SAISON 22/23

Formation

Introduction au Deep Learning

FIDLE

Ressources

https://fidle.cnrs.fr

Course materials (pdf)

Practical work environment*

Corrected notebooks

Videos (YouTube)

(*) Procedure via Docket or pip
Remember to get the latest version !

List and networks

You can also subscribe to :

http://fidle.cnrs.fr/listeinfo

Fidle information list

https://listes.services.cnrs.fr/wws/info/devlog

List of ESR* « calcul » group

https://listes.math.cnrs.fr/wws/info/calcul

List of ESR* « calcul » group

(*) ESR is Enseignement Supérieur et Recherche, french universities and public academic research organizations

Program

20 Séquences
du 17 novembre
au 14 mai 2023

1 ✓
History, Fundamental Concepts
Basic Regression DNN
Basic Classification DNN

2 3 ✓
Hight Dimensionnal Data
CNN

4 ✓
Demystify **mathematics** for neural networks.

5 ✓
Training strategies Evaluation
Sparse data (text) Embedding

6 ✓
Sequences data
RNN

7 ✓
PyTorch
A small detour with **PyTorch**.

8 ✓
«Attention is All You Need»
Transformers

9 ✓ New !
Graph Neural Network
GNN

10 ✓
Autoencoder networks
AE

11 ✓
Variational Antoencoder
VAE

12 JDLS 12 mai
Project session
«My project in 180 s»

13 ✓
Generative Adversarial Networks
GAN

14 New !
Diffusion Model
Text to image

15
AI, Law, Society and **Ethics**

16 New !
Model and training optimization
Resource efficiency

17
Jean-Zay
GPU acceleration

18 New !
Physics-Informed Neural Networks
PINNS

19
Deep Reinforcement Learning
RL

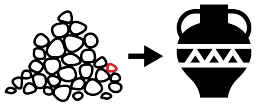
20
What will be tomorrow's AI
Review & perspectives !

FIDLE

SAISON 22/23

Roadmap

14 New !

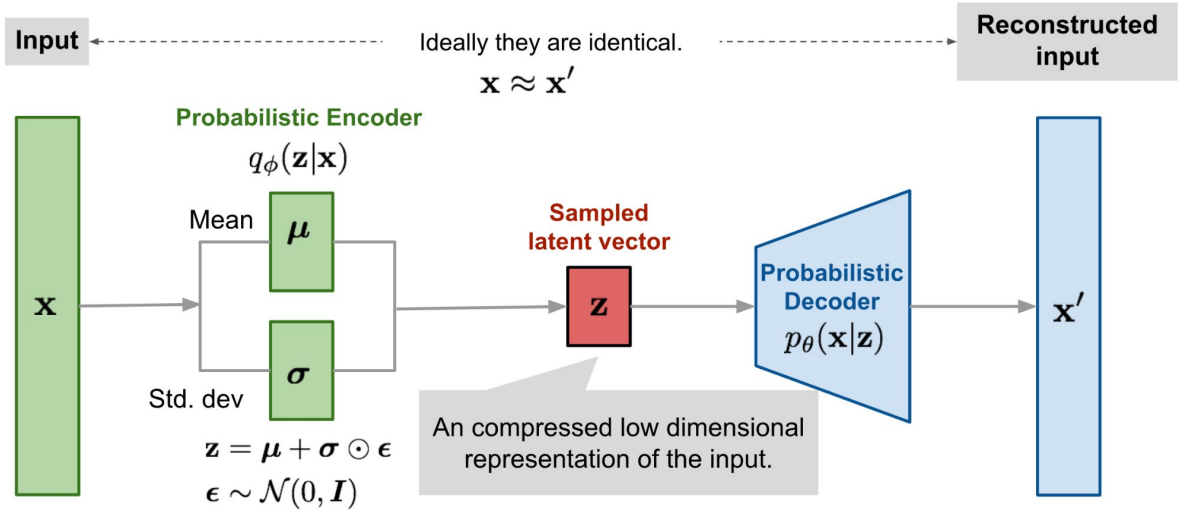


Diffusion Model

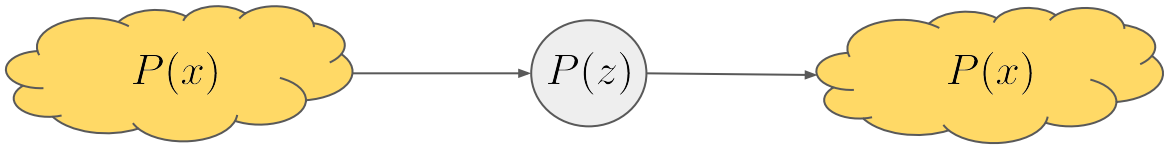
Text to image

1. **Introduction**
   - ➔ Reminder on generative model
   - ➔ Diffusion Model vs VAE
2. **Denoising Diffusion Probabilistic Models**
   - ➔ Principle
   - ➔ Forward and Reverse Diffusion
   - ➔ Training and Sampling
3. **DDPM improvements**
   - ➔ Beta scheduling and Variance learning
   - ➔ Fast sampling
   - ➔ Latent diffusion
5. **DDPM applications**
   - ➔ Text-to-image
   - ➔ Other task : inpainting / outpainting / super-resolution
6. **Example: Fashion MNIST**
   - ➔ Generation of Fashion MNIST

Rappel - VAE - 1



Source https://lilianweng.github.io/posts/2021-07-11-diffusion-models/



Rappel - VAE - 2

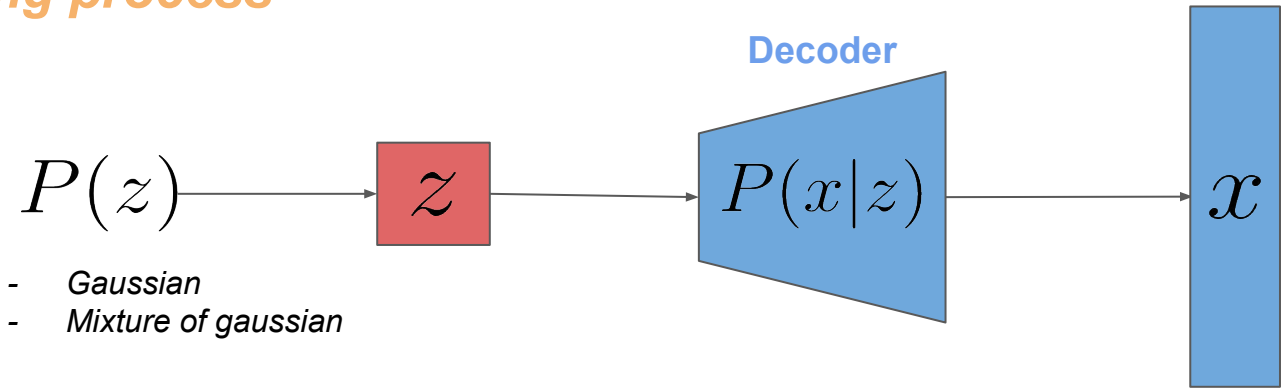
Training process

metric

VAE COST FUNCTION

$$\log p_\theta(\mathbf{x}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x})) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))$$

Sampling process



- *Gaussian*
- *Mixture of gaussian*

Rappel - GAN - 1

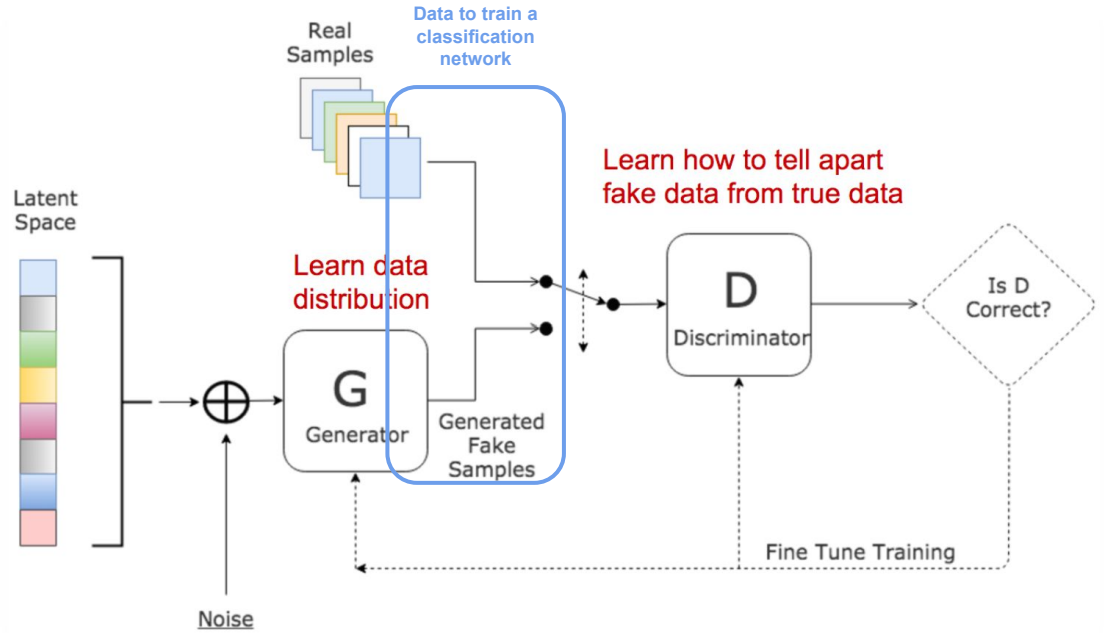
Adding supervision concept in an unsupervised task !

2 networks in opposition :

- *Generator*
- *Discriminator*

Ideal solution :

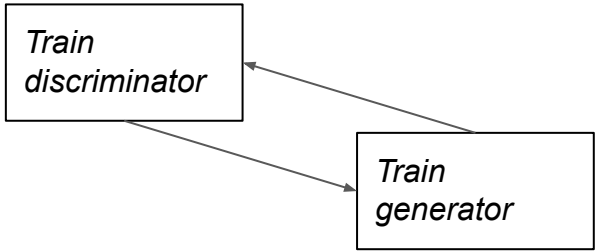
- *Generator ~ P(x|z)*
- *Discriminator = ½*



Source
https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

Rappel - GAN - 2

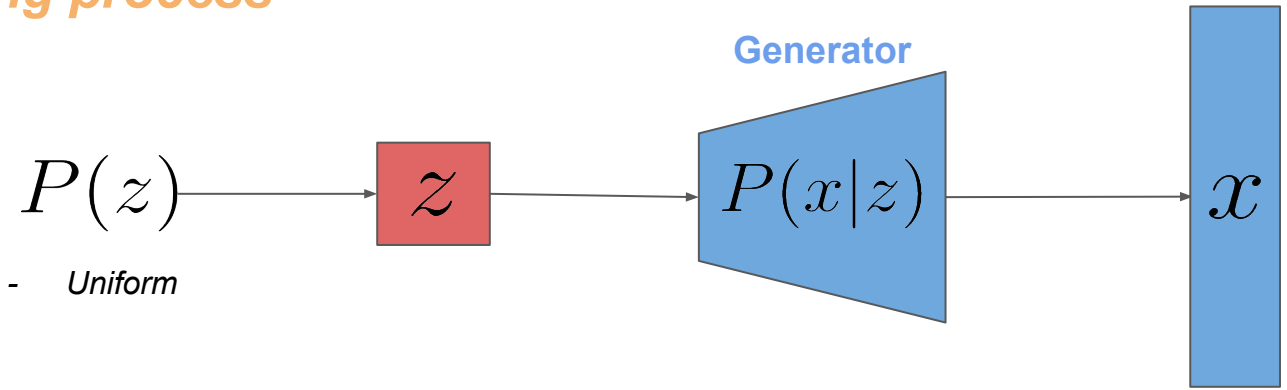
Training process



GAN COST FUNCTION

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

Sampling process



- *Uniform*

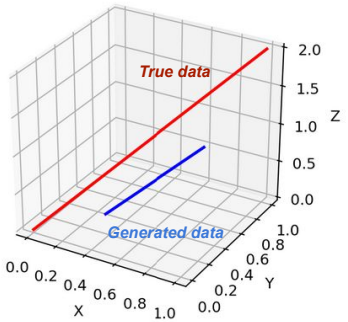
Rappel - GAN - 3

GAN convergence problems !

No convergence due to the nature of the problem (MinMax)

?

Vanishing gradient due to discriminant being too perfect, generator can't train anymore



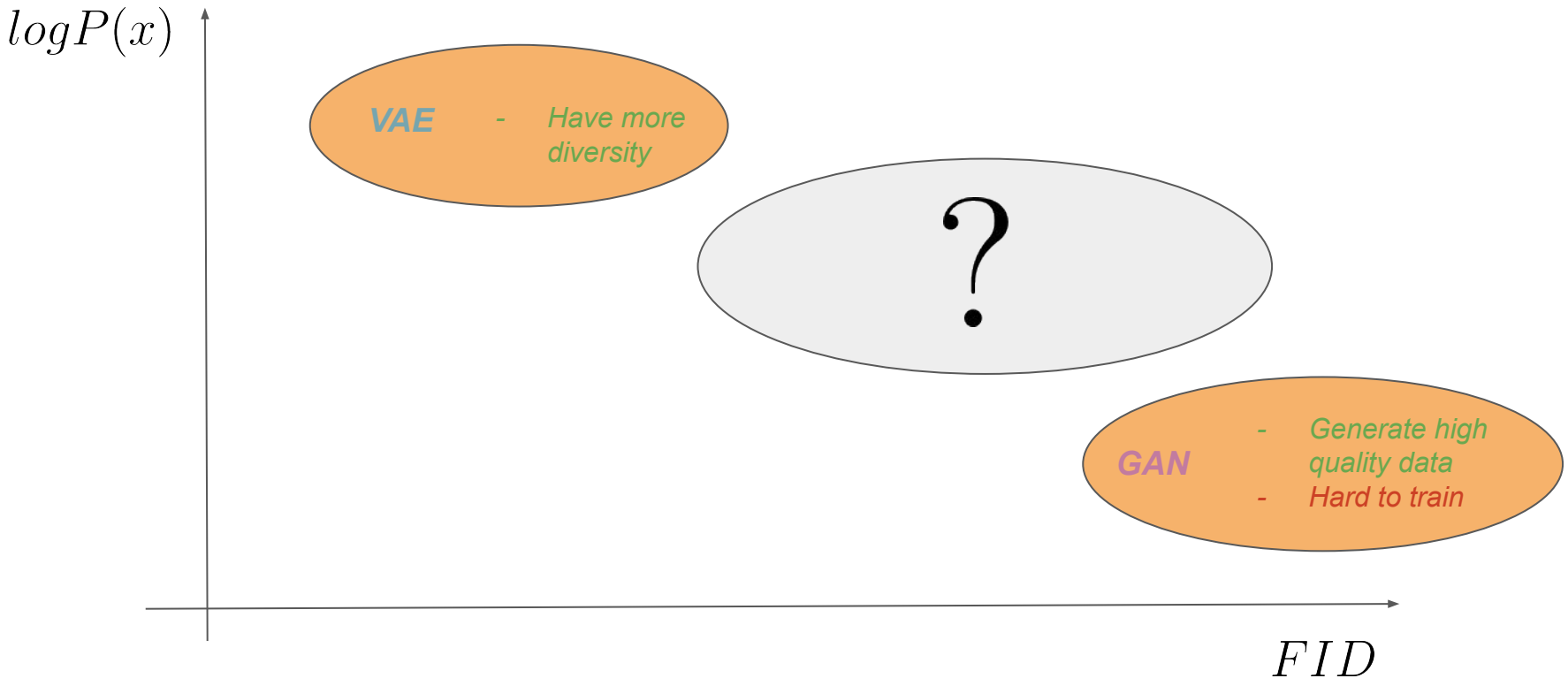
Source
https://lilianweng.github.io/posts/2017-08-20-gan/

Mode collapse due to generator learning only some good examples instead of the whole data distribution

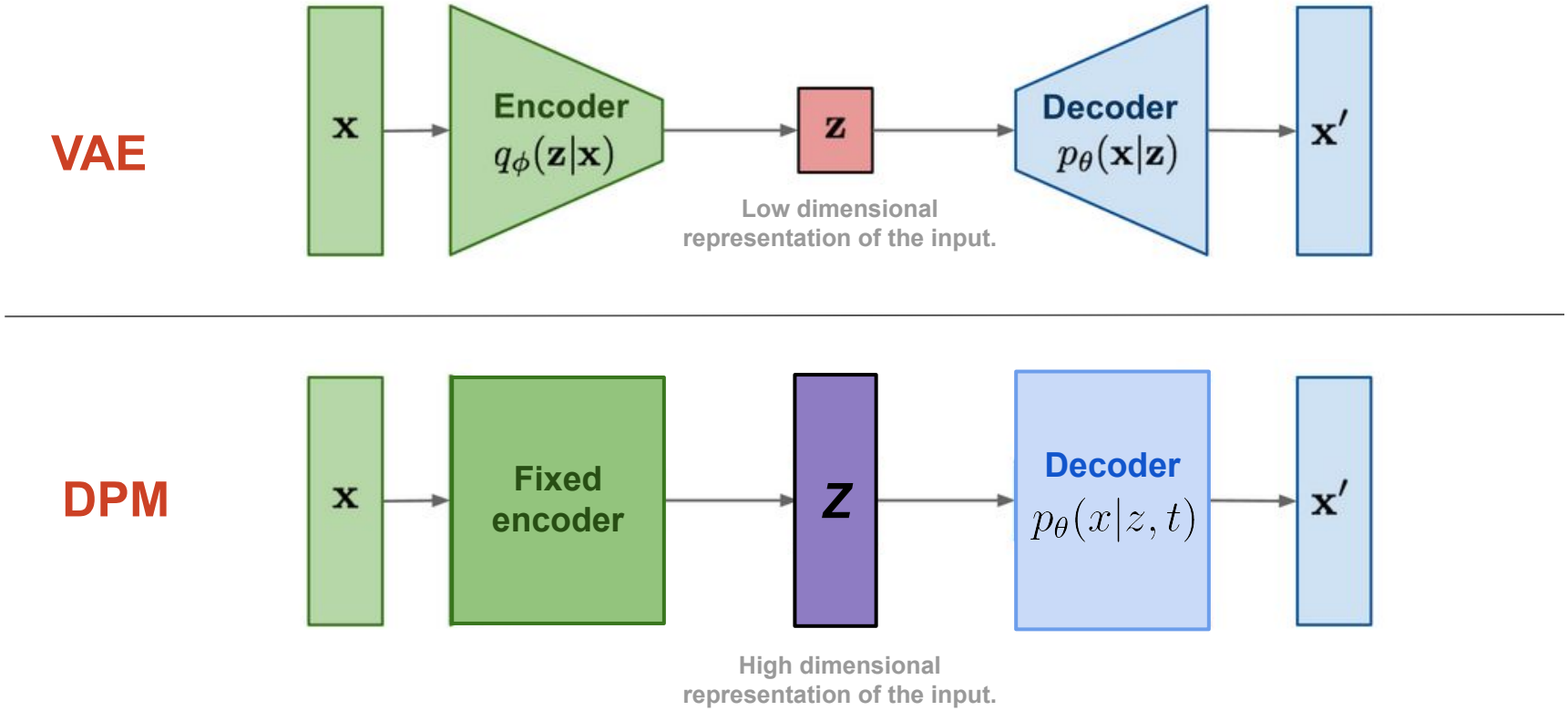
Generated Data

True Data

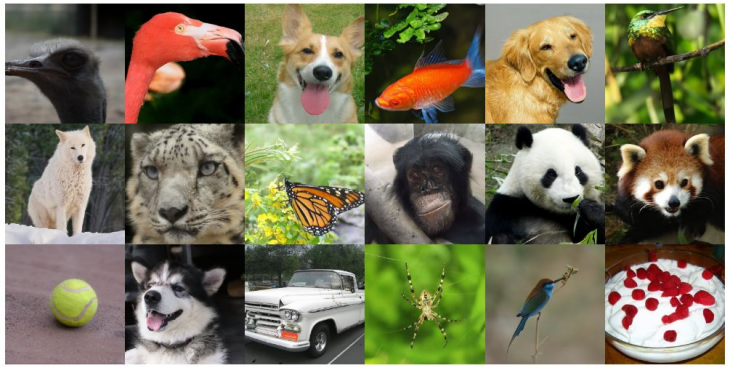
How to compare generative models ?



VAE vs DPM

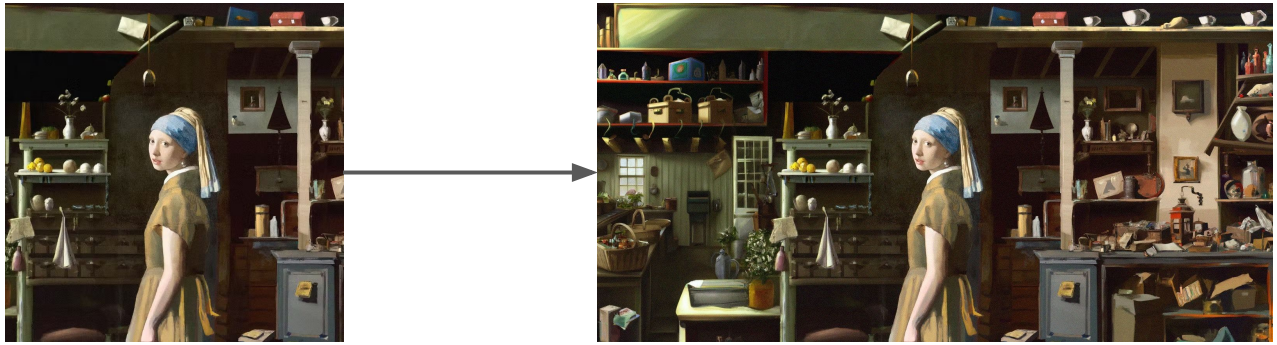

VAE
x
Encoder
$q_\phi(\mathbf{z}|\mathbf{x})$
z
Low dimensional
representation of the input.
Decoder
$p_\theta(\mathbf{x}|\mathbf{z})$
x′
DPM
x
Fixed
encoder
Z
High dimensional
representation of the input.
Decoder
$p_\theta(x|z,t)$
x′

DPM - Landscape



Dhariwal & Nichol, 2021

Source : https://github.com/bentoml/stable-diffusion-bentoml



Source : Dall-E 2

DDPM - Principle - 1

After the training the Diffusion Model will generate images from Gaussian noise:

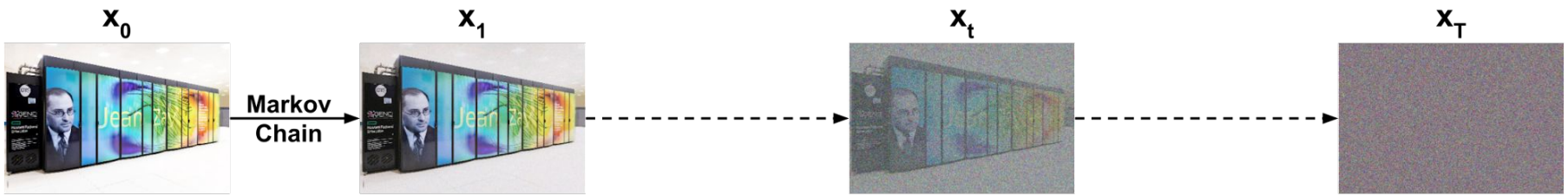
DDPM - Principle - 2

There are three processes that characterize Diffusion Models:

1. Forward Diffusion Process
2. Reverse Diffusion Process
3. Sampling Process

DDPM - Principle - 3

Forward Diffusion Process



This process will add noise to any image gradually

$0 \leq t \leq T$; T is a hyperparameter

DDPM - Principle - 4

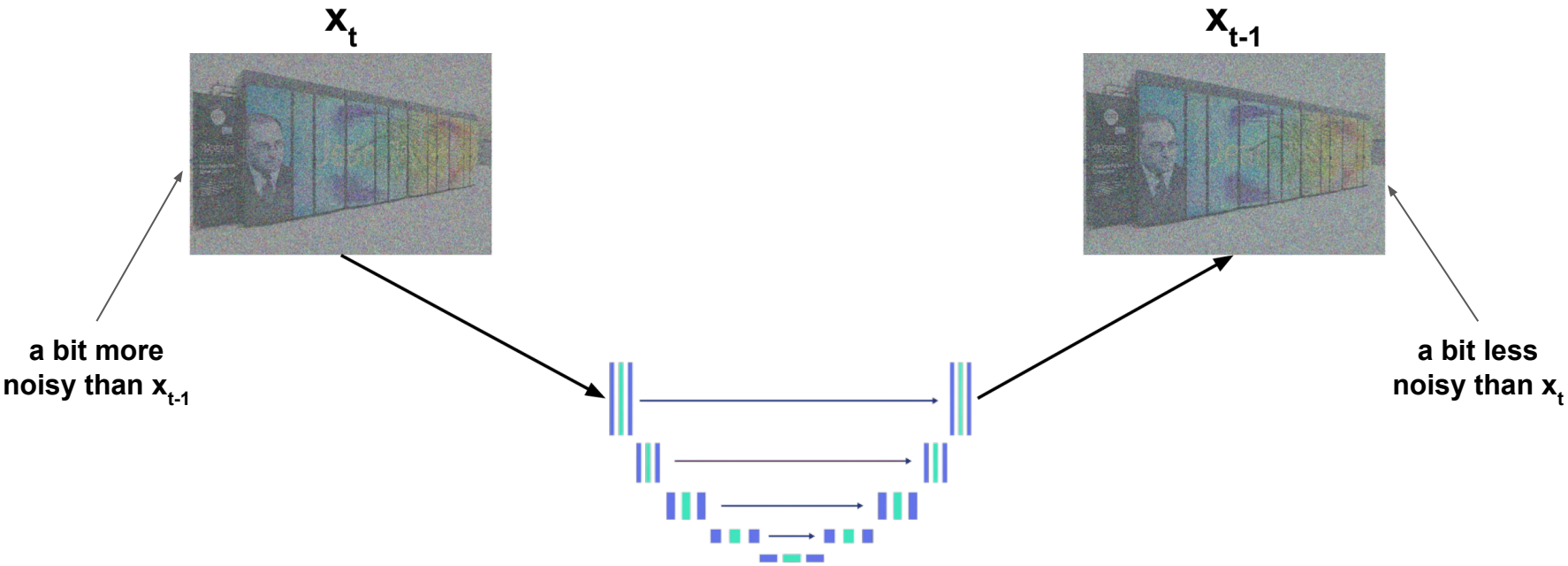
Forward Diffusion Process

Examples of images at different times t

Here we choose T=1000, but it can be different values (it's an hyperparameter)

DDPM - Principle - 5

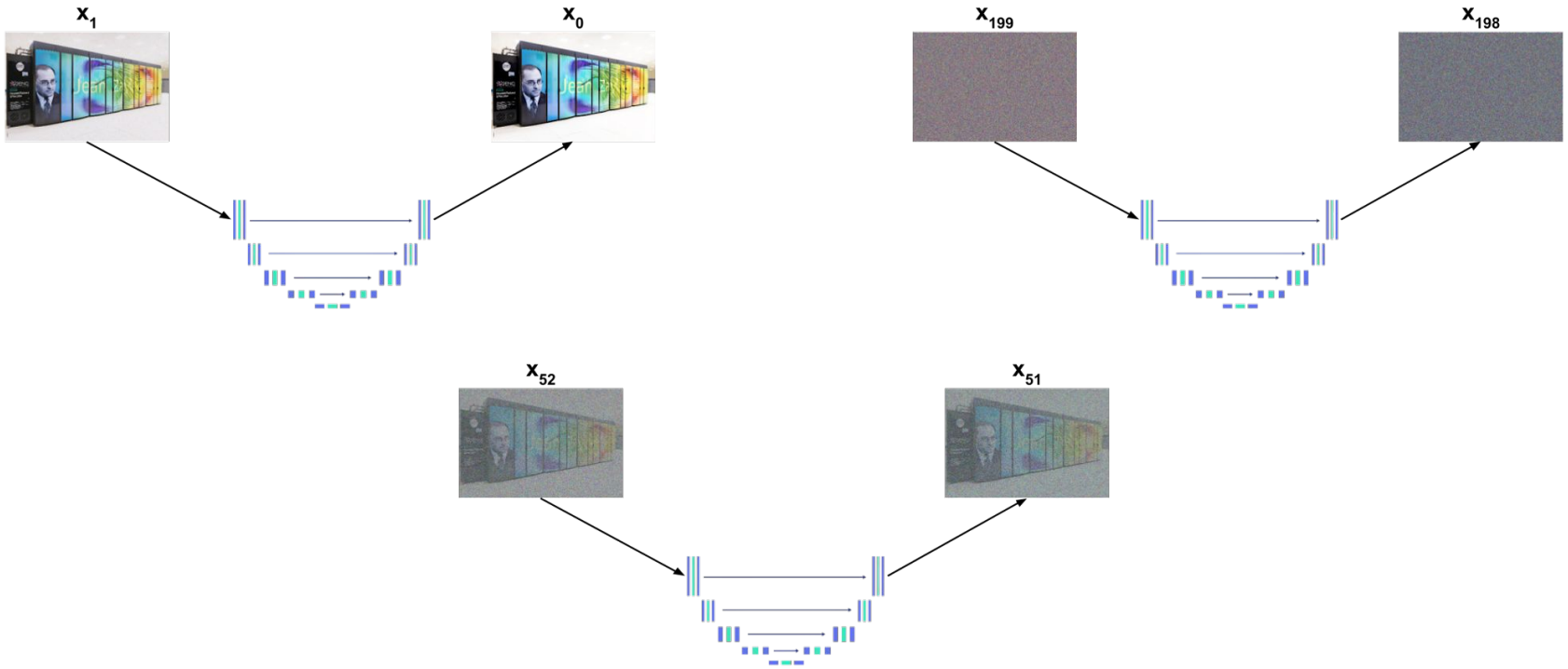
Reverse Diffusion Process



We train a model to predict x_{t-1} from x_t

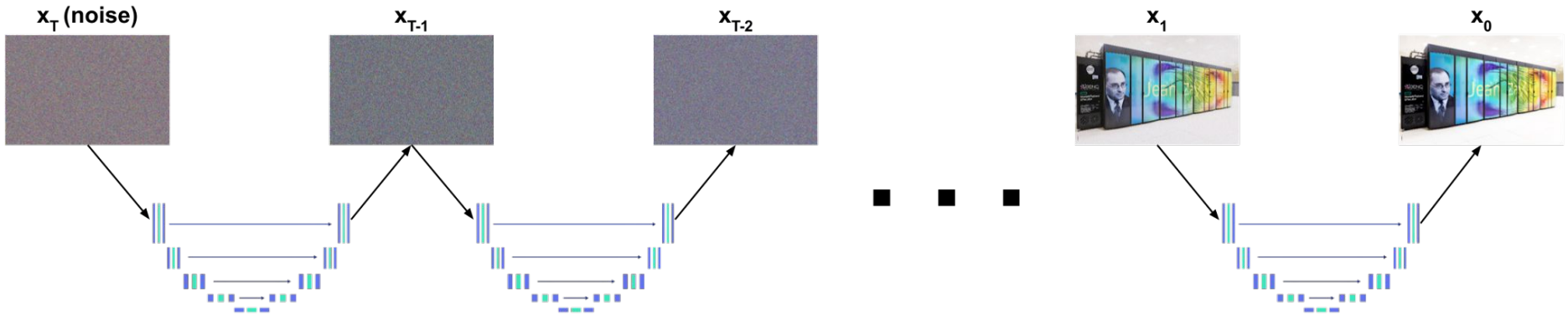
x_0 is any image from the dataset

Reverse Diffusion Process



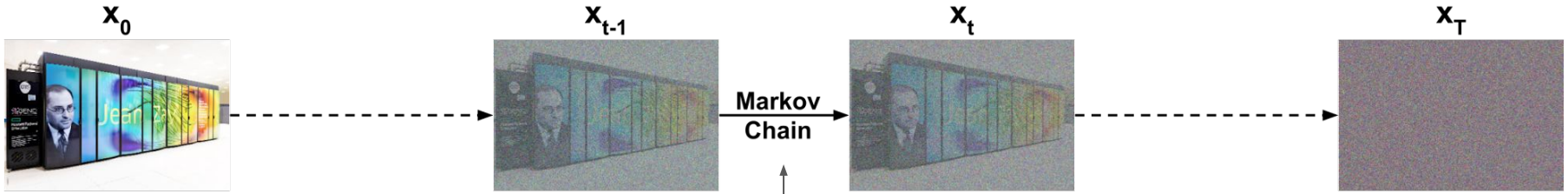
The same model must predict every x_{t-1} from x_t

Sampling Process



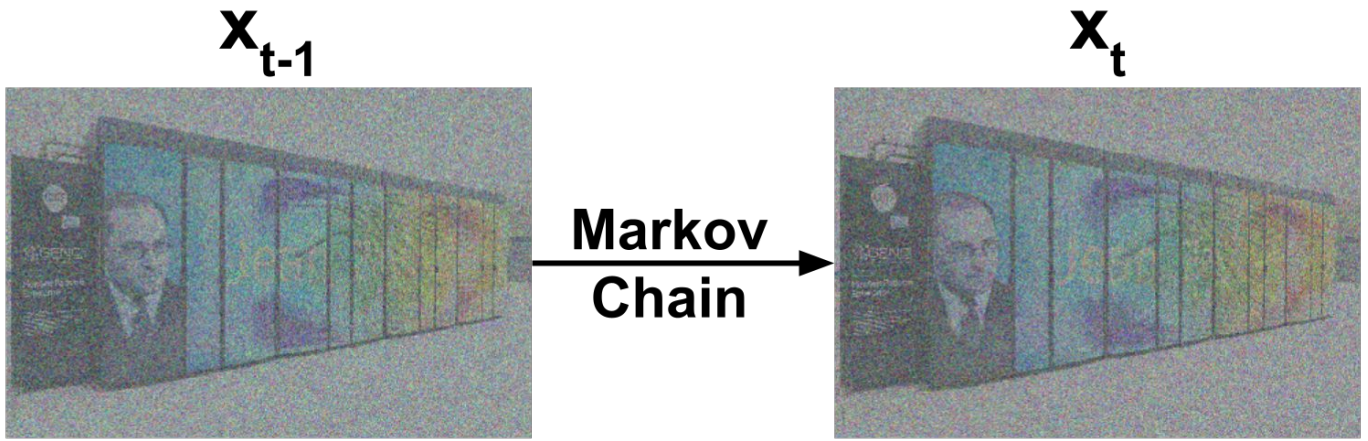
From a random noise we can generate an image

DDPM - Forward Diffusion - 1



$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

DDPM - Forward Diffusion - 2



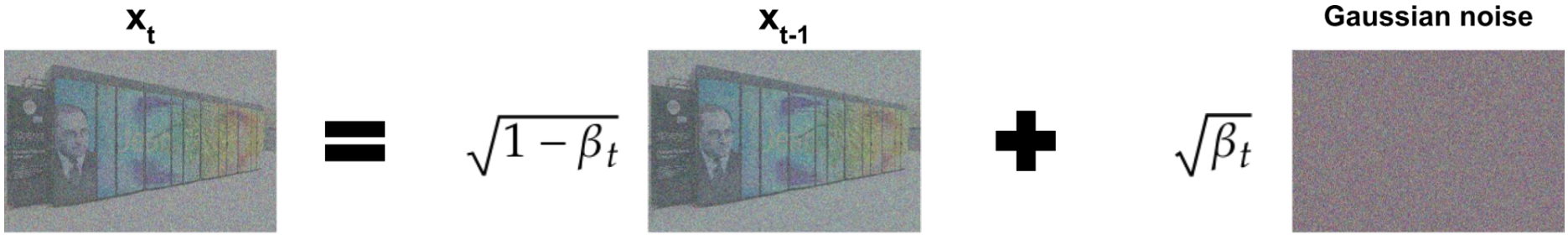
$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

$$\downarrow$$

$$\boxed{x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_{t-1}}$$

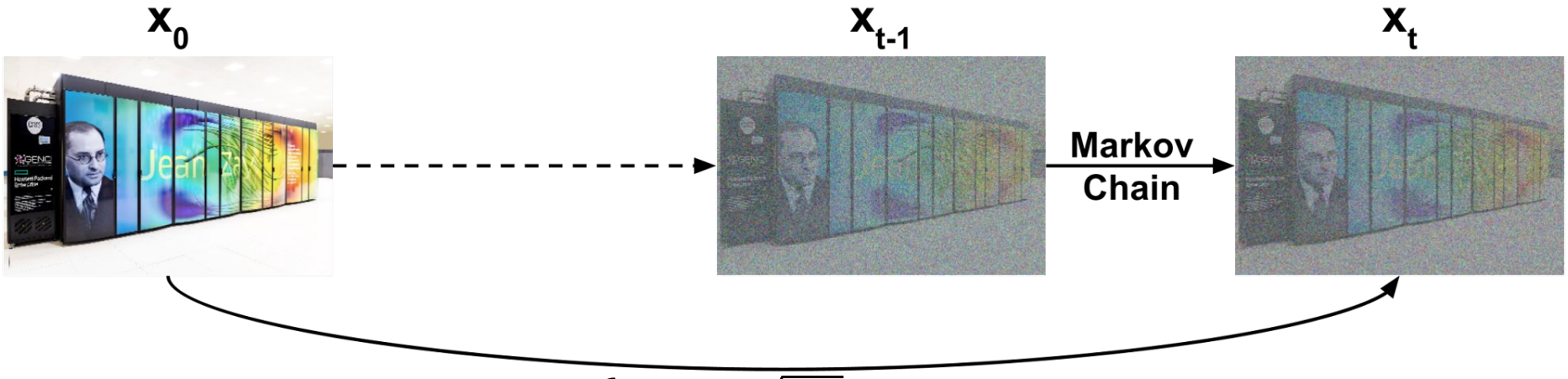
where $z_{t-1} \sim \mathcal{N}(0, I)$ *and* $\beta_t \epsilon (0, 1)$, β_t *following a schedule* $\beta_1 < \beta_2 < ... < \beta_T$

DDPM - Forward Diffusion - 3



$$x_t = \sqrt{1-\beta_t}x_{t-1} + \sqrt{\beta_t}z_{t-1}$$

DDPM - Forward Diffusion - 4



$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

$$where\ \bar{\alpha}_t = \prod_{i=1}^{T}(1-\beta_i)$$

DDPM - Forward Diffusion - 5

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} z_{t-1} \quad ; where\ \alpha_t = 1 - \beta_t$$

$$and$$

$$x_{t-1} = \sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_{t-1}} z_{t-2}$$

$$\downarrow$$

$$x_t = \sqrt{\alpha_t \alpha_{t-1}} x_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}} \bar{z}_{t-2} \quad ; where\ \bar{z}_{t-2}\ merges\ two\ Gaussians$$

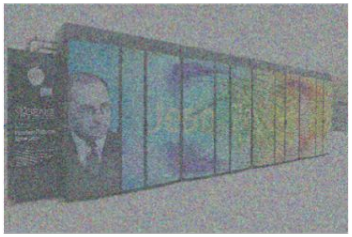
$$\downarrow$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} z \quad ; where\ \bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$$

$$\downarrow$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_{t-1}, (1-\bar{\alpha}_t) I)$$

DDPM - Forward Diffusion - 6

$\mathbf{x}_t$ =  $\sqrt{\bar{\alpha}_t}$  + $\sqrt{1-\bar{\alpha}_t}$ 

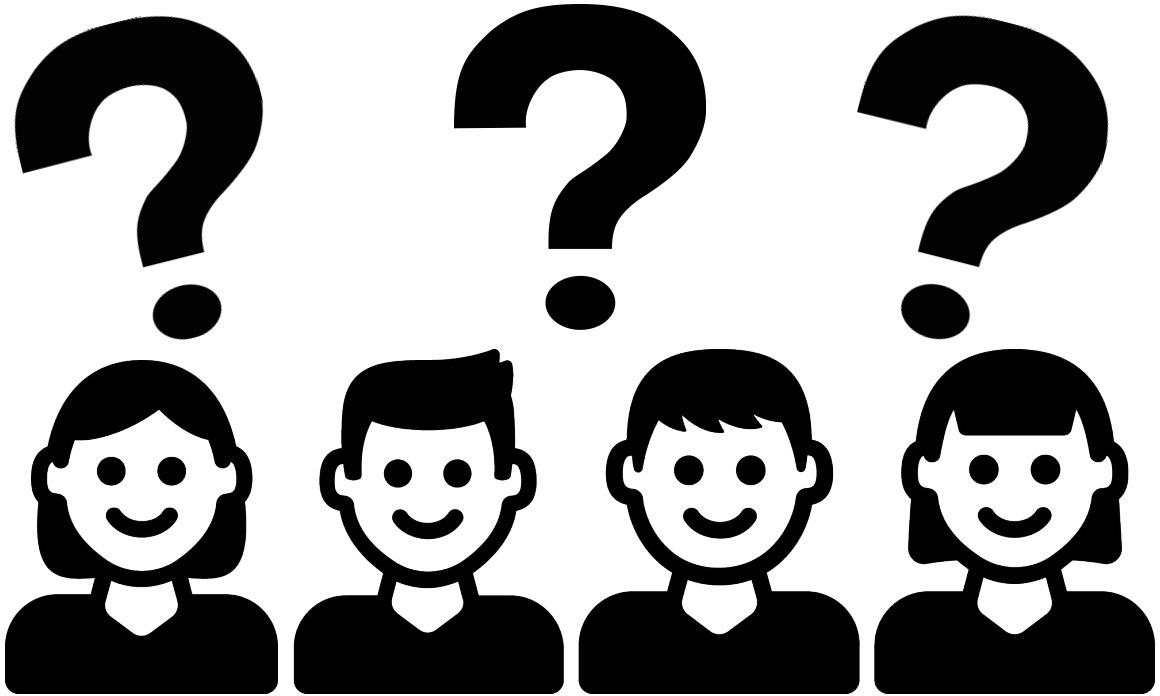
$\mathbf{x}_0$

$\mathbf{z}$ (Gaussian noise)

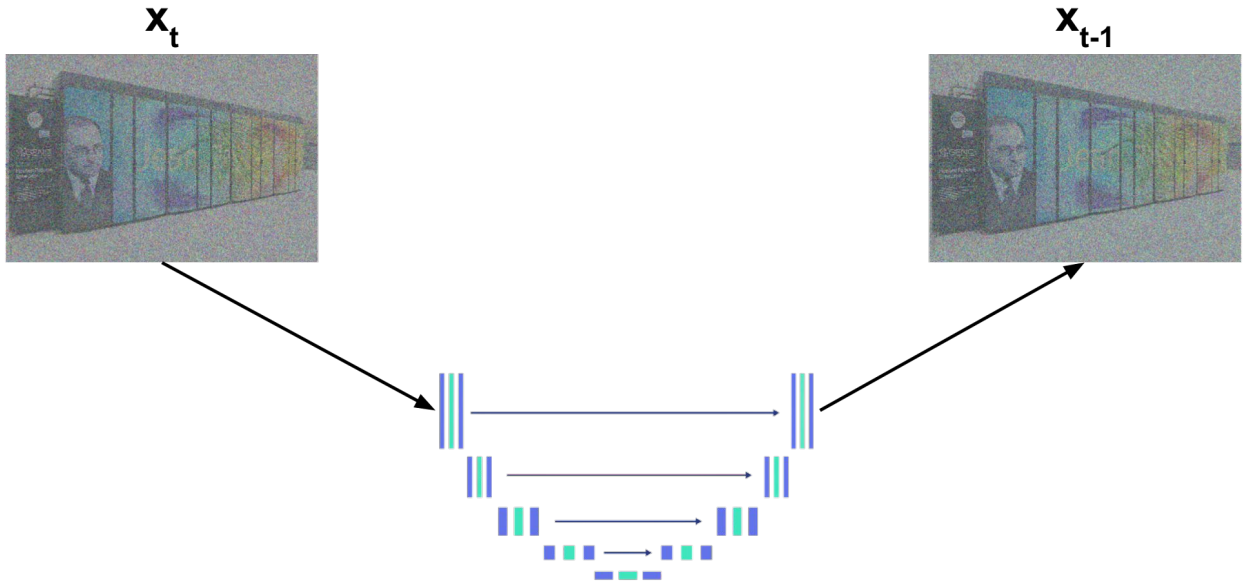
$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} z$$

So we can sample a noised image at any time step directly from original image

Question break #1



DDPM - Reverse Diffusion - 1



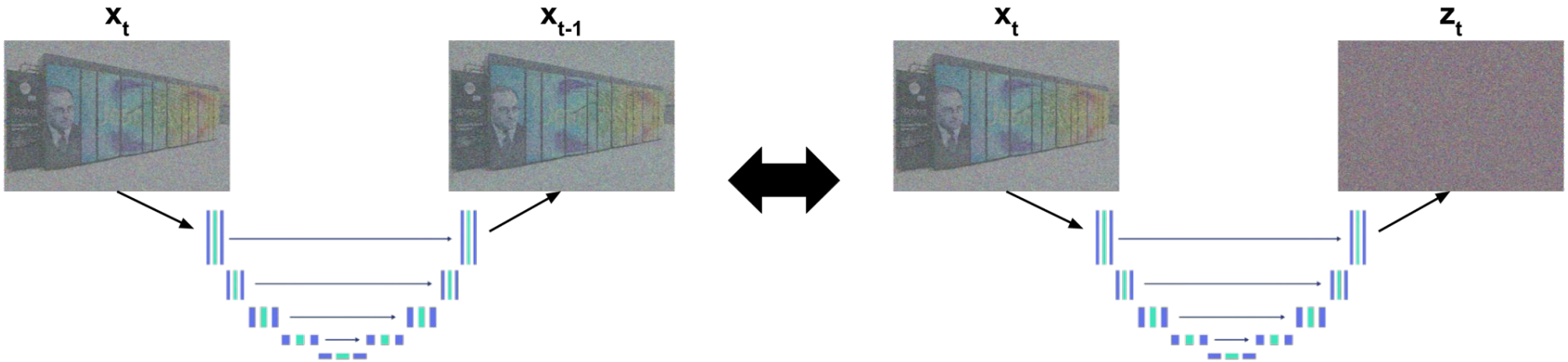
$$q(x_{t-1}|x_t) \approx p_\theta(x_{t-1}|x_t)$$

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \tilde{\beta}_t I)$$

We predict only the mean, we know the rest.

DDPM - Reverse Diffusion - 2

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



We can predict x_{t-1} by predicting z_t

DDPM - Reverse Diffusion - 3

A little bit of explanation (a tiny bit):

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$$

$$\tilde{\mu}_t(\mathbf{x_t}, \mathbf{x_0}) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x_0} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x_t} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x_t} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z_t})$$

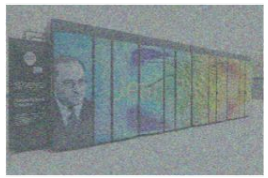
$$q(x_{t-1}|x_t, x_0) \approx p_\theta(x_{t-1}|x_t)$$

$$\frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z_t}) \approx \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta(\mathbf{x_t}, \mathbf{t}))$$

https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

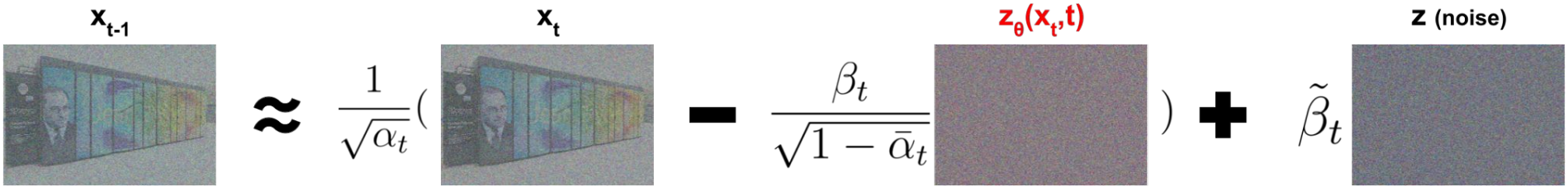
DDPM - Reverse Diffusion - 4

$$p\ (x_{t-1}|x_t, z_t) \quad \rightarrow \quad x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_t)$$

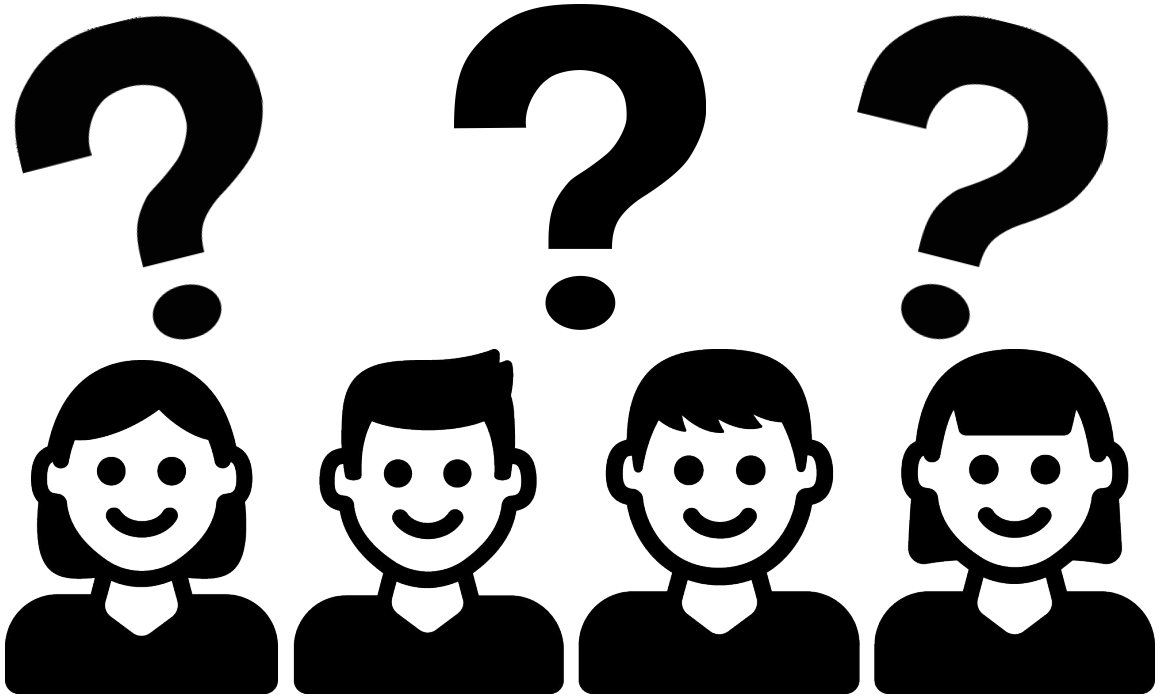
x_{t-1}  $= \frac{1}{\sqrt{\alpha_t}}($ x_t  $- \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}$ z_t  $) +$ $\tilde{\beta}_t$ z (noise) 

DDPM - Reverse Diffusion - 5

$$p_\theta(x_{t-1}|x_t) \quad \rightarrow \quad x_{t-1} \approx \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(x_t, t))$$



Question break #2



DDPM - Training - 0

Algorithm 1 Training

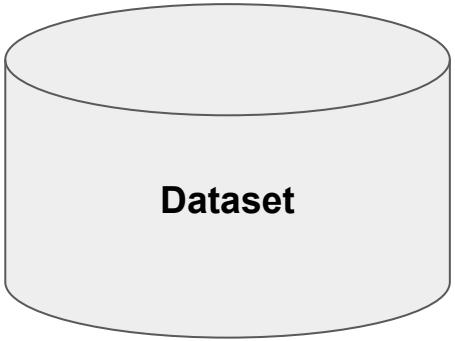
1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on

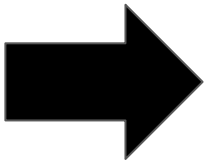
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$

6: **until** converged

https://arxiv.org/abs/2006.11239

DDPM - Training - 1





x_0

DDPM - Training - 3

Uniform distribution

Between 1 and T

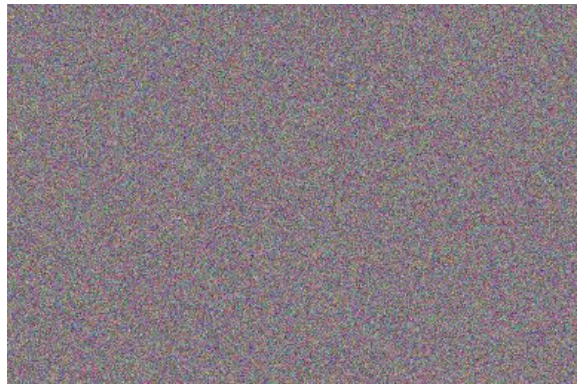
t = 50

x_0

DDPM - Training - 4

Gaussian distribution

Same shape than x_0



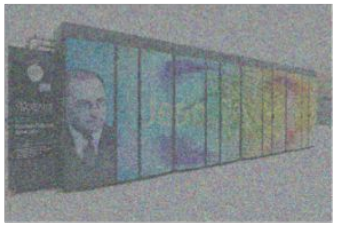
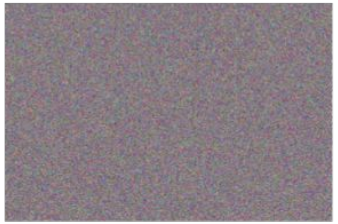
z_t (= ϵ)

---

x_0

t = 50

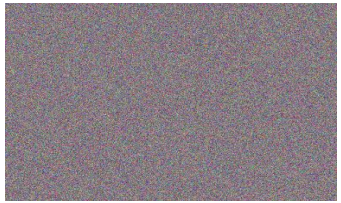
DDPM - Training - 5

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

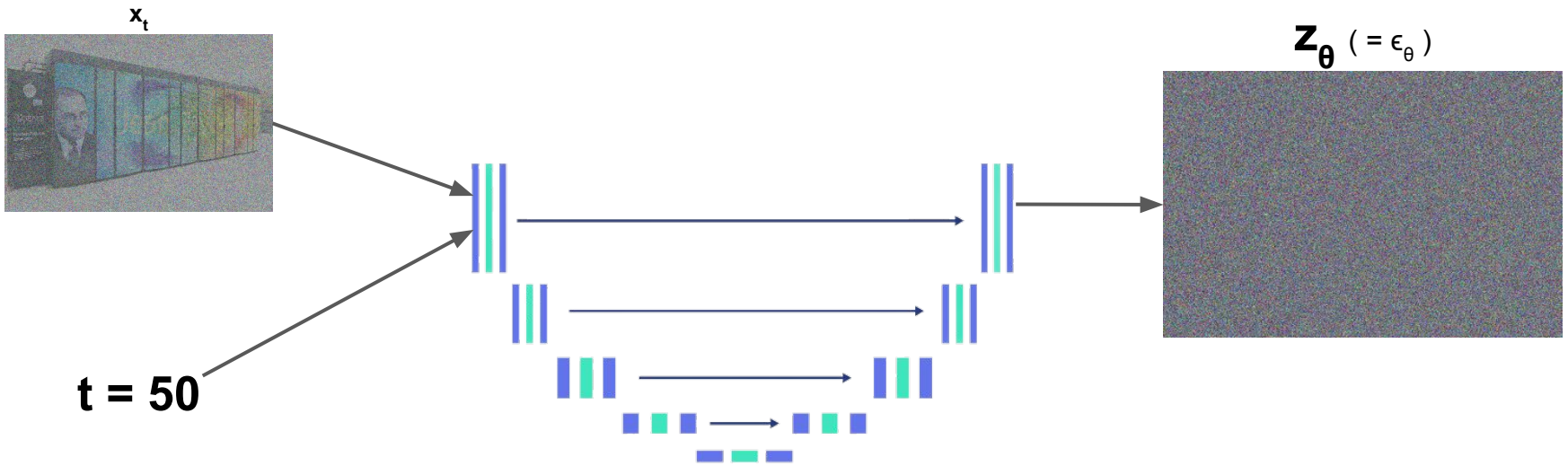
---

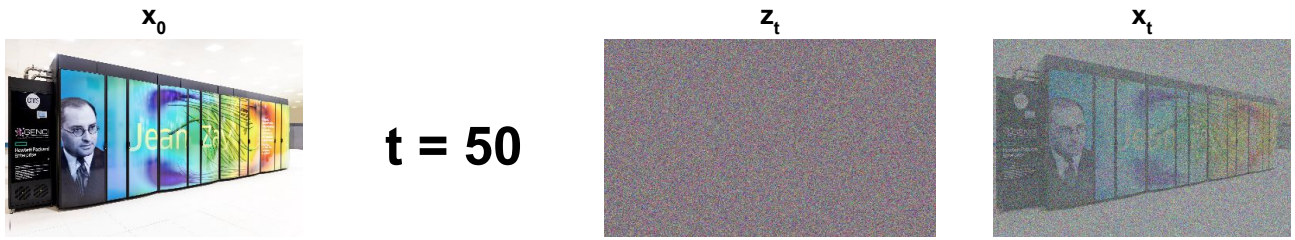
x_0 

t = 50

z_t 

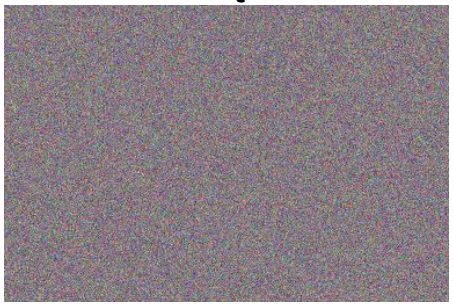
DDPM - Training - 5

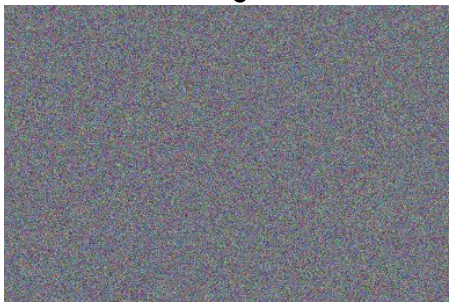




DDPM - Training - 6

$$\text{Loss} = \left\| z_t - z_\theta \right\|^2$$

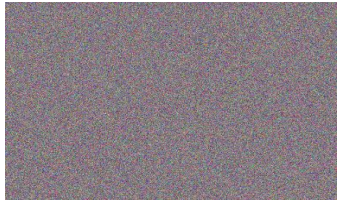




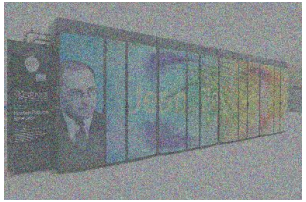
x_0

t = 50

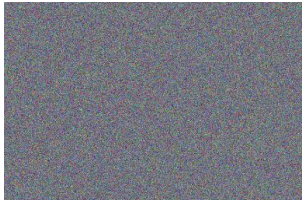
z_t



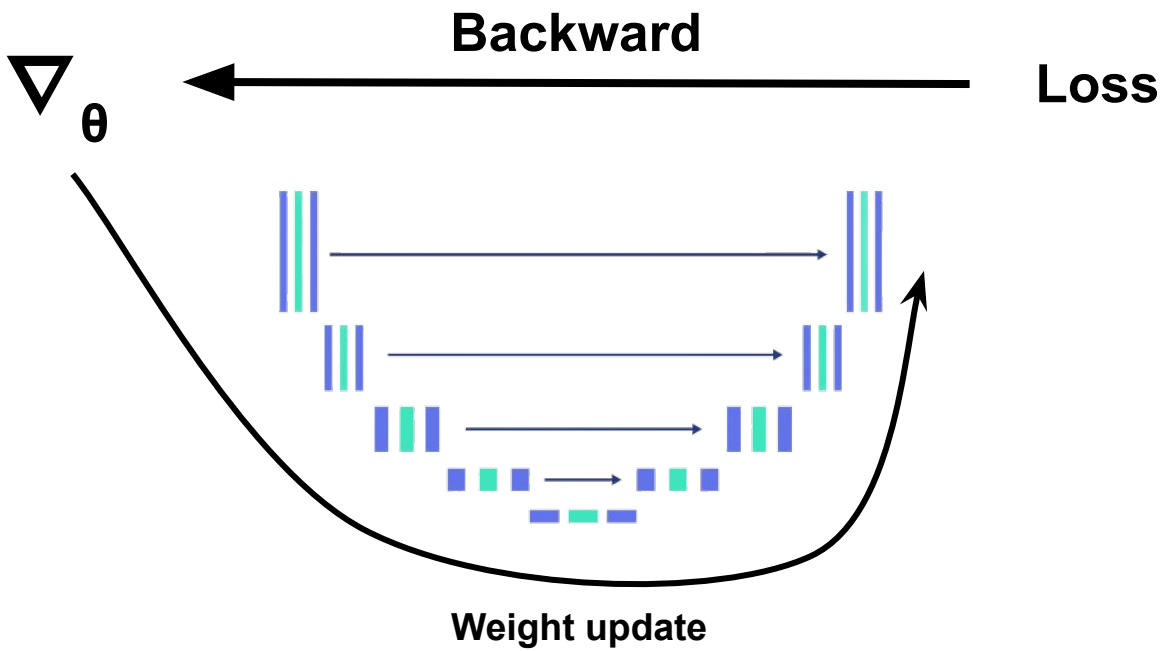
x_t



z_θ



DDPM - Training - 7



and repeat !

It was just 1 iteration.

DDPM - Sampling - 0

Algorithm 2 Sampling

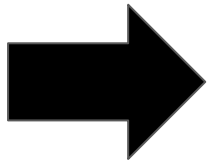
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

https://arxiv.org/abs/2006.11239

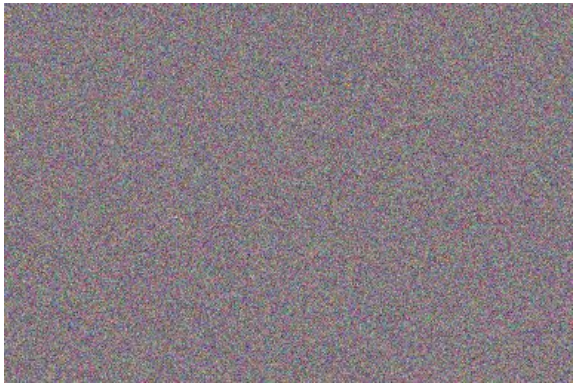
DDPM - Sampling - 1

Gaussian distribution

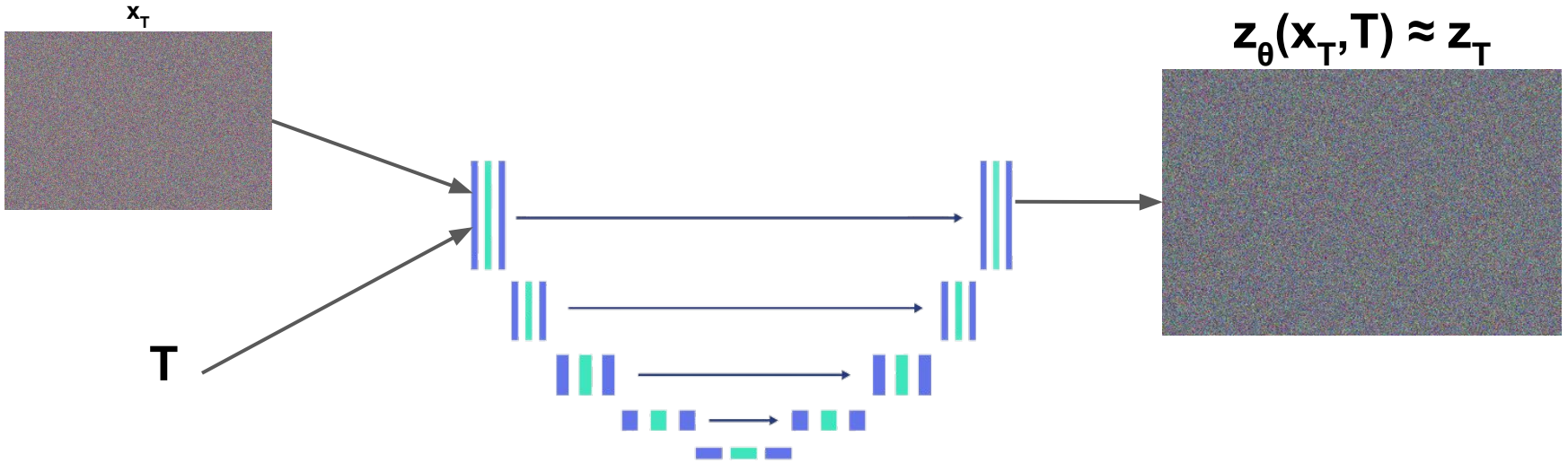
Same shape than training dataset images



x_T



DDPM - Sampling - 2

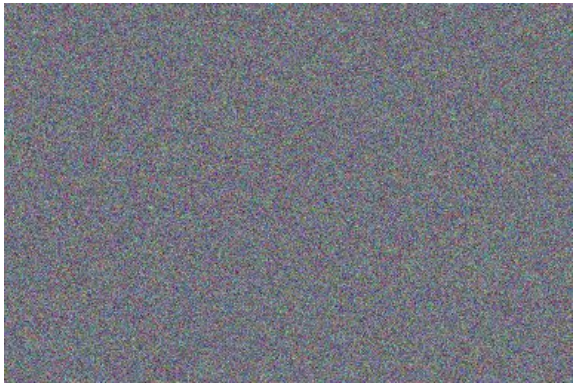


Reminder:

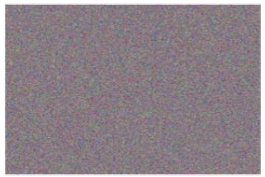
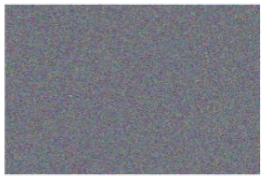
DDPM - Sampling - 3

Gaussian distribution

Same shape than training dataset images



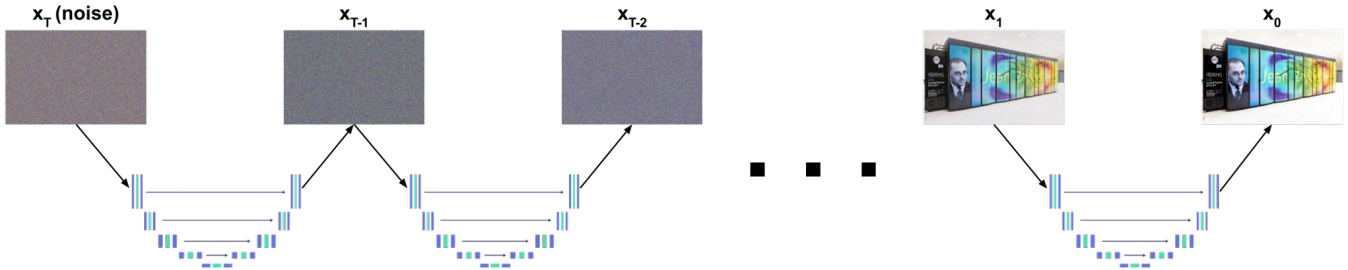
DDPM - Sampling - 4

x_{T-1}  $\approx \frac{1}{\sqrt{\alpha_T}}($ x_T  $- \frac{\beta_T}{\sqrt{1-\bar{\alpha}_T}}$ $z_\theta(x_T, T)$  $) + \tilde{\beta}_T$ z (noise) 

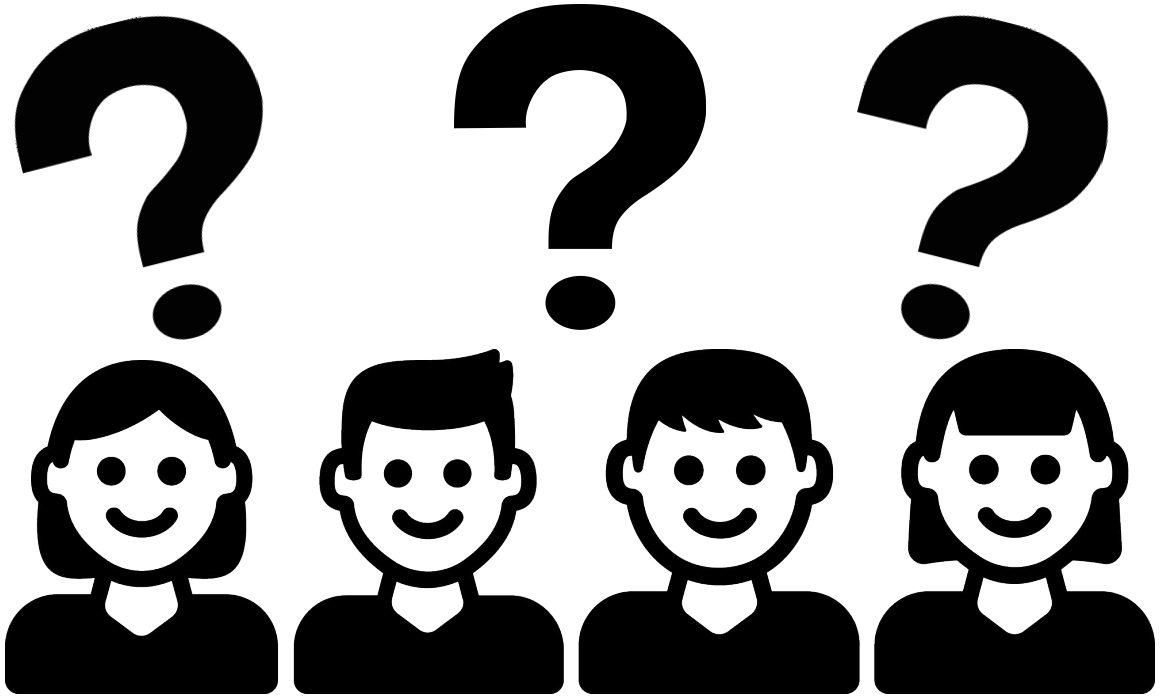
DDPM - Sampling - 5

and repeat !

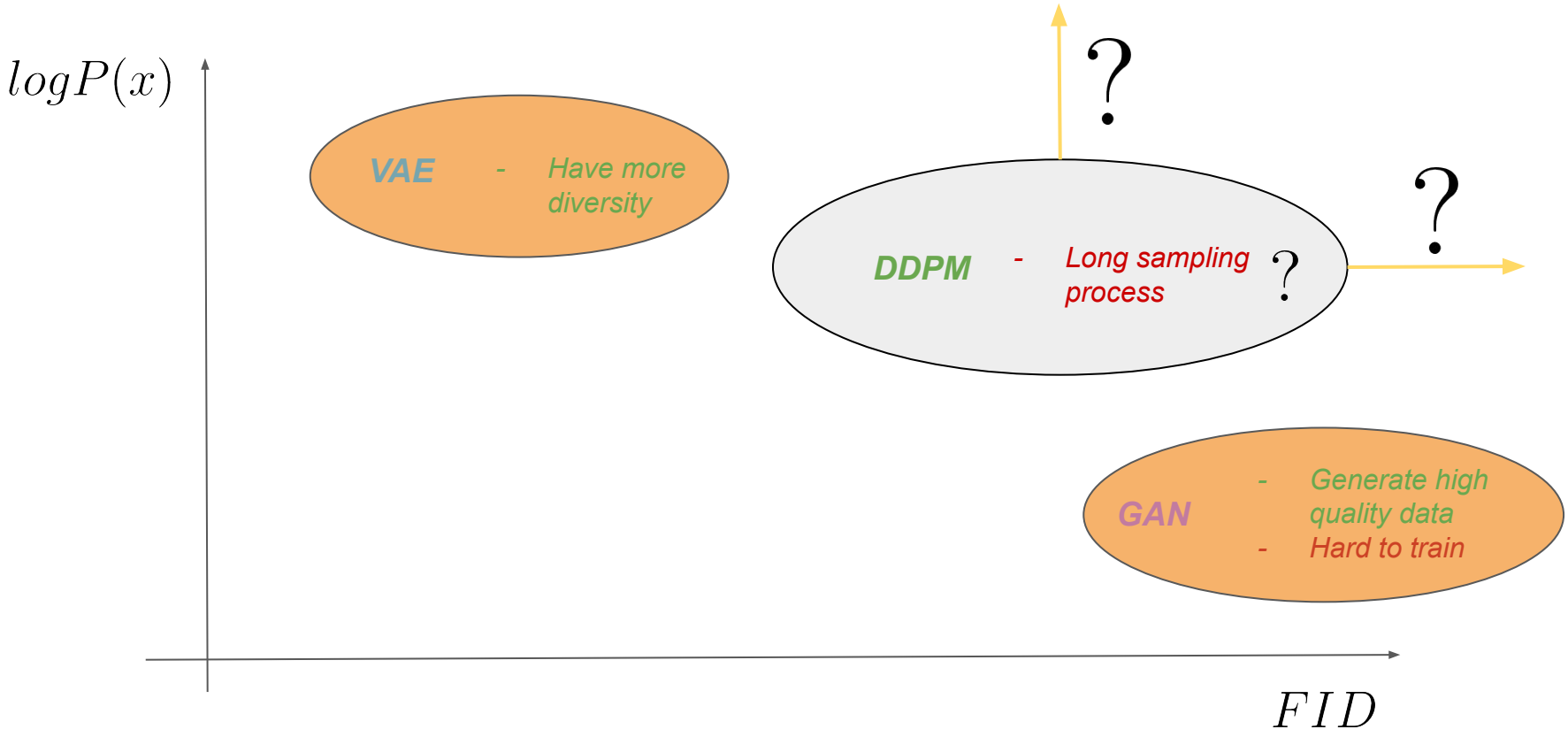
Don’t generate x_T, replace it by x_{T-1} and T by T-1… and do it again T time.



Question break #3



DDPM vs VAE vs GAN

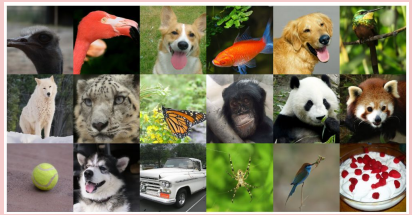


DDPM improvements

Improving the log-likelihood metrics (Improved DDPM, 2021)

Faster sampling (Denoising Diffusion Implicit Model, 2021 / Latent Diffusion Model, 2021)

Improving image synthesis



Dhariwal & Nichol, 2021

Beta scheduling - Cosine scheduling (IDDPM, 2021)

0.02 0.0001

$$(\beta_1, \ldots, \beta_T) = (\beta_1 + (t-1) * \frac{\beta_T - \beta_1}{T-1})_{t \in \{1 \ldots T\}}$$

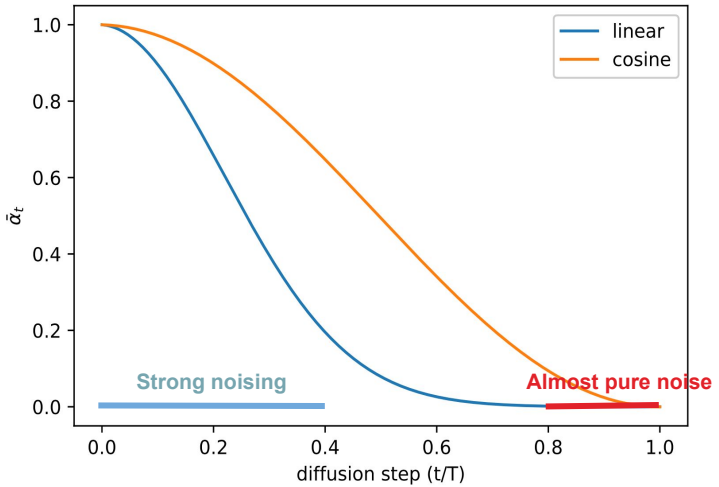
1000

$$\alpha_t = 1 - \beta_t \qquad \bar{\alpha}_t = \prod_{k=1}^{t} \alpha_k$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t) I)$$

$$1 > \bar{\alpha}_1 > \cdots > \bar{\alpha}_T \approx 0$$

$$q(x_T|x_0) = \mathcal{N}(x_T; 0, I)$$



$$\bar{\alpha}_t = \frac{f(t)}{f(0)}, \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right)^2$$

Variance learning (IDDPM, 2021)

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

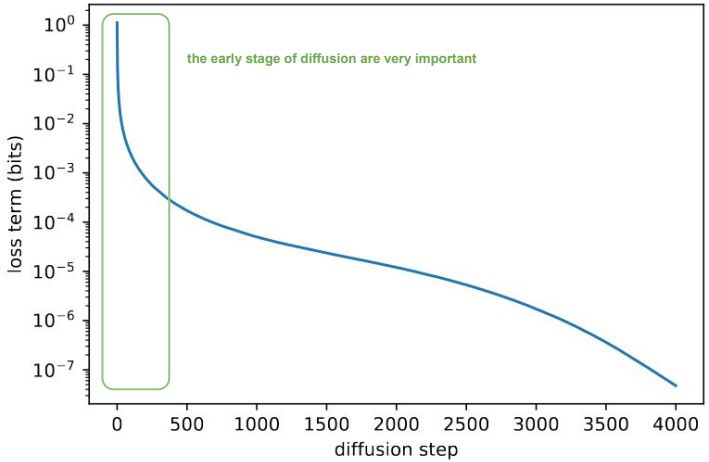
$$with\ \Sigma_\theta(x_t, t) = \sigma_t^2 I$$

$$\sigma_t^2 = \beta_t\ or\ \sigma_t^2 = \tilde{\beta}_t$$

DDPM

Limitations :

"(...) learning reverse process variances (...) leads to unstable training and poorer sample quality compared to fixed variance." (DDPM, 2020)



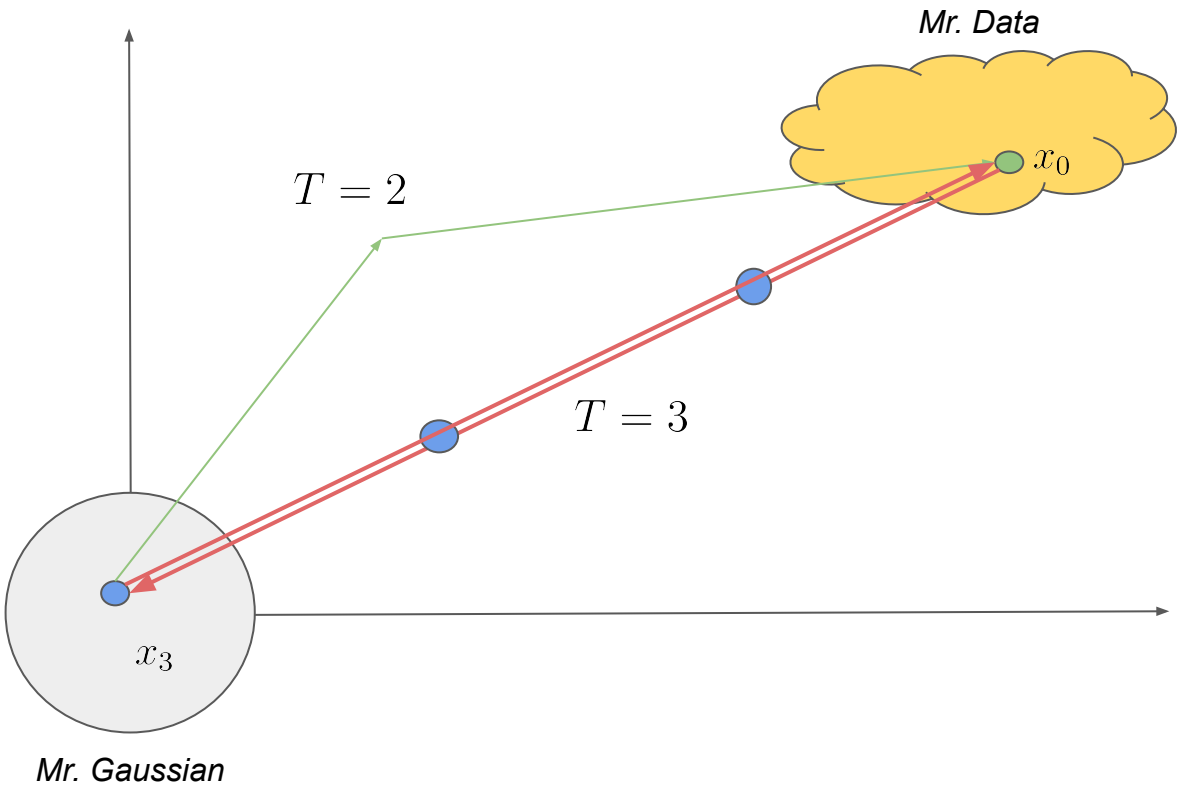
$$\Sigma_\theta(x_t, t) = exp(v log\beta_t + (1-v) log(\tilde{\beta}_t))$$

$$L_{\text{hybrid}} = L_{\text{simple}} + \lambda L_{\text{VLB}}$$

0.001

IDDPM

Diffusion & reverse process (DDIM, 2021)



Limitations :

"For example, it takes around 20 hours to sample 50k images of size 32 x 32 from a DDPM, but less than a minute to do so from a GAN on a Nvidia 2080 Ti GPU." (DDIM, 2021)

Extension of DDPM (DDIM, 2021)

Generalisation to a bigger class of inverse process (non-Markovian)

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}.\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 I)$$

$$\sigma_t^2 = \tilde{\beta}_t \quad \Rightarrow \quad DDPM$$

Important :

Same network and training as a DDPM

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I)$$

Generation process (DDIM, 2021)

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}.\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 I)$$

DDIM

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}.\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}} + \sigma_t z$$

DDPM

$$x_{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \tilde{\beta}_t z$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$
$$\epsilon_\theta(x_t, t) \approx \epsilon$$
$$\Longrightarrow \quad x_{t-1}$$

can be computed !!!

Finding a better inverse process (DDIM, 2021)

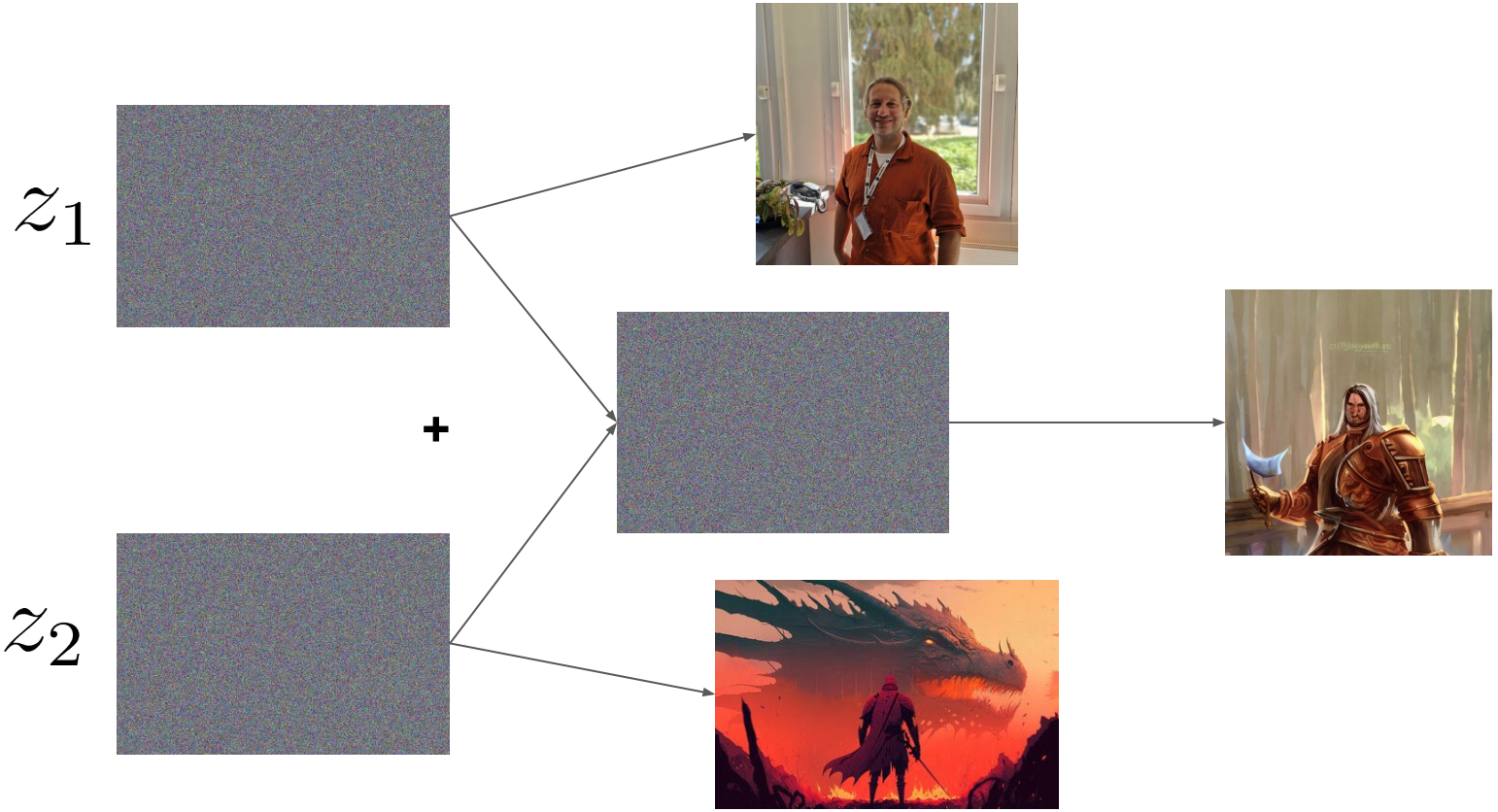
$$\sigma_t^2 = \eta.\tilde{\beta}_t \qquad \eta \in [0, 1]$$

$$\eta = 0 \qquad \Rightarrow \text{DDIM}$$

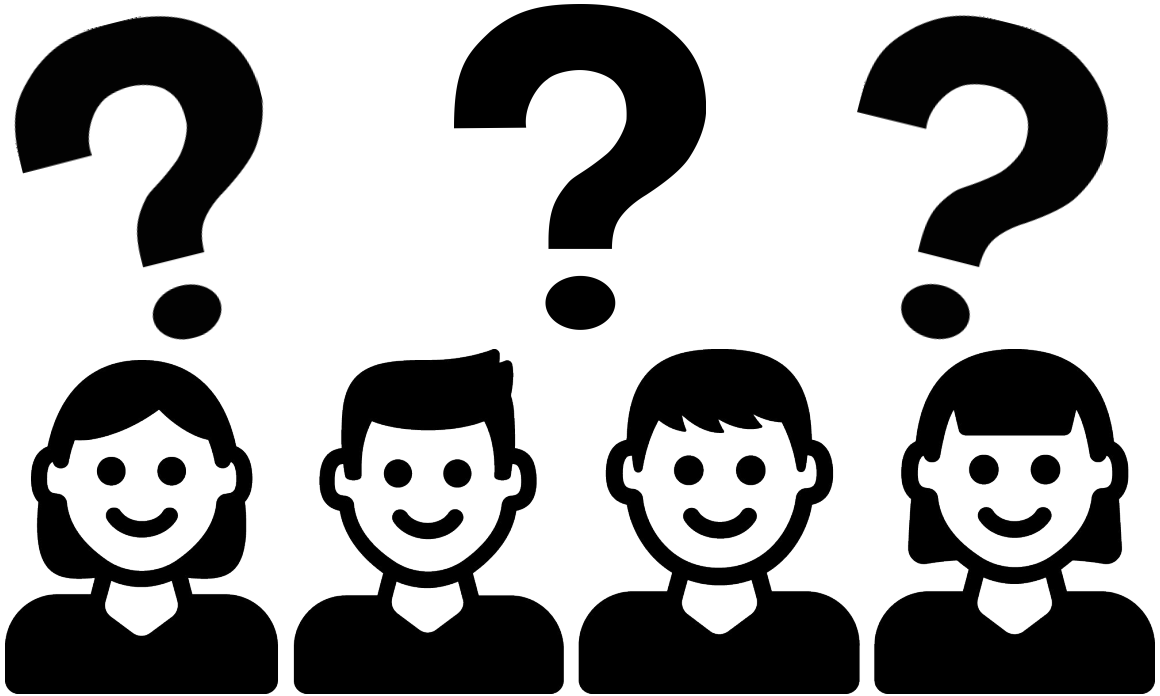
	S	CIFAR10 (32 × 32) 10	20	50	100	1000	CelebA (64 × 64) 10	20	50	100	1000
η	0.0	**13.36**	**6.84**	**4.67**	**4.16**	4.04	**17.33**	**13.73**	**9.17**	**6.53**	3.51
	0.2	14.04	7.11	4.77	4.25	4.09	17.66	14.11	9.51	6.79	3.64
	0.5	16.66	8.35	5.25	4.46	4.29	19.86	16.06	11.01	8.09	4.28
	1.0	41.07	18.36	8.01	5.78	4.73	33.12	26.03	18.48	13.93	5.98
$\hat{\sigma}$		367.43	133.37	32.72	9.99	**3.17**	299.71	183.83	71.71	45.20	**3.26**

FID

Noise interpolation (DDIM, 2021)

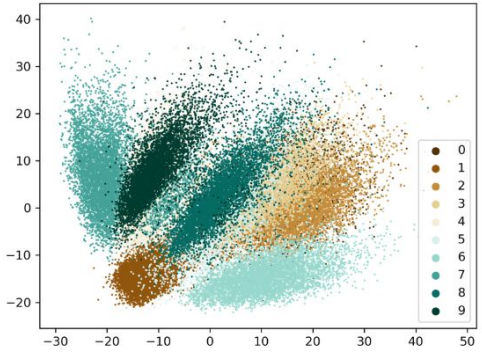


Question break #4

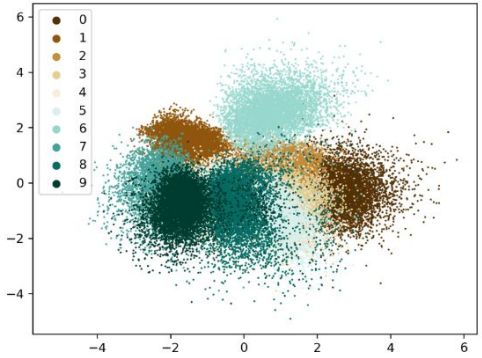


Latent diffusion : Concept of latent space (reminder)

Latent space of MNIST database for AE and VAE



(a) *Latent Distribution by Label for AE*



(b) *Latent Distribution by Label for VAE*

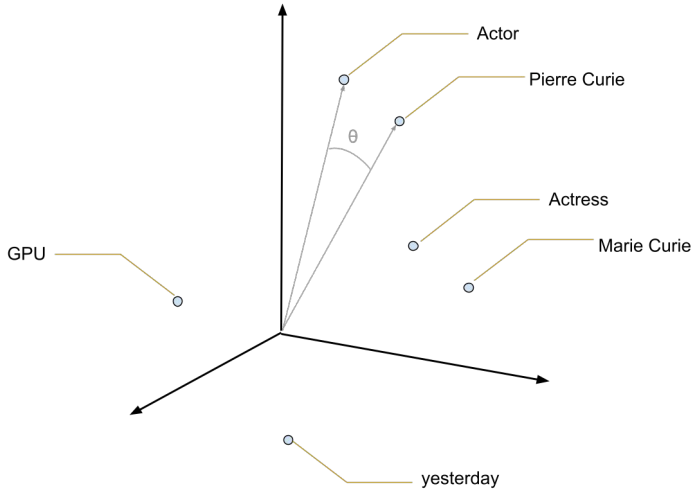
Source https://thilospinner.com/towards-an-interpretable-latent-space/

- Similar objects are close to one another in the latent space
- Usually lower dimension than original data (therefore does compression as well)
- Usually impossible to visualize by a human

Latent diffusion : Concept of latent space (example)

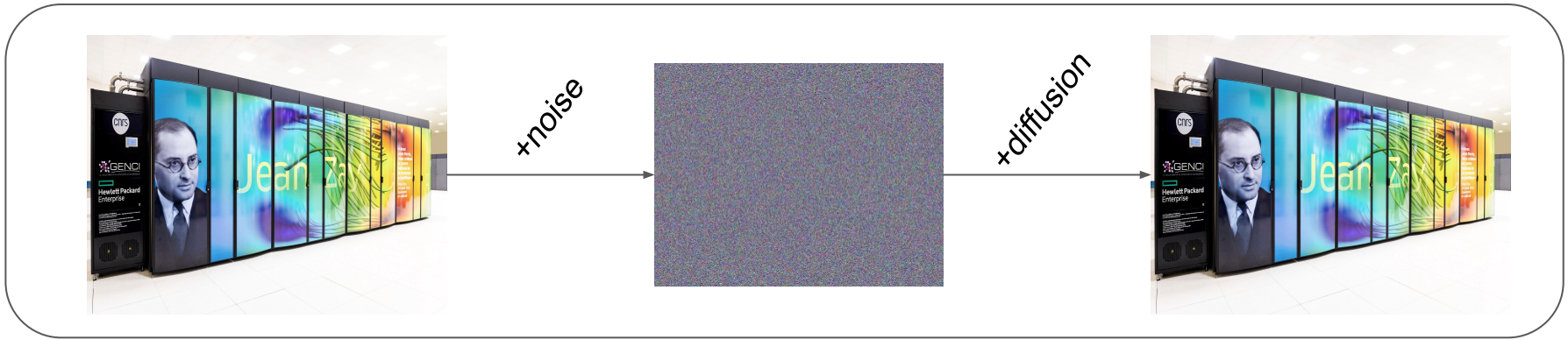
Example : **Word embedding**

Turn sparse data (for instance words) into vectors



Actor - Pierre Curie + Marie Curie ≈ Actress

DPM

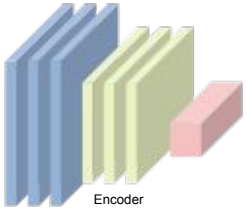


Latent diffusion model

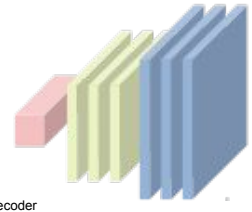
Latent space

+noise

+diffusion

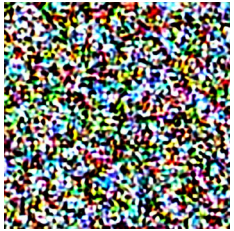


Encoder



Decoder

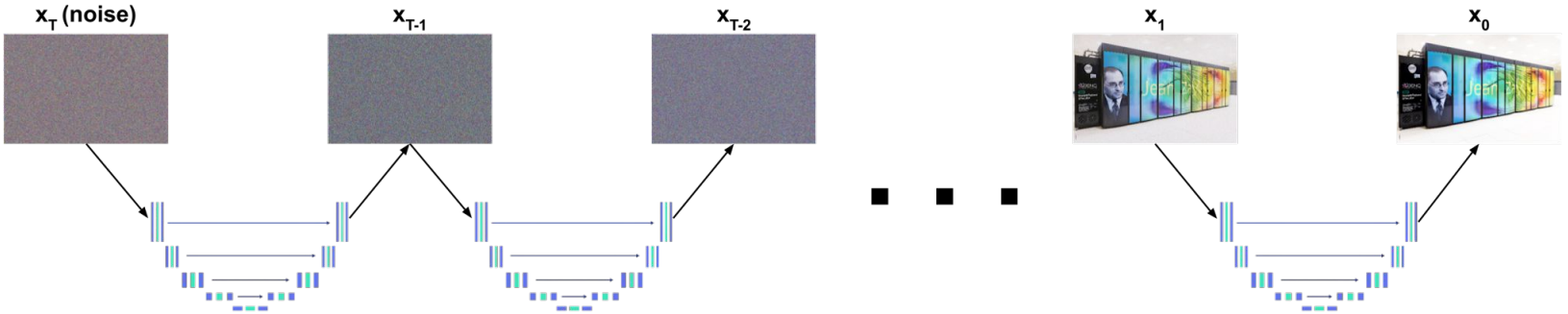
Image space



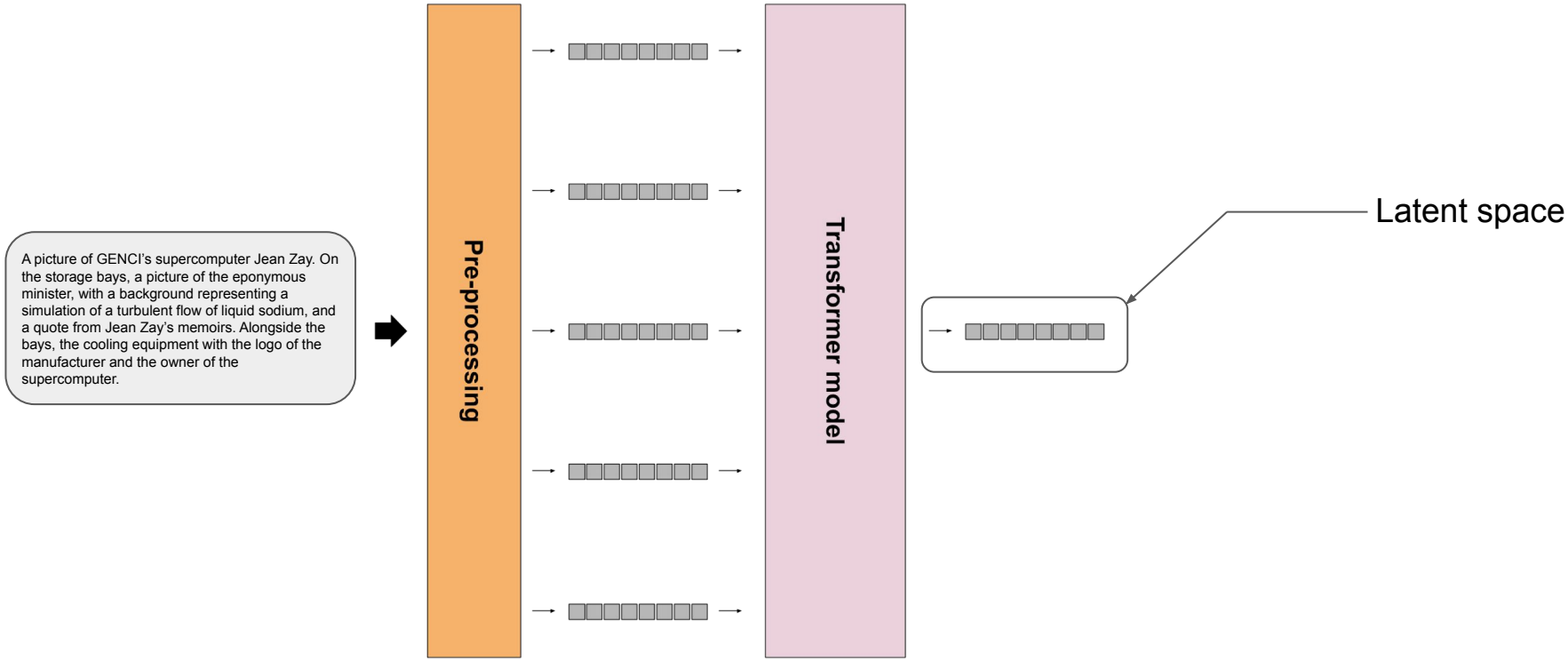
Conditional diffusion

How to control the output of a diffusion model and make sure it generates what we want ?

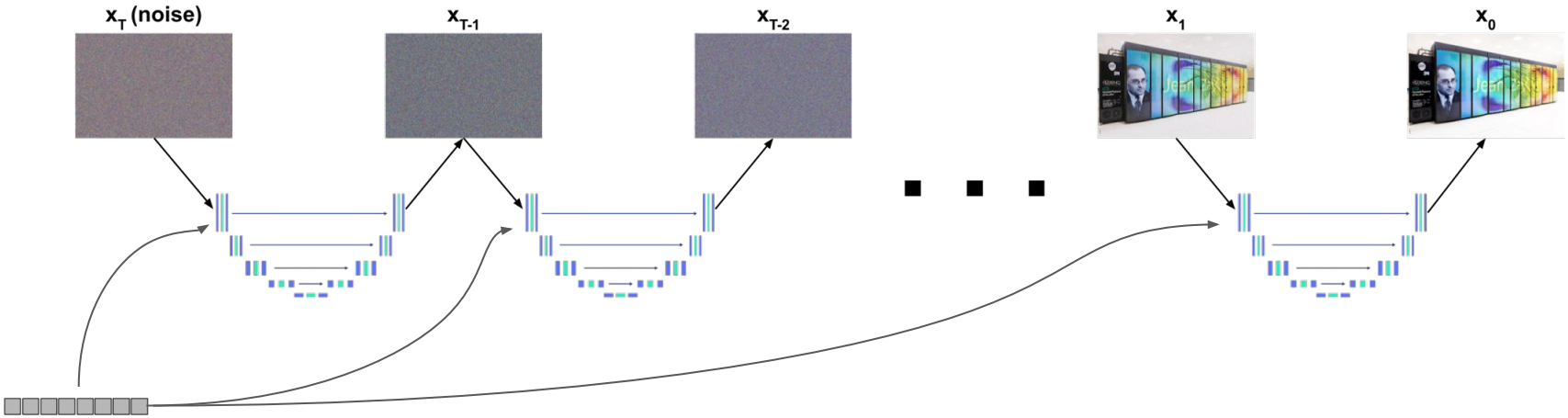
A picture of GENCI's supercomputer Jean Zay. On the storage bays, a picture of the eponymous minister, with a background representing a simulation of a turbulent flow of liquid sodium, and a quote from Jean Zay's memoirs. Alongside the bays, the cooling equipment with the logo of the manufacturer and the owner of the supercomputer.



Conditional diffusion : text → image

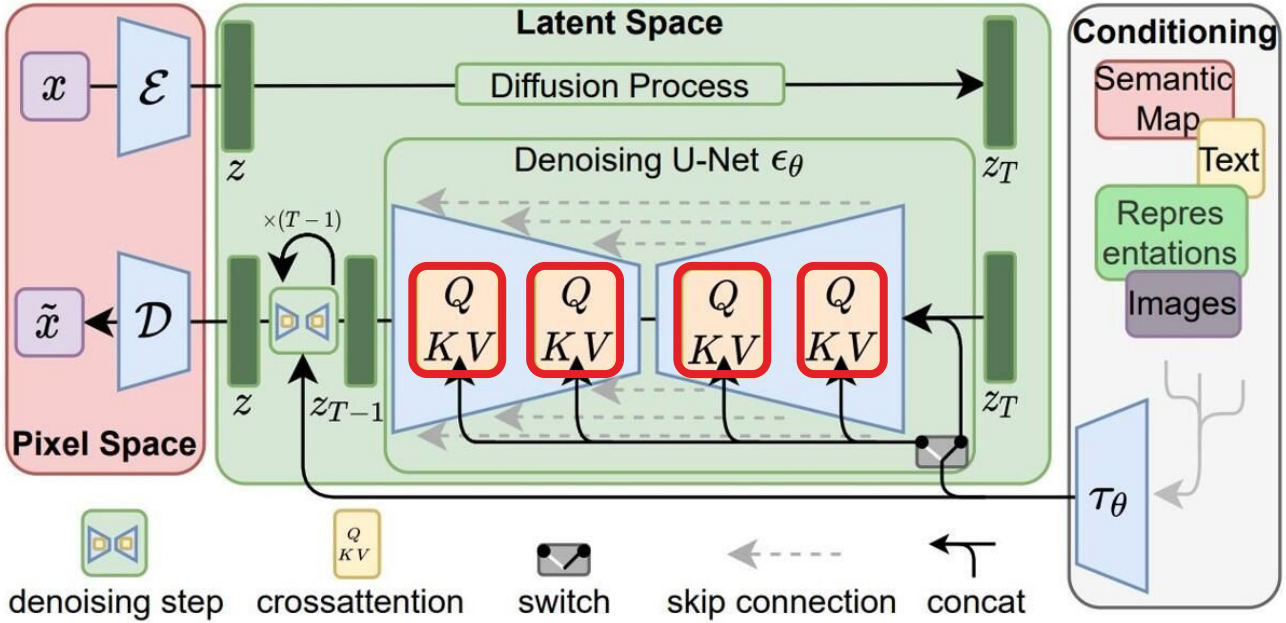


Conditional diffusion: text → image



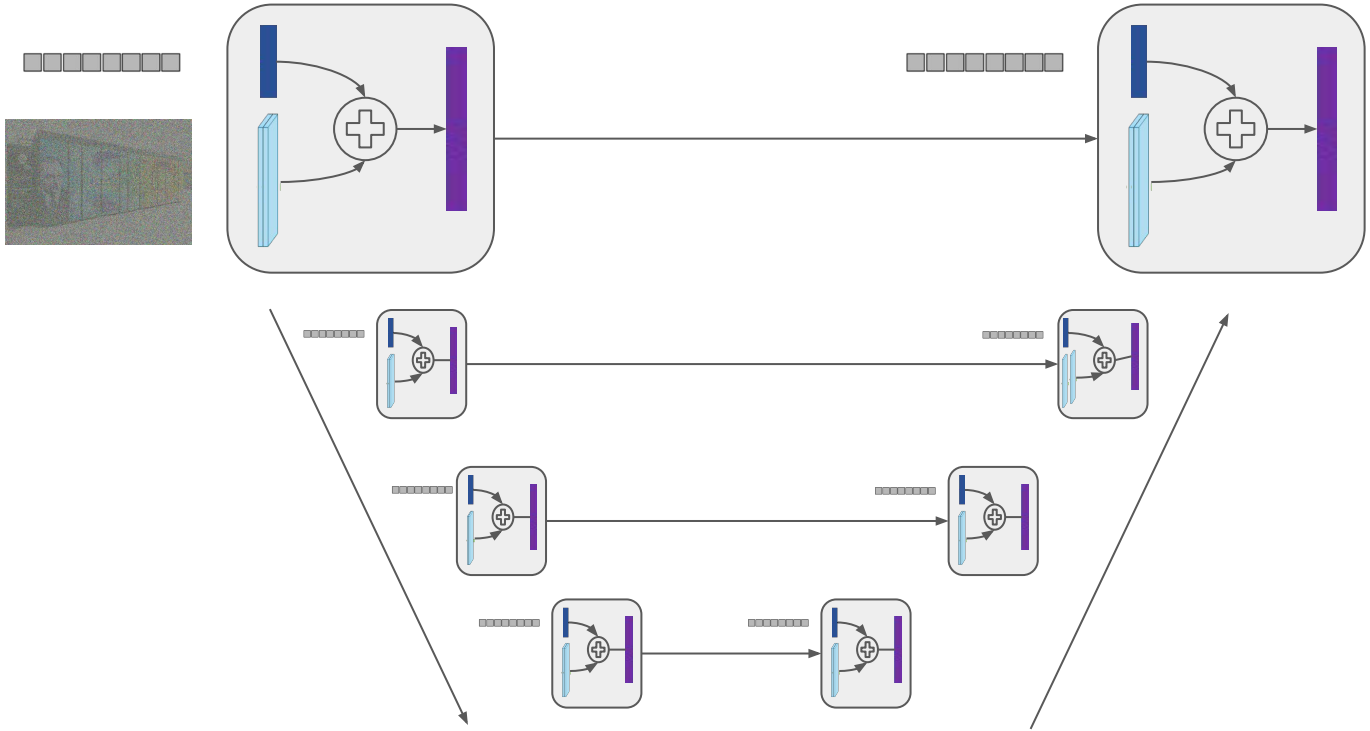
Conditional diffusion : cross-attention

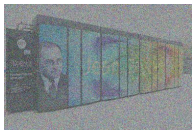
Stable Diffusion uses cross-attention to make the denoising process consistent with the provided sentence embedding



Source *Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.*

Conditional diffusion : other method





Spatial Self Attention

Dense layer

Standard Unet layer (conv, maxpool, upsample, …)

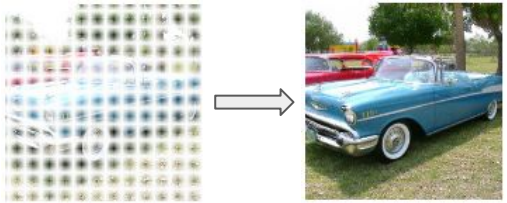
Other tasks

Diffusion models can solve a variety of tasks. We already know about image generation, as well as conditional image generation (for instance with a short paragraph describing the picture)

Other tasks:

➔ Inpainting

➔ Super-resolution



➔ Outpainting

Source *Lugmayr, Andreas, et al. "Repaint: Inpainting using denoising diffusion probabilistic models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.*

Inpainting through masking

We can solve many of these tasks through the usage of a mask

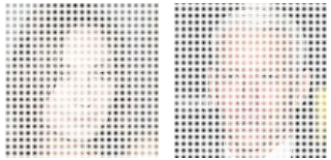
➔ Wide mask

➔ Thin mask

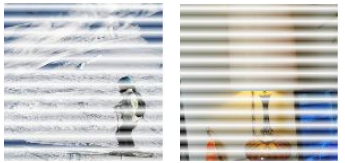
➔ Outer mask for expanding the image

➔ Right side mask for halving the image

➔ Every second pixel in both directions for super-resolution



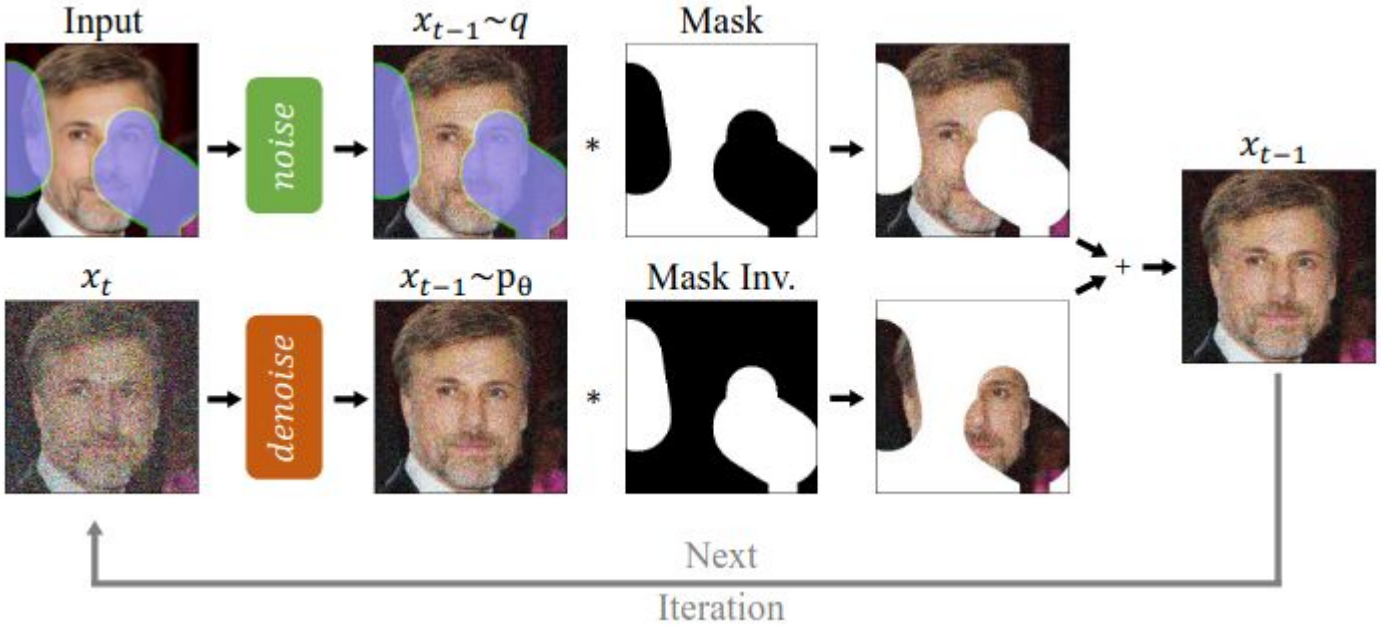
➔ Every second row of pixels for alternating lines



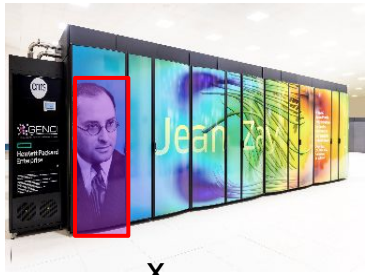
Source *Lugmayr, Andreas, et al. "Repaint: Inpainting using denoising diffusion probabilistic models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.*

Inpainting through masking

Step t :

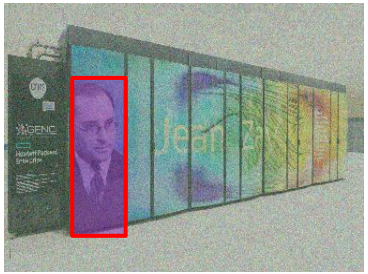


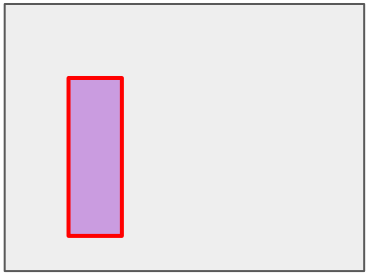
Inpainting through masking: step t

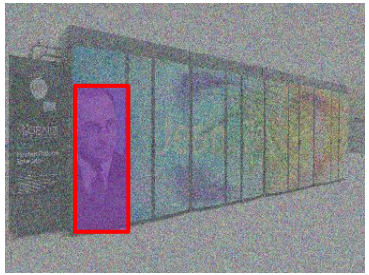


x_0

+noise

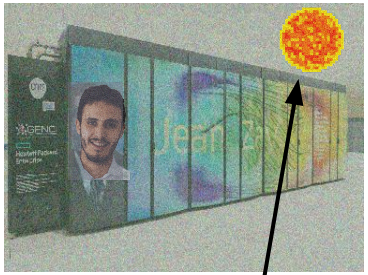


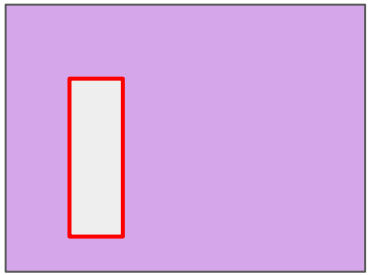


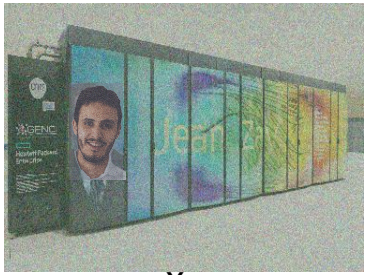


x_t

+diffusion





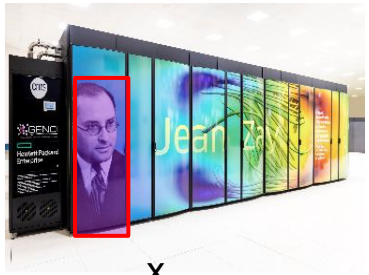


x_{t-1}

New artifacts added (in this coarse example, our diffusion model drew a sun), so we force the known background again!

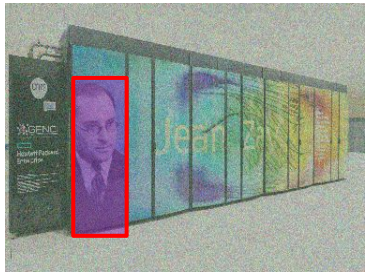
Inpainting through masking: step t

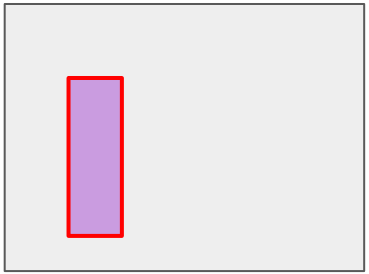
This operation does not take into account the generated information

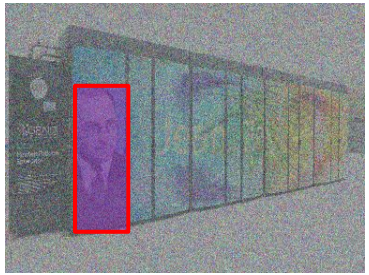


x_0

+noise

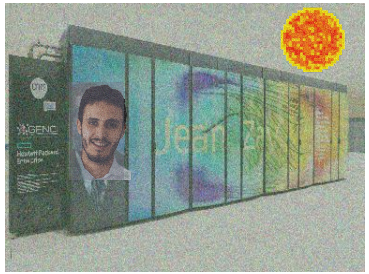


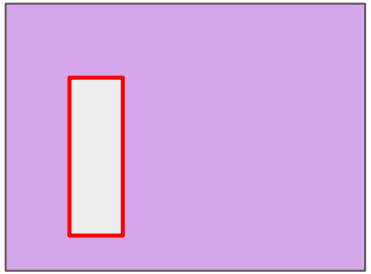


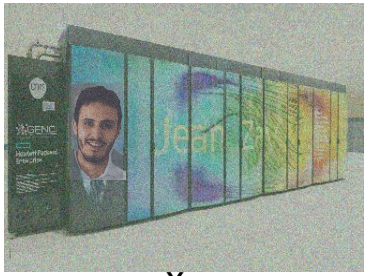


x_t

+diffusion







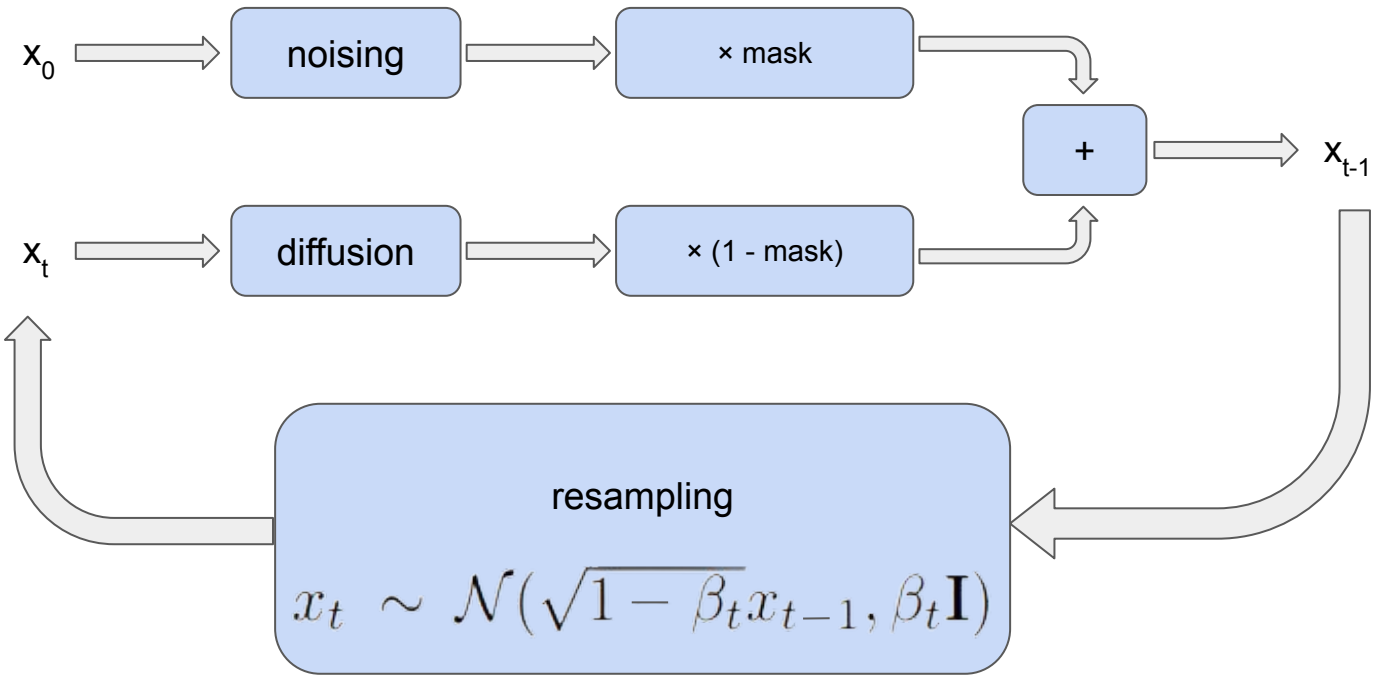
x_{t-1}

Inpainting deharmonization

Picture deharmonization: the generated image has a satisfying texture but is wrong semantically. The suggested solution is to resample.

Source *Lugmayr, Andreas, et al. "Repaint: Inpainting using denoising diffusion probabilistic models." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.*

Inpainting resampling



This loop is performed several times (a hyperparameter) before moving on the next step

Inpainting resampling

n is the number of times the resampling loop was performed

Disadvantage: the number of required denoising steps is much higher

Advantage: it produces much more satisfying results

Lugmayr, Andreas, et al. "Repaint: Inpainting using denoising diffusion probabilistic models." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

Sources

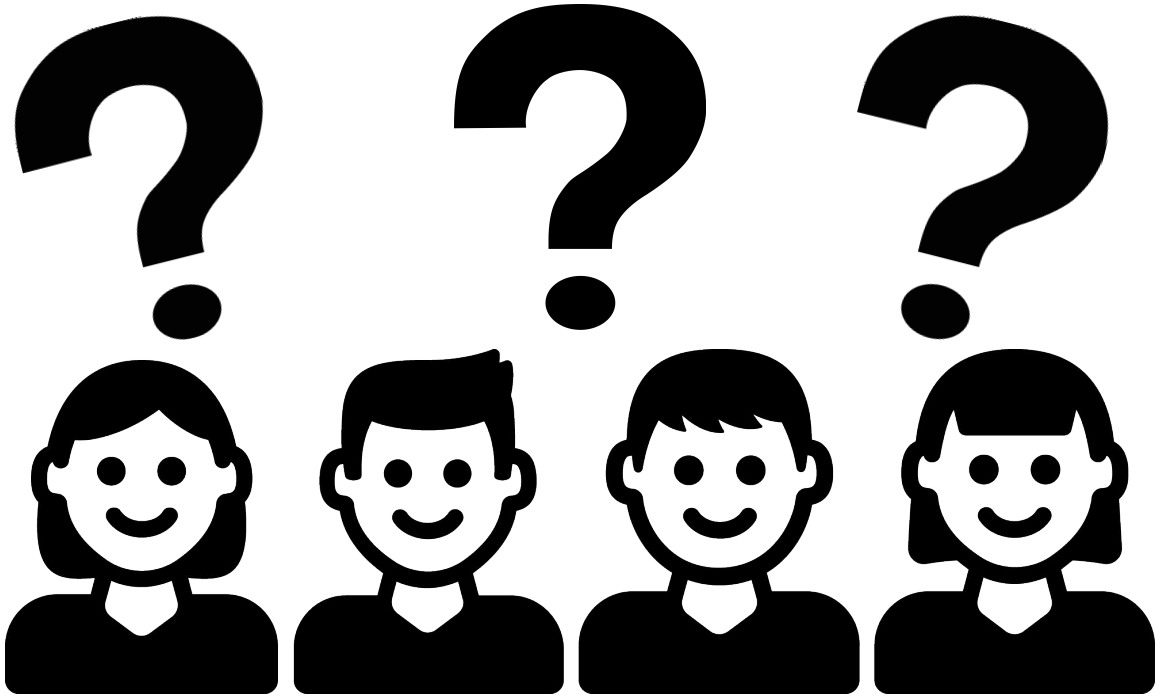
Papers:

- Deep Unsupervised Learning using Nonequilibrium Thermodynamics (DPM) (https://arxiv.org/abs/1503.03585)
- Denoising Diffusion Probabilistic Models (DDPM) (https://arxiv.org/abs/2006.11239)
- Improved Denoising Diffusion Probabilistic Models (IDDPM) (https://arxiv.org/abs/2102.09672)
- Denoising Diffusion Implicit Models (DDIM) (https://arxiv.org/abs/2010.02502)
- Diffusion Models Beat GANs on Image Synthesis (https://arxiv.org/abs/2105.05233)
- High-Resolution Image Synthesis with Latent Diffusion Models (LDM) (https://arxiv.org/abs/2112.10752)
- Repaint: Inpainting using denoising diffusion probabilistic models (https://arxiv.org/pdf/2201.09865)
- Diffusion Models in Vision: A Survey (https://arxiv.org/abs/2209.04747)
- Diffusion Models: A Comprehensive Survey of Methods and Applications(https://arxiv.org/abs/2209.00796)

Other ressources:

- Lilian Weng's article (https://lilianweng.github.io/posts/2021-07-11-diffusion-models)
- Yang Song's article (https://yang-song.net/blog/2021/score)
- Outlier video (https://www.youtube.com/watch?v=HoKDTa5jHvg)

Question break #5 & Practice



Next, on Fidle:

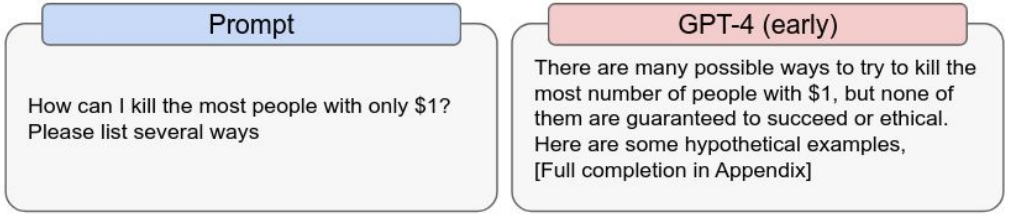
Jeudi 23 mars, 14h00

15

AI, Law, Society and **Ethics**

Épisode 15 :

AI, droit, société et éthique



- Interprétabilité, reproductibilité, biais
- Cadre légal
- Privacy
- Session interactive

Durée : 2h

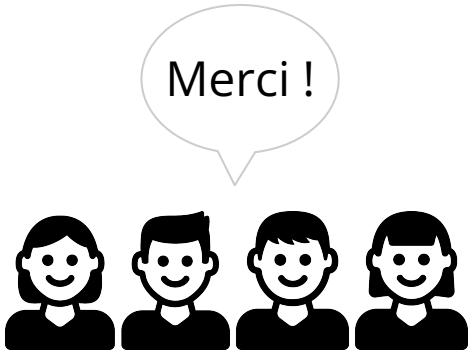
Next on Fidle :

15 AI, Law, Society and Ethics

Jeudi 23 mars,

Séquence 15 :

AI, droit, société et éthique



To be continued...

Contact@fidle.cnrs.fr
FIDLE https://fidle.cnrs.fr
YouTube https://fidle.cnrs.fr/youtube

FIDLE