Brief Introduction to Diffusion Models

Eric Qu

zq32@duke.edu

Class of 2023
Duke Kunshan University

June 17, 2022

Demos

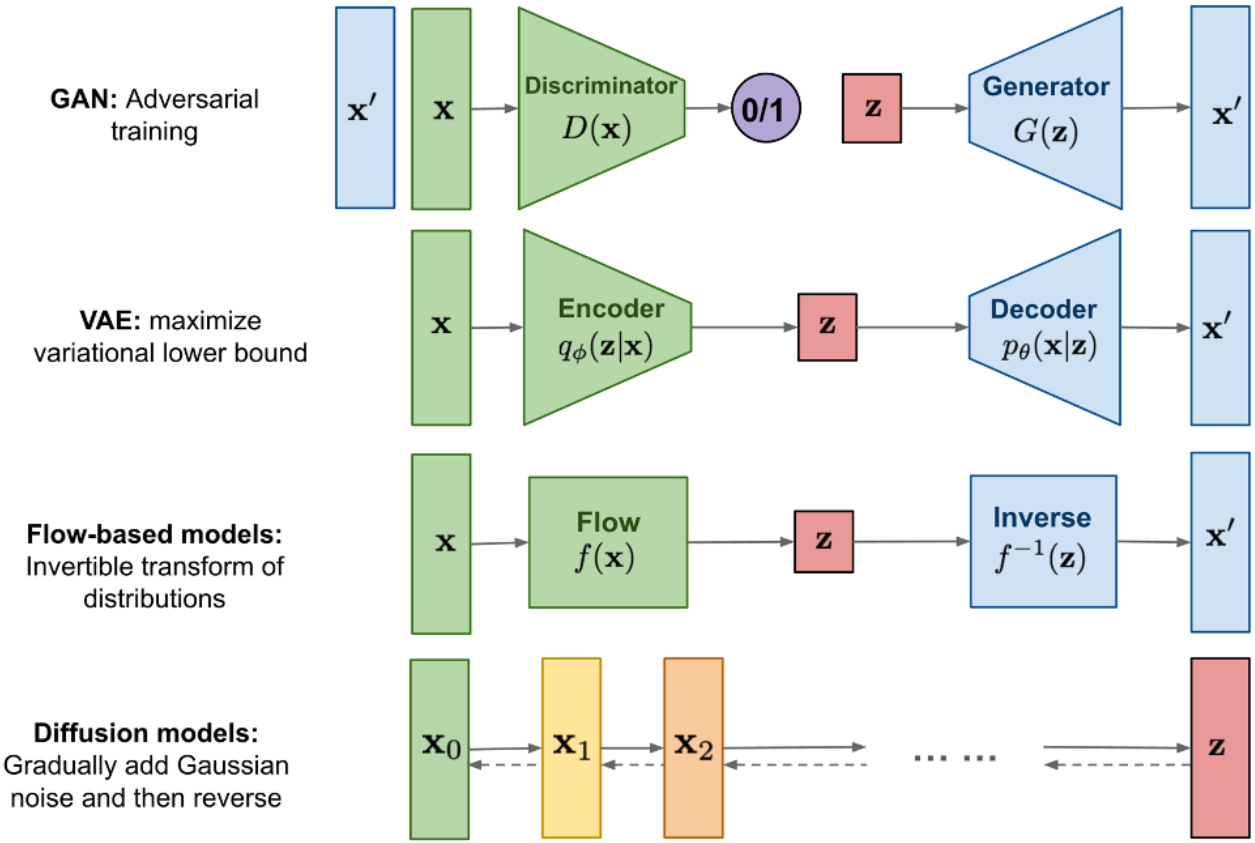
- Diffusion model is a generative model (like VAE, GAN)
- Recently, it achieved SOTA on image generation.
- Representative models: Disco Diffusion, DALLE 2

Timelines

- Deep unsupervised learning using nonequilibrium thermodynamics (Stanford) (ICML 2015)
- Denoising diffusion probabilistic models (Berkeley) (NIPS 2020)
- Denoising diffusion implicit models (Stanford) (ICLR 2021 Poster)
- Improved denoising diffusion probabilistic models (OpenAI) (ICML 2021 Poster)
- Diffusion Models Beat GANs on Image Synthesis (OpenAI) (NIPS 2021 Spotlight)
- Hierarchical Text-Conditional Image Generation with CLIP Latents (DALLE 2) (OpenAI) (2022)

Recall: Generative Models

- Generally, we want to map the data distribution to a (simple) latent distribution that we could easily sample from.



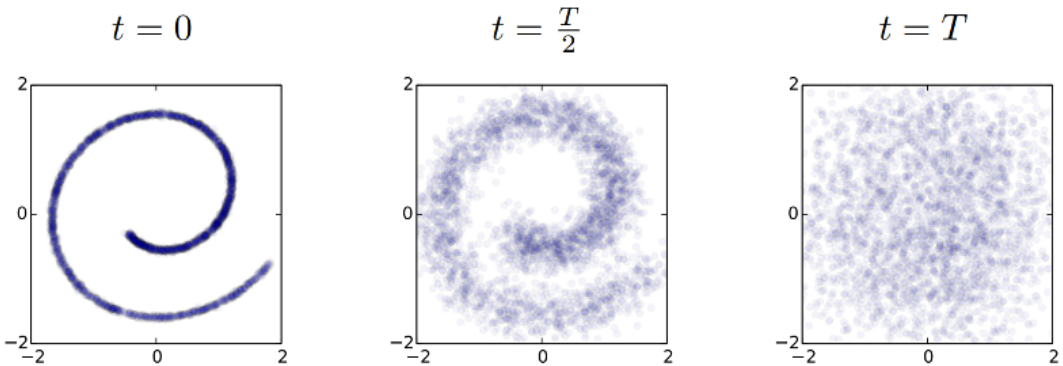
Diffusion Process

- Diffusion: inspired from nonequilibrium thermodynamics. (Jarzynski equality)
  - We define a transformation $\mathcal{T}$ that maps complex data distribution p_{complex} to a known simple prior distribution p_{prior}:

$$\mathbf{x}_0 \sim p_{\text{complex}} \Rightarrow \mathcal{T}(\mathbf{x}_0) \sim p_{\text{prior}}.$$

  - Inspired by thermodynamics, we use **Markov Chain** to model $\mathcal{T}$ by defining $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, $t \in 1, 2, \dots T$ that turns $\mathbf{x}_0$ to $\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_T$. When $T \to \infty$, we want $\mathbf{x}_T \sim p_{\text{prior}}$.

$q\left(\mathbf{x}^{(0\cdots T)}\right)$



Diffusion Process

- How to choose $q(\mathbf{x}_t|\mathbf{x}_{t-1})$?
- There are many choices, such as Gaussian or Binomial. We use Gaussian as a exmaple.

$$
\begin{aligned}
q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) &= \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right) \\
q\left(\mathbf{x}_T\right) &= p_{\text{prior}}\left(\mathbf{x}_T\right) = \mathcal{N}\left(\mathbf{x}_T; 0, \mathbf{I}\right), \quad T \to \infty
\end{aligned}
\tag{1}
$$

- Using reparametrization trick, we have

$$
\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\mathbf{z}_{t-1}, \quad \mathbf{z}_{t-1} \in \mathcal{N}(0, \mathbf{I}) \tag{2}
$$

- Here β_t controls the mixture of $\mathbf{x}_{t-1}$ and standard normal $\mathbf{z}$.
- Therefore, the diffusion process is adding Gaussian noise to the data step by step.

Diffusion Process

- Why do we have $\sqrt{1-\beta_t}$ and $\sqrt{\beta_t}$?
- Suppose $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{T} \alpha_i$,

$$
\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{\alpha_t\left(1-\alpha_{t-1}\right)}\mathbf{z}_{t-2} + \sqrt{1-\alpha_t}\mathbf{z}_{t-1} \\
&= \sqrt{\alpha_t\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\overline{\mathbf{z}}_{t-2} \\
&= \ldots \\
&= \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\mathbf{z}
\end{aligned}
\tag{3}
$$

  where we use the addition of Gaussian is still Gaussian.

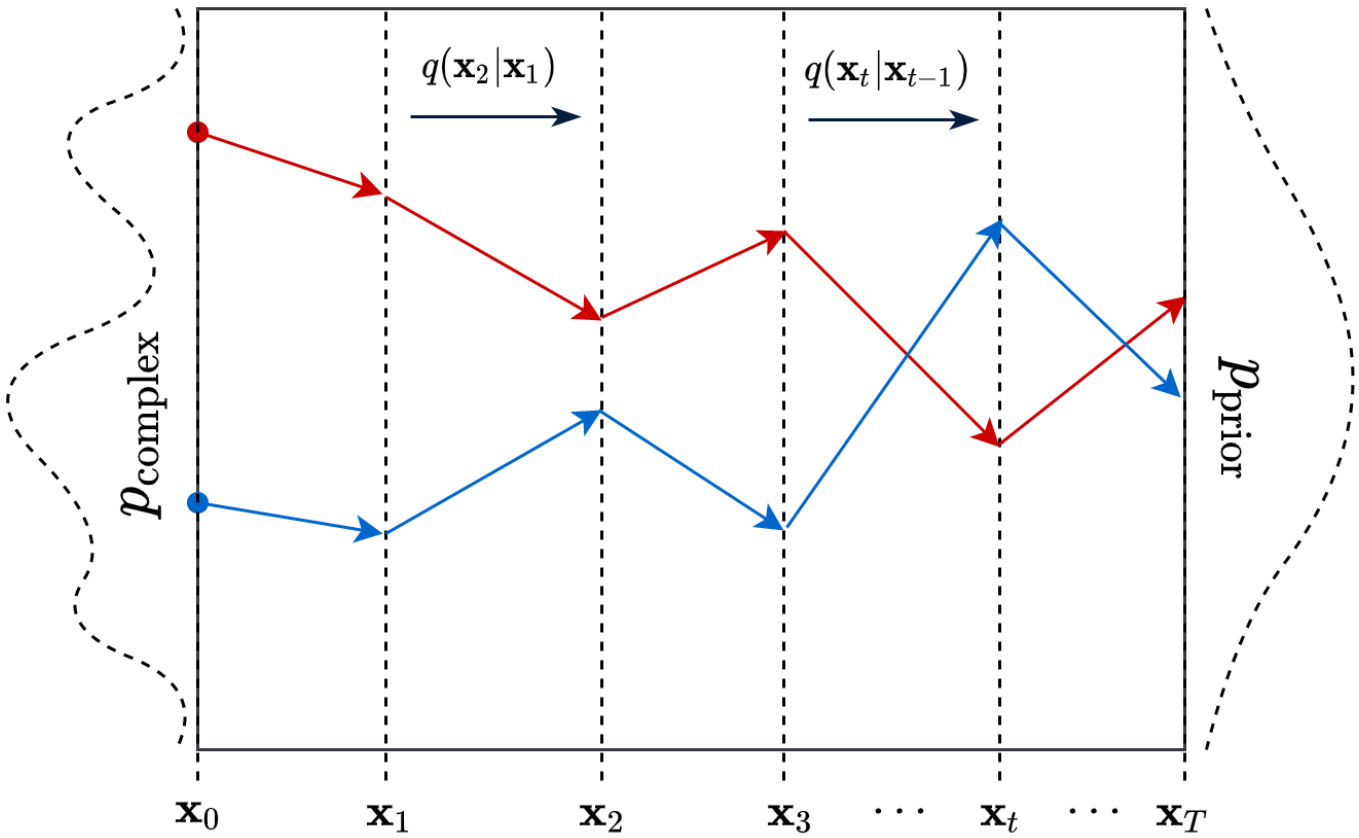
- In other words,

$$
q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right) \tag{4}
$$

- Since $\beta_t \in (0,1)$, $\alpha_t \in (0,1)$. When $t \to \infty$, $\bar{\alpha}_t \to 0$.
- Therefore, $\sqrt{1-\beta_t}$ and $\sqrt{\beta_t}$ ensures $q\left(\mathbf{x}_T\right) = \mathcal{N}(0, \mathbf{I})$

Diffusion Process

- In the paper, β_t is a interpolation between 0.0001 and 0.02 and the process has 4000 steps.
- There are no trainable parameter in the diffusion process.



Reverse Process

- If we reverse the diffusion process and sequentially sample $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$, $t \in T, T-1, \ldots, 0$, we could get $p_{\text{prior}} \rightarrow p_{\text{complex}}$
- What is the form of the reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$?
- By *Feller 1949*, the reverse of a continuous diffusion process has the same form as the forward process.
- Therefore, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is also Gaussian. But it is hard to write the distribution explicitly.
- We use neural network to approximate it!

$$\boldsymbol{p}_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right), \boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right) \tag{5}$$

Reverse Process

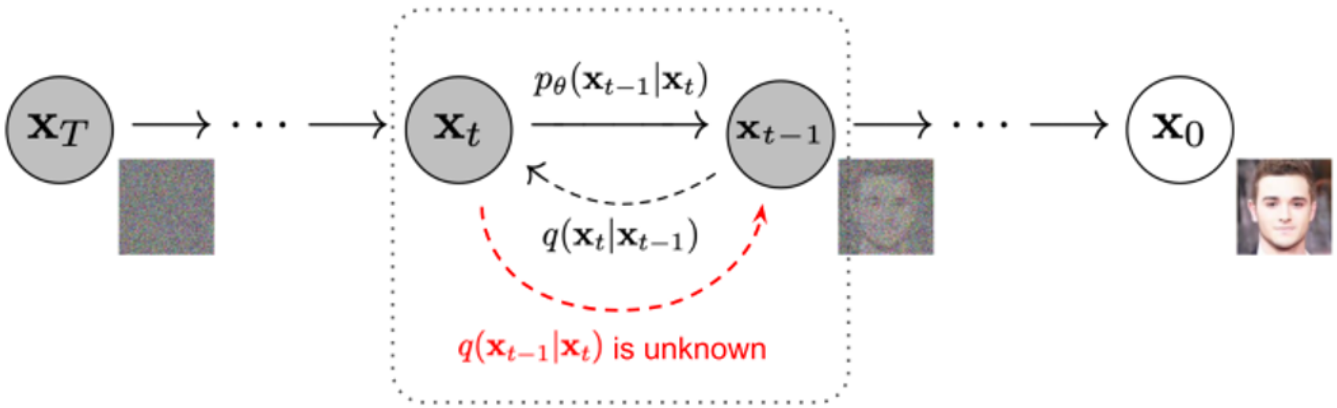
- Here, we use $p_\theta(\mathbf{x}_0)$ to approximate the data distribution,

$$p_\theta(\mathbf{x}_0) = \int p_\theta(\mathbf{x}_{0:T}) \, \mathrm{d}\mathbf{x}_{0:T} \tag{6}$$

where $p_\theta(\mathbf{x}_{0:T})$ is the joint distribution of $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T$.

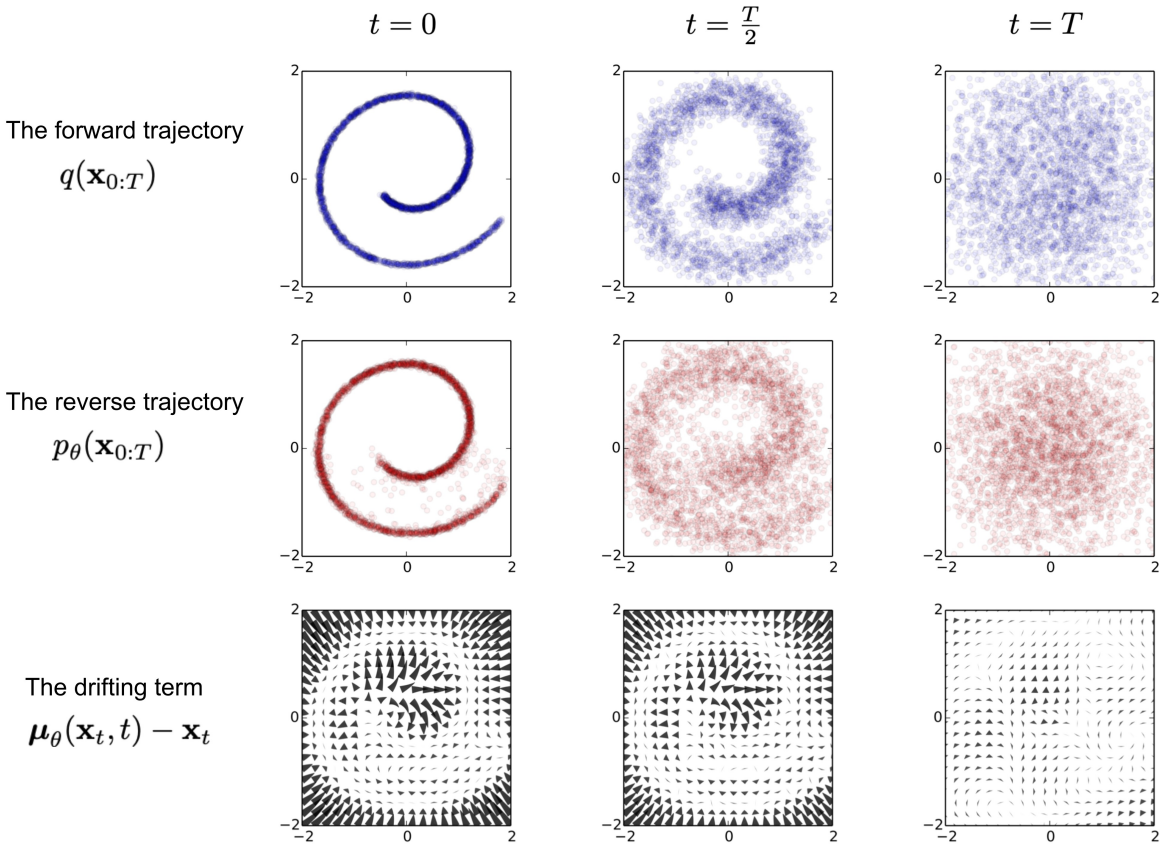
- Using Markov property and conditional probability, we have

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) \tag{7}$$



Reverse Process

- Why Gaussian could reverse the diffusion?
- Each $\mathbf{x}_t$ corresponds to a Gaussian in $t-1$. This is a special Gaussian s.t. its covariances is small $\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)=\sigma_t^2 \mathbf{I}$, $\sigma_t^2 \approx \beta_t$
- We just need a good prediction of the mean $\boldsymbol{\mu}_\theta\left(\mathbf{x}_t, t\right)$



Optimization

- To get $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$, we maximum the log likelihood

$$
\begin{aligned}
\mathcal{L} &= -\mathbb{E}_{q(\mathbf{x}_0)} \log p_\theta\left(\mathbf{x}_0\right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int p_\theta\left(\mathbf{x}_{0:T}\right) d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\int q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)} d\mathbf{x}_{1:T} \right) \\
&= -\mathbb{E}_{q(\mathbf{x}_0)} \log \left(\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)} \right) \\
&\leq -\mathbb{E}_{q(\mathbf{x}_{0:T})} \log \frac{p_\theta\left(\mathbf{x}_{0:T}\right)}{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)} \\
&= \mathbb{E}_{q(\mathbf{x}_{0:T})} \left[\log \frac{q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0:T}\right)} \right]
\end{aligned}
\tag{8}
$$

where $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})$

Optimization

- This is basically the same with VAE
- The variational lower bound is

$$
\begin{aligned}
L_{\mathrm{VLB}} &= \mathbb{E}_{q(\mathbf{x}_{0:T})}\left[\log \frac{q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{0:T}\right)}\right] \\
&= \mathbb{E}_q\left[\log \frac{\prod_{t=1}^{T} q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_T\right)\prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=1}^{T}\log \frac{q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T}\log \frac{q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)} + \log \frac{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)}\right] \\
&= \mathbb{E}_q\left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T}\log \left(\frac{q\left(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)}\frac{q\left(\mathbf{x}_t|\mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1}|\mathbf{x}_0\right)}\right) + \log \frac{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)}\right]
\end{aligned}
\tag{9}
$$

Optimization

- The variational lower bound is

$$
\begin{aligned}
&= \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \left(\frac{q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)} \frac{q\left(\mathbf{x}_t|\mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1}|\mathbf{x}_0\right)} \right) + \log \frac{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)} + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_t|\mathbf{x}_0\right)}{q\left(\mathbf{x}_{t-1}|\mathbf{x}_0\right)} + \log \frac{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta\left(\mathbf{x}_T\right) + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)} + \log \frac{q\left(\mathbf{x}_T|\mathbf{x}_0\right)}{q\left(\mathbf{x}_1|\mathbf{x}_0\right)} + \log \frac{q\left(\mathbf{x}_1|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)} \right] \\
&= \mathbb{E}_q \left[\log \frac{q\left(\mathbf{x}_T|\mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_T\right)} + \sum_{t=2}^{T} \log \frac{q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0\right)}{p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)} - \log p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right) \right] \\
&= \mathbb{E}_q [\underbrace{-\log p_\theta\left(\mathbf{x}_0|\mathbf{x}_1\right)}_{L_0}] + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)\right)}_{L_{t-1}} \\
&\quad + \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T|\mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_T\right)\right)}_{L_T}
\end{aligned}
$$

Optimization

- Here, the variational lower bound has three terms

1. Entropy L_0: $\mathbb{E}_q[-\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]$

- This is turning the last hidden layer to output.
- We know $p_\theta(\mathbf{x}_0|\mathbf{x}_1) = \mathcal{N}\left(\boldsymbol{\mu}_\theta(\mathbf{x}_1, 1), \sigma^2\mathbf{I}\right)$
- In principle, since the entropy of a multivariate Gaussian only depends on its covariance, this term is constant.
- However, we often need to discretize the output. This depends on the type of data. For images, we take an integral.

2. KL divergence L_{t-1}: $D_{\mathrm{KL}}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))$

- More on this Later

3. KL divergence L_T: $D_{\mathrm{KL}}(q(\mathbf{x}_T|\mathbf{x}_0) \| p_\theta(\mathbf{x}_T))$

- Since $\mathbf{x}_T$ & $\mathbf{x}_0$ are sampled from fixed dist., this is constant.

$D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0\right)\|p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)\right)$

- This is the difference between real reverse distribution $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0\right)$ and the trained distribution $p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right)$
- We know that $\boldsymbol{p}_\theta\left(\mathbf{x}_{t-1}\mid\mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1};\boldsymbol{\mu}_\theta\left(\mathbf{x}_t,t\right),\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t,t\right)\right)$
- We can beak $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0\right)$ using Bayes Theorem

$$q\left(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_0\right)=q\left(\mathbf{x}_t\mid\mathbf{x}_{t-1},\mathbf{x}_0\right)\frac{q\left(\mathbf{x}_{t-1}\mid\mathbf{x}_0\right)}{q\left(\mathbf{x}_t\mid\mathbf{x}_0\right)}=q\left(\mathbf{x}_t\mid\mathbf{x}_{t-1}\right)\frac{q\left(\mathbf{x}_{t-1}\mid\mathbf{x}_0\right)}{q\left(\mathbf{x}_t\mid\mathbf{x}_0\right)}$$

$$\propto\exp\left(-\frac{1}{2}\left(\frac{\left(\mathbf{x}_t-\sqrt{\alpha_t}\mathbf{x}_{t-1}\right)^2}{\beta_t}+\frac{\left(\mathbf{x}_{t-1}-\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0\right)^2}{1-\bar{\alpha}_{t-1}}-\frac{\left(\mathbf{x}_t-\sqrt{\bar{\alpha}_t}\mathbf{x}_0\right)^2}{1-\bar{\alpha}_t}\right)\right)$$

$$=\exp\left(-\frac{1}{2}\left(\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)\mathbf{x}_{t-1}^2-\left(\frac{2\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{2\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)\mathbf{x}_{t-1}+C\left(\mathbf{x}_t,\mathbf{x}_0\right)\right)\right) \tag{11}$$

- Therefore, $q\left(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0\right)$ is also a Gaussian, with the form

$$q\left(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_0\right)=\mathcal{N}\left(\mathbf{x}_{t-1};\tilde{\boldsymbol{\mu}}\left(\mathbf{x}_t,\mathbf{x}_0\right),\tilde{\beta}_t\mathbf{I}\right) \tag{12}$$

- Thus, we have

$$\tilde{\beta}_t = 1/\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right)=\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\cdot\beta_t$$

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\mathbf{x}_0\right) &= \left(\frac{\sqrt{\alpha_t}}{\beta_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_{t-1}}\mathbf{x}_0\right)/\left(\frac{\alpha_t}{\beta_t}+\frac{1}{1-\bar{\alpha}_{t-1}}\right) \\ &= \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0\end{aligned} \tag{13}$$

- With the explicit form, we can directly write the KL divergence

$$L_t = \mathbb{E}_q\left[\frac{1}{2\left\|\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t,t\right)\right\|_2^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\mathbf{x}_0\right)-\boldsymbol{\mu}_\theta\left(\mathbf{x}_t,t\right)\right\|^2\right]+C \tag{14}$$

- In practice, we set $\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t,t\right)=\sigma_t^2\mathbf{I}$ and let $\sigma_t^2=\beta_t$ or $\tilde{\beta}_t$
- It seems that we only need to train $\boldsymbol{\mu}_\theta\left(\mathbf{x}_t,t\right)$ with L2 norm.

- Can we go further?
- We could try to cancel out the $\mathbf{x}_0$ input of $\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\mathbf{x}_0\right)$
- Recall that from (3): $\mathbf{x}_t=\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\mathbf{z}$

  We could get $\mathbf{x}_0=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t\right)$
- Plug it in and we have

$$\begin{aligned}\tilde{\boldsymbol{\mu}}_t&=\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t+\frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t\right)\\&=\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right)\end{aligned}$$

- Further plug that in and we have

$$L_t - C = \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\boldsymbol{\mu}}_t\left(\mathbf{x}_t,\mathbf{x}_0\right)-\boldsymbol{\mu}_\theta\left(\mathbf{x}_t,t\right)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_t\right)-\frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t-\frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta\left(\mathbf{x}_t,t\right)\right)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{\beta_t^2}{2\alpha_t\left(1-\bar{\alpha}_t\right)\sigma_t^2}\left\|\mathbf{z}_t-\mathbf{z}_\theta\left(\mathbf{x}_t,t\right)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{x}_0,\mathbf{z}_t}\left[\frac{\beta_t^2}{2\alpha_t\left(1-\bar{\alpha}_t\right)\sigma_t^2}\left\|\mathbf{z}_t-\mathbf{z}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\mathbf{z}_t,t\right)\right\|^2\right] \tag{15}$$

- In pratice, we remove the scaling coefficient and use this simplified lower bound

$$L_{\text{simple}}\left(\theta\right) := \mathbb{E}_{t,\mathbf{x}_0,\epsilon_t}\left[\left\|\epsilon_t-\mathbf{z}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0+\sqrt{1-\bar{\alpha}_t}\epsilon_t,t\right)\right\|^2\right] \tag{16}$$

Summary

- Here is the training and sampling process.
- In training, we are learning the noise.
- In sampling, we need to go layer by layer.
- This is really slow.

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

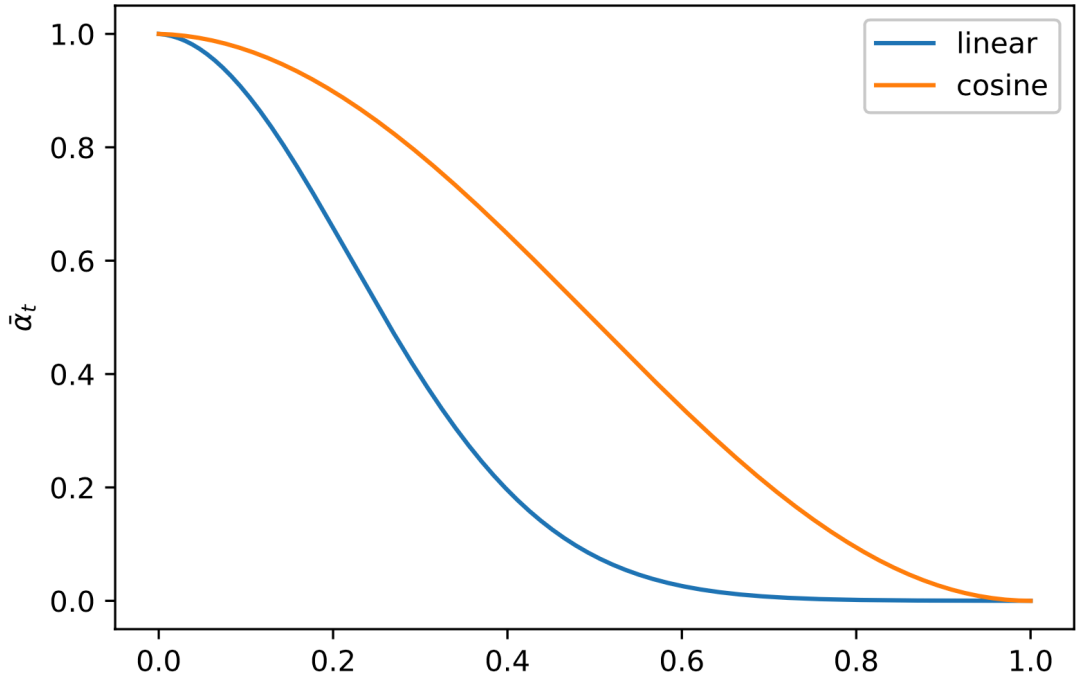
Algorithm 2 Sampling

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** $\mathbf{x}_0$

Improved DDPM

- Nichol & Dhariwal (2021) proposed a better way to schedule variance β_t
- It uses a cosine-based function

$$\beta_t = \text{clip}\left(1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}, 0.999\right) \quad \bar{\alpha}_t = \frac{f(t)}{f(0)} \quad f(t) = \cos\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right) \tag{17}$$



Improved DDPM

- Nichol & Dhariwal (2021) also to learn $\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)$ instead of setting it to $\sigma_t^2 \mathbf{I}$
- However, L_{simple} does not contain $\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right)$. They had to use $L = L_{\text{simple}} + \lambda L_{\text{VLB}}$, $\lambda = 0.001$.

$$\boldsymbol{\Sigma}_\theta\left(\mathbf{x}_t, t\right) = \exp\left(\mathbf{v} \log \beta_t + (1 - \mathbf{v}) \log \tilde{\beta}_t\right) \tag{18}$$

Model	ImageNet	CIFAR
Glow (Kingma & Dhariwal, 2018)	3.81	3.35
Flow++ (Ho et al., 2019)	3.69	3.08
PixelCNN (van den Oord et al., 2016c)	3.57	3.14
SPN (Menick & Kalchbrenner, 2018)	3.52	-
NVAE (Vahdat & Kautz, 2020)	-	2.91
Very Deep VAE (Child, 2020)	3.52	2.87
PixelSNAIL (Chen et al., 2018)	3.52	2.85
Image Transformer (Parmar et al., 2018)	3.48	2.90
Sparse Transformer (Child et al., 2019)	3.44	**2.80**
Routing Transformer (Roy et al., 2020)	**3.43**	-
DDPM (Ho et al., 2020)	3.77	3.70
DDPM (cont flow) (Song et al., 2020b)	-	2.99
Improved DDPM (ours)	**3.53**	**2.94**

Conditional DDPM

- Dhariwal & Nichol (2021) (beats GAN)
- Train a classifier $f_\phi\left(y \mid \mathbf{x}_t, t\right)$ on noisy image $\mathbf{x}_t$
- Use gradients $\nabla_{\mathbf{x}} \log f_\phi\left(y \mid \mathbf{x}_t, t\right)$ to guide sampling process

Algorithm 1 Classifier guided diffusion sampling, given a diffusion model $(\mu_\theta(x_t), \Sigma_\theta(x_t))$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\mu, \Sigma \leftarrow \mu_\theta(x_t), \Sigma_\theta(x_t)$
  $x_{t-1} \leftarrow$ sample from $\mathcal{N}(\mu + s\Sigma \nabla_{x_t} \log f_\phi(y|x_t), \Sigma)$
end for
return x_0

Algorithm 2 Classifier guided DDIM sampling, given a diffusion model $\epsilon_\theta(x_t)$, classifier $f_\phi(y|x_t)$, and gradient scale s.

Input: class label y, gradient scale s
$x_T \leftarrow$ sample from $\mathcal{N}(0, \mathbf{I})$
for all t from T to 1 **do**
  $\hat{\epsilon} \leftarrow \epsilon_\theta(x_t) - \sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log f_\phi(y|x_t)$
  $x_{t-1} \leftarrow \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\hat{\epsilon}}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}$
end for
return x_0

References

[1] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, pp. 2256–2265, PMLR, 2015.

[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[3] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2020.

[4] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *International Conference on Machine Learning*, pp. 8162–8171, PMLR, 2021.

[5] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[6] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.