



INSTITUT DU  
DÉVELOPPEMENT ET DES  
RESSOURCES EN  
INFORMATIQUE  
SCIENTIFIQUE



# Diffusion Model

AI DevTalks #5



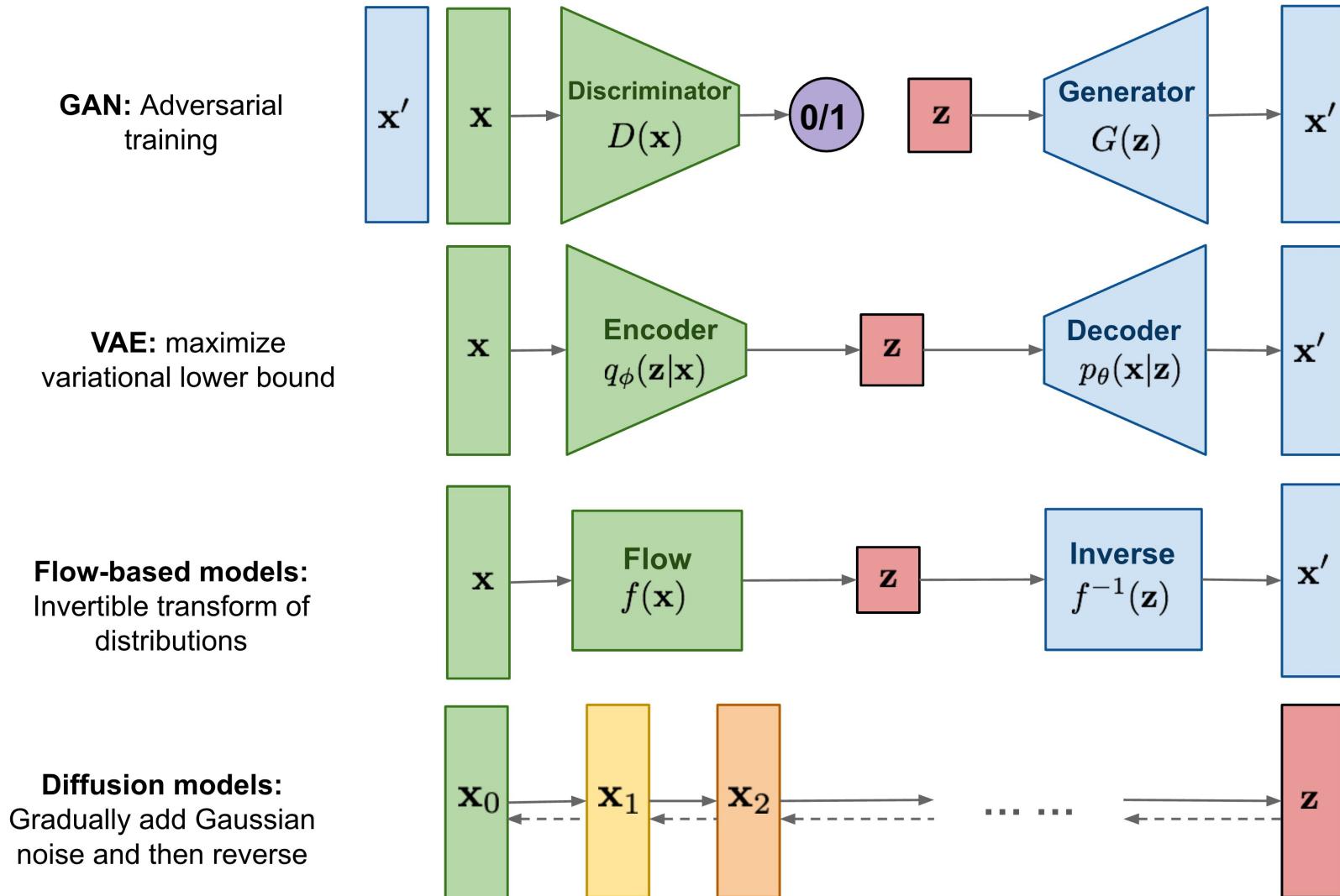
# Diffusion Probabilistic Model



A man teaching diffusion model to people in front of a whiteboard, anime style



# Generative models





# Why diffusion model ?



Sprouts in the shape of text 'Imagen' coming out of a fairytale book.



A photo of a Shiba Inu dog with a backpack riding a bike. It is wearing sunglasses and a beach hat.



A high contrast portrait of a very happy fuzzy panda dressed as a chef in a high end kitchen making dough. There is a painting of flowers on the wall behind him.



Teddy bears swimming at the Olympics 400m Butterfly event.



A cute corgi lives in a house made out of sushi.



A cute sloth holding a small treasure chest. A bright golden glow is coming from the chest.



A brain riding a rocketship heading towards the moon.



A dragon fruit wearing karate belt in the snow.



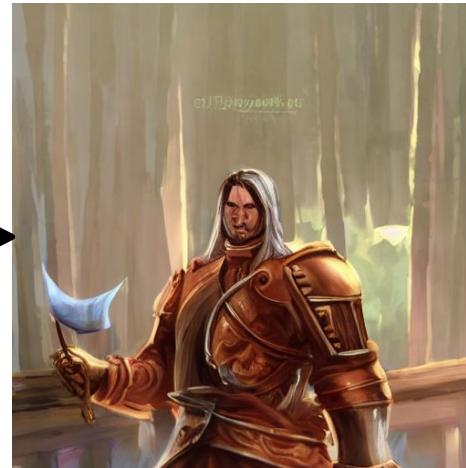
A strawberry mug filled with white sesame seeds. The mug is floating in a dark chocolate sea.



# Why diffusion model ?



A fantasy knight

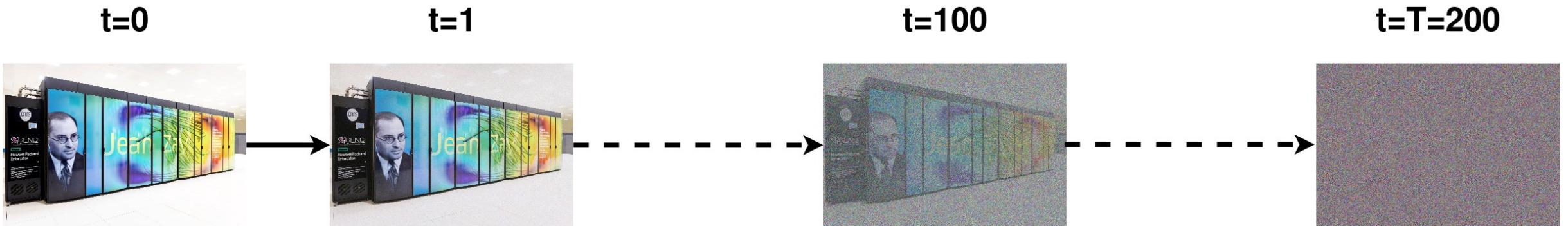


Dora the explorer





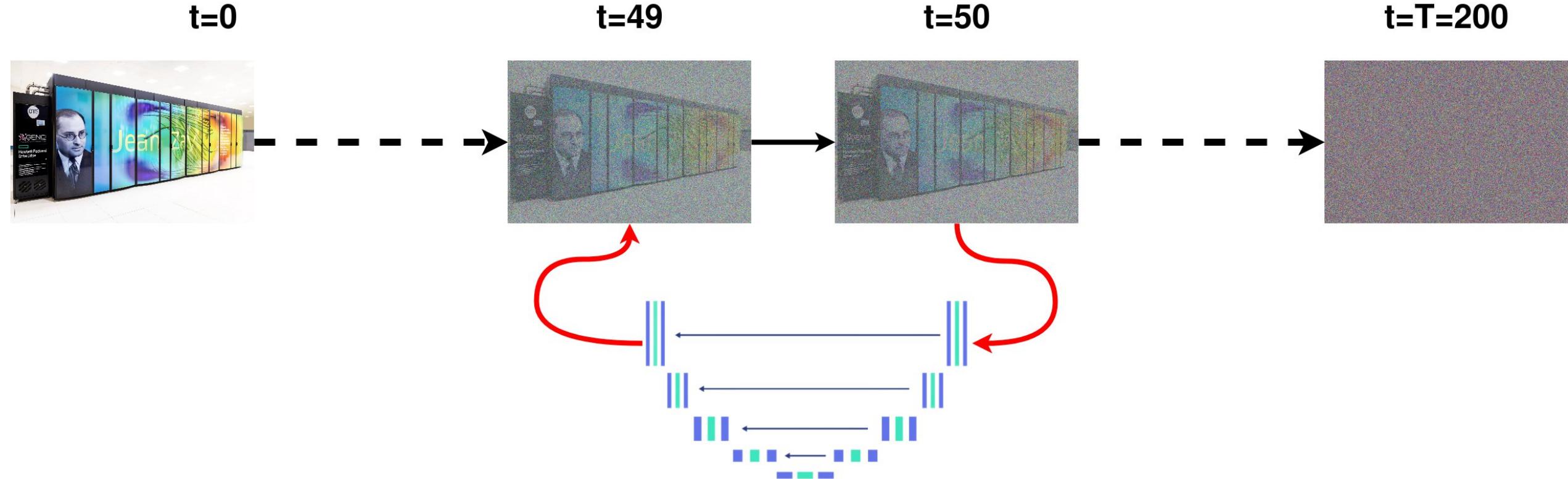
# Diffusion model in short



**Forward diffusion process = fixed Markov chain algorithm**



# Diffusion model in short



**Reverse diffusion process = trained model**



# Diffusion model in short

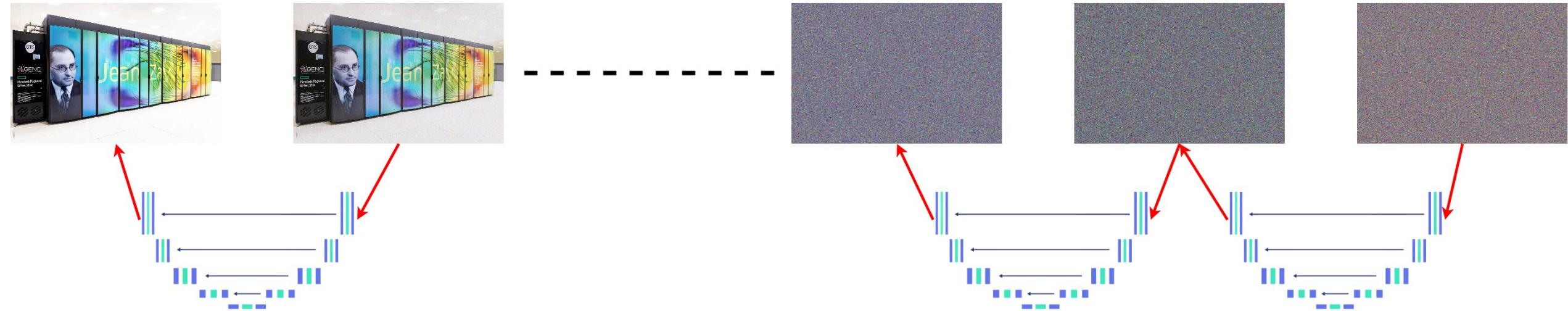
200

199

2

1

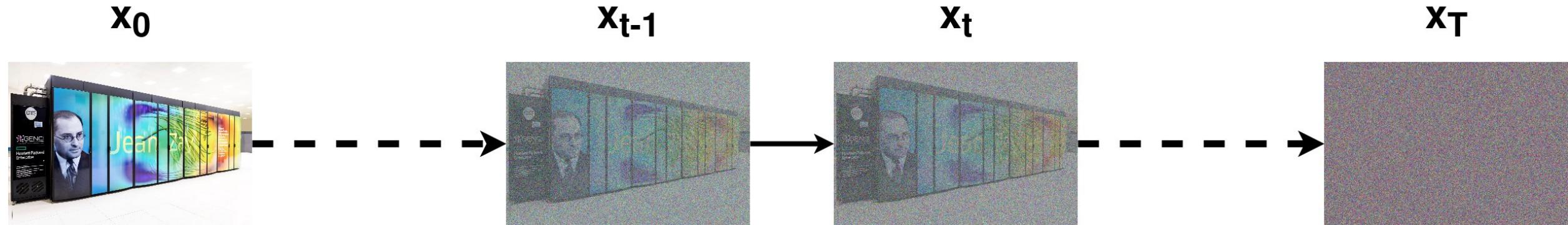
0



**Generation process = reverse diffusion T times from noise**



# Forward diffusion process



$\beta_t \in (0,1)$     $\beta_t$  follow a schedule where  $\beta_1 < \beta_2 < \dots < \beta_T$

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

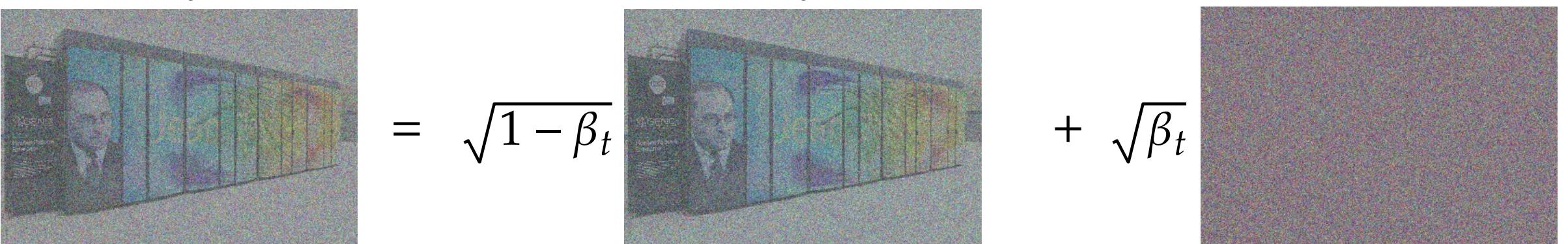
$$x_t = \sqrt{1 - \beta_t} x_{t-1} + \sqrt{\beta_t} z_{t-1} \quad ; \text{ where } z_{t-1} \sim N(0, I)$$



# Forward diffusion process

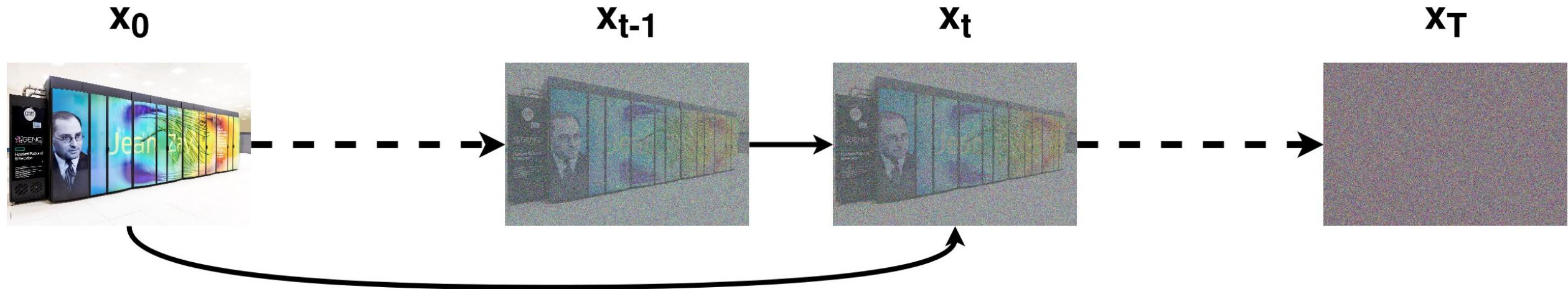
$$\mathbf{x}_t = \sqrt{1 - \beta_t} \mathbf{x}_{t-1} + \sqrt{\beta_t} \text{noise}$$

The equation illustrates the forward diffusion process. It shows a sequence of images:  $\mathbf{x}_t$  (original image),  $\mathbf{x}_{t-1}$  (intermediate state), and "noise" (highly noisy image). The transformation is represented by the equation, where the original image is scaled by  $\sqrt{1 - \beta_t}$  and added to noise scaled by  $\sqrt{\beta_t}$ .





# Forward diffusion process



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

$$x_t = \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1 - \alpha_t\alpha_{t-1}}\bar{z}_{t-2}$$

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

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



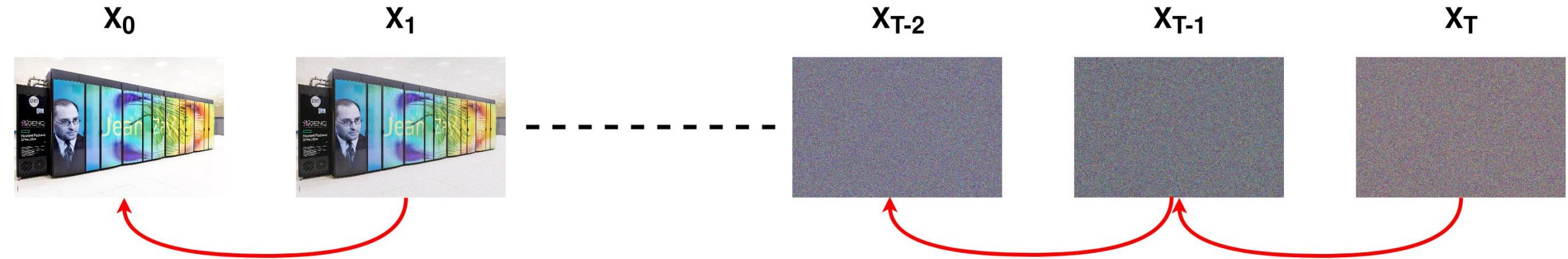
# Forward diffusion process

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \mathbf{z}_t \text{ (noise)}$$

The equation illustrates the forward diffusion process. It shows a server rack labeled  $\mathbf{x}_t$  on the left, which is composed of two parts:  $\sqrt{\bar{\alpha}_t} \mathbf{x}_0$  (the original image) and  $\sqrt{1 - \bar{\alpha}_t} \mathbf{z}_t$  (noise). The middle part,  $\mathbf{x}_0$ , is a clear image of a man's face, while the right part,  $\mathbf{z}_t$  (noise), is a dark, grainy image.



# Reverse diffusion process



$$q(x_{t-1} | x_t) \approx p_\theta(x_{t-1} | x_t) = \begin{array}{c} \text{Diagram showing a flow from } x_t \text{ to } x_{t-1} \text{ through a series of latent variables } z_1, z_2, \dots, z_d. \\ \text{The flow is represented by arrows pointing from } x_t \text{ down to } z_1, z_2, \dots, z_d, \text{ and then from each } z_i \text{ up to } x_{t-1}. \end{array}$$

$$p_\theta(x_{t-1} | x_t) = \mathbf{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 I)$$



# Reverse diffusion process

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

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

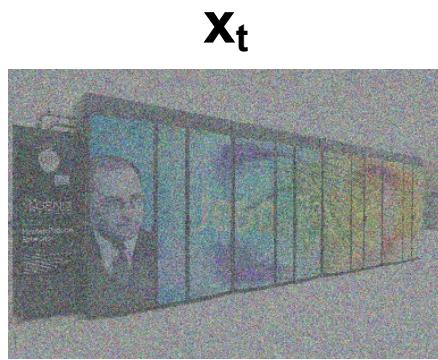
$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1 - \bar{\alpha}_t} x_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t = \frac{1}{\sqrt{\bar{\alpha}_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} z_t \right)$$

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

$$\frac{1}{\sqrt{\bar{\alpha}_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} z_t \right) \approx \frac{1}{\sqrt{\bar{\alpha}_t}} \left( x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} z_\theta(x_t, t) \right)$$



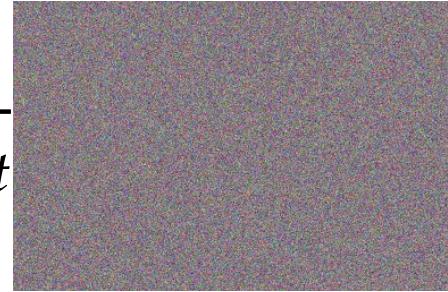
# Reverse diffusion process



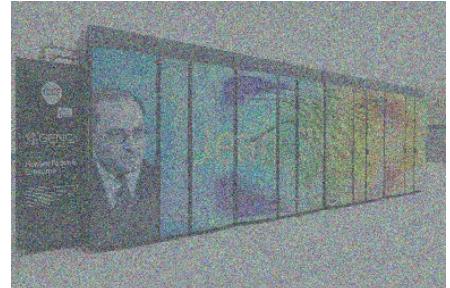
$$= \sqrt{\bar{\alpha}_t}$$



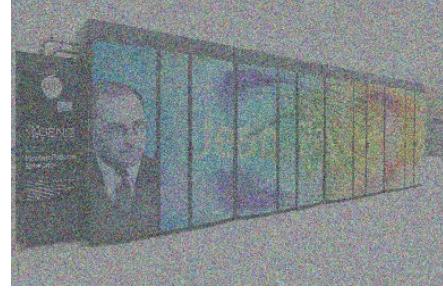
$$+ \sqrt{1 - \bar{\alpha}_t}$$



$\mathbf{x}_{t-1}$



$$= \frac{1}{\sqrt{\bar{\alpha}_t}} ($$



$$- \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}$$

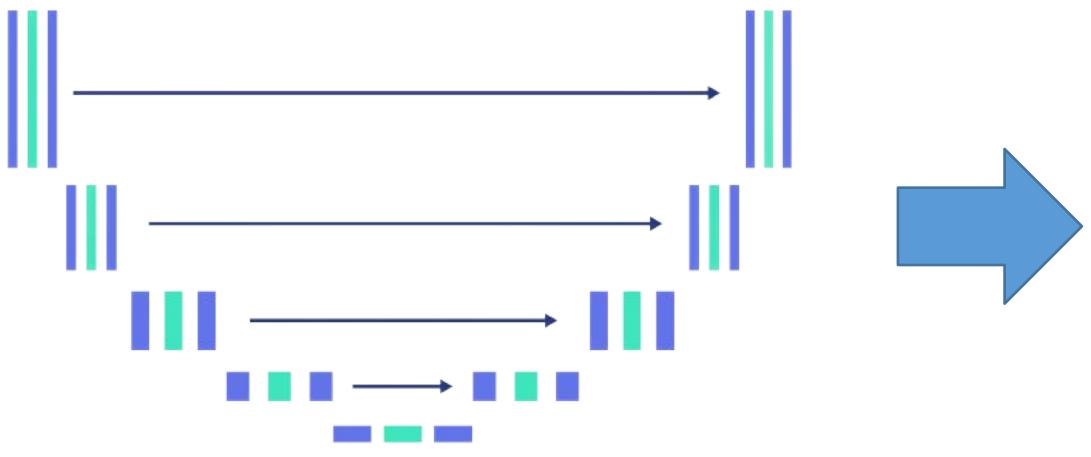


$$+ \tilde{\beta}_t )$$





# Reverse diffusion process



## Loss to train the model

negative log-likelihood:  $-\log p_\theta(x_0)$  (not possible)

=>

variational lower bound:

$$-\log p_\theta(x_0) \leq -\log p_\theta(x_0) + D_{KL}(q(x_{1:T} | x_0) \| p_\theta(x_{1:T} | x_0))$$

==>

$$\mathbf{L}_t = E \left[ \| \boldsymbol{\mu}_t - \mu_\theta(x_t, t) \|^2 \right]$$

$$\mathbf{L}_t = E \left[ \| \boldsymbol{z}_t - z_\theta(x_t, t) \|^2 \right]$$



# Training

---

## 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

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

6: **until** converged

---



# Training

---

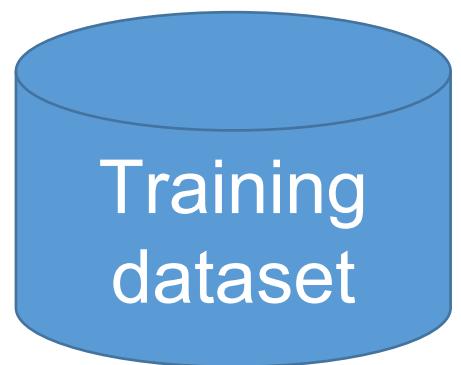
## Algorithm 1 Training

---

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

6: until converged
```

---





# Training

---

## Algorithm 1 Training

---

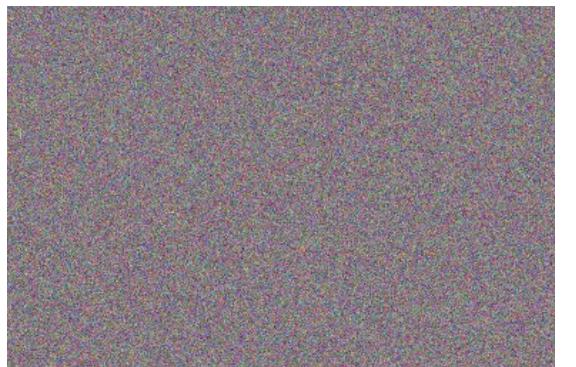
```
1: repeat
2:    $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 
3:    $t \sim \text{Uniform}(\{1, \dots, T\})$ 
4:    $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 
5:   Take gradient descent step on
      
$$\nabla_{\theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\theta}(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, t)\|^2$$

6: until converged
```

---

$$t = 50$$

$$\boldsymbol{\epsilon} =$$





# Training

---

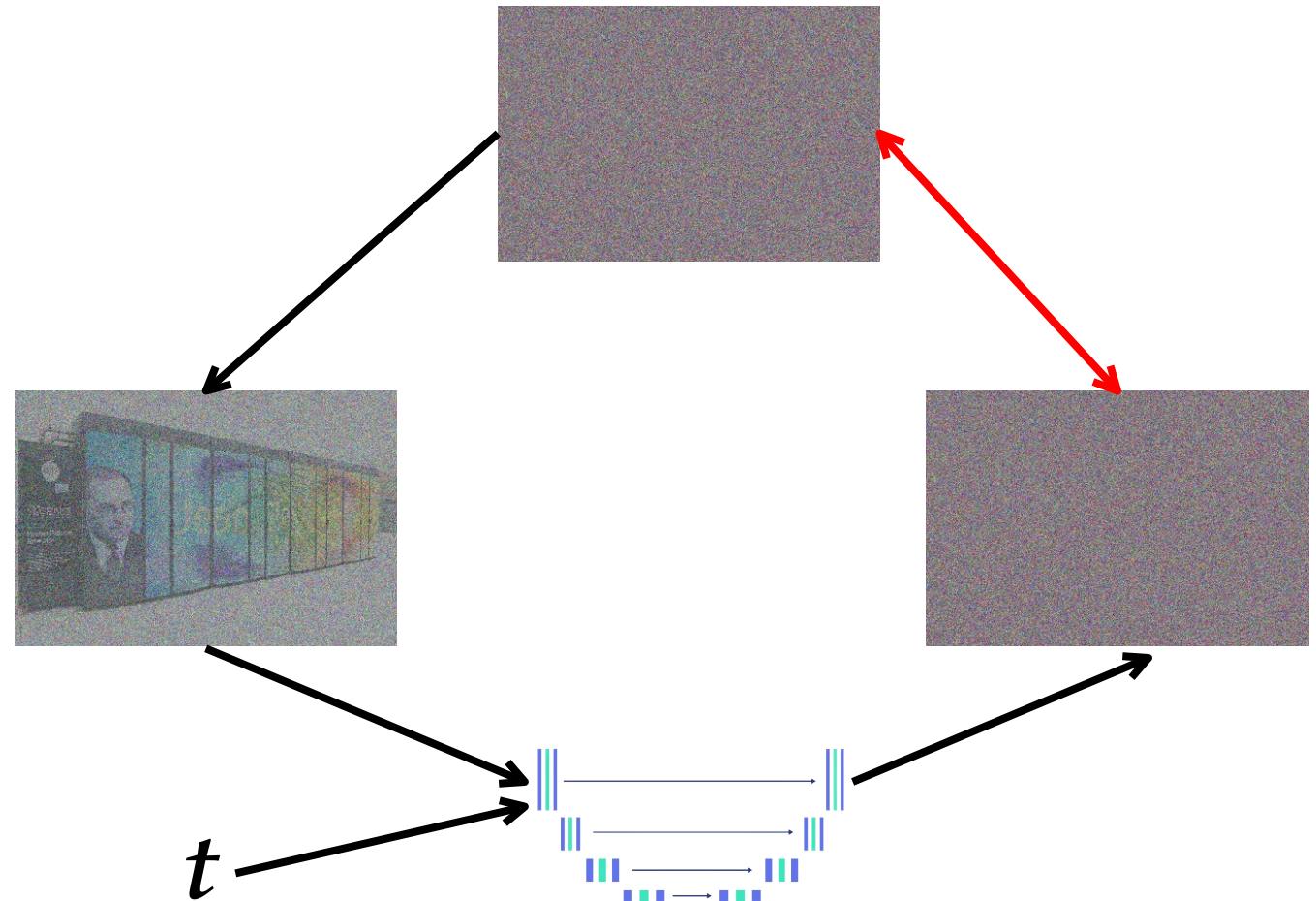
## Algorithm 1 Training

---

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

6: until converged
```

---





# Sampling

---

## Algorithm 2 Sampling

---

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

---



# Sampling

---

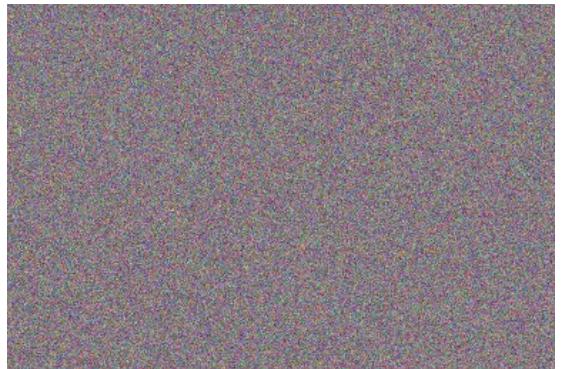
## Algorithm 2 Sampling

---

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

---

$$\mathbf{x}_T =$$





# Sampling

---

## Algorithm 2 Sampling

---

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

---

$\mathbf{z} =$



or

$\mathbf{z} =$





# Sampling

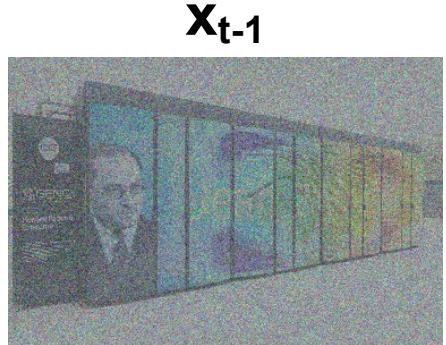
---

## Algorithm 2 Sampling

---

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

---



$\mathbf{x}_{t-1}$

=

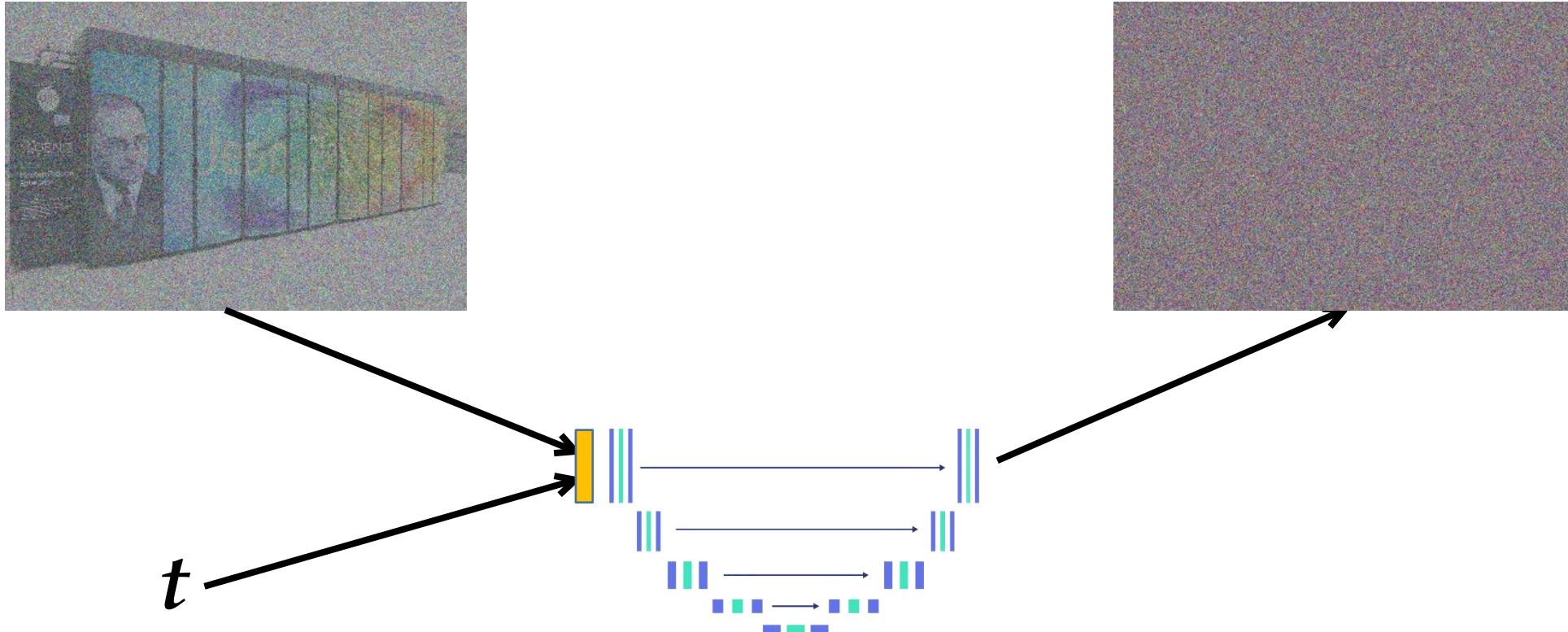
$\mathbf{z}_t$

$\mathbf{z}$  (noise)

$$\frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \mathbf{z}_t + \tilde{\beta}_t \mathbf{z} \right)$$

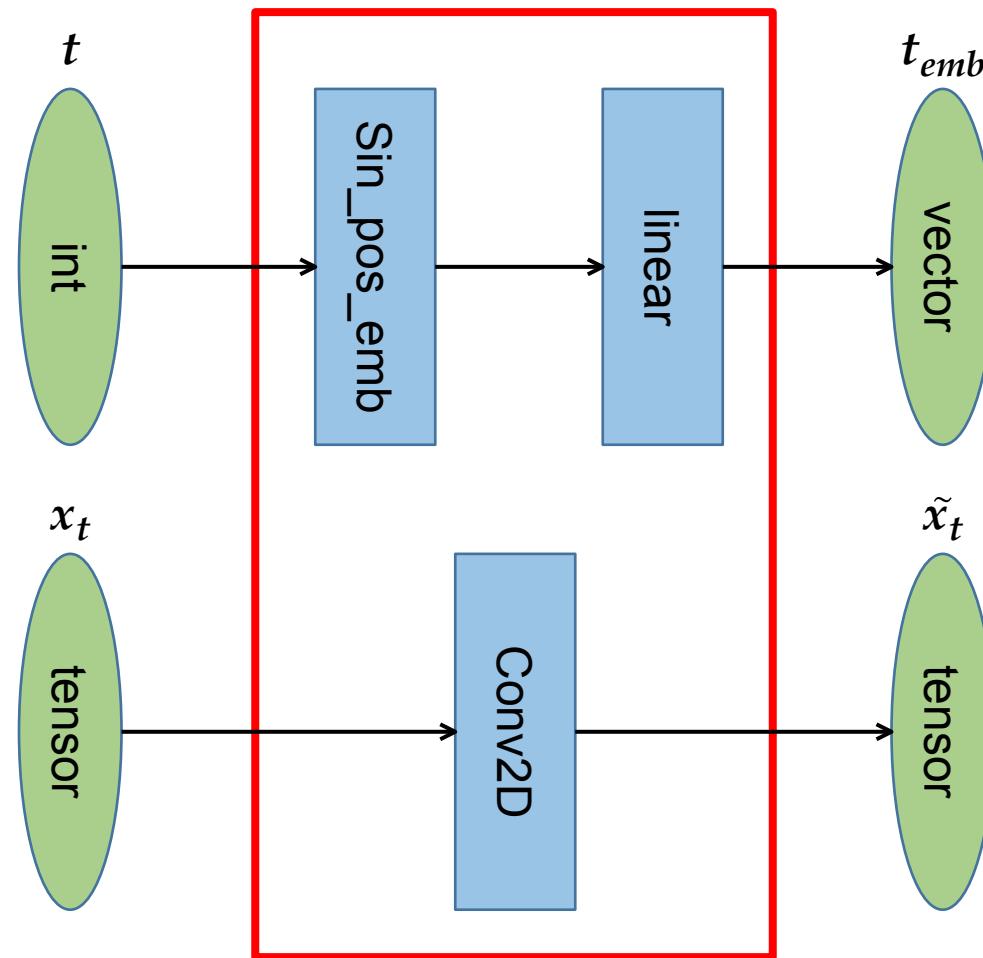


# Model example



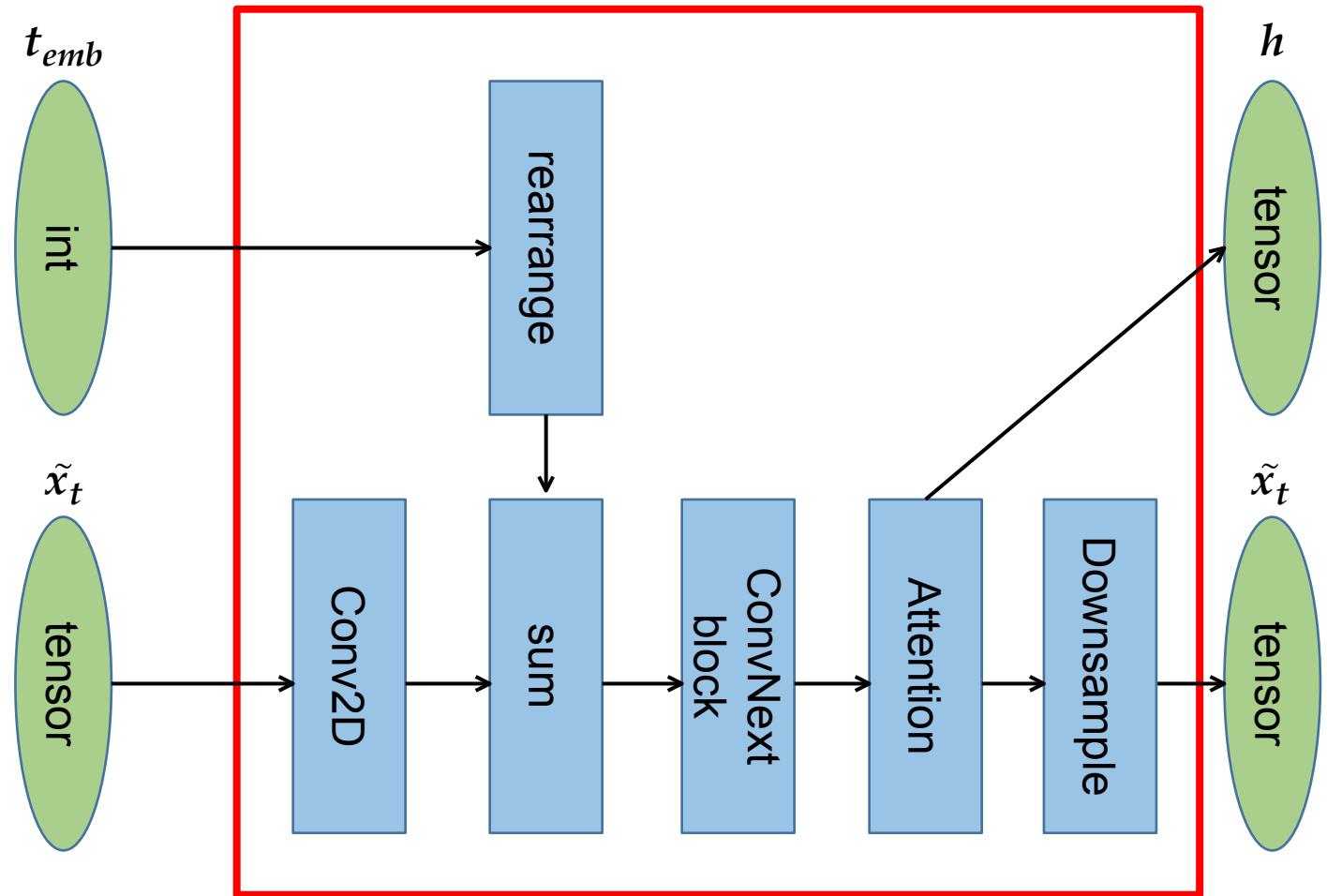


# Model example



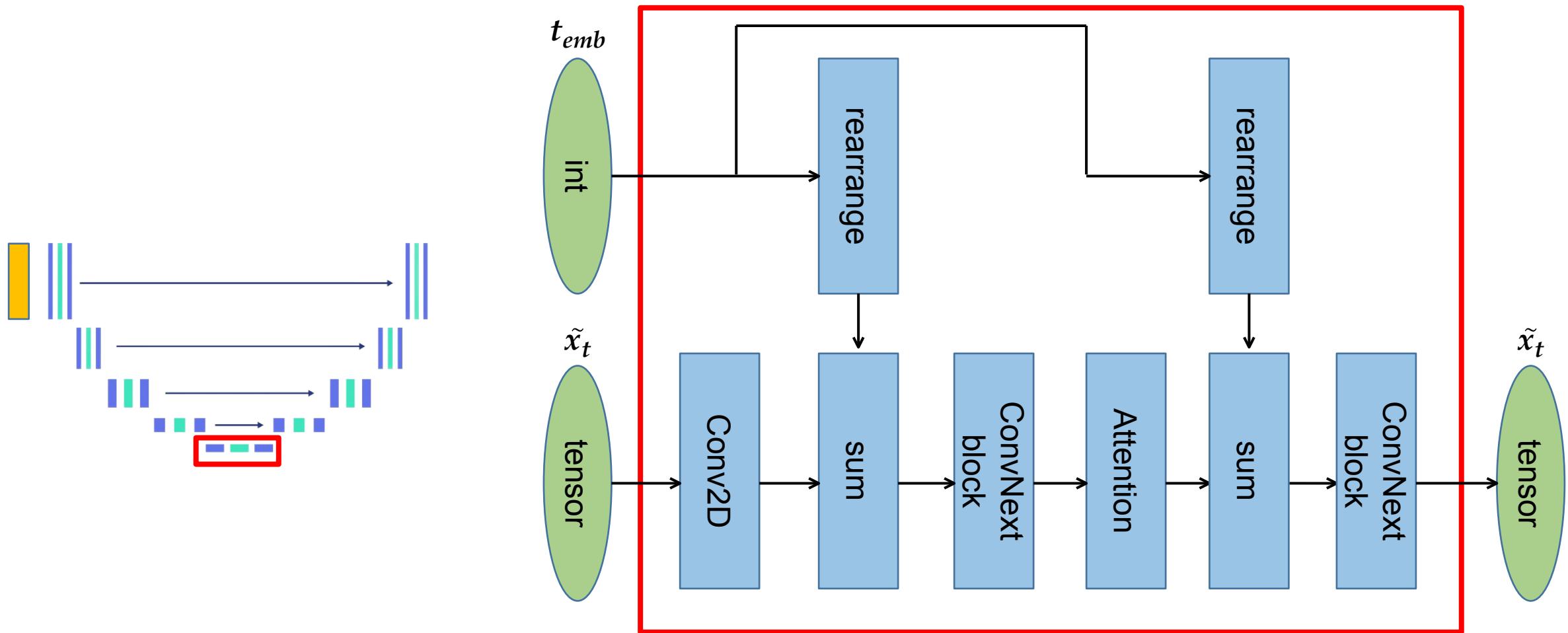


# Model example



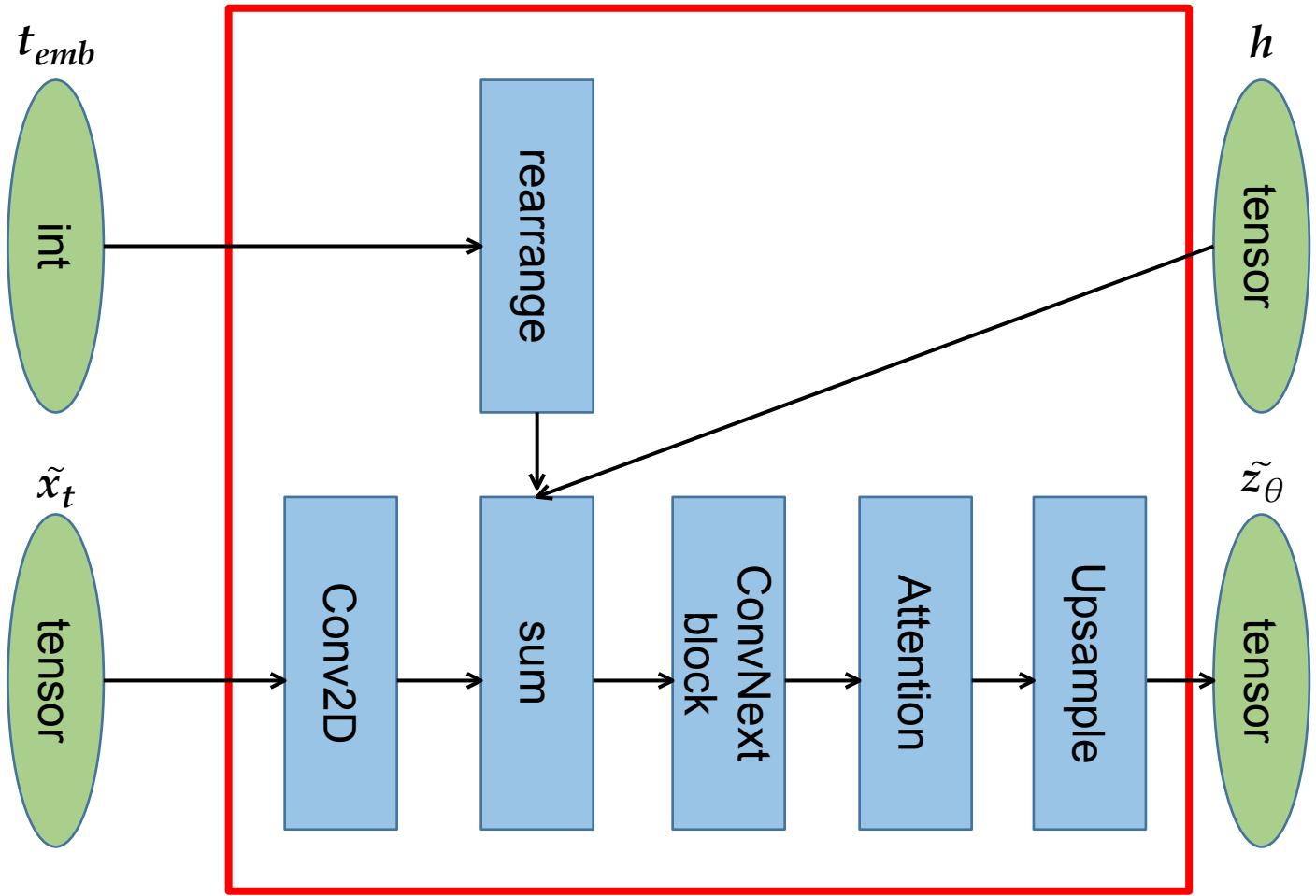
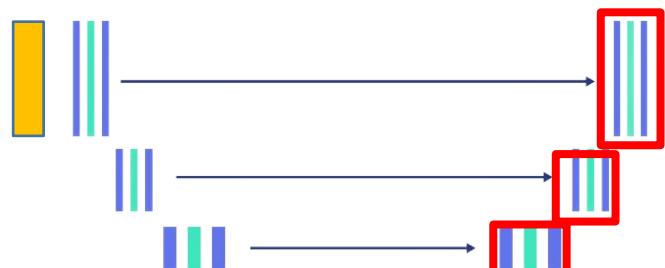


# Model example



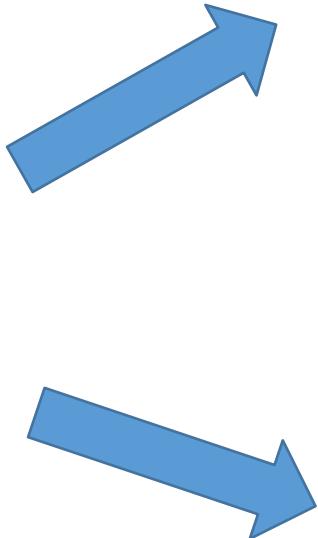


# Model example





# Text-to-image



**The French supercomputer Jean-Zay**



# Text-to-image

The French supercomputer Jean-Zay



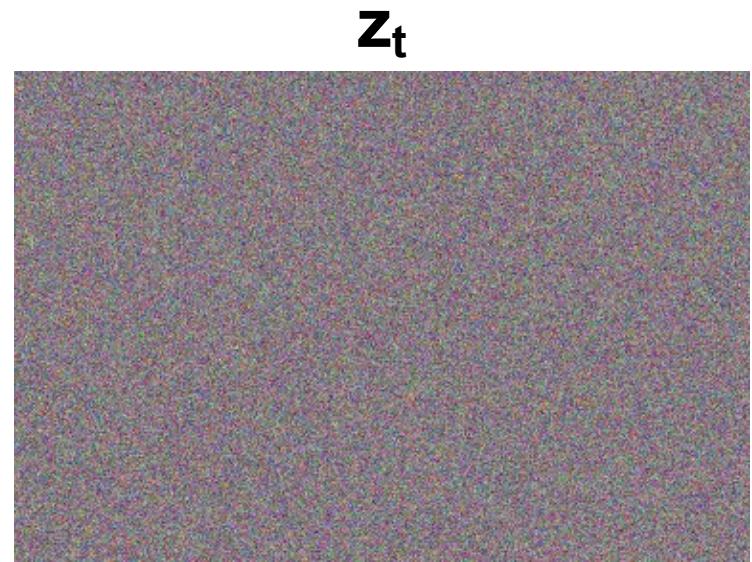
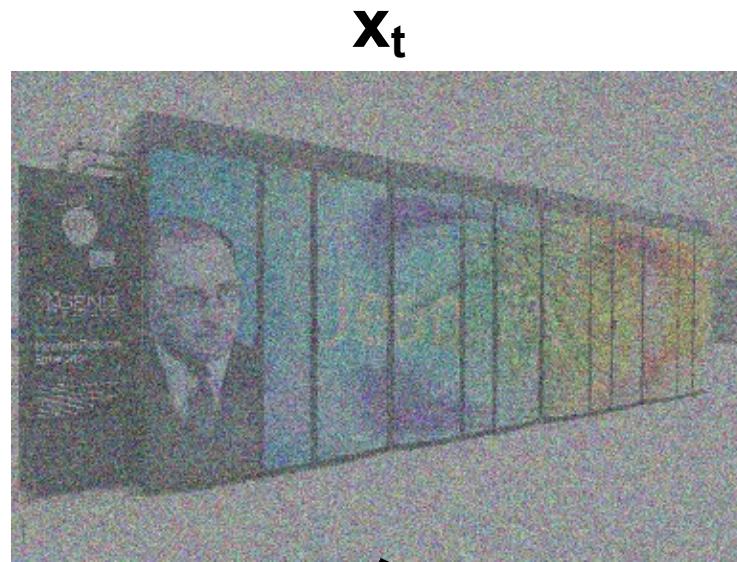
Text embedding model



(0.961    0.230    - 0.567 ...)

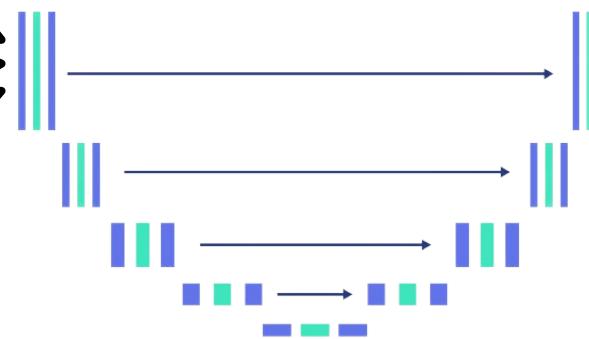


# Text-to-image



$t$

(0.961    0.230    - 0.567 ...)





# Biblio

- **Deep Unsupervised Learning using Nonequilibrium Thermodynamics :**  
<https://arxiv.org/abs/1503.03585>
- **Generative Modeling by Estimating Gradients of the Data Distribution :**  
<https://arxiv.org/abs/1907.05600>
- **Denoising Diffusion Probabilistic Models :** <https://arxiv.org/abs/2006.11239>
- **Improved Denoising Diffusion Probabilistic Models :** <https://arxiv.org/abs/2102.09672>
- **GLIDE: Towards Photorealistic Image Generation and Editing with Text-Guided Diffusion Models :** <https://arxiv.org/abs/2112.10741>
- **Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding :**  
<https://arxiv.org/abs/2205.11487>
- **Lilian Weng article “What are Diffusion Models?” :** <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>
- **Outlier video “Diffusion Models | Paper Explanation | Math Explained” :**  
<https://www.youtube.com/watch?v=HoKDTa5jHvg>