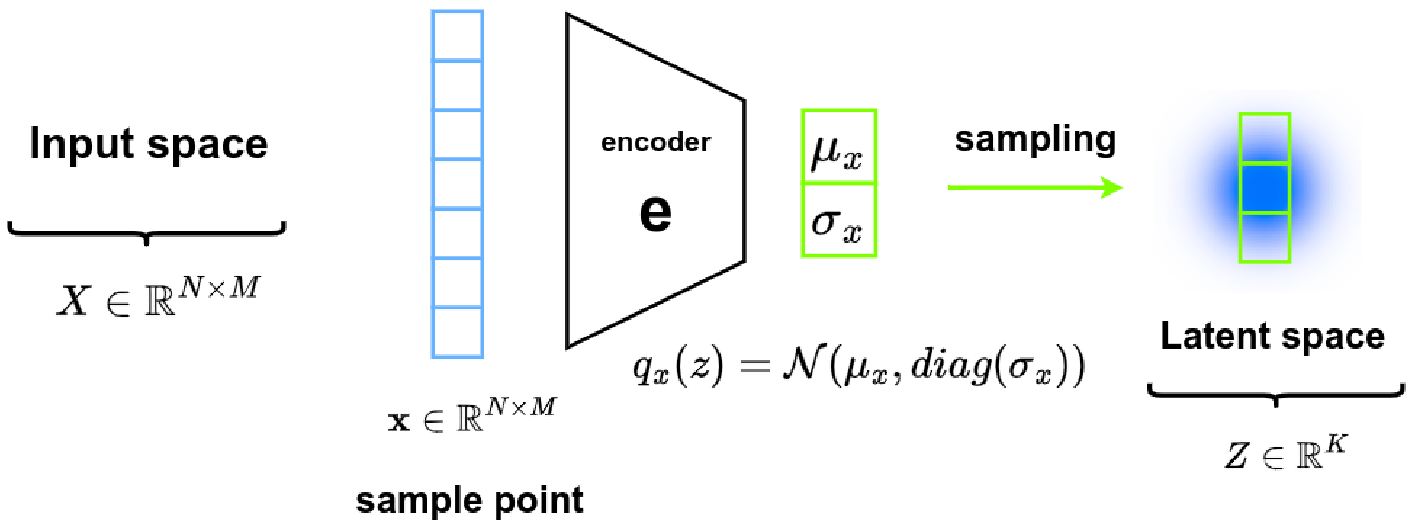
  - By imposing continuity constraints

  - By imposing completeness constraints

- Probabilistic framework: *continuity*

  → Introduction of local regularizations of the latent space

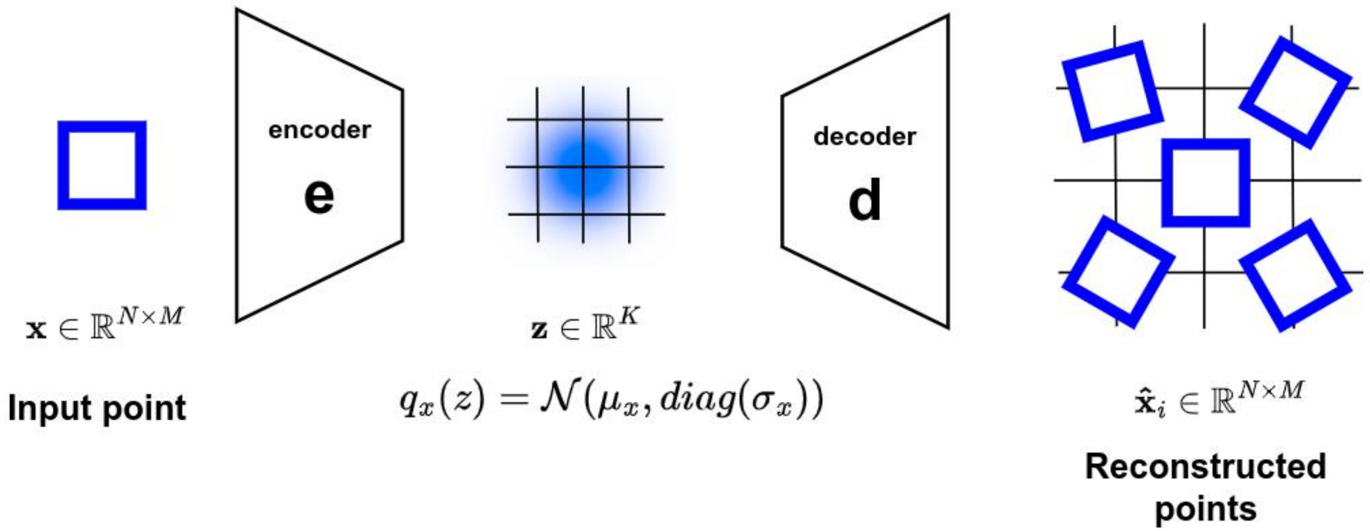
  → Each input data x is encoded as a Gaussian distribution $q_x(z) = \mathrm{N}(\mu_x, diag(\sigma_x))$



Key concepts

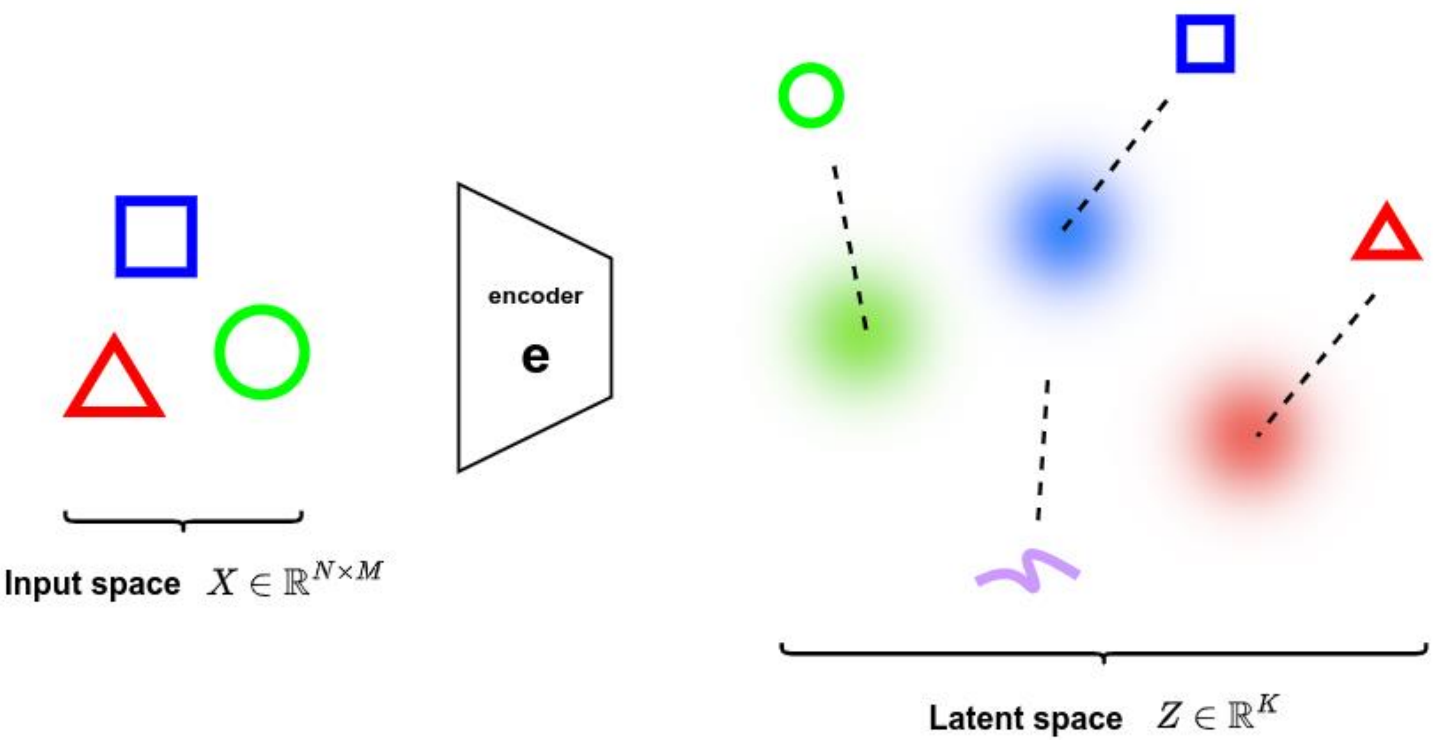
- Probabilistic framework: *continuity*

  → Sampling from a local region of the latent space produces similar results



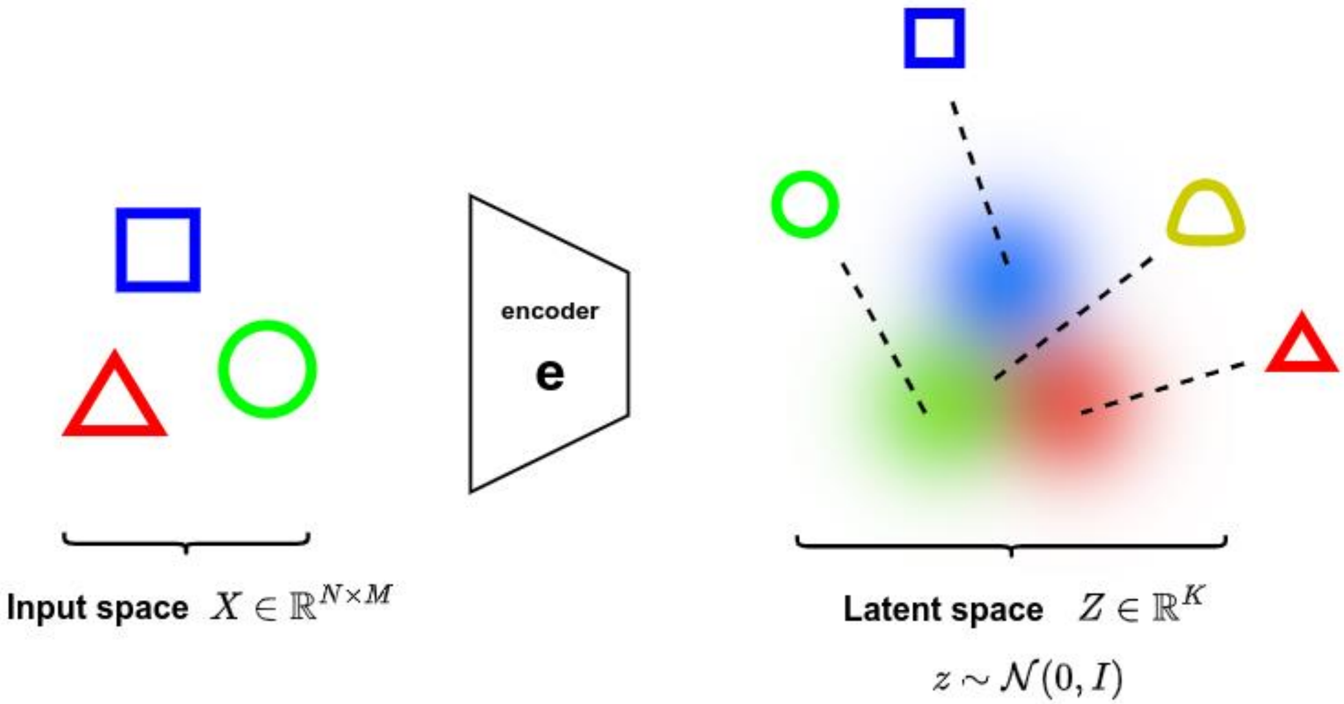
- Probabilistic framework: *completeness*

  → Encourage that every reconstructed point in the latent space produces consistent results



- Probabilistic framework: *completeness*

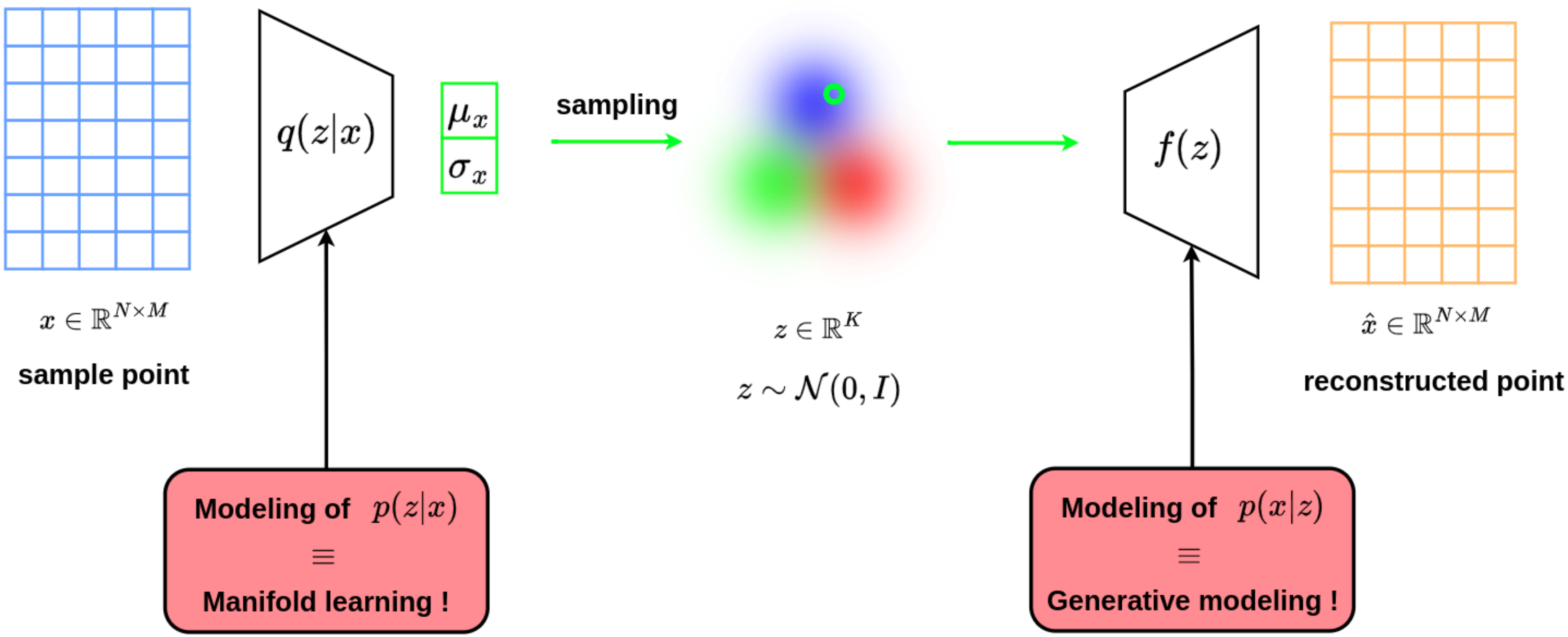
  - ➔ Impose that all distributions $q_x(z)$ are close to a standard normal distribution $\mathrm{N}(0, \mathrm{I})$
  - ➔ Variances close to 1 => limits the generation of point distributions
  - ➔ Means close to 0 => encourages distributions that are close to each other



Key concepts

- Probabilistic framework: *continuity & completeness*

  → Architecture of VAEs



Variational Auto-Encoder

Mathematical formulation

What is the purpose of generative models?

- How to generate synthetic faces?

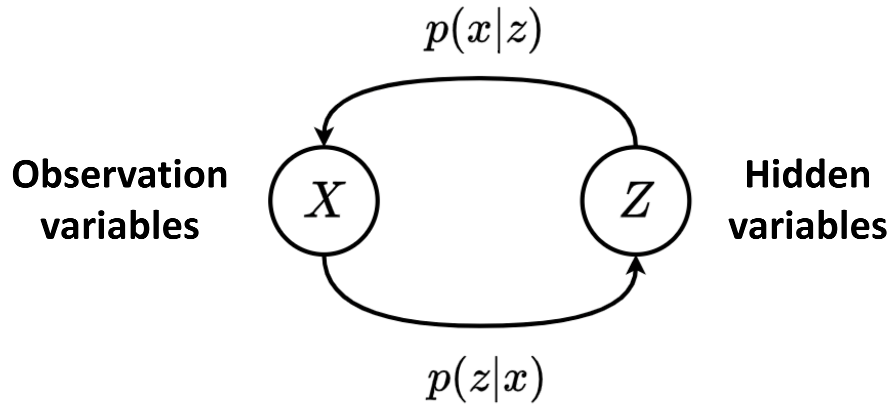
  - ➔ Let $p(\cdot)$ be the distribution that represents human faces

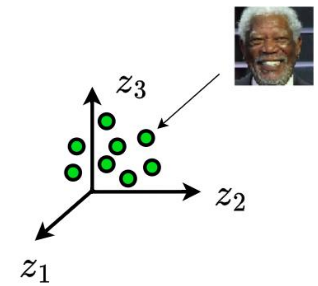
  - ➔ We want to find a model f that generates samples x whose probability $p(x)$ is maximal

$$f^* = \arg\max_f \; p(x) \quad \textbf{with } x \textbf{ generated by a model } f(\cdot)$$

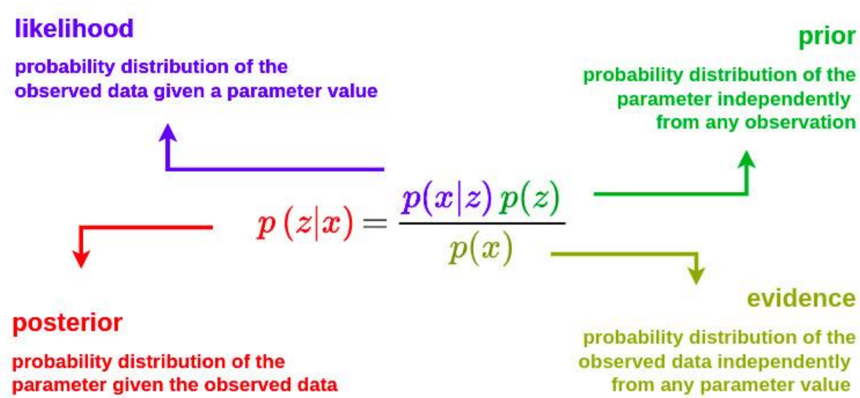
  - ➔ In this case, the generated samples resemble human faces from the training dataset

- Modeling a hidden variable z to reduce the complexity of the problem



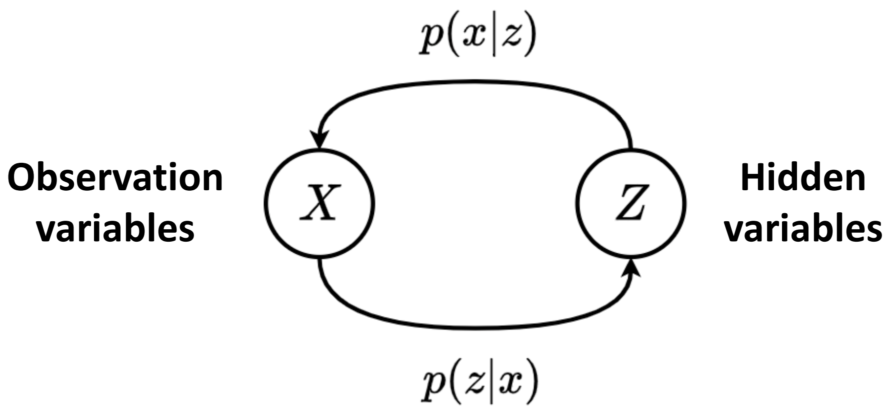


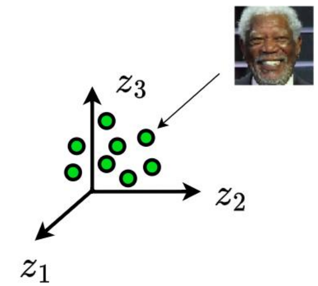
→ Reminder of Bayes' theorem



Probabilistic framework

- Mathematical formulation





→ The distribution $p(z|x)$ is generally complex to model

Approximation of $p(z|x)$ by a simple and computable function $q(z|x)$ that will allow efficient sampling of z

- Variational inference
  - ➔ Statistical approximation technique for complex distributions, here $p(z|x)$
  - ➔ Definition of a parameterized family of distributions
    - e.g., family of Gaussians with parameters μ_x, σ_x modeled by functions to be determined
  - ➔ Find the best approximation of the target distribution in this family
  - ➔ The best element of the family minimizes an approximation error measure between two distributions
    - Kullback-Leibler divergence function is often used

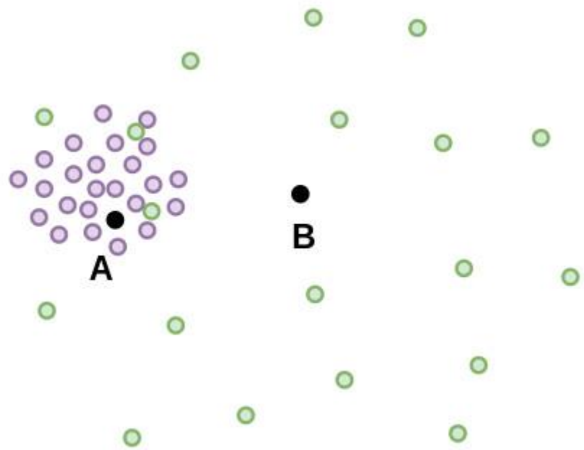
Kullback-Leibler divergence function

➔ Distance measure between two distributions via relative entropy

$$D_{KL}(p \parallel q) = \int p(x) \cdot \log\left(\frac{p(x)}{q(x)}\right) dx$$

➔ D_{KL} is a measure that is always positive $D_{KL}(p||q) \geq 0$

➔ D_{KL} is a nonsymmetric measure $D_{KL}(p||q) \neq D_{KL}(q||p)$



- For the purple distribution, the distance AB is large
- For the green distribution, the distance AB is moderate
- The notion of distance differs depending on the distributions

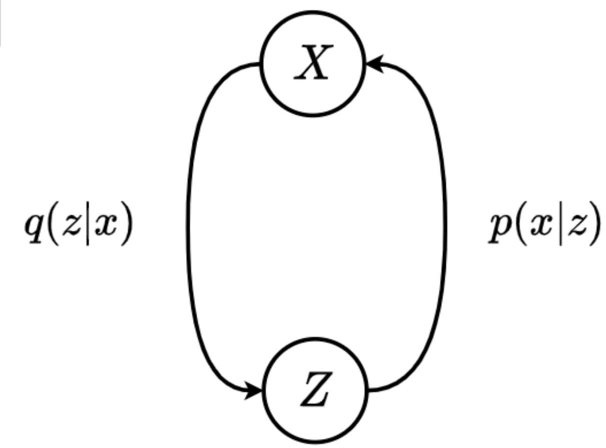
Probabilistic framework

- Variational inference

  - ➔ $p(z|x)$ is approximated by a family of functions $q(z|x)$
  - ➔ $q(z|x)$ is modeled by a Gaussian distribution aligned with the axes

$$q(z|x) = \mathcal{N}\left(\mu_x, \sigma_x\right) = \mathcal{N}\left(g(x), diag(h(x))\right)$$

  - ➔ $g(x)$ and $h(x)$ are functions that represent the means μ_x and the covariances σ_x
  - ➔ Measure of approximation between the two distributions $p(z|x)$ et $q(z|x)$



$$(g^*, h^*) = \underset{(g,h)}{\arg\min}\ D_{KL}\left(q(z|x) \parallel p(z|x)\right)$$

- Variational inference

  - By playing with the expressions of $p(x)$, it is possible to find the following definitions and relationships

$$log\ p(x) \geq \int q(z|x)\, log \left(\frac{p(x,z)}{q(z|x)} \right) dz$$

$$log\ p(x) \geq ELBO$$

$$log\ p(x) = ELBO + D_{KL}\left(q(z|x)\ \|\ p(z|x)\right)$$

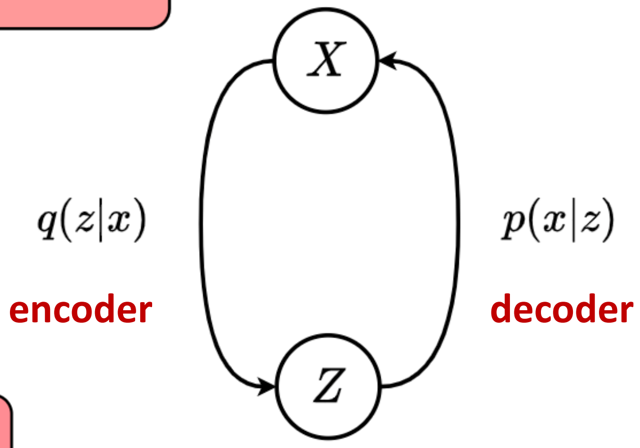
  - ELBO is a lower bound of $\log p(x)$
  - Maximizing ELBO amounts to maximizing $\log p(x)$
  - If we maximize $\log p(x)$, then we minimize $D_{KL}(\, q(z|x)\ ||\ p(z|x)\,)$

- Optimization process

$$(f^*, g^*, h^*) = \underset{(f,g,h)}{\arg\min} \left(\mathbb{E}_{z\sim q(z|x)}\left[\alpha\|x - f(z)\|^2\right] + D_{KL}\left(q(z|x) \parallel p(z)\right)\right)$$

- Deep learning loss function

$$\text{loss} = \alpha\|x - f(z)\|^2 + D_{KL}\left(\mathcal{N}\left(g(x), diag\left(h(x)\right)\right), \mathcal{N}\left(0, I\right)\right)$$

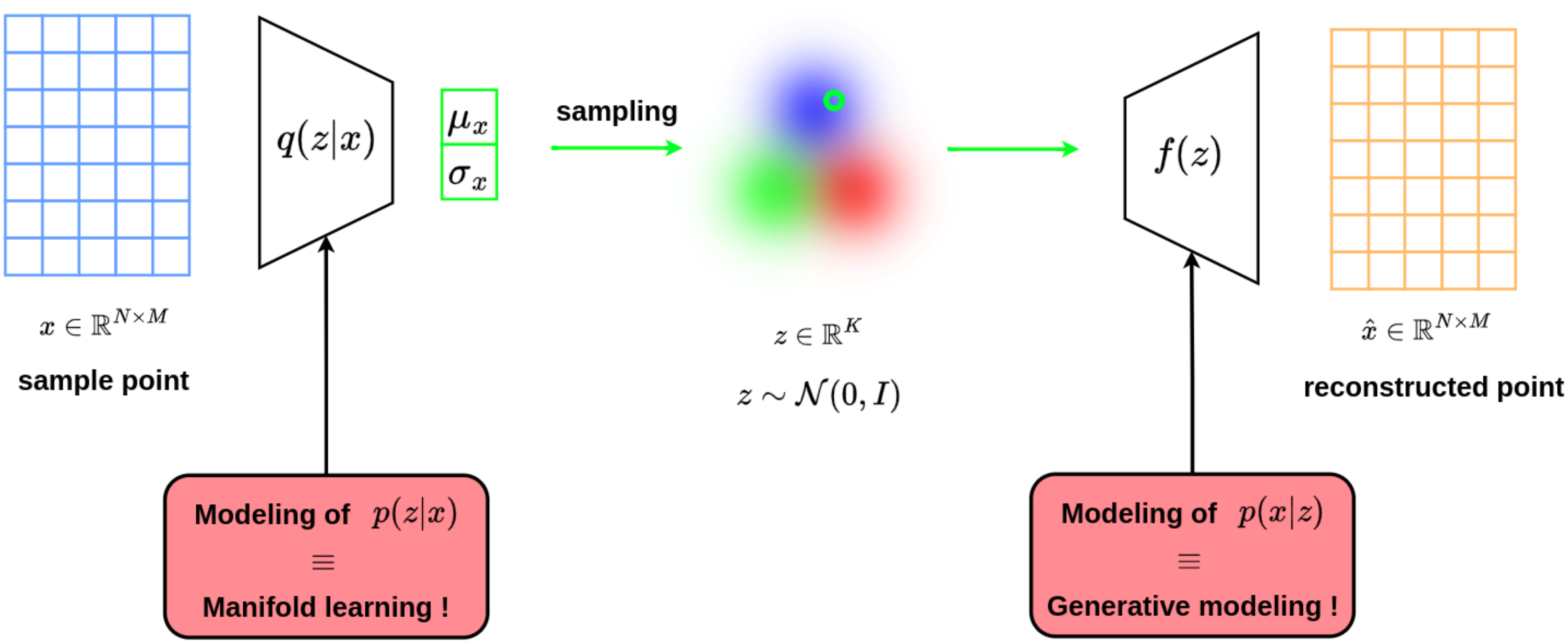


➔ $g(\cdot)$ et $h(\cdot)$ are modelled by an encoder

➔ $f(\cdot)$ is modelled by a decoder

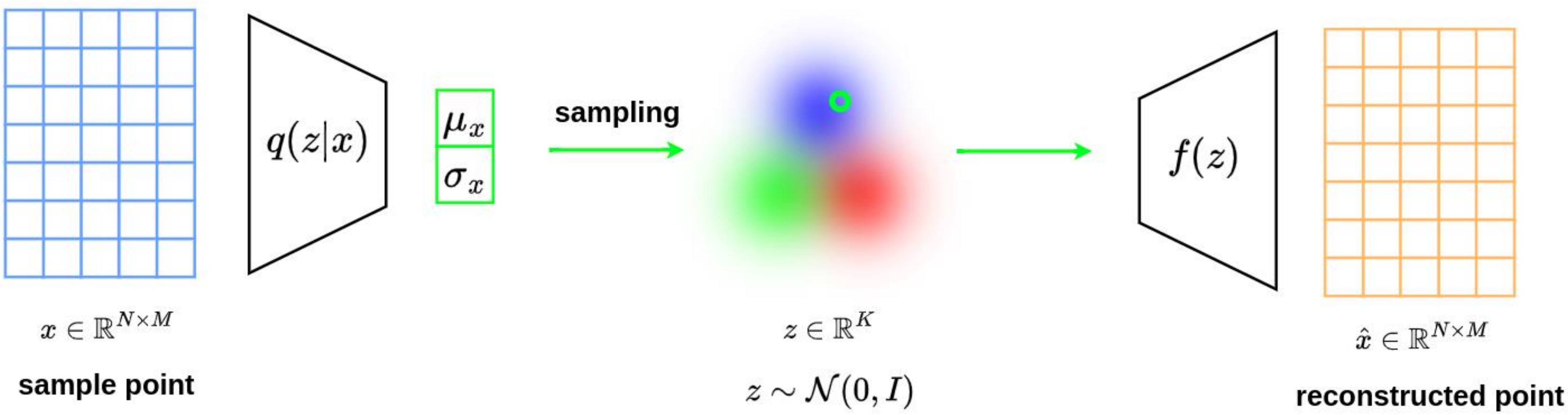
- Interpretation of the loss function

$$\text{loss} = D_{KL}\left(\mathcal{N}\left(g(x), diag\left(h(x)\right)\right), \mathcal{N}\left(0, I\right)\right) + \alpha \|x - f(z)\|^2$$



- Interpretation of the loss function

$$\text{loss} = D_{KL}\left(\mathcal{N}\left(g(x), diag\left(h(x)\right)\right), \mathcal{N}\left(0, I\right)\right) + \alpha\|x - f(z)\|^2$$

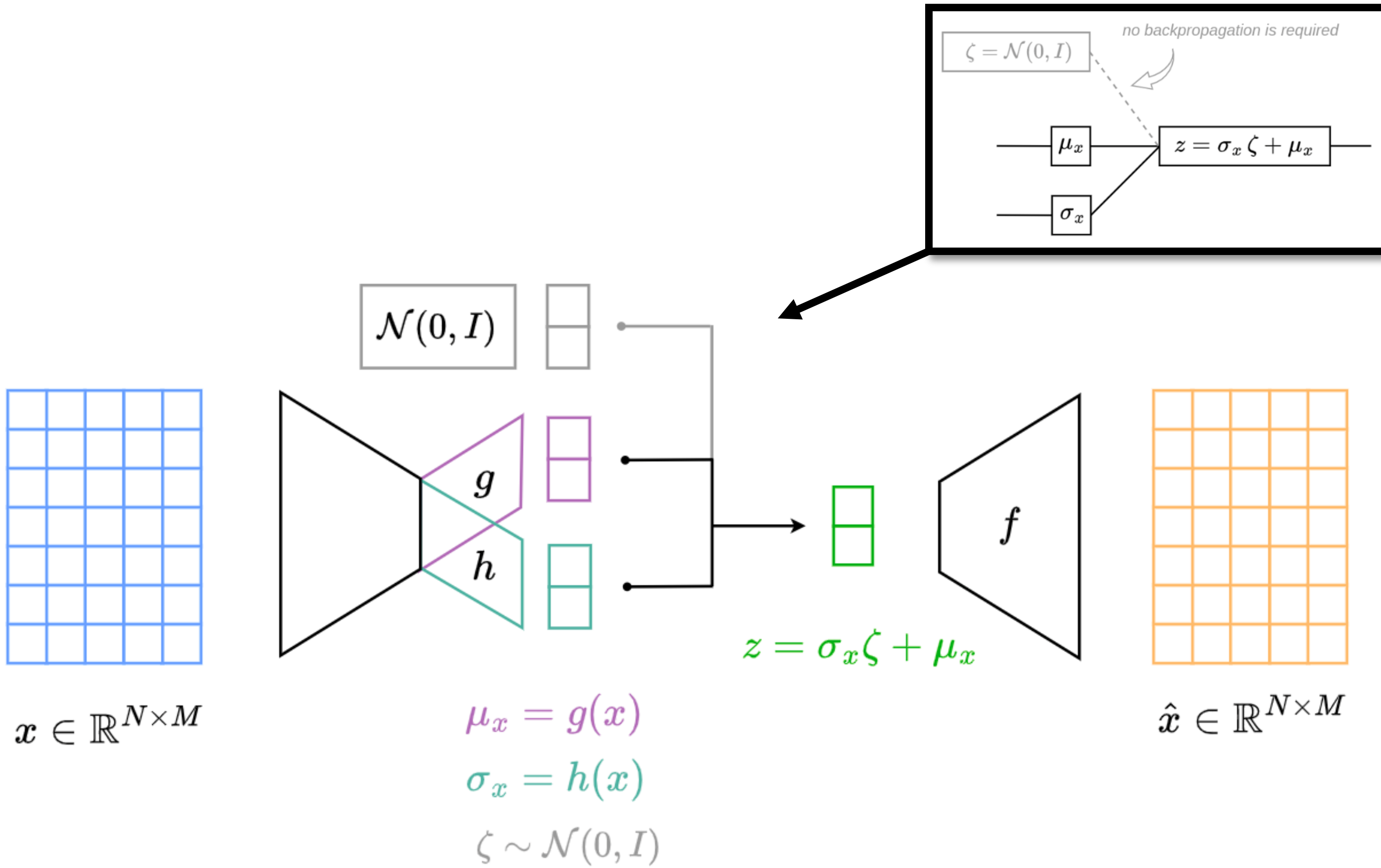


→ $\mathcal{N}\left(g(x), h(x)\right)$ **imposes a local *continuity* constraint**

→ $\mathcal{N}\left(\cdot, \mathcal{N}(0, I)\right)$ **imposes a global *completeness* constraint**

Implementation through deep learning

- Reparameterization trick



Practical application

The obsession is to master the latent space !

Structuration of the latent space: AR-VAE

- VAE latent space regularization based on image attributes

  - Structured latent space generation

    → Specific attributes with continuous values must be coded according to specific dimensions

$$Loss = \mathcal{L}_{recons} + \beta \mathcal{L}_{KLD} + \gamma \sum_{l=0}^{L-1} \mathcal{L}_{r_l, a_l}$$

Data attachment term

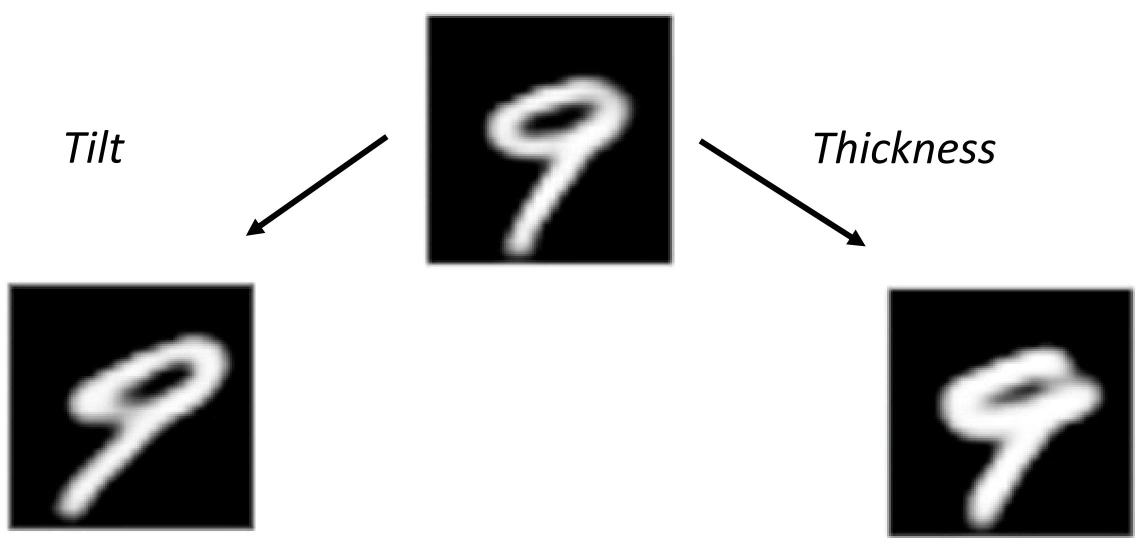
KL divergence term

Attribute regularization term

Structuring latent space

Structuration of the latent space: AR-VAE

- Attribute regularization term
  - What is an attribute ?
    - → Measurement performed in image space to characterize a target object
    - → E.g.: handwritten digits (MNIST database)
      - Attributes: line thickness, inclination, length, area, ...
    - → Pre-training image attribute measurements used as input data



▶ Attribute regularization term

- During the learning phase

➔ Computation for each attribute a of a distance matrix $D_a \in \mathbb{R}^{m \times m}$ from the m images $\{x_i\}_{1 \leq i \leq m}$ present in the current batch

$$D_a(i,j) = a(x_i) - a(x_j) \quad \text{with} \quad i,j \in [0,m)$$

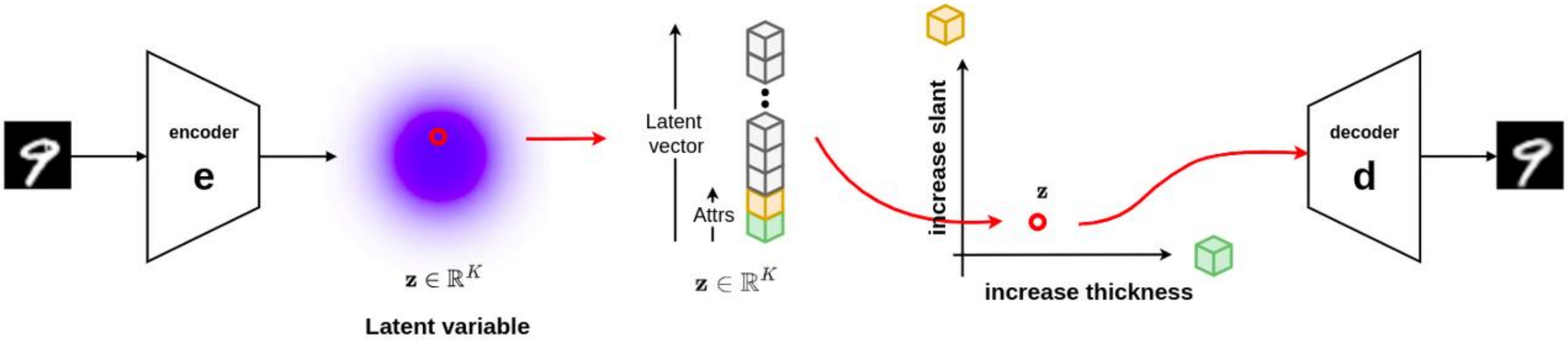
➔ Computation for each attribute r of a distance matrix $D_r \in \mathbb{R}^{m \times m}$ from the m latent vector $\{z_i\}_{1 \leq i \leq m}$ corresponding to the images in the current batch

$$D_r(i,j) = z_i^r - z_j^r \quad \text{with} \quad i,j \in [0,m)$$

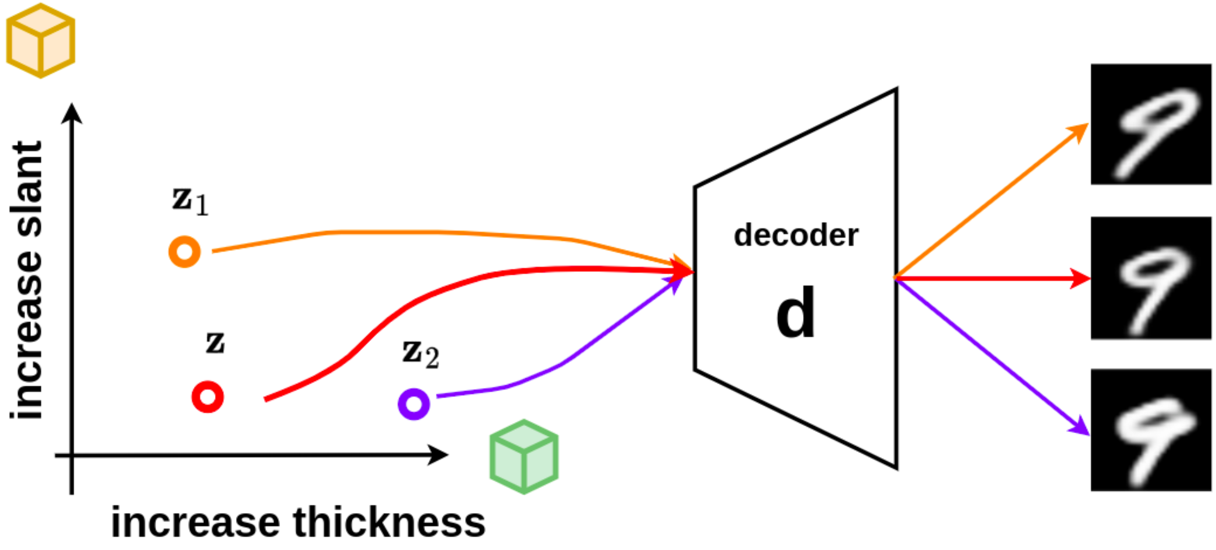
➔ Introduction of the following loss term

$$\mathcal{L}_{r,a} = MAE\left(\tanh\left(D_r\right) - sign\left(D_a\right)\right)$$

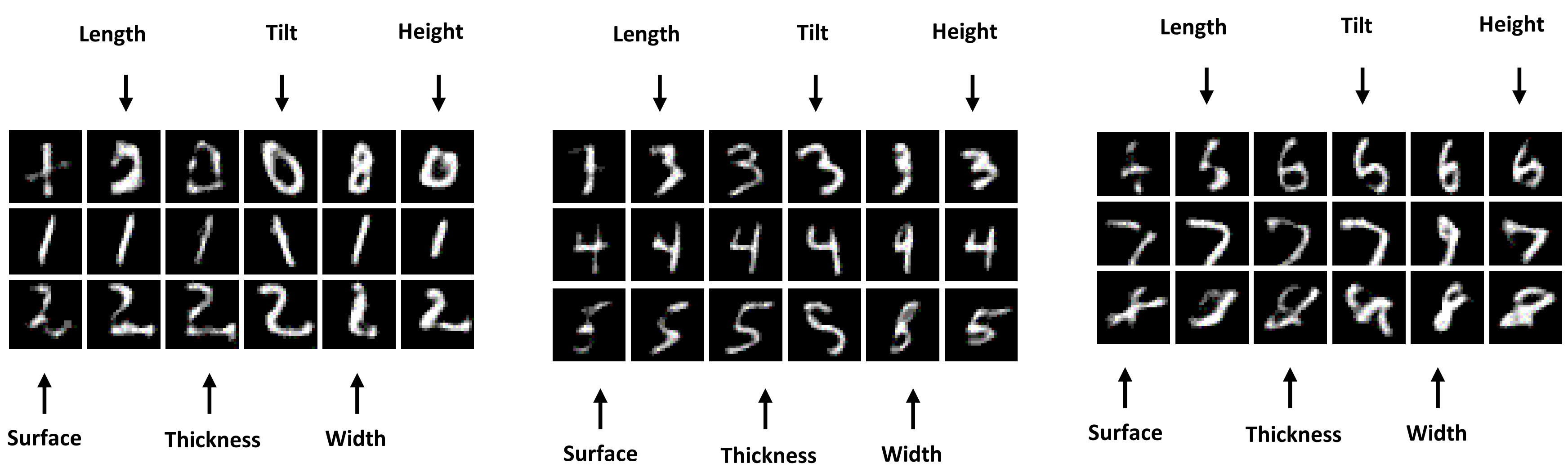
- Generate a latent space structured according to attributes



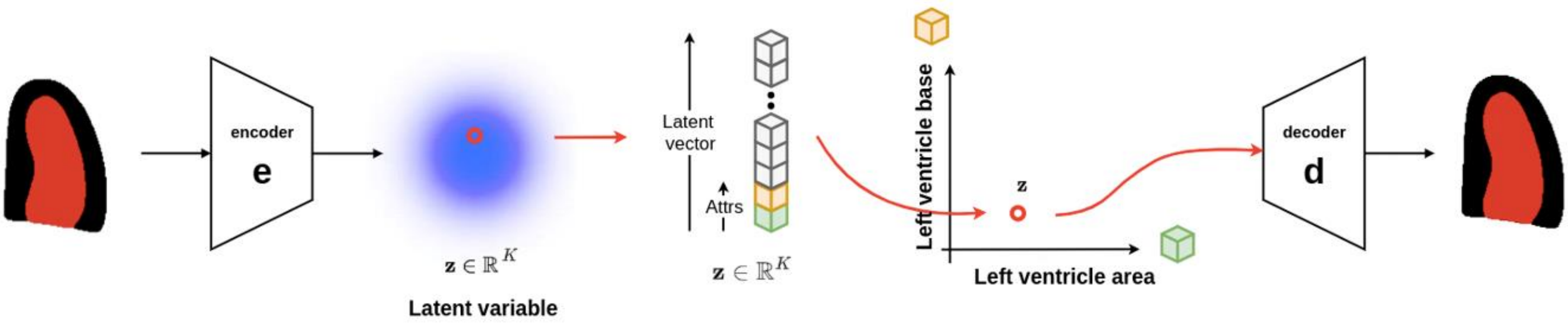
- Generate a latent space structured according to attributes
  - Sampling of the structured latent space



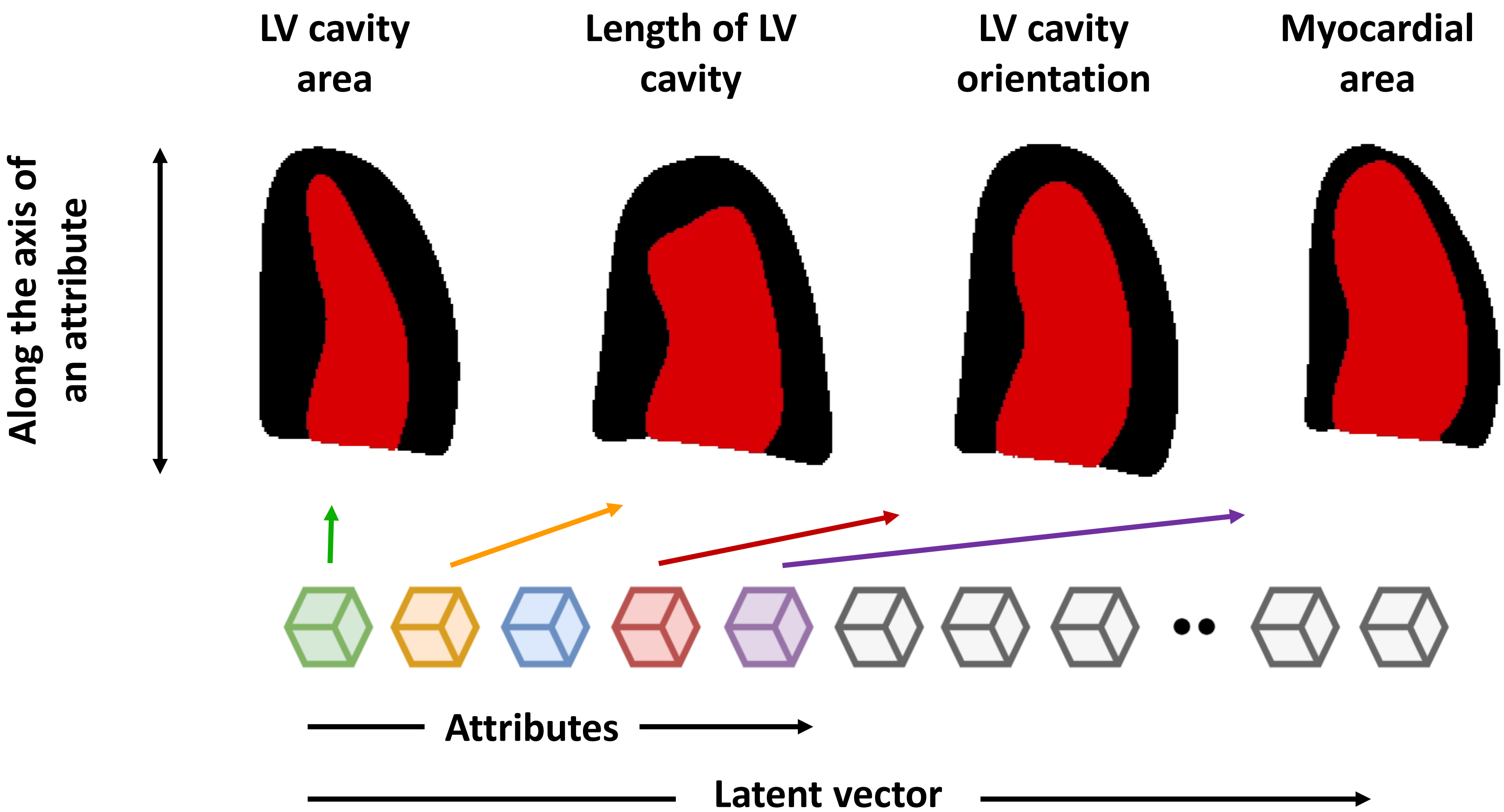
- Generate a latent space structured according to attributes
  - Sampling of the structured latent space
    - → Specific attributes: surface, length, thickness, inclination, width, height
    - → Each column corresponds to a traverse along a regularized dimension



- Application example: representation of cardiac shapes
  - Generation of a latent space structured according to the following attributes
    - ➔ Left ventricular (LV) cavity: surface area, length, basal width, orientation
    - ➔ Myocardial surface
    - ➔ Epicardial wall center



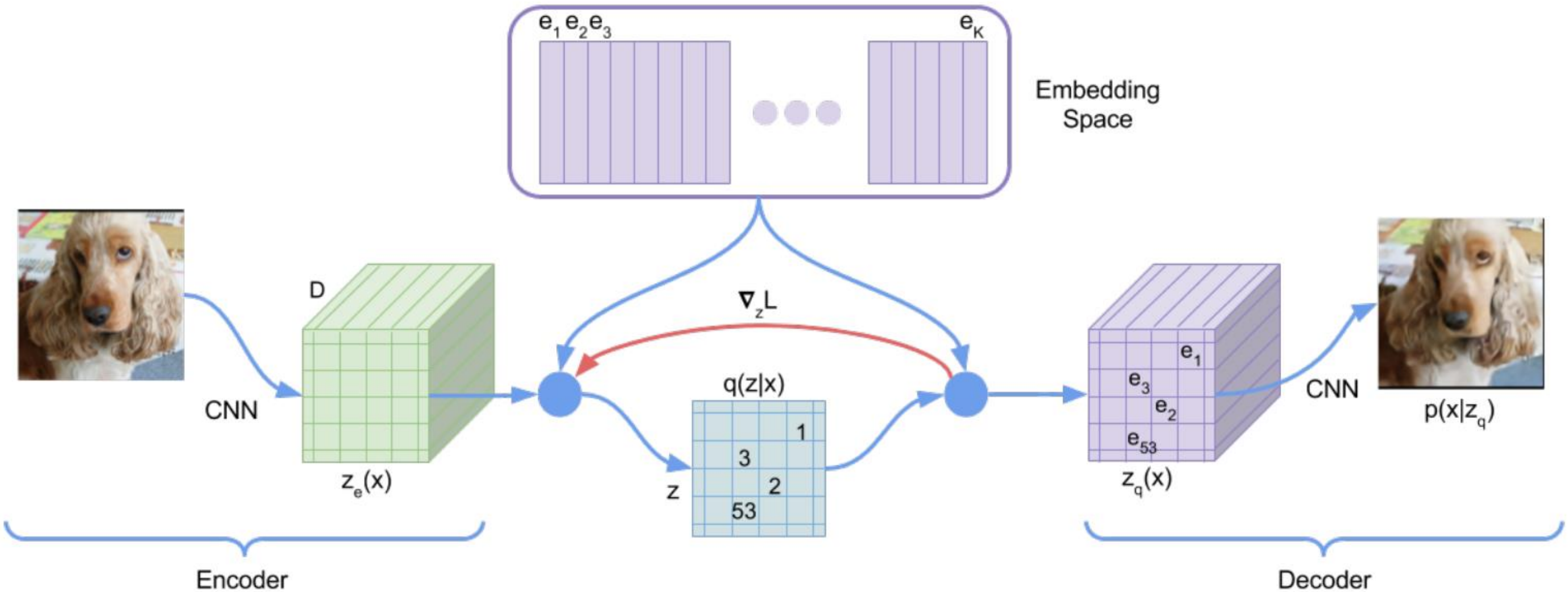
Structuration of the latent space: AR-VAE



Variational auto-encoders with vector quantization

Another VAE-inspired method: VQ-VAE

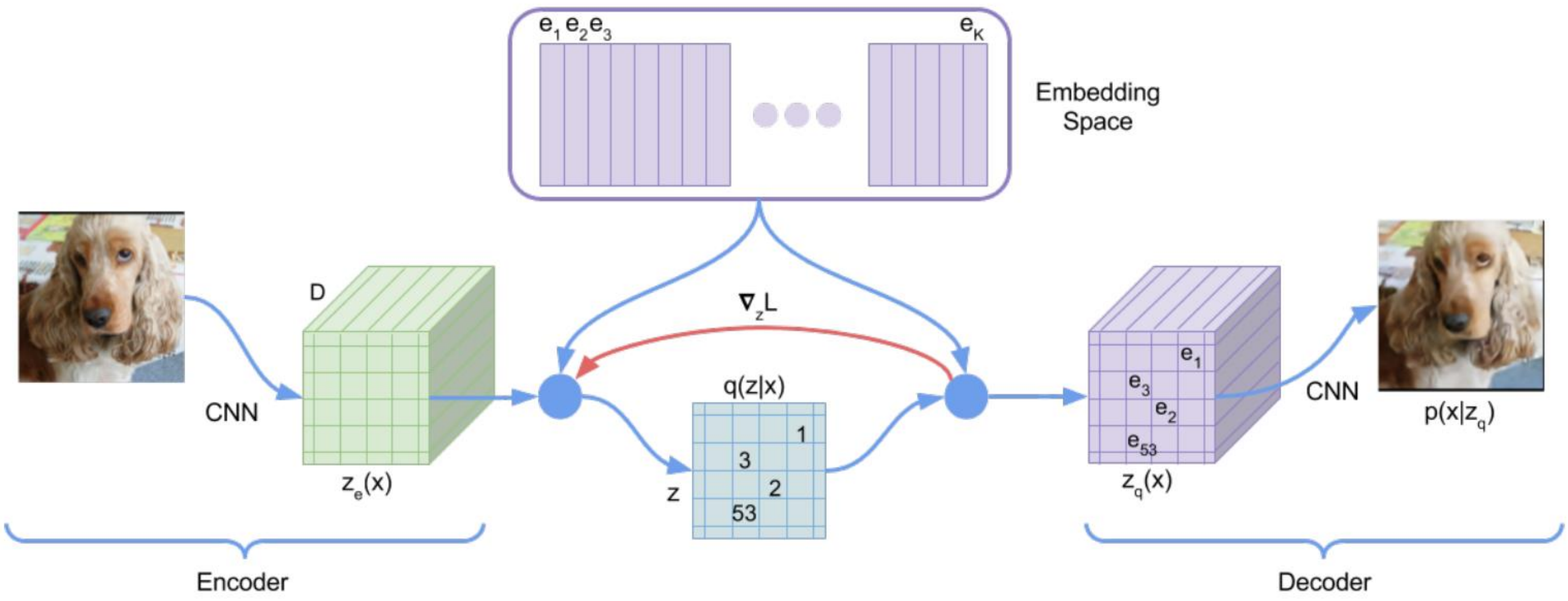
- Joint learning of an auto-encoder and a discrete latent space representation



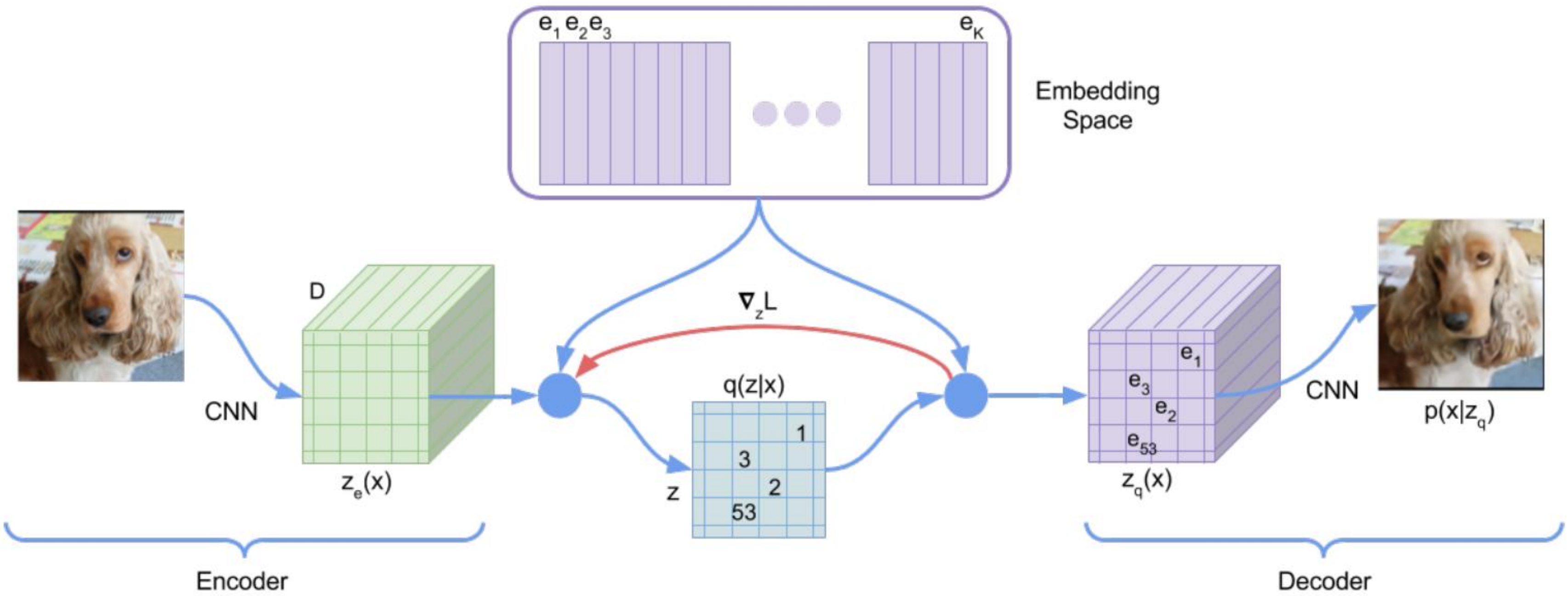
- The latent space is defined by the set of vectors $\{e_i\}_{i\in[1,K]}$ that are learned

Another VAE-inspired method: VQ-VAE

- The encoder outputs a matrix of size $[M \times M \times D]$ corresponding to $[M \times M]$ vectors of size D

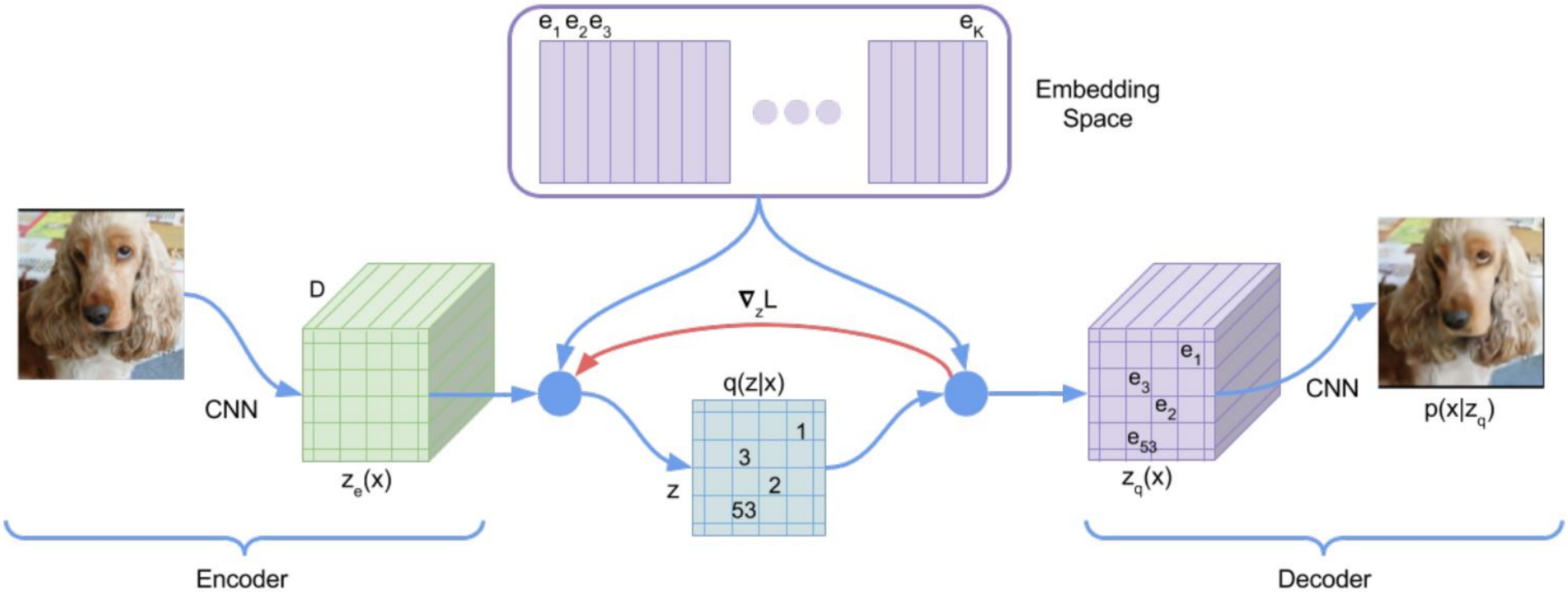


- Each encoder vector is compared with vectors in latent space, and the number of the closest vector is assigned in discrete space $q(z|x)$



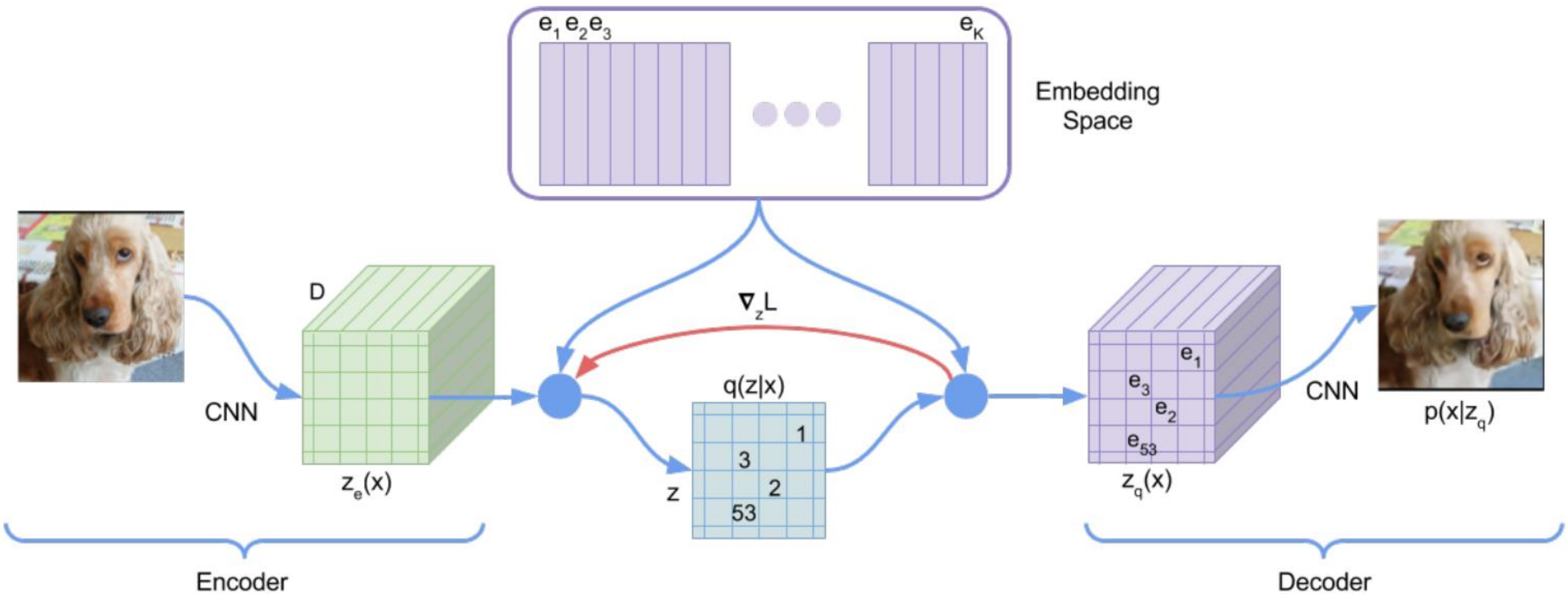
Another VAE-inspired method: VQ-VAE

- The decoder input corresponds to a matrix of size $[M \times M]$ where each component is a vector of size D
- Each component corresponds to a vector in latent space chosen according to its number in discrete space $q(z|x)$



- The loss function to be minimized is as follows

$$\mathcal{L} = log\left(p(x|z_q)\right) + \|sg\left[z_e(x)\right] - e\|_2 + \beta\|z_e(x) - sg\left[e\right]\|_2$$



$sg[\cdot]$: Identity function in forward and null function in backward

That's all folks

- Data representation