Denoising Diffusion Probabilistic and Implicit Models

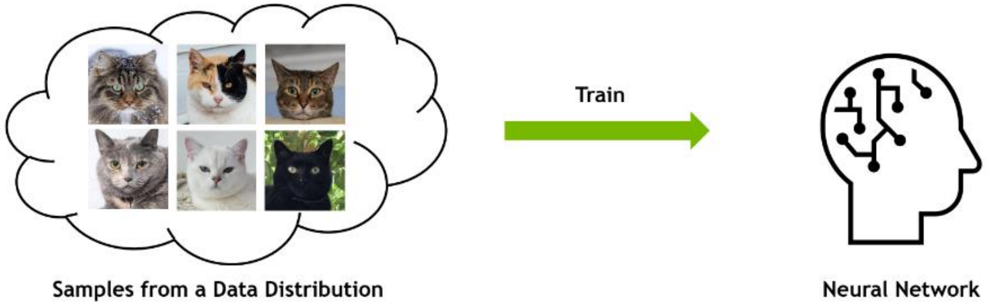
Presented by: Itamar Salazar-Reque

Outline

- The problem
- The solution
- Summary of DDPM and DDIM
- Mathematical Details
- Results

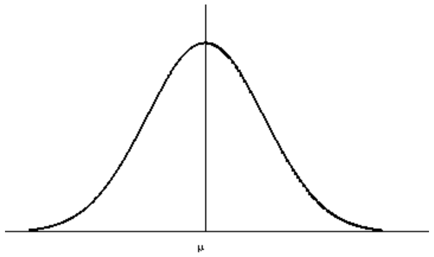
The problem

- Learning to generate data

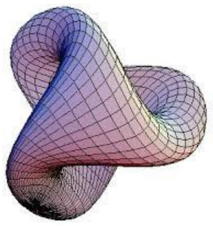


- Why is this a problem?

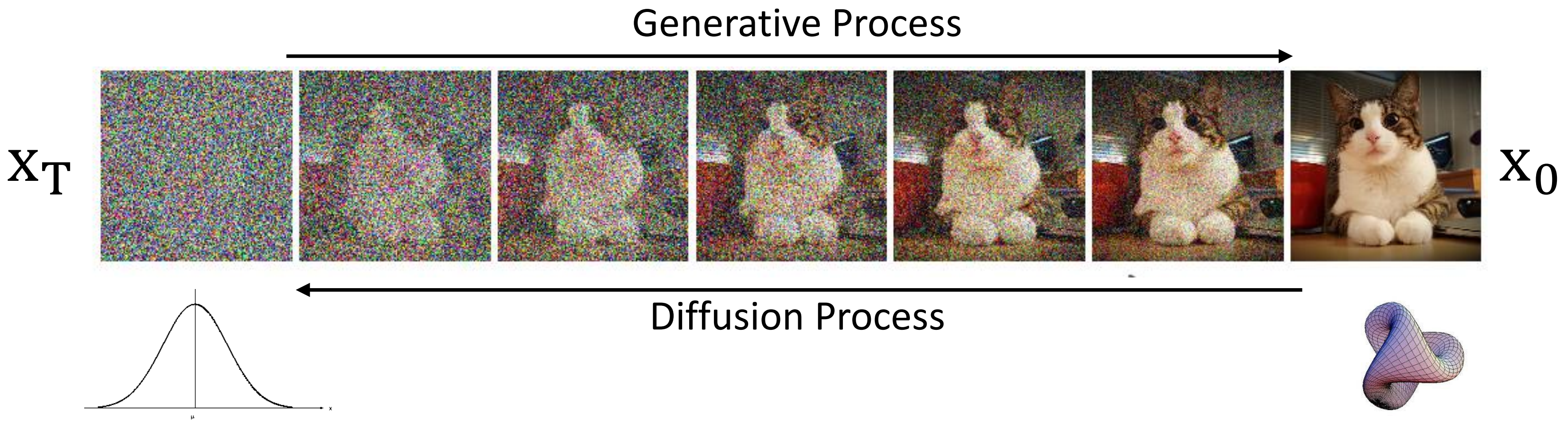
Sampling from this is easy because we know the distribution.

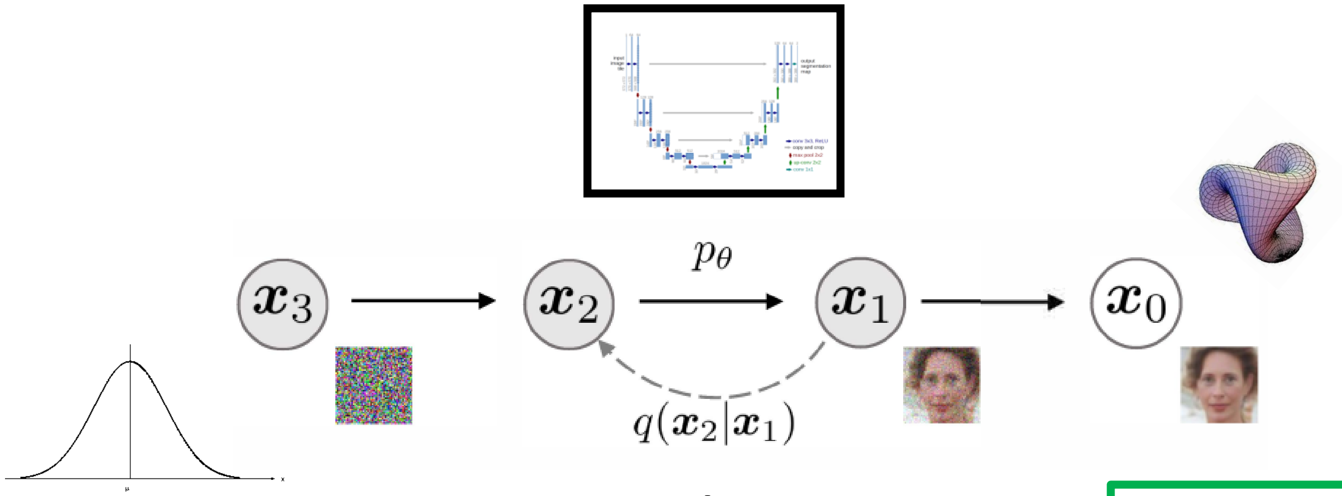


Sampling from this is complicated. What is the data distribution?



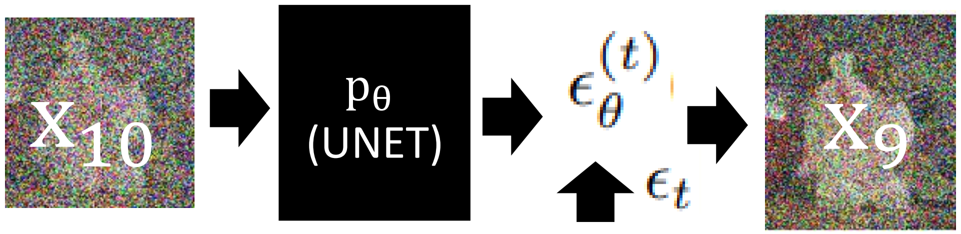
The solution





Markovian

Markovian

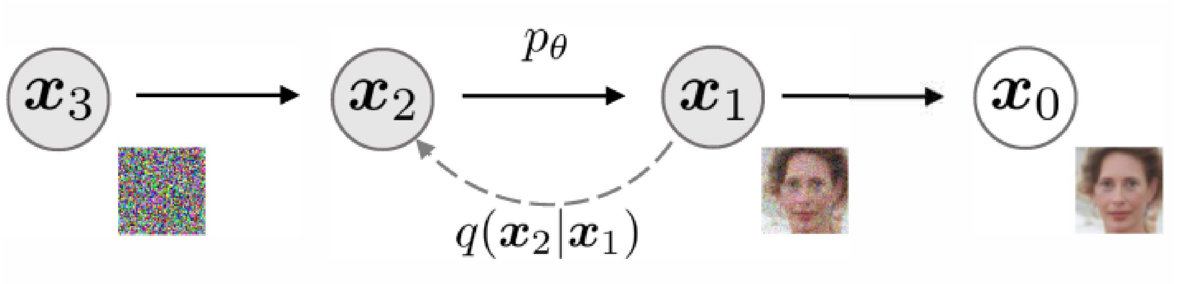


$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \epsilon_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\left\| \epsilon_\theta^{(t)}(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon_t) - \epsilon_t \right\|_2^2 \right]$$

Summary of DDPM and DDIM

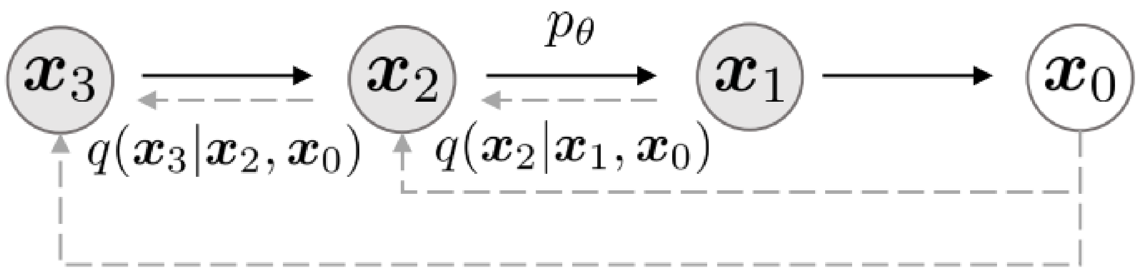
- Denoising Diffusion **Probabilistic** Models → Denoising Diffusion **Implicit** Models

DDPM



- **Difussion Process: (←)**
  - From image to noise ($x_T \leftarrow x_0$)
  - Adds some Gaussian noise to the image
  - Assume Markovian process
- **Generative Process: (→)**
  - From noise to image ($x_T \rightarrow x_0$), Markovian
  - Recovers original image
  - p_θ: A neural network
- **Sampling:** How to recoverer an image from noise?
  - Apply p_θ iteratively ($x_T \rightarrow x_{T-1} \dots \rightarrow x_1 \rightarrow x_0$)

DDIM



- **Difussion Process** (Actually this is not a difussive process but whatever)**:**
  - From image to noise ($x_0 \rightarrow x_T$)
  - Adds some Gaussian noise to the image
  - **Non-markovian process which lead to:**
    - The same training objective used in DDPM
    - A generalized sampling process (dependent on $\boldsymbol{\sigma}$)
- **Generative Process:**
  - From noise to image ($x_T \rightarrow x_0$), Non-Markovian
  - Recovers original image
  - p_θ: Neural network
- **Sampling:**
  - **Requires less iterations (when σ is properly chosen)**

$$p_\theta(\boldsymbol{x}_{0:T})$$

Generative Process

$$p_\theta(\boldsymbol{x}_0) = \int p_\theta(\boldsymbol{x}_{0:T}) \mathrm{d}\boldsymbol{x}_{1:T},$$

We need a function to optimize the parameters

Diffusion Process

Because the process is Markovian

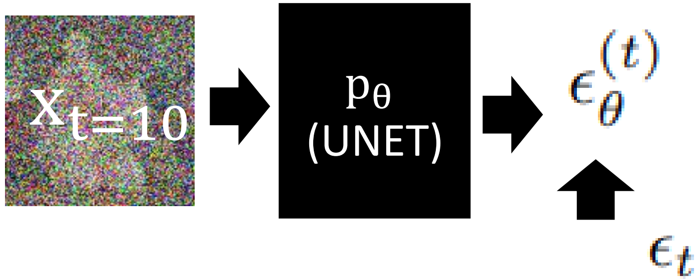
$$p_\theta(\boldsymbol{x}_{0:T}) := p_\theta(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$$

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)$$

Because is Markovian

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}),$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) := \mathcal{N}\left(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}\boldsymbol{x}_{t-1}, \left(1 - \frac{\alpha_t}{\alpha_{t-1}}\right)\boldsymbol{I}\right)$$



We need this to create data for NN training.

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) := \int q(\boldsymbol{x}_{1:t}|\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_{1:(t-1)} = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\boldsymbol{I});$$

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\boldsymbol{I});$$

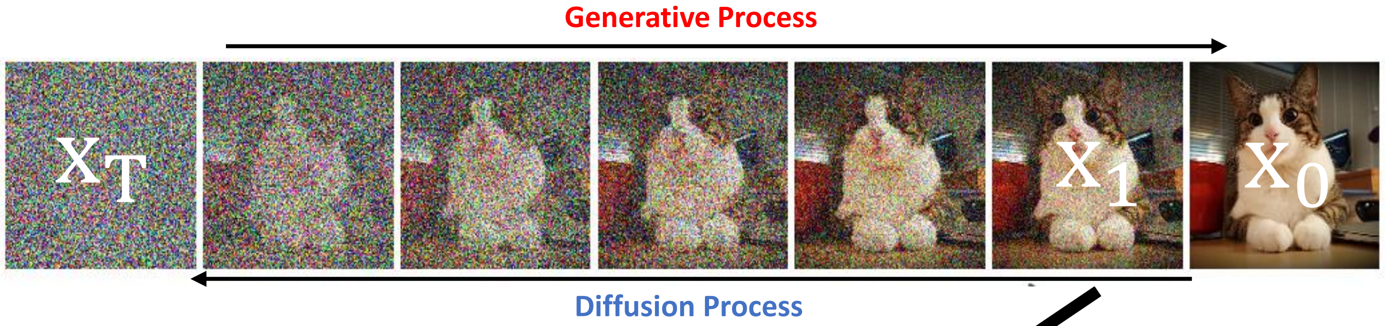
$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

Variational Lower Bound

$$\max_\theta \mathbb{E}_{q(\boldsymbol{x}_0)}[\log p_\theta(\boldsymbol{x}_0)]$$

$$\leq \max_\theta \mathbb{E}_{q(\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_T)}\left[\log p_\theta(\boldsymbol{x}_{0:T}) - \log q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)\right]$$

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{\boldsymbol{x}_0 \sim q(\boldsymbol{x}_0), \epsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})}\left[\left\|\epsilon_\theta^{(t)}(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon_t) - \epsilon_t\right\|_2^2\right]$$



This is the reverse. We want p_θ to emulate this.

$$p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$$

$$q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) := q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) \prod_{t=2}^{T} q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$$

But we do not have $\boldsymbol{x}_0$ in the generative process!! (i.e. in the sampling)

$$\boxed{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2} \cdot \frac{\boldsymbol{x}_t - \sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}}, \sigma_t^2 \boldsymbol{I}\right)}$$

$$\boxed{\begin{aligned} \boldsymbol{x}_0 &= (\boldsymbol{x}_t - \sqrt{1-\alpha_t} \cdot \epsilon_\theta^{(t)}(\boldsymbol{x}_t))/\sqrt{\alpha_t} = f_\theta^{(t)}(\boldsymbol{x}_t) \\ p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) &= q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, f_\theta^{(t)}(\boldsymbol{x}_t)) \end{aligned}}$$

The mean function is chosen to order to ensure that

$$\boxed{q_\sigma(\boldsymbol{x}_t|\boldsymbol{x}_0) = \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\boldsymbol{I})}$$

$$\boxed{J_\sigma(\epsilon_\theta) := \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})}[\log q_\sigma(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) - \log p_\theta(\boldsymbol{x}_{0:T})]}$$

$$\boxed{= \mathbb{E}_{\boldsymbol{x}_{0:T} \sim q_\sigma(\boldsymbol{x}_{0:T})}\left[q_\sigma(\boldsymbol{x}_T|\boldsymbol{x}_0) + \sum_{t=2}^{T} \log q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) - \sum_{t=1}^{T} \log p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) - \log p_\theta(\boldsymbol{x}_T)\right]}$$

$$\boldsymbol{x}_t = \sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\epsilon, \quad \text{where} \quad \epsilon \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}).$$

$$\boxed{J_\sigma = L_\gamma + C.}$$

$$\boxed{q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)=\mathcal{N}\left(\sqrt{\alpha_{t-1}}\boldsymbol{x}_0+\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{\boldsymbol{x}_t-\sqrt{\alpha_t}\boldsymbol{x}_0}{\sqrt{1-\alpha_t}},\sigma_t^2\boldsymbol{I}\right).}$$

$$p_\theta^{(t)}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)=\ q_\sigma(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t,\boldsymbol{x}_0)$$

$$\boldsymbol{x}_0=(\boldsymbol{x}_t-\sqrt{1-\alpha_t}\cdot\epsilon_\theta^{(t)}(\boldsymbol{x}_t))/\sqrt{\alpha_t}$$

$$\boldsymbol{x}_{t-1}=\qquad \text{mean} \qquad + \qquad \text{std}\times\epsilon$$

$$\boldsymbol{x}_{t-1}=\sqrt{\alpha_{t-1}}\underbrace{\left(\frac{\boldsymbol{x}_t-\sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}{\sqrt{\alpha_t}}\right)}_{\text{“ predicted } \boldsymbol{x}_0\text{”}}+\underbrace{\sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta^{(t)}(\boldsymbol{x}_t)}_{\text{“direction pointing to } \boldsymbol{x}_t\text{”}}+\underbrace{\sigma_t\epsilon_t}_{\text{random noise}}$$

This a general expresión for sampling.
We can obtain $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$ applying this iteratively.

How is this better?

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

This is Probabilistic

$\sigma_t=0$ We "eliminate" the stochasticity of the sampling so We can compute $\boldsymbol{x}_0$ directly in one step!!!

...but in practice this is not good because we depend on our **"predicted $\boldsymbol{x}_0$".**

So, we ended up using just fewer iterations, like: $10\to 8\to 6\to 4\to 2\to 0$

Thanks!

Itamar Salazar-Reque