Diffusion Network - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

Diffusion models construct a generative distribution over data by *inverting* a fixed noising (diffusion) process. A tractable *forward* Markov chain gradually destroys structure in a data sample by adding Gaussian noise. The *reverse* chain—parameterized by a neural network—is trained to denoise step-by-step, yielding samples from the data distribution via iterative refinement. We derive the model starting from a linear-Gaussian diffusion, show a variational (ELBO) training objective, prove closed-form posteriors for the forward process, and explain standard parameterizations (noise-, data-, and v-prediction) as well as the ancestral sampling procedure.

1.2 Data and Notation

Let the training set be $\mathcal{D} = \{\mathbf{x}_0^{(i)}\}_{i=1}^n$, with each $\mathbf{x}_0 \in \mathbb{R}^d$ (e.g. a flattened image). Fix a number of diffusion steps $T \in \mathbb{N}$ and a *variance schedule*

$$\beta_t \in (0,1), \quad \alpha_t \triangleq 1 - \beta_t, \quad \bar{\alpha}_t \triangleq \prod_{s=1}^{t} \alpha_s, \qquad t = 1, \dots, T.$$

We use $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$ for a Gaussian with mean $\boldsymbol{\mu}$ and covariance Σ, $\mathrm{KL}(\cdot\|\cdot)$ for Kullback–Leibler divergence, and bold symbols for vectors/tensors.

1.3 Model Formulation: Forward (Diffusion) Process

The forward (noising) chain q is a fixed Markov process that gradually corrupts $\mathbf{x}_0$:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) \;=\; \mathcal{N}\big(\sqrt{\alpha_t}\,\mathbf{x}_{t-1},\, \beta_t\mathbf{I}\big), \qquad t = 1, \dots, T. \tag{1}$$

By composition of linear Gaussians, we obtain a closed form "one-shot" marginal:

$$q(\mathbf{x}_t \mid \mathbf{x}_0) \;=\; \mathcal{N}\big(\sqrt{\bar\alpha_t}\,\mathbf{x}_0,\, (1-\bar\alpha_t)\mathbf{I}\big), \qquad \text{so } \; \mathbf{x}_t = \sqrt{\bar\alpha_t}\,\mathbf{x}_0 + \sqrt{1-\bar\alpha_t}\,\boldsymbol{\epsilon}, \;\; \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

Proof (by induction on t). *Proof.* For $t = 1$, Eqn. (2) equals Eqn. (1) with $\bar\alpha_1 = \alpha_1$. Assume $q(\mathbf{x}_{t-1} \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar\alpha_{t-1}}\mathbf{x}_0, (1-\bar\alpha_{t-1})\mathbf{I})$. Convolving with Eqn. (1) gives

$$\mathbb{E}[\mathbf{x}_t \mid \mathbf{x}_0] = \sqrt{\alpha_t}\,\sqrt{\bar\alpha_{t-1}}\mathbf{x}_0 = \sqrt{\bar\alpha_t}\mathbf{x}_0, \quad \mathrm{Var}(\mathbf{x}_t \mid \mathbf{x}_0) = \alpha_t(1-\bar\alpha_{t-1})\mathbf{I} + \beta_t\mathbf{I} = (1-\bar\alpha_t)\mathbf{I}.$$

Thus Eqn. (2) holds for t, proving the claim. ■

1.4 Reverse (Generative) Process

The generative model is a *time-reversal* of the diffusion:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{3}$$

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\big(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t),\, \sigma_t^2\mathbf{I}\big), \qquad t = T, \dots, 1, \tag{4}$$

where $\boldsymbol{\theta}$ are neural network parameters. Common choices fix $\sigma_t^2 \in \{\beta_t,\, \tilde\beta_t\}$ (see below) or learn a diagonal covariance.

1.5 Closed-Form Forward Posterior

Because the forward process is linear-Gaussian, the exact posterior is Gaussian:

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}\big(\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0),\, \tilde\beta_t\mathbf{I}\big), \tag{5}$$

with

$$\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) = \frac{\sqrt{\bar\alpha_{t-1}}\,\beta_t}{1-\bar\alpha_t}\,\mathbf{x}_0 \;+\; \frac{\sqrt{\alpha_t}\,(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}\,\mathbf{x}_t, \tag{6}$$

$$\tilde\beta_t = \frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}\,\beta_t. \tag{7}$$

Proof (Gaussian conditioning). *Proof.* From Eqn. (1) and Eqn. (2), the pair $(\mathbf{x}_{t-1}, \mathbf{x}_t) \mid \mathbf{x}_0$ is jointly Gaussian with known means/covariances. Conditioning a joint Gaussian yields Eqns. (5)–(7) by the standard formula $\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(b-\mu_b)$, $\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$. Algebra recovers the coefficients shown (details omitted for space). ■

1.6 Variational Objective (ELBO)

The goal is to maximize the log-likelihood of the data:

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

where the joint distribution is defined by:

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

The data log-likelihood satisfies a telescoping variational bound:

$$\begin{aligned}\log p_{\boldsymbol{\theta}}(\mathbf{x}_0) &= \log \mathbb{E}_{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\left[\frac{p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T}\mid\mathbf{x}_0)}\right] \\ &\geq \mathbb{E}_q\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}_T) + \sum_{t=1}^{T}\log p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}\mid\mathbf{x}_t) - \sum_{t=1}^{T}\log q(\mathbf{x}_t\mid\mathbf{x}_{t-1})\right] \triangleq -\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{\theta}),\end{aligned} \tag{8}$$

by Jensen's inequality. Rearranging into KL terms using $q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$ and the exact posterior Eqn. (5) gives

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_q\Big[\underbrace{\mathrm{KL}(q(\mathbf{x}_T\mid\mathbf{x}_0)\,\|\,p(\mathbf{x}_T))}_{\mathcal{L}_T} + \sum_{t=2}^{T}\underbrace{\mathrm{KL}(q(\mathbf{x}_{t-1}\mid\mathbf{x}_t,\mathbf{x}_0)\,\|\,p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}\mid\mathbf{x}_t))}_{\mathcal{L}_t} - \underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{x}_0\mid\mathbf{x}_1)}_{\mathcal{L}_1}\Big]. \tag{9}$$

If $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, then $\mathcal{L}_T$ is constant in $\boldsymbol{\theta}$. The training objective reduces to minimizing the sum of KL divergences $\mathcal{L}_t$ for $t \geq 2$ and the (optional) decoder term $\mathcal{L}_1$.

1.7 Parameterizations and the "Simple" Loss

With Eqn. (5) known in closed form, a natural parameterization is to set

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \;\equiv\; \tilde{\boldsymbol{\mu}}_t\big(\mathbf{x}_t, \widehat{\mathbf{x}}_0^{\boldsymbol{\theta}}(\mathbf{x}_t, t)\big), \tag{10}$$

i.e. predict a proxy $\widehat{\mathbf{x}}_0^{\boldsymbol{\theta}}$ and plug into the exact posterior mean. A convenient alternative is to predict the forward noise $\boldsymbol{\epsilon}$ in Eqn. (2):

$$\widehat{\mathbf{x}}_0^{\boldsymbol{\theta}}(\mathbf{x}_t, t) \;=\; \frac{1}{\sqrt{\bar{\alpha}_t}}\left(\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right). \tag{11}$$

Plugging Eqn. (11) into Eqn. (10) and choosing $\sigma_t^2 = \tilde{\beta}_t$ yields (after algebra)

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\right). \tag{12}$$

ELBO $\Rightarrow$ noise MSE. With Gaussian KLs, each $\mathcal{L}_t$ is a quadratic in the mean mismatch with weight $(2\sigma_t^2)^{-1}$. Under Eqn. (12) and $\sigma_t^2 = \tilde{\beta}_t$, the KL is proportional to $\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\|_2^2$ for $\mathbf{x}_t$ sampled via Eqn. (2). This motivates the widely used *simple loss*

$$\mathcal{L}_{\mathrm{simple}}(\boldsymbol{\theta}) \;=\; \mathbb{E}_{\mathbf{x}_0\sim\mathcal{D},\, t\sim\mathcal{U}\{1:T\},\, \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}\big(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon},\ t\big)\right\|_2^2, \tag{13}$$

which is a reweighted version of the ELBO. In practice, Eqn. (13) is used for training.

Score connection. The score of $q(\mathbf{x}_t \mid \mathbf{x}_0)$ equals

$$\nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t\mid\mathbf{x}_0) = -\frac{1}{1-\bar{\alpha}_t}\left(\mathbf{x}_t - \sqrt{\bar{\alpha}_t}\mathbf{x}_0\right) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon},$$

so learning $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is equivalent to learning the score up to a scalar: $s_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t}\log q(\mathbf{x}_t) \approx -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\,\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)$.

Other parameterizations. Besides ϵ-prediction, two equivalent forms are common:

$$x_0\textbf{-prediction:}\quad \widehat{\mathbf{x}}_0^{\boldsymbol{\theta}}(\mathbf{x}_t, t)\text{ directly; recover }\boldsymbol{\mu}_{\boldsymbol{\theta}}\text{ via Eqn. (10).} \tag{14}$$

$$v\textbf{-prediction:}\quad \mathbf{v} \triangleq \sqrt{\bar{\alpha}_t}\,\boldsymbol{\epsilon} - \sqrt{1-\bar{\alpha}_t}\,\mathbf{x}_0,\quad \mathbf{v}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\text{ predicted, with }\quad \widehat{\mathbf{x}}_0 = \frac{\sqrt{\bar{\alpha}_t}\mathbf{x}_t - \sqrt{1-\bar{\alpha}_t}\mathbf{v}_{\boldsymbol{\theta}}}{\bar{\alpha}_t}. \tag{15}$$

All are affine transforms of one another and induce equivalent samplers when combined with the exact posterior mean.

1.8 Sampling (Ancestral Denoising)

Given a trained network (e.g. $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$) and $\sigma_t^2 = \tilde{\beta}_t$, the ancestral sampler is:

$$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \text{for } t = T, \dots, 1:$$

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \, \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \right) \; + \; \mathbf{1}\{t > 1\} \sqrt{\tilde{\beta}_t} \, \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{16}$$

At $t = 1$, the final noise term is omitted, yielding $\mathbf{x}_0$.

1.9 Continuous-Time Limit and SDE View (Optional)

Let $t \in [0, 1]$ and choose a variance-preserving (VP) SDE

$$\mathrm{d}\mathbf{x} = -\tfrac{1}{2}\beta(t)\mathbf{x}\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\mathbf{W}_t, \tag{17}$$

whose time-discretization recovers Eqn. (1). Time-reversal yields the generative SDE

$$\mathrm{d}\mathbf{x} = \Big[-\tfrac{1}{2}\beta(t)\mathbf{x} - \beta(t)\nabla_{\mathbf{x}} \log q_t(\mathbf{x}) \Big] \mathrm{d}\bar{t} + \sqrt{\beta(t)}\,\mathrm{d}\bar{\mathbf{W}}_t, \tag{18}$$

where q_t is the forward marginal and $\bar{t}$ runs backward. Replacing the unknown score by $s_{\boldsymbol{\theta}}(\mathbf{x}, t)$ gives score-based sampling with predictor–corrector or ODE solvers; the discrete DDPM sampler Eqn. (16) is a particular Euler–Maruyama scheme.

1.10 Algorithm (Training and Sampling)

Training (noise-prediction).

1. Sample $\mathbf{x}_0 \sim \mathcal{D}$, $t \sim \mathcal{U}\{1{:}T\}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
2. Form $\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}$.
3. Minimize $\big\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) \big\|_2^2$ (optionally with per-t weights).

Sampling (ancestral).

1. Initialize $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
2. For $t = T, \dots, 1$, update $\mathbf{x}_{t-1}$ by Eqn. (16).
3. Output $\mathbf{x}_0$.

1.11 Summary of Variables and Their Dimensions

- $\mathbf{x}_0 \in \mathbb{R}^d$: data vector (e.g. image), d features/pixels.
- $T \in \mathbb{N}$: number of diffusion steps.
- $\beta_t \in (0, 1)$: variance schedule; $\alpha_t = 1 - \beta_t$; $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$.
- $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: standard Gaussian noise in the forward process.
- $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$: forward marginal.
- $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\tilde{\boldsymbol{\mu}}_t, \tilde{\beta}_t\mathbf{I})$: exact posterior, with $\tilde{\boldsymbol{\mu}}_t$ and $\tilde{\beta}_t$ given by Eqns. (6)–(7).
- $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \sigma_t^2\mathbf{I})$: learned reverse transition.
- $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot, t)$: neural network predicting forward noise (or, equivalently, score/$\mathbf{x}_0$/$\mathbf{v}$).
- $\mathcal{L}_{\text{ELBO}}$: variational training objective; $\mathcal{L}_{\text{simple}}$ in Eqn. (13) is its practical surrogate.

1.12 Summary

Starting from a linear-Gaussian diffusion q that admits the closed-form marginal Eqn. (2) and posterior Eqn. (5), the reverse Markov chain $p_{\boldsymbol{\theta}}$ is trained by maximizing a variational lower bound Eqn. (9). Parameterizing the reverse mean via a prediction of either $\mathbf{x}_0$ or the forward noise $\boldsymbol{\epsilon}$ leads to the efficient "simple" MSE loss Eqn. (13), which is equivalent (up to constants/weights) to minimizing the Gaussian KLs in the ELBO. Sampling proceeds by ancestral denoising Eqn. (16), beginning at $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and iteratively applying learned denoisers to produce $\mathbf{x}_0$. In the continuous-time limit, diffusion connects to score-based generative modeling via time-reversed SDEs, with the discrete DDPM sampler recovered as an Euler discretization.