ITCS 6156/8156 Spring 2024
Machine Learning

Deep Generative Models

Instructor: Hongfei Xue
Email: hongfei.xue@charlotte.edu
Class Meeting: Mon & Wed, 4:00 PM – 5:15 PM, Denny 109

UNIVERSITY OF NORTH CAROLINA CHARLOTTE

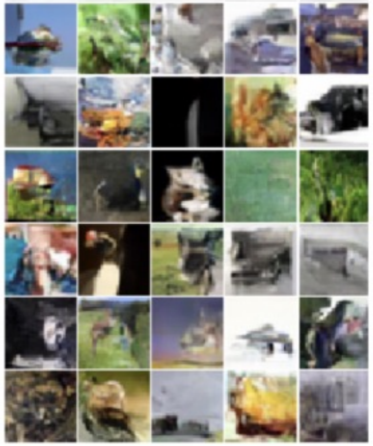
Some content in the slides is based on Dr. Ruohan Gao's lectures

Where We Came From

VAEs, 2013

GANs, 2014

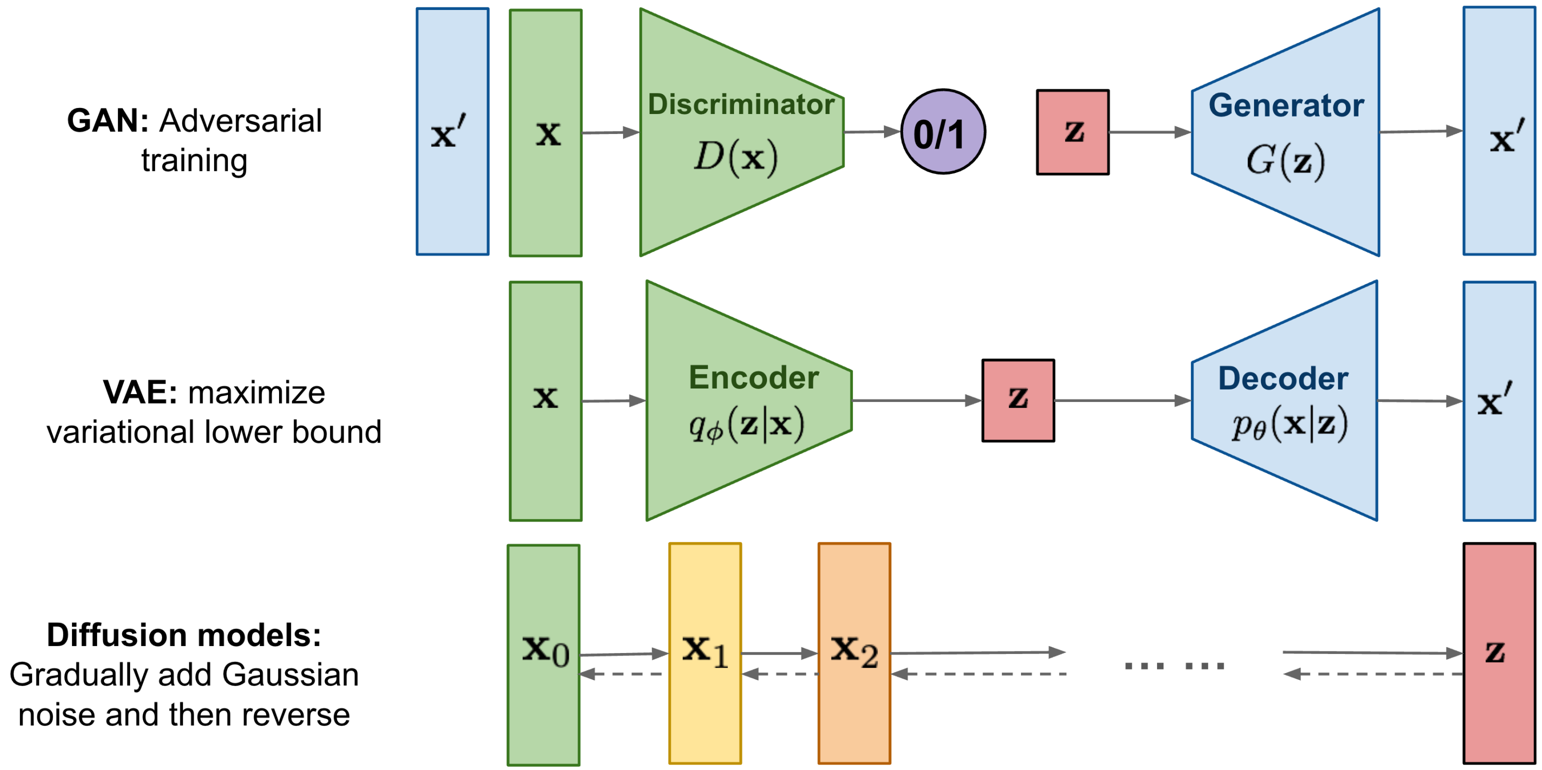
PixelCNN, 2016



BigGAN, 2019

Imagen, 2022

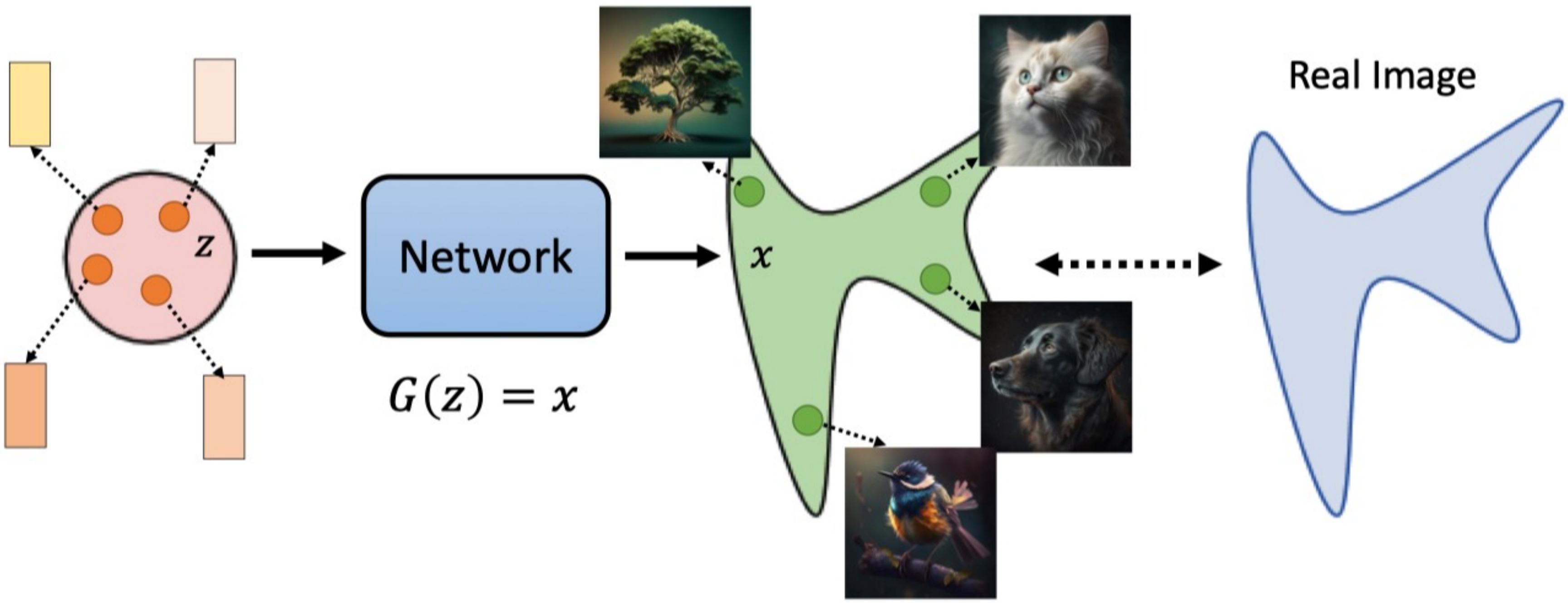
Generative Models



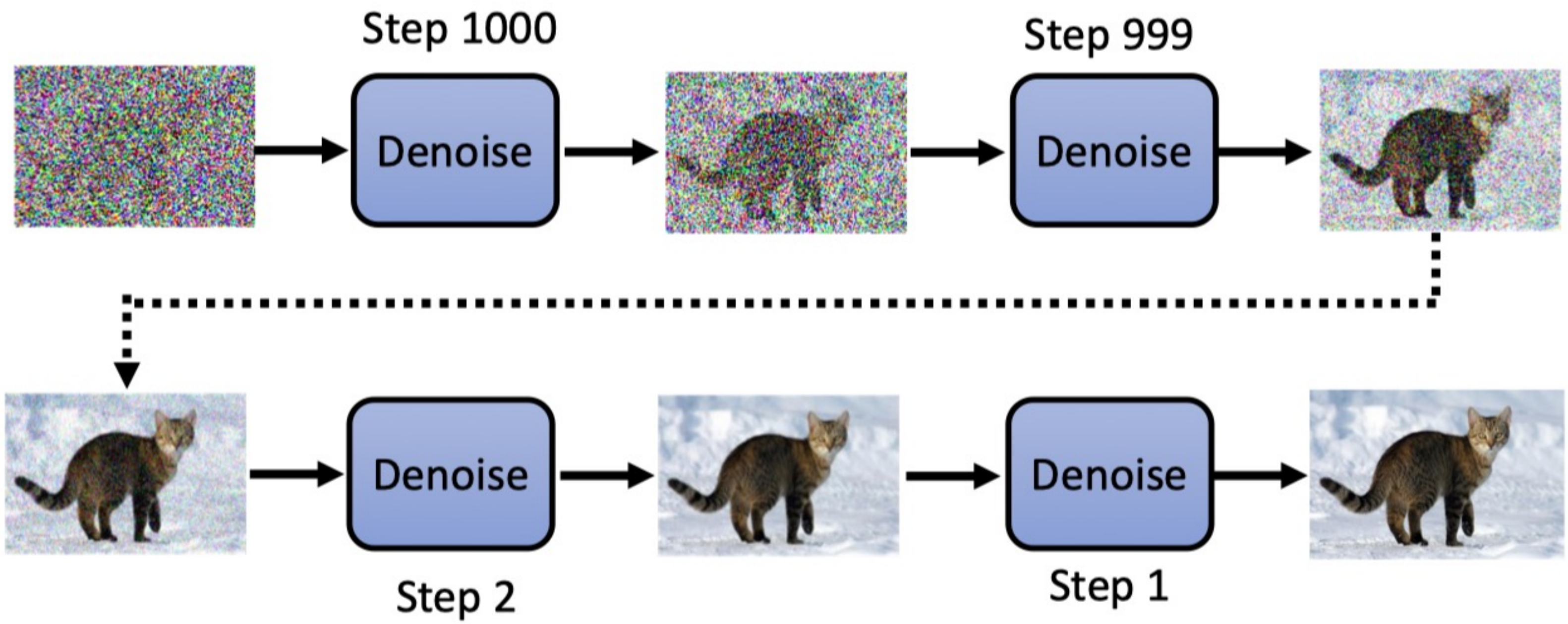
GAN: Hard to train two networks; hard to converge; biased discriminator

Denoising Diffusion Probabilistic Models (DDPM), 2020, https://arxiv.org/abs/2006.11239
Image source: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

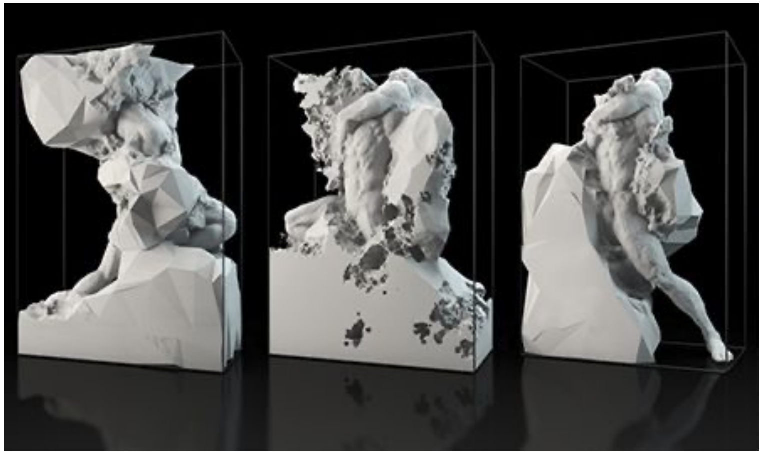
Goal of Generative Model



How the Diffusion Model Works?



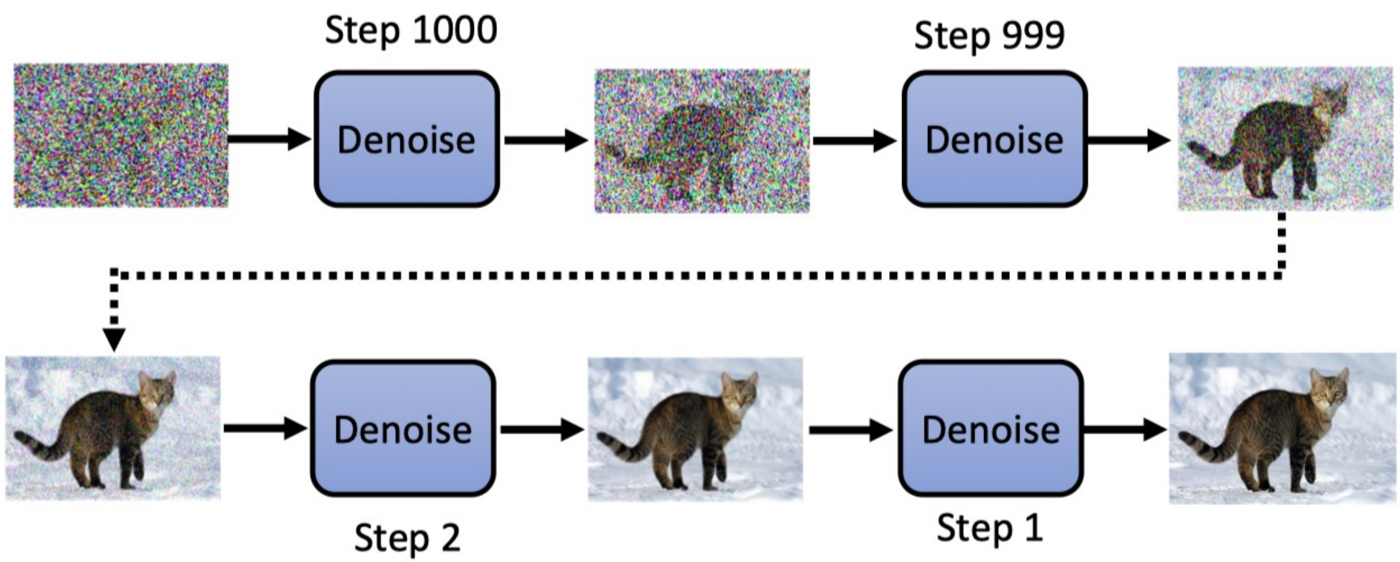
Reverse Process

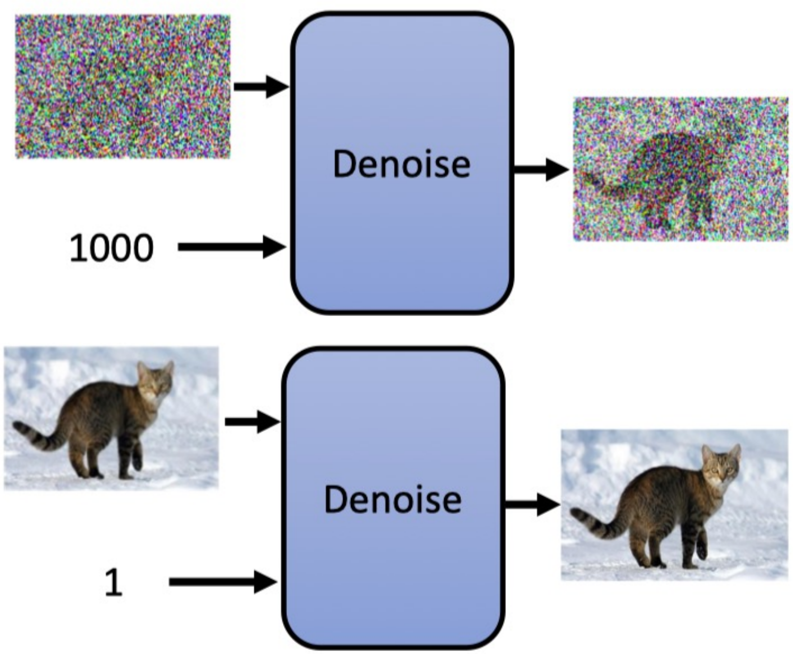


The sculpture is already complete within the marble block, before I start my work. It is already there, I just have to chisel away the superfluous material.

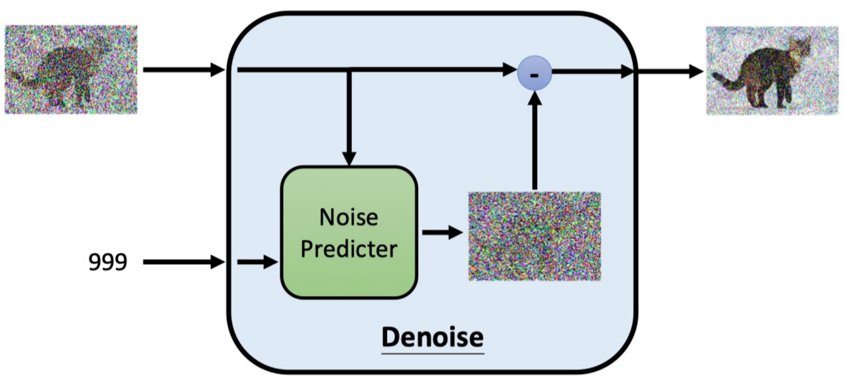
— **Michelangelo**

Denoising (Reverse) Process

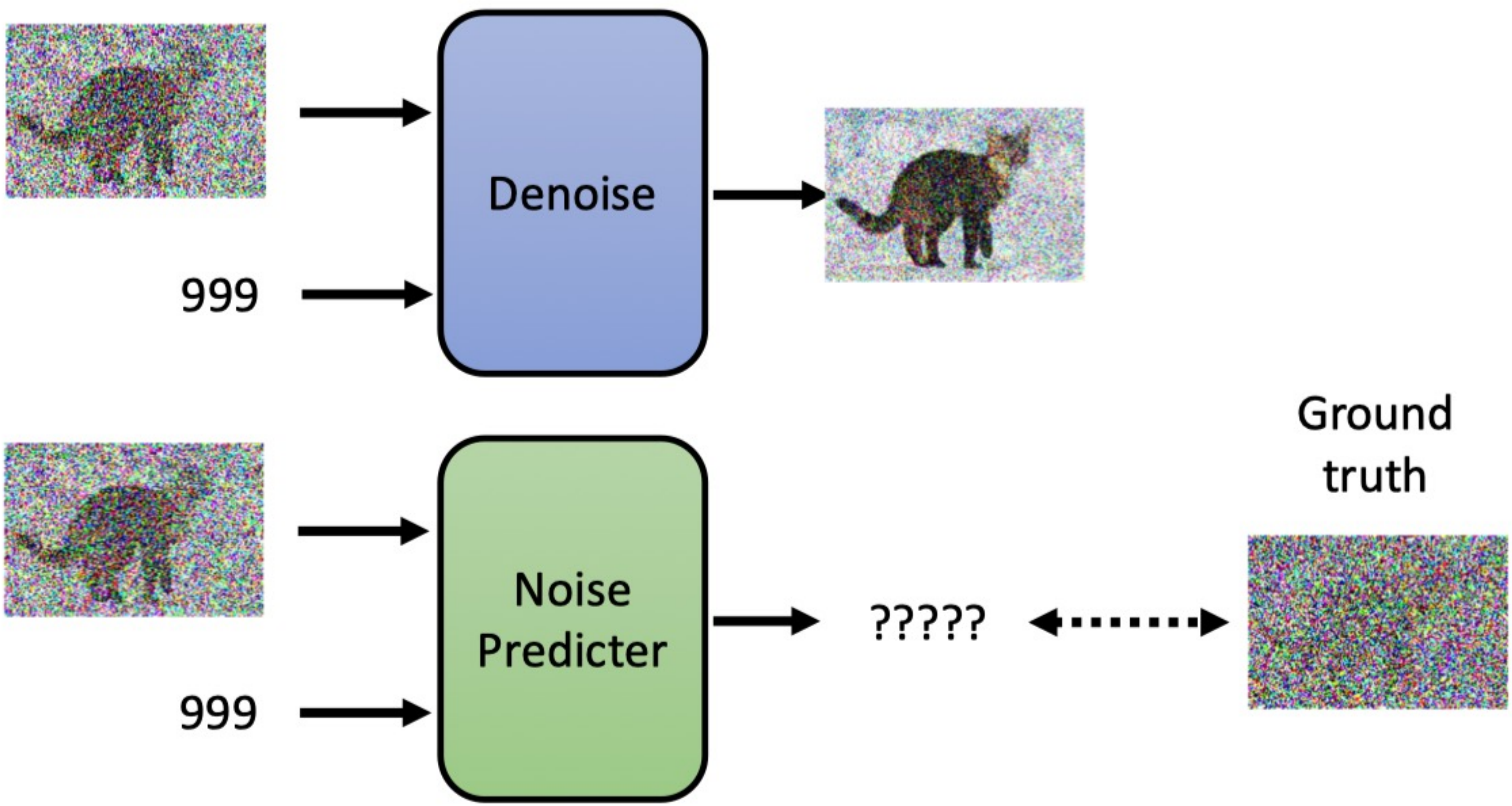




Denoise:

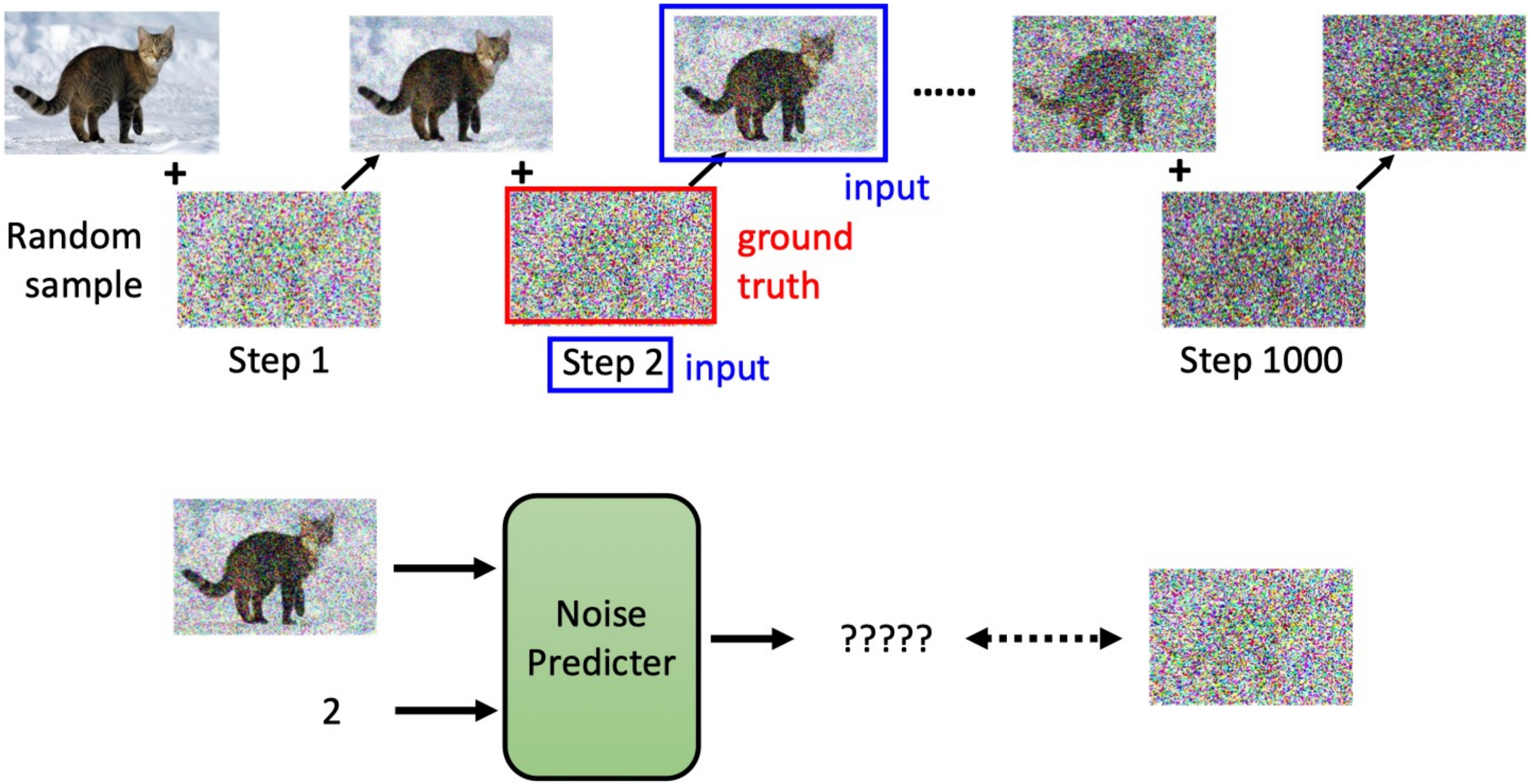


Train a Noise Predictor

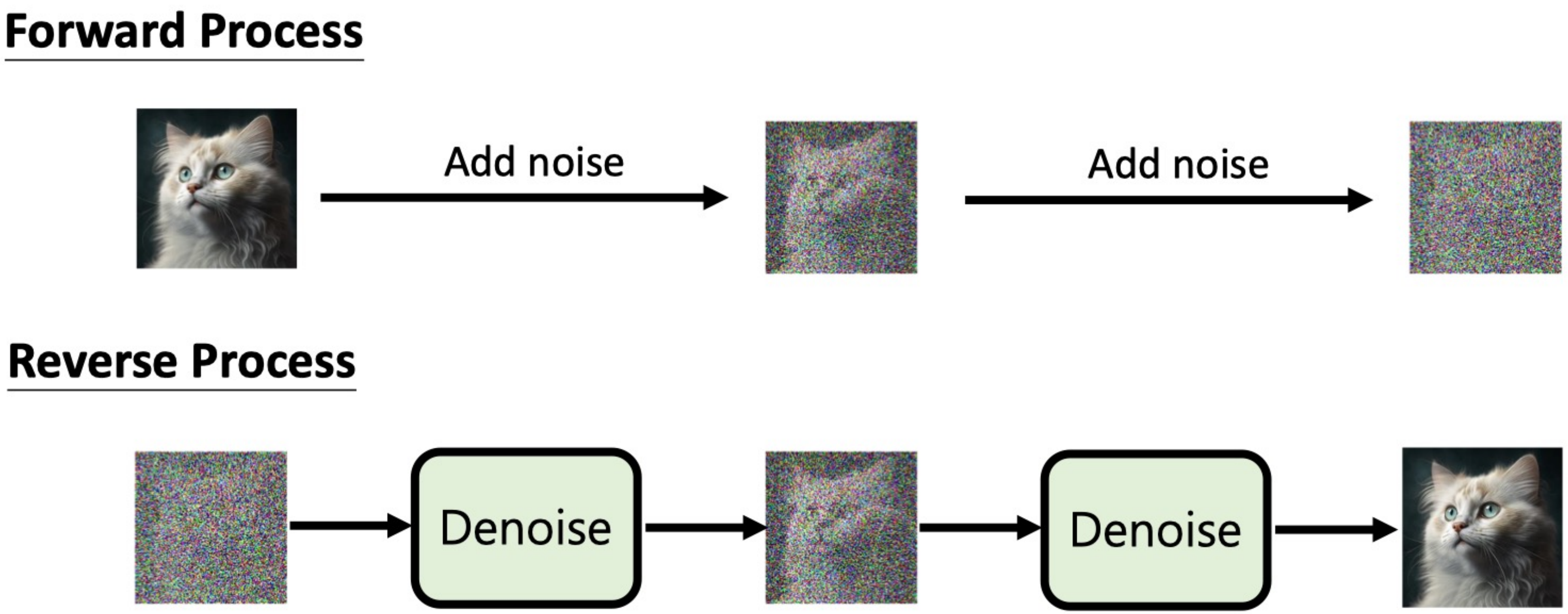


Forward/Diffusion Process

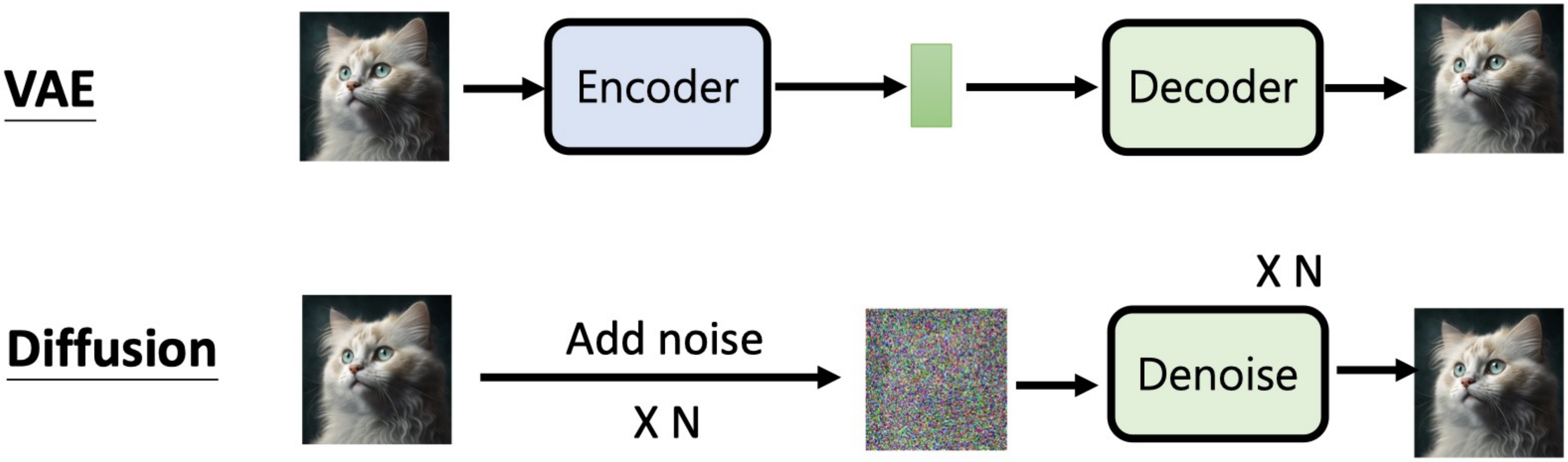
Forward/Diffusion Process:



Diffusion Model



VAE vs. Diffusion Model



Denoising Diffusion Probabilistic Models (DDPM)

Algorithm 1 Training

1: **repeat**
2: $\quad \mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $\quad t \sim \mathrm{Uniform}(\{1, \ldots, T\})$
4: $\quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: $\quad$ 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

Algorithm 2 Sampling

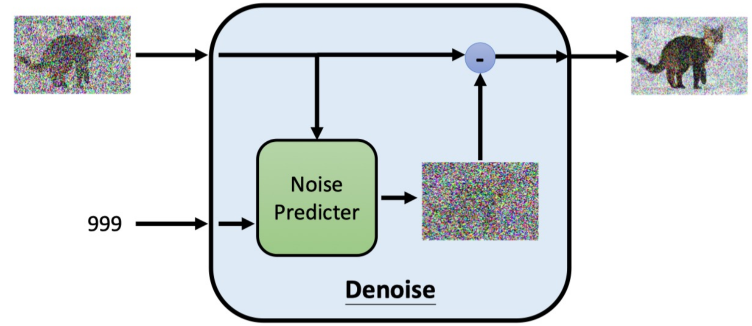
1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,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

Denoising Diffusion Probabilistic Models (DDPM)

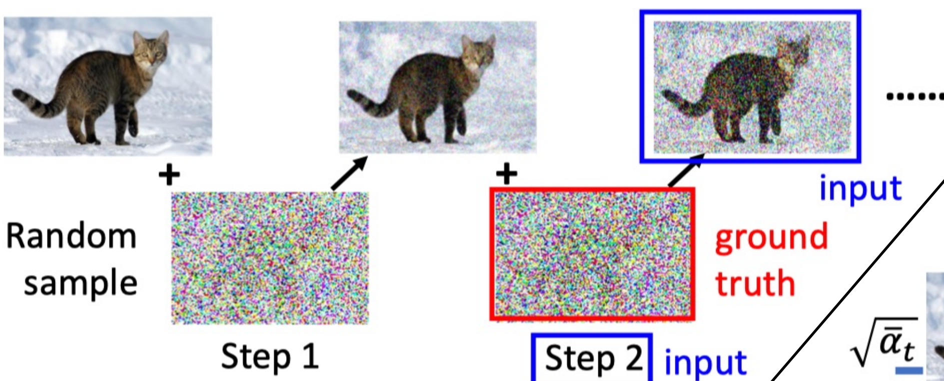
Algorithm 1 Training

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

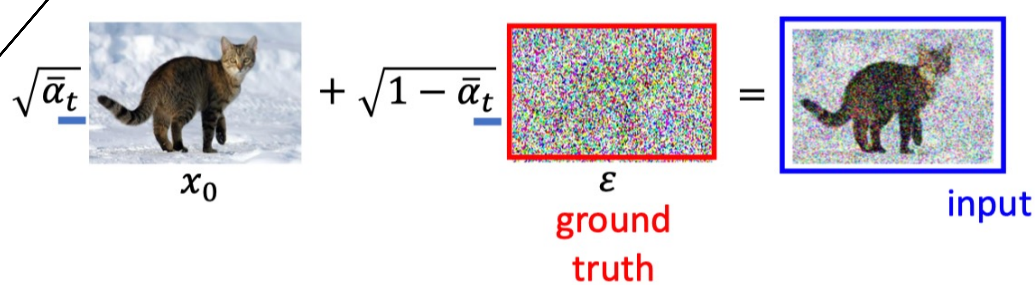
Why x_0 not x_{t-1}?



What I told you:



Real implementation:

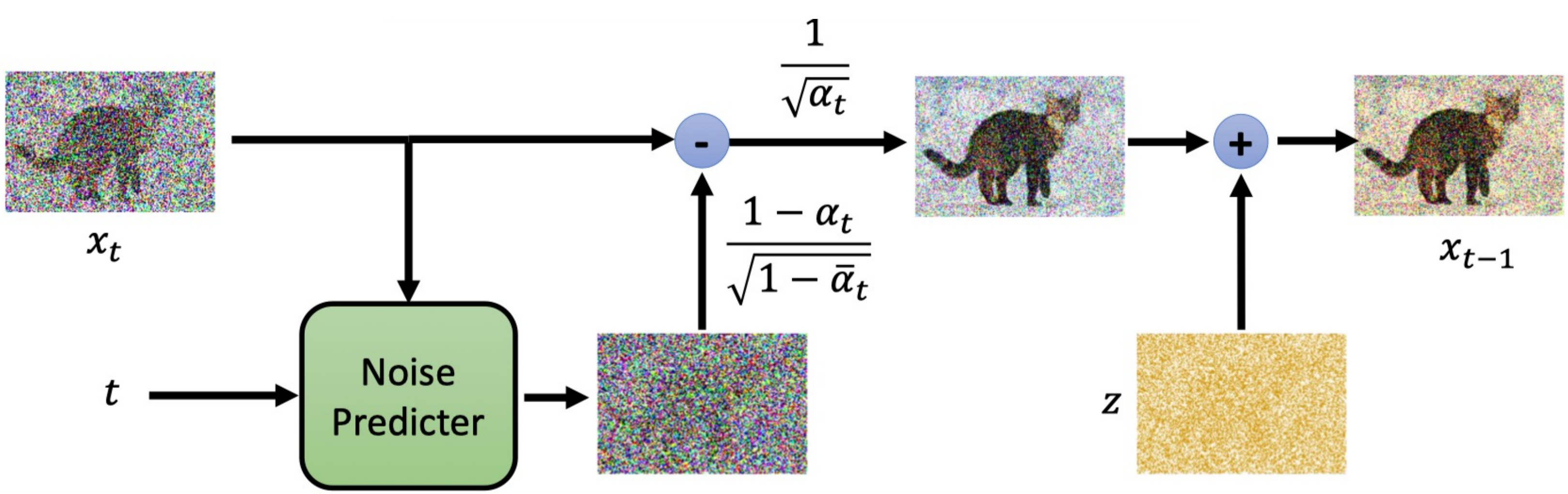


Denoising Diffusion Probabilistic Models (DDPM)

Algorithm 2 Sampling

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

Sample and add a noise during the denoising steps?!



Probabilistic Explanation

$$P_\theta(x_0) = \int_{x_1:x_T} P(x_T)P_\theta(x_{T-1}|x_T)\dots P_\theta(x_{t-1}|x_t)\dots P_\theta(x_0|x_1)dx_1{:}\,x_T$$

$$\begin{aligned}
\log p(\boldsymbol{x}) &= \log \int p(\boldsymbol{x}_{0:T})d\boldsymbol{x}_{1:T} \\
&= \log \int \frac{p(\boldsymbol{x}_{0:T})q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}d\boldsymbol{x}_{1:T} \\
&= \log \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] \\
&\geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=2}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})\prod_{t=1}^{T-1} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\prod_{t=1}^{T-1} p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})\prod_{t=1}^{T-1} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \prod_{t=1}^{T-1} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\sum_{t=1}^{T-1} \log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})}\right] + \sum_{t=1}^{T-1} \mathbb{E}_{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right] + \mathbb{E}_{q(\boldsymbol{x}_{T-1},\boldsymbol{x}_T|\boldsymbol{x}_0)}\left[\log \frac{p(\boldsymbol{x}_T)}{q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1})}\right] + \sum_{t=1}^{T-1} \mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_t,\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \\
&= \underbrace{\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{x}_0|\boldsymbol{x}_1)\right]}_{\text{reconstruction term}} - \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{T-1}|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_T|\boldsymbol{x}_{T-1}) \parallel p(\boldsymbol{x}_T))\right]}_{\text{prior matching term}} \\
&\quad - \sum_{t=1}^{T-1} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_{t-1},\boldsymbol{x}_{t+1}|\boldsymbol{x}_0)}\left[D_{\mathrm{KL}}(q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \parallel p_{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}))\right]}_{\text{consistency term}}
\end{aligned}$$

VAE Maximize $logP_\theta(x)$ $\longrightarrow$ Maximize $\mathbb{E}_{\boxed{q(z|x)}}\left[log\left(\frac{P(x,z)}{q(z|x)}\right)\right]$

Encoder

DDPM Maximize $logP_\theta(x_0)$ $\longrightarrow$ Maximize $\mathbb{E}_{\boxed{q(x_1:x_T|x_0)}}\left[log\left(\frac{P(x_0:x_T)}{q(x_1:x_T|x_0)}\right)\right]$

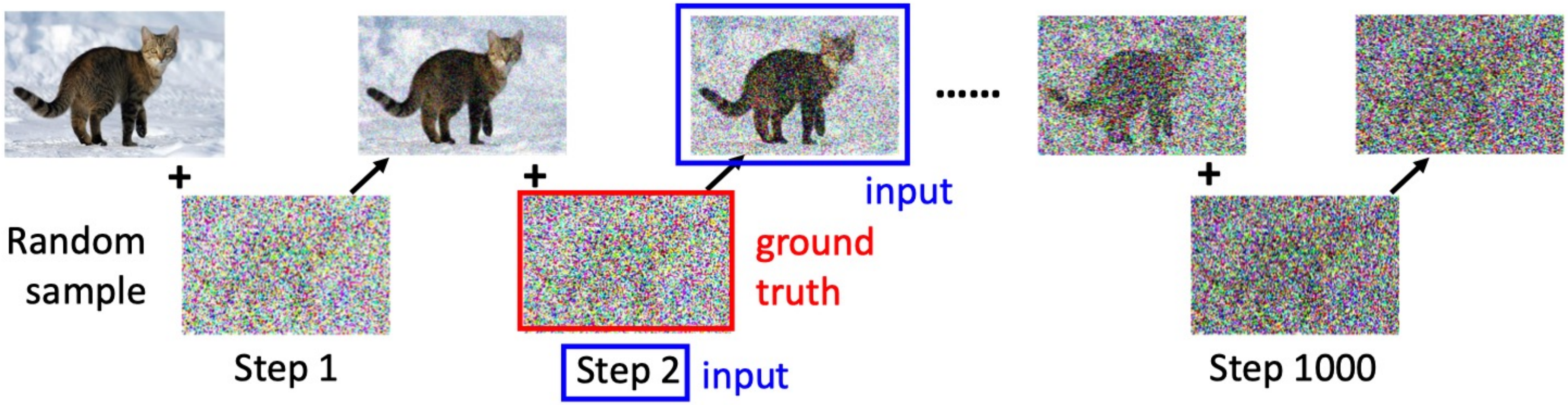
Forward Process
(Diffusion Process)

$$q(x_1:x_T|x_0) = q(x_1|x_0)q(x_2|x_1)\dots q(x_T|x_{T-1})$$

https://arxiv.org/pdf/2208.11970.pdf

Forward/Diffusion Process

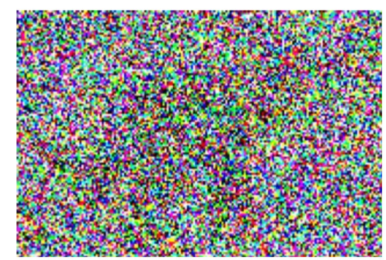
- We add noise step by step:



- We have α_t to control how much noise we want to add.

$= \sqrt{\alpha_t}$

$+\sqrt{1-\alpha_t}$



x_t x_{t-1} z_t

- Equation: $x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t} z_1$
- α_t decreases when t increases.

Forward/Diffusion Process

- We have α_t to control how much noise we want to add.

x_t

$= \sqrt{\alpha_t}$

x_{t-1}

$+\sqrt{1-\alpha_t}$

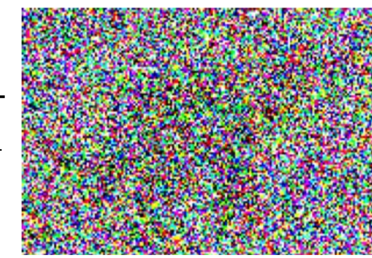
z_t

x_{t-1}

$= \sqrt{\alpha_{t-1}}$

x_{t-2}

$+\sqrt{1-\alpha_{t-1}}$



z_{t-1}

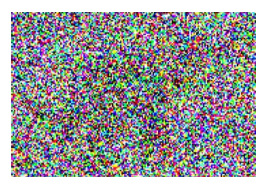
- Combine them, we have:

x_t

$=\sqrt{\alpha_t \alpha_{t-1}}$

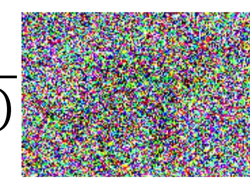
x_{t-2}

$+\sqrt{1-\alpha_t}$



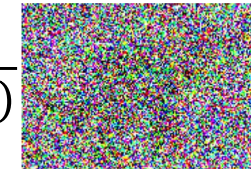
z_t

$+\sqrt{\alpha_t(1-\alpha_{t-1})}$



z_{t-1}

Forward/Diffusion Process

 $= \sqrt{\alpha_t \alpha_{t-1}}$  $+ \sqrt{1-\alpha_t}$  $+ \sqrt{\alpha_t(1-\alpha_{t-1})}$ 

x_t x_{t-2} z_t z_{t-1}

- Let's formulate it:

$$\begin{aligned} x_t &= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_{t-1}}z_{t-1}\right) + \sqrt{1-\alpha_t}z_t \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \left(\sqrt{\alpha_t(1-\alpha_{t-1})}z_{t-1} + \sqrt{1-\alpha_t}z_t\right) \end{aligned}$$

- We know that $z_t, z_{t-1}, \ldots, \sim \mathcal{N}(0, I)$.
- So $\sqrt{\alpha_t(1-\alpha_{t-1})}z_{t-1} \sim \mathcal{N}\left(0, \alpha_t(1-\alpha_{t-1})\right)$, and $\sqrt{1-\alpha_t}z_t \sim \mathcal{N}(0, 1-\alpha_t)$
- We also know that $\mathcal{N}(0, {\sigma_1}^2 I) + \mathcal{N}(0, {\sigma_2}^2 I) = \mathcal{N}(0, ({\sigma_1}^2 + {\sigma_2}^2)I)$.

$$\begin{aligned} x_t &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \left(\sqrt{\alpha_t(1-\alpha_{t-1})}z_{t-1} + \sqrt{1-\alpha_t}z_t\right) \\ &= \sqrt{\alpha_t\alpha_{t-1}}x_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\tilde{z}_{t-1} \\ &= \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\tilde{z}_1 \end{aligned}$$

Where $\bar{\alpha}_t = \alpha_t\alpha_{t-1}, \ldots, \alpha_1$

$z_t, z_{t-1}, \ldots, \sim \mathcal{N}(0, I)$

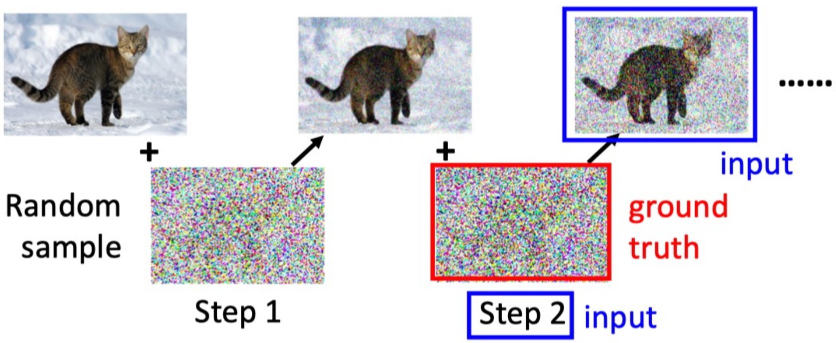
$\tilde{z}_t, \tilde{z}_{t-1}, \ldots, \sim \mathcal{N}(0, I)$

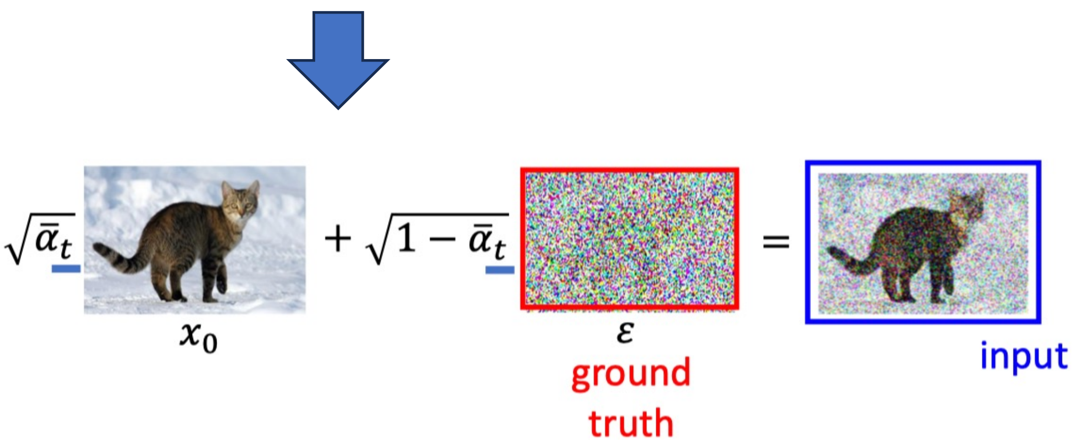
Forward/Diffusion Process

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\tilde{z}_1$$

Where $\bar{\alpha}_t = \alpha_t\alpha_{t-1}, \dots, \alpha_1$

$$\tilde{z}_t, \tilde{z}_{t-1}, \dots, \sim \mathcal{N}(0, I)$$





Algorithm 1 Training

1: **repeat**

2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$

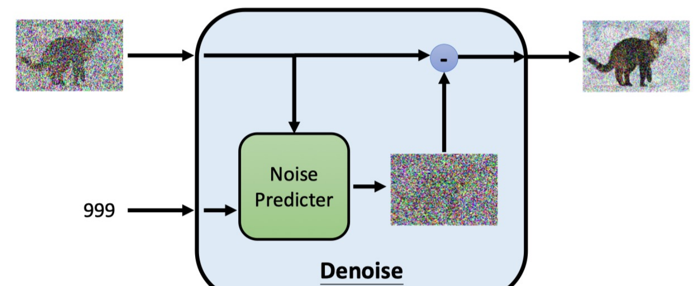
3: $t \sim \mathrm{Uniform}(\{1, \dots, T\})$

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



Denoising/Reverse Process

- Goal: $q(x_{t-1}|x_t)$, but we don't know how to calculate it. We only know $q(x_t|x_{t-1})$.
- Using Bayes Rule we have:

$$q(x_{t-1}|x_t) = q(x_t|x_{t-1})\frac{\boxed{q(x_{t-1})}}{\boxed{q(x_t)}}$$

Hard to model directly.

- Instead, we can model $q(x_{t-1}|x_t, x_0)$
- Using Bayes Rule we have:

$$q(x_{t-1}|x_t, x_0) = q(x_t|x_{t-1}, x_0)\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)}$$

- For each term, we have:

$$\boxed{x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\tilde{z}_1}$$

$$q(x_{t-1}|x_0) = \sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1-\bar{\alpha}_{t-1}}z \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0, 1-\bar{\alpha}_{t-1}\right)$$
$$q(x_t|x_0) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}x_0, 1-\bar{\alpha}_t\right)$$
$$q(x_t|x_{t-1}, x_0) = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}z \sim \mathcal{N}\left(\sqrt{\alpha_t}x_{t-1}, 1-\alpha_t\right)$$

- So, we have:

$$q(x_{t-1}|x_t, x_0) \propto \exp\left(-\frac{1}{2}\left(\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t} + \frac{\left(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{1-\bar{\alpha}_{t-1}} - \frac{\left(x_t-\sqrt{\bar{\alpha}_t}x_0\right)^2}{1-\bar{\alpha}_t}\right)\right), \text{ let } 1-\alpha_t = \beta_t$$

$$= \exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right){x_{t-1}}^2 - \left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t + \frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1} + C(x_t, x_0)\right)\right), \text{ C is a constant}$$

Denoising/Reverse Process

$$q(x_{t-1}|x_t,x_0) \propto \exp(-\frac{1}{2}(\frac{(x_t-\sqrt{\alpha_t}x_{t-1})^2}{\beta_t}+\frac{\left(x_{t-1}-\sqrt{\bar{\alpha}_{t-1}}x_0\right)^2}{1-\bar{\alpha}_{t-1}}-\frac{\left(x_t-\sqrt{\bar{\alpha}_t}x_0\right)^2}{1-\bar{\alpha}_t})),\text{ let } 1-\alpha_t=\beta_t$$

$$=\exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right){x_{t-1}}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)x_{t-1}+C(x_t,x_0)\right)\right),\text{ C is a constant}$$

- For normal distribution we have: $\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)=\exp(-\frac{1}{2}(\frac{1}{\sigma^2}x^2-\frac{2\mu}{\sigma^2}x+\frac{\mu^2}{\sigma^2}))$
- So, we have:

$$\sigma^2=\frac{1}{\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)}$$ is a constant

$$\frac{2\mu}{\sigma^2}=\left(\frac{2\sqrt{\alpha_t}}{\beta_t}x_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}x_0\right)$$

We can estimate x_{t-1} from x_t, x_0

We don't know this in reverse process

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

Actually, we even don't need the reverse process if we know this. LOL.

Denoising/Reverse Process

$$\sigma^2 = \frac{1}{\left(\frac{\alpha_t}{\beta_t} + \frac{1}{1-\bar{\alpha}_{t-1}}\right)} \qquad \tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} x_0$$

- But we have: $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t} z_t$
- So, $x_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(x_t - \sqrt{1-\bar{\alpha}_t} z_t)$

Estimated by the neural network

- Finally, we have: $$\tilde{\mu}_t = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \boxed{z_t}\right)$$

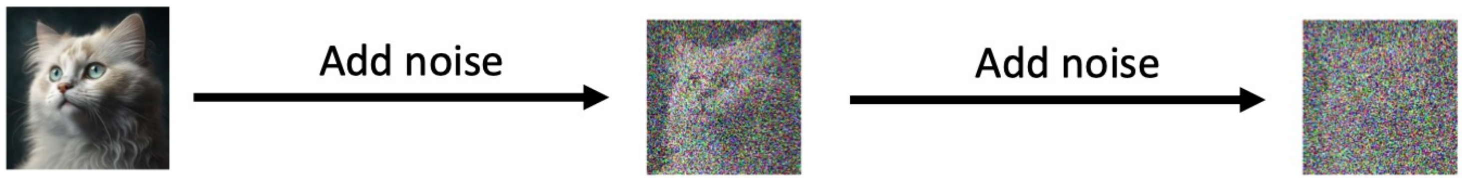
Algorithm 2 Sampling

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: $\quad \mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\right) + \boxed{\sigma_t \mathbf{z}}$
5: **end for**
6: **return** $\mathbf{x}_0$

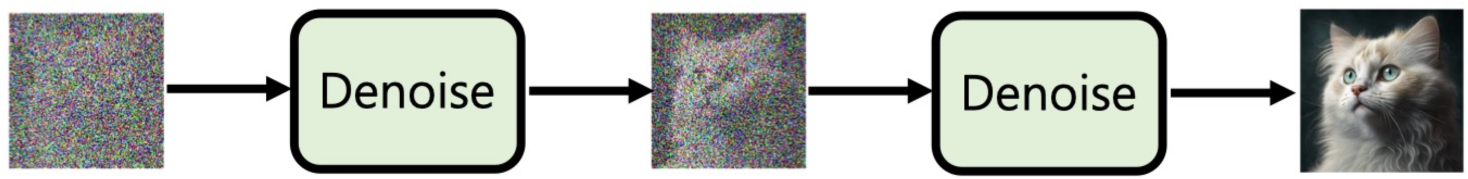
Sampling from the data distribution

Diffusion Model

Forward Process



Reverse Process



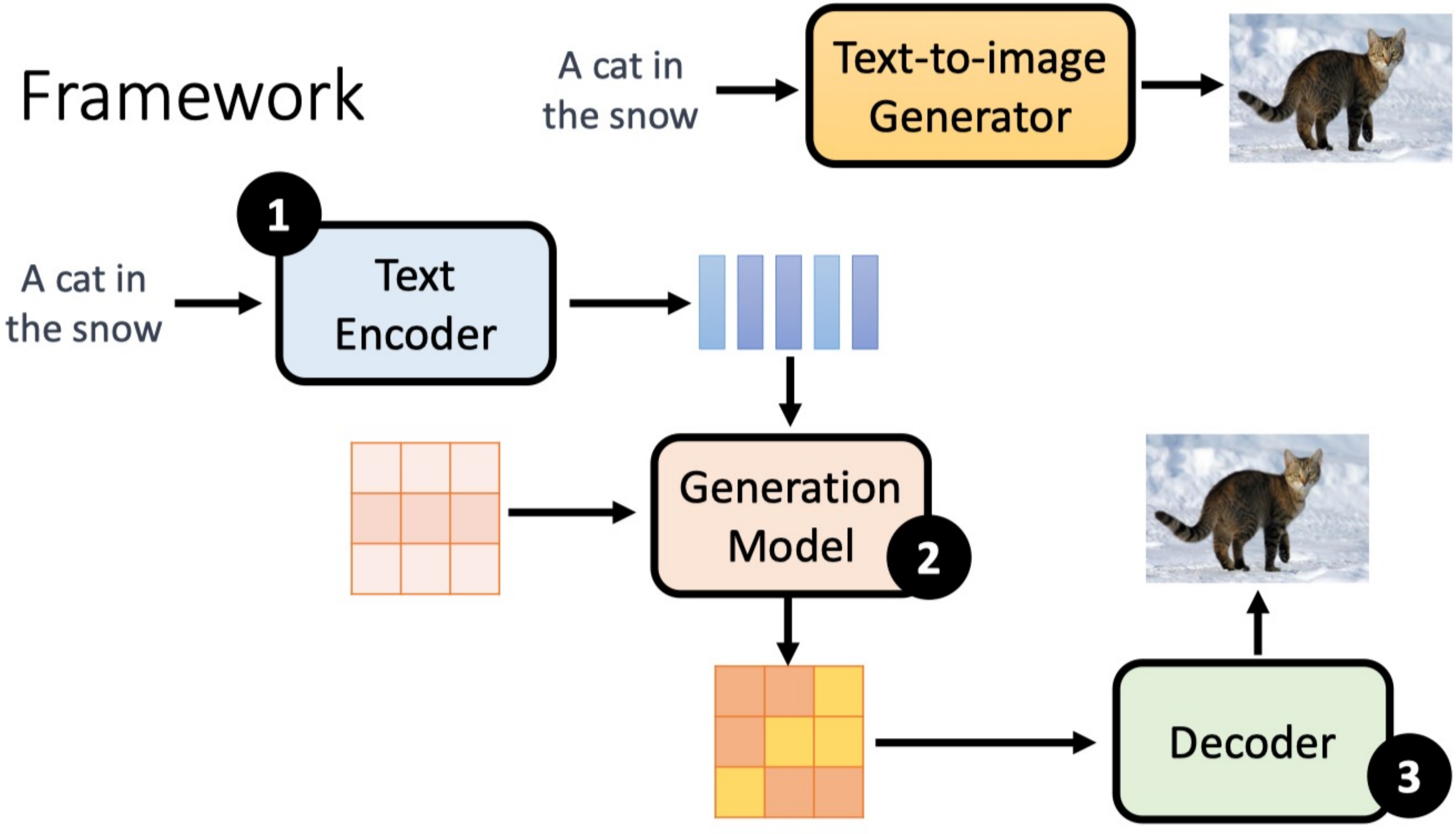
Algorithm 1 Training

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

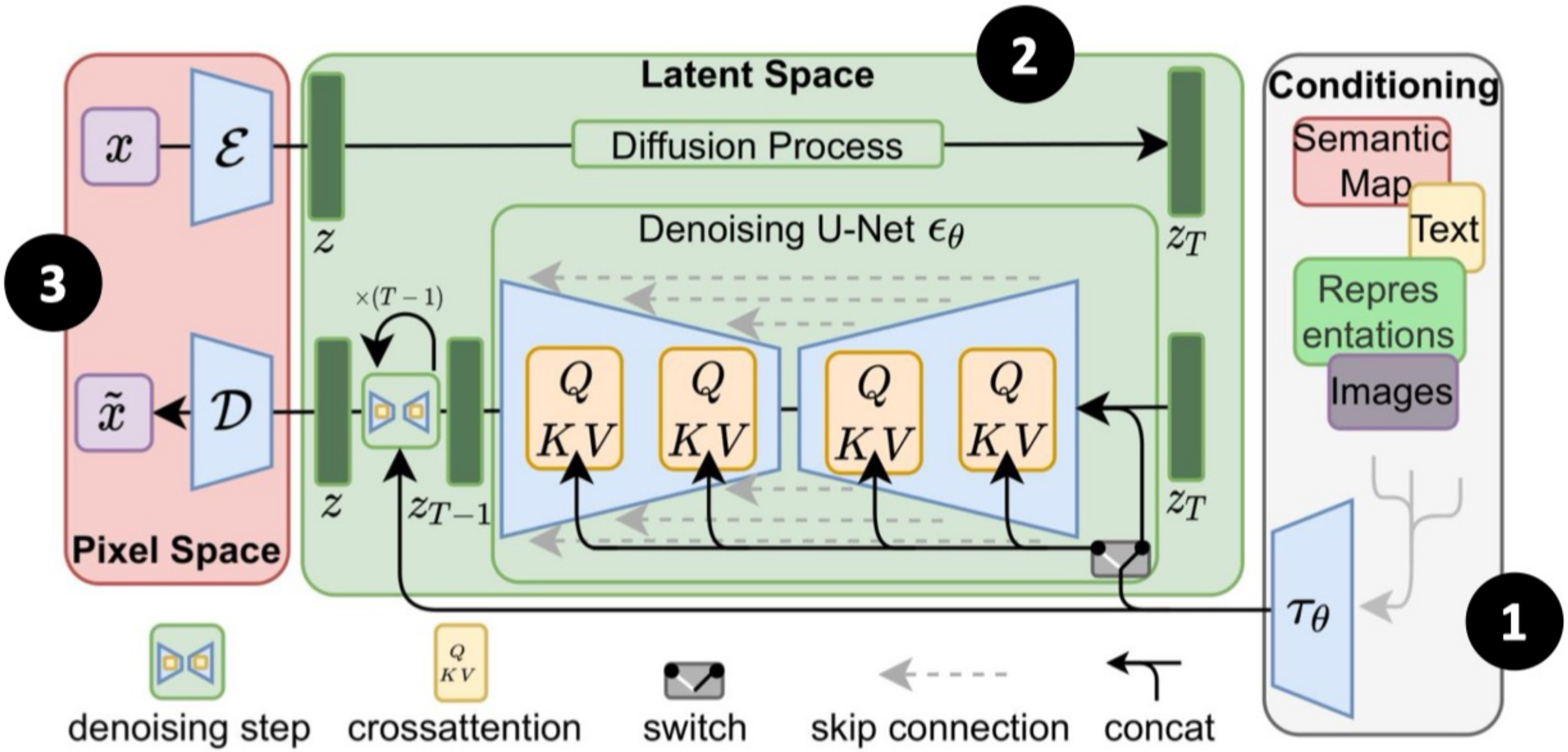
Algorithm 2 Sampling

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
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 $\mathbf{x}_0$

Stable Diffusion

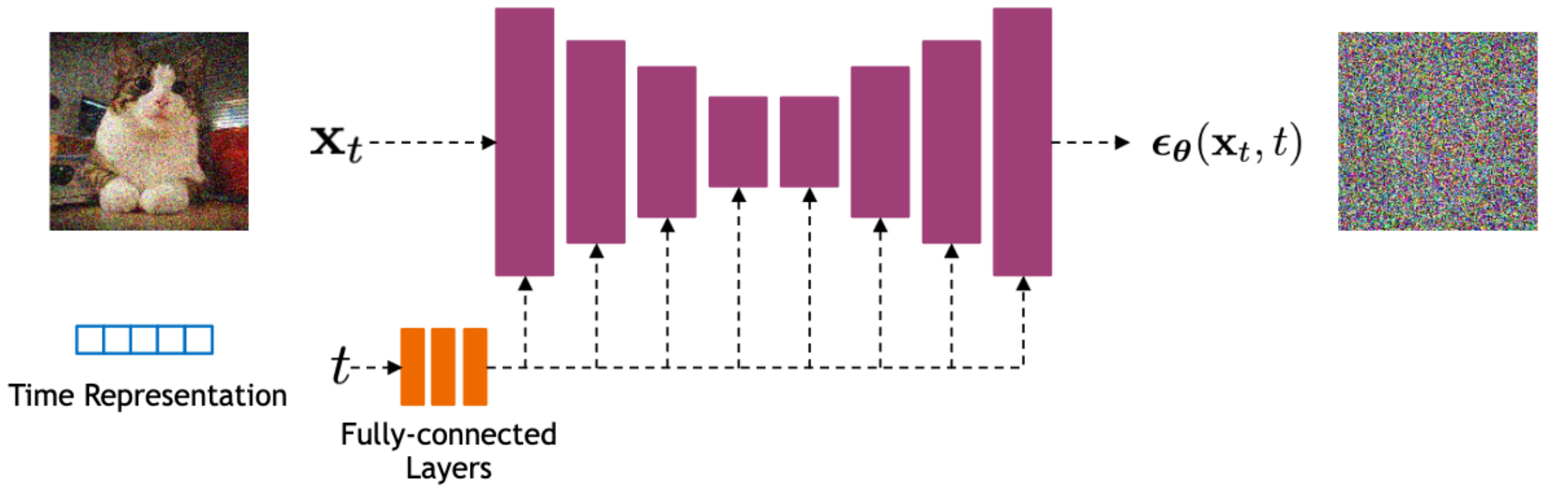


Stable Diffusion



Network Architectures

Diffusion models often use U-Net architectures with ResNet blocks and self-attention layers to represent $\epsilon_\theta(\mathbf{x}_t, t)$



Time representation: sinusoidal positional embeddings or random Fourier features.

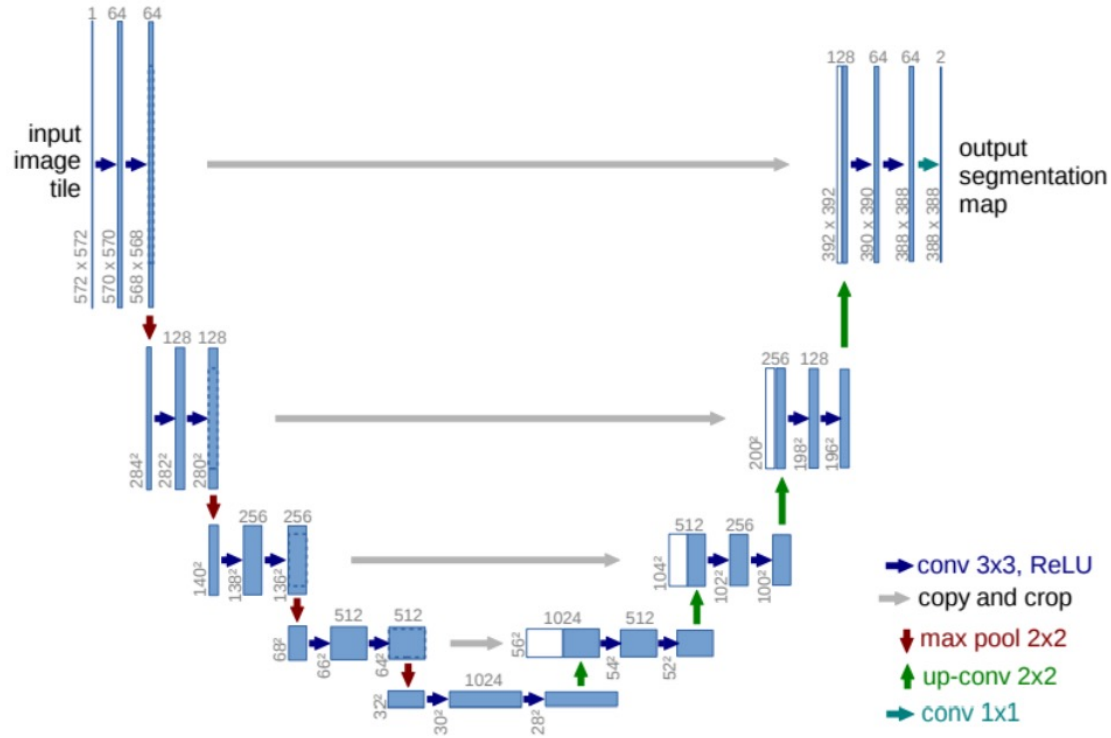
U-Net

Contracting path

- block consists of:
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
  - max-pooling with stride of 2 (downsample)
- repeat the block N times, doubling number of channels

Expanding path

- block consists of:
  - 2x2 convolution (upsampling)
  - concatenation with contracting path features
  - 3x3 convolution
  - 3x3 convolution
  - ReLU
- repeat the block N times, halving the number of channels



U-Net

- Originally designed for applications to biomedical segmentation
- Key observation is that the output layer has the **same** dimensions as the input image (possibly with different number of channels)

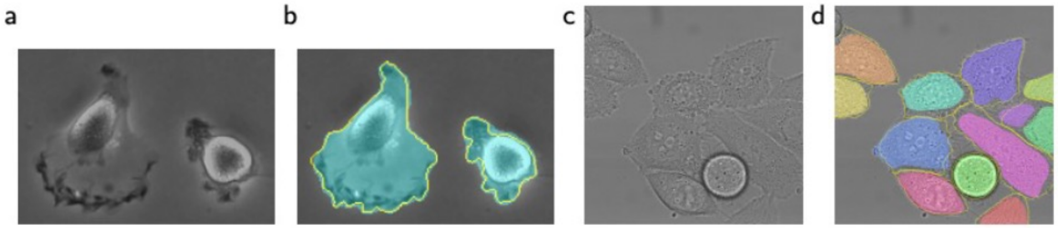
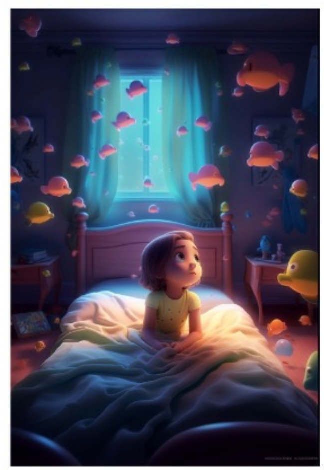
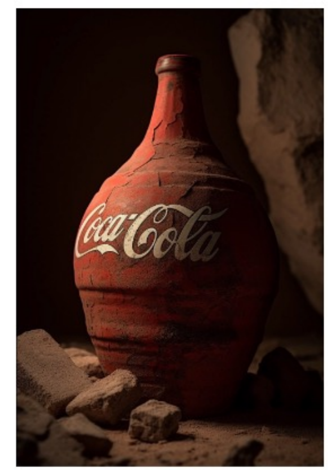


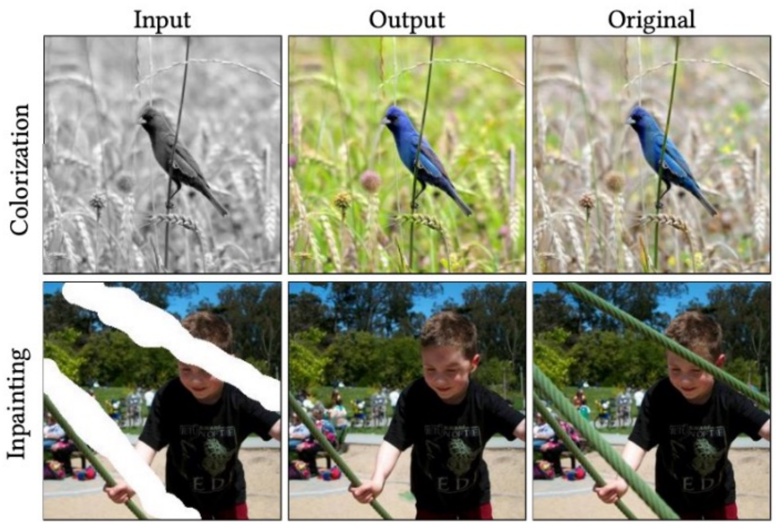
Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

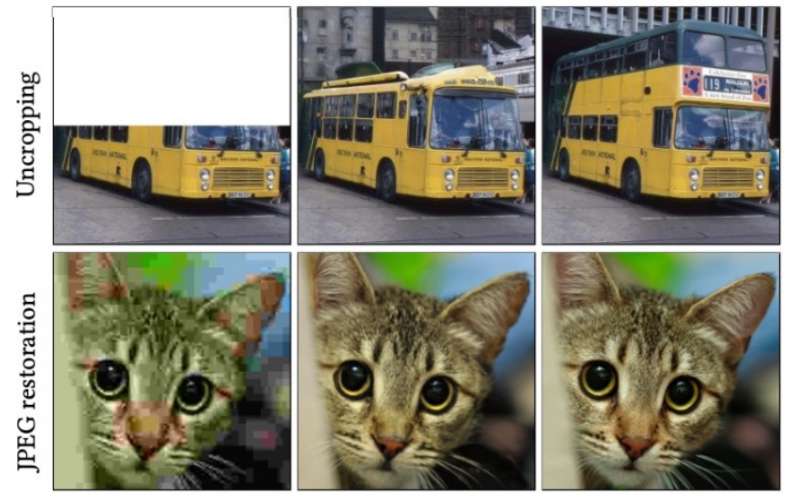
Applications: AI Art

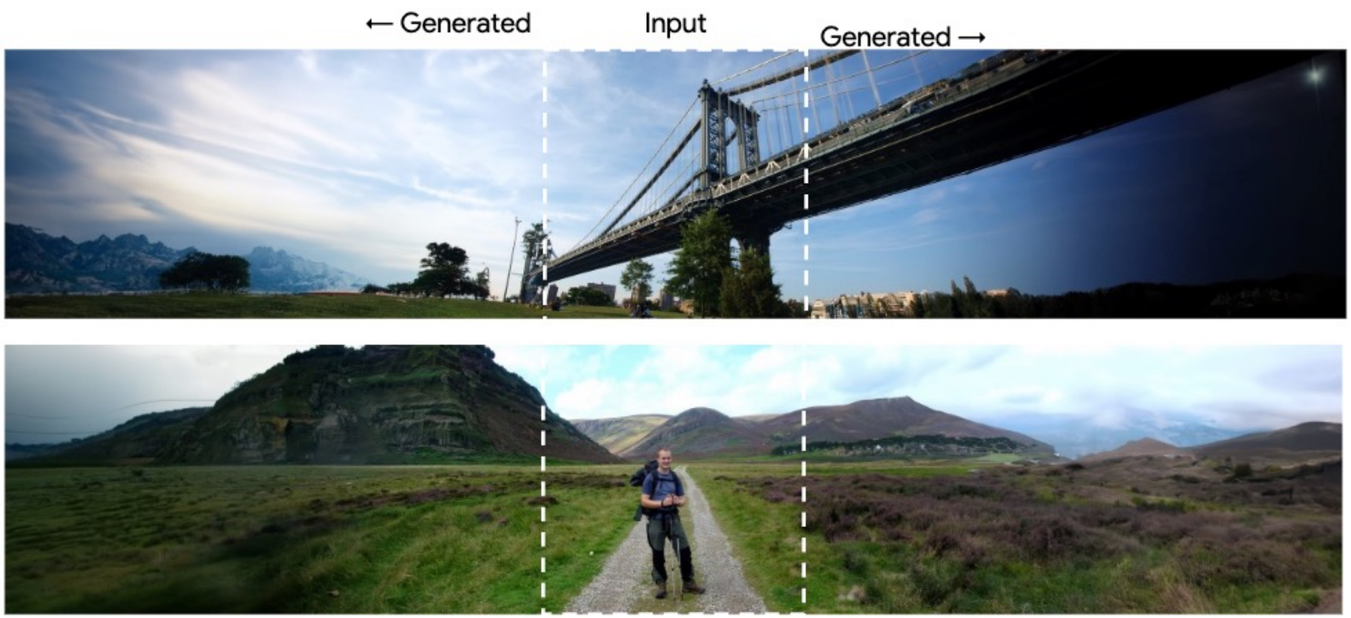




Applications: Colorization, Inpainting, Restoration, Outfilling







Questions?

UNIVERSITY OF NORTH CAROLINA
CHARLOTTE