DIFFUSION MODELS: MATH AND DERIVATIONS

Wenhan Gao
Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794, USA
wenhan.gao@stonybrook.edu

1 Denoising Diffusion Probabilistic Models (DDPM)

1.1 Forward Process

The *forward process* or *diffusion process* is a Markov chain that gradually adds Gaussian noise to the data according to:

$$q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) := \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \left(1-\alpha_t\right)\mathbf{I}\right),$$

where $\alpha_t := 1 - \beta_t$, $\beta_t \in (0,1)$ is the noise schedule with $\beta_0 = 0$ and $\beta_1 = 1$.

• To sample, $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

As a Markov chain, we can also write

$$q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, \left(1-\bar{\alpha}_t\right)\mathbf{I}\right) \text{ with } \bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s.$$

• To sample, $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Proof. By induction,

$$\begin{aligned}
\mathbf{x}_t &= \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon_t \\
&= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\mathbf{x}_{t-2} + \sqrt{1-\alpha_{t-1}}\epsilon_{t-1}\right) + \sqrt{1-\alpha_t}\epsilon_t \\
&= \underbrace{\sqrt{\alpha_{t-1}\alpha_t}}_{\mu = \sqrt{\prod_{s=t-1}^{t}\alpha_s}}\mathbf{x}_{t-2} + \underbrace{\sqrt{\alpha_t\left(1-\alpha_{t-1}\right)}\epsilon_{t-1} + \sqrt{1-\alpha_t}\epsilon_t}_{\sigma^2 = 1 - \prod_{s=t-1}^{t}\alpha_s}.
\end{aligned}$$

□

1.2 Reverse Process

The joint distribution $p_\theta\left(\mathbf{x}_{0:T}\right)$ is called the *reverse process*, and it is defined as a Markov chain with learned Gaussian transitions starting at $p\left(\mathbf{x}_T\right) = \mathcal{N}\left(\mathbf{x}_T; \mathbf{0}, \mathbf{I}\right)$, where

$$p_\theta\left(\mathbf{x}_{0:T}\right) := p\left(\mathbf{x}_T\right)\prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right), \quad p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right) := \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \mathbf{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right).$$

1.3 Optimization Objective: ELBO

We aim to maximize the log-likelihood $\log p_\theta\left(\mathbf{x}_0\right)$, which can be reformulated as maximizing the variational lower bound:

$$\underbrace{\mathbb{E}_{q(\mathbf{x}_1 \mid \mathbf{x}_0)}\left[\log p_\theta\left(\mathbf{x}_0 \mid \mathbf{x}_1\right)\right]}_{L_0\text{: reconstruction term}} - \underbrace{D_{\mathrm{KL}}\left(q\left(\mathbf{x}_T \mid \mathbf{x}_0\right) \| p\left(\mathbf{x}_T\right)\right)}_{L_T\text{: prior matching term}} - \underbrace{\sum_{t=2}^{T}\mathbb{E}_{q(\mathbf{x}_t \mid \mathbf{x}_0)}\left[D_{\mathrm{KL}}\left(q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \| p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)\right)\right]}_{L_{t-1}\text{: denoising matching terms}}. \tag{1.1}$$

• The posterior transition distribution $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right), \tilde{\beta}_t\mathbf{I}\right)$ with

$$\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right) := \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t}\mathbf{x}_t \text{ and } \tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{1.2}$$

Proof of the Variational Lower Bound. This proof is adopted from Luo (2022) with added details.

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

□

Proof of the Posterior Transition Distribution. Note that

$$
q(\mathbf{x}_{t-1}|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, \left(1-\bar{\alpha}_{t-1}\right)\mathbf{I}\right),
$$

and

$$
q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)\mathbf{I}\right)
$$

$$
\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})
$$

$$
\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\epsilon \quad \Rightarrow \quad \mathbf{x}_{t-1} \sim \mathcal{N}\left(\frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t, \frac{1-\alpha_t}{\alpha_t}\mathbf{I}\right).
$$

Note that this is just a reparameterization; it does not define $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. Then,

$$
q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) \propto \underbrace{\mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, \left(1-\bar{\alpha}_{t-1}\right)\mathbf{I}\right)}_{\text{Prior prediction from } \mathbf{x}_0} \cdot \underbrace{\mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_t, \frac{1-\alpha_t}{\alpha_t}\mathbf{I}\right)}_{\text{Posterior estimate from } \mathbf{x}_t},
$$

which can be solved by the product of Gaussians formula:

$$
\mathcal{N}\left(x; \mu_1, \Sigma_1\right) \cdot \mathcal{N}\left(x; \mu_2, \Sigma_2\right) \propto \mathcal{N}(x; \mu, \Sigma), \text{ with } \Sigma = \left(\Sigma_1^{-1} + \Sigma_2^{-1}\right)^{-1} \text{ and } \mu = \Sigma\left(\Sigma_1^{-1}\mu_1 + \Sigma_2^{-1}\mu_2\right).
$$

Remark 1.1. *At a high level, $\mathbf{x}_0$ and $\mathbf{x}_t$ define a trajectory, and our goal is to determine the probability of $\mathbf{x}_{t-1}$ along this path. Intuitively, whether we move forward from $\mathbf{x}_0$ to $\mathbf{x}_{t-1}$ or backward from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$, we should arrive at the same point $\mathbf{x}_{t-1}$. Therefore, this probability can be expressed as the product of two distributions, one conditioned on $\mathbf{x}_0$ and the other on $\mathbf{x}_t$.*

On page 12 of Luo (2022), a direct proof is provided. We provide an alternative proof, as our proof might give a better intuitive idea in the remark above. Moreover, our proof is actually equivalent to Luo (2022). To see this,

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

which is also a product of two Gaussians; however, as the unknown is $\mathbf{x}_{t-1}$. We apply a simple reparameterization

$$\begin{aligned}
q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)=\mathcal{N}\left(\mathbf{x}_t ; \sqrt{\alpha_t} \mathbf{x}_{t-1},\left(1-\alpha_t\right) \mathbf{I}\right) & \propto \exp \left\{-\left[\frac{\left(\mathbf{x}_t-\sqrt{\alpha_t} \mathbf{x}_{t-1}\right)^2}{2\left(1-\alpha_t\right)}\right]\right\} \\
& =\exp \left\{-\left[\frac{\mathbf{x}_t^2-2 \sqrt{\alpha_t} \mathbf{x}_t \mathbf{x}_{t-1}+\alpha_t \mathbf{x}_{t-1}^2}{2\left(1-\alpha_t\right)}\right]\right\} \\
& =\exp \left\{-\left[\frac{\mathbf{x}_{t-1}^2-2 \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t \mathbf{x}_{t-1}+\frac{1}{\alpha_t} \mathbf{x}_t^2}{2\left(\frac{1-\alpha_t}{\alpha_t}\right)}\right]\right\} \\
& =\exp \left\{-\left[\frac{\left(\mathbf{x}_{t-1}-\frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t\right)^2}{2\left(\frac{1-\alpha_t}{\alpha_t}\right)}\right]\right\} \propto \mathcal{N}\left(\mathbf{x}_{t-1} ; \frac{1}{\sqrt{\alpha_t}} \mathbf{x}_t, \frac{1-\alpha_t}{\alpha_t} \mathbf{I}\right),
\end{aligned}$$

which establishes the equivalence between Luo (2022) and our proof. □

1.4 Denoising Loss

We aim to learn $p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t\right)=\mathcal{N}\left(\mathbf{x}_{t-1} ; \mu_\theta\left(\mathbf{x}_t, t\right), \mathbf{\Sigma}_\theta\left(\mathbf{x}_t, t\right)\right)$ that minimizes the KL divergence:

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

In general, set $\mathbf{\Sigma}_\theta\left(\mathbf{x}_t, t\right)=\sigma_t^2$ for some variance σ_t^2, then from the KL divergence of two Gaussians (see Appendix A.2), we have

$$L_{t-1}=\mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t}\left[\frac{1}{2 \sigma_t^2}\left\|\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)-\mu_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right]+C, \ C \text{ independent of } \theta.$$

From (1.2), we have the exact form of the variance, $\sigma_t^2=\tilde{\beta}_t$, and $C=0$.

As in the reverse process, we have $\mathbf{x}_t$, it seems that we should learn to predict $\mathbf{x}_0$. We will show that predicting $\mathbf{x}_0$ is equivalent to predicting noise. Note that

$$\tilde{\mu}_t\left(\mathbf{x}_t, \mathbf{x}_0\right)=\frac{\sqrt{\bar{\alpha}_{t-1}} \beta_t}{1-\bar{\alpha}_t} \mathbf{x}_0+\frac{\sqrt{\alpha_t}\left(1-\bar{\alpha}_{t-1}\right)}{1-\bar{\alpha}_t} \mathbf{x}_t.$$

Since in the reverse process, we do not have $\mathbf{x}_0$, we want to express $\mathbf{x}_0$ with $\mathbf{x}_t$.

$$\mathbf{x}_t=\sqrt{\bar{\alpha}_t} \mathbf{x}_0+\sqrt{1-\bar{\alpha}_t} \epsilon, \quad \text{for some } \epsilon \sim \mathcal{N}(0, \mathbf{I}),$$

then

$$\mathbf{x}_0=\frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}}-\frac{\sqrt{1-\bar{\alpha}_t} \epsilon}{\sqrt{\bar{\alpha}_t}}=\frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t-\sqrt{1-\bar{\alpha}_t} \cdot \epsilon\right).$$

Now,

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

Since $\mathbf{x}_t$ is known, we let

$$\mu_\theta\left(\mathbf{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta\left(\mathbf{x}_t, t\right)\right),$$

and the denoising matching terms becomes

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t\left(1-\bar{\alpha}_t\right)}\left\|\epsilon - \epsilon_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right], \text{ where } \mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon. \tag{1.3}$$

Note that the reconstruction term, which is the log-probability under a Gaussian distribution, can also be expressed as a MSE (see Appendix A.2).

1.5 Signal-to-Noise Ratio and Loss Weighting

Following Kingma et al. (2023), we define the signal-to-noise ratio (SNR) as

$$\mathrm{SNR}(t) = \frac{\bar{\alpha}_t}{\bar{\beta}_t}.$$

Since $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{\bar{\beta}_t}\epsilon$, as the name implies, SNR represents the ratio between the original signal and the noise. SNR decreases over time, confirming the notion that the input becomes increasingly noisy over time. Note that SNR determines both $\bar{\alpha}_t$ and $\bar{\beta}_t$ as $\bar{\alpha}_t^2 = \bar{\beta}_t^2$.

Eq. (1.3) simplifies to

$$L_{t-1} = \mathbb{E}_{\mathbf{x}_0,\epsilon}\left[\frac{1}{2}\cdot\left(\frac{\mathrm{SNR}(t-1)}{\mathrm{SNR}(t)} - 1\right)\left\|\epsilon - \epsilon_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right].$$

In practice, uniform sampling over time is taken instead of summing all the denoising matching terms. So more generally, the loss can be expressed as a weighted MSE

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[w(t)\left\|\epsilon - \epsilon_\theta\left(\mathbf{x}_t, t\right)\right\|^2\right].$$

this weighting term is often applied to ensure that certain timesteps (and thus noise levels) aren't over- or under-emphasized during training. In DDPM, $w(t) = 1,\ \forall t$.

1.6 Training and Sampling

The training and sampling algorithms provided in DDPM (Ho et al., 2020) is provided below. In summary, the sampling process aims to approximate the mean at time $t-1$ by x_t (given) and x_0 (approximated, i.e., denoised by the neural network).

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** $\mathbf{x}_0$

Common confusion about DDPM:

1. Why do we need so many steps in the reverse process? It seems that in every step we have already predicted $\mathbf{x}_0$ and use it to sample $\mathbf{x}_{t-1}$.
   - You do not need $\hat{\mathbf{x}}_0$ to be perfect in each step, just good enough to take a small step in the right direction. In early denoising steps, $\hat{\mathbf{x}}_0$ is very noisy and the update is small. In later steps, x_t becomes less noisy, so $\hat{\mathbf{x}}_0$ becomes more accurate. Think of $\hat{\mathbf{x}}_0$ as a noisy estimate of a target, and you are doing one update step toward it each time, just like the gradient descent.

1.7 Relevance to Score Matching

In probability, the *score function* is $\nabla_x \log q(x)$; it indicates the direction in which the probability density $q(x)$ increases the most. Instead of learning the full probability $q(x)$, *score matching* learns the score function to move in the direction of higher data likelihood, at every noise level.

DDPM relates to score matching simply by Tweedie's formula, which states that for a Gaussian random variable $\mathbf{z} \sim \mathcal{N}(\mathbf{z}; \mu_z, \mathbf{\Sigma}_z)$,

$$\mathbb{E}[\mu_z \mid \mathbf{z}] = \mathbf{z} + \mathbf{\Sigma}_z \nabla_{\mathbf{z}} \log q(\mathbf{z}).$$

Consider,

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

by Tweedie's formula:

$$\mathbb{E}[\mu_{x_t} \mid \mathbf{x}_t] = \mathbf{x}_t + (1-\bar{\alpha}_t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = \sqrt{\bar{\alpha}_t}\mathbf{x}_0.$$

Rearrange terms:

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 - (1-\bar{\alpha}_t)\nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t \mid \mathbf{x}_0).$$

We also know that

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

It is easy to see the connection between the score function and the noise: $\nabla \log q_t(\mathbf{x}_t \mid \mathbf{x}_0) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon$. Thus, denoising is equivalent to score matching, up to a constant factor dependent on time. Score-based models might offer some insight into diffusion models; the readers are encouraged to read Sec. 3.

Since the Gaussian under consideration is isotropic, the derivation is straightforward. To help understanding of Tweedie's formula, we present an alternative derivation that does not rely on it directly.

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0, (1-\bar{\alpha}_t)\,\mathbf{I}\right) = \frac{1}{(2\pi)^{d/2}(1-\bar{\alpha}_t)^{d/2}} \exp\left(-\frac{1}{2(1-\bar{\alpha}_t)}\left\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0\right\|^2\right),$$

then

$$\begin{aligned}
\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0) &= \nabla_{\mathbf{x}_t}\left(-\frac{d}{2}\log(2\pi) - \frac{d}{2}\log(1-\bar{\alpha}_t) - \frac{1}{2(1-\bar{\alpha}_t)}\left\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0\right\|^2\right) \\
&= -\frac{1}{2(1-\bar{\alpha}_t)}\nabla_{\mathbf{x}_t}\left\|\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0\right\|^2
\end{aligned}$$

$$
\begin{aligned}
&= -\frac{1}{2(1-\bar{\alpha}_t)} \cdot 2\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0\right) \\
&= -\frac{1}{1-\bar{\alpha}_t}\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0\right) \\
&= -\frac{1}{1-\bar{\alpha}_t}\left(\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}\right) \\
&= -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}.
\end{aligned}
$$

2 Denoising Diffusion Implicit Models (DDIM)

2.1 Accelerated Sampling

During training, we never use the Markov transition probability $p\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$; instead, we rely solely on the marginal[1] distributions $p\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$. As a result, the training of DDPM inherently includes the training signals for any subsequence of time steps: $\Delta t, 2\Delta t, 3\Delta t, \ldots, T$, for any $\Delta t \geq 1$ that divides T.

From Eq. (1.2), we can interpret the sampling process as an interpolation between the estimated clean data $\hat{\mathbf{x}}_0$ and the noisy input $\mathbf{x}_t$. This perspective allows us skip intermediate steps during sampling if the trajectory is smooth. Thus, we want to reduce randomness in the sampling process as randomness injects high-frequency variation.

2.2 DDIM Sampling

We begin by presenting the DDIM (Song et al., 2020) deterministic sampling algorithm to provide a high-level overview of its mechanism. We then discuss the mathematical insights and justifications.

Algorithm 3 DDIM Sampling

$\Delta t = \frac{T}{N_{\text{steps}}}$
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, T-\Delta t, T-2\Delta t, \ldots, \Delta t$ **do**
  $\hat{\mathbf{x}}_0 = \frac{\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\cdot\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\bar{\alpha}_t}}$
  $\mathbf{x}_{t-\Delta t} = \sqrt{\bar{\alpha}_{t-\Delta t}} \cdot \hat{\mathbf{x}}_0 + \sqrt{1-\bar{\alpha}_{t-\Delta t}} \cdot \epsilon_\theta(\mathbf{x}_t, t)$
end for
return $\mathbf{x}_0$

Recall $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. DDIM sampling works in two steps as a linear interpolation between $\hat{\mathbf{x}}_0$ and $\mathbf{x}_t$:

1. Predict $\hat{\mathbf{x}}_0$ from the noisy input $\mathbf{x}_t$
2. Move toward $\mathbf{x}_{t-\Delta t}$ deterministically using a formula that guides the sample along a trajectory derived from $\mathbf{x}_0$ and $\mathbf{x}_t$.

Unlike DDPM, DDIM sampling does not add new noise in each step, making the process deterministic and more smooth.

2.3 Non-Markovian Forward Processes

The posterior transition distribution, i.e., Eq. (1.2), derived in DDPM follows a Gaussian distribution with variance $\tilde{\beta}_t\mathbf{I}$. However, the above sampling is deterministic (no variance); we now dive into more mathematical insights and justifications as to why we can do this.

Recall that in DDPM, the posterior transition distribution is derived with the one-step forward transition: $q\left(\mathbf{x}_t \mid \mathbf{x}_{t-1}\right)$, which defines the joint distribution $q\left(\mathbf{x}_{1:T} \mid \mathbf{x}_0\right)$ given the intial state $q\left(\mathbf{x}_0\right)$. Although DDPM is defined as a Markovian process, we never use the joint distribution during training. The DDPM objective depends only on the marginals $q\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$. There are many joint distributions that lead to the same marginals. Using only marginals, we can derive an alternative posterior transition distribution, $q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right)$.

[1] Here, $p\left(\mathbf{x}_t \mid \mathbf{x}_0\right)$ is referred to as the marginal distribution because we do not track the intermediate steps. We marginalize the intermediate steps out. The marginal here means marginal over path. Depending on the context, $p(\mathbf{x}_t)$ could also be called marginal, which is marginalizing over initial states.

Without the Markovian assumption, this distribution can be more flexible; it only needs to satisfy:

$$\int q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) q\left(\mathbf{x}_t \mid \mathbf{x}_0\right) d\mathbf{x}_t = q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0\right). \tag{2.1}$$

Generally, let

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \zeta_t \mathbf{x}_t + \xi_t \mathbf{x}_0, \sigma_t^2 \mathbf{I}\right).$$

Then, we have

	Distribution	**Sampling**
$q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)$	$\mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right)$	$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\boldsymbol{\epsilon}_1$
$q(\mathbf{x}_t \mid \mathbf{x}_0)$	$\mathcal{N}\left((\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I}\right)$	$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_2$
$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$	$\mathcal{N}(\mathbf{x}_{t-1}; \zeta_t \mathbf{x}_t + \xi_t \mathbf{x}_0, \sigma_t^2 \mathbf{I})$	$\mathbf{x}_{t-1} = \zeta_t \mathbf{x}_t + \xi_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_3$
$\int q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t \mid \mathbf{x}_0)\, d\mathbf{x}_t$	$q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)$	$\mathbf{x}_{t-1} = \zeta_t \mathbf{x}_t + \xi_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_3$ $= \zeta_t(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_2) + \xi_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_3$ $= (\zeta_t\sqrt{\bar{\alpha}_t} + \xi_t)\mathbf{x}_0 + (\zeta_t\sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_2 + \sigma_t \boldsymbol{\epsilon}_3)$

From Eq. 2.1, we have

$$\sqrt{\bar{\alpha}}_{t-1} = \zeta_t \sqrt{\bar{\alpha}_t} + \xi_t, \quad 1 - \bar{\alpha}_{t-1} = \zeta_t^2 \left(1 - \bar{\alpha}_t\right) + \sigma_t^2.$$

There are three variables, but only two equations; letting σ_t be a free variable, we have

$$\zeta_t = \sqrt{\frac{1 - \bar{\alpha}_{t-1} - \sigma_t^2}{1 - \bar{\alpha}_t}}, \quad \xi_t = \sqrt{\bar{\alpha}_{t-1}} - \sqrt{\frac{\bar{\alpha}_t\left(1 - \bar{\alpha}_{t-1} - \sigma_t^2\right)}{1 - \bar{\alpha}_t}}.$$

Rearranging terms, we have the following:

$$q\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0\right) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2 \mathbf{I}\right).$$

Note that this is exactly Eq. (7) in Song et al. (2020) with a change in notation in α_t; we adopt the DDPM notation ($\bar{\alpha}_t$) for consistency. Now, one can generate a sample $\mathbf{x}_{t-1}$ from a sample $\mathbf{x}_t$ via:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \underbrace{\left(\frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)}{\sqrt{\bar{\alpha}_t}}\right)}_{\text{predicted } \mathbf{x}_0} + \underbrace{\sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)}_{\text{direction pointing to } \mathbf{x}_t} + \underbrace{\sigma_t \boldsymbol{\epsilon}_t}_{\text{random noise}}. \tag{2.2}$$

During training, we only used the marginals, which are unchanged, so we can use the same model ϵ_θ for any value of σ_t. In principle, we do not have constraints on σ_t, but different choices of σ_t will lead to different characteristics in the sampling process.

- When $\sigma_t = \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}} \cdot \sqrt{1 - \frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$ for all t, the forward process is Markovian, and the generative process becomes a DDPM.
- When $\sigma_t = 0$ for all t except $t = 1$, the sampling process becomes deterministic. More precisely, this deterministic sampling corresponds to an implicit probabilistic model, which is what the *I* in DDIM stands for.
- Smaller noise leads to better sampling quality, particularly when the number of sampling steps is small.

2.4 Relevance to ODEs

Consider the DDIM update (2.2) without noise:

$$\mathbf{x}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right)}{\sqrt{\bar{\alpha}_t}}\right)\mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \boldsymbol{\epsilon}_\theta\left(\mathbf{x}_t, t\right).$$

It can be rewritten as

$$\frac{\mathbf{x}_t}{\sqrt{\bar{\alpha}_t}} - \frac{\mathbf{x}_{t-1}}{\sqrt{\bar{\alpha}_{t-1}}} = \left(\frac{\sqrt{1 - \bar{\alpha}_t}}{\sqrt{\bar{\alpha}_t}} - \frac{\sqrt{1 - \bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_{t-1}}}\right)\epsilon_\theta\left(\mathbf{x}_t, t\right).$$

Assume large enough T, we can introduce a continuous variable s and define

- $\mathbf{x}(s)$ as the continuous analogue of $\mathbf{x}_t$,

- $\bar{\alpha}(s)$ as a smooth function replacing $\bar{\alpha}_t$,
- and $\epsilon_\theta(\mathbf{x}(s), s)$ as the continuous version of the predicted noise.

The discrete update, in the limit $\Delta s \to 0$, becomes

$$\frac{d}{ds}\left(\frac{\mathbf{x}(s)}{\sqrt{\bar{\alpha}(s)}}\right) = \frac{d}{ds}\left(\frac{\sqrt{1-\bar{\alpha}(s)}}{\sqrt{\bar{\alpha}(s)}}\right)\epsilon_\theta\left(\mathbf{x}(s), s\right).$$

Cleaning up, we have

$$\frac{d}{ds}\mathbf{x}(s) = \frac{1}{2\bar{\alpha}(s)} \cdot \frac{d}{ds}\bar{\alpha}(s) \cdot \mathbf{x}(s) + \sqrt{\bar{\alpha}(s)} \cdot \frac{d}{ds}\left(\frac{\sqrt{1-\bar{\alpha}(s)}}{\sqrt{\bar{\alpha}(s)}}\right) \cdot \epsilon_\theta\left(\mathbf{x}(s), s\right).$$

3 Score-based Models

3.1 General Description

Any arbitrary probability distribution can be written in the form of an energy function: $q(\mathbf{x}) = \frac{1}{Z}e^{-f(\mathbf{x})}$, where $f(\mathbf{x})$ is the energy function. Taking the derivative of the log of both sides:

$$\begin{aligned}
\nabla_{\mathbf{x}} \log q(\mathbf{x}) &= \nabla_{\mathbf{x}} \log\left(\frac{1}{Z}e^{-f(\mathbf{x})}\right) \\
&= \nabla_{\mathbf{x}} \log \frac{1}{Z} + \nabla_{\mathbf{x}} \log e^{-f(\mathbf{x})} \\
&= -\nabla_{\mathbf{x}} f(\mathbf{x}).
\end{aligned}$$

The score function $\nabla_{\mathbf{x}} \log q(\mathbf{x})$ points to the direction that minimizes the energy function. Score-based models aim to learn a score model $s_\theta \approx \nabla_{\mathbf{x}} \log q(\mathbf{x})$ by minimizing the Fisher divergence:

$$\mathcal{L}(\theta) = \mathbb{E}_{q(\mathbf{x})}\left[\|\mathbf{s}_\theta(\mathbf{x}) - \nabla \log q(\mathbf{x})\|_2^2\right].$$

The relevance of score models in diffusion models is introduced in Sec. 1.7, and we will further explore score functions and their connection to stochastic differential equations (SDEs) and diffusion processes in Sec. 4.

3.2 Learning Score-based Models

The optimization objective contains $\nabla \log q(\mathbf{x})$, which is infeasible because it requires access to the unknown data score. So, the idea is to train a model $s_\theta(\mathbf{x})$ to approximate $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ without needing to know $p(\mathbf{x})$ or $\nabla \log q(\mathbf{x})$directly. The family of methods that achieve this is called score matching. As a particular example, Hyvärinen (2005) defines a loss function that only involves s_θ and sampling from $q(\mathbf{x})$:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_{q(\mathbf{x})}\left[\|\mathbf{s}_\theta(\mathbf{x}) - \nabla \log q(\mathbf{x})\|_2^2\right] \\
&= \int q(\mathbf{x}) \left\|\nabla_{\mathbf{x}} \log q(\mathbf{x}) - s_\theta(\mathbf{x})\right\|^2 d\mathbf{x} \\
&= \int q(\mathbf{x}) \left(\left\|\nabla_{\mathbf{x}} \log q(\mathbf{x})\right\|^2 - 2s_\theta(\mathbf{x})^\top \nabla_{\mathbf{x}} \log q(\mathbf{x}) + \left\|s_\theta(\mathbf{x})\right\|^2\right) dx.
\end{aligned}$$

Note that the first term $\int q(\mathbf{x}) \left\|\nabla_{\mathbf{x}} \log q(\mathbf{x})\right\|^2 d\mathbf{x}$ does not depend on θ and can be ignored during optimization. Note that

$$\begin{aligned}
\int q(\mathbf{x}) s_\theta(\mathbf{x})^\top \nabla_{\mathbf{x}} \log q(\mathbf{x})\, d\mathbf{x} &= \int q(\mathbf{x}) s_\theta(\mathbf{x})^\top \frac{\nabla_{\mathbf{x}} q(\mathbf{x})}{q(\mathbf{x})}\, d\mathbf{x} \\
&= \int s_\theta(\mathbf{x})^\top \nabla_{\mathbf{x}} q(\mathbf{x})\, d\mathbf{x}
\end{aligned}$$

$$= -\int q(\mathbf{x}) \nabla_{\mathbf{x}} \cdot s_\theta(\mathbf{x}) d\mathbf{x}$$
$$= -\mathbb{E}_{q(\mathbf{x})} \left[\mathrm{Tr}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x})) \right].$$

The loss can be simplified as

$$\mathcal{L}(\theta) = \mathbb{E}_{q(\mathbf{x})} \left[\frac{1}{2} \left\| s_\theta(\mathbf{x}) \right\|^2 + \mathrm{Tr}(\nabla_{\mathbf{x}} s_\theta(\mathbf{x})) \right].$$

At a high level,

- $\frac{1}{2} \left\| s_\theta(\mathbf{x}) \right\|^2$ penalizes overly large gradients, encouraging smooth, stable estimates.
- $\mathrm{Tr}\left(\nabla_{\mathbf{x}} s_\theta(\mathbf{x})\right)$ is the Laplacian of the log-likelihood, it measures how much the vector field is expanding or contracting at a point. If it is negative, it is pulling in (like a sink).
- If you are not familiar with vector calculus, you can think of the second term (the divergence, or trace of the Jacobian) as a second derivative. When it is smaller (more negative), it suggests that the function is curving downward (concave), which typically corresponds to a local maximum.

There are many other parameterizations of score matching; the readers can refer to Denoising Score Matching (Vincent, 2011) and Noise Conditional Score Network (NCSN) (Song and Ermon, 2019) to begin with.

3.3 Sampling: Langevin Dynamics

Langevin dynamics is an iterative process; it starts with an arbitrary prior distribution $\mathbf{x}_0 \sim \pi(\mathbf{x})$, and then iterates the following:

$$\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + c \nabla_{\mathbf{x}} \log q(\mathbf{x}) + \sqrt{2c} \epsilon_t, \quad t = 0, 1, \cdots, K$$

where $\epsilon_t \sim \mathcal{N}(0, I)$. When $c \to 0$ and $K \to \infty$, $\mathbf{x}_K$ converges to a sample from $\mathbf{q}(\mathbf{x})$ under certain regularity conditions[2]. Once we have trained a score-based model $\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log q(\mathbf{x})$, we can sample from $q(\mathbf{x})$ by replacing $\nabla_{\mathbf{x}} \log q(\mathbf{x})$ with $\mathbf{s}_\theta(\mathbf{x})$.

Common confusion about Langevin dynamics:

1. Why add noise?
   - Langevin dynamics is derived from a stochastic differential equation that has $q(x)$ as its stationary distribution. Without noise, the process becomes a deterministic gradient ascent on $\log q(x)$, which just leads to mode seeking, not proper sampling.
   - If we only follow the gradient (i.e., no noise), we could easily get trapped at local maxima of $\log q(x)$. The Gaussian noise helps the process escape local modes, such as in simulated annealing.
   - The random walk component (noise) ensures that the process is ergodic (ignore this if you do not know stochastic processes).

4 SDE and Probability Flow ODE

Diffusion models can be described continuously as stochastic differential equations (SDEs). The connection is established in Song et al. (2021). With the SDE pespective,

The connection between diffusion models and stochastic differential equations (SDEs) is established in Song et al. (2021). Using the Fokker-Planck equation, it also presents equivalent forms of the reverse SDE

(or probablity flow ODE when the variance is set to 0) that samples from the same distribution.

Consequently, both DDPM and DDIM can be seen as discrete forms of the SDEs or ODEs.

4.1 SDE and Its Reverse

The general form of a *Stochastic Differential Equation (SDE)* is:

$$d\mathbf{x}_t = f\left(\mathbf{x}_t, t\right) dt + g\left(\mathbf{x}_t, t\right) dW_t,$$

[2]Within the scope of this write-up, regularity conditions typically refer to the dominated convergence condition, although in some contexts, continuity and compact support may also suffice. These are standard assumptions to make in most machine learning settings.

where $f(\mathbf{x}_t, t)$ is the drift term that describes the deterministic trend of the process over time, $g(\mathbf{x}_t, t)$ is the diffusion term that scales the randomness introduced into the system, and dW_t is an infinitesimal increment of a Wiener process (also known as Brownian motion or random walk).

In general, when f is linear and g is not state-dependent, the SDE is a Gaussian process: $\mathbf{x}_t$ is Gaussian for all $t \geq 0$ given $\mathbf{x}_0$ as assumed in diffusion models (Gaussian marginals). Thus, we consider only this specific case in the scope of this write-up. The derivation in this subsection is based on an asymptotic analysis by assuming infinitesimal time increments. Consider appropriate drift and diffusion terms, this SDE corresponds to DDPM/diffusion models. The derivation of the marginal distribution of an SDE that corresponds to that of DDPM is provided in Sec. 4.3.

The SDE can be discretized as

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = f(\mathbf{x}_t, t)\,\Delta t + g(t)\sqrt{\Delta t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Rearrange,

$$\mathbf{x}_{t+\Delta t} = \underbrace{\mathbf{x}_t + f(\mathbf{x}_t, t)\,\Delta t}_{\text{deterministic}} + \underbrace{g(t)\sqrt{\Delta t}\,\boldsymbol{\epsilon}}_{\text{stochastic}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

In probability,

$$p(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t+\Delta t}; \mathbf{x}_t + f(\mathbf{x}_t, t)\,\Delta t, g(t)^2 \Delta t \cdot \mathbf{I}\right) \propto \exp\left(-\frac{\left\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - f(\mathbf{x}_t, t)\,\Delta t\right\|^2}{2g(t)^2 \Delta t}\right)$$

By Bayes, the reverse

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{x}_{t+\Delta t}) &= \frac{p(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t)\, p(\mathbf{x}_t)}{p(\mathbf{x}_{t+\Delta t})} \\ &= p(\mathbf{x}_{t+\Delta t} \mid \mathbf{x}_t) \exp\left(\log p(\mathbf{x}_t) - \log p(\mathbf{x}_{t+\Delta t})\right) \\ &\propto \exp\left(-\frac{\left\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - f(\mathbf{x}_t, t)\,\Delta t\right\|^2}{2g(t)^2 \Delta t} + \log p(\mathbf{x}_t) - \log p(\mathbf{x}_{t+\Delta t})\right). \end{aligned}$$

Assuming $\Delta t \to 0$, a first-order Taylor expansion on $\log p(\mathbf{x}_{t+\Delta t})$ gives

$$\log p(\mathbf{x}_{t+\Delta t}) \approx \log p(\mathbf{x}_t) + (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) \cdot \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) + \Delta t \frac{\partial}{\partial t} \log p(\mathbf{x}_t).$$

Thus,

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{x}_{t+\Delta t}) &\propto \exp\left(-\frac{\left\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \left[f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right]\Delta t\right\|^2}{2g(t)^2 \Delta t} + \underbrace{\mathcal{O}(\Delta t)}_{\text{dropped as } \Delta t \to 0}\right) \\ &= \exp\left(-\frac{\left\|\mathbf{x}_t - \mathbf{x}_{t+\Delta t} + \left[f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right]\Delta t\right\|^2}{2g(t)^2 \Delta t}\right) \\ &\approx \exp\left(-\frac{\left\|\mathbf{x}_t - \mathbf{x}_{t+\Delta t} + \left[f(\mathbf{x}_{t+\Delta t}, t+\Delta t) - g(t+\Delta t)^2 \nabla_{\mathbf{x}_{t+\Delta t}} \log p(\mathbf{x}_{t+\Delta t})\right]\Delta t\right\|^2}{2g(t+\Delta t)^2 \Delta t}\right), \end{aligned}$$

where the last equation follows as a first-order approximation by assuming $\Delta t \to 0$. This is a Gaussian with mean $\mathbf{x}_{t+\Delta t} - \left[f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right]\Delta t$ and variance $g(t+\Delta t)^2 \Delta t \cdot \mathbf{I}$. Taking $\Delta t \to 0$, this probability corresponds to the reverse SDE:

$$d\mathbf{x}_t = \left[f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] dt + g(t) dW_t. \tag{4.1}$$

4.2 Sampling and Training

Given the reverse SDE (4.1), we can sample through the reverse process:

$$\mathbf{x}_t = \mathbf{x}_{t+\Delta t} - \left[f(\mathbf{x}_{t+\Delta t}, t+\Delta t) - g(t+\Delta t)^2 \nabla_{\mathbf{x}_{t+\Delta t}} \log p_{t+\Delta t}(\mathbf{x}_{t+\Delta t})\right]\Delta t - g(t+\Delta t)\sqrt{\Delta t}\boldsymbol{\epsilon}, \ \boldsymbol{\epsilon} \sim \mathcal{N}(0, 1).$$

As the marginal $p_t(\mathbf{x}_t \mid \mathbf{x}_0)$ is accessible (a sample derivation is shown in Sec. 4.3), we aim to express the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ in terms of these marginals:

$$\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log\left(\mathbb{E}_{\mathbf{x}_0}\left[p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right]\right)$$

$$
\begin{aligned}
&= \frac{\nabla_{\mathbf{x}_t}\left(\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]\right)}{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]} \\
&= \frac{\mathbb{E}_{\mathbf{x}_0}\left[\nabla_{\mathbf{x}_t} p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]}{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]} \\
&= \frac{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right) \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]}{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]},
\end{aligned}
$$

where the third equality assumes regularity conditions allowing gradient and expectation to commute. Note that the score function can, in principle, be computed analytically by evaluating the entire dataset, though this is computationally prohibitive, as we have to compute for each generation trajectory.

Similar to Sec. 3, we learn a score model to approximate the score function. For any $\mathbf{x}_t$, the score function can be learned through

$$
\min_\theta \left(\frac{\mathbb{E}_{\mathbf{x}_0}\left[p_t(\mathbf{x}_t \mid \mathbf{x}_0)\left\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right\|^2\right]}{\mathbb{E}_{\mathbf{x}_0}\left[p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right]} \right). \tag{4.2}
$$

Since $\mathbb{E}_{\mathbf{x}_0}\left[p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right]$ is independent of θ, Eq. (4.2) reduces to

$$
\min_\theta \mathbb{E}_{\mathbf{x}_0}\left[p_t(\mathbf{x}_t \mid \mathbf{x}_0)\left\|\mathbf{s}_\theta(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t \mid \mathbf{x}_0)\right\|^2\right]. \tag{4.3}
$$

Last but not least, Eq. (4.3) is for a fixed $\mathbf{x}_t$, we should take the expectation over $\mathbf{x}_t$, and the final optimization objective becomes

$$
\min_\theta \mathbb{E}_{\mathbf{x}_0, \mathbf{x}_t \sim p_t(\mathbf{x}_t \mid \mathbf{x}_0)}\left[\left\|\mathbf{s}_\theta\left(\mathbf{x}_t, t\right) - \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right\|^2\right]. \tag{4.4}
$$

As we have shown in Sec. 1.7, the score function is the noise. It is straightforward to see that Eq. (4.4) is equivalent to denoising, up to a time-dependent weighting term.

Remark 4.1. *Most readers are familiar with learning to approximate a function using MSE. It might seem different here. Our aim is to learn the score function $\nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t\right)$, but here the MSE is between the score model and the marginal score function. In fact, MSE works the same even if the objective is expressed as a weighted average and the underlying reasoning is rather similar. We will show why Eq. (4.3) learns $\mathbf{s}_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t\right)$.*

Proof. First, expand the loss function

$$
\begin{aligned}
L\left(\mathbf{s}_\theta\right) &= \mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\left\|\mathbf{s}_\theta\left(\mathbf{x}_t, t\right) - \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right\|^2\right] \\
&= \mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\left(\left\|\mathbf{s}_\theta\right\|^2 - 2\mathbf{s}_\theta \cdot \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right) + \left\|\nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right\|^2\right)\right] \\
&= \left\|\mathbf{s}_\theta\right\|^2 \cdot \underbrace{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]}_{=p_t\left(\mathbf{x}_t\right)} - 2\mathbf{s}_\theta \cdot \mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right) \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right] + \text{ const}
\end{aligned}
$$

Now,

$$
\frac{\partial L}{\partial \mathbf{s}_\theta} = 2p_t\left(\mathbf{x}_t\right)\mathbf{s}_\theta - 2\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right) \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right].
$$

Setting $\frac{\partial L}{\partial \mathbf{s}_\theta}$ equal to 0, we have

$$
\mathbf{s}_\theta^\star\left(\mathbf{x}_t\right) = \frac{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right) \nabla_{\mathbf{x}_t} \log p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]}{\mathbb{E}_{\mathbf{x}_0}\left[p_t\left(\mathbf{x}_t \mid \mathbf{x}_0\right)\right]} = \nabla_{\mathbf{x}_t} \log p\left(\mathbf{x}_t\right).
$$

□

4.3 Diffusion Models are SDEs: VP-SDE

Consider the SDE

$$
d\mathbf{x}_t = -\frac{\beta(t)}{2}\mathbf{x}_t\, dt + \sqrt{\beta(t)}\, dW_t,
$$

define

$$
\gamma_t = \exp\left(-\frac{1}{2}\int_0^t \beta(s)\, ds\right), \quad Y_t = \gamma_t^{-1}\mathbf{x}_t.
$$

Note that

$$dY_t = d(\gamma_t^{-1}\mathbf{x}_t) = \gamma_t^{-1}d\mathbf{x}_t + \mathbf{x}_t\, d(\gamma_t^{-1}).$$

Since $\gamma_t^{-1} = e^{\frac{1}{2}\int_0^t \beta(s)ds}$, we have

$$\frac{d}{dt}\gamma_t^{-1} = \frac{1}{2}\beta(t)\gamma_t^{-1} \Rightarrow d(\gamma_t^{-1}) = \frac{1}{2}\beta(t)\gamma_t^{-1}dt,$$

then

$$\begin{aligned} dY_t &= \gamma_t^{-1}\left(-\frac{1}{2}\beta(t)\mathbf{x}_t dt + \sqrt{\beta(t)}dW_t\right) + \mathbf{x}_t \cdot \frac{1}{2}\beta(t)\gamma_t^{-1}dt \\ &= \gamma_t^{-1}\sqrt{\beta(t)}dW_t. \end{aligned}$$

Integrating from 0 to t:

$$\begin{aligned} Y_t &= Y_0 + \int_0^t \gamma_s^{-1}\sqrt{\beta(s)}dW_s \\ &= \mathbf{x}_0 + \int_0^t \gamma_s^{-1}\sqrt{\beta(s)}dW_s. \end{aligned}$$

So,

$$\mathbf{x}_t = \gamma_t Y_t = \gamma_t\mathbf{x}_0 + \gamma_t\int_0^t \gamma_s^{-1}\sqrt{\beta(s)}dW_s.$$

Thus $\mathbf{x}_t \mid \mathbf{x}_0 \sim \mathcal{N}(\gamma_t\mathbf{x}_0, \sigma_t^2 I)$, where the mean is γ_t and the variance is

$$\sigma_t^2 = \mathrm{Var}\left(\gamma_t\int_0^t \gamma_s^{-1}\sqrt{\beta(s)}dW_s\right).$$

By Itô isometry (see Appendix A.3),

$$\sigma_t^2 = \gamma_t^2\int_0^t \left(\gamma_s^{-1}\right)^2\beta(s)ds.$$

Note that

$$\gamma_s^{-1} = e^{\frac{1}{2}\int_0^s \beta(r)dr} \Rightarrow (\gamma_s^{-1})^2 = e^{\int_0^s \beta(r)dr}.$$

So

$$\sigma_t^2 = \gamma_t^2\int_0^t e^{\int_0^s \beta(r)dr}\beta(s)ds.$$

Simplify this by change of variables. Let

$$u(s) = \int_0^s \beta(r)dr \Rightarrow \frac{du}{ds} = \beta(s),$$

then

$$\sigma_t^2 = \gamma_t^2\int_0^t e^{u(s)}\beta(s)ds = \gamma_t^2\int_0^t e^{u(s)}\frac{du}{ds}ds = \gamma_t^2\int_{u(0)}^{u(t)} e^u du = \gamma_t^2(e^{u(t)} - e^{u(0)}).$$

Since $u(0) = 0$, and $u(t) = \int_0^t \beta(s)ds$, we get

$$\sigma_t^2 = \gamma_t^2(e^{\int_0^t \beta(s)ds} - 1).$$

Moreover,

$$\gamma_t^2 = e^{-\int_0^t \beta(s)ds} \Rightarrow \sigma_t^2 = \left(e^{-\int_0^t \beta(s)ds}\right)\left(e^{\int_0^t \beta(s)ds} - 1\right) = 1 - e^{-\int_0^t \beta(s)ds}.$$

In conclusion,

$$\mathbf{x}_t \mid \mathbf{x}_0 \sim \mathcal{N}\left(\gamma_t\mathbf{x}_0, \sigma_t^2 I\right),$$

where

$$\underbrace{\gamma_t = e^{-\frac{1}{2}\int_0^t \beta(s)\,ds}}_{\text{mean decay factor}} \quad \text{and} \quad \underbrace{\sigma_t^2 = 1 - e^{-\int_0^t \beta(s)\,ds}}_{\text{variance growth}}.$$

Note that $\sigma_t^2 = 1 - \gamma_t^2$. With appropriate $\beta(s)$ such that $\gamma_t = \bar{\alpha}_t$, the marginals at any given time, $\mathbf{x}_t \mid \mathbf{x}_0$, follow exactly as in a diffusion process. Specifically, let $\beta(s) = -\frac{2}{\sqrt{\bar{\alpha}_s}} \frac{d\sqrt{\bar{\alpha}_s}}{ds}$, then

$$\gamma_t = e^{-\frac{1}{2}\int_0^t \beta(s)\, ds} = e^{\int_0^t \frac{1}{\sqrt{\bar{\alpha}_s}} \frac{d\sqrt{\bar{\alpha}_s}}{ds}\, ds}.$$

Note that

$$\frac{1}{\sqrt{\bar{\alpha}_s}} \frac{d\sqrt{\bar{\alpha}_s}}{ds} = \frac{d}{ds}\left[\ln\left(\sqrt{\bar{\alpha}_s}\right)\right],$$

thus

$$\gamma_t = e^{\int_0^t \frac{d}{ds}\left[\ln\left(\sqrt{\bar{\alpha}_s}\right)\right] ds} = \sqrt{\bar{\alpha}_s}.$$

Suppose that $\mathbf{x}_0 \sim \mathcal{N}(0, 1)$, the process is constructed to keep the total variance normalized as time progresses: $\mathrm{Var}\left(\mathbf{x}_t\right) = \bar{\alpha}_t + (1 - \bar{\alpha}_t) = 1$. Therefore, this SDE is called the Variance Preserving SDE (VP-SDE).

Another common type of SDE for diffusion models is the Variance Exploding SDE (VE-SDE), given by

$$d\mathbf{x}_t = \sqrt{\frac{d\left(\sigma_t^2\right)}{dt}} dW_t$$
$$\mathbf{x}_t \mid \mathbf{x}_0 \sim \mathcal{N}\left(\mathbf{x}_0, \sigma_t^2 I\right),$$

where σ_t increases over time and is not bounded by 1. Usually, at the end, $\sigma_T >> \mathbf{x}_0$ dominates the mean $\mathbf{x}_0$, so we can just sample standard Gaussian noise scaled by σ_T.

4.4 Probability Flow ODE

Similar to Section 2, we argue that the training process depends solely on the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$. Therefore, as long as the distribution $p_t(\mathbf{x}_t)$ remains the same, the resulting trained score model remains the same. In particular, the same marginal distribution $p_t(\mathbf{x}_t)$ can arise from different underlying SDEs, offering greater flexibility. SDE describes how individual sample paths of a stochastic process evolve over time. To study the distribution, the Fokker-Planck equation (Risken and Frank, 1996) describes how the probability distribution (density) evolves over time.

Given an diffusion SDE with state-independent diffusion coefficient:

$$d\mathbf{x}_t = f\left(\mathbf{x}_t, t\right) dt + g\left(t\right) dW_t, \tag{4.5}$$

the corresponding reverse SDE is

$$d\mathbf{x}_t = \left[f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] dt + g(t) dW_t. \tag{4.6}$$

The Fokker-Planck equation that "solves" Eq. (4.5) is given by[3]

$$\frac{\partial}{\partial t} p_t(\mathbf{x}_t) = -\nabla_{\mathbf{x}_t} \cdot \left[f(\mathbf{x}_t, t) p_t(\mathbf{x}_t)\right] + \frac{1}{2} g(t)^2 \nabla_{\mathbf{x}_t}^2 p_t(\mathbf{x}_t). \tag{4.7}$$

We can adjust the drift term to compensate for changes in the diffusion coefficient. For any function $\sigma(t)$ such that $\sigma(t)^2 \leq g(t)^2$, Eq. (4.7) can be rewritten as

$$\begin{aligned} \frac{\partial}{\partial t} p_t(\mathbf{x}_t) &= -\nabla_{\mathbf{x}} \cdot \left[f(\mathbf{x}_t, t) p_t(\mathbf{x}_t) - \frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} p_t(\mathbf{x}_t)\right] + \frac{1}{2}\sigma(t)^2 \nabla_{\mathbf{x}_t}^2 p_t(\mathbf{x}_t) \\ &= -\nabla_{\mathbf{x}_t} \cdot \left[\left(f(\mathbf{x}_t, t) - \frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right) p_t(\mathbf{x}_t)\right] + \frac{1}{2}\sigma(t)^2 \nabla_{\mathbf{x}_t}^2 p_t(\mathbf{x}_t). \end{aligned} \tag{4.8}$$

Eq. (4.8) can be seen as the Fokker-Planck equation with drift $\left(f(\mathbf{x}_t, t) - \frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right)$ and diffusion coefficient $\sigma(t)$. Following Eq. (4.5) and Eq. (4.6), the forward and reverse SDEs, respectively, are

$$d\mathbf{x}_t = \left(f(\mathbf{x}_t, t) - \frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right) dt + \sigma\left(t\right) dW_t, \tag{4.9}$$

and

$$\begin{aligned} d\mathbf{x}_t &= \left[\left(f(\mathbf{x}_t, t) - \frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right) - \sigma(t)^2 \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] dt + \sigma(t) dW_t \\ &= \left[f(\mathbf{x}_t, t) - \frac{1}{2}\left(g(t)^2 + \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)\right] dt + \sigma(t) dW_t. \end{aligned} \tag{4.10}$$

[3]The proof can be easily found online.

When $\sigma(t) = 0$, Eq. (4.10) reduces to an ODE, termed the probability flow ODE. The sampling is deterministic. It can be shown that the Euler-Maruyama discretization of the probability flow ODE is equivalent to the ODE derived in Sec. 2.4. In summary, Eq. (4.5)-(4.6) and Eq. (4.9)-(4.10) define the same probability flow $p_t(\mathbf{x}_t)$. Since the score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ is unknown, we can only train with Eq. (4.5); however, we can sample with Eq. (4.10) with smaller or even no variance.

Remark 4.2. *To intuitively understand this, we know that the score function, i.e., the gradient of the log probability, points to the high-density regions.*

- *For the forward SDE (Eq. (4.5) and Eq. (4.9)), randomness (the Wiener process) causes the density to spread out (diffuse). If you reduce the randomness from $g(t)$ to $\sigma(t)$, the density will spread less. To compensate for it, we include a correction term $-\frac{1}{2}\left(g(t)^2 - \sigma(t)^2\right) \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ in the drift pointing towards the low-density regions.*
- *For the reverse SDE (Eq. (4.6) and Eq. (4.10)), the probability flows from low-density regions to high-density regions. In the reverse process, when we reduce the randomness, it might get over-condensed in high-density regions (recall that noise spreads out the density). Therefore, we reduce the "step-size" of moving towards the score direction from $g(t)^2$ to $\frac{1}{2}\left(g(t)^2 + \sigma(t)^2\right)$.*

4.5 Understanding the Fokker-Planck Equation

Consider the general form of a SDE

$$d\mathbf{x}_t = f\left(\mathbf{x}_t, t\right) dt + g\left(\mathbf{x}_t, t\right) dW_t.$$

The "solution" to this SDE will be a probability density function $p_t(\mathbf{x}_t)$ such that $\mathbf{x}_t \sim p_t(\mathbf{x}_t)$. In general, the mean, $\mathbb{E}[p(x,t)]$, is the deterministic solution to $\frac{d\mathbf{x}}{dt} = f\left(\mathbf{x}_t, t\right)$ while the variance, $\mathrm{Var}[p(x,t)]$, incorporates the function $g\left(\mathbf{x}_t, t\right)$. Rather than tracking individual sample paths, we want to understand how the distribution over possible states $\mathbf{x}_t$ evolves over time. This is given by the corresponding Fokker-Planck equation:

$$\frac{\partial}{\partial t} p_t(\mathbf{x}_t) = -\nabla_{\mathbf{x}_t} \cdot [f(\mathbf{x}_t, t) p_t(\mathbf{x}_t)] + \frac{1}{2} g(\mathbf{x}_t, t)^2 \nabla^2_{\mathbf{x}_t} p_t(\mathbf{x}_t).$$

Remark 4.3. *To intuitively understand the Fokker-Planck equation, we explain each term below. A more detailed and mathematical explanation can be found in Peter E. Holderrieth's blog.*

- $\frac{\partial}{\partial t} p_t(\mathbf{x}_t)$*: This term describes how the probability density evolves over time, reflecting the net inflow*[4] *of the probability mass.*
- $-\nabla_{\mathbf{x}_t} \cdot [f(\mathbf{x}_t, t) p_t(\mathbf{x}_t)]$*: This negative divergence term reflects the deterministic net inflow*[5] *under the influence of f. Think of f as a wind field, it pushes the probability density in certain directions.*
- $\frac{1}{2} g(\mathbf{x}_t, t)^2 \nabla^2_{\mathbf{x}_t} p_t(\mathbf{x}_t)$*: This term describes how fast the probability mass spreads out. $\nabla^2 p$ (the Laplacian of the density) measures how curved the density is. High curvature $\rightarrow$ sharp peak $\rightarrow$ probability "wants" to flow outward to flatten it out. So, diffusion drives flow from high to low concentration.*

Table 1: Comparison of drift and diffusion in SDEs and Fokker-Planck equations

Feature	Drift $f\left(\mathbf{x}_t, t\right)$	Diffusion $g\left(\mathbf{x}_t, t\right)$
Role in SDE	Deterministic evolution	Randomness
Role in Fokker-Planck	Convection (shift of density)	Spreading (dispersion of density)
Mathematical	First derivative (gradient) Indicates direction of flow	Second derivative (Laplacian) Indicates spread/curvature
Analogy	Wind pushing a particle	Heat causing it to diffuse

[4] A positive value of $\frac{\partial}{\partial t} p_t(\mathbf{x}_t)$ indicates an inflow of probability mass, i.e., an increase in density at the point $\mathbf{x}_t$ over time. A negative value of $\frac{\partial}{\partial t} p_t(\mathbf{x}_t)$ indicates an outflow. By conservation of probability, a positive divergence term indicates a net outflow.

[5] This term is negative as divergence indicates a net outflow, so negating the divergence term indicates a net inflow.

References

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*.

Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

Kingma, D. P., Salimans, T., Poole, B., and Ho, J. (2023). Variational diffusion models.

Luo, C. (2022). Understanding diffusion models: A unified perspective.

Risken, H. and Frank, T. (1996). *The Fokker-Planck Equation: Methods of Solution and Applications*. Springer Series in Synergetics. Springer Berlin Heidelberg.

Song, J., Meng, C., and Ermon, S. (2020). Denoising diffusion implicit models. *arXiv:2010.02502*.

Song, Y. and Ermon, S. (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

A Mathematical Preliminary

A.1 Entropy, Cross-entropy, and KL Divergence

1. Entropy $H(p)$ is a measure of the uncertainty in the distribution.
$$H(p) = \mathbb{E}_{X\sim p}[-\log p(X)]$$
   - Non-negativity: $H(p) \geq 0$, with equality if and only if p is a degenerate distribution (all probability mass is on one outcome).
2. Cross-entropy $H(p, q)$ measures the expected number of bits needed to encode the data from p using the distribution q.
$$H(p, q) = \mathbb{E}_{X\sim p}[-\log q(X)]$$
   - Non-negativity: Cross-entropy is always non-negative.
   - Asymmetric: Cross-entropy is not symmetric, i.e., $H(p, q) \neq H(q, p)$.
   - Lower Bound: Cross-entropy $H(p, q)$ is greater than or equal to the entropy $H(p)$, i.e., $H(p, q) \geq H(p)$.
   - Equality: $H(p, q) = H(p)$ if and only if $p = q$, i.e., when the distributions are the same.
3. KL Divergence $D_{\mathrm{KL}}(p\|q)$ is a measure of how one probability distribution diverges from another.
$$D_{\mathrm{KL}}(p\|q) = \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right]$$
   - Non-negativity: $D_{\mathrm{KL}}(p\|q) \geq 0$, with equality if and only if $p = q$. This is a consequence of Jensen's inequality.
   - Asymmetry: KL divergence is not symmetric, meaning $D_{\mathrm{KL}}(p\|q) \neq D_{\mathrm{KL}}(q\|p)$.
   - Relation to Cross-Entropy: The KL divergence can be expressed as the difference between the cross-entropy and the entropy:
$$D_{\mathrm{KL}}(p\|q) = H(p, q) - H(p).$$

A.2 Log-likelihoods and KL Divergence of Gaussians

Log Likelihood for Multivariate Gaussian

- For a multivariate Gaussian distribution $x \sim \mathcal{N}(\mu, \Sigma)$ with mean vector $\mu \in \mathbb{R}^D$ and covariance matrix $\Sigma \in \mathbb{R}^{D\times D}$ (i.e., $\theta = [\mu, \Sigma]$), the probability density function is:
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$
- Assuming Σ is diagonal:
$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{D/2}\prod_{i=1}^{D}\sigma_i} \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-\mu_i)^2}{\sigma_i^2}\right).$$

- The log-likelihood is:

$$\log p(x;\mu,\Sigma) = \log\left(\frac{1}{(2\pi)^{D/2}\prod_{i=1}^{D}\sigma_i}\right) - \frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-\mu_i)^2}{\sigma_i^2}$$
$$= -\frac{D}{2}\log(2\pi) - \sum_{i=1}^{D}\log\sigma_i - \frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-\mu_i)^2}{\sigma_i^2}.$$

- The log-likelihood consists of 3 terms:
  - A constant term: $-\frac{D}{2}\log(2\pi)$.
  - A variance-dependent term: $-\sum_{i=1}^{D}\log\sigma_i$.
  - A MSE-like term penalizing deviations from the mean: $-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-\mu_i)^2}{\sigma_i^2}$.

KL Divergence for Two Multivariate Gaussians

- If $p \sim \mathcal{N}(\mu_p, \Sigma_p)$ and $q \sim \mathcal{N}(\mu_q, \Sigma_q)$, then

$$\mathrm{KL}(p\|q) = \frac{1}{2}\left(\log\frac{|\Sigma_q|}{|\Sigma_p|} - D + \mathrm{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu_q-\mu_p)^\top\Sigma_q^{-1}(\mu_q-\mu_p)\right).$$

- Assuming diagonal covariance matrices:

$$D_{\mathrm{KL}}(p\|q) = \frac{1}{2}\sum_{i=1}^{D}\left(\log\frac{\sigma_{q,i}^2}{\sigma_{p,i}^2} - 1 + \frac{\sigma_{p,i}^2}{\sigma_{q,i}^2} + \frac{(\mu_{p,i}-\mu_{q,i})^2}{\sigma_{q,i}^2}\right).$$

- The KL divergence consists of 3 terms:
  - $\frac{1}{2}\sum_{i=1}^{D}\log\frac{\sigma_{q,i}^2}{\sigma_{p,i}^2}$: accounts for the difference in variances.
  - $\frac{1}{2}\sum_{i=1}^{D}\frac{\sigma_{p,i}^2}{\sigma_{q,i}^2}$: measures the scaling difference in variances.
  - $\frac{1}{2}\sum_{i=1}^{D}\frac{(\mu_{p,i}-\mu_{q,i})^2}{\sigma_{q,i}^2}$: penalizes differences in the means (MSE normalized by q's variance).

Special Case: Fixed Variance

- If the variances in the Gaussian distributions are fixed (i.e., constants and not learnable), then maximizing the log-likelihood or minimizing the KL divergence reduces to minimizing the mean squared error (MSE) between the means of the distributions.

A.3 Itô Isometry

Suppose $f(t)$ is square-integrable, i.e.:

$$\mathbb{E}\left[\int_0^T f(t)^2 dt\right] < \infty.$$

Itô's isometry says

$$\mathbb{E}\left[\left(\int_0^T f(t)dW_t\right)^2\right] = \mathbb{E}\left[\int_0^T f(t)^2 dt\right].$$

In terms of variance,

$$\mathrm{Var}\left(\int_0^T f(t)dW_t\right) = \mathbb{E}\left[\left(\int_0^T f(t)dW_t\right)^2\right],$$

because the Itô integral has mean zero

$$\mathbb{E}\left[\int_0^T f(t)dW_t\right] = 0.$$