

Two Unified Perspectives on DDPM

Zhijie Chen

October 25, 2026

In this article, we introduce two different perspectives on Denoising Diffusion Probabilistic Models (DDPM): one from stochastic analysis (continuous-time Markov chains, CTMC), and one from discrete-time Markov chains (used by the original DDPM paper [1]). The material for the former is drawn from Zhang [2].

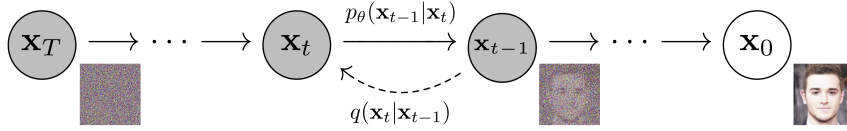


Figure 1: Forward diffusion and reverse denoising processes in DDPM [1].

DDPM consists of two stages. The forward process systematically destroys a complex data sample by gradually adding Gaussian noise to it. The reverse process learns to denoise a noisy sample step by step to recover a sample from the data distribution (e.g., the uniform distribution on a large image dataset).

1 The Stochastic Analysis Perspective

1.1 The forward process

Consider an Ornstein–Uhlenbeck (OU) process

$$dX_t = -X_t dt + \sqrt{2} dW_t, \quad (1)$$

where $(W_t)_{t \geq 0}$ is a standard Wiener process. Its solution is (by applying Itô's formula to $f(t, X_t) = e^t X_t$)

$$X_t \stackrel{d}{=} e^{-t} X_0 + \sqrt{1 - e^{-2t}} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (2)$$

As $t \rightarrow \infty$, X_t converges in distribution to the standard Gaussian distribution. Hence, we can use (2) to simulate the OU process and systematically destroy a data sample X_0 .

1.2 The reverse process

To learn the denoising process, consider the time-reversed process of (1)

$$Y_t = X_{T-t}, \quad 0 \leq t \leq T,$$

where T is chosen sufficiently large such that the law of X_T is close to the standard Gaussian distribution. A tedious calculation (see Zhang [3]) yields the SDE for the reverse process:

$$dY_t = (Y_t + 2\nabla \log \pi_{T-t}(Y_t)) dt + \sqrt{2} d\bar{W}_t, \quad (3)$$

where π_t is the density of X_t .

To simulate (3) (e.g., using Euler–Maruyama discretization), it suffices to know the value of $\nabla \log \pi_t(x)$ (known as the score function) for every t and x . However, this function is generally intractable to compute directly. So we approximate it using a neural network. The following formula makes it convenient to learn the score function.

Theorem 1 (Tweedie’s formula). *Let $Y = X + \sqrt{t} \cdot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ is independent of X . Let p_Y be the density of Y . Then*

$$\mathbb{E}[X \mid Y = y] = y + t \nabla \log p_Y(y).$$

Recall from (2) that

$$\underbrace{X_t}_Y = \underbrace{e^{-t} X_0}_X + \underbrace{\sqrt{1 - e^{-2t}}}_{\sqrt{t}} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

Since we can draw samples from the data distribution, we can construct (x_0, x_t) pairs from the forward process, and train the network to predict x_0 from x_t (and t). Use the ordinary MSE loss $\|x_0 - \hat{x}_0\|^2$. The solution to this easy optimization problem is

$$\hat{x}_0^*(x_t, t) = \mathbb{E}[x_0 \mid x_t].$$

Substituting it into Theorem 1 yields the score function.

2 The Discrete-Time Markov Chain Perspective

The original implementation of DDPM differs from the approach described above in several ways. In particular, it uses discrete-time Markov chains instead of Itô diffusions to model the forward and reverse processes.

2.1 Noise schedule

Instead of using a CTMC (OU process), the original DDPM selects a set of timesteps $(\tau_t)_{0 \leq t \leq T}$ and only considers the discrete-time Markov chain $x_t = X_{\tau_t}$, $0 \leq t \leq T$. Specifically, DDPM defines

$$0 < \alpha_t < 1, \quad \beta_t = 1 - \alpha_t, \quad \bar{\alpha}_t = \prod_{s=1}^t \alpha_s, \quad 1 \leq t \leq T,$$

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{\beta_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (4)$$

Or, equivalently,

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (5)$$

Set $\tau_t = -\frac{1}{2} \log \bar{\alpha}_t$ and we can see that (5) is completely equivalent to (2). Hence, the noise schedule is just about selecting a discrete set of timesteps.

The original DDPM paper sets $T = 1000$ and linearly increases β_t from 10^{-4} to 0.02. The smallness of β_t will be used in the asymptotic expansion below.

2.2 Prediction target

DDPM usually directly predicts the noise in (5) using a neural network $\epsilon_\theta(x_t, t)$. The ordinary MSE loss is used:

$$L_{\text{simple}} = \mathbb{E}_{x_0, t, \epsilon} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2], \quad t \sim \text{Uniform}\{1, \dots, T\}.$$

We show that this new prediction target is essentially just a reparameterization of score matching. Let $s_\theta(x_t, t)$ be the score estimator. The optimal noise predictor is $\epsilon^*(x_t, t) = \mathbb{E}[\epsilon \mid x_t]$. Take the conditional expectation of both sides of (5) given x_t , and combine it with Theorem 1.

$$s_\theta(x_t, t) = -\frac{\epsilon_\theta(x_t, t)}{\sqrt{1 - \bar{\alpha}_t}}, \quad \hat{x}_0 := \mathbb{E}[x_0 \mid x_t] = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}. \quad (6)$$

2.3 Gaussian posterior

We will need the Gaussian posterior $q(x_{t-1} \mid x_t, x_0)$ in what follows.

$$\begin{aligned} q(x_{t-1} \mid x_t, x_0) &= \frac{q(x_{t-1} \mid x_0)q(x_t \mid x_{t-1}, x_0)}{q(x_t \mid x_0)} \\ (\text{Markov property}) \quad &= \frac{q(x_{t-1} \mid x_0)q(x_t \mid x_{t-1})}{q(x_t \mid x_0)} \\ &= \phi\left(\frac{x_{t-1} - \sqrt{\bar{\alpha}_{t-1}}x_0}{\sqrt{1 - \bar{\alpha}_{t-1}}}\right) \phi\left(\frac{x_t - \sqrt{\bar{\alpha}_t}x_{t-1}}{\sqrt{1 - \bar{\alpha}_t}}\right) \phi\left(\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1 - \bar{\alpha}_t}}\right)^{-1} \\ (\text{by a tedious calculation}) \quad &= \mathcal{N}(\tilde{\mu}_t, \tilde{\beta}_t I), \end{aligned}$$

where ϕ is the standard Gaussian density, and

$$\tilde{\mu}_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t}x_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t}x_t, \quad \tilde{\beta}_t = \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t. \quad (7)$$

2.4 Sampling

2.4.1 Definition

DDPM defines¹

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 I),$$

where σ_t^2 usually equals β_t or $\tilde{\beta}_t$, and

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right). \quad (8)$$

Here $\mu_\theta(x_t, t)$ is obtained by substituting $\hat{x}_0$ for x_0 in (7) (note that we no longer condition on x_0). During sampling, we start with $x_T \sim \mathcal{N}(0, I)$ and sample

$$x_{t-1} = \mu_\theta(x_t, t) + \sigma_t z, \quad t = T, \dots, 1, \quad (9)$$

where we resample $z \sim \mathcal{N}(0, I)$ at each step, and set $z = 0$ at the final step $t = 1$.

Note that although we substitute $\hat{x}_0$ for x_0 to obtain $\mu_\theta(x_t, t)$, which seems inaccurate, it still holds that $\mu^*(x_t, t) = \mathbb{E}[x_{t-1} \mid x_t]$:

$$\mathbb{E}[x_{t-1} \mid x_t] = \mathbb{E}[\mathbb{E}[x_{t-1} \mid x_t, x_0] \mid x_t] = \mathbb{E}[\tilde{\mu}_t \mid x_t] = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1 - \bar{\alpha}_t} \hat{x}_0 + \frac{\sqrt{\bar{\alpha}_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t = \mu^*(x_t, t).$$

¹Note that although $q(x_{t-1} \mid x_t, x_0)$ is Gaussian, $q(x_{t-1} \mid x_t)$ in general is not.

2.4.2 Connection to the reverse SDE

We now show that the two perspectives are almost equivalent. Suppose $\sigma_t^2 = \beta_t$. Combining (9), (8), and (6) yields

$$\begin{aligned} x_{t-1} &= \frac{x_t + \beta_t s_\theta(x_t, t)}{\sqrt{1 - \beta_t}} + \sqrt{\beta_t} z \\ (\text{Taylor expand in } \beta_t) \quad &= x_t + \frac{\beta_t}{2} x_t + \beta_t s_\theta(x_t, t) + O(\beta_t^2) + \sqrt{\beta_t} z \\ (h := \beta_t/2) \quad &\approx x_t + (x_t + 2s_\theta(x_t, t))h + \sqrt{2h}z. \end{aligned}$$

This coincides with the Euler–Maruyama discretization of (3). Moreover, the step size h also matches that in the discretized forward process (4):

$$h_{\text{forward}} = \tau_t - \tau_{t-1} = -\frac{1}{2} \log(1 - \beta_t) \approx \frac{\beta_t}{2}.$$

2.5 Loss function

2.5.1 Reparameterization changes the loss weights across timesteps

Section 1.2 uses equally weighted original-image MSE, while now we use equally weighted noise MSE. (6) shows the difference between the two:

$$\|\epsilon - \epsilon_\theta\|^2 = \frac{\bar{\alpha}_t}{1 - \bar{\alpha}_t} \|x_0 - \hat{x}_0\|^2.$$

Therefore, noise MSE is the signal-to-noise ratio-weighted original-image MSE. The reparameterization changes the loss weights across timesteps and shifts the focus of training.

2.5.2 An ELBO-based loss function

We can use the evidence lower bound (ELBO) to derive an alternative loss function. To do this, define the model’s joint distribution $p_\theta(x_{0:T})$ and the forward distribution $q(x_{1:T} | x_0)$. They factorize as

$$p_\theta(x_{0:T}) = \phi(x_T) \prod_{t=1}^T p_\theta(x_{t-1} | x_t), \quad q(x_{1:T} | x_0) = q(x_T | x_0) \prod_{t=2}^T q(x_{t-1} | x_t, x_0). \quad (10)$$

Here we use the Markov property. Now ELBO gives

$$\begin{aligned} \log p_\theta(x_0) &\geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T} | x_0)} \right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{\phi(x_T)}{q(x_T | x_0)} + \sum_{t=2}^T \log \frac{p_\theta(x_{t-1} | x_t)}{q(x_{t-1} | x_t, x_0)} + \log p_\theta(x_0 | x_1) \right] \\ &= -L_{\text{VLB}}, \end{aligned}$$

where VLB stands for variational lower bound and

$$L_{\text{VLB}} = \mathbb{E} \left[\underbrace{D_{\text{KL}}(q(x_T | x_0) \parallel \phi(x_T))}_{L_T} + \sum_{t=2}^T \underbrace{D_{\text{KL}}(q(x_{t-1} | x_t, x_0) \parallel p_\theta(x_{t-1} | x_t))}_{L_{t-1}} - \underbrace{\log p_\theta(x_0 | x_1)}_{\text{reconstruction loss}} \right].$$

Let’s look at L_T first. Because

$$q(x_T | x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_T} x_0, (1 - \bar{\alpha}_T)I), \quad p(x_T) = \mathcal{N}(0, I),$$

L_T does not depend on θ and thus can be ignored during training.

Now consider L_{t-1} . We already know that

$$q(x_{t-1} \mid x_t, x_0) = \mathcal{N}(\tilde{\mu}_t, \tilde{\beta}_t I), \quad p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(\mu_\theta, \sigma_t^2 I).$$

Hence

$$D_{\text{KL}} = \frac{\|\tilde{\mu}_t - \mu_\theta\|^2}{2\sigma_t^2} + \underbrace{\frac{d}{2} \left(\frac{\tilde{\beta}_t}{\sigma_t^2} - 1 + \log \frac{\sigma_t^2}{\tilde{\beta}_t} \right)}_{\text{not depending on } \theta}.$$

$$L_{t-1} = \mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1 - \bar{\alpha}_t)} \|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right] + C_t.$$

Here d is the data dimension and C_t is a constant that does not depend on θ .

Now consider the reconstruction loss. Direct calculation yields

$$L_0 = \mathbb{E}_{x_0, \epsilon} \left[\frac{\beta_1}{2\sigma_1^2 \alpha_1} \|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right] + C.$$

Note that in the original DDPM implementation, L_{simple} is adopted instead of this fancier ELBO-based loss L_{VLB} , because the former just works better.

References

- [1] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.
- [2] Chihao Zhang. Advanced algorithms, lecture 7: Markov semigroup, langevin dynamics, and DDPMs, 2025. URL <https://chihaozhang.com/teaching/AA2025/lectures/lec7/lec7.html>. Lecture notes.
- [3] Chihao Zhang. Stochastic processes, lecture 14: Markov processes, langevin monte carlo, and diffusion models, 2025. URL <https://chihaozhang.com/teaching/SP2025/lec14.pdf>. Lecture notes.