Diffusion Models - Design
04/08/2026

Diffusion models — also called Denoising Diffusion Probabilistic Models or score-based generative models. Rather than encoding data into a latent space or training adversarially, diffusion models learn to reverse a gradual noising process. They have emerged as the dominant architecture for high-fidelity image synthesis, underpinning systems such as DALL·E 2, Stable Diffusion, Imagen, and Midjourney.

The central intuition is elegant: if one can learn to undo noise addition step by step, then starting from pure noise and repeatedly applying the learned denoising operation will eventually yield a sample from the data distribution.

Conceptual Framework: The Two Processes

Diffusion models are defined by two stochastic processes operating in opposite temporal directions:

1. The forward process q: A fixed (non-learned), Markovian process that progressively corrupts a data sample $\mathbf{x}_0$ by adding Gaussian noise over T discrete timesteps, eventually transforming it into approximately pure Gaussian noise $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

2. The reverse process p_θ: A learned process, parameterized by neural network weights θ, that attempts to recover a clean sample from noise by iteratively denoising from $\mathbf{x}_T$ back to $\mathbf{x}_0$.

> *Markov property*: A stochastic process has the Markov property if the future state depends only on the present state, not on the history of past states. Formally, $p(\mathbf{x}_t \mid \mathbf{x}_{t-1}, \mathbf{x}_{t-2}, \ldots, \mathbf{x}_0) = p(\mathbf{x}_t \mid \mathbf{x}_{t-1})$.
>
> *Gaussian distribution* $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$: A probability distribution fully characterized by its mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. The identity covariance $\mathbf{I}$ denotes independent, unit-variance dimensions — "isotropic" noise.

The Forward (Noising) Process

Given a data sample $\mathbf{x}_0 \sim q(\mathbf{x})$ drawn from the true data distribution, the forward process defines a sequence of noisy latent variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ via the transition kernel:

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

where $\{\beta_t\}_{t=1}^T$ is a noise schedule — a monotonically increasing $0 < \beta_1 < \beta_2 < \cdots < \beta_T < 1$ sequence that controls how rapidly noise is added at each step. Typical values range from $\beta_1 \approx 10^{-4}$ to $\beta_T \approx 0.02$.

The full joint distribution of the forward trajectory is:

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

The Reparameterisation Trick and Closed-Form Marginals. A key computational advantage of the Gaussian forward process is that the marginal distribution $q(\mathbf{x}_t \mid \mathbf{x}_0)$ — the distribution of $\mathbf{x}_t$ given the original clean image, bypassing all intermediate steps — can be expressed in closed form.

Define $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. Using the reparameterisation of *Gaussian* random variables — the fact that if $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then $\boldsymbol{\mu} + \boldsymbol{\sigma}\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$ — one can show by induction that:

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

Derivation sketch: At step 1:
$\mathbf{x}_1 = \sqrt{\alpha_1}\,\mathbf{x}_0 + \sqrt{1-\alpha_1}\,\boldsymbol{\epsilon}_1$

At step 2:
$\mathbf{x}_2 = \sqrt{\alpha_2}\,\mathbf{x}_1 + \sqrt{1-\alpha_2}\,\boldsymbol{\epsilon}_2 = \sqrt{\alpha_2\alpha_1}\,\mathbf{x}_0 + \sqrt{\alpha_2(1-\alpha_1)}\,\boldsymbol{\epsilon}_1 + \sqrt{1-\alpha_2}\,\boldsymbol{\epsilon}_2$

Since the sum of independent Gaussians is Gaussian, and since variances add:

$$\mathbf{x}_2 \sim \mathcal{N}(\sqrt{\alpha_1\alpha_2}\,\mathbf{x}_0,\ (1-\alpha_1\alpha_2)\mathbf{I}) = \mathcal{N}\big(\sqrt{\bar{\alpha}_2}\,\mathbf{x}_0,\ (1-\bar{\alpha}_2)\mathbf{I}\big)$$

Proceeding by induction to step t yields the general result. This means that to generate a noisy version of $\mathbf{x}_0$ at any arbitrary timestep t, one simply samples:

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

This is computationally crucial: training does not require iterating through all T steps; any $(\mathbf{x}_0, \mathbf{x}_t)$ pair can be generated in $\mathcal{O}(1)$ time.

As $t \to T$, $\bar{\alpha}_T \to 0$ and the marginal approaches:

$$q(\mathbf{x}_T \mid \mathbf{x}_0) \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$$

i.e., the signal is completely destroyed and $\mathbf{x}_T$ becomes indistinguishable from pure *Gaussian* noise. This is the starting point for generation.

The Reverse (Denoising) Process

If the true data distribution $q(\mathbf{x}_0)$ were known, the reverse conditional distribution $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ would be well-defined via Bayes' theorem:

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

However, this requires knowledge of the marginal $q(\mathbf{x}_{t-1})$, which involves integrating over all possible data points — a high-dimensional integral that is computationally intractable.

Key theoretical insight (Feller, 1949; Anderson, 1982): When β_t is small and T is large, the reverse process $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is also approximately *Gaussian*. This permits the reverse process to be approximated by a learned *Gaussian*:

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

The parameters $\boldsymbol{\mu}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are outputs of a neural network conditioned on both the current noisy image $\mathbf{x}_t$ and the timestep t.

While $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ is intractable, the conditioned reverse $q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0)$ is tractable, because we can apply *Bayes*' theorem using the closed-form forward marginals:

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

where:

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

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

This result follows from computing the product of *Gaussian* densities in the *Bayes* expansion and completing the square. It tells us: if we knew $\mathbf{x}_0$, the optimal denoising step would be given by a linear combination of $\mathbf{x}_t$ and $\mathbf{x}_0$ with analytically known coefficients.

Training Objective: The Evidence Lower Bound

Diffusion models are trained by maximising the log-likelihood of the data under the model:

$$\max_\theta \mathbb{E}_{q(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0)]$$

Since $p_\theta(\mathbf{x}_0)$ requires marginalising over all latent trajectories — again intractable — one instead maximises a variational lower bound, known as the Evidence Lower Bound (ELBO). The ELBO follows from *Jensen's* inequality applied to the concave $\log$ function:

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_q\left[\log \frac{p_\theta(\mathbf{x}_{0:T})}{q(\mathbf{x}_{1:T} \mid \mathbf{x}_0)}\right] =: \mathcal{L}_{\text{ELBO}}$$

> *Jensen's inequality*: For a concave function *f* and random variable X : $f(\mathbb{E}[X]) \geq \mathbb{E}[f(X)]$.
>
> *KL divergence* $D_{\text{KL}}(q\|p)$: A measure of how much probability distribution q differs from p, defined as $D_{\text{KL}}(q\|p) = \mathbb{E}_q[\log q - \log p] \geq 0$, with equality iff $q = p$.

After algebraic manipulation, the ELBO decomposes into a sum of KL divergences between the learned reverse steps and the tractable posterior:

$$-\mathcal{L}_{\text{ELBO}} = \underbrace{D_{\text{KL}}(q(\mathbf{x}_T \mid \mathbf{x}_0) \,\|\, p(\mathbf{x}_T))}_{\mathcal{L}_T\ \text{(prior matching)}} + \sum_{t=2}^{T} \underbrace{D_{\text{KL}}(q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0) \,\|\, p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t))}_{\mathcal{L}_{t-1}\ \text{(denoising matching)}} - \underbrace{\mathbb{E}_q[\log p_\theta(\mathbf{x}_0 \mid \mathbf{x}_1)]}_{\mathcal{L}_0\ \text{(reconstruction)}}$$

The dominant terms are the denoising matching terms $\mathcal{L}_{t-1}$, which penalise discrepancy between the learned reverse transitions and the optimal (tractable) posteriors.

Ho et al. (2020) introduced a crucial simplification: instead of directly parameterising $\boldsymbol{\mu}_\theta$, one trains a neural network $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ to predict the noise $\boldsymbol{\epsilon}$ that was added to $\mathbf{x}_0$ to produce $\mathbf{x}_t$. Given this, the predicted mean is:

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

The denoising matching terms then reduce — after dropping weighting constants shown empirically to stabilise training — to a remarkably simple mean squared error objective:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{t,\,\mathbf{x}_0,\,\boldsymbol{\epsilon}}\left[\left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\left(\sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\epsilon},\ t\right)\right\|^2\right]$$

This is the core training objective: at each step, sample a random timestep t, add the corresponding amount of noise to a training image, and train the network to predict what noise was added. The procedure is summarised below.

Training and Sampling Algorithms

Algorithm 1: DDPM Training

---

repeat
  x_0 ~ q(x_0) ▷ Sample from dataset
  t ~ Uniform({1, …, T}) ▷ Random timestep
  ε ~ $\mathcal{N}$(0, I) ▷ Sample noise
  Take gradient step on:
    $\nabla_\theta \|\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}\,x_0 + \sqrt{1-\bar{\alpha}_t}\,\varepsilon, t)\|^2$
until converged

Algorithm 2: DDPM Sampling

---

x_t ~ $\mathcal{N}$(0, I)
for t = T, T−1, …, 1 do
  z ~ $\mathcal{N}$(0, I) if t > 1, else z = 0
  $x_{t-1} = (1/\sqrt{\alpha_t})(x_t - \beta_t/\sqrt{1-\bar{\alpha}_t}\cdot\varepsilon_\theta(x_t, t)) + \sqrt{\tilde{\beta}_t}\cdot z$
end for
return x_0

The sampling process involves T sequential neural network evaluations, which for typical $T = 1000$ is computationally expensive — a key motivation for accelerated samplers.

Continuous-Time Formulation: Stochastic Differential Equations

Song et al. (2020) unified discrete diffusion models and score-based generative models by recasting the forward process as a Stochastic Differential Equation (SDE):

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\,dt + g(t)\,d\mathbf{w}$$

where:
- $\mathbf{f}(\mathbf{x}, t)$ is the drift coefficient — a deterministic vector field guiding the mean trajectory;
- $g(t)$ is the diffusion coefficient — a scalar controlling noise amplitude;
- $d\mathbf{w}$ denotes the increment of a standard *Wiener* process (*Brownian* motion), with $\mathbf{w}(t) - \mathbf{w}(s) \sim \mathcal{N}(\mathbf{0}, (t-s)\mathbf{I})$ for $t > s$.

> *Wiener process (Brownian motion)*: A continuous-time stochastic process $\mathbf{w}(t)$ with independent, normally distributed increments. It is the continuous-time limit of a random walk.

For the Variance Preserving (VP-SDE) case (corresponding to DDPM), the drift and diffusion take the form:

$$\mathbf{f}(\mathbf{x}, t) = -\tfrac{1}{2}\beta(t)\mathbf{x}, \qquad g(t) = \sqrt{\beta(t)}$$

where $\beta(t)$ is a continuous noise schedule function interpolating the discrete $\{\beta_t\}$.

A foundational result (*Anderson*, 1982) states that the reverse of any *Itô* SDE is also an *Itô* SDE. Specifically, the reverse of above equatoin is:

$$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] dt + g(t)\, d\bar{\mathbf{w}}$$

where $d\bar{\mathbf{w}}$ is a reverse-time *Wiener* process and $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ is the score function of the marginal distribution at time t.

> *Score function*: The gradient of the log-probability density with respect to the data: $\mathbf{s}(\mathbf{x}) := \nabla_{\mathbf{x}} \log p(\mathbf{x})$. Intuitively, it points in the direction of increasing probability mass — towards the heart of the data distribution.

Equation above is the mathematical justification for the entire diffusion model paradigm: to reverse the noising process, one needs only the score function $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. A neural network $\mathbf{s}_\theta(\mathbf{x}, t)$ trained to approximate this score enables generation via numerical SDE integration.

The connection between score estimation and the noise-prediction objective is made precise by *Tweedie's* formula, which relates the score function to the optimal denoising estimate:

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

Therefore:

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) = -\frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)}{\sqrt{1 - \bar{\alpha}_t}}$$

This demonstrates that predicting noise is equivalent to estimating the score function. The two formalisms — DDPM and score-based models — are mathematically unified through this identity.

Concluding Remarks

Diffusion models represent a mathematically coherent and empirically powerful framework for generative modeling, grounded in stochastic process theory, variational inference, and score matching. Their principal intellectual contributions are:

1. The reformulation of generation as iterative denoising, replacing the adversarial training instability of GANs with a stable mean-squared-error objective;
2. The unification of discrete-time and continuous-time perspectives through the SDE framework, enabling principled accelerated sampling;
3. The demonstration that a learned score function is sufficient to characterize and sample from complex high-dimensional distributions.

The mathematical thread from the closed-form marginals of the forward process, through the tractable posterior, to the simplified noise-prediction objective, and finally to Anderson's reverse-time SDE constitutes one of the more elegant derivational arcs in contemporary machine learning.