Generative AI for medical imaging

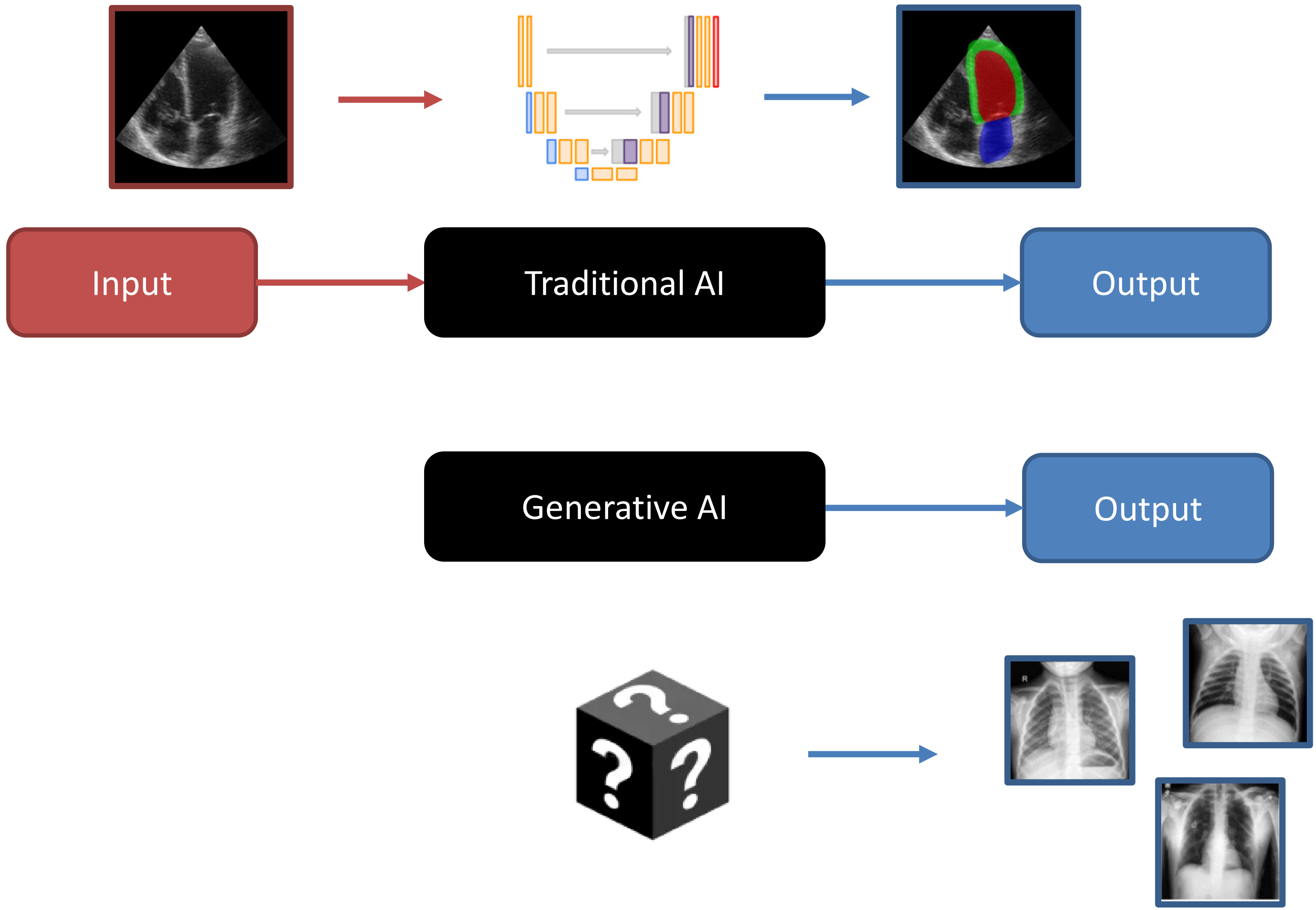
Recent advances and potential benefits

Olivier Bernard

olivier.bernard@insa-lyon.fr

Generative AI for imaging

Generative AI for medical imaging



- Key challenges

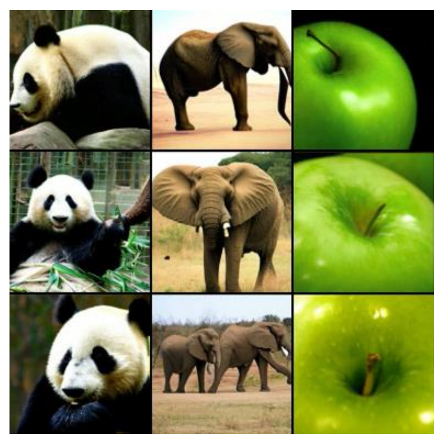
Generative ability

Conditioning

Multimodality

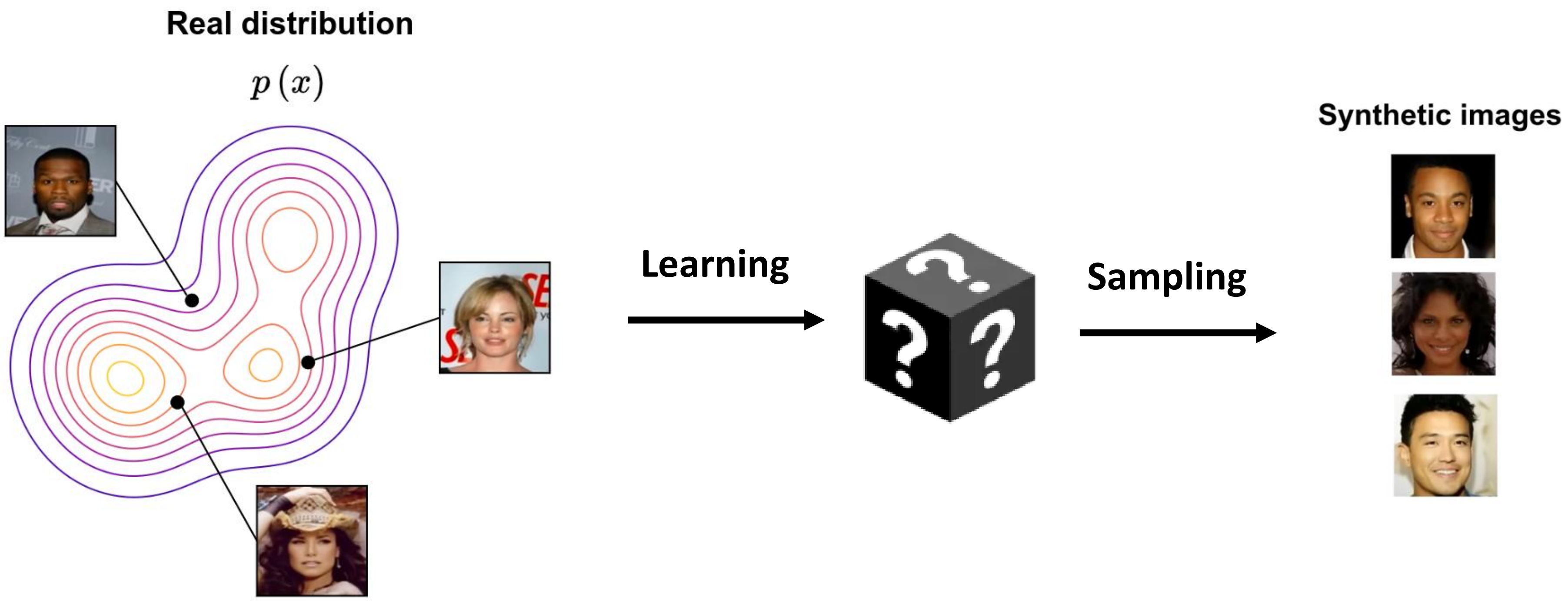
Real images

Synthetic images

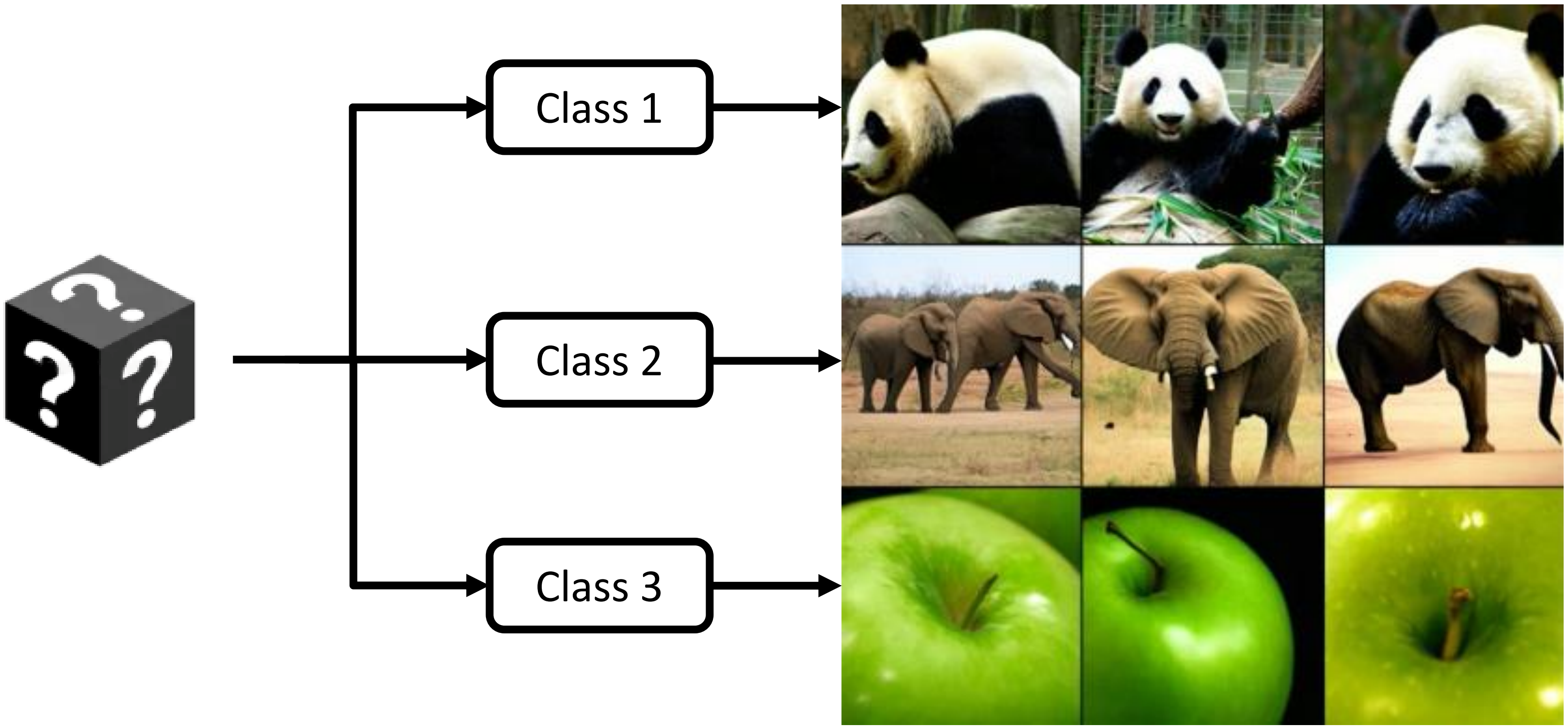


An Asian girl in ancient rides a giant panda

Generative ability



Conditioning

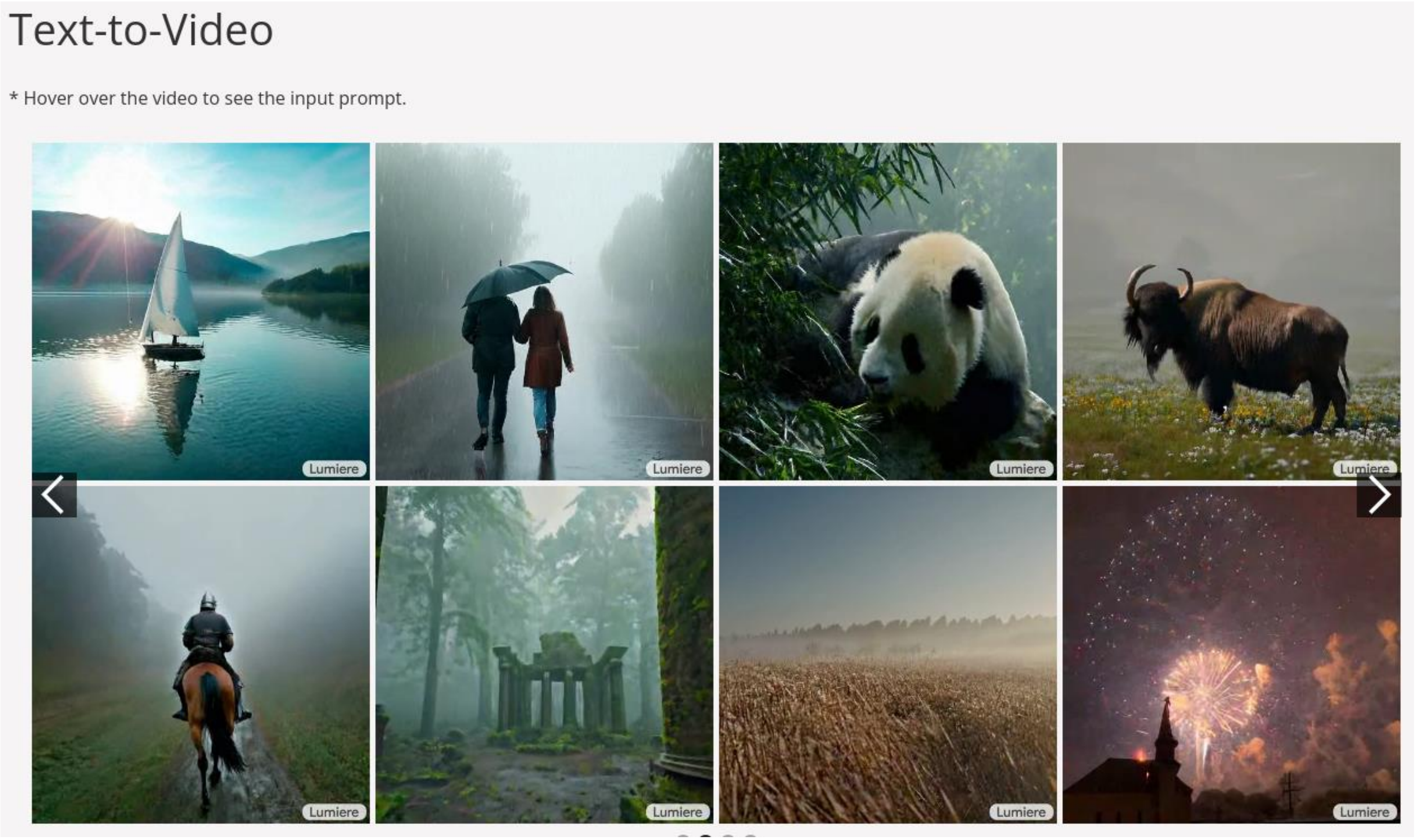


Multimodality

An Asian girl in ancient coarse linen clothes rides a giant panda and carries a wooden cage. A chubby little girl with two buns walks on the snow. High-precision clothing texture, real tactile skin, foggy white tone, low saturation, retro film texture, tranquil atmosphere, minimalism, long-range view, telephoto lens

Generative AI for medical imaging

Multimodality



https://lumiere-video.github.io/#section_image_to_video

Multimodality

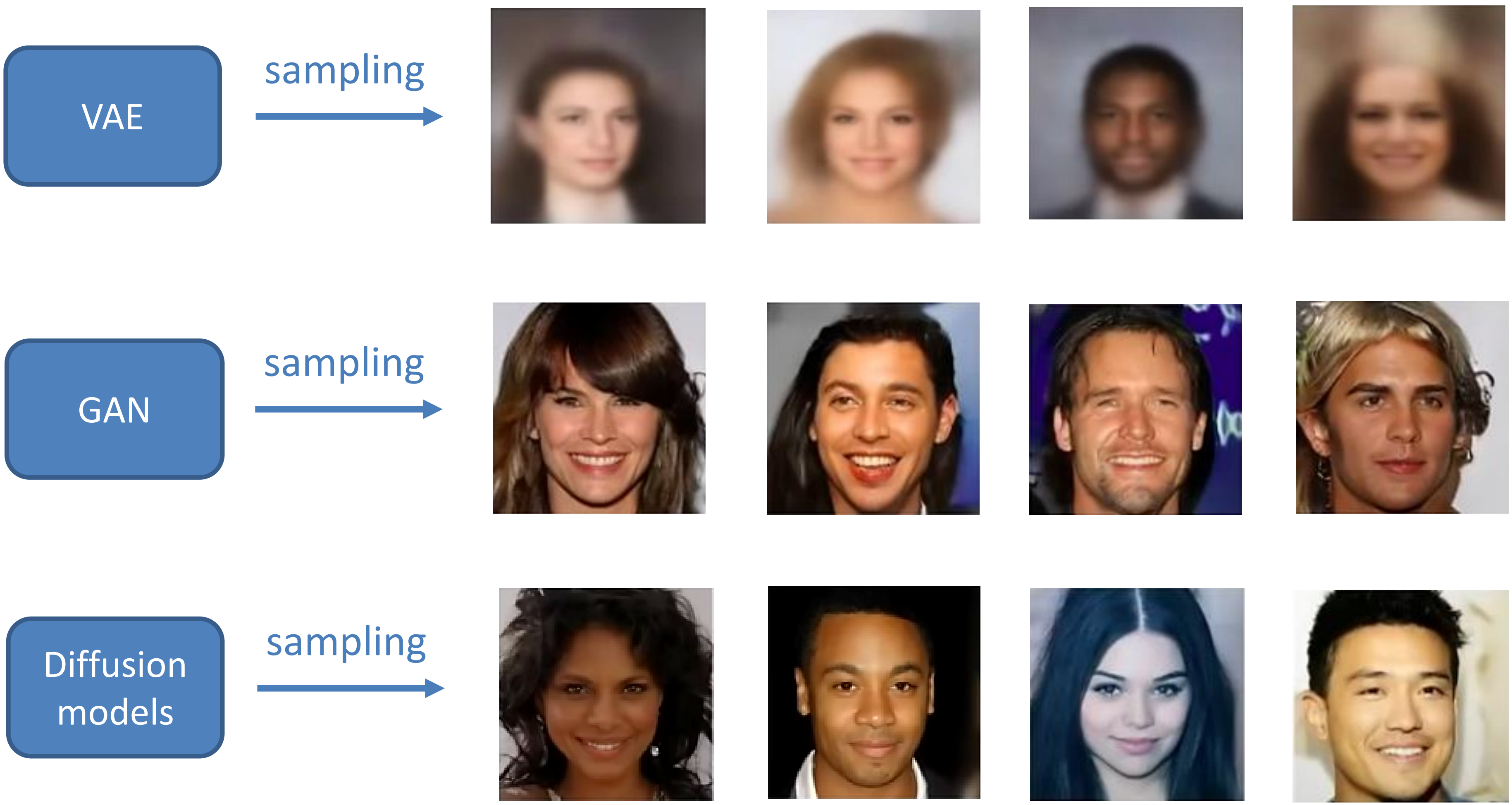
Image-to-Video

* Hover over the video to see the input image and prompt.

https://lumiere-video.github.io/#section_image_to_video

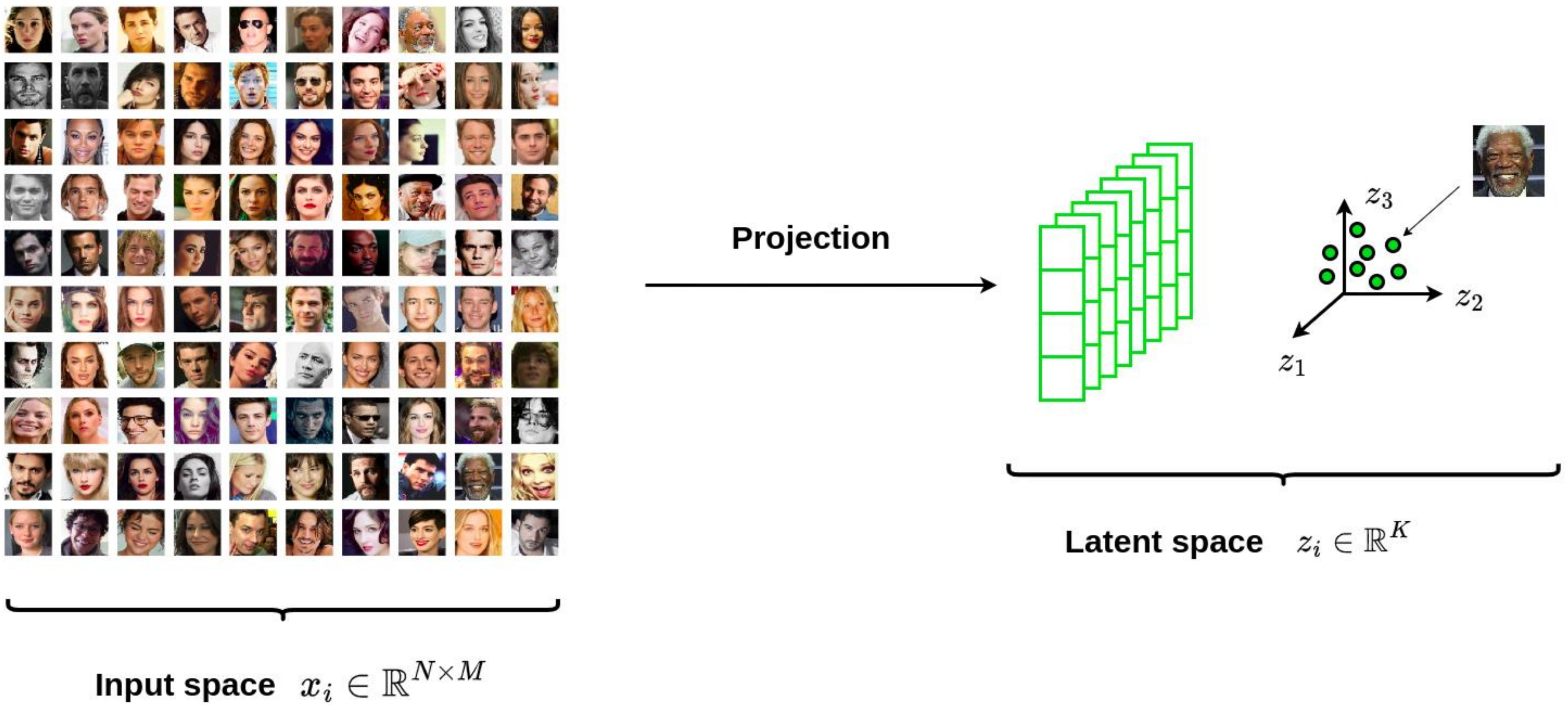
Generative AI for medical imaging

- Family of networks



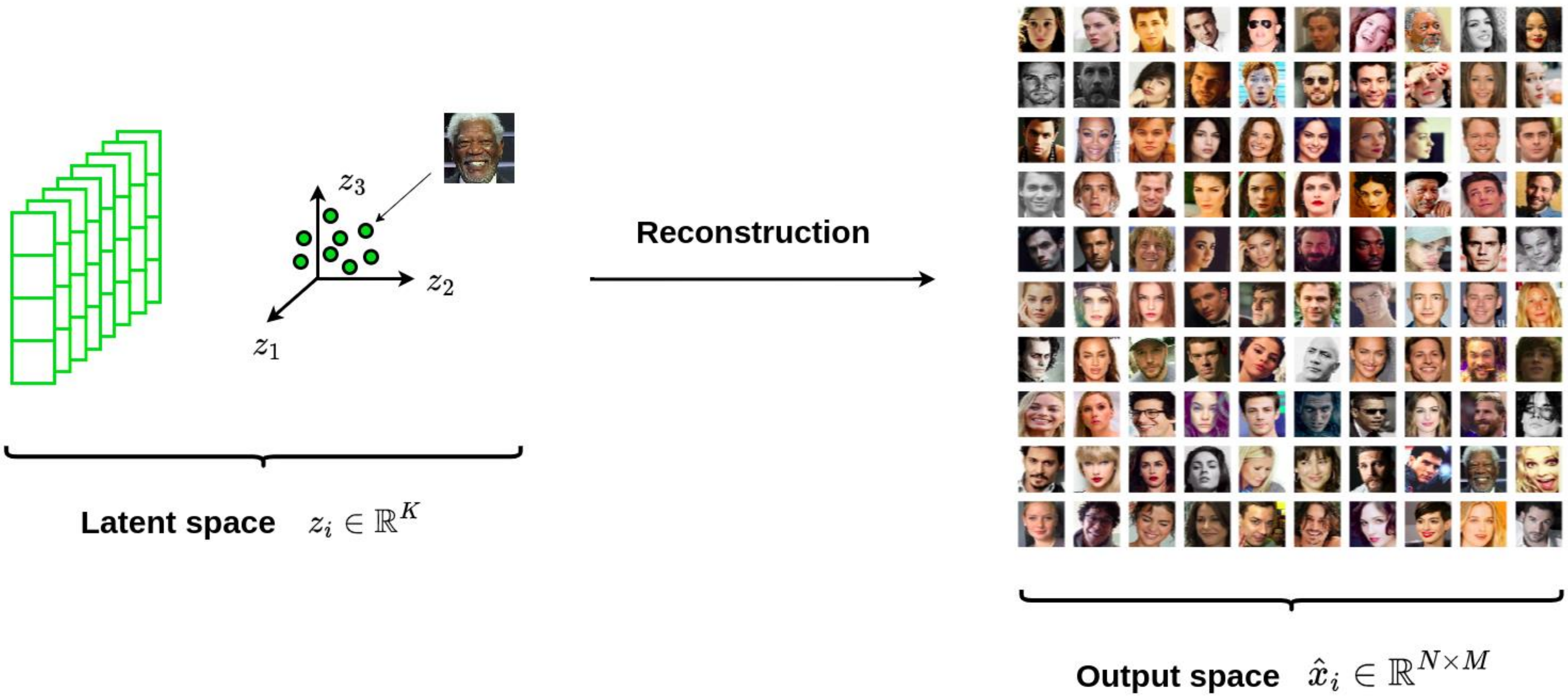
Variational auto-encoders

- Projection of the data into a lower dimensional space called latent space
- Interest: generating a more compact and interpretable representation



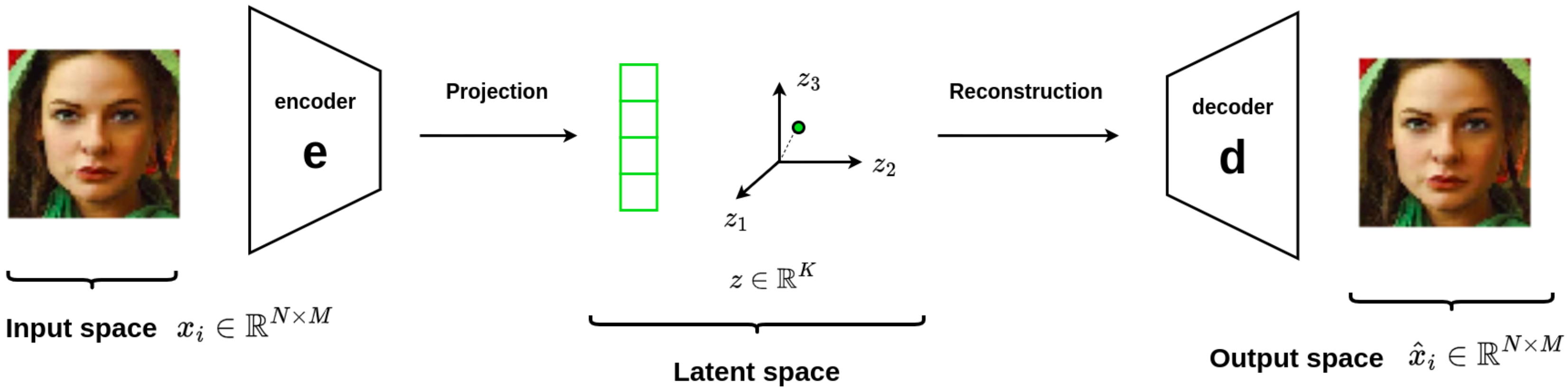
How to learn a complex distribution ?

- How to learn a relevant latent representation ?



Auto-encoder framework

- Standard encoder / decoder architecture

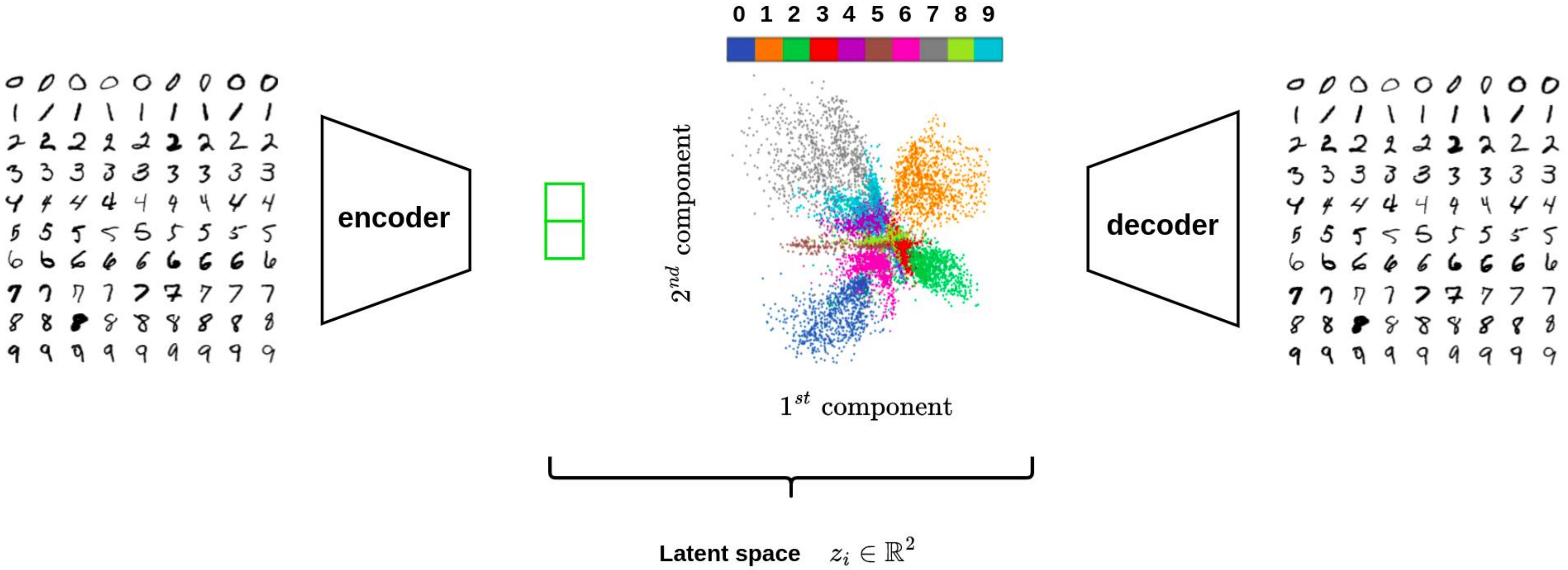


- Deep learning loss function

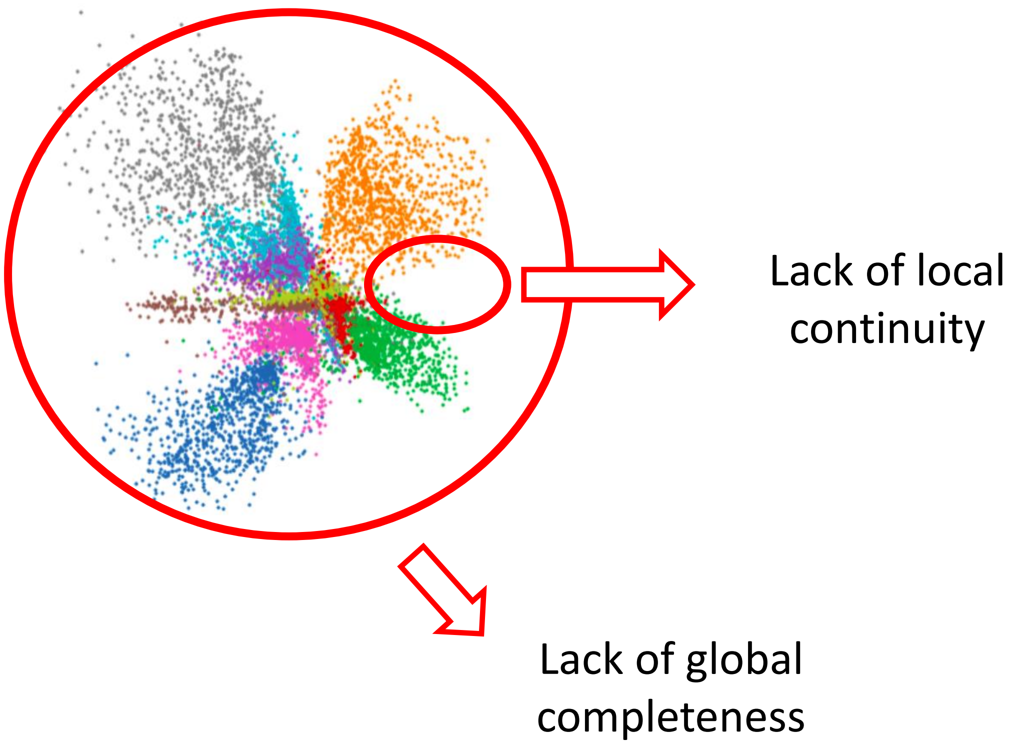
$$\text{loss} = \|x - \hat{x}\|^2$$

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

- Illustration from MNIST dataset
  - (train,test) = (60 000, 10 000)
  - Input image size: 32x32 / latent space K=16 (compression factor around 64)

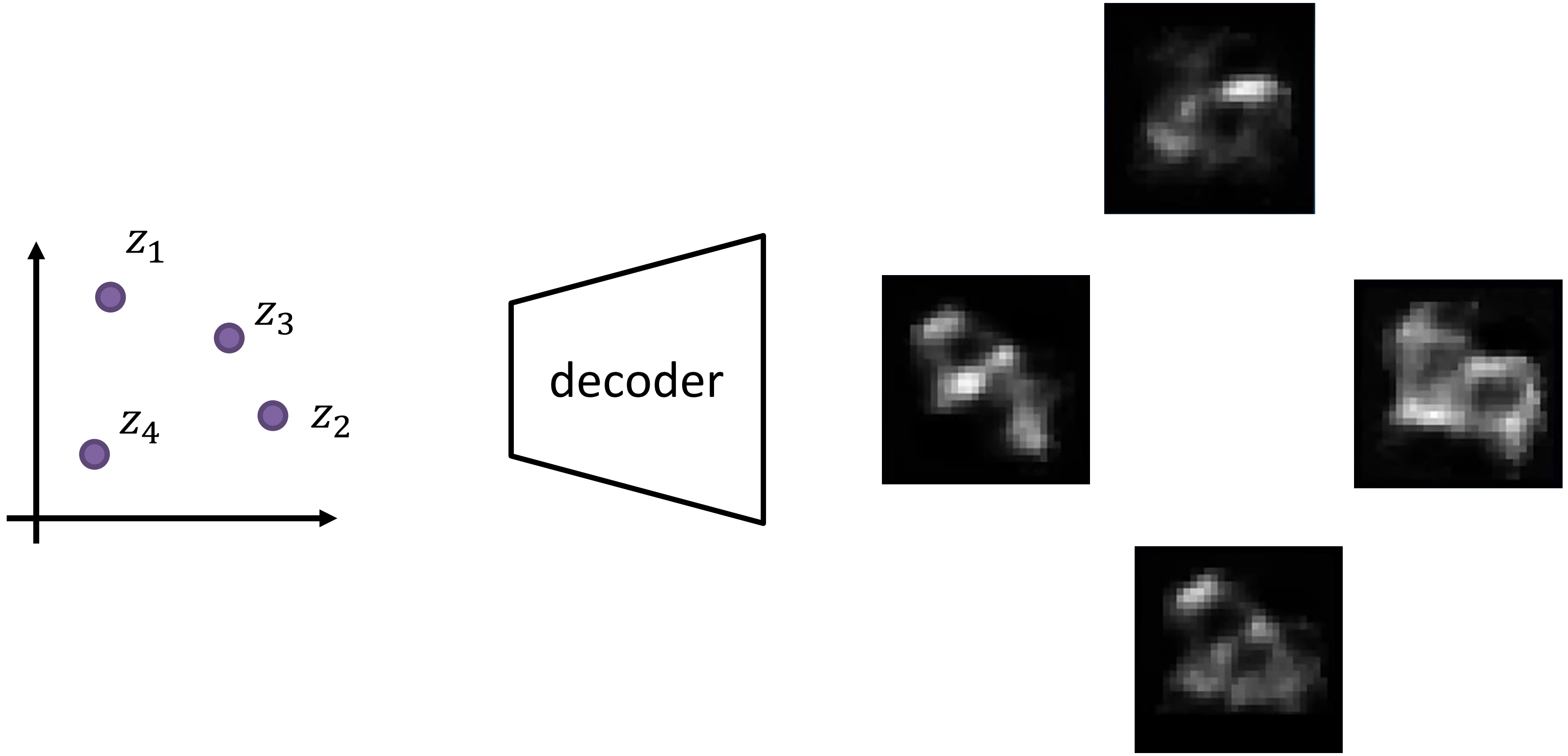


- Needs to better control the structure of the latent space



Auto-encoder weaknesses

- Sampling random latent vector

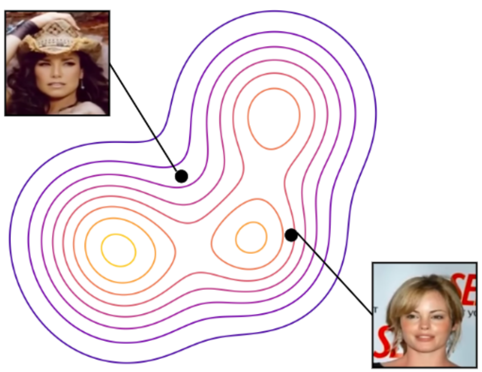


Variation Auto Encoder framework

- Starting point

Intractable distribution

$$p\left(x\right)$$

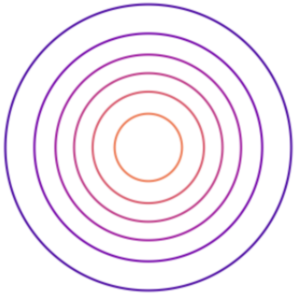


Data distribution

$$x \sim p\left(X\right) \in \mathbb{R}^{N \times M}$$

Controlled distribution

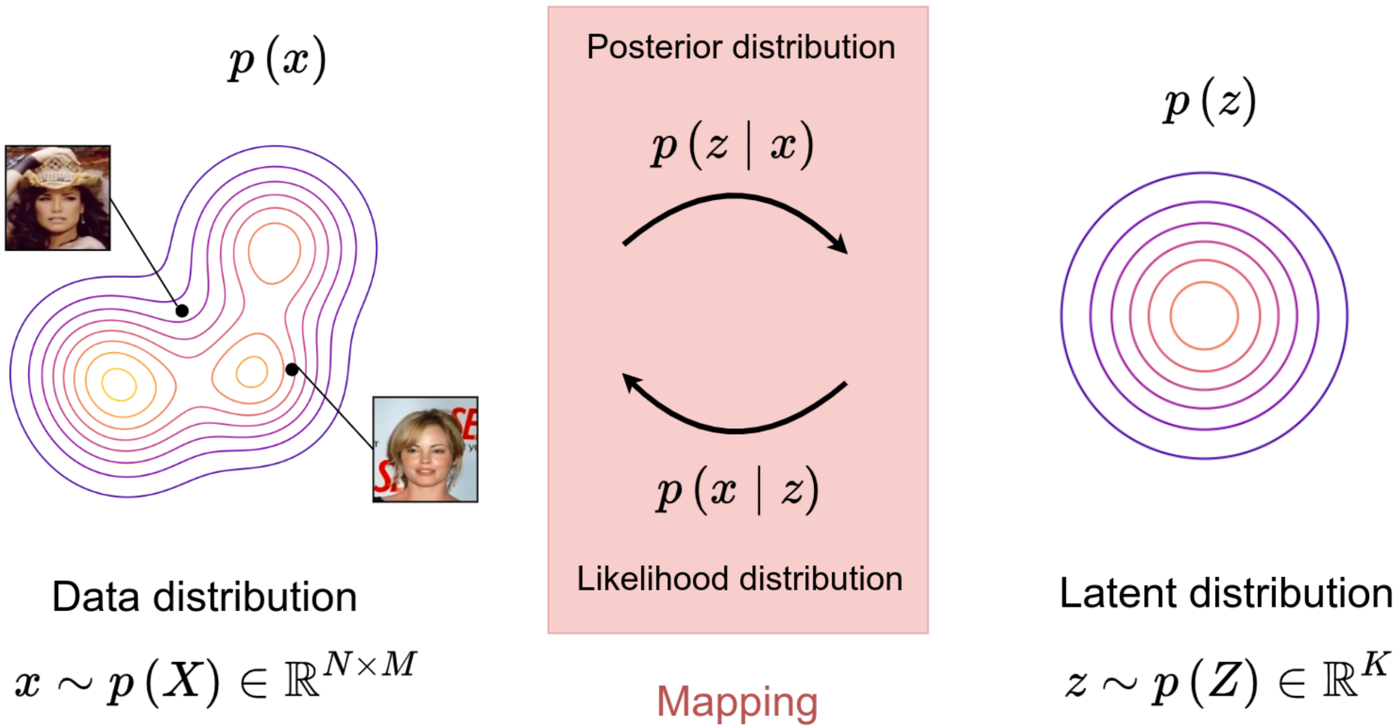
$$p\left(z\right)$$



Latent distribution

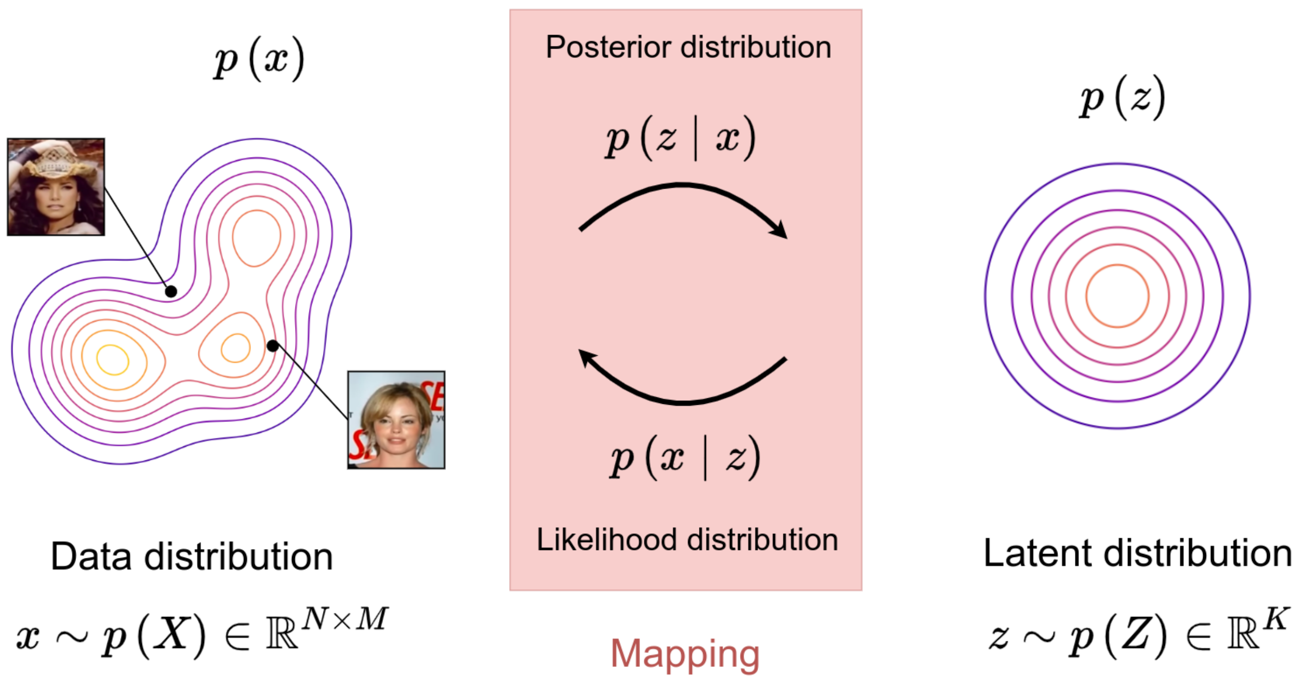
$$z \sim p\left(Z\right) \in \mathbb{R}^{K}$$

- Creating a mapping between the two distributions
  - → Through Bayesian statistics



Variation Auto Encoder framework

- **Strong assumptions**
  - Latent distribution $p(z)$ is assumed to be a normal distribution
  - The likelihood distribution is $p(x|z)$ assumed to be a Gaussian distribution whose parameters need to be learned
  - The posterior distribution $p(z|x)$ is intractable and needs to be approximated



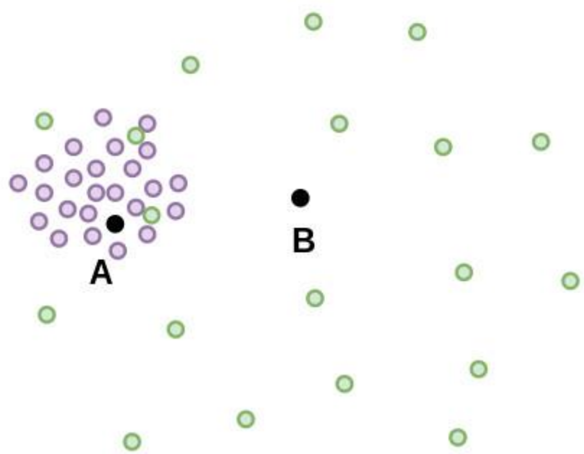
- Approximation of the posterior through variational inference
  - → Statistical approximation technique for complex distributions, here $p(z|x)$
  - → Definition of a parameterized family of distributions
    - e.g., family of Gaussian distributions with parameters μ_x, σ_x modeled by functions to be 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

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

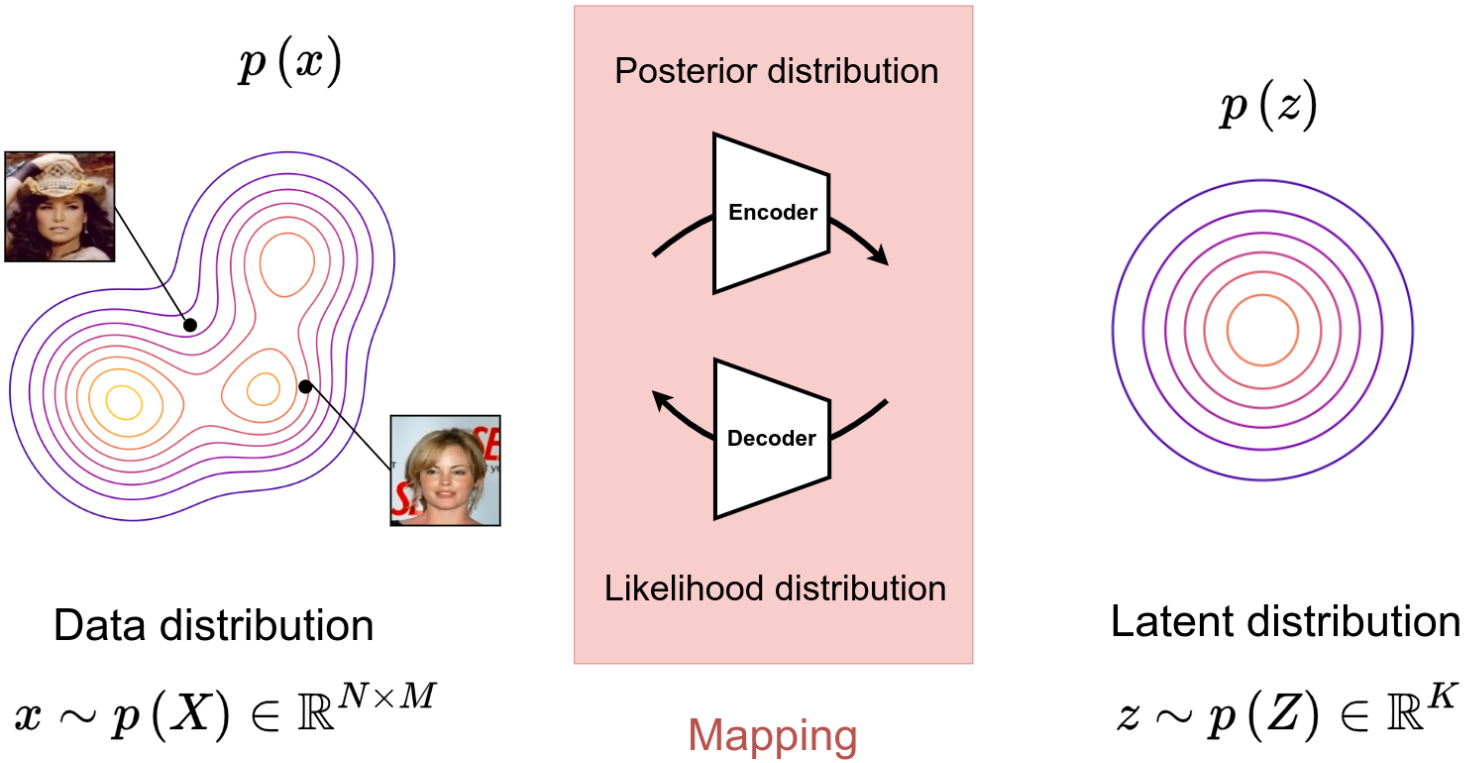
  - D_{KL} is a measure that is always positive $D_{KL}(p||q) \geq 0$
  - D_{KL} is a nonsymmetric measure $D_{KL}(p||q) \neq D_{KL}(q||p)$



- 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

Variation Auto Encoder framework

- Enforce a structured latent space with reduced dimensionalities
  - → Through Bayesian statistics



Variation Auto Encoder framework

- The prior is modeled through a Gaussian distribution

$$\boxed{p(z) = \mathcal{N}(0, I)}$$

- The likelihood is modeled through a Gaussian distribution

$$\boxed{p(x \mid z) = \mathcal{N}(\mu_z, \sigma_z) = \mathcal{N}\left(f(z), cI\right)}$$

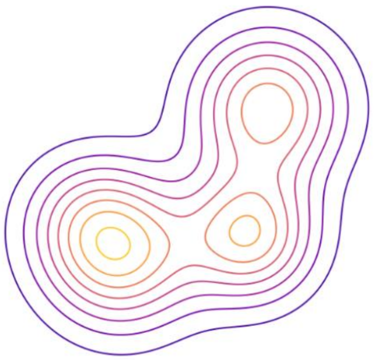
- The posterior is approximated by an axis-aligned Gaussian distribution

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

Variation Auto Encoder framework

- Optimization process
  - Sample a new data from the original data distribution
  - Pick a sample x that maximize $p(x)$, or $\log(p(x))$

$p\left(x\right)$



$$\boxed{\log\left(p(x)\right)}$$

$$\log\left(p(x)\right) = \log\left(\int p(x,z)dz\right)$$

Marginal distribution

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

Expectation reformulation

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

Jensen's inequality

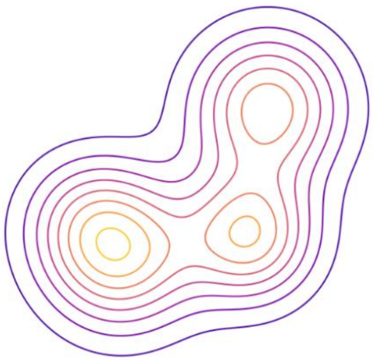
$$\boxed{\log\left(p(x)\right) \geq \mathbb{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)}{q(z \mid x)}\right)\right]}$$

- Evidence lower bound (ELBO)

$$\log\left(p(x)\right) \geq \boxed{\mathbb{E}_{q(z|x)}\left[\log\left(\frac{p(x,z)}{q(z\mid x)}\right)\right]}$$

ELBO

$p\left(x\right)$



→ Maximization of the ELBO

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

Bayes' formula
$p(x,z) = p(x|z)\,p(z)$

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

$$\mathcal{L}(x) = \mathbb{E}_{q(z|x)}\left[\log\left(p(x\mid z)\right) + \log\left(\frac{p(z)}{q(z\mid x)}\right)\right]$$

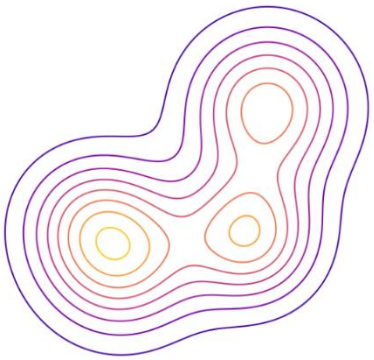
Kullback-Liebler divergence D_{KL}

$$\boxed{\mathcal{L}(x) = \mathbb{E}_{q(z|x)}\left[\log\left(p(x\mid z)\right)\right] - D_{KL}\left(q(z\mid x)\,\|\,p(z)\right)}$$

- ELBO maximization

$$\boxed{\mathcal{L} = \mathbb{E}_{z\sim q_x}\left[log\left(p(x|z)\right)\right] - D_{KL}\left(q(z|x) \parallel p(z)\right)}$$

$p\left(x\right)$



➔ Exploitation of the Gaussian assumption of the likelihood

$$p(x|z) = \mathcal{N}\left(f(z), cI\right)$$

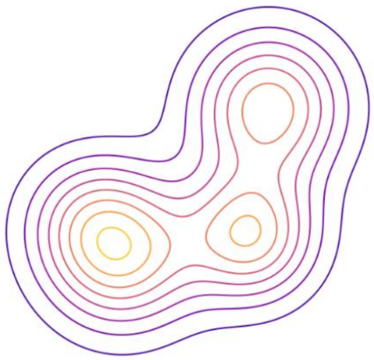
$$\boxed{\mathcal{L} \propto \mathbb{E}_{z\sim q_x}\left[-\alpha\|x - f(z)\|^2\right] - D_{KL}\left(q(z|x) \parallel p(z)\right)}$$

Variation Auto Encoder framework

- Optimization process

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

$p(x)$



- Deep learning loss function

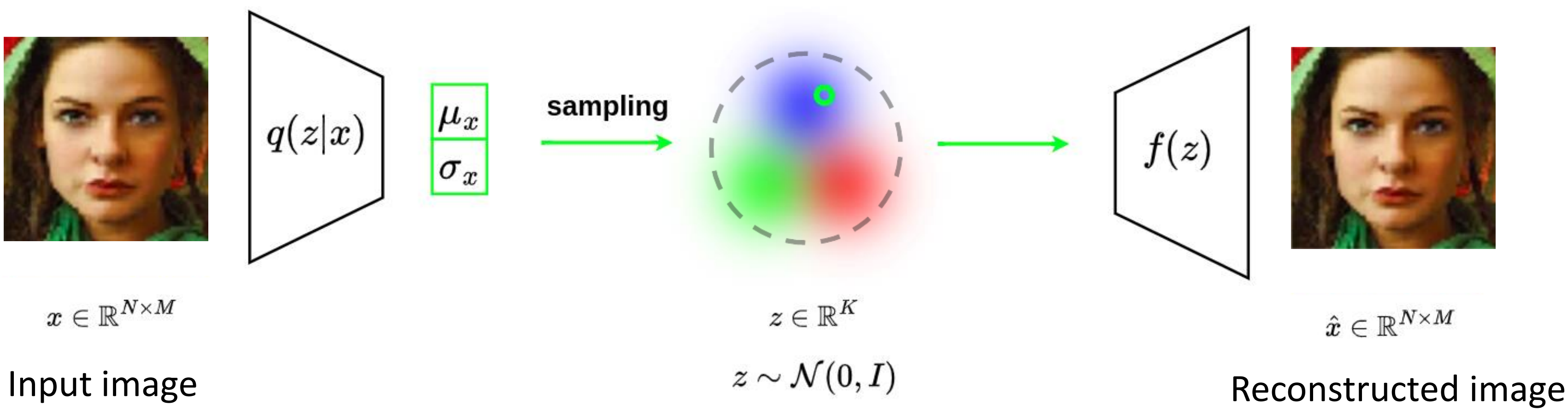
$$\text{loss} = \alpha\|x - f(z)\|^2 + D_{KL}\left(\mathcal{N}\left(g(x), diag\left(h(x)\right)\right), \mathcal{N}\left(0, I\right)\right)$$

➔ $g(\cdot)$ and $h(\cdot)$ are modeled through an encoder

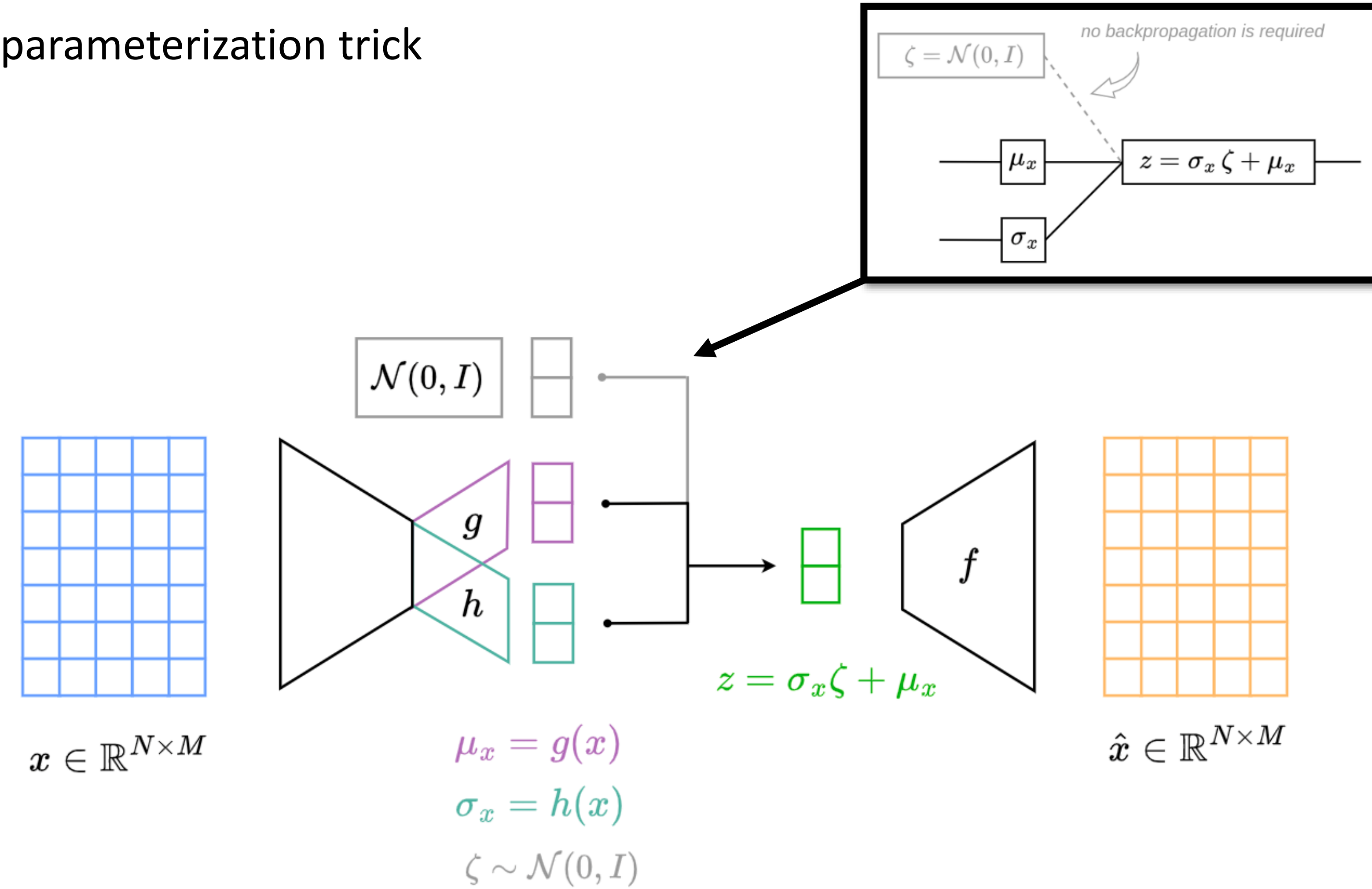
➔ $f(\cdot)$ is modeled through a decoder

- Interpretation of the loss function

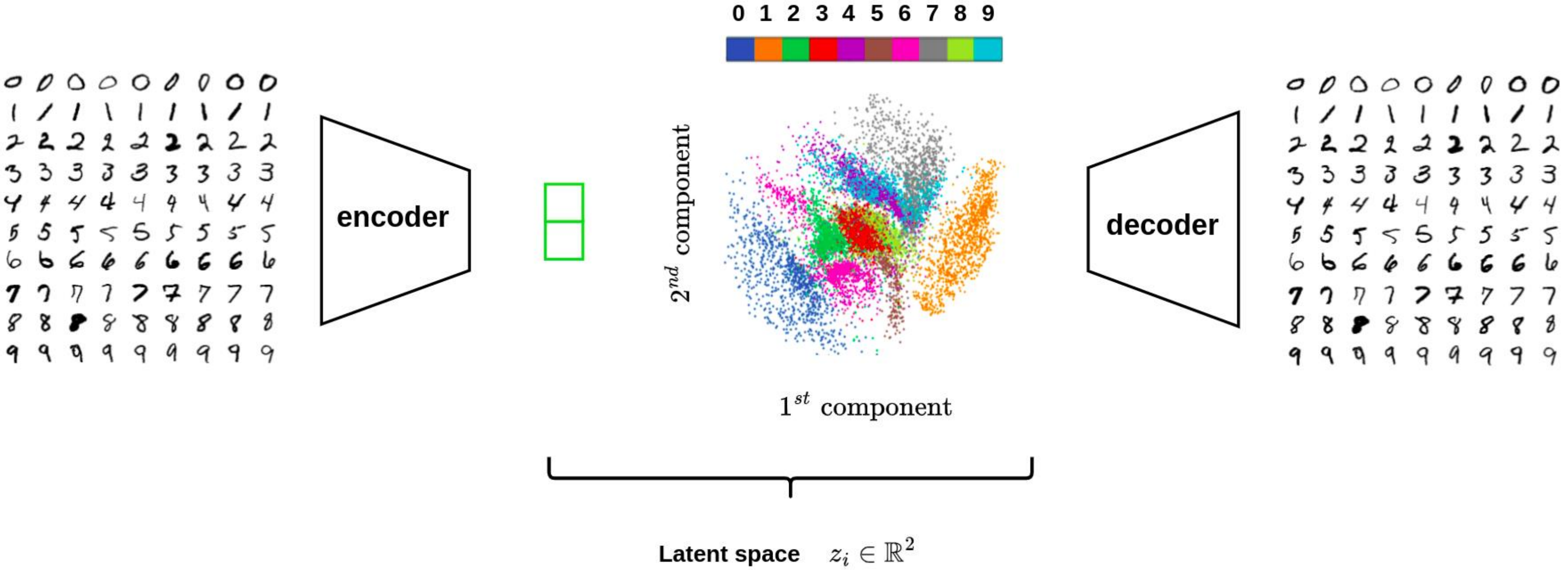
$$\text{loss} = \underbrace{\alpha\|x - f(z)\|^2}_{\textit{Data attachment}\text{ term}} + \underbrace{D_{KL}}_{\textit{Completeness}\text{ constraint}}\underbrace{\left(\mathcal{N}\left(g(x), diag\left(h(x)\right)\right)\right.}_{\textit{Continuity}\text{ constraint}}, \mathcal{N}\left(0, I\right))$$



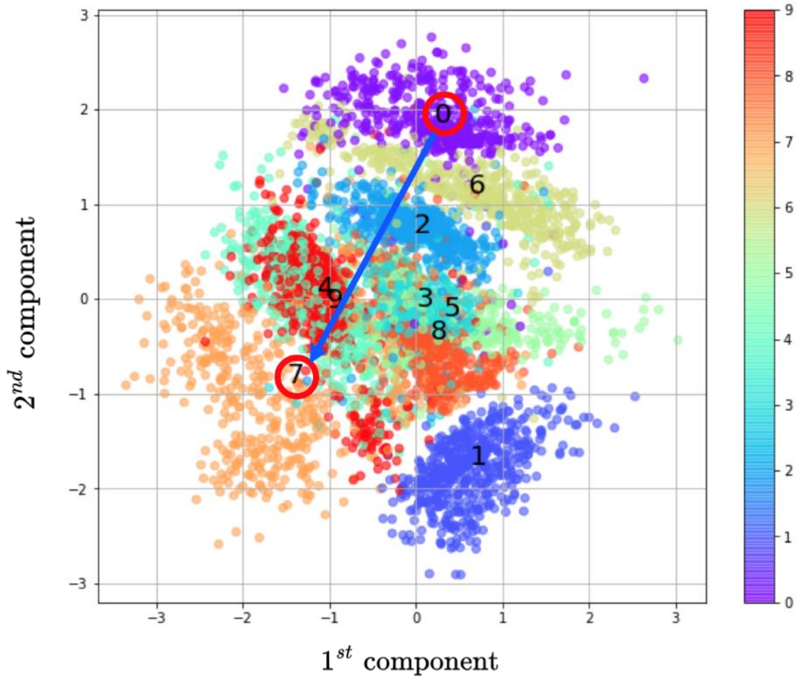
- Reparameterization trick



- Illustration from MNIST dataset
  - (train,valid,test) = (50 000,10 000,10 000)
  - Input image size: 28x28 / latent space K=16 (compression factor around 50)

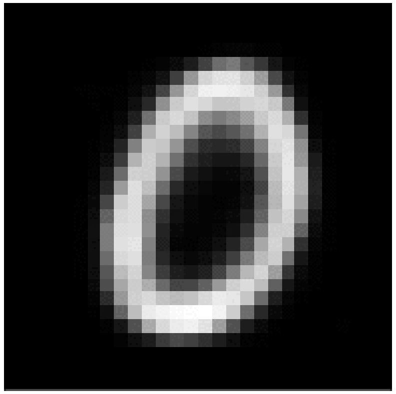


- Generative model with variational framework



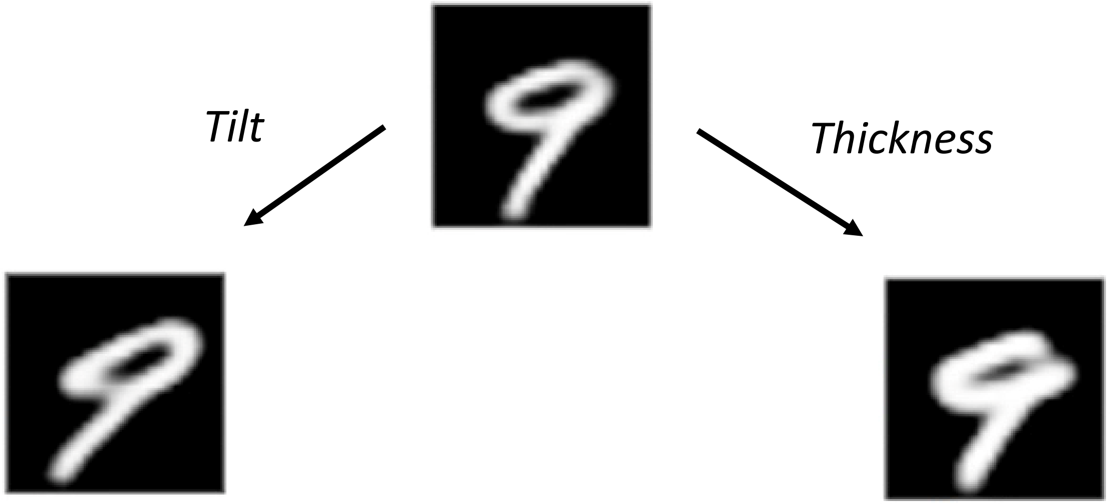
Linear interpolation into the latent space

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



Reinforcement of the generative process

- Structuration of latent space based on image attributes
  - 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



- Structuration of latent space based on image attributes
  - Each attribute are coded according to a specific latent dimension

$$Loss = \underbrace{\mathcal{L}_{recons}}_{\text{Data attachment term}} + \underbrace{\underbrace{\beta \mathcal{L}_{KLD}}_{\text{KL divergence term}} + \underbrace{\gamma \sum_{l=0}^{L-1} \mathcal{L}_{r_l, a_l}}_{\text{Attribute regularization term}}}_{\text{Structuring latent space}}$$

Structuration of the latent space: AR-VAE

- 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

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

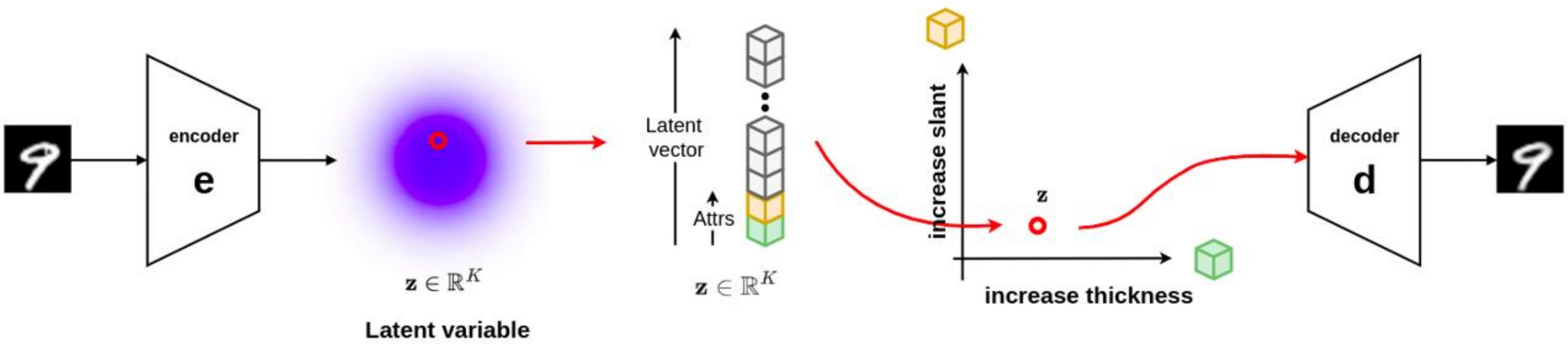
→ Computation for each attribute r of a distance matrix $D_r \in \mathbb{R}^{m\times m}$ from the m latent vector $\{z_i\}_{1\leq i\leq m}$ corresponding to the images in the current batch

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

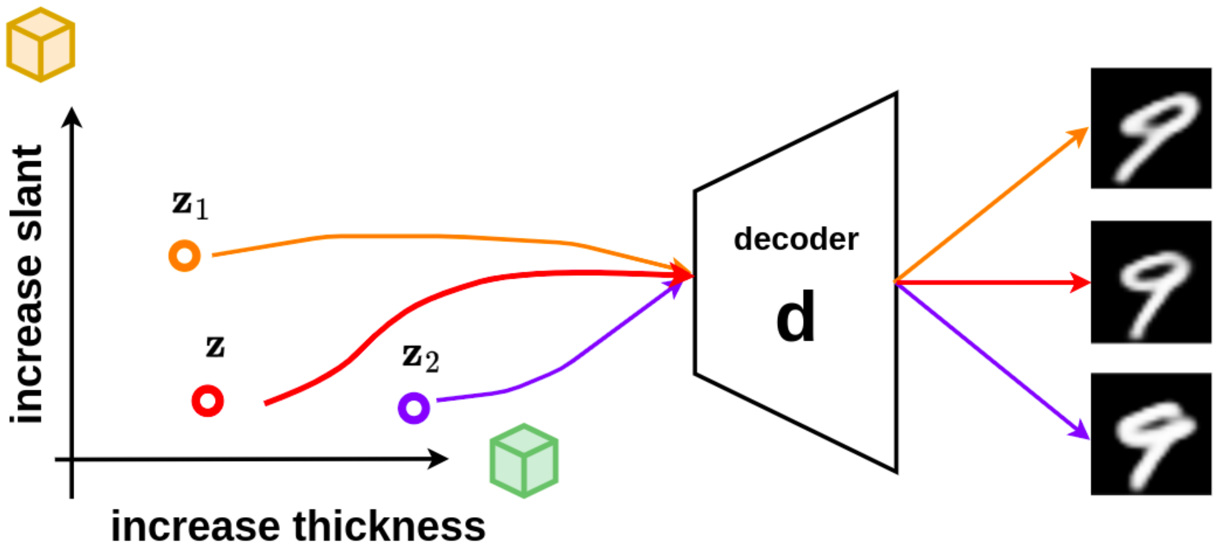
→ Introduction of the following loss term

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

- Generate a latent space structured according to attributes

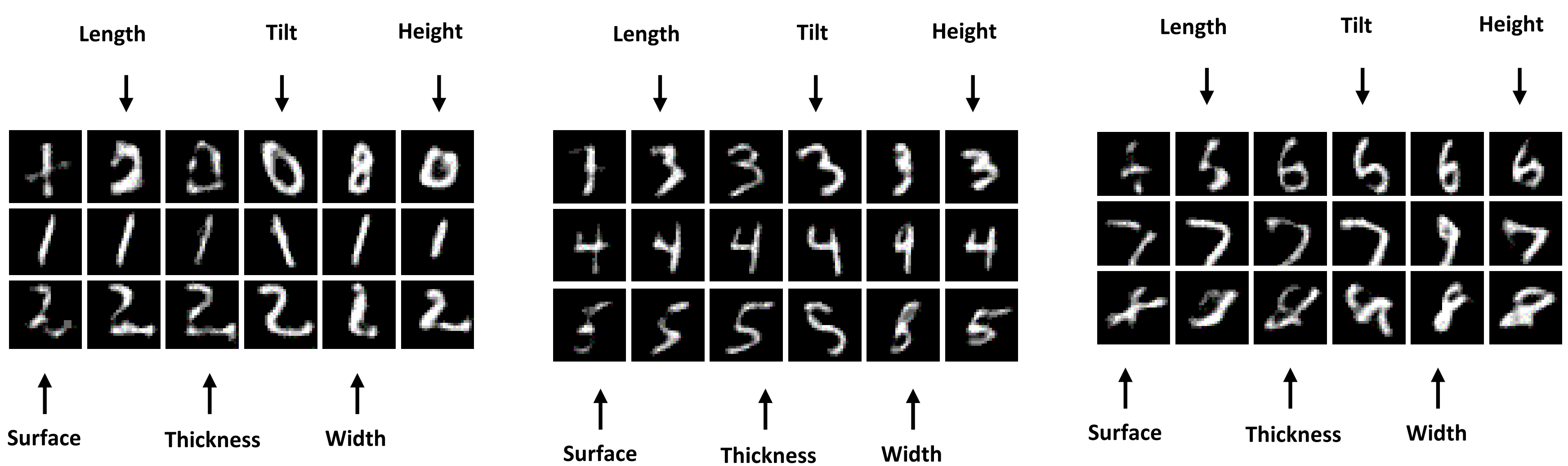


- Generate a latent space structured according to attributes
  - Sampling of the structured latent space



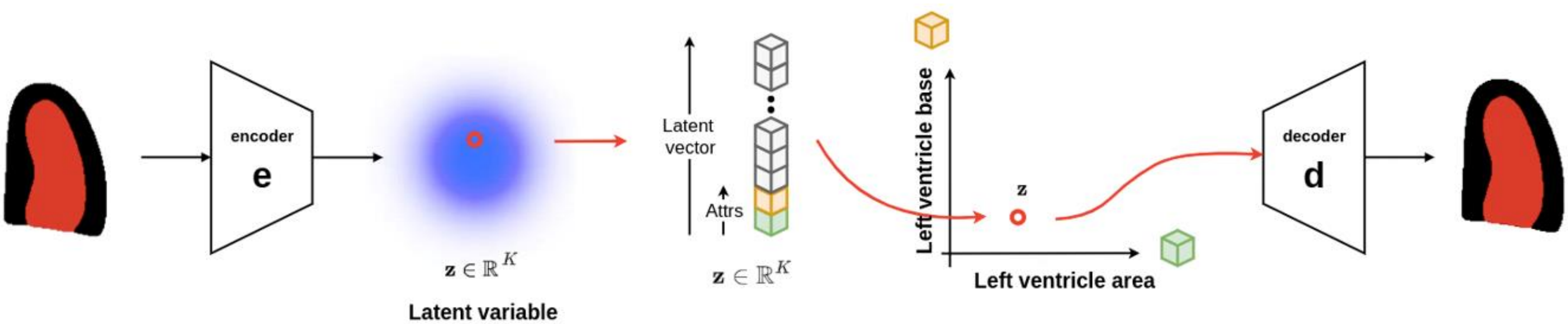
Structuration of the latent space: AR-VAE

- 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

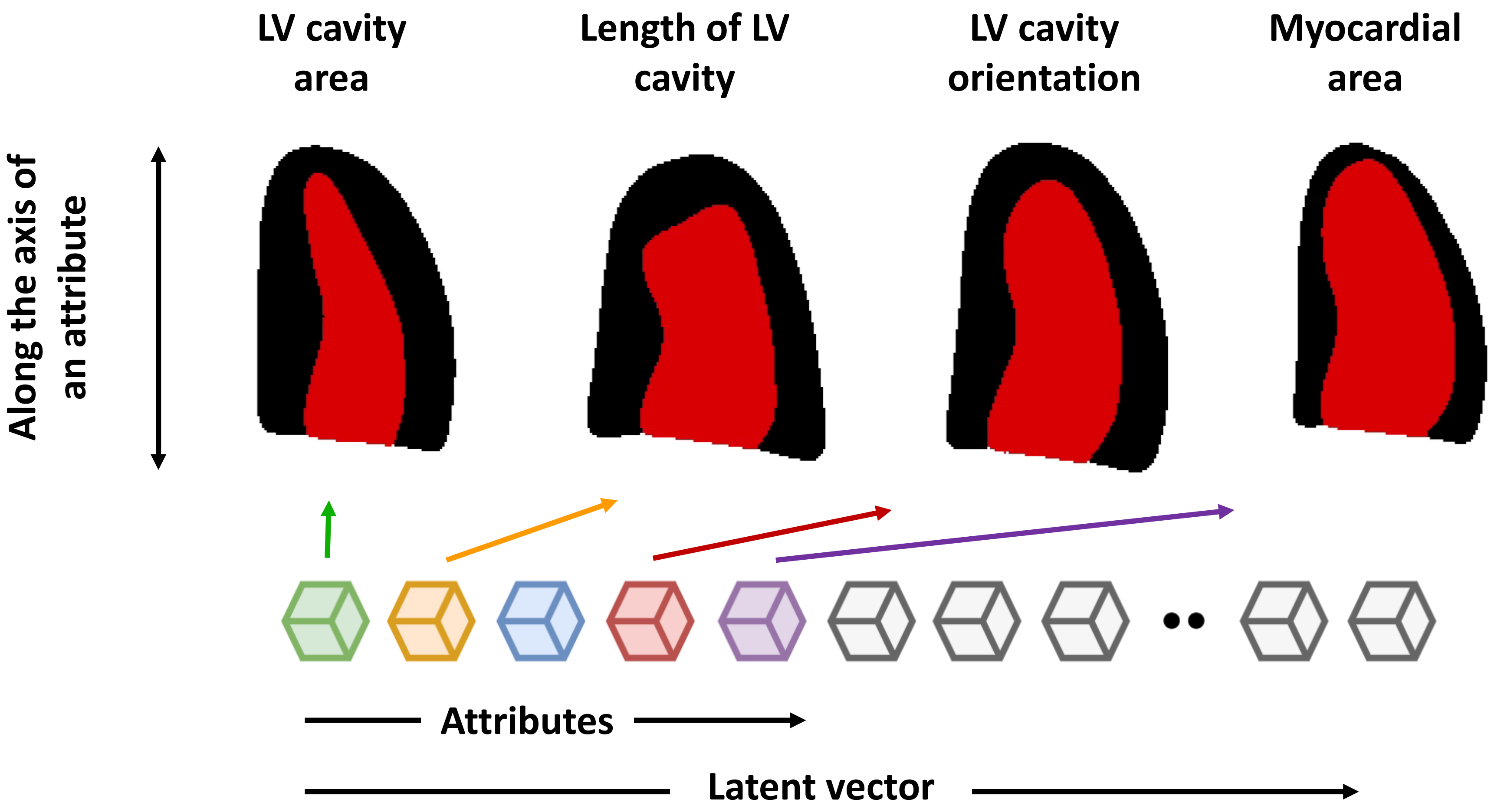


Medical applications

- 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

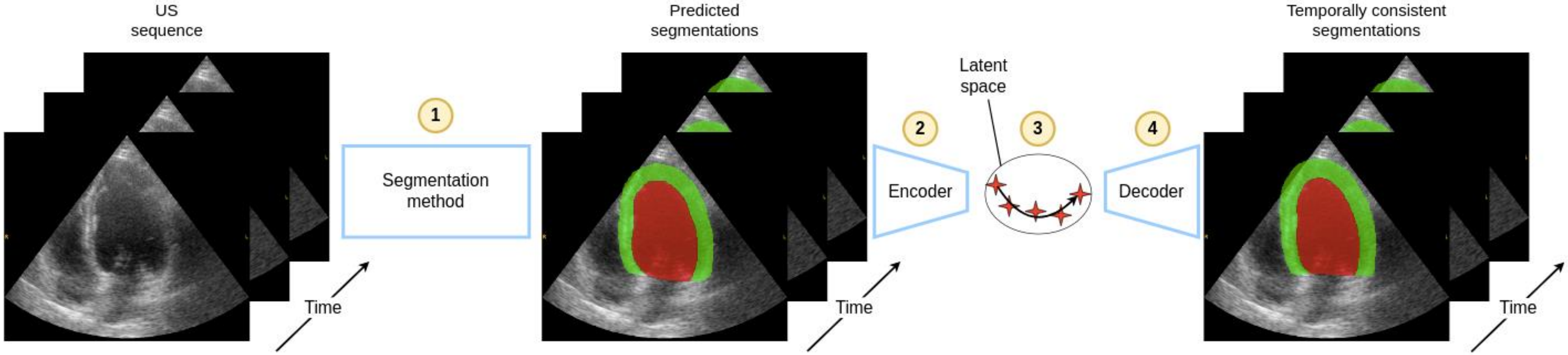


Shape representation



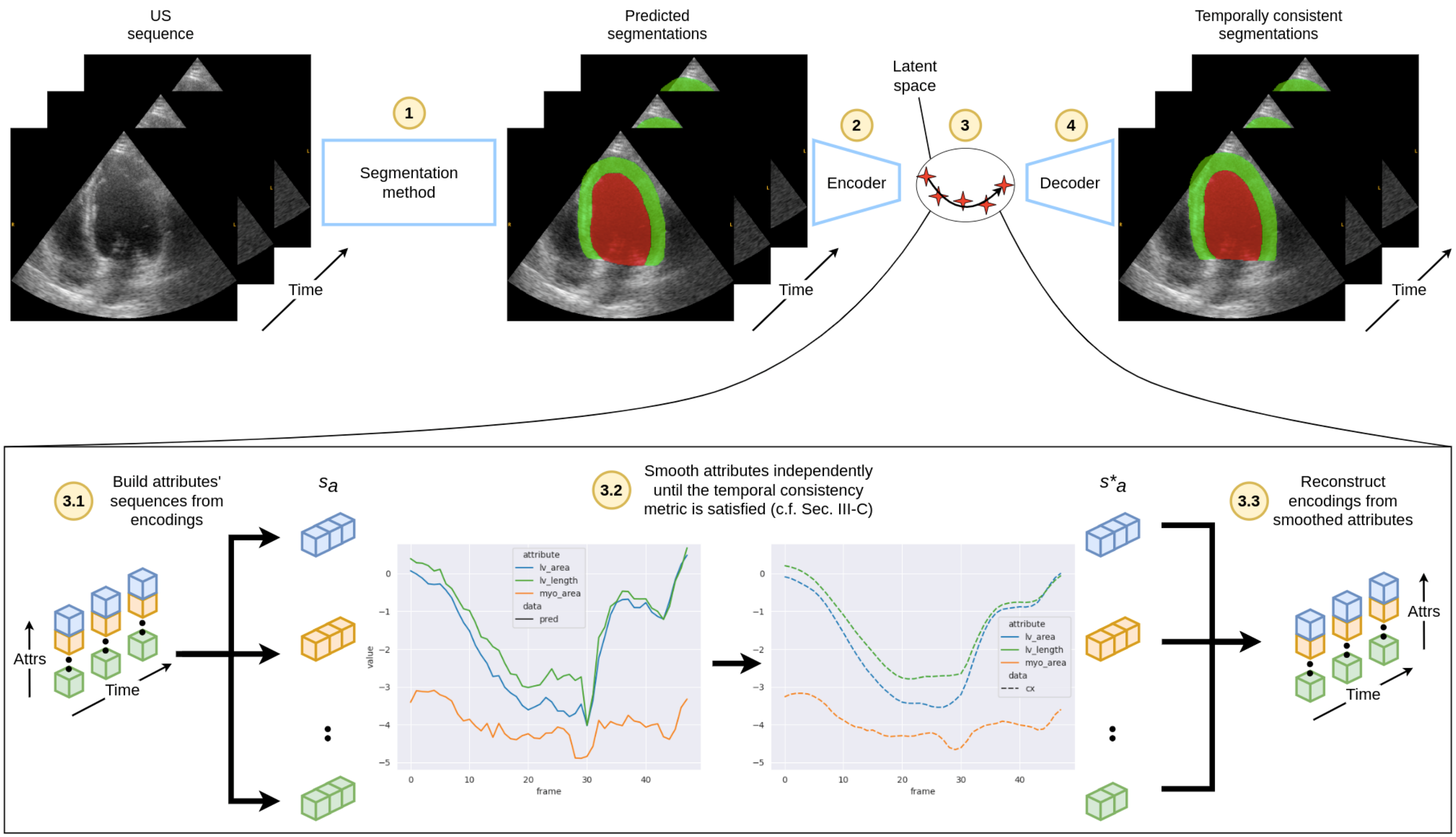
Cardiac segmentation with temporal consistency

- Post-processing to ensure temporal consistency



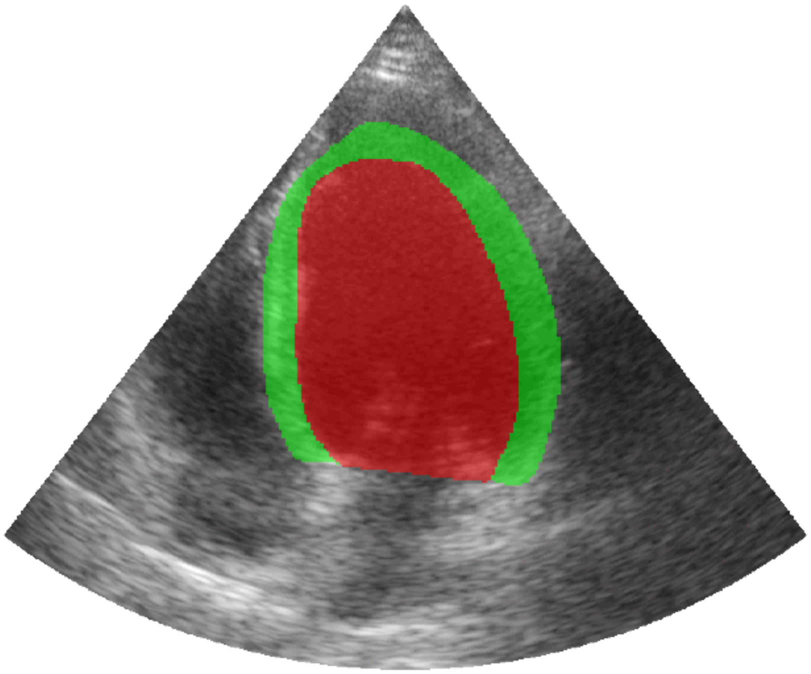
[Painchaud, IEEE TMI, 2022]

Cardiac segmentation with temporal consistency

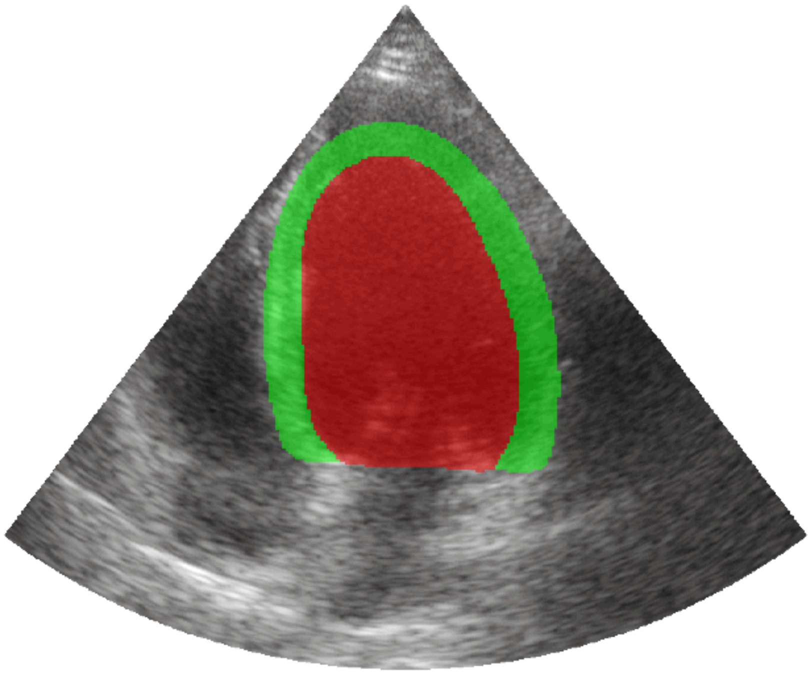


- Some post-processing examples

Original segmentation

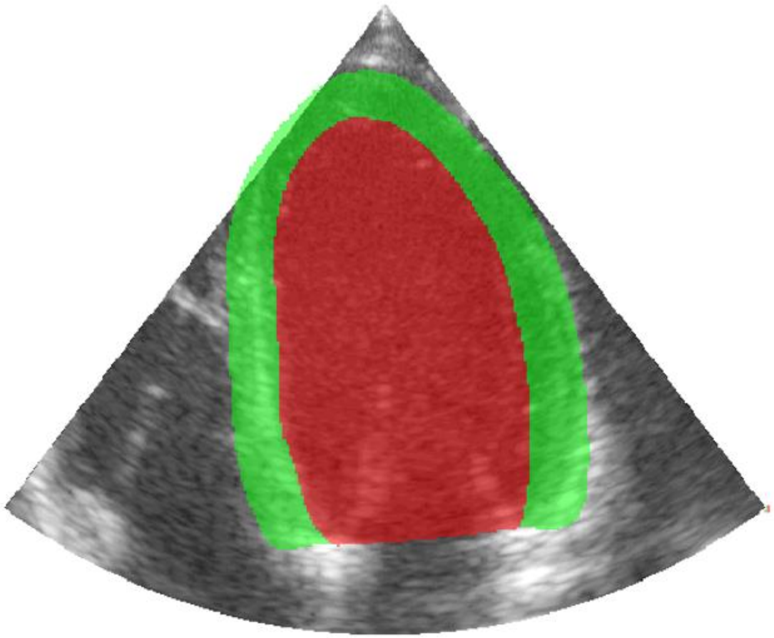


Post-processed segmentation

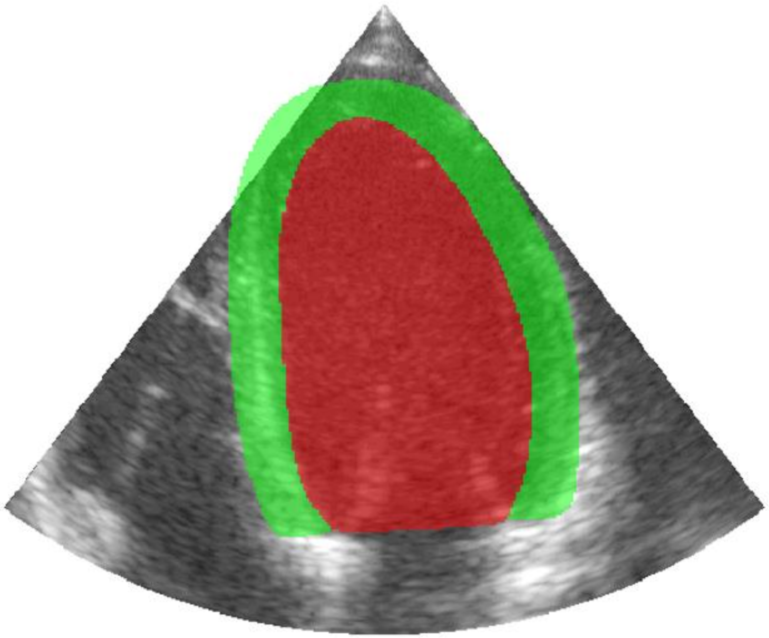


- Some post-processing examples

Original segmentation



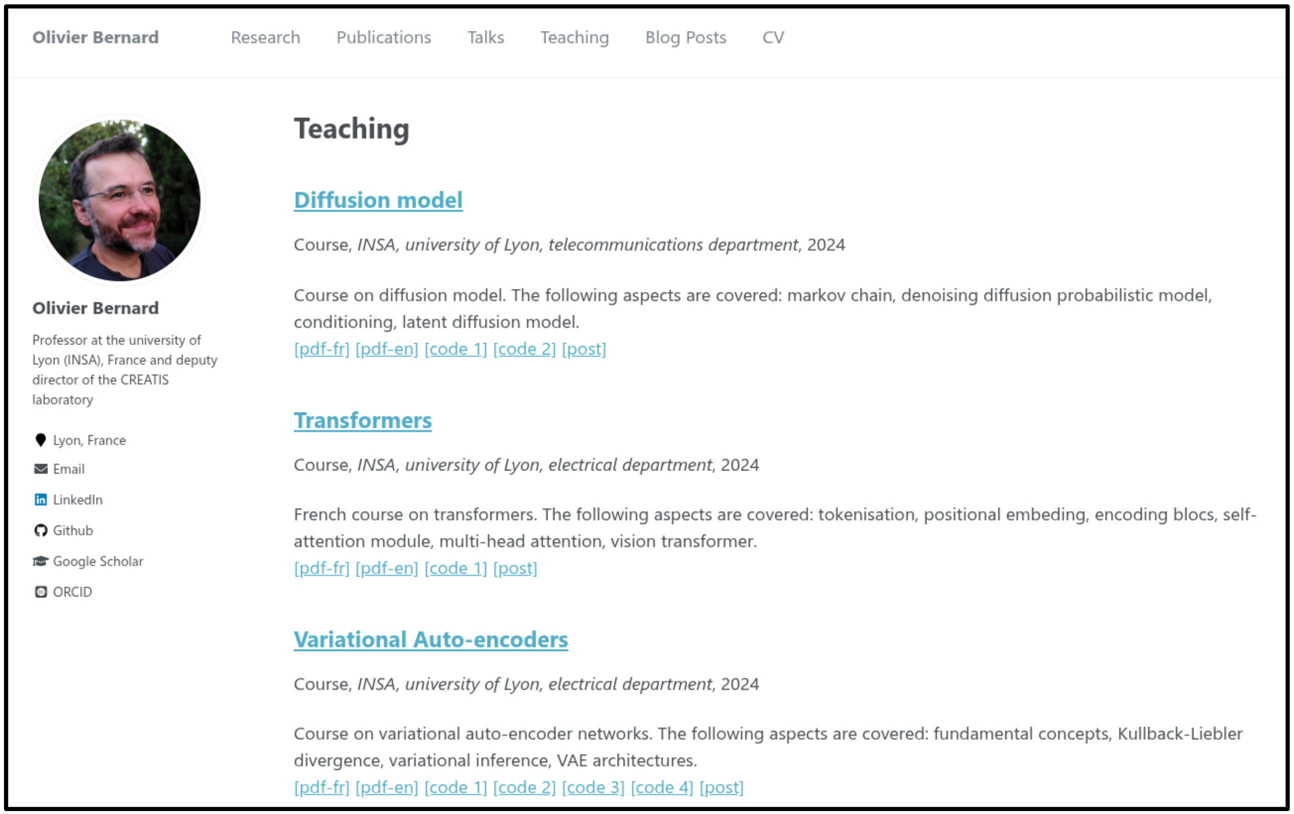
Post-processed segmentation



Hands-on session

Hands-on session

- https://olivier-bernard-creatis.github.io//teaching/


Olivier Bernard   Research   Publications   Talks   Teaching   Blog Posts   CV

Olivier Bernard

Professor at the university of Lyon (INSA), France and deputy director of the CREATIS laboratory

Lyon, France
Email
LinkedIn
Github
Google Scholar
ORCID

Teaching

Diffusion model

Course, *INSA, university of Lyon, telecommunications department*, 2024

Course on diffusion model. The following aspects are covered: markov chain, denoising diffusion probabilistic model, conditioning, latent diffusion model.
[pdf-fr] [pdf-en] [code 1] [code 2] [post]

Transformers

Course, *INSA, university of Lyon, electrical department*, 2024

French course on transformers. The following aspects are covered: tokenisation, positional embeding, encoding blocs, self-attention module, multi-head attention, vision transformer.
[pdf-fr] [pdf-en] [code 1] [post]

Variational Auto-encoders

Course, *INSA, university of Lyon, electrical department*, 2024

Course on variational auto-encoder networks. The following aspects are covered: fundamental concepts, Kullback-Liebler divergence, variational inference, VAE architectures.
[pdf-fr] [pdf-en] [code 1] [code 2] [code 3] [code 4] [post]

Diffusion models

- Best current methods for synthetic image generation
- Allows generating images in a *conditioned* form
- Many software solutions, such as Midjourney, DALL-E

An Asian girl in ancient coarse linen clothes rides a giant panda and carries a wooden cage. A chubby little girl with two buns walks on the snow. High-precision clothing texture, real tactile skin, foggy white tone, low saturation, retro film texture, tranquil atmosphere, minimalism, long-range view, telephoto lens

What is the purpose of diffusion models?

- Family of diffusion networks

Denoising Diffusion Probabilistic models

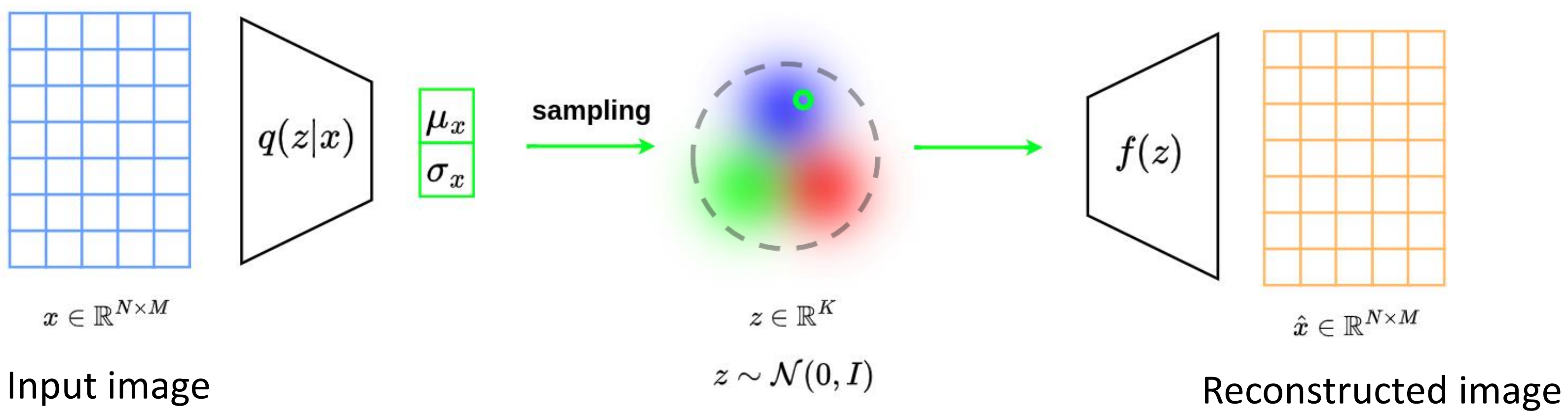
Score-based methods

Normalizing flow methods

Intuition behind diffusion models

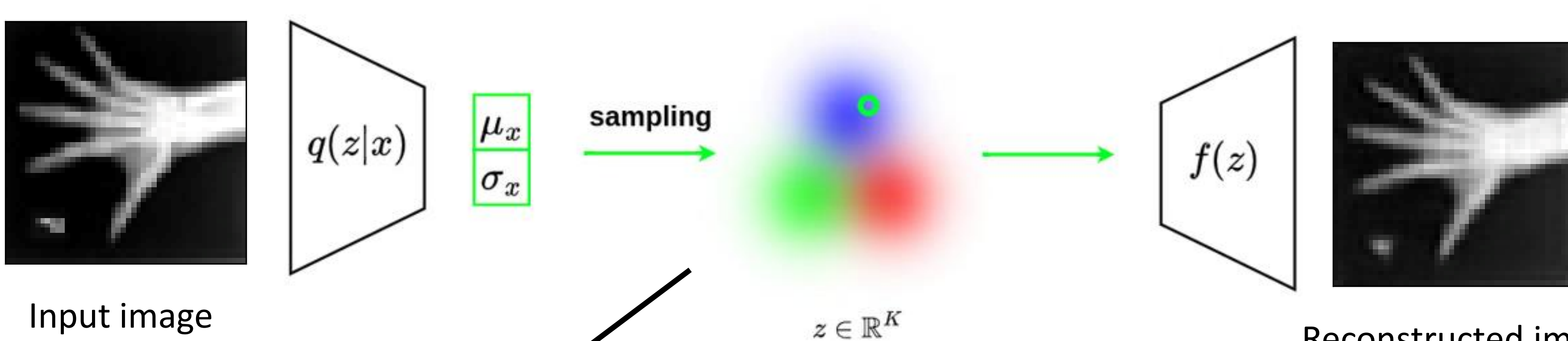
- Completeness is expressed as a ***soft constraint*** !

➔ $\mathcal{N}(g(x), diag(h(x)))$ and $\mathcal{N}(0, I)$ should remain close in terms of distributional distance



Sampling from the latent space $\mathcal{N}(0, I)$ does not guarantee to obtain a reconstructed image from the target distribution

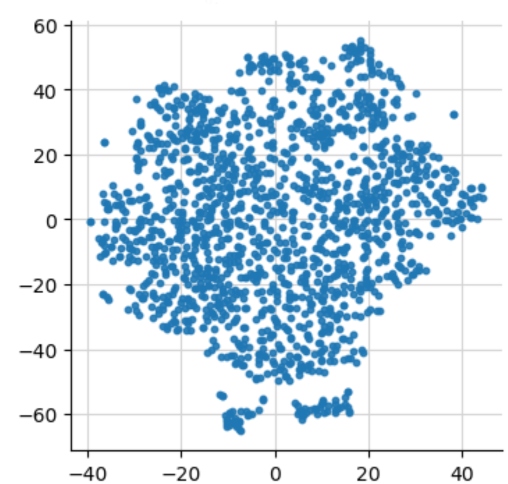
- Illustration from Mednist dataset
  - (train,valid,test) = (1491,373,223)
  - Input image size: 48x48 / latent space K=432 (compression factor around 5)



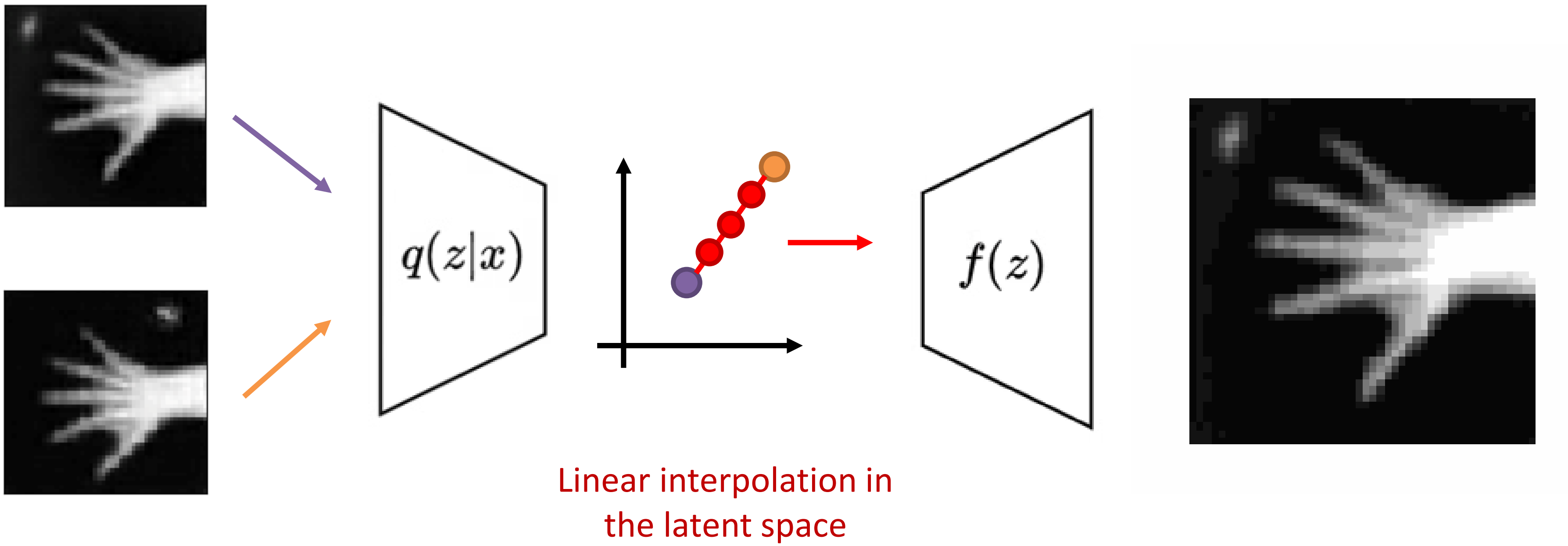
Input image

Reconstructed image

t-SNE visualization of the latent space using the training data as sample points

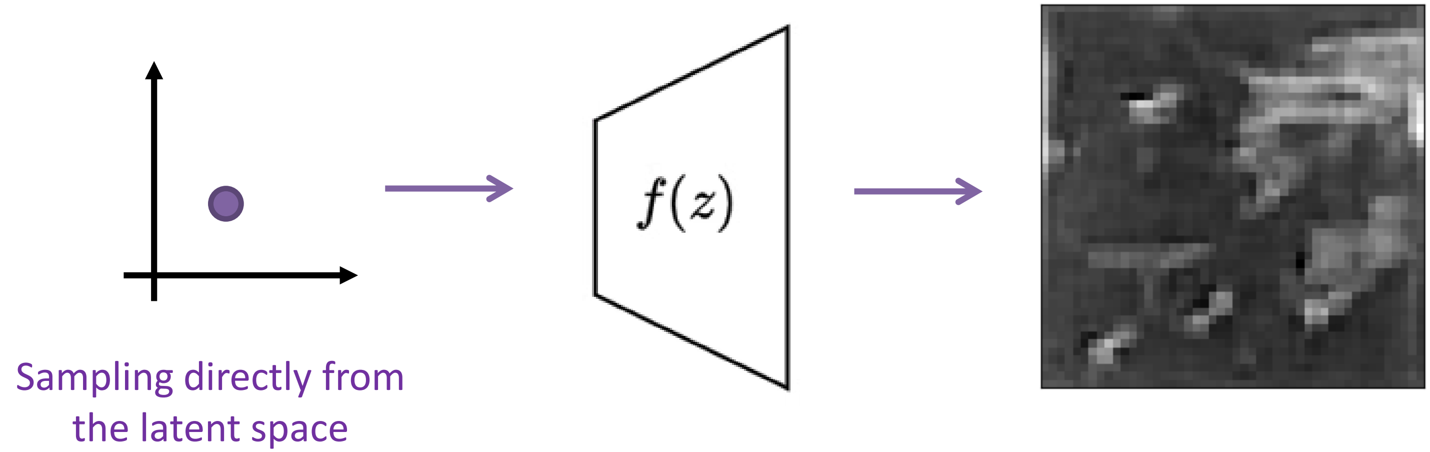


- Linear interpolation between two real images



- Sampling directly from the latent space

$z \in \mathbb{R}^{(K)}$ with $z_i \sim \mathcal{N}(0, I)$



A soft constraint on the latent space to remain close to $\mathcal{N}(0, I)$ is not sufficient to build generative models that effectively learn a target distribution

The denoising diffusion probabilistic models

DDPM

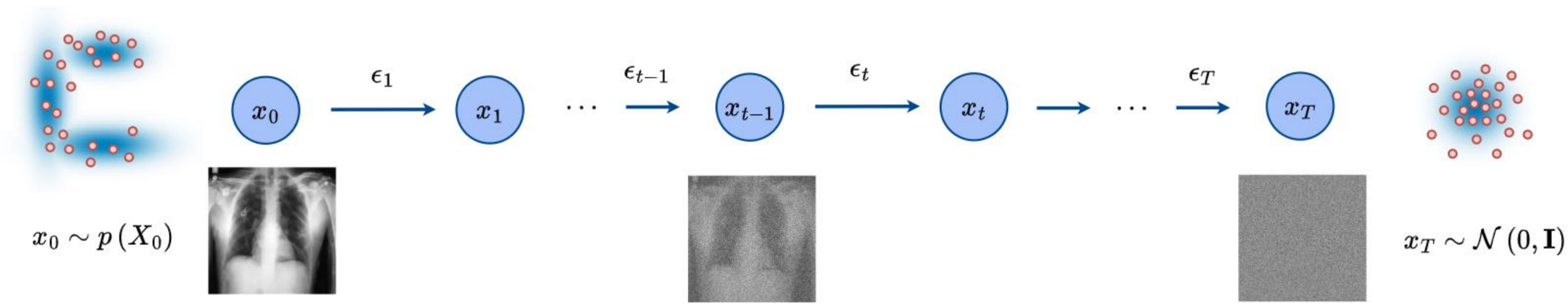
All the mathematics are described in the following blog

https://creatis-myriad.github.io/tutorials/2023-11-30-tutorial-ddpm.html

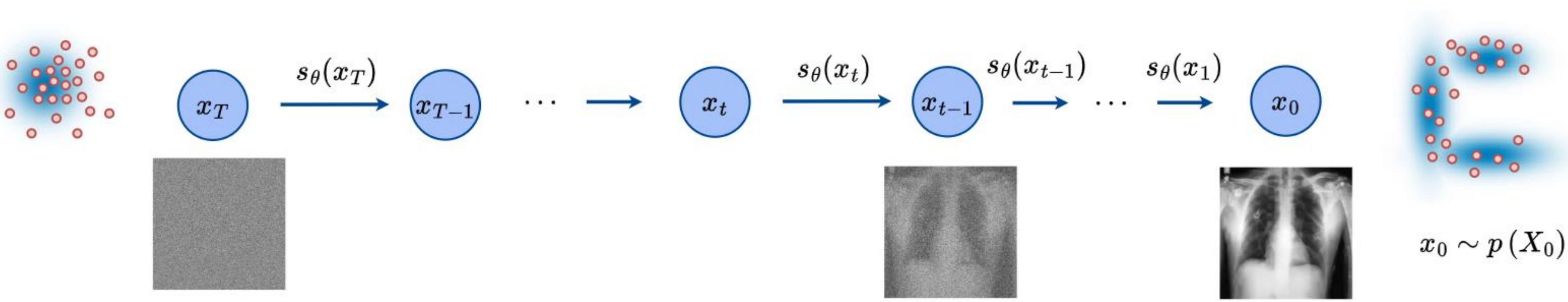
Basic idea of denoising diffusion model

How can a hard constraint be enforced to ensure a direct transformation from the latent space (modeled as a Gaussian) to the target distribution?

- Noising process



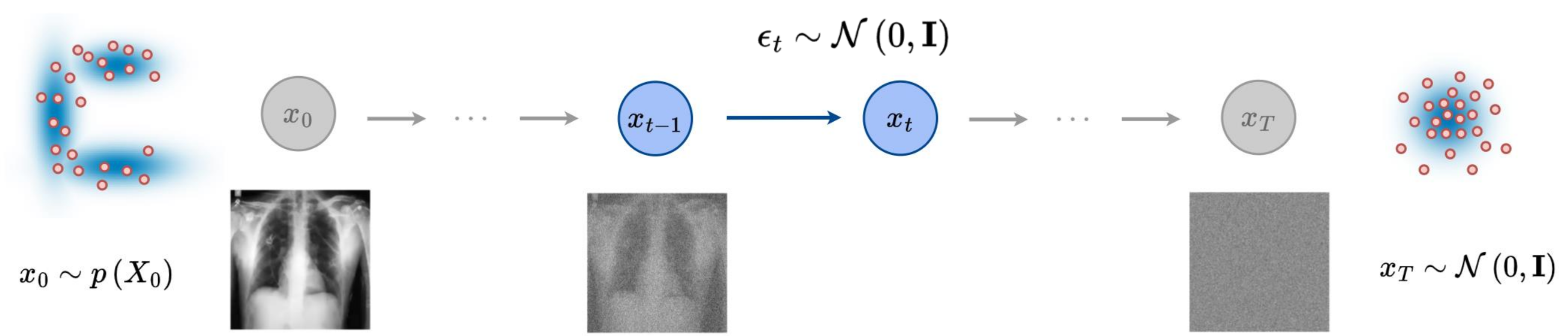
- Learning of the denoising process



Noising process (forward diffusion process)

- Modeled as a sequence of normal distributions (Markov chain process)

$$q\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(\left(\sqrt{1-\beta_t}\right) x_{t-1}, \beta_t \, \mathbf{I}\right)$$

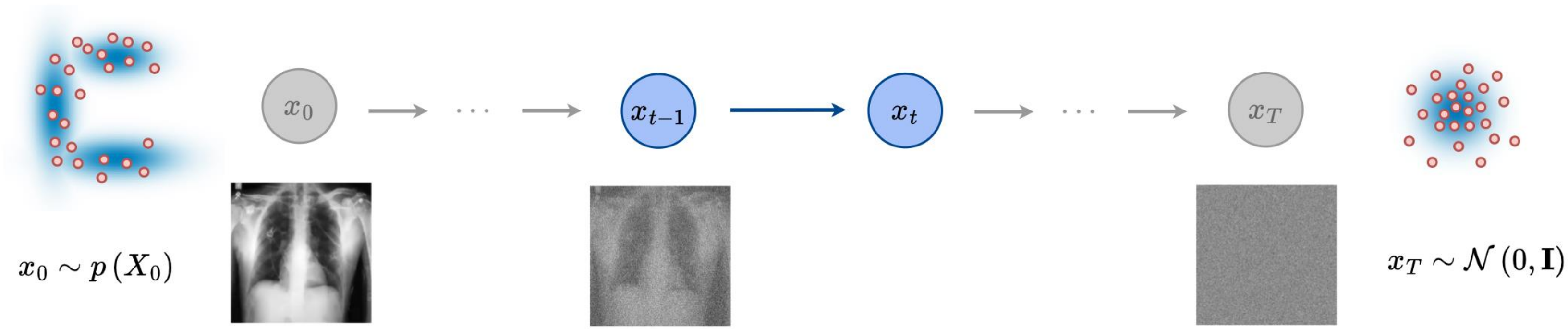


$$q\left(x_t \mid x_{t-1}\right) = \left(\sqrt{1-\beta_t}\right) x_{t-1} + \beta_t \, \mathbf{I} \, . \, \epsilon_t$$

Noising process (forward diffusion process)

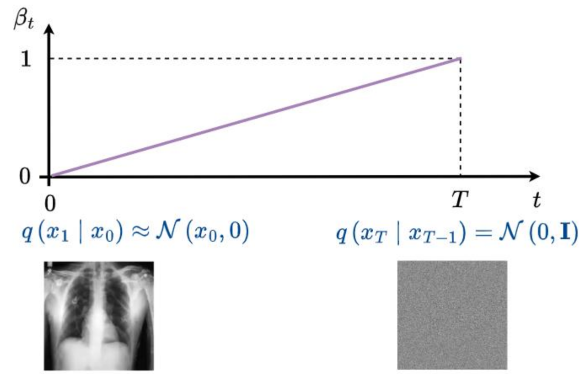
- Modeled as a sequence of normal distributions (Markov chain process)

$$q\left(x_t \mid x_{t-1}\right) = \mathcal{N}\left(\left(\sqrt{1-\beta_t}\right) x_{t-1}, \beta_t \mathbf{I}\right)$$



- β_t : variance varying over the iterative process from 0 to 1

$$\begin{aligned} &\text{if} \quad \beta_t = 0, \quad \text{then} \quad q(x_t \mid x_{t-1}) = x_{t-1} \\ &\text{if} \quad \beta_t = 1, \quad \text{then} \quad q(x_t \mid x_{t-1}) = \mathcal{N}(0, \mathbf{I}) \end{aligned}$$

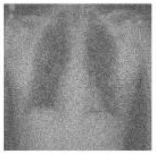


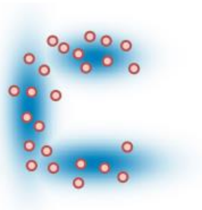
Noising / denoising processes

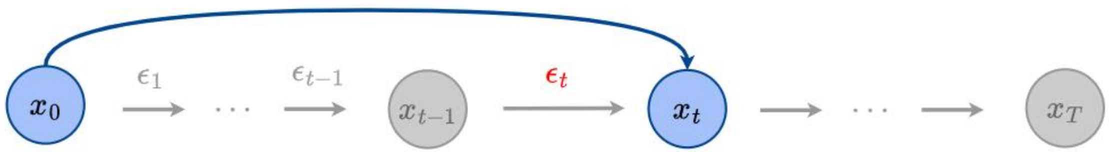
- Noising process

$$\begin{cases} \alpha_t = 1 - \beta_t \\ \bar{\alpha}_t = \prod_{k=1}^{t} \alpha_k \\ \epsilon_t \sim \mathcal{N}(0, \mathbf{I}) \end{cases}$$

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon_t$$

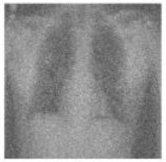


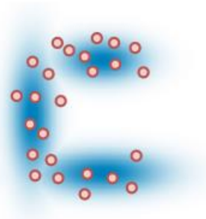


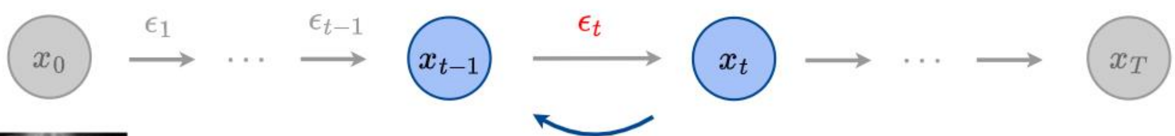


$x_0 \sim p(X_0)$

- Denoising process



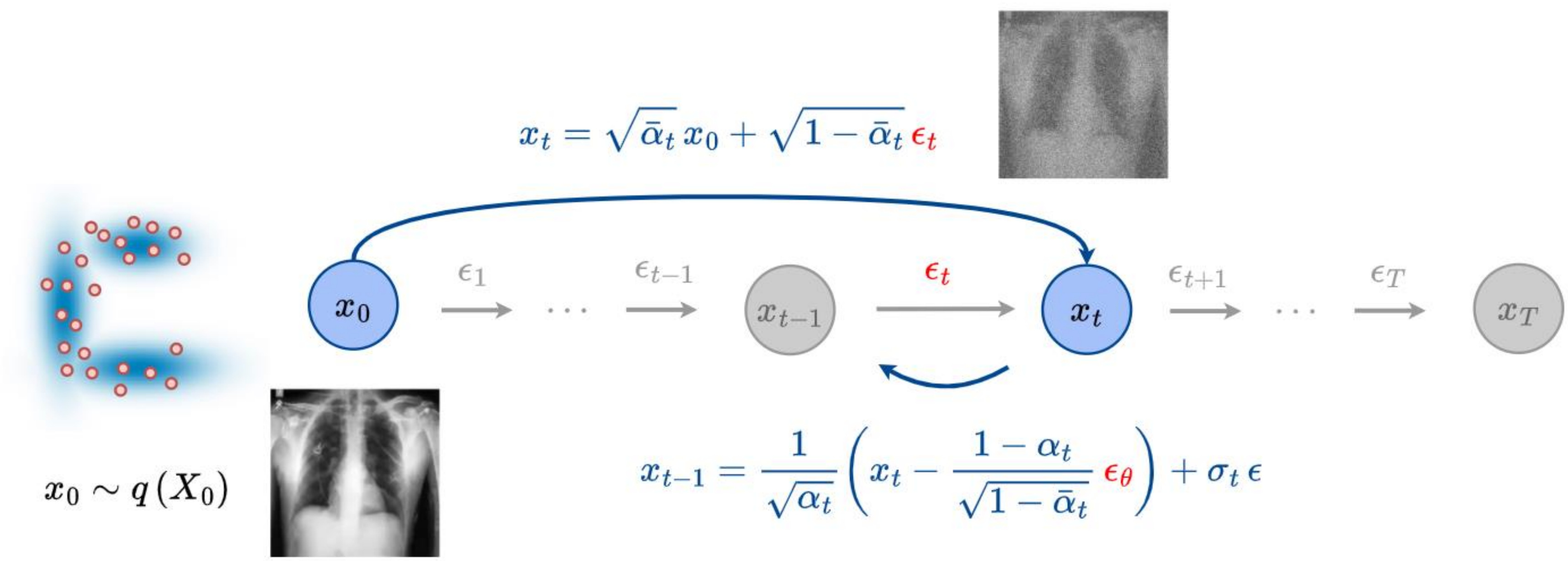




$x_0 \sim p(X_0)$

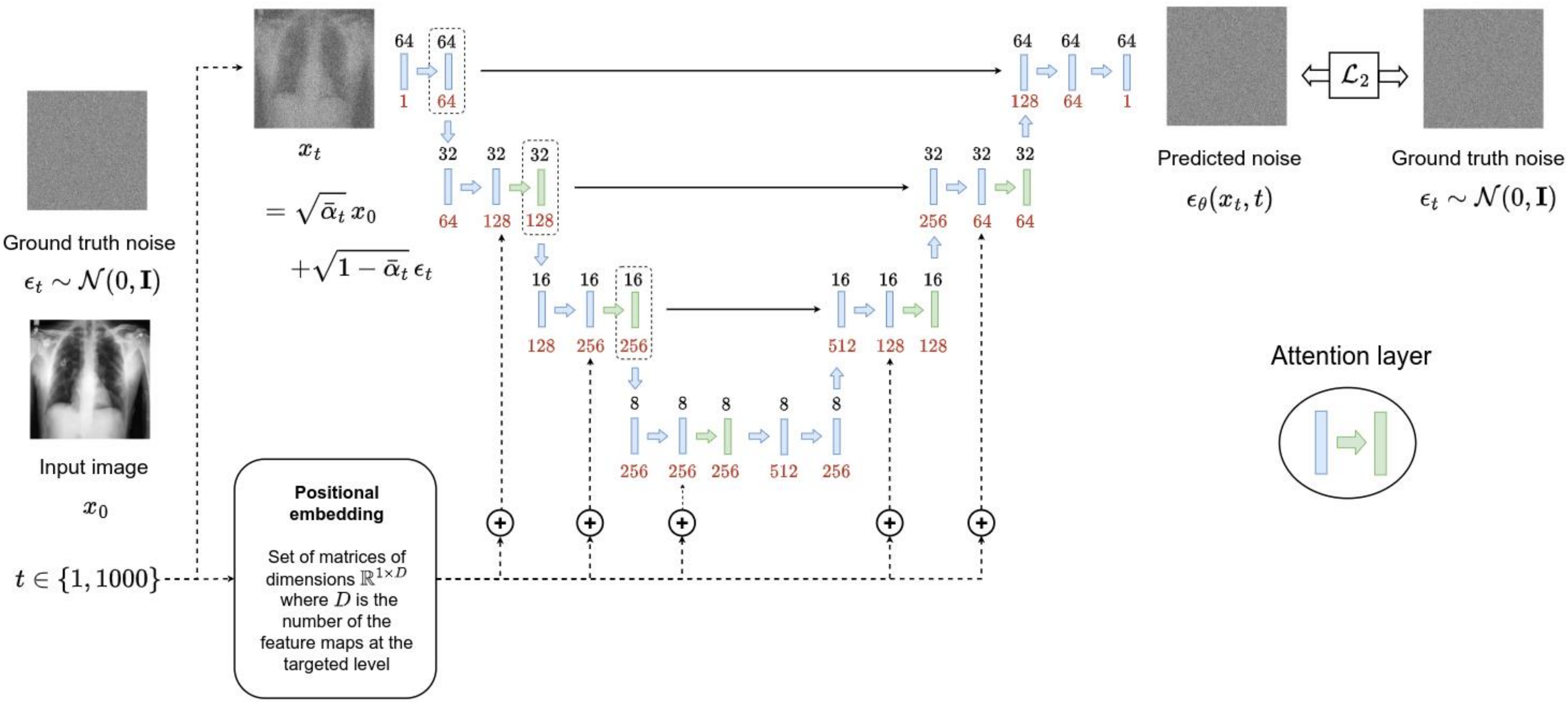
$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\, \epsilon_t \right) + \sigma_t\, \epsilon$$

- Choose a random step $t \in \{1, \cdots, T\}$
- Train a U-Net model to predict the noise pattern ε_θ to remove from x_t

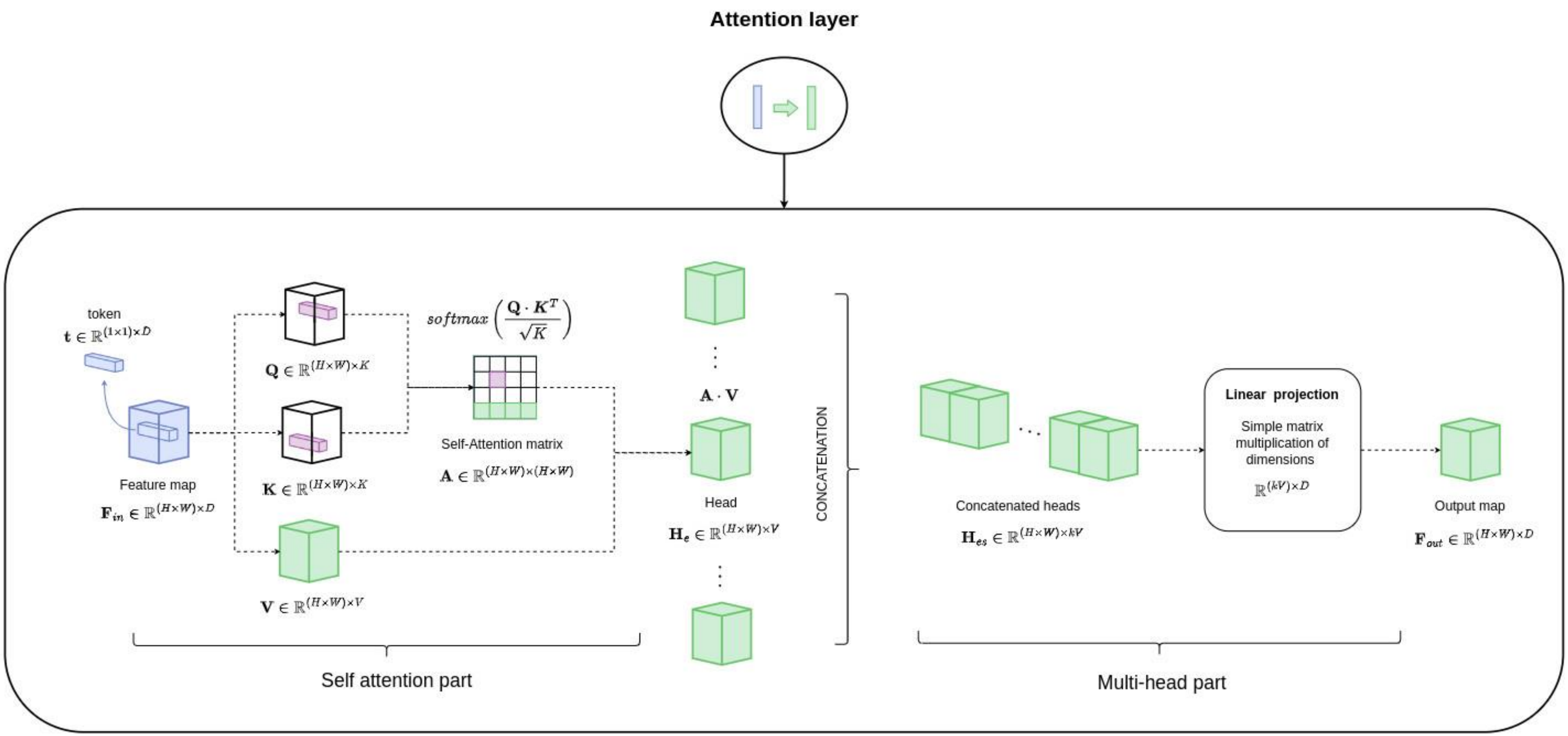


Standard U-Net with attention layers and position encoding to integrate temporal information

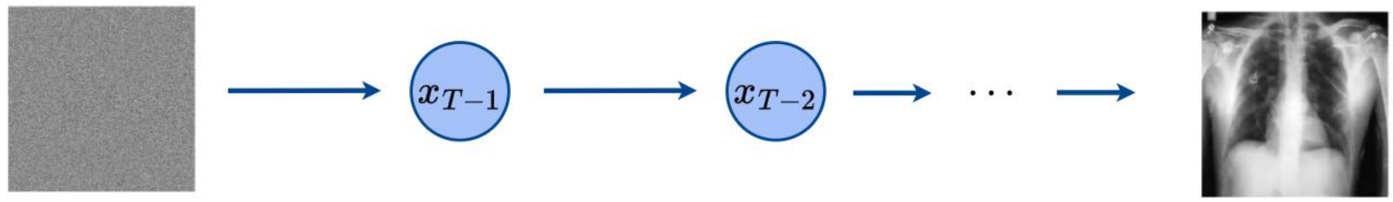
➔ Integration of t is necessary because the added noise varies over time



➔ Attention layer



Inference: generation of synthetic data



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

x_0

- Generate a random image $x_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{N \times M}$
- At each step from T to 0, use the U-Net model to compute

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t \epsilon$$

U-Net

with $\quad \epsilon \sim \mathcal{N}(0, \mathbf{I})$

Mathematical formalism

Mathematical formulation

- Useful notations

$$q(x_1, \cdots, x_T \mid x_0) = q(x_1 \mid x_0)\, q(x_2 \mid x_1, x_0) \cdots q(x_T \mid x_{T-1}, \cdots, x_0)$$

Complete forward process

$$q(x_1, \cdots, x_T \mid x_0) = q(x_1 \mid x_0)\, q(x_2 \mid x_1) \cdots q(x_T \mid x_{T-1})$$

Markov chain

$$q(x_{1:T} \mid x_0) = q(x_1 \mid x_0)\, q(x_2 \mid x_1) \cdots q(x_T \mid x_{T-1})$$

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})$$

Compact reformulation

$$\boxed{q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})}$$

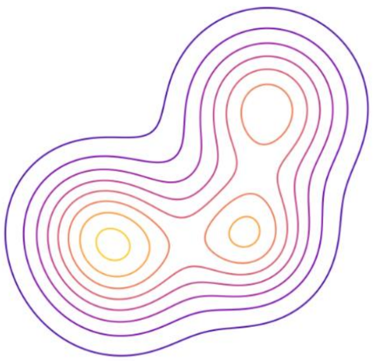
Complete forward process

$$\boxed{p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p(x_{t-1} \mid x_t)}$$

Complete reverse process

Mathematical formulation

- Optimization process
  - → Maximization of $\log\left(p_\theta(x)\right)$ / Minimization of $-\log(p_\theta(x))$



$$\boxed{-\log\left(p_\theta(x)\right)}$$

$$-\log\left(p_\theta(x)\right) = -\log\left(\int p_\theta(x_{0:T})\,dx_{1:T}\right)$$

← Marginal distribution

$$-\log\left(p_\theta(x)\right) = -\log\left(\int \frac{q(x_{1:T} \mid x_0)}{q(x_{1:T} \mid x_0)}\, p_\theta(x_{0:T})\,dx_{1:T}\right)$$

Expectation reformulation

$$-\log\left(p_\theta(x)\right) = -\log\left(\mathbb{E}_{q(x_{1:T}|x_0)}\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right]\right)$$

Jensen's inequality

$$\boxed{-\log\left(p_\theta(x)\right) \leq -\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log\left(\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right)\right]}$$

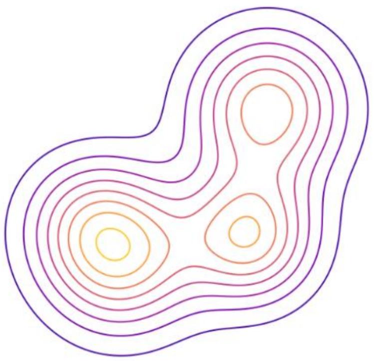
Mathematical formulation

- Evidence lower bound (ELBO)

$$-\log\left(p_\theta(x)\right) \leq \boxed{-\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log\left(\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right)\right]}$$

ELBO

$p\left(x\right)$



→ Minimization of the ELBO

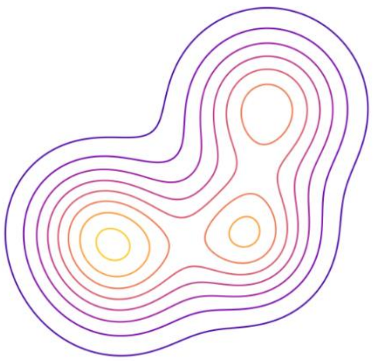
$$-\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log\left(\frac{p_\theta(x_{0:T})}{q(x_{1:T} \mid x_0)}\right)\right]$$

$$\vdots$$

$$\boxed{-\mathbb{E}_q\left[\underbrace{D_{KL}\left(q(x_T \mid x_0) \,\|\, p_\theta(x_T)\right)}_{\text{No parameter to be learned}} + \sum_{t>1} D_{KL}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\right) - \underbrace{\log\left(p_\theta(x_0 \mid x_1)\right)}_{\text{Very small}}\right]}$$

- ELBO minimization

$$\mathcal{L} = -\mathbb{E}_q \left[\sum_{t>1} D_{KL}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\right) \right]$$



→ Exploitation of the Gaussian properties of the forward process and modeling of the reverse process using Gaussian distribution

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(\tilde{\mu}_t, \tilde{\beta}_t\right) \qquad p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(\mu_\theta, \sigma_\theta\right)$$

→ Reformulation

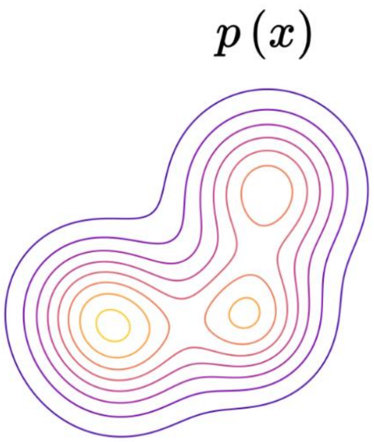
$$D_{KL}\left(q(x_{t-1} \mid x_t, x_0) \,\|\, p_\theta(x_{t-1} \mid x_t)\right) = -\frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t - \mu_\theta \right\|^2$$

$$\mathcal{L} = \mathbb{E}_q \left[\sum_{t>1} \frac{1}{2\sigma_t^2} \left\| \tilde{\mu}_t - \mu_\theta(x_t, t) \right\|^2 \right]$$

Mathematical formulation

- ELBO minimization

$$\mathcal{L} = \mathbb{E}_q \left[\sum_{t>1} \frac{1}{2\sigma_t^2} \|\tilde{\mu}_t - \mu_\theta(x_t, t)\|^2 \right]$$



→ Expressions of means

$$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_t \right) \qquad \mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

$$\mathcal{L} = \mathbb{E}_q \left[\sum_{t>1} \frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \right]$$

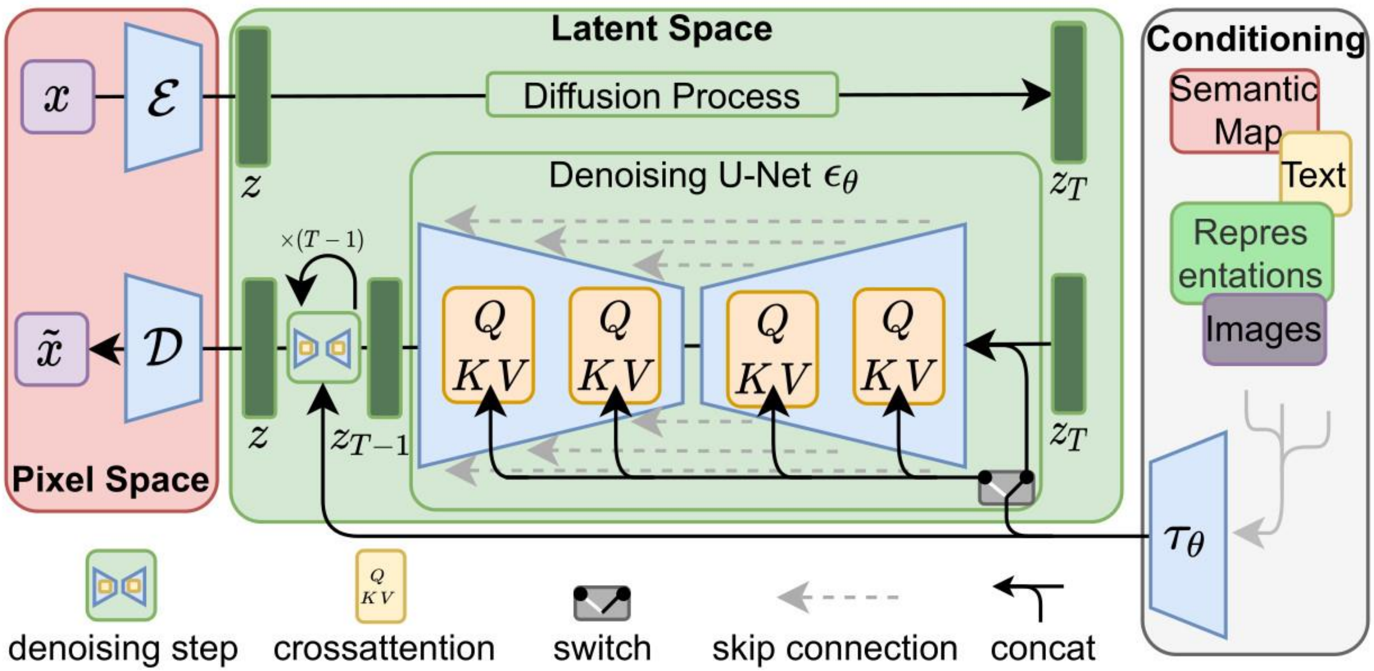
→ Simplifications

$$\mathcal{L} = \mathbb{E}_{q,t} \left[\|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \right]$$

Practical application

Latent diffusion models

- VAE is learned independently of DDPM and its architecture is fixed
  - Efficiently reduce the dimensionality of the input space
  - Efficiently initiate the Gaussian diffusion process
- LDM architecture



- Properties

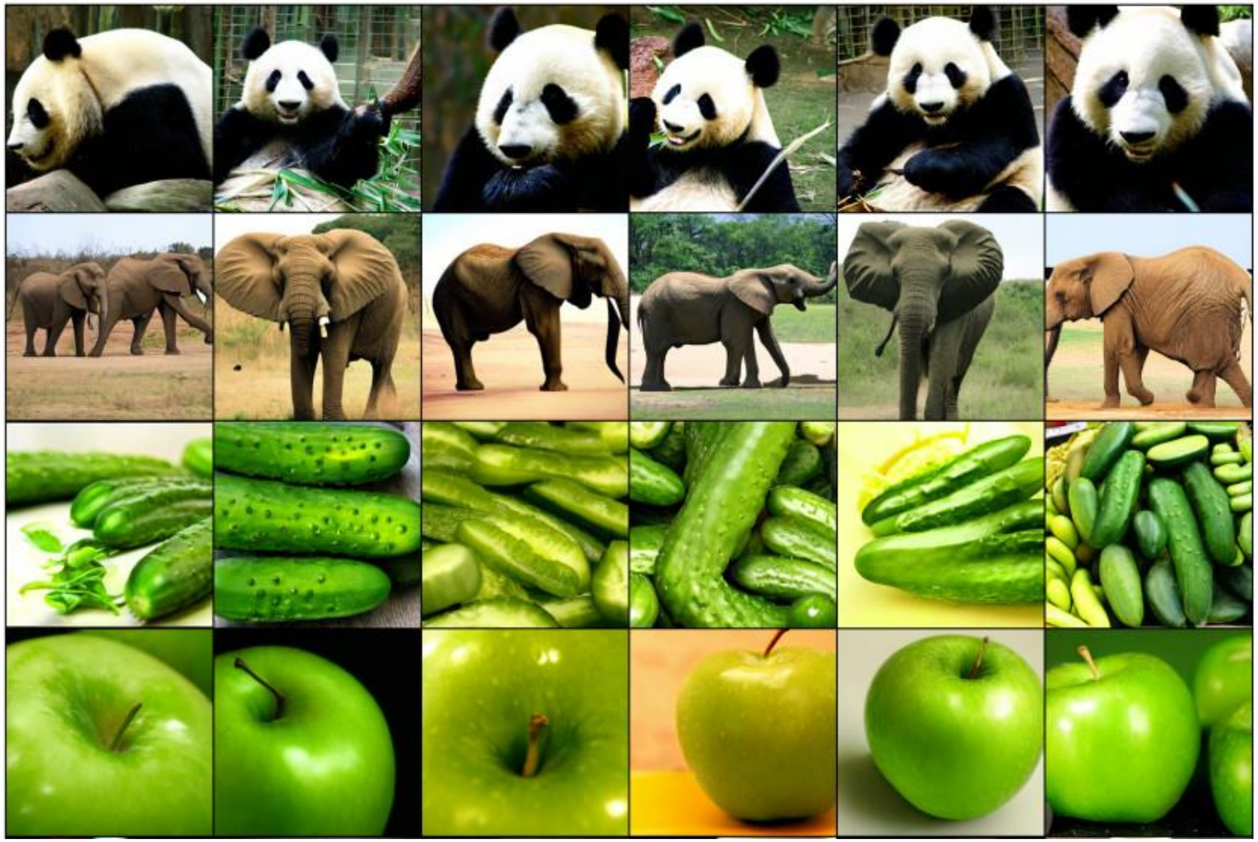
Parameters	LDM – 256×256
z dimensions	$64 \times 64 \times 3$
Diffusion steps	1000
Noise scheduler (β_t)	linear
Number of parameters	274 Million
Channels	224
Channel multiplier	$1, 2, 3, 4$
Levels for attention	$2, 3, 4$
Number of head	1
Batch size	48
Iterations	410 k
Learning rate	$9.6\, e^{-5}$

- Random generation of synthetic images *without conditioning* learned from the CelebA-HQ database

Random samples on the CelebA-HQ dataset

- Random generation of synthetic images *with conditioning on the class* learned from the ImageNet database

Random class conditional samples on the ImageNet dataset



- Random generation of synthetic images *with conditioning on masks* learned from the Flickr-landscapes database

Latent diffusion model (LDM)

- Random generation of synthetic images *with conditioning on text* learned from LAION-400M database

  → Using the BERT tokenizer

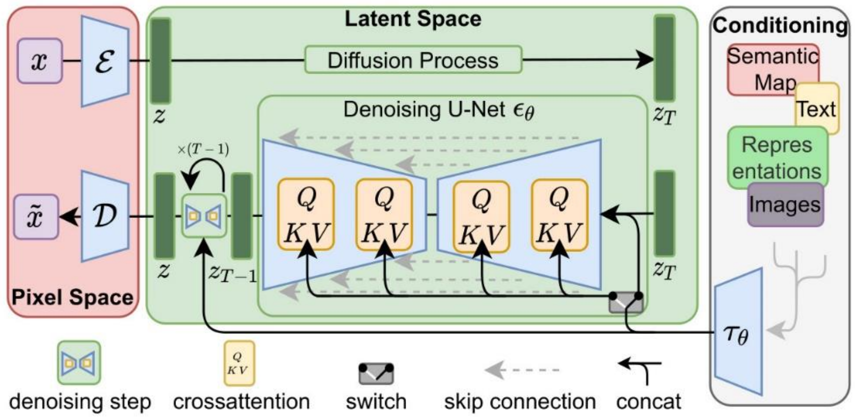
  → This model has over 1.45 billion parameters!

'A painting of the last supper by Picasso.'

Medical applications

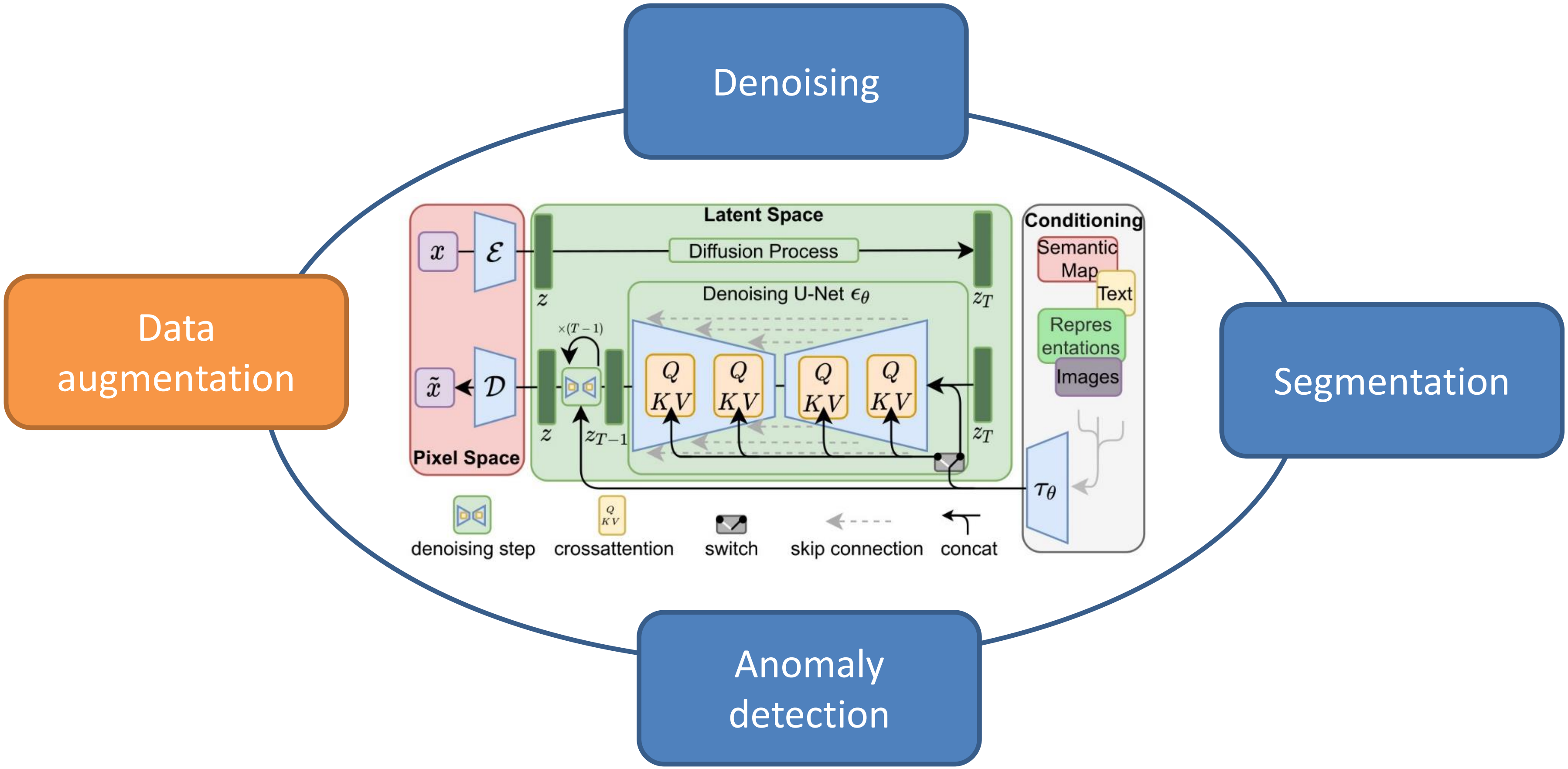
Denoising

Data augmentation



Segmentation

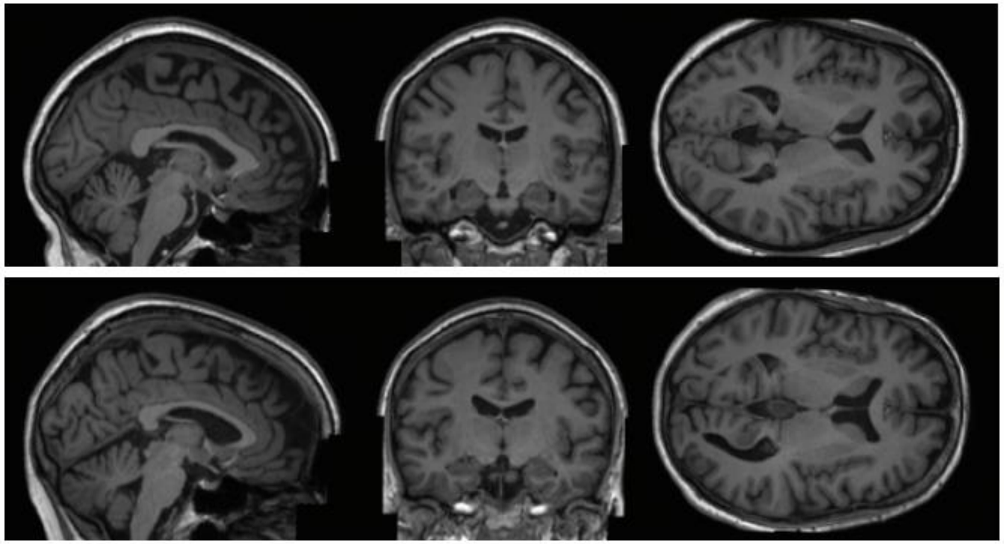
Anomaly detection


Denoising
Data augmentation
Segmentation
Anomaly detection
Latent Space
Diffusion Process
Denoising U-Net ϵ_θ
Pixel Space
Conditioning
Semantic Map
Text
Repres entations
Images
denoising step
crossattention
switch
skip connection
concat

Diffusion models for data augmentation

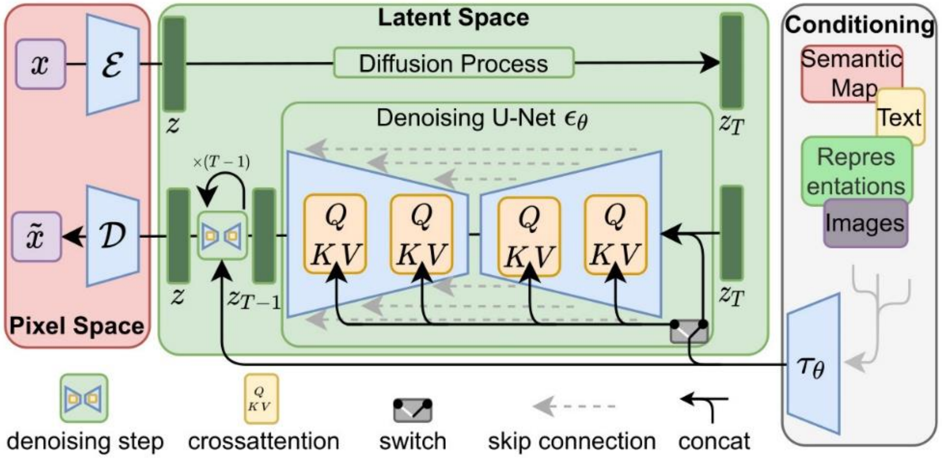
- Synthetic dataset generation for brain MR volumes [Walter et al., MICCAI workshop 2022]

- UK Biobank dataset
  - 3D MR volumes (T1w)
  - Training: 31,740 patients
  - with covariables: age (44 to 82 years), gender (53% women), brain structure volumes
  - Quality of synthetic data measured using FID: Fréchet Inception Distribution



Diffusion models for data augmentation

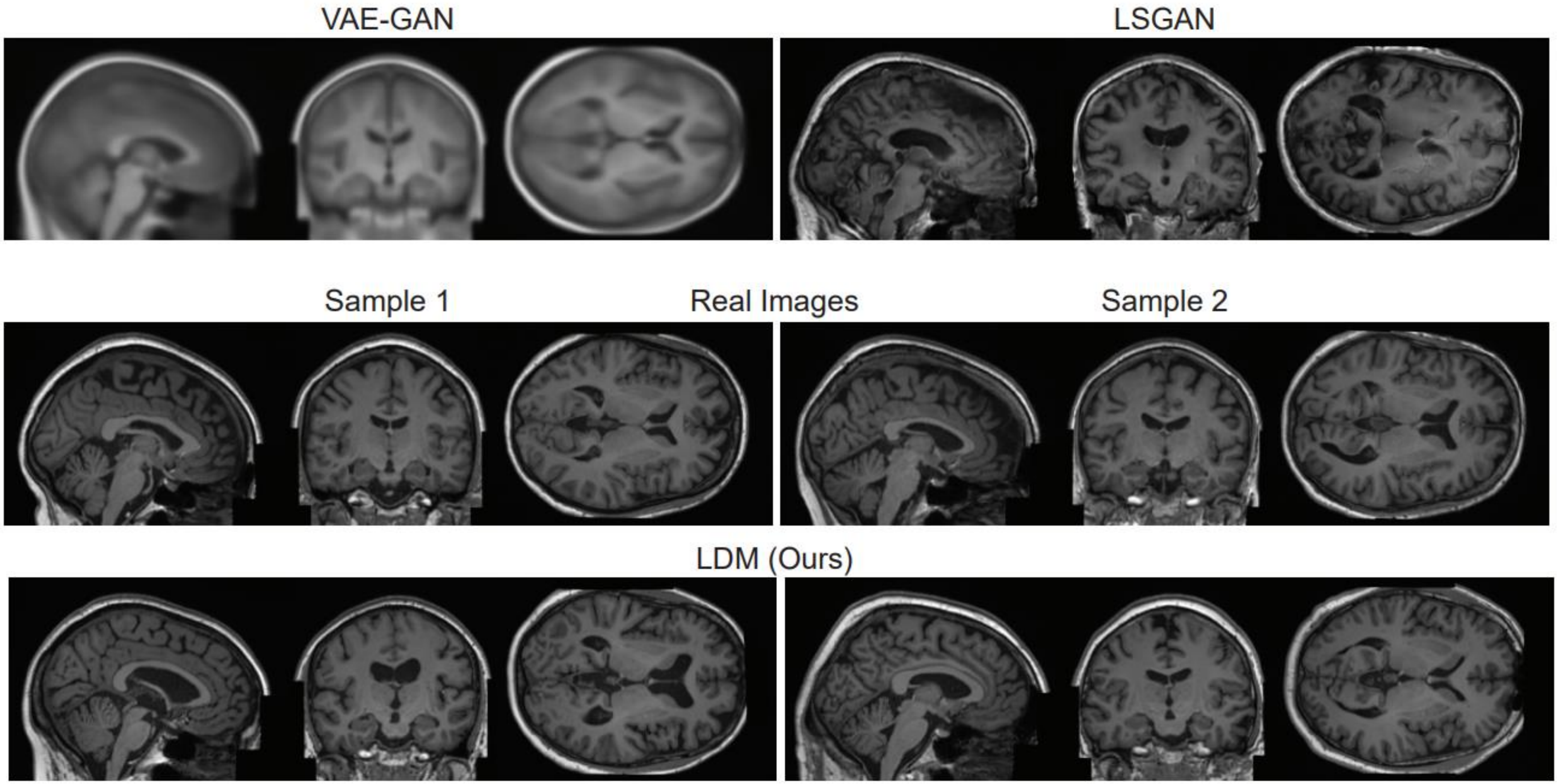
- VAE
  - 3D convolutions
  - Latent space dimension: 20 x 28 x 20
- DDPM
  - 3D convolutions
  - T=1000 time steps
  - Conditioning: vector encoding of each covariable



Diffusion models for data augmentation

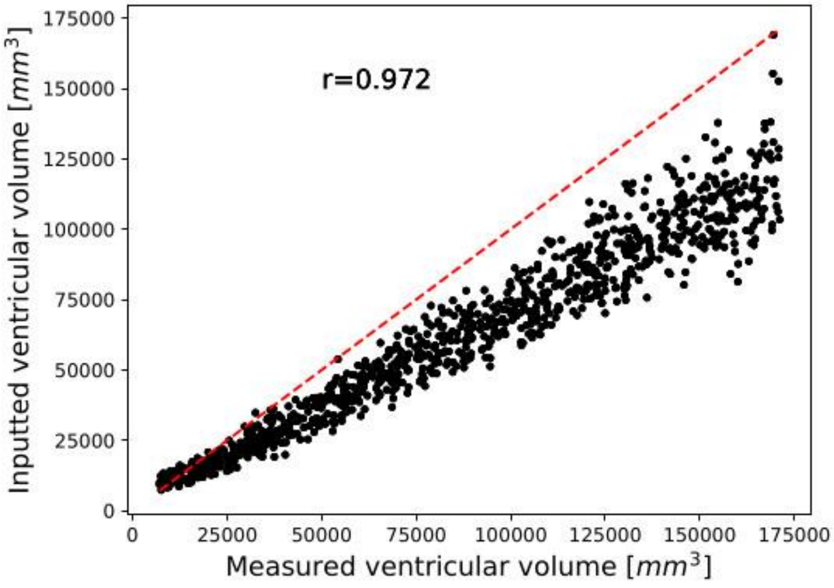
- Results
  - FID: generated from 1,000 samples drawn from each of the two distributions to be compared

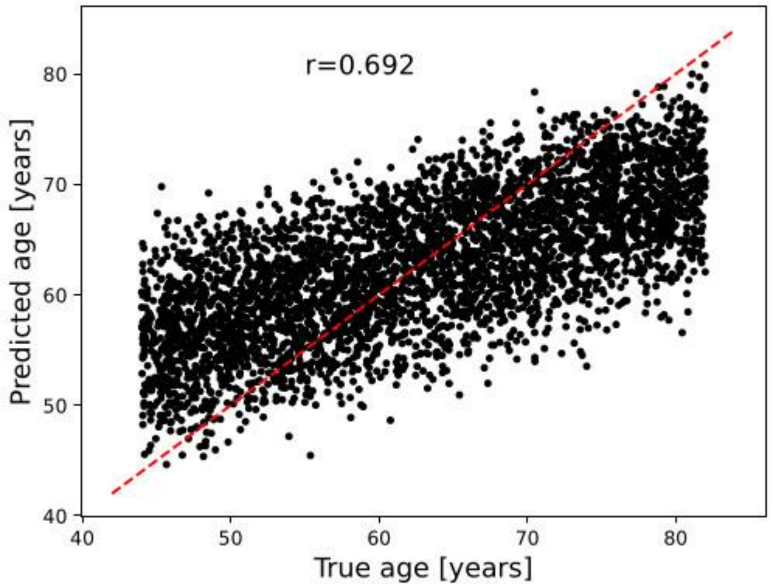
	FID ↓
LSGAN	0.0231
VAE-GAN	0.1576
LDM	**0.0076**
Real images	0.0005



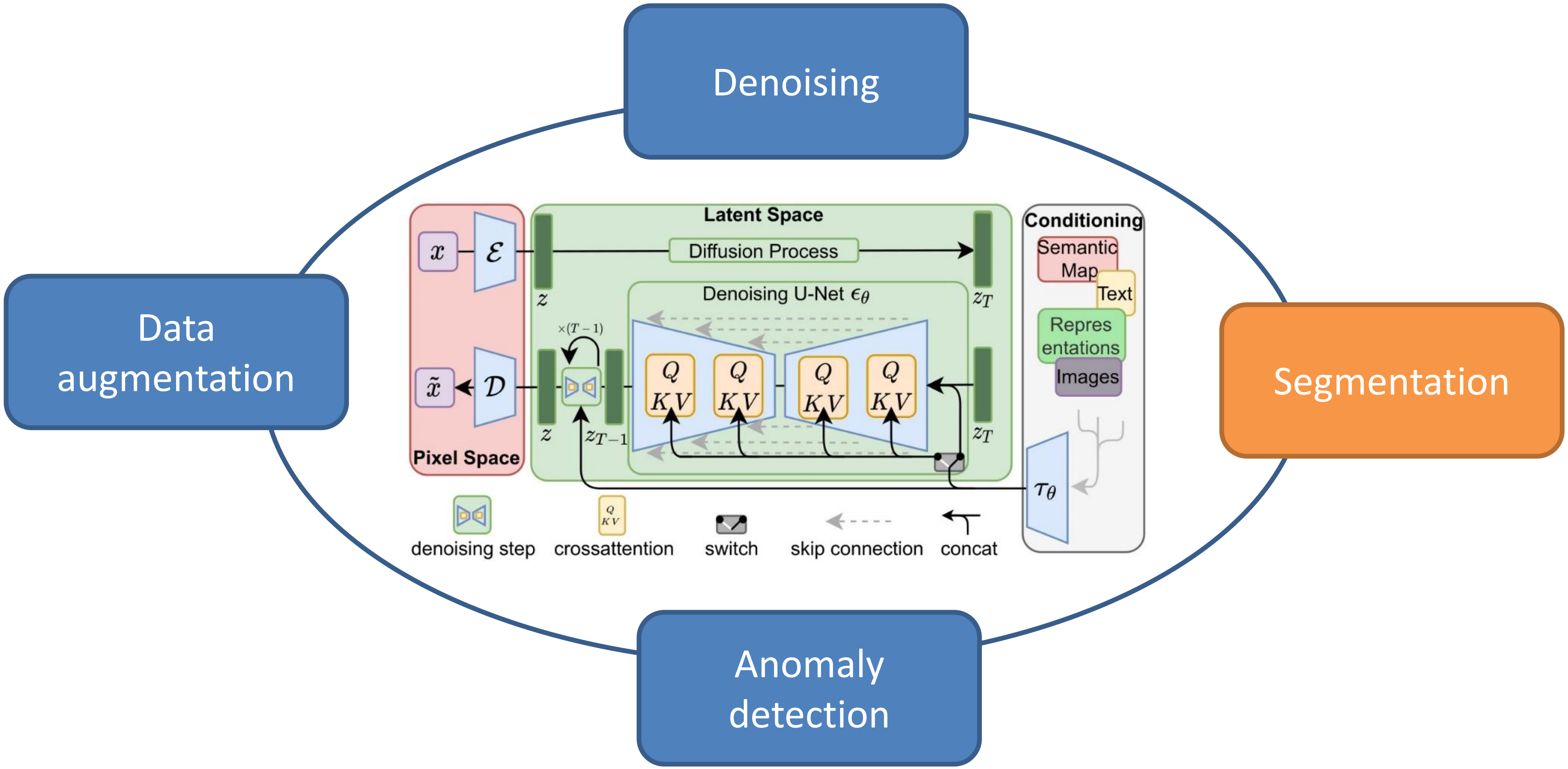
Diffusion models for data augmentation

- Results
  - SynthSeg model was used to automatically measure brain volumes from synthetic data
  - A 3D CNN trained from the UK biobank was used to automatically predict the age from the synthetic data





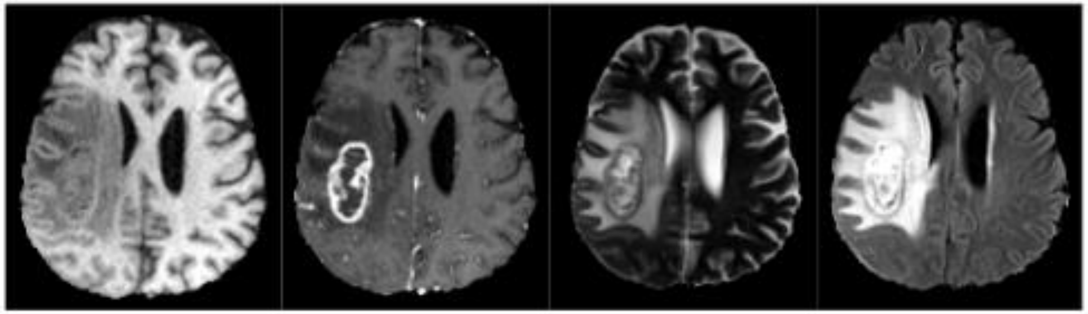
- Synthetic dataset of 100,000 human brain was generated and made publicly available with the conditioning information
- Promote data sharing with privacy guarantees


Denoising
Data augmentation
Segmentation
Anomaly detection
Latent Space
Diffusion Process
Denoising U-Net ϵ_θ
Pixel Space
Conditioning
Semantic Map
Text
Representations
Images
denoising step
crossattention
switch
skip connection
concat

Diffusion models for image segmentation

- Segmentation of tumors from MR images [Wolleb et al., MIDL 2022]

- BRATS2020 dataset
  - 4 different MR sequences per patient (T1, T2, T1ce, FLAIR)
  - Training: 332 patients with 3D volumes sequences => 16,998 2D images
  - Testing: 37 patients with 3D volumes sequences => 1,082 2D images

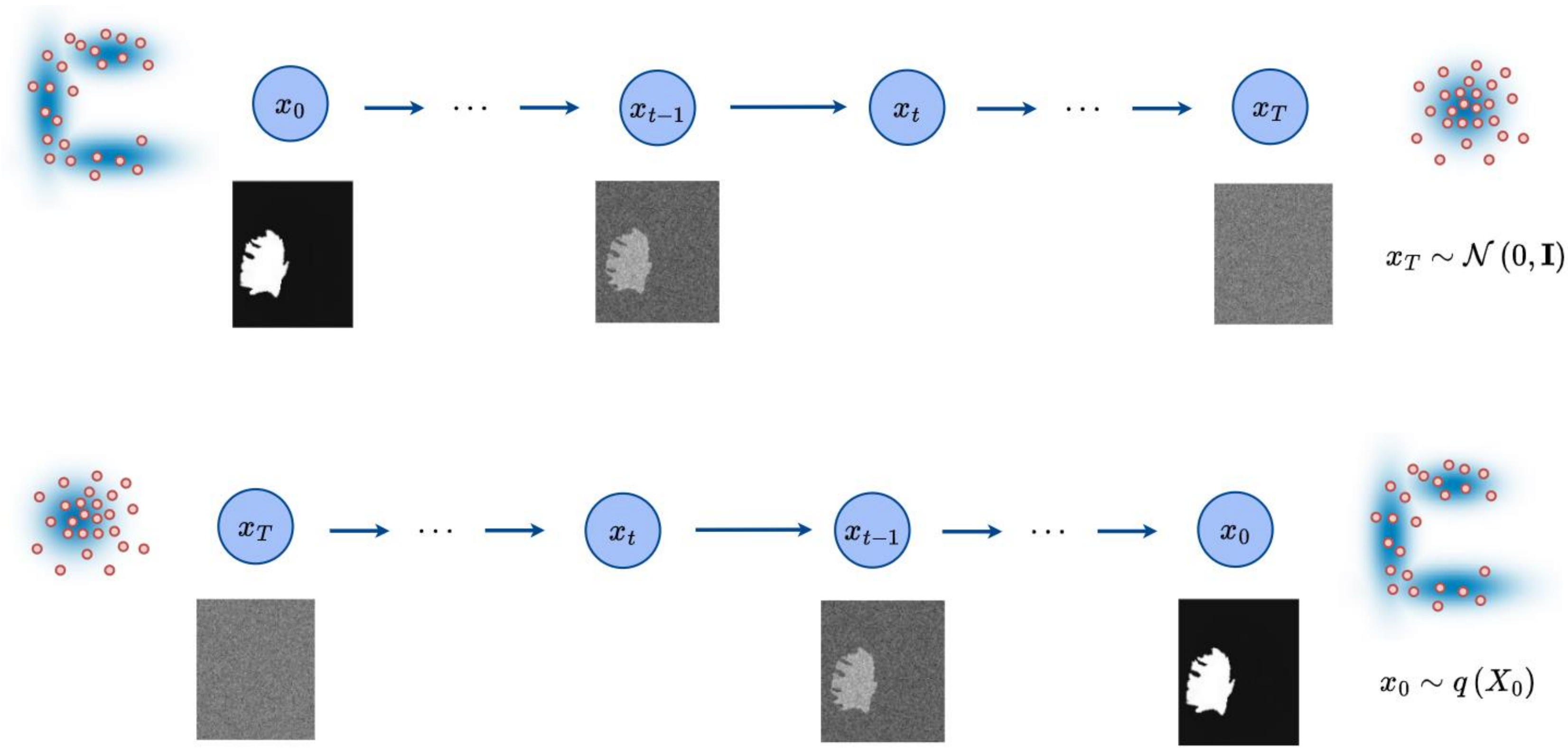


4 MR inputs per patient (T1, T2, T1ec, FLAIR)

Mask output

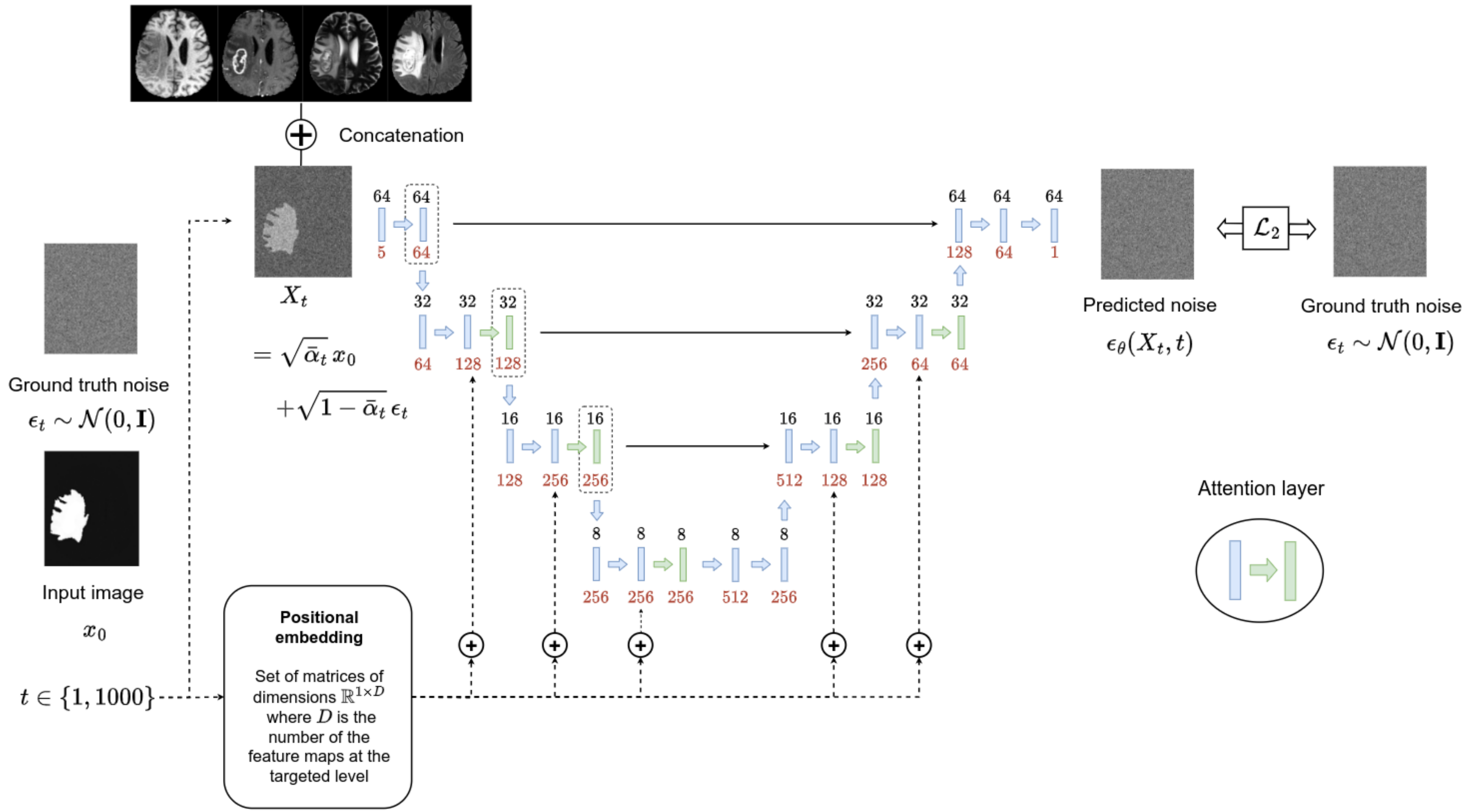
Diffusion models for image segmentation

- Learn the underlying distribution of tumor segmentation masks



Diffusion models for image segmentation

- Conditioning with the 4 MR images using concatenation scheme



Diffusion models for image segmentation

- At inference time: modelling of the segmentation uncertainty

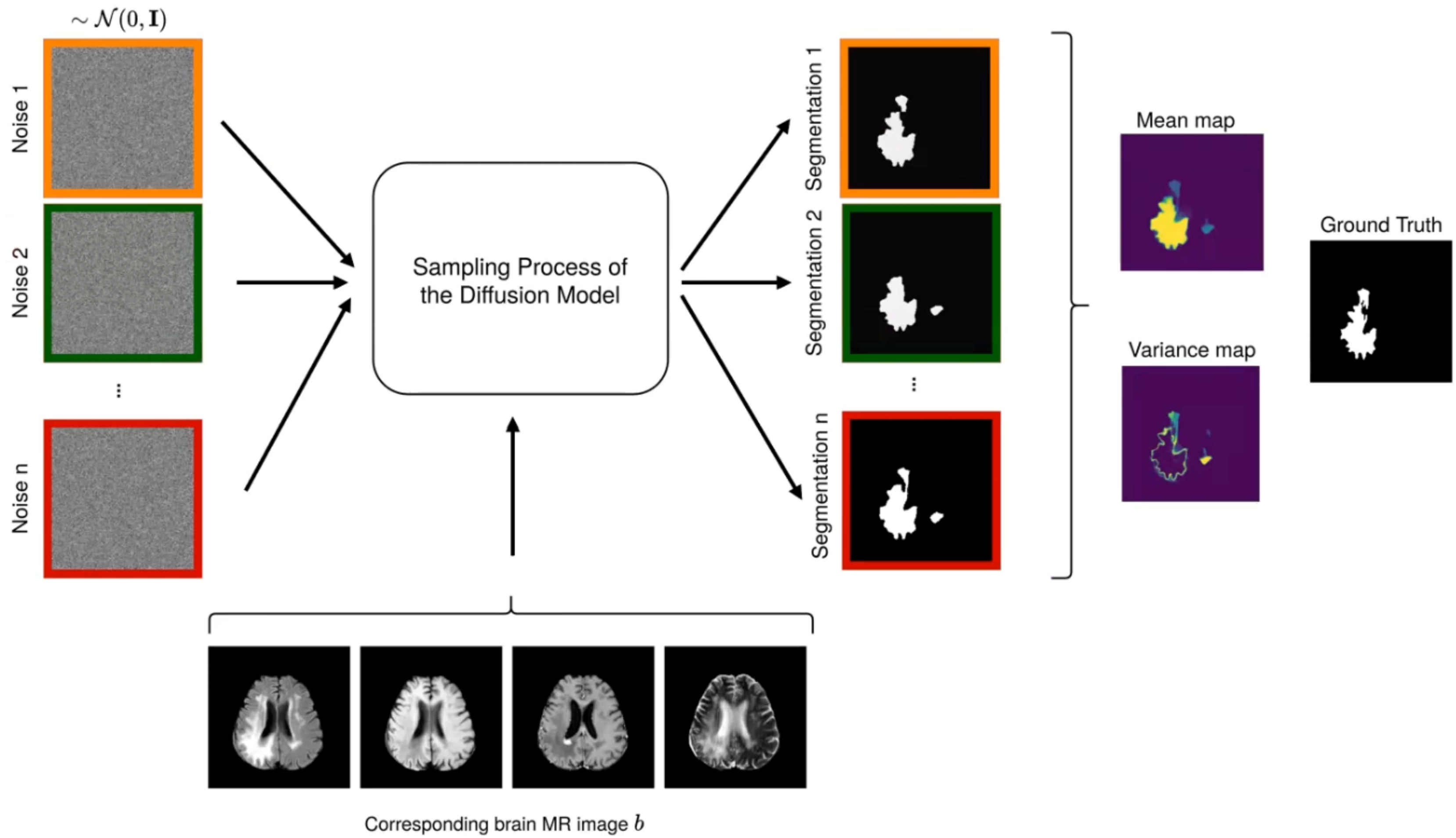


Illustration taken from https://www.youtube.com/watch?v=US9CzPrT2H8

Diffusion models for image segmentation

- Results

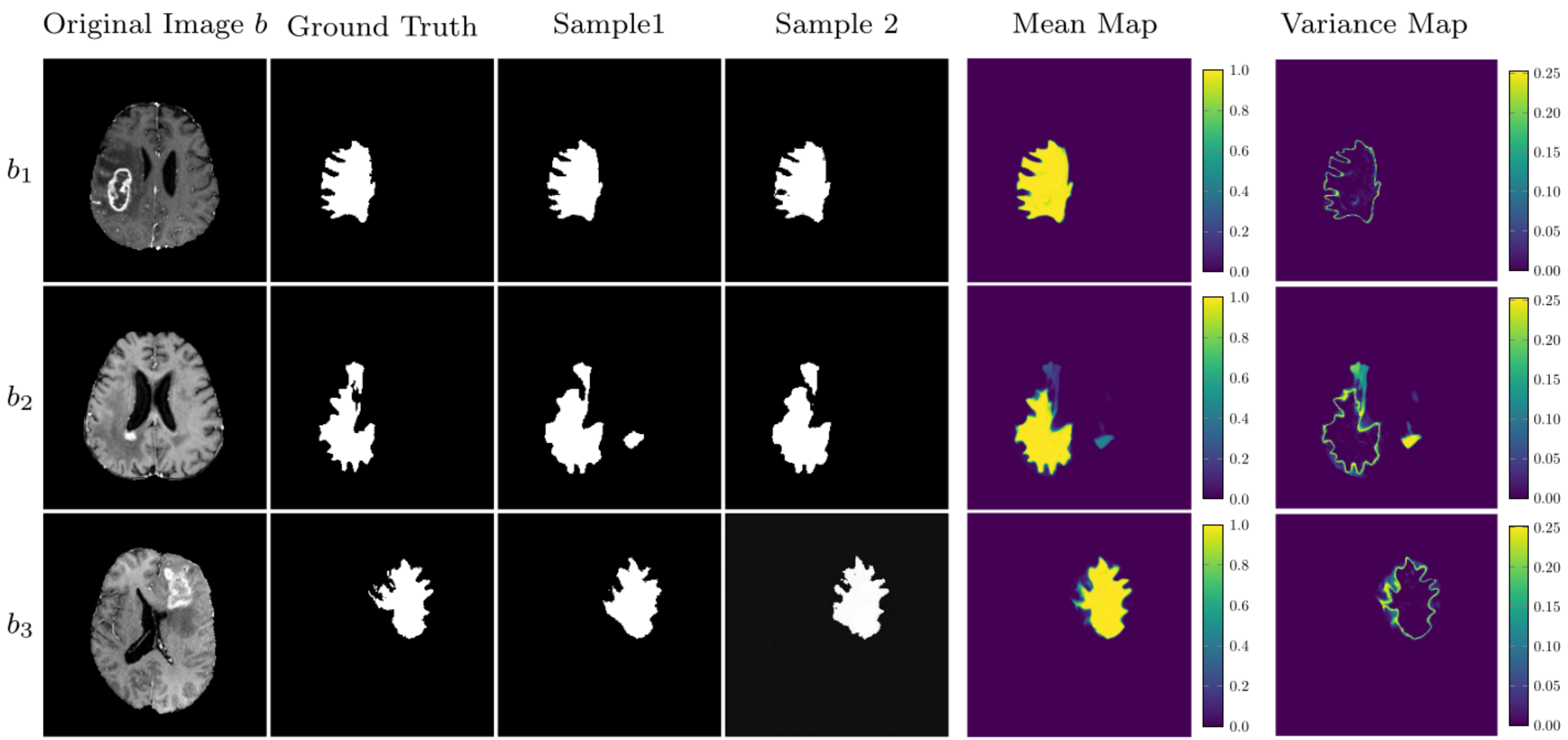
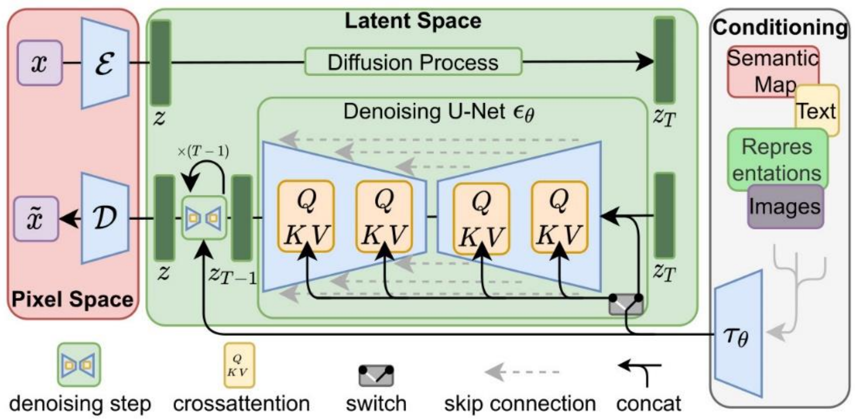


Illustration taken from https://arxiv.org/pdf/2112.03145

Denoising

Data augmentation



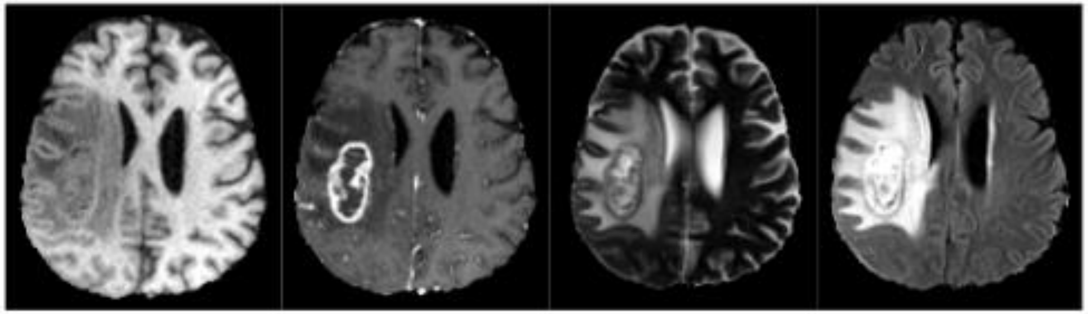
Segmentation

Anomaly detection

Diffusion models for anomaly detection

- Anomaly detection from MR images [Wolleb et al., MICCAI 2024]

- BRATS2020 dataset
  - 4 different MR sequences per patient (T1, T2, T1ce, FLAIR)
  - Training: 332 patients with 3D volumes sequences => 16,998 2D images
  - 5,598 healthy 2D slices (without tumor) / 10,607 disease 2D slices

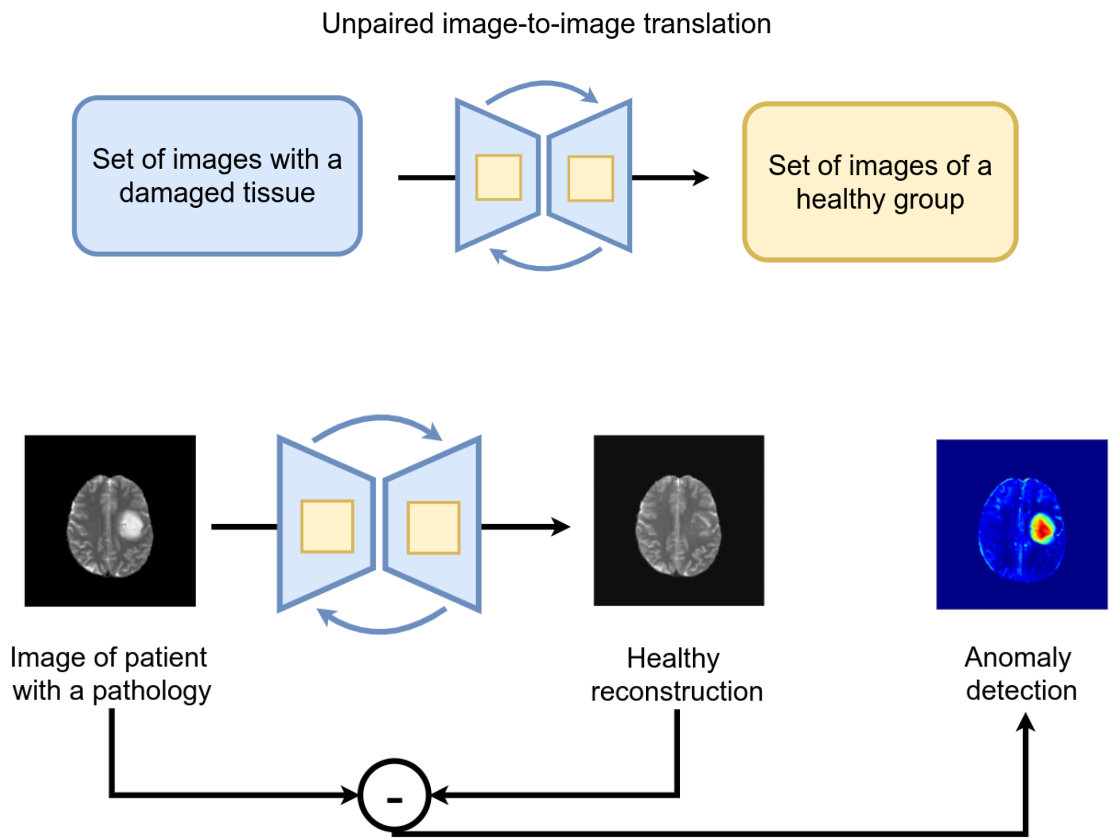


4 MR inputs per patient (T1, T2, T1ec, FLAIR)

Mask output

Diffusion models for anomaly detection

- General idea

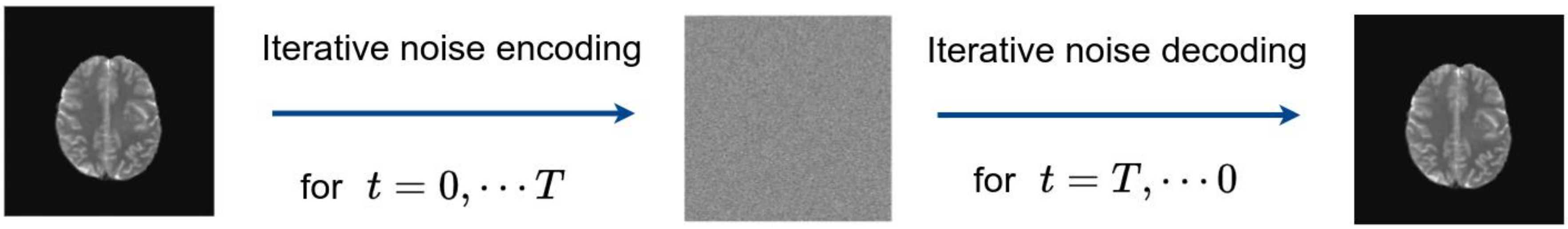


How to preserve spatial anatomical information using a diffusion process?

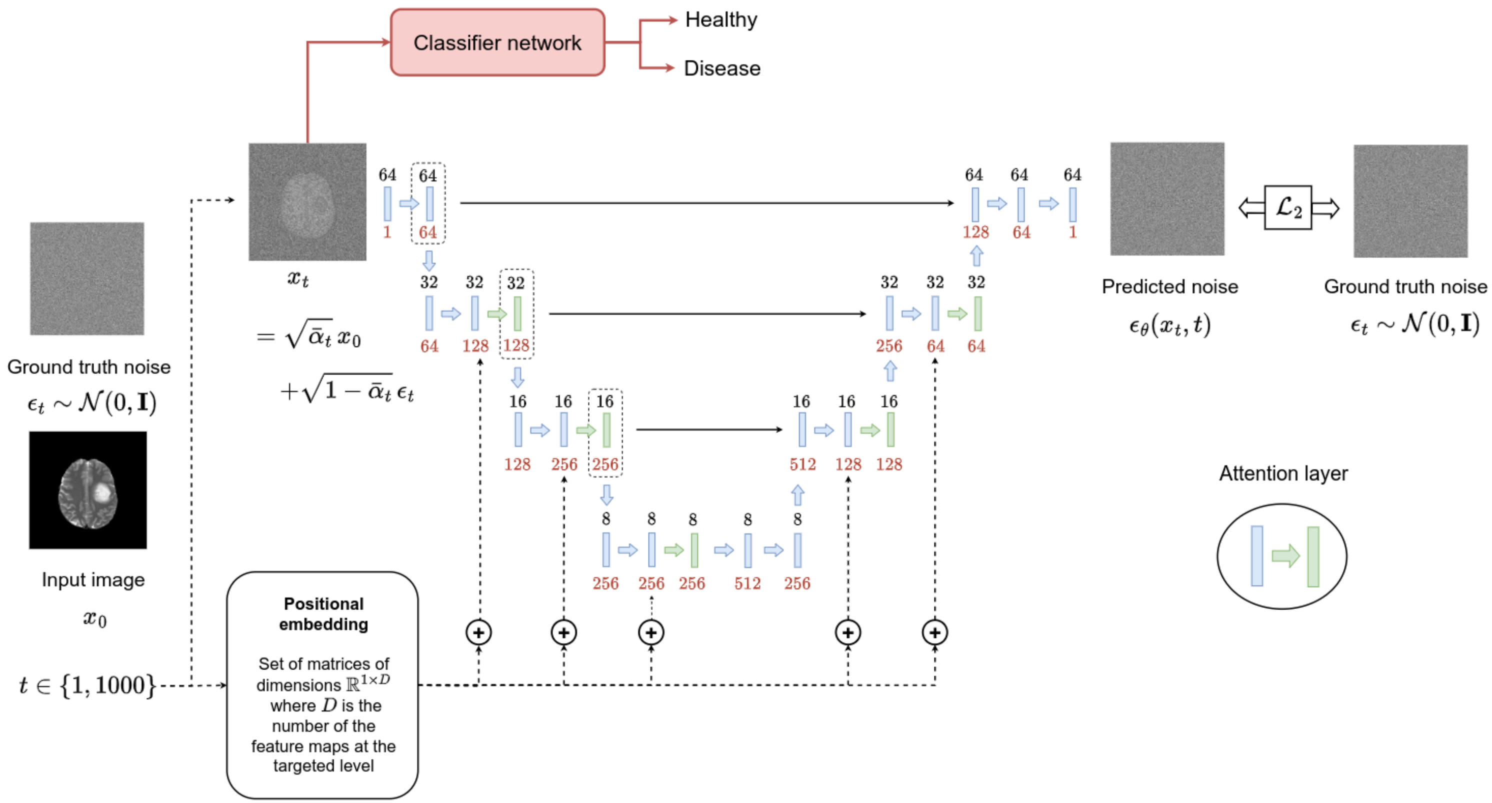
Diffusion models for anomaly detection

- Denoising Diffusion Implicit Models (DDIM)

  - Reformulation of the diffusion process
  - Remove the random component $\sigma_t \epsilon$
  - Make the diffusion process deterministic



- Main algorithm – part 1
  - Train a classical DDPM on the dataset containing healthy and disease images
  - Train a classifier network C to predict the class label (healthy vs disease) from any noisy images x_t



- Main algorithm – part 2
  - Use DDIM process
  - Compute the gradient of the classifier to guide the removing of anomaly regions

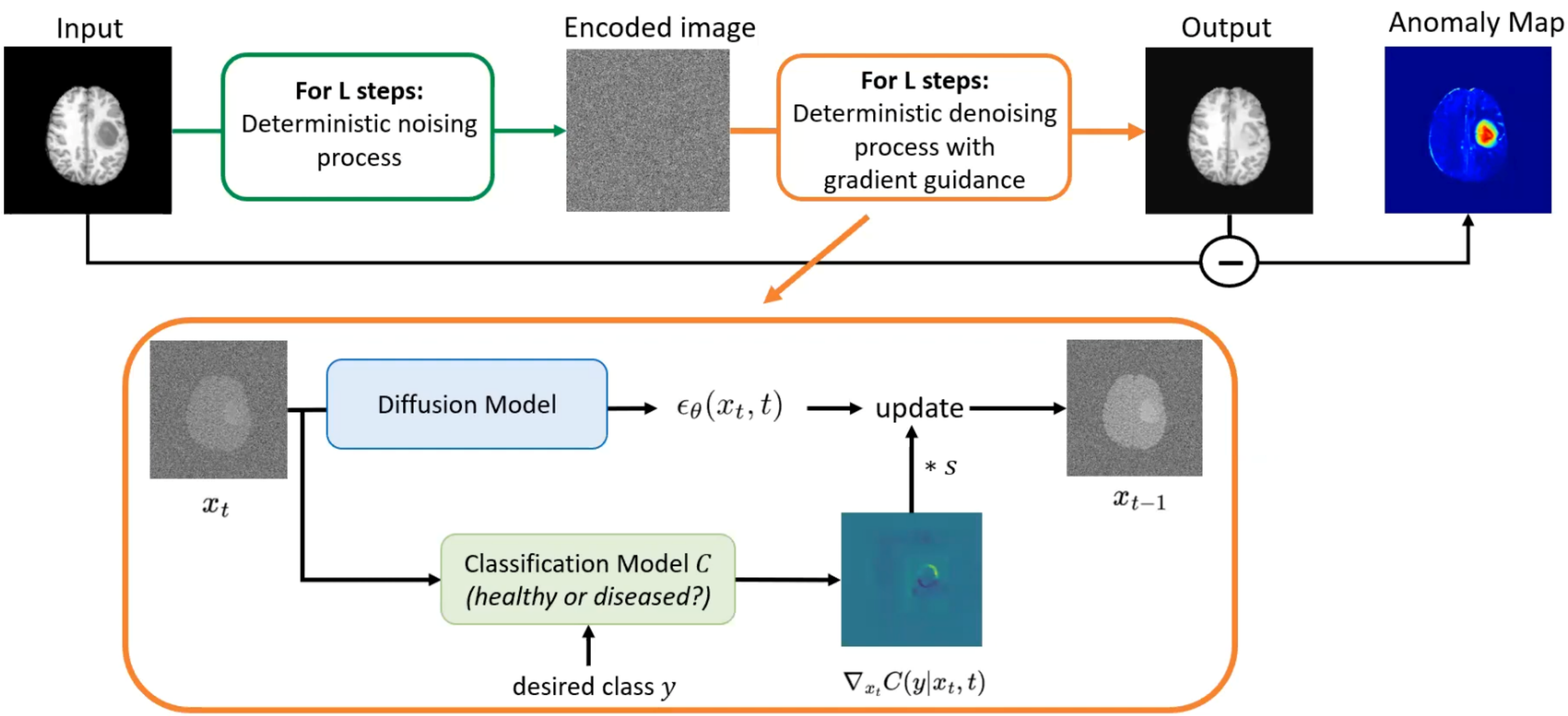
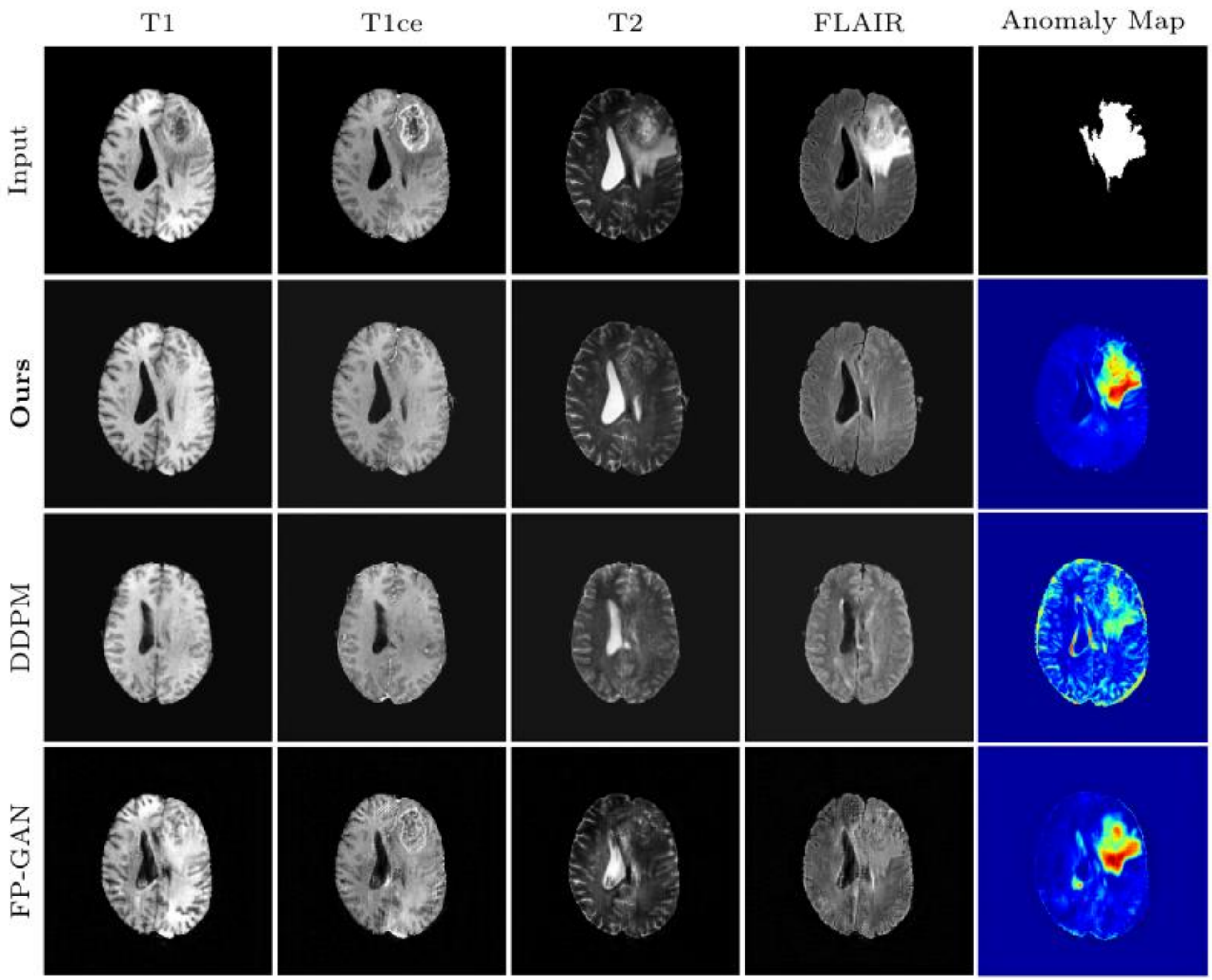


Illustration taken from https://arxiv.org/pdf/2203.04306

Diffusion models for anomaly detection

- Result on an image with a tumor



Diffusion models for anomaly detection

- Result on an image without any tumor

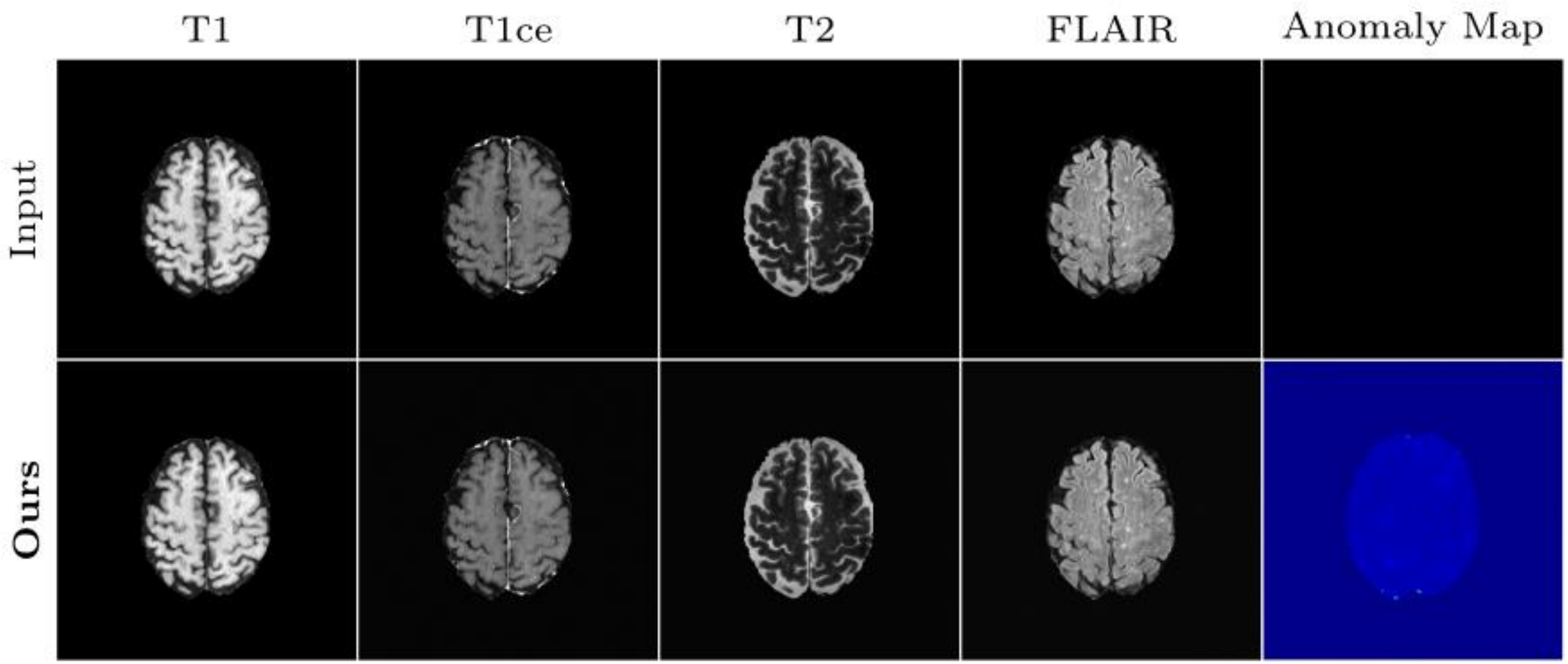


Illustration taken from https://arxiv.org/pdf/2203.04306

That’s all folks

- Key idea
  - Generating sample z according to $p(z|x)$, implies that z has been generated by images from the original data distribution $p(x)$
  - If can reconstruct vectors back into images, we will effectively generate new samples from our original data distribution
  - We need to know the latent distribution, which is assume to be a normal distribution
  - This allows to compute the likelihood p(x/z)
  - The only unknown remains the true posterior p(z/x)
  - Thanks to variational inference, we approximate it using a Gaussian distribution q(z/x)
  - This Gaussian will have parameters mu and sigma that we need to learn, which is an optimization process known as variational Bayes
  - We will train an encoder to estimate these parameters mu and sigma from the images
  - Then we used a decoder to reconstruct images from the latent variables that are sampled from the approximate posterior