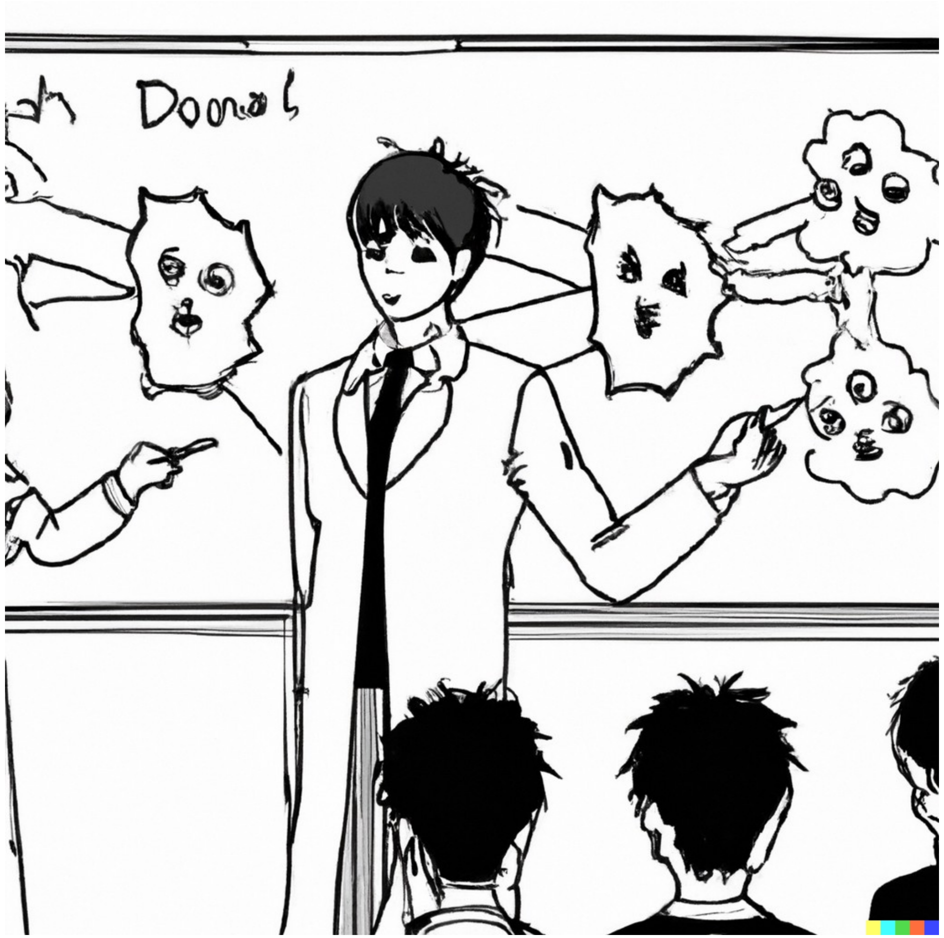
INSTITUT DU
DÉVELOPPEMENT ET DES
RESSOURCES EN
INFORMATIQUE
SCIENTIFIQUE

Diffusion Model

AI DevTalks #5

13 septembre 2022

Diffusion Probabilistic Model



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

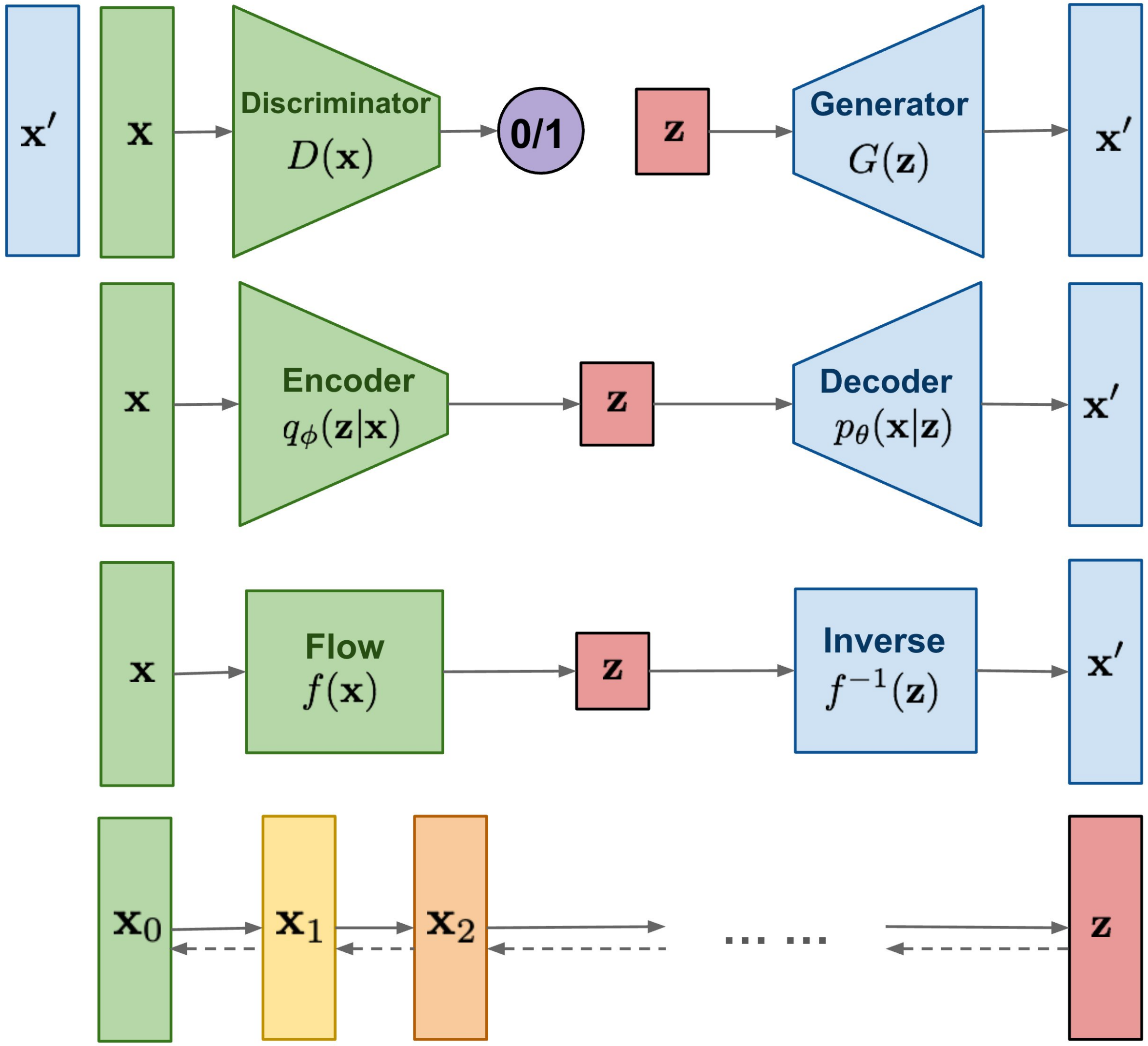
Generative models

GAN: Adversarial training

VAE: maximize variational lower bound

Flow-based models: Invertible transform of distributions

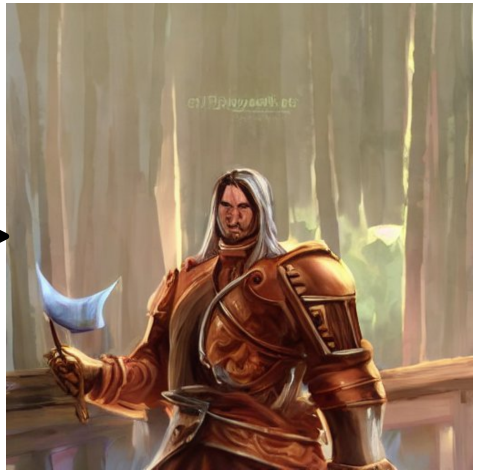
Diffusion models: Gradually add Gaussian noise and then reverse



Why diffusion model ?

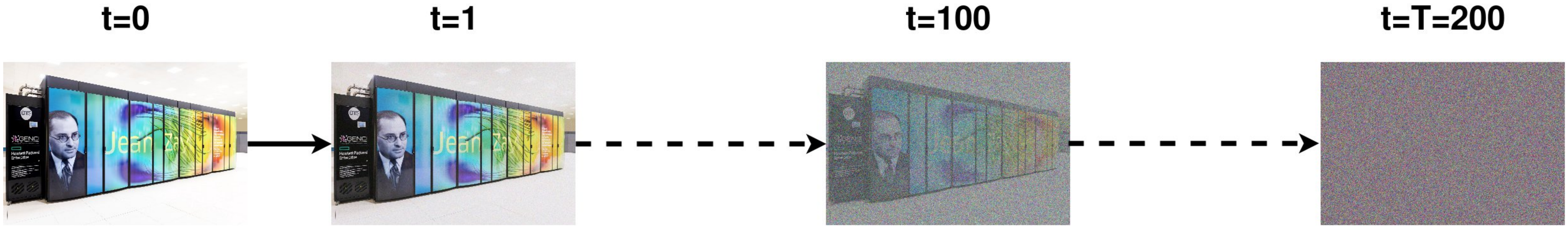
Why diffusion model ?

A fantasy knight



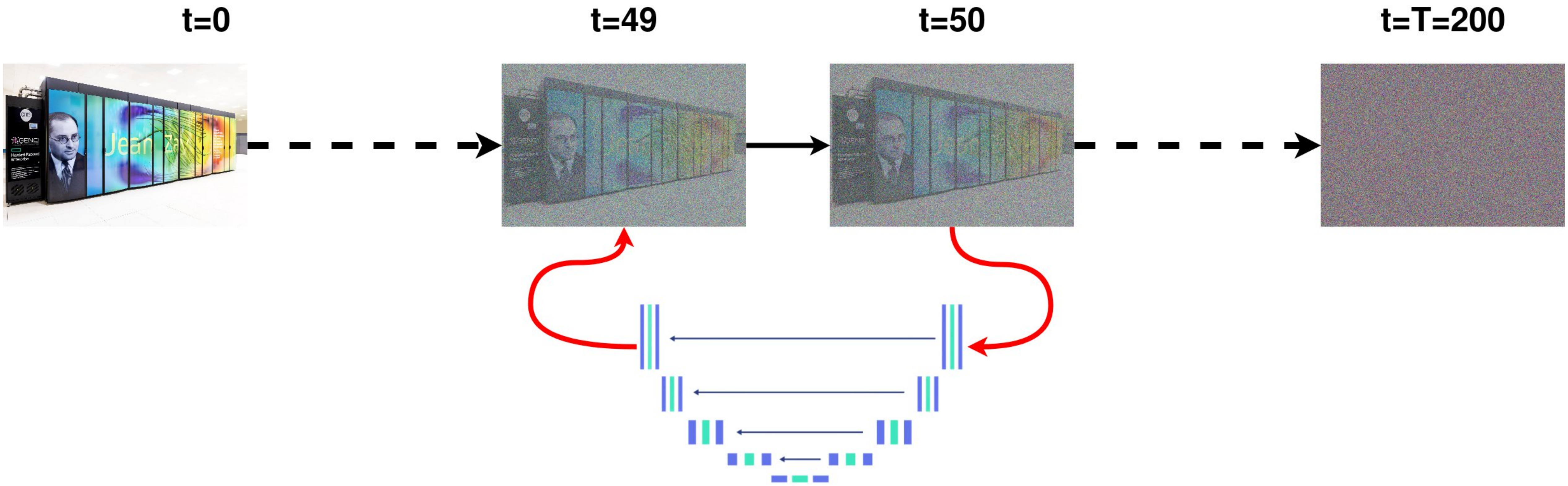
Dora the explorer

Diffusion model in short



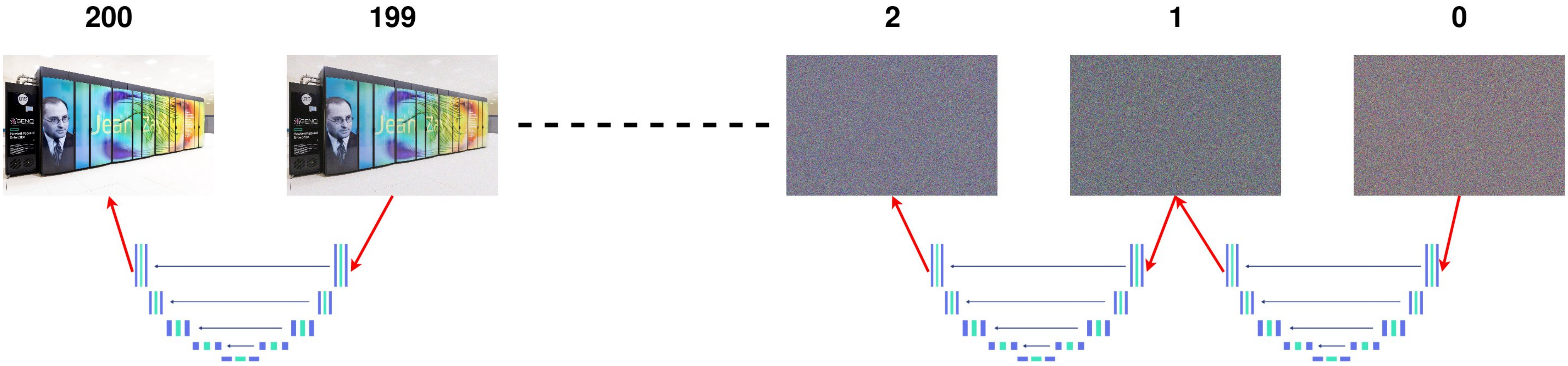
Forward diffusion process = fixed Markov chain algorithm

Diffusion model in short



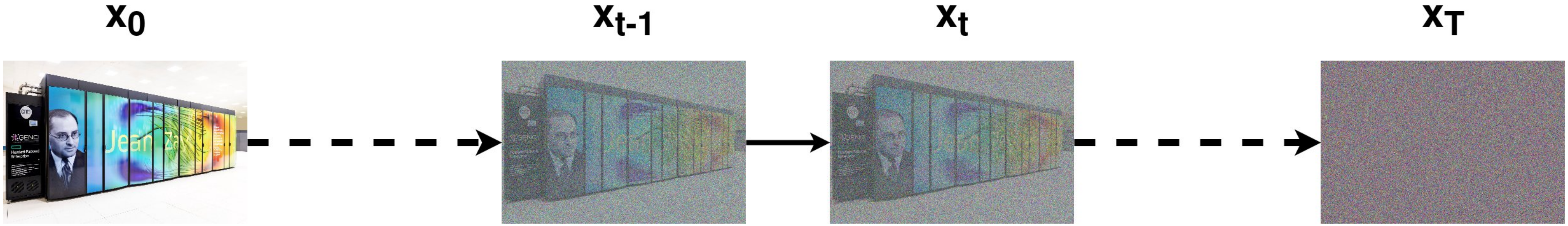
Reverse diffusion process = trained model

Diffusion model in short



Generation process = reverse diffusion T times from noise

Forward diffusion process



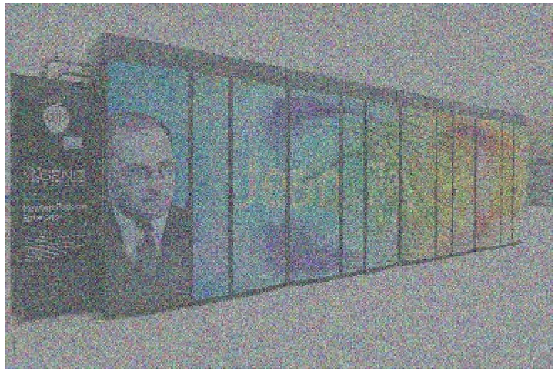
$$\boldsymbol{\beta_t} \in (0,1) \qquad \boldsymbol{\beta_t} \text{ follow a schedule where } \beta_1 < \beta_2 < \cdots < \beta_T$$

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

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

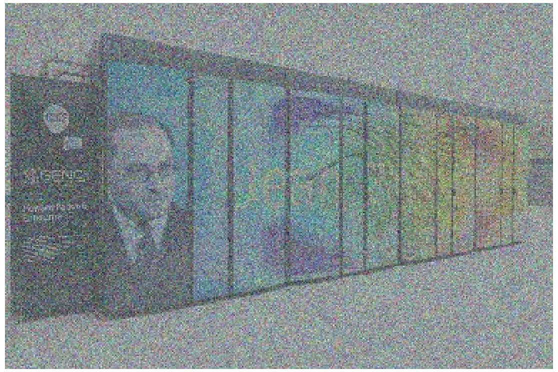
Forward diffusion process

x_t



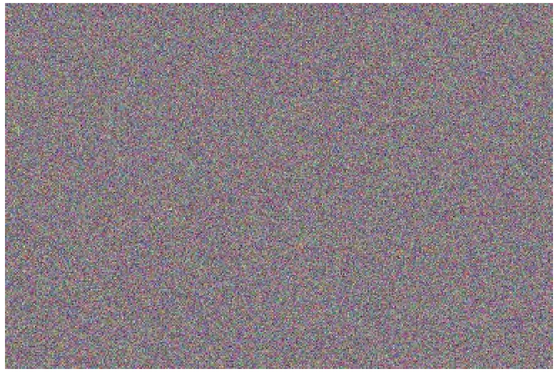
$$= \sqrt{1-\beta_t}$$

x_{t-1}

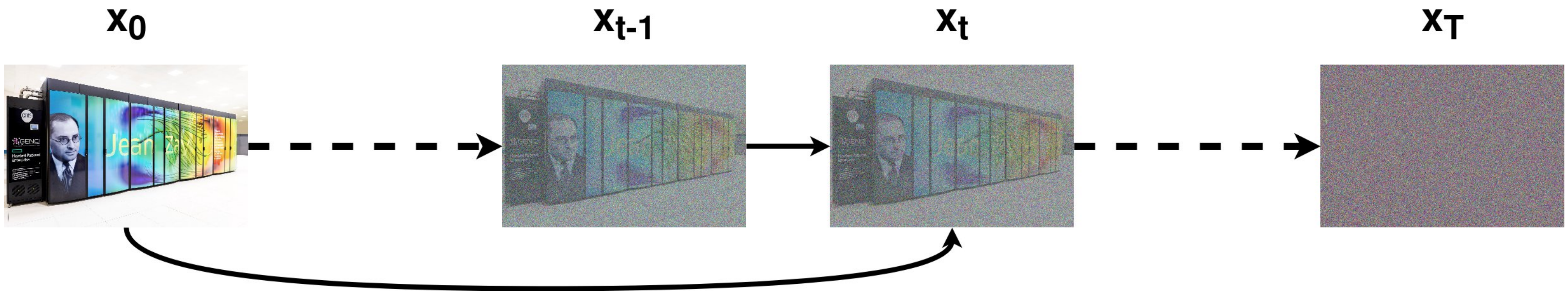


$$+ \sqrt{\beta_t}$$

noise



Forward diffusion process



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

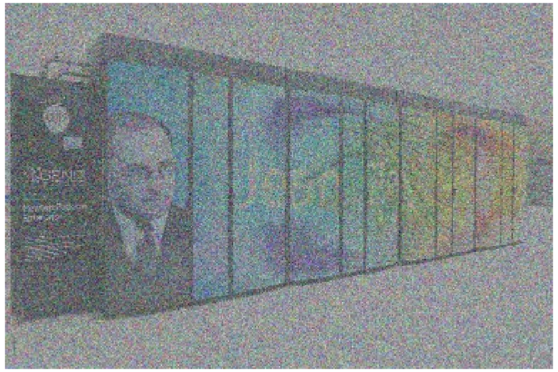
$$\boldsymbol{x_t} = \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{z}_{t-2}$$

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

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

Forward diffusion process

x_t

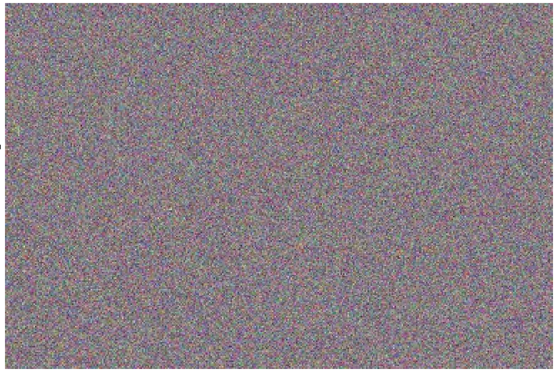


$= \sqrt{\bar{\alpha}_t}$

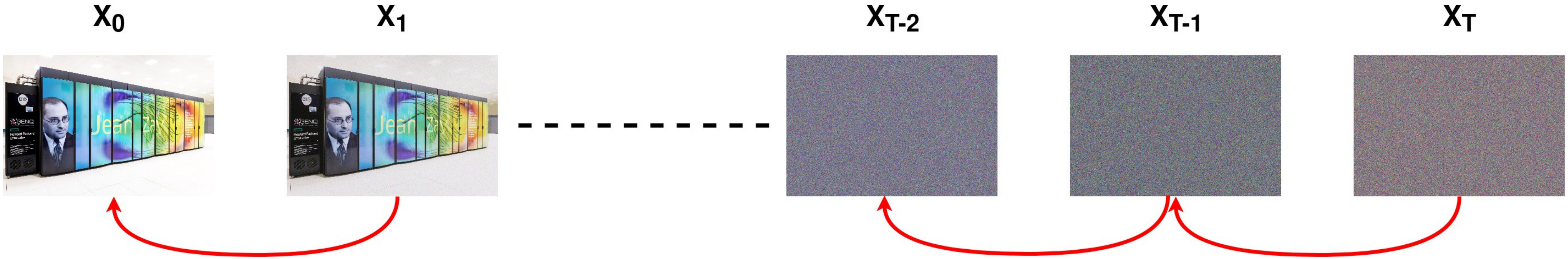
x_0

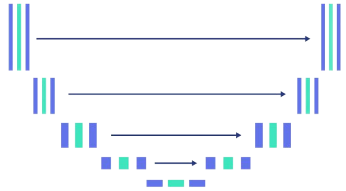
$+ \sqrt{1 - \bar{\alpha}_t}$

z_t (noise)



Reverse diffusion process



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

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

Reverse diffusion process

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

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

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

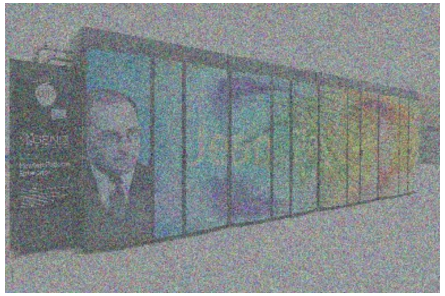
$$q(\, x_{t-1} \mid x_t \,, x_0\,) \approx p_\theta(\, x_{t-1} \mid x_t\,)$$

=>

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

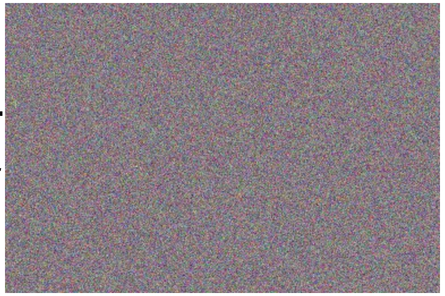
Reverse diffusion process

x_t **x_0** **z_t (noise)**



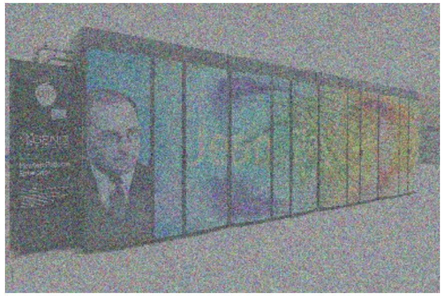
$$= \sqrt{\bar{\alpha}_t}$$

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

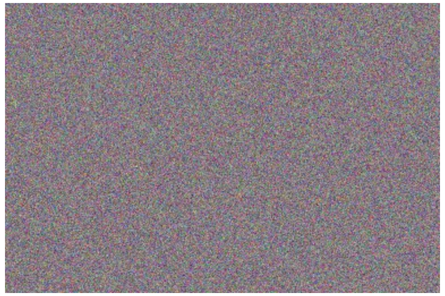


x_{t-1} **x_t** **z_t** **z (noise)**

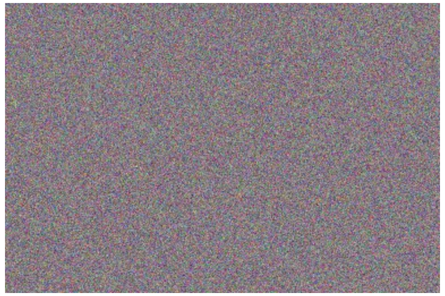
$$= \frac{1}{\sqrt{\alpha_t}} ($$



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

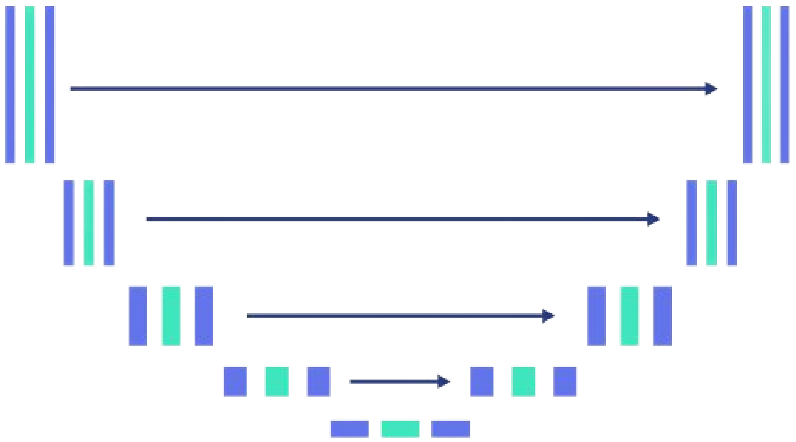


$$+ \tilde{\beta}_t$$



$$)$$

Reverse diffusion process



Loss to train the model

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

=>

variational lower bound:

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

==>

$$L_t = E\left[\| \mu_t - \mu_\theta(x_t, t) \|^2\right]$$

$$L_t = E\left[\| z_t - z_\theta(x_t, t) \|^2\right]$$

Training

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on

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

until converged

Training

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
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

Training

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
  $$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
until converged

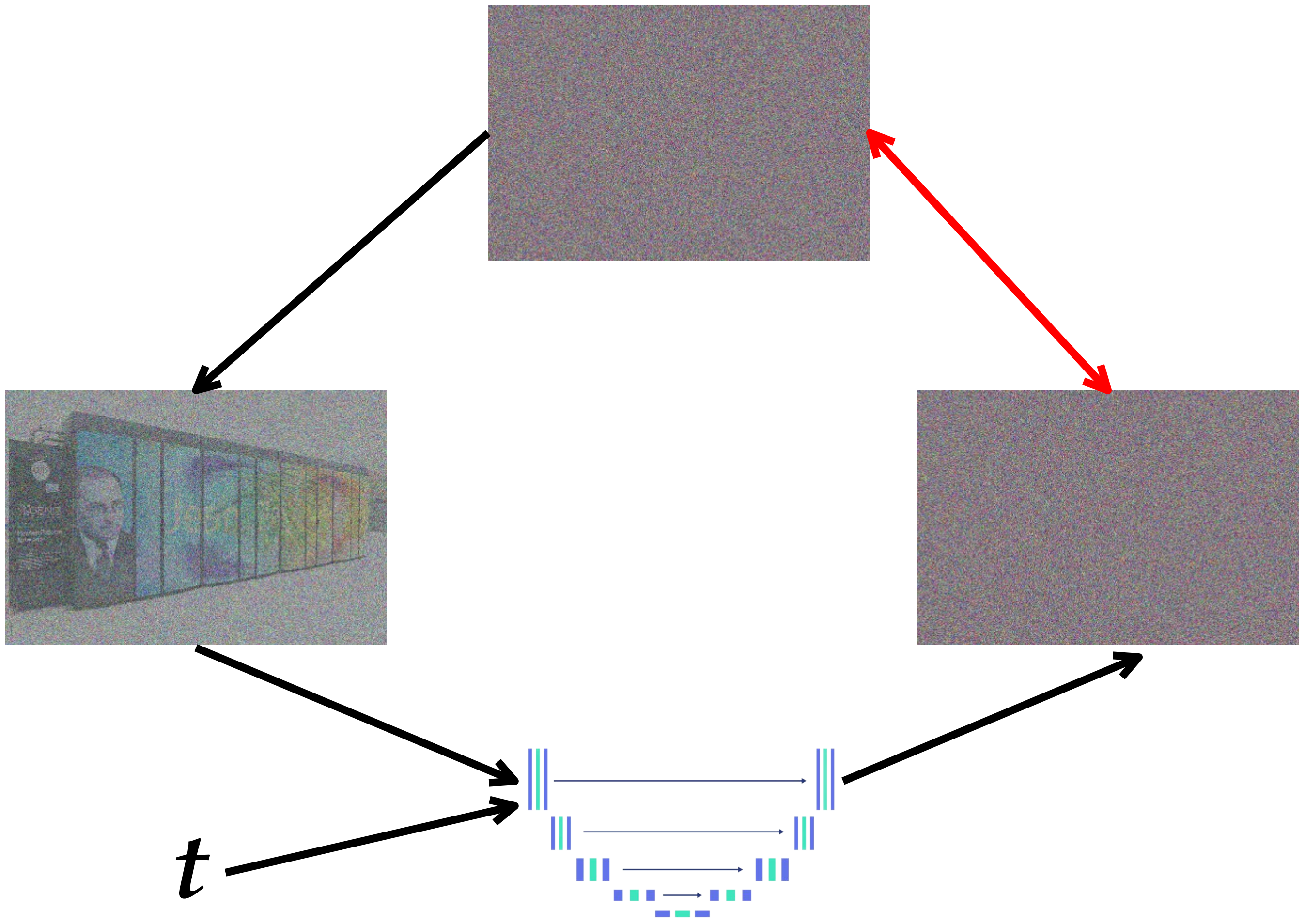
$$t = 50$$

$$\epsilon =$$

Training

Algorithm 1 Training

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
  $t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Take gradient descent step on
    $\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
until converged



Sampling

Algorithm 2 Sampling

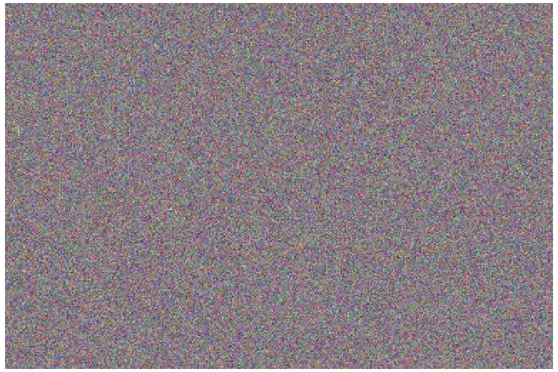
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
  $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

Sampling

Algorithm 2 Sampling

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

$x_T =$



Sampling

Algorithm 2 Sampling

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

z =

or

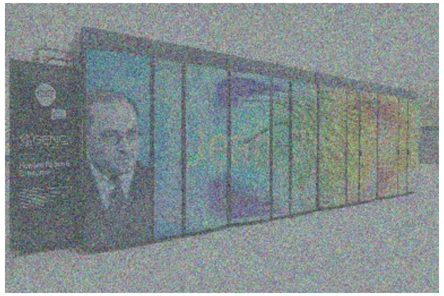
z =

Sampling

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **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** $\mathbf{x}_0$

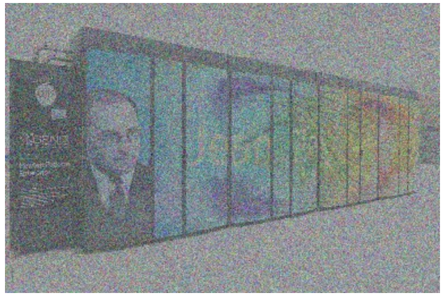
x_{t-1}



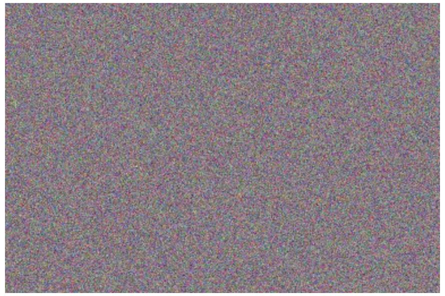
$=$

x_t **z_t** **z (noise)**

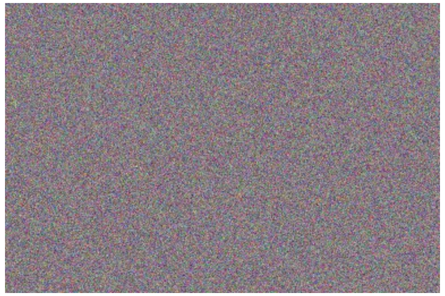
$\frac{1}{\sqrt{\alpha_t}}($



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

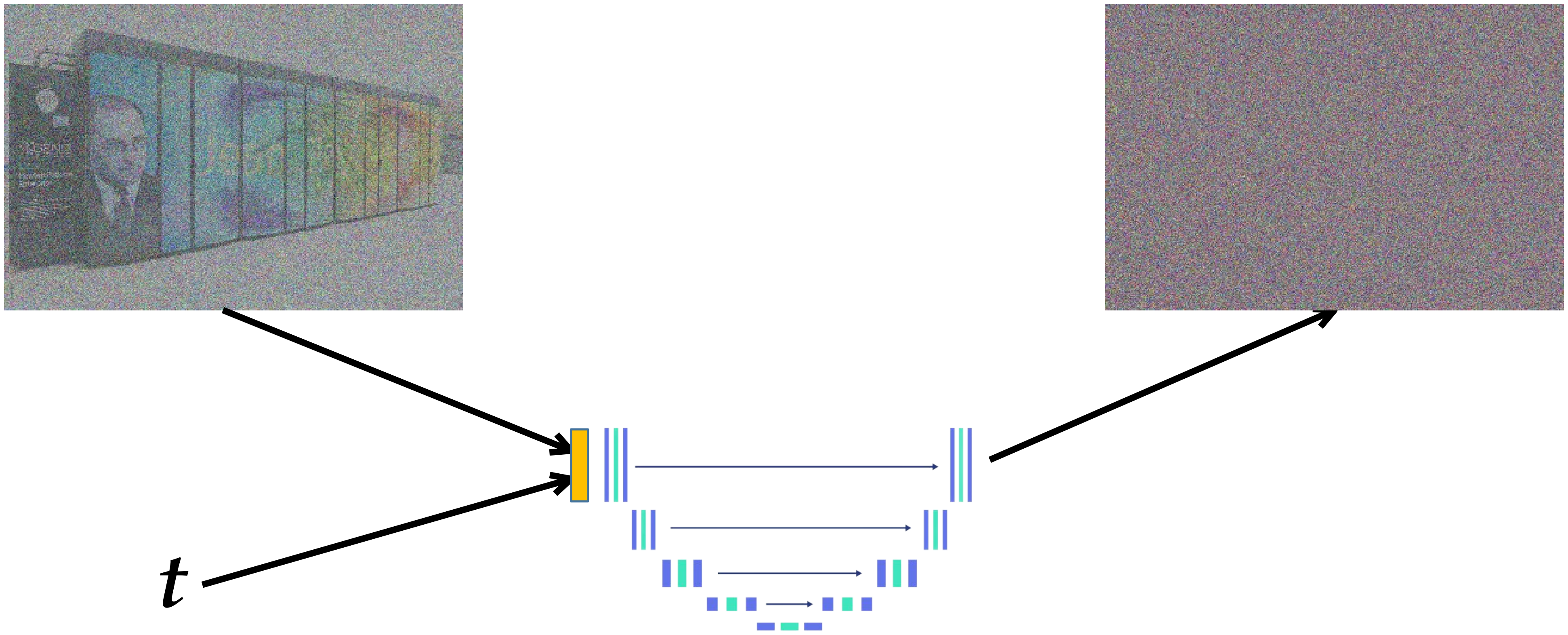


$+ \tilde{\beta}_t$

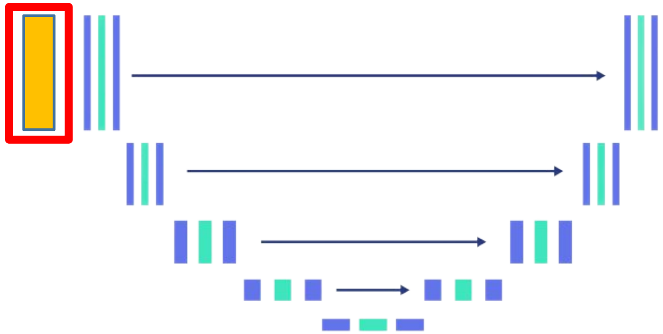


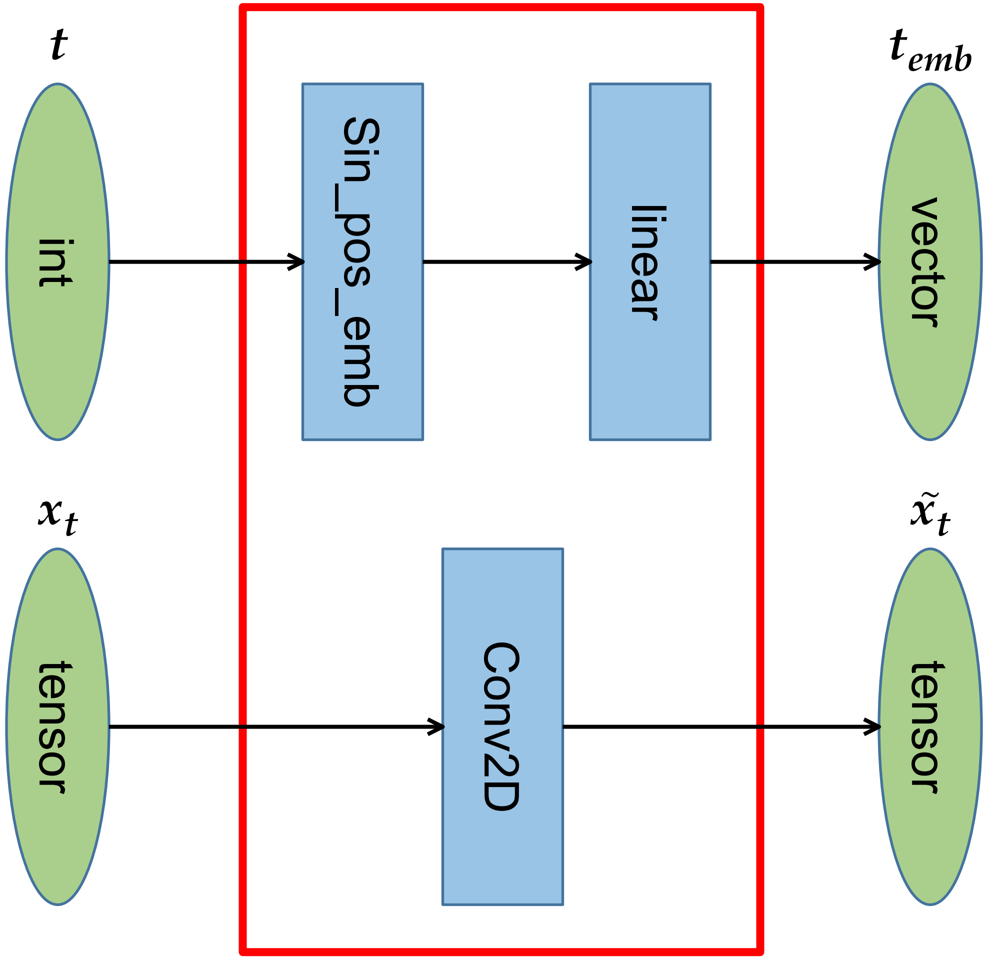
$)$

Model example

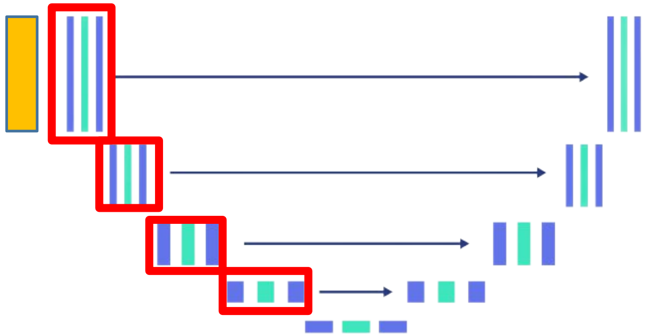


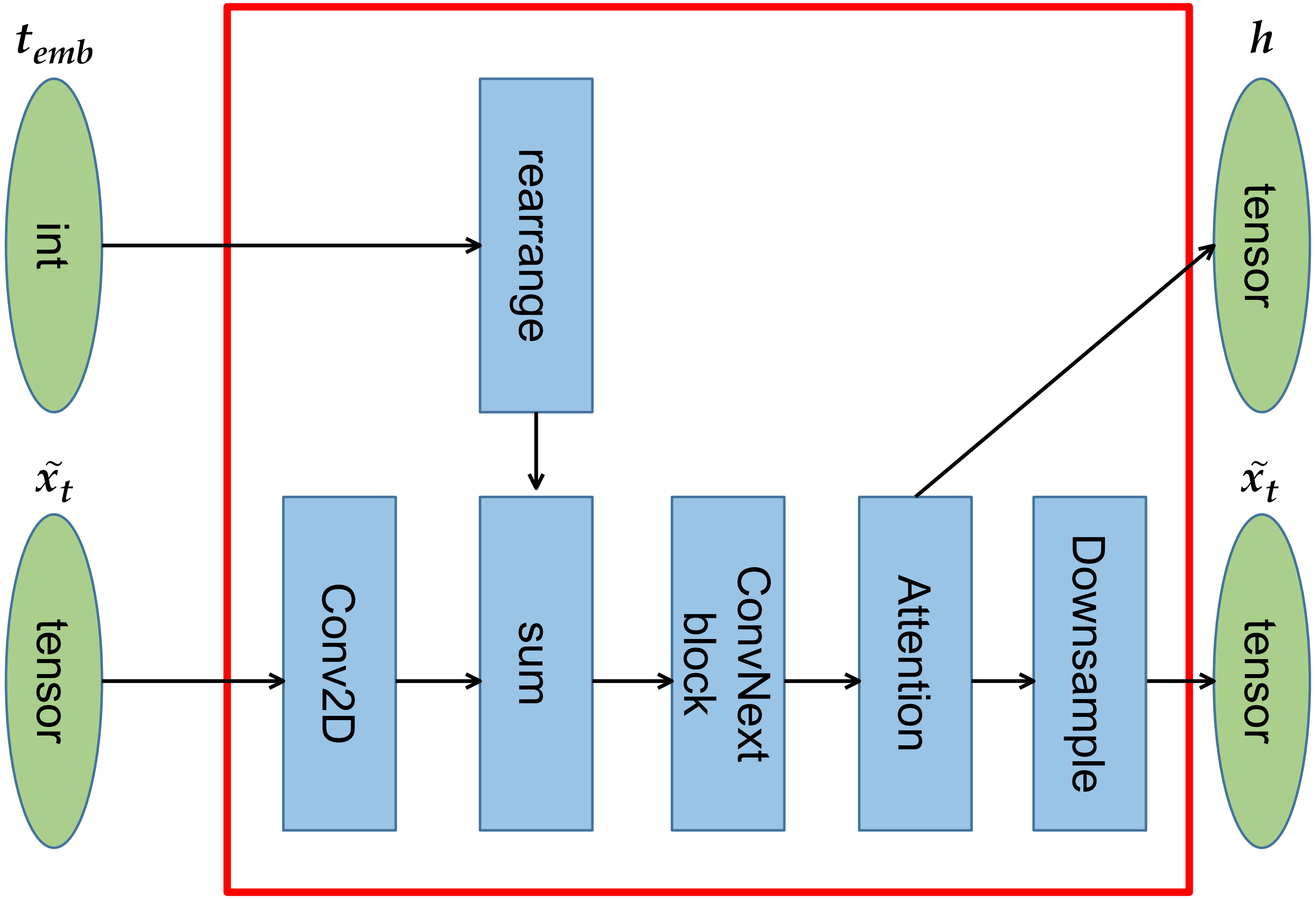
Model example



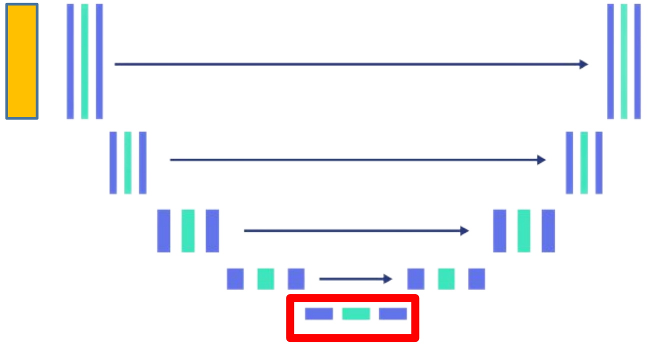


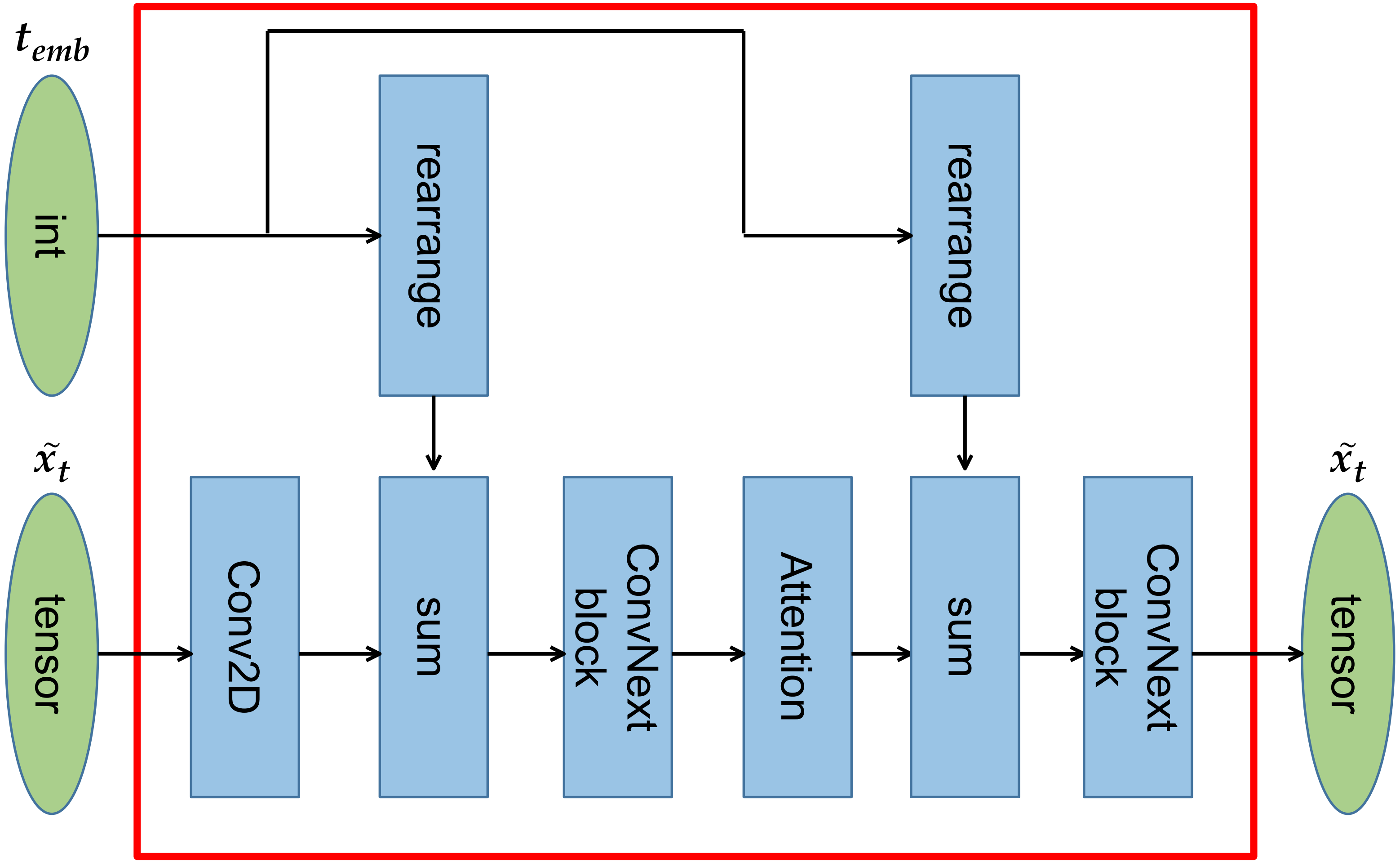
Model example



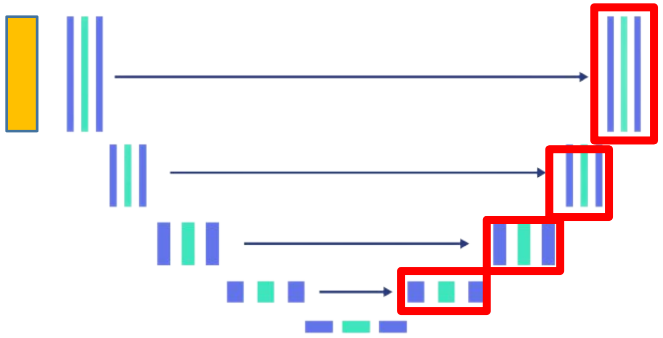


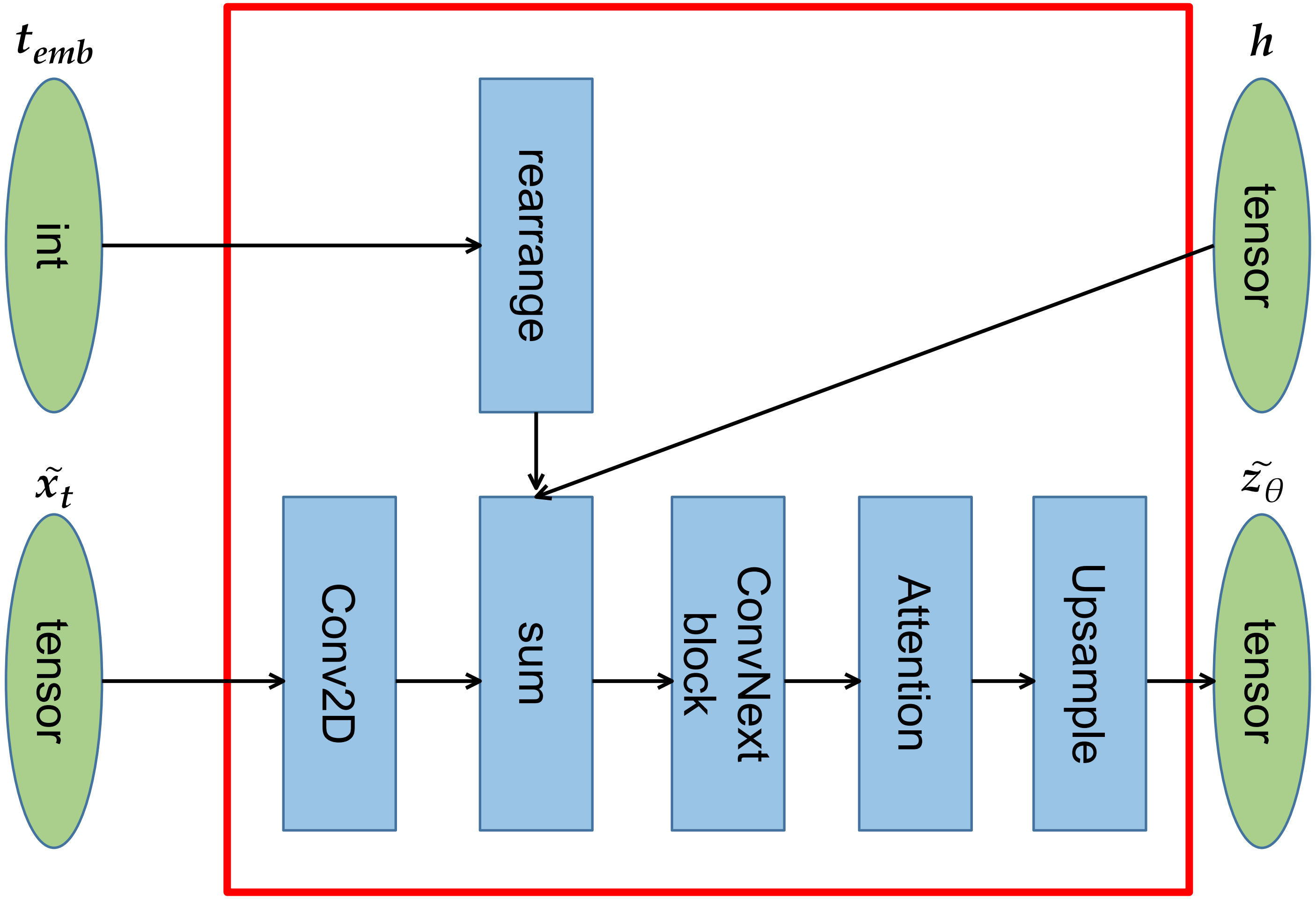
Model example



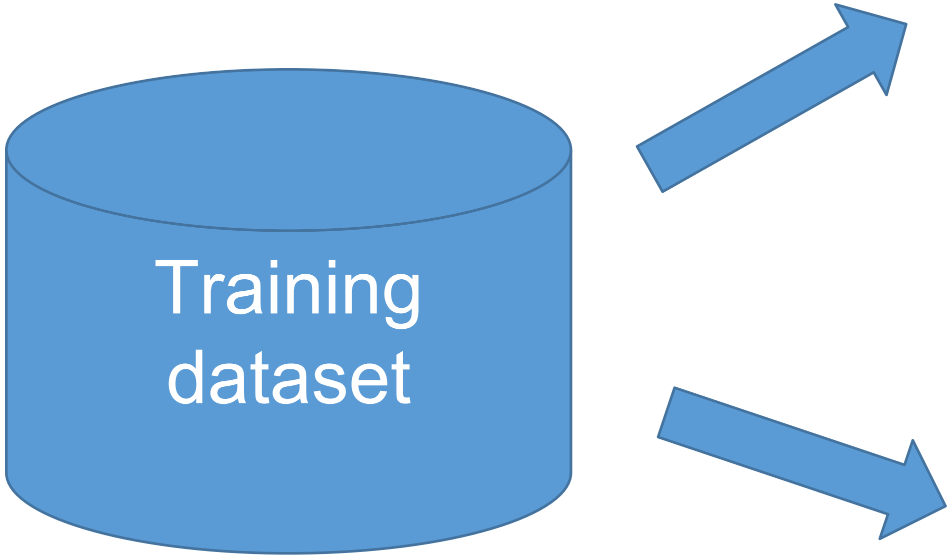


Model example





Text-to-image



The French supercomputer Jean-Zay

Text-to-image

The French supercomputer Jean-Zay

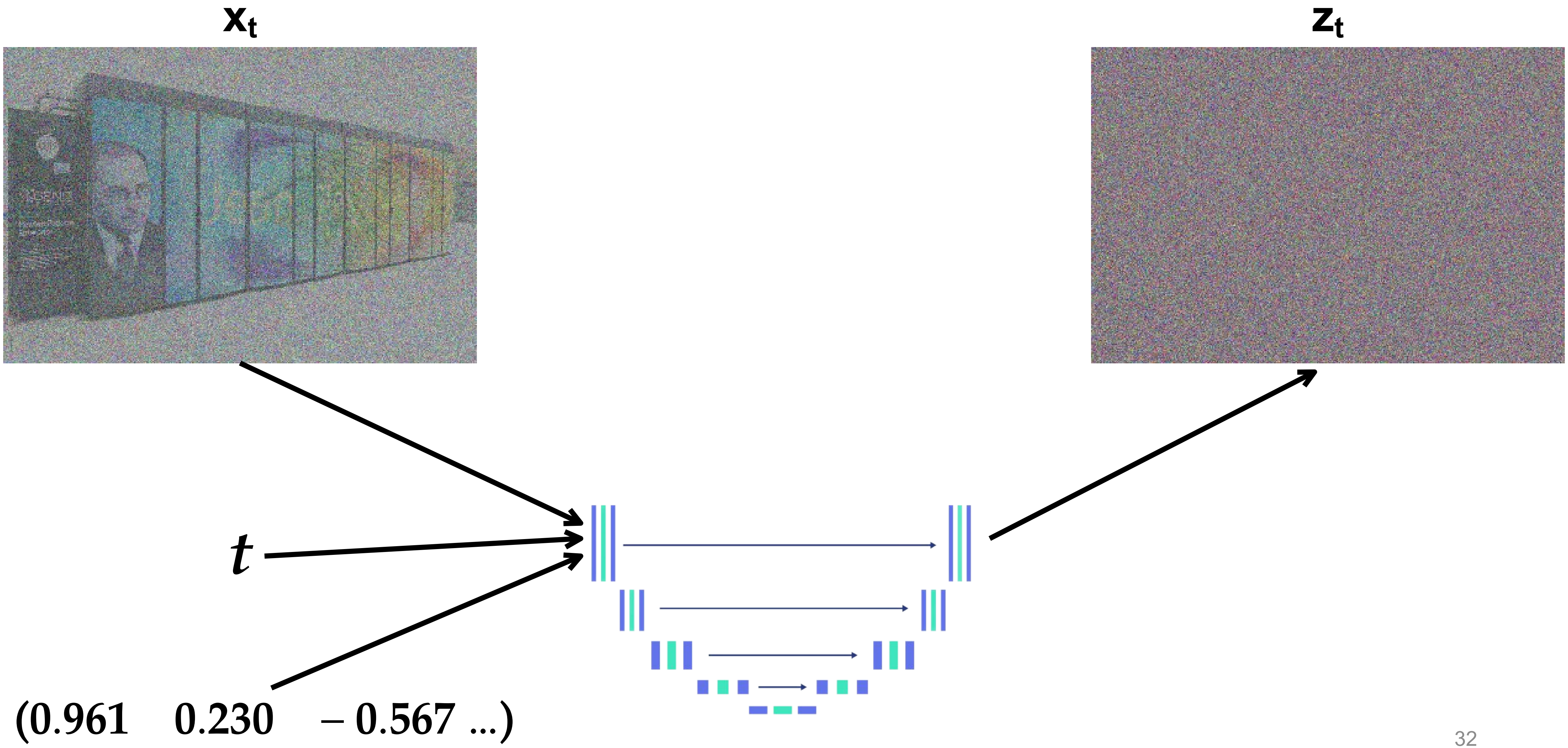
↓

Text embedding model

↓

(0.961 0.230 – 0.567 ...)

Text-to-image



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