Introduction for AI in Robotics
Diffusion Model

Byungchul Kim

1 Overview

This document aims to explain basic math for machine learning and to introduce how DDPM paper simplified the loss function.

- The diffusion model generates data from nois

S1.1. Diffusion Model Overview Diffusion model has shown supprising

Diffusion model consists of two process: (1) forward noising process and (2) reverse denoising process. The forward process can be represented in

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

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)) \tag{2}$$

What distinguishes diffusion models from other types of latent variable models is that the approximate posterior $q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$, called the *forward process* or *diffusion process*, is fixed to a Markov chain that gradually adds Gaussian noise to the data according to a variance schedule $\beta_1, \ldots, \beta_T$:

$$\begin{aligned} q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) &:= \prod_{t=1}^{T} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}), \\ q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) &:= \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\, \mathbf{x}_{t-1}, \beta_t \mathbf{I}), \qquad \alpha_t = 1 - \beta_t \end{aligned} \tag{3}$$

M1.1. Mathematical Expression of Machine Learning (ML) Model

In a traditional setting, one may consider a deterministic relationship between input and output:

$$y = f_\theta(x), \tag{4}$$

where x is a given input and θ denotes the parameters of the model. The notation f_θ emphasizes that the function is parameterized and that θ is learned (or fitted) through optimization. We can refer this case as *deterministic model*.

However, in machine learning, inputs are typically viewed as random variables drawn from an underlying data distribution, commonly expressed as $x \sim p(x)$. This reflects the fact that models are trained on datasets sampled from a population rather than on a single deterministic input.

Accordingly, the relationship between input and output is often modeled probabilistically as a conditional distribution:

$$y \sim p_\theta(y \mid x), \tag{5}$$

which captures uncertainty in the output given an input. We can refer this case as *stochastic model*. Importantly, this formulation generalizes the deterministic mapping in Eq. (4), as a deterministic model can be interpreted as a special case of $p_\theta(y \mid x)$ (e.g., a distribution with zero variance).

Note that depending on the task, the objective may instead be to model the data distribution itself. In unsupervised or generative settings, the goal is to learn $p(x)$ directly by approximating it with a parameterized model $p_\theta(x)$, from which new samples $x \sim p_\theta(x)$ can be generated.

M1.2. Definition and Interpretation of Expectation

When models are trained on data distributions, their objective functions are defined in terms of expectations rather than evaluations on a single instance. Therefore, we also briefly explain how we obtain the expectation of a function under certain distribution.

The expectation of a function $f(\mathbf{x})$ with respect to a probability density $p(\mathbf{x})$ is defined as

$$\mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \int f(\mathbf{x})\, p(\mathbf{x})\, d\mathbf{x}. \tag{6}$$

Here, $p(\mathbf{x})$ denotes a probability density function, while $f(\mathbf{x})$ is an arbitrary function of $\mathbf{x}$ and does not need to be a probability distribution. Intuitively, the expectation represents the average value of $f(\mathbf{x})$ when $\mathbf{x}$ is sampled from $p(\mathbf{x})$:

$$\mathbf{x} \sim p(\mathbf{x}), \qquad \mathbb{E}_{p(\mathbf{x})}[f(\mathbf{x})] = \mathbb{E}[f(\mathbf{x})]. \tag{7}$$

M1.3. Bayes' Rule In stochastic models, the models are expressed in terms of conditional probabilities ((5)). Therefore, it is useful to keep in mind of Bayes' rule, which relates conditional and marginal distributions:

$$p(y \mid x) = \frac{p(x, y)}{p(x)} = \frac{p(x \mid y)\, p(y)}{p(x)}. \tag{8}$$

Here, $p(x \mid y)$ is called the likelihood, $p(y)$ is the prior, and $p(x)$ is the marginal likelihood (or evidence), given by

$$p(x) = \int p(x \mid y)\, p(y)\, dy. \tag{9}$$

Bayes' rule is particularly useful because directly modeling $p(y \mid x)$ can be challenging. If we can easily obtain $p(x \mid y)$ instead of $p(y \mid x)$, we can use Bayes' rule. Note that, eq(9) is sometimes useful when we also need to know $p(y)$ and $p(x)$ if we use Bayes' rule.

M1.4. Kullback–Leibler (KL) Divergence

In stochastic models, it is often useful to quantify how different two probability distributions are. The Kullback–Leibler (KL) divergence provides a measure of discrepancy between a reference distribution $p(\mathbf{x})$ and an approximating distribution $q(\mathbf{x})$, and is defined as

$$D_{\mathrm{KL}}(p \,\|\, q) = \int p(\mathbf{x})\, \log \frac{p(\mathbf{x})}{q(\mathbf{x})}\, d\mathbf{x}. \tag{10}$$

The KL divergence can be interpreted as the expected log-difference between the two distributions when samples are drawn from $p(\mathbf{x})$:

$$D_{\mathrm{KL}}(p \,\|\, q) = \mathbb{E}_{p(\mathbf{x})} \left[\log \frac{p(\mathbf{x})}{q(\mathbf{x})} \right]. \tag{11}$$

Properties:

- $D_{\mathrm{KL}}(p \,\|\, q) \geq 0$, and $D_{\mathrm{KL}}(p \,\|\, q) = 0$ if and only if $p = q$ (almost everywhere).
- It is *not* symmetric, i.e., $D_{\mathrm{KL}}(p \,\|\, q) \neq D_{\mathrm{KL}}(q \,\|\, p)$.
- It is not a true distance metric (does not satisfy symmetry or triangle inequality).

Interpretation: KL divergence measures how well the distribution $q(\mathbf{x})$ approximates $p(\mathbf{x})$. Minimizing $D_{\mathrm{KL}}(p \,\|\, q)$ encourages q to assign high probability to regions where p has high probability.

Role in Machine Learning: KL divergence appears in many machine learning objectives. For example, maximum likelihood estimation can be interpreted as minimizing $D_{\mathrm{KL}}(p_{\mathrm{data}} \,\|\, p_\theta)$, and in variational inference, one often minimizes $D_{\mathrm{KL}}(q \,\|\, p)$ to approximate an intractable distribution.

M1.5. Closed-Form KL Divergence Between Gaussian Distributions

For ELBO derivations, a particularly important case is the KL divergence between two Gaussian distributions. Consider

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_p, \Sigma_p), \qquad q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \Sigma_q), \tag{12}$$

where $\mathbf{z} \in \mathbb{R}^k$. The KL divergence from q to p is defined as

$$D_{\mathrm{KL}}(q \,\|\, p) = \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{p(\mathbf{z})}\right]. \tag{13}$$

Using the log-density of a multivariate Gaussian,

$$\log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Sigma) = -\frac{k}{2}\log(2\pi) - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^T\Sigma^{-1}(\mathbf{z}-\boldsymbol{\mu}), \tag{14}$$

we obtain

$$\log q(\mathbf{z}) - \log p(\mathbf{z}) = -\frac{1}{2}\log|\Sigma_q| + \frac{1}{2}\log|\Sigma_p| - \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_q)^T\Sigma_q^{-1}(\mathbf{z}-\boldsymbol{\mu}_q) + \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\mathbf{z}-\boldsymbol{\mu}_p), \tag{15}$$

since the constant term $-\frac{k}{2}\log(2\pi)$ cancels out. Therefore,

$$D_{\mathrm{KL}}(q \,\|\, p) = \frac{1}{2}\log\frac{|\Sigma_p|}{|\Sigma_q|} + \frac{1}{2}\mathbb{E}_q\left[(\mathbf{z}-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\mathbf{z}-\boldsymbol{\mu}_p)\right] - \frac{1}{2}\mathbb{E}_q\left[(\mathbf{z}-\boldsymbol{\mu}_q)^T\Sigma_q^{-1}(\mathbf{z}-\boldsymbol{\mu}_q)\right]. \tag{16}$$

Now we evaluate the two expectation terms separately. First, since $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}_q, \Sigma_q)$,

$$\mathbb{E}_q\left[(\mathbf{z}-\boldsymbol{\mu}_q)^T\Sigma_q^{-1}(\mathbf{z}-\boldsymbol{\mu}_q)\right] = \mathrm{tr}(\Sigma_q^{-1}\Sigma_q) = k. \tag{17}$$

Next, write

$$\mathbf{z} - \boldsymbol{\mu}_p = (\mathbf{z} - \boldsymbol{\mu}_q) + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p). \tag{18}$$

Then

$$\begin{aligned} &\mathbb{E}_q\left[(\mathbf{z}-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\mathbf{z}-\boldsymbol{\mu}_p)\right] \\ &= \mathbb{E}_q\left[\big((\mathbf{z}-\boldsymbol{\mu}_q) + (\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)\big)^T\Sigma_p^{-1}\big((\mathbf{z}-\boldsymbol{\mu}_q) + (\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)\big)\right]. \end{aligned} \tag{19}$$

Expanding this expression and using $\mathbb{E}_q[\mathbf{z}-\boldsymbol{\mu}_q] = \mathbf{0}$ gives

$$\mathbb{E}_q\left[(\mathbf{z}-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\mathbf{z}-\boldsymbol{\mu}_p)\right] = \mathrm{tr}(\Sigma_p^{-1}\Sigma_q) + (\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\boldsymbol{\mu}_q-\boldsymbol{\mu}_p). \tag{20}$$

Substituting these results back yields the closed-form expression

$$D_{\mathrm{KL}}(q \,\|\, p) = \frac{1}{2}\left[\log\frac{|\Sigma_p|}{|\Sigma_q|} - k + \mathrm{tr}(\Sigma_p^{-1}\Sigma_q) + (\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)^T\Sigma_p^{-1}(\boldsymbol{\mu}_q-\boldsymbol{\mu}_p)\right]. \tag{21}$$

Special case: standard normal prior. If

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, I), \qquad q(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma), \tag{22}$$

then

$$D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})) = \frac{1}{2}\left[-\log|\Sigma| - k + \mathrm{tr}(\Sigma) + \boldsymbol{\mu}^T\boldsymbol{\mu}\right]. \tag{23}$$

If $\Sigma = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_k^2)$ is diagonal, this further reduces to

$$D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})) = \frac{1}{2}\sum_{i=1}^{k}\left(\mu_i^2 + \sigma_i^2 - \log\sigma_i^2 - 1\right). \tag{24}$$

This diagonal form is commonly used in variational autoencoders, where the KL term in the ELBO can be computed analytically.

M1.6. Why do we optimize $\log p_\theta(x)$ instead of $p_\theta(x)$?

In diffusion models (and more generally in probabilistic modeling), the goal is to maximize the likelihood of data:

$$\max_{\theta} \; p_\theta(\mathbf{x}_0). \tag{25}$$

However, in practice, we instead optimize the log-likelihood:

$$\max_{\theta} \; \log p_\theta(\mathbf{x}_0). \tag{26}$$

This transformation is justified by several important properties:

- **Monotonicity:** The logarithm is a strictly increasing function, so the optimizer remains unchanged:

$$\arg\max_{\theta} \; p_\theta(\mathbf{x}_0) = \arg\max_{\theta} \; \log p_\theta(\mathbf{x}_0). \tag{27}$$

- **Numerical Stability:** Probabilities are often extremely small:

$$p_\theta(\mathbf{x}_0) \ll 1. \tag{28}$$

  Taking the logarithm prevents numerical underflow and improves stability.

- **Product to Sum Conversion:** In latent variable models such as DDPM, the likelihood involves a product over multiple steps:

$$p_\theta(\mathbf{x}_{0:T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t). \tag{29}$$

  Taking the logarithm converts this into a sum:

$$\log p_\theta(\mathbf{x}_{0:T}) = \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t), \tag{30}$$

  which is significantly easier to optimize.

- **Gradient Behavior:** The gradient of the log-likelihood has a convenient form:

$$\nabla_\theta \log p_\theta(x) = \frac{1}{p_\theta(x)} \nabla_\theta p_\theta(x), \tag{31}$$

  which avoids vanishing gradients when $p_\theta(x)$ is very small.

- **Connection to KL Divergence and ELBO:** In diffusion models, the log-likelihood enables variational inference:

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)} \right], \tag{32}$$

  leading to the Evidence Lower Bound (ELBO), which is tractable and forms the basis of training.

Summary: We optimize $\log p_\theta(x)$ instead of $p_\theta(x)$ because it preserves the objective while providing numerical stability, simplifying optimization, and enabling tractable learning via ELBO.

M1.7. Marginalization of Joint Probability

In probabilistic modeling, we often start with a joint distribution over multiple random variables:

$$p(x, y). \tag{33}$$

However, in many cases, we are only interested in one variable (e.g., x). To obtain the distribution of x alone, we must remove the dependency on y. This process is called **marginalization**.

$$p(x) = \sum_{y} p(x, y) \quad \text{(discrete case)} \tag{34}$$

$$p(x) = \int p(x, y)\, dy \quad \text{(continuous case)} \tag{35}$$

This transformation is justified by the following interpretations:

- **Summing Over All Possibilities:** The marginal probability $p(x)$ accounts for all possible values of y:

$$p(x) = \sum_y p(x, y). \tag{36}$$

This means we consider every way in which x can occur together with y.

- **Removing Dependency:** Marginalization eliminates the influence of y, allowing us to focus only on x:

$$p(x) = \int p(x, y)\, dy. \tag{37}$$

- **Consistency with Probability Laws:** Using conditional probability, we have:

$$p(x, y) = p(x \mid y)\, p(y). \tag{38}$$

Substituting into marginalization:

$$p(x) = \int p(x \mid y)\, p(y)\, dy, \tag{39}$$

which shows that $p(x)$ is an average over all conditional distributions weighted by $p(y)$.

- **Interpretation as Expectation:** Marginalization can be viewed as an expectation over y:

$$p(x) = \mathbb{E}_{y \sim p(y)} \left[p(x \mid y) \right]. \tag{40}$$

- **Role in Latent Variable Models:** In many machine learning models, y (or z) is a latent variable:

$$p_\theta(x) = \int p_\theta(x, z)\, dz. \tag{41}$$

This integral is often **intractable**, which motivates approximation methods such as variational inference.

S1.2. Defining $q(x_t \mid x_0)$ from $q(x_{t-1} \mid x_t)$

The forward (adding noise) process from $t-1$ to t can be expanded to the process from 0 to t with following process; this will be used in S1.3. We start from rewriting eq(3) as

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0, I). \tag{42}$$

Substituting recursively, we can obtain following equation:

$$\begin{aligned} x_t &= \sqrt{\alpha_t} \left(\sqrt{\alpha_{t-1}} x_{t-2} + \sqrt{\beta_{t-1}} \epsilon_{t-1} \right) + \sqrt{\beta_t} \epsilon_t \\ &= \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{\alpha_t \beta_{t-1}} \epsilon_{t-1} + \sqrt{\beta_t} \epsilon_t \end{aligned} \tag{43}$$

Repeating the substitution until x_0,

$$x_t = \sqrt{\alpha_t \alpha_{t-1} \cdots \alpha_1}\, x_0 + \sum_{s=1}^{t} \left(\sqrt{\beta_s \prod_{j=s+1}^{t} \alpha_j}\, \epsilon_s \right) \tag{44}$$

By defining $\bar{\alpha}_t$ as $\prod_{s=1}^{t} \alpha_s$, we can obtain following equation:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sum_{s=1}^{t} \left(\sqrt{\beta_s \prod_{j=s+1}^{t} \alpha_j}\, \epsilon_s \right) \tag{45}$$

Since the second term is a linear combination of independent Gaussian noises,

$$x_t \mid x_0 \sim \mathcal{N}\left(\sqrt{\bar{\alpha}_t}\, x_0, \left(\sum_{s=1}^{t} \beta_s \prod_{j=s+1}^{t} \alpha_j\right) I\right) \tag{46}$$

Again, by defining $\sum_{s=1}^{t} \beta_s \prod_{j=s+1}^{t} \alpha_j$ as $1 - \bar{\alpha}_t$, we obtain

$$q(x_t \mid x_0) = \mathcal{N}\left(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1 - \bar{\alpha}_t) I\right) \tag{47}$$

S1.3. Derivation of Variational Lower Bound (ELBO)

Now we have to define the loss function which can be used in the training process. Note that, this is not the simplified version (the contribution of the DDPM paper). We start with the objective of maximizing the log-likelihood of the data sample $\mathbf{x}_0$:

$$\begin{aligned} \log p_\theta(\mathbf{x}_0) &= \log \int p_\theta(\mathbf{x}_{0:T})\, d\mathbf{x}_{1:T} \\ &= \log \int q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\, d\mathbf{x}_{1:T} \\ &= \log \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right]. \end{aligned} \tag{48}$$

We obtain the following lower bound by applying Jensen's inequality ($\log \mathbb{E}[X] \geq \mathbb{E}[\log X]$):

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right]. \tag{49}$$

In the DDPM paper, the objective is written as the negative log-likelihood to be minimized:

$$L = \mathbb{E}_q[-\log p_\theta(\mathbf{x}_0)] \tag{50}$$

$$\leq \mathbb{E}_q \left[-\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] \tag{51}$$

$$= \mathbb{E}_q \left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{T} \log \frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}\right]. \tag{52}$$

In diffusion models, expectations in the loss function are taken over latent trajectories generated by the forward process:

$$\mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)}[f(\mathbf{x}_{1:T})] = \int f(\mathbf{x}_{1:T})\, q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)\, d\mathbf{x}_{1:T}. \tag{53}$$

Summary: Marginalization allows us to obtain the distribution of a subset of variables by summing or integrating out the others. It is a fundamental operation in probability theory and plays a central role in latent-variable models and modern machine learning.

S1.4. To derive Equation (5), we rewrite the forward transition $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ using the posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$.

From Bayes' rule:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x_0}) = \frac{q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)\, q(\mathbf{x}_t \mid \mathbf{x}_0)}{q(\mathbf{x}_{t-1} \mid \mathbf{x}_0)}. \tag{54}$$

Substituting this expression into the previous loss formulation and applying logarithm identities results in a telescoping sum where most terms cancel. The final decomposition becomes:

$$L = \mathbb{E}_q\Big[\underbrace{D_{KL}\big(q(\mathbf{x}_T \mid \mathbf{x}_0) \parallel p(\mathbf{x}_T)\big)}_{L_T} + \sum_{t>1} \underbrace{D_{KL}\big(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)\big)}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)}_{L_0}\Big]. \tag{55}$$

- L_T **(Prior Matching)** Compares the final forward noise distribution to the standard Gaussian prior.
- L_{t-1} **(Denoising Matching)** This is the main training term. It forces the learned reverse process $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ to match the true posterior $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$.
- L_0 **(Reconstruction)** The final reconstruction step that converts the slightly noisy state $\mathbf{x}_1$ back into the clean data $\mathbf{x}_0$.

S1.5. Now, we can interpret Eq(55) one by one. L_T
We want to compute the posterior distribution

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0).$$

Using Bayes' rule,

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \propto q(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_0)\, q(\mathbf{x}_{t-1} \mid \mathbf{x}_0). \tag{56}$$

Due to the Markov property of the forward diffusion process,

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

Therefore,

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \propto q(\mathbf{x}_t \mid \mathbf{x}_{t-1})\, q(\mathbf{x}_{t-1} \mid \mathbf{x}_0). \tag{57}$$

The two terms are Gaussian distributions:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}\right) \tag{58}$$
$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_{t-1}; \sqrt{\bar{\alpha}_{t-1}}\mathbf{x}_0, (1-\bar{\alpha}_{t-1})\mathbf{I}\right). \tag{59}$$

Multiplying these two Gaussians results in another Gaussian:

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

The posterior variance becomes

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

and the posterior mean is

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

Lastly, here is a list for the robot and AI related papers: https://www.notion.so/Paper-list-32195d8894488036b30ffb910
source=copy_link