Introduction to Diffusion Models

DDPM, DDIM, the SDE formulation

Dario Shariatian, Giovanni Conforti

March 17, 2025

Introduction

Introduction - Diffusion Models

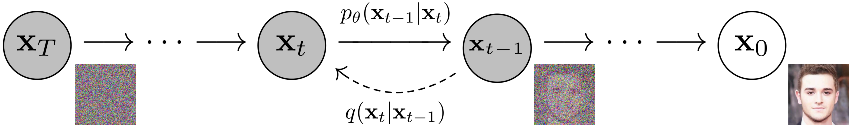

Figure: Forward/backward structure, discrete time [HJA20]

Introduction - Overall Structure

- **Forward process** Define $\{X_t\}_{t=0}^T$ (discrete) or $\{X_t\}_{0\leqslant t\leqslant T}$ (continuous), such that

$$X_0 \sim p_0\ , \quad X_T \sim p_T\ , \tag{1}$$

  where p_0 is the data distribution, and $p_T \approx \mathcal{N}(0, \mathrm{I}_d)$.

- **Backward process** Find corresponding $\{\bar{X}_t\}_{t=0}^T$, such that

$$\bar{X}_0 \sim p_T\ , \quad \bar{X}_t \sim X_{T-t}\ . \tag{2}$$

  Classically, it is characterized by a quantity of interest (e.g., the score $\nabla_x \log p_t(x)$).

- **Generative process** Sample $\{\bar{X}_t^\theta\}_{t=0}^T$, classically a Markov chain approximating the backward process:

$$\bar{X}_0^\theta \sim \mathcal{N}\left(0, \mathrm{I}_d\right)\ , \quad p_{t+1|t}^\theta \approx \bar{p}_{t+1|t}\ , \tag{3}$$

  where $\theta \in \Theta$ parametrizes a family of functions. Then:

$$\bar{X}_T^\theta \sim p_T^\theta \approx p_0\ . \tag{4}$$

Introduction - Further Notations

- p_t is the marginal distribution of X_t
- $p_{t|0}$ is the conditional distribution of X_t given X_0
- $p_{t+1|t}$ is the conditional distribution of X_{t+1} given X_t

Other notations will be easily inferred from this terminology. In particular,

- $\bar{p}$ refers to the backward process $\bar{X}$
- p^θ refers to the generative process $\bar{X}^\theta$

Unofficial convention: running backward in time In most paper, for convenience, the backward and generative process are run backward in time; for instance we write

$$\bar{X}_T^\theta \sim \mathcal{N}(0, \mathrm{I}_d) \ , \quad \bar{X}_0^\theta \sim p_0^\theta \approx p_0 \ . \tag{5}$$

Indeed we get $p_t = \bar{p}_t \approx p_t^\theta$, which makes equations more readable. We will keep this convention.

Introduction - Roadmap

- DDPM (Denoising Diffusion Probabilistic Models) [HJA20]

- DDIM (Denoising Diffusion Implicit Models) [SME20]

- SDEs (Score-Based Generative Modeling through SDEs) [Son+21]

Denoising Diffusion Probabilistic Models (DDPM)

DDPM – Overview

Setup (discrete time):

- $\{X_t\}_{t=0}^{T}$ is a Markov chain with Gaussian transition kernels $p_t(\cdot|\cdot)$
- $X_0 \sim p_0$ (the data), $X_T \sim p_T \approx \mathcal{N}(0, \mathrm{I}_d)$ (the noise)
- The generative process $\{\bar{X}_t^\theta\}_{t=0}^{T}$ will be a Markov chain running in reverse time, with a structured inherited from the true backward process.
- We fit the joint distributions of the two processes with an ELBO loss, like in VAEs.

DDPM – Forward Process

- **Forward process (Markov chain):**

$$X_{t+1} = \sqrt{\alpha_t} X_t + \sqrt{1-\alpha_t}\epsilon_t \ , \tag{6}$$

  where $\{\alpha_t\}_{t=0}^{T-1}$ is a noise schedule, $0 < \alpha_t < 1$.

- **Closed form for $X_t \mid X_0$**, by stability of the Gaussian distribution:

$$p_t(\cdot \mid x_0) \ = \ \mathcal{N}\left(\cdot\ ; \sqrt{\overline{\alpha}_t}\, x_0, \, (1-\overline{\alpha}_t)\mathrm{I}_d\right), \tag{7}$$

  with $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. For large T, X_T is approximately distributed as $\mathcal{N}(0, \mathrm{I}_d)$.

DDPM – Backward Process

- **Reformulating the forward process** Let us examine its joint distribution

$$
\begin{aligned}
p(x_0, \cdots, x_T) &= p_0(x_0) \cdot \prod_{t=1}^{T} p_{t|t-1}(x_t|x_{t-1}) \\
&= p_0(x_0) \cdot p_{1|0}(x_1|x_0) \cdot \prod_{t=2}^{T} p_{t|t-1}(x_t|x_{t-1}, x_0) \\
&= p_0(x_0) \cdot p_{1|0}(x_1|x_0) \cdot \prod_{t=2}^{T} \frac{p_{t-1|t,0}(x_{t-1}|x_t, x_0) p_{t|0}(x_t|x_0)}{p_{t-1|0}(x_{t-1}|x_0)} \quad \text{, by Bayes rule} \\
&= \underbrace{p_0(x_0)}_{\text{data}} \cdot \underbrace{p_{T|0}(x_T|x_0)}_{\text{noise}} \cdot \prod_{t=2}^{T} \underbrace{p_{t-1|t,0}(x_{t-1}|x_t, x_0)}_{\text{Gaussian transitions}}
\end{aligned}
$$

- **Gaussian transitions** $p_{t-1|t,0}(\cdot|x_t, x_0)$ is the density of the Gaussian distribution $\mathcal{N}(\tilde{\mathtt{m}}_t(x_t, x_0), \tilde{\Sigma}_t)$.

DDPM – Backward Process

Gaussian transitions Again, by Bayes rule:

$$
\begin{aligned}
p_{t-1|t,0}(x_{t-1}|x_t, x_0) &= \frac{p_{t|t-1}(x_t|x_{t-1}, x_0)p_{t-1|0}(x_{t-1}|x_0)}{p_{t|0}(x_t|x_0)} \\
&= \frac{p_{t|t-1}(x_t|x_{t-1})p_{t-1|0}(x_{t-1}|x_0)}{p_{t|0}(x_t|x_0)} \\
&\propto \exp\left(-\frac{\|x_t - \sqrt{\alpha_t}x_{t-1}\|^2}{2(1-\alpha_t)} - \frac{\|x_{t-1} - \sqrt{\bar\alpha_{t-1}}x_0\|^2}{2(1-\bar\alpha_{t-1})} - \frac{\|x_t - \sqrt{\bar\alpha_t}x_0\|^2}{2(1-\bar\alpha_t)}\right) \\
&\propto \cdots \\
&\propto \exp\left(-\frac{\|x_{t-1} - \tilde{\mathtt{m}}_t(x_t, x_0)\|^2}{2\tilde\Sigma_t}\right)
\end{aligned}
$$

with

$$
\tilde{\mathtt{m}}_t(x_t, x_0) = \frac{\sqrt{\bar\alpha_{t-1}}(1-\alpha_t)}{1-\bar\alpha_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar\alpha_{t-1})}{1-\bar\alpha_t}x_t \quad \text{and} \quad \tilde\Sigma_t = \frac{1-\bar\alpha_{t-1}}{1-\bar\alpha_t}(1-\alpha_t)\,. \tag{8}
$$

DDPM – Generative Process

- **Backward process**

$$p(x_0, \cdots, x_t) = \underbrace{p_0(x_0)}_{\text{data}} \cdot \underbrace{p_{T|0}(x_T|x_0)}_{\text{noise}} \cdot \prod_{t=2}^{T} \underbrace{p_{t-1|t,0}(x_{t-1}|x_t, x_0)}_{\text{Gaussian transitions}} , \tag{9}$$

where $p_{t-1|t,0}(\cdot|x_t, x_0) = \mathcal{N}(\cdot\ ;\ \tilde{\mathtt{m}}_t(x_t, x_0), \tilde{\Sigma}_t)$. In other words, with $\{\epsilon_t\}_{t=0}^{T} \sim \mathcal{N}(0, \mathrm{I}_d)$ i.i.d. :

$$\bar{X}_{t-1} = \tilde{\mathtt{m}}_t(\bar{X}_t, \bar{X}_0)\bar{X}_t + \tilde{\Sigma}_t^{1/2}\epsilon_{t-1} , \quad \bar{X}_0 \sim p_0, \quad \bar{X}_T = \sqrt{\bar{\alpha}_T}\bar{X}_0 + \sqrt{1-\bar{\alpha}_T}\epsilon_T . \tag{10}$$

- **Generative process** This suggests using the following structure for the generative model

$$p^{\theta}(x_0, \cdots, x_t) = \underbrace{p_T^{\theta}(x_T)}_{\text{noise}} \cdot \prod_{t=1}^{T} \underbrace{p_{t-1|t}^{\theta}(x_{t-1}|x_t)}_{\text{Gaussian transitions}} , \tag{11}$$

with $p_{t-1|t}^{\theta}(\cdot|x_t) = \mathcal{N}(\cdot\ ;\ \hat{\mathtt{m}}_t^{\theta}(x_t), \tilde{\Sigma}_t)$. In other words,

$$\bar{X}_{t-1}^{\theta} = \hat{\mathtt{m}}_t^{\theta}(\bar{X}_t^{\theta}) + \tilde{\Sigma}_t^{1/2}\epsilon_{t-1} , \quad \bar{X}_T^{\theta} = \epsilon_T , \tag{12}$$

with $\{\epsilon_t\}_{t=0}^{T} \sim \mathcal{N}(0, \mathrm{I}_d)$ i.i.d.

DDPM – Training Objective

- **Variational bound (ELBO)** We want to fit p^θ to p:

$$\begin{aligned}\log p_\theta(x_0) &= \log\left(\int p^\theta(X_{0:T})dX_{1:T}\right)\\ &\geqslant \log\left(\mathbb{E}_{p(X_{1:T}|x_0)}\frac{p^\theta(X_{0:T})}{p(X_{1:T}|X_0)}\right)\\ &\geqslant \mathbb{E}_{p(X_{1:T})}\log\left(\frac{p^\theta(X_{0:T})}{p(X_{1:T}|X_0)}\right) \quad \text{By Jensen's ineq.}\\ &= -\mathcal{L}_{\mathrm{ELBO}}(\theta)\end{aligned}$$

Rearranging terms, we obtain

$$\mathcal{L}_{\mathrm{ELBO}}(\theta) = \mathbb{E}\left[\underbrace{\mathrm{KL}(p_{T|0}(\cdot|X_0) \parallel p^\theta_T(\cdot))}_{L_T} + \sum_{t=2}^{T}\underbrace{\mathrm{KL}(p_{t-1|t,0}(\cdot|X_t, X_0) \parallel p^\theta_{t-1|t}(\cdot|X_t))}_{L_{t-1}} \underbrace{- \log p^\theta_{0|1}(X_0|X_1)}_{L_0}\right] .$$

The terms L_T, L_0 are typically neglected.

DDPM – Training Objective

- **Analytical formula for L_{t-1}** KL between Gaussian distribution of equal variance $\tilde{\Sigma}_t$:

$$L_{t-1} = \frac{\|\tilde{\mathtt{m}}_t(X_t, X_0) - \hat{\mathtt{m}}_t^\theta(X_t)\|^2}{2\tilde{\Sigma}_t} .$$

- **ELBO loss**

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{\|\tilde{\mathtt{m}}_t(X_t, X_0) - \hat{\mathtt{m}}_t^\theta(X_t)\|^2}{2\tilde{\Sigma}_t}\right] , \tag{13}$$

  with a choice of time distribution w (e.g., uniform, log-normal...).

- **Denoiser reparameterization** X_t is sampled from $p_{t|0}$ as $X_t = \sqrt{\bar{\alpha}_t}X_0 + \sqrt{1-\bar{\alpha}_t}\bar{\epsilon}_t$, with $\bar{\epsilon}_t \sim \mathcal{N}(0, \mathrm{I}_d)$, so we rewrite

$$\tilde{\mathtt{m}}_t(x_t, \bar{\epsilon}_t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\bar{\epsilon}_t\right) , \quad \hat{\mathtt{m}}_t^\theta(x_t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\hat{\epsilon}_t^\theta\right) , \tag{14}$$

  And, instead of optimizing the real ELBO, we optimize:

$$\mathcal{L}_{\mathrm{simple}}(\theta) = \mathbb{E}\left[\|\bar{\epsilon}_t - \hat{\epsilon}_t^\theta(X_t)\|^2\right]. \tag{15}$$

- **Interpretation:** We learn to predict (or remove) the noise added at each step.

DDPM – Recap

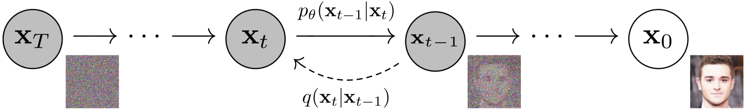

Figure: Forward/generative processes [HJA20]

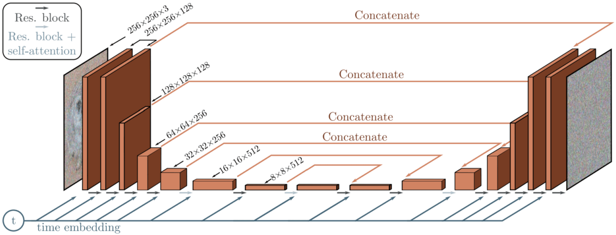

Figure: U-net architecture used for $\hat{\epsilon}_t^\theta$, predicting noise at each timestep [Sin23]

Denoising Diffusion Implicit Models (DDIM)

DDIM

- **Directly define the backward** Remark that, for DDPM, we did not need the forward process to be Markovian, and only benefited from the following expression:

$$p(x_0, \cdots, x_t) = \underbrace{p_0(x_0)}_{\text{data}} \cdot \underbrace{p_{T|0}(x_T|x_0)}_{\text{noise}} \cdot \prod_{t=2}^{T} \underbrace{p_{t-1|t,0}(x_{t-1}|x_t, x_0)}_{\text{Gaussian transitions}} . \tag{16}$$

  This time, we will just come up with a process defined as above.

- **Non-necessarily Markovian process**

$$\bar{X}_0 \sim p_0 , \quad \bar{X}_T|\bar{X}_0 \sim \mathcal{N}(\sqrt{\bar{\alpha}_T}\bar{X}_0, (1-\bar{\alpha}_T)\mathrm{I}_d) , \tag{17}$$

  and

$$\bar{X}_{t-1} = \underbrace{\sqrt{\bar{\alpha}_{t-1}}\bar{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \frac{\bar{X}_t - \sqrt{\bar{\alpha}_t}\bar{X}_0}{\sqrt{1-\bar{\alpha}_t}}}_{\tilde{\mathtt{m}}_t(\bar{X}_t, \bar{X}_0)} + \sigma_t \epsilon_t , \tag{18}$$

  with $\{\epsilon_t\}_{t=1}^{T} \sim \mathcal{N}(0, \mathrm{I}_d)$ i.i.d..

DDIM

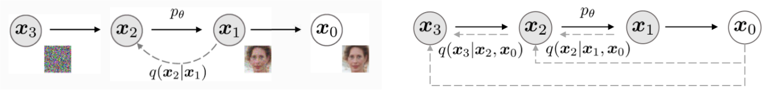

Figure: Non-Markovian forward process [SME20]

Distribution of $X_t|X_0$ Same as DDPM. Informal proof:

$$\bar{X}_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\bar{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \underbrace{\frac{\bar{X}_t - \sqrt{\bar{\alpha}_t}\bar{X}_0}{\sqrt{1-\bar{\alpha}_t}}}_{\text{noise term at time } t} + \sigma_t \epsilon_t$$

$$\stackrel{d}{=} \sqrt{\bar{\alpha}_{t-1}}\bar{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2+\sigma_t^2}\hat{\epsilon}_t\,, \quad \hat{\epsilon}_t \sim \mathcal{N}(0, \mathrm{I}_d) \quad \text{(Stability of Gaussian)}$$

$$\stackrel{d}{=} \sqrt{\bar{\alpha}_{t-1}}\bar{X}_0 + \sqrt{1-\bar{\alpha}_{t-1}}\hat{\epsilon}_t\,.$$

DDIM – Generative process

- **Generative process** Exactly the same as for DDPM:

$$p^\theta(x_0, \cdots, x_t) = \underbrace{p^\theta_T(x_T)}_{\text{noise}} \cdot \prod_{t=1}^{T} \underbrace{p^\theta_{t-1|t}(x_{t-1}|x_t)}_{\text{Gaussian transitions}} , \tag{19}$$

with $p^\theta_{t-1|t}(\cdot|x_t) = \mathcal{N}(\cdot\ ;\ \hat{\mathtt{m}}^\theta_t(x_t), \sigma_t^2 \mathrm{I}_d)$. In other words,

$$\bar{X}^\theta_{t-1} = \hat{\mathtt{m}}^\theta_t(\bar{X}^\theta_t) + \sigma_t \epsilon_{t-1} , \quad \bar{X}^\theta_T = \epsilon_T , \tag{20}$$

with $\{\epsilon_t\}_{t=0}^T \sim \mathcal{N}(0, \mathrm{I}_d)$ i.i.d.

- **Deterministic generation** when $\sigma_t = 0$ for all t.

DDIM – Training Objective

- **ELBO loss**

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{\|\tilde{\mathtt{m}}_t(\bar{X}_t, X_0) - \hat{\mathtt{m}}_t^\theta(\bar{X}_t)\|^2}{2\sigma_t^2}\right] , \tag{21}$$

with a choice of time distribution w (e.g., uniform, log-normal...).

- **Denoiser-reparameterization** With

$$\tilde{\mathtt{m}}_t(x_t) = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \hat{\epsilon}_t^\theta(x_t) , \tag{22}$$

we optimize

$$\mathcal{L}_{\text{simple}}(\theta) = \mathbb{E}\left[\|\hat{\epsilon}_t - \hat{\epsilon}_t^\theta(\sqrt{\bar{\alpha}_{t-1}}\bar{X}_0 + \sqrt{1 - \bar{\alpha}_{t-1}}\hat{\epsilon}_t)\|^2\right]. \tag{23}$$

Same loss and same neural network, but with deterministic generation

	Bedroom (256 × 256)				Church (256 × 256)			
dim(τ)	10	20	50	100	10	20	50	100
DDIM ($\eta = 0.0$)	**16.95**	**8.89**	**6.75**	**6.62**	**19.45**	**12.47**	**10.84**	10.58
DDPM ($\eta = 1.0$)	42.78	22.77	10.81	6.81	51.56	23.37	11.16	**8.27**

Figure: FID↓ for LSUN datasets. dim(τ) is the number of reverse steps/network calls [SME20]

SDE Formulation

SDE Formulation – Overview

Key Ideas

- Diffusion models can be viewed as discretizations of continuous-time stochastic processes.
- The forward process is described by a Stochastic Differential Equation (SDE), and the reverse process is characterized by a reverse-time SDE.
- This formulation unifies DDPM, DDIM, and other diffusion models under a single framework.

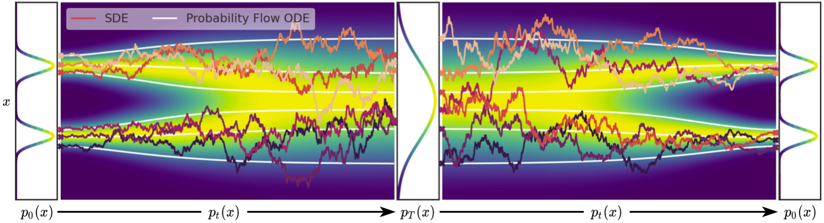

Figure: Forward/Backward Diffusion (SDE) [Son+21]

SDE Formulation – Forward Process

- **Forward process** Defined by an SDE of the form:

$$\mathrm{d}X_t = \mu_t(X_t)\mathrm{d}t + \sigma_t \mathrm{d}W_t \ , \tag{24}$$

where:

- X_t is the state at time $t \in [0, T]$.
- μ_t is the drift term.
- σ_t is the diffusion coefficient.
- W_t is a standard Wiener process (Brownian motion).

SDE Formulation – Forward Process

- **Variance Preserving (VP) SDE**. The VP-SDE is the continuous-time counterpart of the discrete Ornstein-Uhlenbeck process used in DDPM:

$$\mathrm{d}X_t = -\frac{1}{2}\beta_t X_t \mathrm{d}t + \sqrt{\beta_t}\mathrm{d}W_t \; , \tag{25}$$

where $\beta_t = 1 - \alpha_t$ is the noise schedule. At $t = T$, $X_T \sim \mathcal{N}(0, \mathrm{I}_d)$. Indeed, with a Euler scheme using time discretization steps h_t at time t:

$$\begin{aligned}
X_{t+h_t} &= X_t - \frac{1}{2}\beta_t h_t X_t \mathrm{d}t + \sqrt{\beta_t}\sqrt{h_t}\epsilon_t \; , \quad \epsilon_t \sim \mathcal{N}(0, \mathrm{I}_d) \\
X_{t+h_t} &= (1 - \frac{1}{2}\beta_t h_t) X_t \mathrm{d}t + \sqrt{\beta_t h_t}\epsilon_t \\
X_{t+h_t} &\approx \sqrt{1 - \beta_t h_t} X_t \mathrm{d}t + \sqrt{\beta_t h_t}\epsilon_t \; .
\end{aligned}$$

So we find the DDPM forward process after applying the map $(\beta_t, t) \mapsto \beta_t h_t$.

- **Variance Exploding (VE) SDE**. The VE-SDE is inspired by prior score-matching approaches using Langevin dynamics. In this case:

$$\mathrm{d}X_t = \sigma_t \mathrm{d}W_t \; . \tag{26}$$

SDE Formulation – Backward Process

- **Reverse-time SDE** A reverse-time SDE yields a stochastic backward process:

$$\mathrm{d}\bar{X}_t = \left[\mu_t(\bar{X}_t) - \sigma_t^2 \nabla_x \log p_t(\bar{X}_t)\right] dt + \sigma_t \mathrm{d}W_t \ , \tag{27}$$

  where $\nabla_x \log p_t(\bar{X}_t)$ is the **score function** of the marginal distribution p_t.

- **Reverse-time ODE** A reverse-time ODE yields a deterministic backward process:

$$\mathrm{d}\bar{X}_t = \left[\mu_t(\bar{X}_t) - \frac{1}{2}\sigma_t^2 \nabla_x \log p_t(\bar{X}_t)\right] \mathrm{d}t \ . \tag{28}$$

The score function is key to reversing the diffusion process.

SDE Formulation – Score-Matching

The score function is approximated by a neural network $s_\theta(x, t) \approx \nabla_x \log p_t(x)$.

- **Score-matching objective**

$$\mathcal{L}_{\mathrm{SM}}(\theta) = \mathbb{E}_{t,x_t}\left[\left\|s_\theta(x_t, t) - \nabla_x \log p_t(x_t)\right\|^2\right] , \tag{29}$$

  but the true score is not available.

- **Denoising score-matching** In practice, we use denoising score matching, which is an equivalent objective function:

$$\mathcal{L}_{\mathrm{DSM}}(\theta) = \mathbb{E}\left[\left\|s_\theta(X_t, t) - \nabla_{x_t} \log p_{t|0}(X_t|X_0)\right\|^2\right] . \tag{30}$$

  A proof is in [Vin11].

SDE Formulation – Denoising Score-Matching for the VP-SDE

- **Denoising loss for the VP-SDE** the transition kernel $p_{t|0}(x_t|x_0)$ is Gaussian:

$$p_{t|0}(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathrm{I}_d) \,, \tag{31}$$

where $\bar{\alpha}_t = \exp(-\int_0^t \beta_s ds)$. Thus:

$$\nabla_{x_t} \log p_{t|0}(x_t|x_0) = -\frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{1-\bar{\alpha}_t} \,. \tag{32}$$

Let $\epsilon_t = \frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1-\bar{\alpha}_t}}$ be the quantity corresponding to the noise term added at timestep t. Then:

$$\nabla_{x_t} \log p_{t|0}(x_t|x_0) = -\frac{\epsilon_t}{\sqrt{1-\bar{\alpha}_t}} \,. \tag{33}$$

The denoising score-matching objective becomes:

$$\mathcal{L}_{\mathsf{DSM}}(\theta) = \mathbb{E}\left[\|s_\theta(X_t, t) + \frac{\epsilon_t}{\sqrt{1-\bar{\alpha}_t}}\|^2\right] \,. \tag{34}$$

A straightforward reparameterization shows this is equivalent to learning to predict the noise ϵ_t added during the forward process.

SDE Formulation – Generative Process

- **Stochastic Sampling**

$$d\bar{X}_t = \left[\mu_t(\bar{X}_t) - \sigma_t^2 s_\theta(\bar{X}_t, t)\right] \mathrm{d}t + \sigma_t \mathrm{d}W_t \ . \tag{35}$$

Start from $\bar{X}_T \sim \mathcal{N}(0, \mathrm{I}_d)$ and solve the SDE backward in time. This is similar to DDPM.

- **Deterministic Sampling**

$$d\bar{X}_t = \left[\mu_t(\bar{X}_t) - \frac{1}{2}\sigma_t^2 s_\theta(\bar{X}_t, t)\right] \mathrm{d}t \ . \tag{36}$$

This corresponds to deterministic sampling, similar to DDIM. The ODE formulation enables faster and more stable sampling.

SDE Formulation

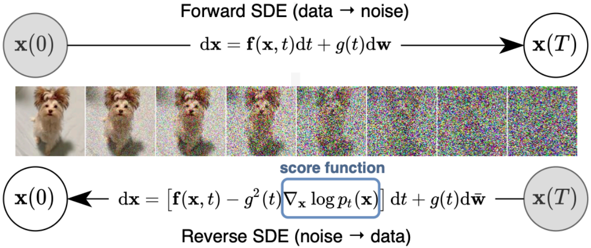

Figure: SDE-based generative model

Classifier-Free Guidance (CFG)

Conditioning the Generative Model

- **Straightforward solution** Train a conditioned model $s_\theta(x_t, t, c)$ where c is the class attribute.
- **Empirical improvement: Classifier guidance** By Bayes formula:

$$p_t(x_t|c) \propto p_t(c|x_t)p_t(x_t) \ , \tag{37}$$

so

$$\nabla_x \log p_t(x_t|c) = \nabla_x \log p_t(c|x_t) + \nabla_x \log p_t(x_t) \ , \tag{38}$$

and in addition to the usual score model $s_\theta(\cdot, t)$, we train a classifier $p_\theta(c, t|\cdot)$ (and sample smartly from its gradient log). In practice, it has been observed that increasing *guidance scale* ω achieves better quality to the expense of diversity, i.e., sampling from a modified conditioned score model:

$$\nabla_x \log \tilde{p}_t(x_t|c) = \omega\nabla_x \log p_t(c|x_t) + \nabla_x \log p_t(x_t) \ , \tag{39}$$

Classifier-Free Guidance (CFG) – Motivation

Motivation

- Training an additional classifier is computationally costly and introduces complexities.
- Classifier-Free Guidance (CFG) provides a way to leverage conditioning information directly through a single neural network without an explicit classifier.
- We leverage only the idea of guidance and remove classifier

References: [HS22]

Classifier-Free Guidance – Main Idea

Key Ideas

- During training, the model learns both conditional and unconditional score functions by randomly dropping conditioning information (e.g., 10 % of the time).
- During sampling, the unconditional and conditional models are combined, amplifying the effect of the conditioning information with guidance scale:

$$\nabla_x \log \tilde{p}_t(x_t|c) = \omega(\nabla_x \log p_t(x_t|c) - \nabla_x \log p_t(x_t)) + \nabla_x \log p_t(x_t) \, . \tag{40}$$

Formally: Define two score functions:

$$\epsilon_\theta(x_t, t), \quad \epsilon_\theta(x_t, t, c), \tag{41}$$

where $\epsilon_\theta(x_t, t)$ is parameterizes the unconditional score and $\epsilon_\theta(x_t, t, c)$ parameterizes the conditional score (given c).

CFG – Sampling Procedure

CFG sampling formula: Use the same sampling equations but with $\tilde{\epsilon}_\theta$ defined as:

$$\tilde{\epsilon}_\theta(x_t, t, c) = \epsilon_\theta(x_t, t) + \omega \cdot (\epsilon_\theta(x_t, t, c) - \epsilon_\theta(x_t, t)) , \tag{42}$$

where:

- $\epsilon_\theta(x_t, t, c)$ is the conditional prediction.
- $\epsilon_\theta(x_t, t)$ is the unconditional prediction.
- $\omega \geqslant 1$ is the guidance scale, controlling the conditioning signal.

Interpretation:

- For $\omega = 1$: standard conditional sampling.
- For $\omega \to \infty$, generated samples become strongly conditioned, often sharper but less diverse.

CFG – Practical Impact

Effect of CFG:

- Enables high-quality conditional generation without explicit classifier training.
- Empirically shown to greatly improve sample fidelity (e.g., better Inception and FID scores).
- Widely adopted in text-to-image generative models such as Stable Diffusion, DALL · E 2.

Elucidated Diffusion Models (EDM)

Elucidated Diffusion Models (EDM) – Overview

[Kar+22]

- Comprehensive study systematically exploring the design space of diffusion models.
- Identifies key choices and their influence on performance.

Three main aspects explored:

- **Training formulation**
- **Noise schedules**
- **Sampling methods**

Elucidated Diffusion Models – Design Choices

Key identified design aspects:

- **Noise schedule β_t:** How quickly noise is introduced and removed.
- **Denoising parameterization:** Predicting noise (ϵ-prediction), data directly, or velocity.
- **Loss weighting scheme:** How much to emphasize different timesteps during training.
- **Sampler choice:** Euler-Maruyama, Heun's method, or other numerical integrators.
- **Continuous vs. Discrete time formulations:** Choosing discretization schemes.

Go further

Go further

- Generative models with other stochastic processes (e.g., PDMPs), and generator matching
- Stochastic interpolants
- Flow/Bridge matching and diffusion Schrödinger bridge
- Heavy-tailed diffusion
- Discrete data (e.g., text) or mixed type data
- Riemannian generative models
- ...

Thanks for listening

Reference I

[Vin11] Pascal Vincent. "A connection between score matching and denoising autoencoders". In: *Neural Comput.* 23.7 (July 2011), pp. 1661–1674. ISSN: 0899-7667. DOI: 10.1162/NECO_a_00142. URL: https://doi.org/10.1162/NECO_a_00142.

[HJA20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[SME20] Jiaming Song, Chenlin Meng, and Stefano Ermon. "Denoising Diffusion Implicit Models". In: *CoRR* abs/2010.02502 (2020). arXiv: 2010.02502. URL: https://arxiv.org/abs/2010.02502.

[Son+21] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. 2021. arXiv: 2011.13456 [cs.LG].

[HS22] Jonathan Ho and Tim Salimans. *Classifier-Free Diffusion Guidance*. 2022. arXiv: 2207.12598 [cs.LG]. URL: https://arxiv.org/abs/2207.12598.

[Kar+22] Tero Karras et al. *Elucidating the Design Space of Diffusion-Based Generative Models*. 2022. arXiv: 2206.00364 [cs.CV].

Reference II

[Sin23] Vaibhav Singh. *An In-Depth Guide to Denoising Diffusion Probabilistic Models DDPM – Theory to Implementation.* https://learnopencv.com/denoising-diffusion-probabilistic-models/. [Online; accessed 11-Feburary-2025]. 2023.