Generative Modelling with Score Diffusion

From VAE to Score Diffusion

Sanchez Victor, PhD Candidate

What is a generative model?

Definition

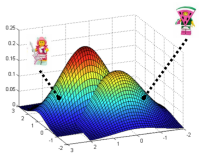
Let ν a probability distribution of interest. A generative model aims to generate new samples of ν, given the knowledge of some samples.



Original Samples



Generated Samples



Target Distribution

Generative Models - “First generation”

Learn a transport map $x = F_\theta(z), \quad z \sim \mu_0; \qquad$ write $q_\theta = (F_\theta)_{\#}\mu_0$

Normalizing Flow - [5]

Estimate the exact likelihood via an invertible map

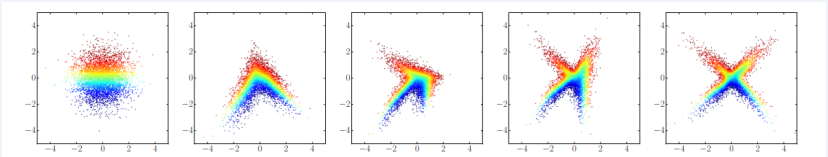


Figure 1: Example of a 4-step flow transforming samples from a standard-normal base density to a cross-shaped target density.

Figure: From : Papamakarios et al. Normalizing flows for probabilistic modeling and inference.

Generative Models - “First generation”

Learn a transport map $x = F_\theta(z), \quad z \sim \mu_0; \qquad$ write $q_\theta = (F_\theta)_{\#}\mu_0$

Normalizing Flow - [5]

Estimate the exact likelihood via an invertible map

$$\log q_\theta(x) = \log \mu_0\big(F_\theta^{-1}(x)\big) + \log \left|\det \nabla F_\theta^{-1}(x)\right|$$

$$\underbrace{\min_\theta \ \mathbb{E}_{x\sim\nu}\left[-\log q_\theta(x)\right]}_{\text{maximum likelihood}} \iff \underbrace{\min_\theta \ D_{\mathrm{KL}}(\nu \,\|\, q_\theta)}_{\text{data-to-model KL}}$$

Generative Models - “First generation”

Generative Adversarial Network - [2]

Distribution matching through a discriminator

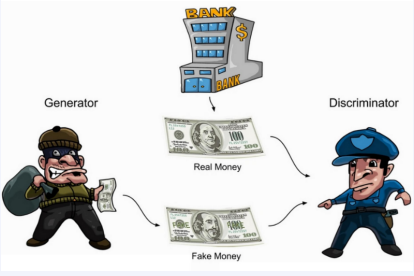


Figure: Atienza, Rowel. Advanced Deep Learning with Keras : Apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy gradients, and more. Packt Publishing Ltd, 2018.

Generative Models - “First generation”

Generative Adversarial Network - [2]

Distribution matching through a discriminator

$$\min_{\theta} \max_{D} \; \mathbb{E}_{x \sim \nu} \log D(x) + \mathbb{E}_{z \sim \mu_0} \log \left(1 - D(F_\theta(z))\right)$$

Generative Models - “First generation”

Variational Autoencoder - [1]

Maximize the ELBO directly, jointly over an encoder q_ϕ and a decoder p_θ.

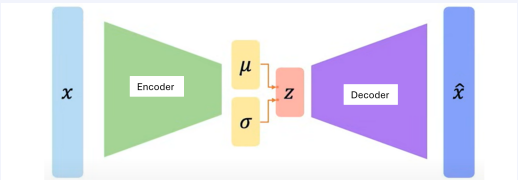


Figure: Source

Generative Models - “First generation”

Variational Autoencoder - [1]

Maximize the ELBO directly, jointly over an encoder q_ϕ and a decoder p_θ.

$$\mathrm{ELBO} = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{reconstruction}} - \underbrace{D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big)}_{\text{prior matching}}$$

Generative Models - “Second generation”

Learn a time-dependent field; generate samples by simulating its dynamics.

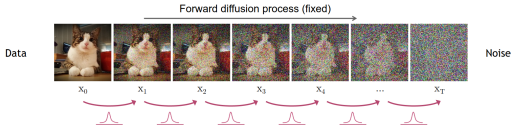


Figure: Source

Sampling uses multiple network evaluations.
What we will study today !

Generative Models - “Second generation”

Learn a time-dependent field; generate samples by simulating its dynamics.

Diffusion models - [3]

Denoising along a noising process

$$x_t = \sqrt{\bar{\alpha}_t}\, x + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, \qquad \min_\theta \; \mathbb{E}_{t,x,\varepsilon} \left\| \varepsilon_\theta(x_t, t) - \varepsilon \right\|^2 .$$

$$\text{Sampling:} \quad X_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(X_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \varepsilon_\theta(X_t, t) \right) + s_t \xi_t, \qquad X_T \sim \mathcal{N}(0, I).$$

Sampling uses multiple network evaluations.
What we will study today !

Generative Models - “Second generation”

Learn a time-dependent field; generate samples by simulating its dynamics.

Flow Matching - [4]

Velocity regression along an interpolation

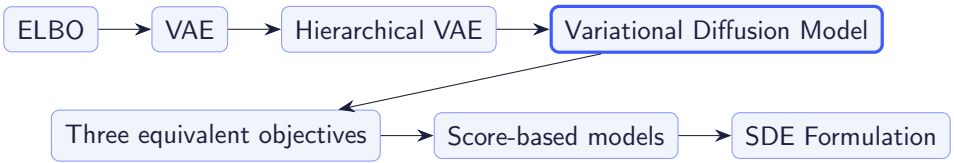
$$x_t = (1-t)z + tx, \qquad \min_\theta \ \mathbb{E}_{t,z,x} \left\| v_\theta(x_t, t) - (x - z) \right\|^2, \qquad z \sim \mu_0, \ x \sim \nu.$$

$$\text{Sampling:} \quad \dot{X}_t = v_\theta(X_t, t), \qquad X_0 \sim \mu_0.$$

Sampling uses multiple network evaluations.
Not treated in this course but the theory is close to Diffusion.

Road from VAE to diffusion

Each step is a small generalization of the one before.



Core claim from Luo et al. :

A diffusion model is a Markovian hierarchical VAE with a **fixed** noising encoder, and training it reduces to a **denoising regression**.

Latent variables and the evidence

What we observe, x, is generated from unseen latents z, like shadows on the wall of Plato's cave.

Two ways to get $p(x)$

$$p(x) = \int p(x, z)\, dz$$

$$p(x) = \frac{p(x, z)}{p(z \mid x)}$$

Why it is hard

- The integral over all z is intractable for complex models
- The true posterior $p(z \mid x)$ is not available

We usually want latents of *lower* dimension than x: compression that can uncover structure.

The Evidence Lower Bound (ELBO)

Introduce a flexible approximate posterior $q_\phi(z \mid x)$ and bound the log-likelihood from below.

$$\log p(x) = \underbrace{\mathbb{E}_{q_\phi(z|x)}\left[\log \frac{p(x,z)}{q_\phi(z|x)}\right]}_{\text{ELBO}} + D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z|x)\big)$$

Why a lower bound

The gap is a KL divergence, which is never negative, so the ELBO cannot exceed the evidence.

Why maximize it

$\log p(x)$ does not depend on ϕ. So raising the ELBO **lowers the KL**, pushing q_ϕ toward the true posterior.

Variational Autoencoder

Maximize the ELBO directly, jointly over an encoder q_ϕ and a decoder p_θ.

$$\mathrm{ELBO} = \underbrace{\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]}_{\text{reconstruction}} - \underbrace{D_{\mathrm{KL}}\big(q_\phi(z|x) \,\|\, p(z)\big)}_{\text{prior matching}}$$

Typical choices

$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x) I)$ and $p(z) = \mathcal{N}(z; 0, I)$. The KL is analytic; reconstruction uses Monte Carlo.

Reparameterization trick

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

Makes sampling differentiable.

Hierarchical VAE: stacking latents

In a Markovian HVAE, each latent is generated only from the one above it: a stack of VAEs.



$$p(x, z_{1:T}) = p(z_T)\, p_\theta(x|z_1) \prod_{t=2}^{T} p_\theta(z_{t-1}|z_t)$$

$$q_\phi(z_{1:T}|x) = q_\phi(z_1|x) \prod_{t=2}^{T} q_\phi(z_t|z_{t-1})$$

The same ELBO applies, with the joint and the posterior factorized along the chain.

A Variational Diffusion Model is a Markovian HVAE with three restrictions

These assumptions make the ELBO tractable and scalable.

1 · Same dimension

The latent dimension equals the data dimension. Write data as x_0 and latents as $x_1, \ldots, x_T$.

2 · Fixed encoder

Each step is a pre-defined linear Gaussian: noise is added to the previous output. Nothing to learn.

3 · Pure noise at the end

The noise schedule is set so that x_T is a standard Gaussian.

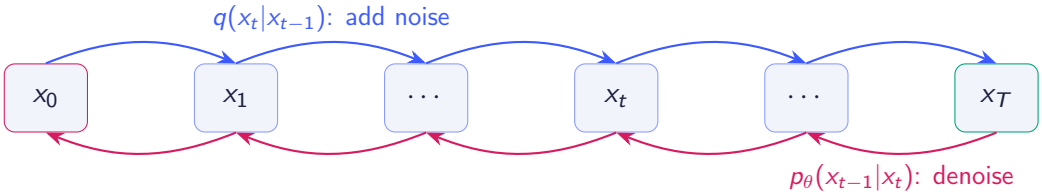
$$q(x_t|x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{\alpha_t}\, x_{t-1}, (1-\alpha_t)I\big)$$

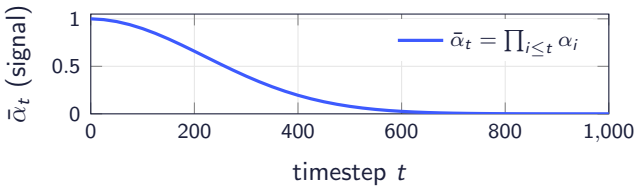
$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon_t, \qquad \epsilon \sim \mathcal{N}(0, I)$$

Variance-preserving: the scale of the latents stays comparable across steps.

Forward process: from image to noise

Data x_0 is noised step by step into x_T; the model learns to run the chain backwards.





Illustrative linear schedule, $\beta_t = 1 - \alpha_t$ over $T = 1000$.

Jump straight to any noise level

Repeated reparameterization, plus "a sum of independent Gaussians is Gaussian", collapses the chain.

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \epsilon_0, \qquad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$$

$$q(x_t|x_0) = \mathcal{N}\big(x_t;\, \sqrt{\bar{\alpha}_t}\, x_0,\, (1-\bar{\alpha}_t) I\big)$$

Training

Noise any clean image to any level in one shot, so training can sample random timesteps.

Sampling

Draw $x_T \sim \mathcal{N}(0, I)$, then run the learned denoising steps $p_\theta(x_{t-1}|x_t)$ for T steps.

The ELBO of a diffusion model

A rewrite using Bayes' rule gives a lower-variance form: each term involves at most one random variable.

$$\begin{aligned}\mathrm{ELBO} &= \mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)] \\ &\quad - D_{\mathrm{KL}}\big(q(x_T|x_0) \,\|\, p(x_T)\big) \\ &\quad - \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\Big[D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t)\big)\Big]\end{aligned}$$

Reconstruction

Predict the data from the first latent, as in a vanilla VAE.

Prior matching

No trainable parameters, and effectively zero for large T.

Denoising matching

Make each learned step $p_\theta(x_{t-1}|x_t)$ match the tractable true step $q(x_{t-1}|x_t, x_0)$. This term dominates the cost.

The ground-truth denoising step is a Gaussian

By Bayes' rule, $q(x_{t-1}|x_t, x_0) \propto q(x_t|x_{t-1})\, q(x_{t-1}|x_0)$, and completing the square gives:

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}\big(x_{t-1};\, \mu_q(x_t, x_0),\, \sigma_q^2(t) I\big)$$

$$\mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})\, x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\, x_0}{1-\bar{\alpha}_t}$$

$$\sigma_q^2(t) = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}$$

Everything here is known and fixed. Only x_0 is unavailable at sampling time, which is what the network must supply.

Training reduces to denoising

Give the approximate step the same variance. The KL between two Gaussians collapses to a difference of means.

$$\arg\min_{\theta} D_{\mathrm{KL}}\big(q(x_{t-1}|x_t, x_0) \,\|\, p_\theta(x_{t-1}|x_t)\big) = \arg\min_{\theta} \frac{1}{2\sigma_q^2(t)} \|\mu_\theta - \mu_q\|_2^2$$

Choose μ_θ like μ_q, with a network $\hat{x}_\theta(x_t, t)$ in place of x_0:

$$\arg\min_{\theta} \frac{1}{2\sigma_q^2(t)} \frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2} \left\|\hat{x}_\theta(x_t, t) - x_0\right\|_2^2$$

In words: train a network to predict the original image from a noisy version of it and the time index.

Three equivalent interpretations

x_0 can be reparameterized in two other ways, giving three ways to train the same model.

Network predicts	Mean $\mu_\theta(x_t, t)$	Loss
Image $\hat{x}_\theta \approx x_0$	$\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{x}_\theta}{1-\bar{\alpha}_t}$	$\|\hat{x}_\theta - x_0\|^2$
Noise $\hat{\epsilon}_\theta \approx \epsilon_0$	$\frac{1}{\sqrt{\alpha_t}}x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\hat{\epsilon}_\theta$	$\|\epsilon_0 - \hat{\epsilon}_\theta\|^2$
Score $s_\theta \approx \nabla \log p(x_t)$	$\frac{1}{\sqrt{\alpha_t}}x_t + \frac{1-\alpha_t}{\sqrt{\alpha_t}}s_\theta$	$\|s_\theta - \nabla \log p(x_t)\|^2$

Score $\leftrightarrow$ noise

$$\nabla \log p(x_t) = -\frac{1}{\sqrt{1-\bar{\alpha}_t}}\epsilon_0$$

Where the score comes from

Tweedie's formula:
$\mathbb{E}[\mu_z|z] = z + \Sigma_z \nabla \log p(z)$. Some works found noise prediction works better empirically.

Why model the score?

Energy-based models need a normalizing constant Z_θ. The score does not.

Energy model

$$p_\theta(x) = \frac{1}{Z_\theta} e^{-f_\theta(x)}$$

$$\nabla_x \log p_\theta(x) = -\nabla_x f_\theta(x) \approx s_\theta(x)$$

The score is a vector field

At each point it says which direction in data space raises the likelihood: it points toward the modes.

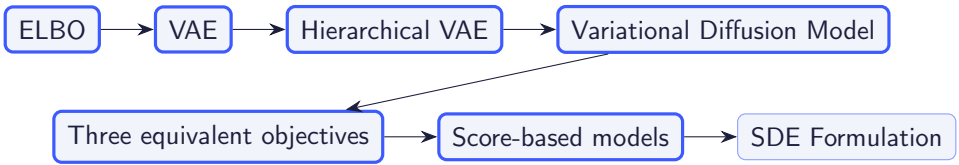
Langevin dynamics sampling

$$x_{i+1} \leftarrow x_i + c\,\nabla \log p(x_i) + \sqrt{2c}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I)$$

Follow the score from a random start. The noise term keeps samples from collapsing onto a single mode. Score matching trains s_θ without the ground-truth score.

Back to the Road from VAE to diffusion

Each step is a small generalization of the one before.



Core claim from Luo et al. :

A diffusion model is a Markovian hierarchical VAE with a **fixed** noising encoder, and training it reduces to a **denoising regression**.

Let's take a break

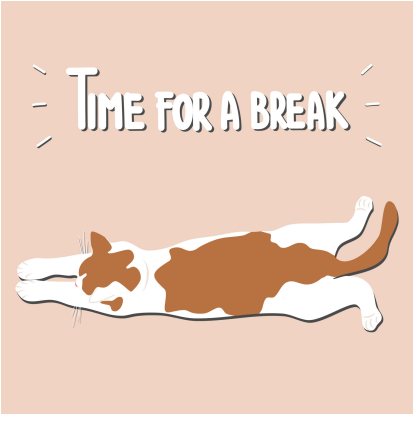
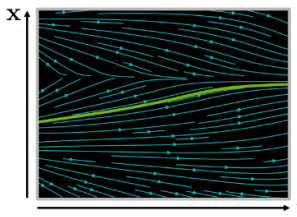


Figure: Cat taking a break

Recalls about Differential Equations

Ordinary Differential Equation (ODE):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}(\mathbf{x}, t) \quad \text{or} \quad \mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t$$



Analytical Solution: $\mathbf{x}(t) = \mathbf{x}(0) + \int_0^t \mathbf{f}(\mathbf{x}, \tau)\mathrm{d}\tau$

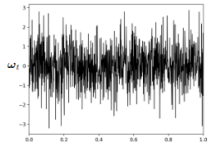
Iterative Numerical Solution: $\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), t)\Delta t$

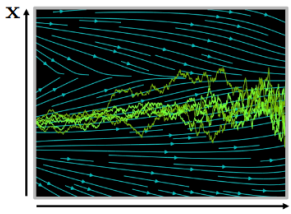
Stochastic Differential Equation (SDE):

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \underbrace{\mathbf{f}(\mathbf{x}, t)}_{\text{drift coefficient}} + \underbrace{\sigma(\mathbf{x}, t)\boldsymbol{\omega}_t}_{\text{diffusion coefficient}}$$

$$\left(\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + \sigma(\mathbf{x}, t)\mathrm{d}\boldsymbol{\omega}_t \right)$$

Wiener Process (Gaussian White Noise)



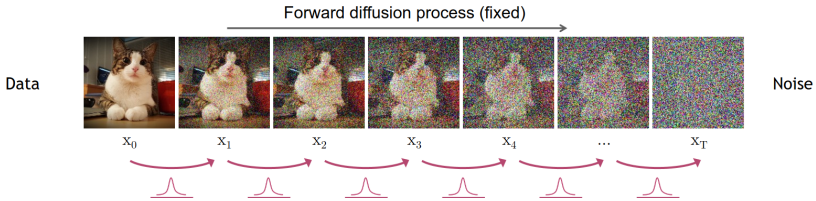


$$\mathbf{x}(t + \Delta t) \approx \mathbf{x}(t) + \mathbf{f}(\mathbf{x}(t), t)\Delta t + \sigma(\mathbf{x}(t), t)\sqrt{\Delta t}\mathcal{N}(\mathbf{0}, \mathbf{I})$$

Figure: ODE vs SDE models

Forward Diffusion Process

Let's now consider the forward diffusion process again :

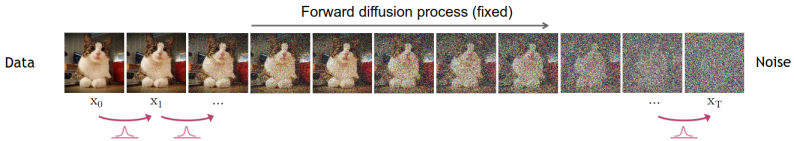


$$q(x_t|x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{\alpha_t}\, x_{t-1}, (1-\alpha_t)I\big)$$

$$\begin{aligned} x_t &= \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \\ &= \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon, \qquad \beta_t = 1-\alpha_t \qquad \epsilon \sim \mathcal{N}(0, I) \end{aligned}$$

Forward Diffusion Process

Consider the limit of many small steps:

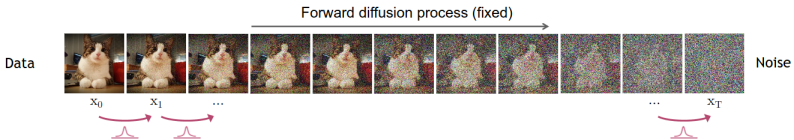


$$q(x_t|x_{t-1}) = \mathcal{N}\big(x_t; \sqrt{1-\beta_t}\, x_{t-1},\, \beta_t I\big)$$

$$\begin{aligned} x_t &= \sqrt{1-\beta_t}\, x_{t-1} + \sqrt{\beta_t}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I) \\ &= \sqrt{1-\beta(t)\Delta_t}\, x_{t-1} + \sqrt{\beta(t)\Delta_t}\, \epsilon \end{aligned}$$

Forward Diffusion Process

Consider the limit of many small steps:



$$x_t = \sqrt{1-\beta(t)\Delta_t}\, x_{t-1} + \sqrt{\beta(t)\Delta_t}\,\epsilon, \qquad \epsilon \sim \mathcal{N}(0, I)$$

$$\approx x_{t-1} - \frac{\beta(t)\Delta_t}{2} x_{t-1} + \sqrt{\beta(t)\Delta_t}\,\epsilon \qquad \text{Taylor expansion}$$

Forward Diffusion Process as Stochastic Differential Equation (SDE)

Consider the limit of many small steps:

$$x_t \approx x_{t-1} - \frac{\beta(t)\Delta_t}{2} x_{t-1} + \sqrt{\beta(t)\Delta_t}\,\epsilon$$

SDE Forward Diffusion Process

$$dx_t = -\frac{1}{2}\beta(t)x_t dt + \sqrt{\beta(t)}dw_t$$

SDE Formulation

Diffusing data into noise with an SDE [6]

Stochastic Differential Equation

$$\boldsymbol{dx} = f(\boldsymbol{x}, t)\boldsymbol{dt} + g(t)\boldsymbol{dw}$$

$f(\boldsymbol{x}, t)$
Drift: the deterministic pull on the data.

$g(t)$
Diffusion: how strongly noise enters at time t.

Standard Wiener process (Brownian motion).

Reversing time gives another SDE (Anderson, 1982)

Stochastic Differential Equation

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

Time runs backward
t is a negative step from T to 0, and $\bar{}$ is a reverse-time Wiener process.

One unknown: the score
$\nabla \log p$ is the gradient of the log marginal density at time t.

Learn the score, simulate the reverse SDE, generate samples.

Estimating the score with a neural network

Training objective

$$\min_{\theta} \; \mathbb{E}_t \Big\{ \lambda(t) \, \mathbb{E}_{(0)} \mathbb{E}_{(t)|(0)} \Big[\big\| \boldsymbol{s}_\theta((t), t) - \nabla_{(t)} \log p_{0t}((t) \mid (0)) \big\|^2 \Big] \Big\}$$

Time-dependent model
One network $\boldsymbol{s}_\theta(, t)$ estimates the score at every noise level.

Denoising target
The perturbation kernel is Gaussian, so its score has a closed form.

Weighting
$\lambda(t)$ sets how much each noise level counts in the loss.

Three SDE families

SDE	Drift $f(,t)$	Diffusion $g(t)$	Matches
VE	0	$\sqrt{[\sigma^2(t)]/t}$	SMLD
VP	$-\frac{1}{2}\beta(t)$	$\sqrt{\beta(t)}$	DDPM
sub-VP	$-\frac{1}{2}\beta(t)$	$\sqrt{\beta(t)\left(1-e^{-2\int_0^t \beta(s)s}\right)}$	new here

- **VE** (variance exploding): noise variance grows without bound.
- **VP** (variance preserving): variance stays bounded.
- **sub-VP**: variance never exceeds the VP SDE's.

Earlier methods are discretizations

SMLD $\leftrightarrow$ VE SDE

SMLD's increasing noise scales $\sigma_1 < \cdots < \sigma_N$ form a Markov chain. As $N \to \infty$, that chain becomes the VE SDE.

DDPM $\leftrightarrow$ VP SDE

DDPM's chain $_i = \sqrt{1-\beta_i}\,_{i-1} + \sqrt{\beta_i}\,_{i-1}$ converges to the VP SDE as $N \to \infty$.

One framework: the noise schedule and the sampler become separate design choices.

Solving the reverse SDE numerically

Reverse diffusion

Discretize the reverse SDE with the same scheme used for the forward SDE.

Ancestral sampling

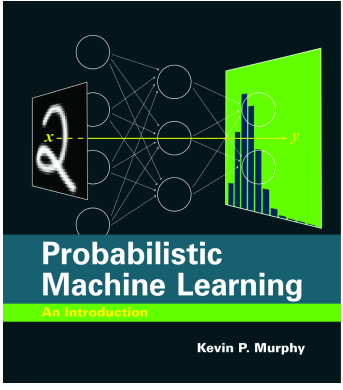
DDPM's sampler is one particular discretization of the reverse VP SDE.

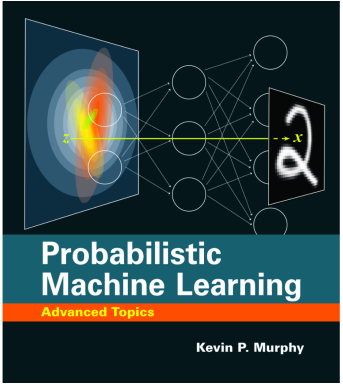
Predictor–corrector

A new general scheme: an SDE solver steps, then a score-based MCMC step refines.

Continuous time frees us to pick any solver, rather than being locked to a fixed discrete chain.

Resources available online for free to go beyond on deep learning and generative modelling:





Bibliography:

[1] P Kingma Diederik and Welling Max. "An introduction to variational autoencoders". In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[2] Ian Goodfellow et al. "Generative adversarial networks". In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models". In: *Advances in neural information processing systems* 33 (2020), pp. 6840–6851.

[4] Yaron Lipman et al. "Flow matching for generative modeling". In: *arXiv preprint arXiv:2210.02747* (2022).

[5] George Papamakarios et al. "Normalizing flows for probabilistic modeling and inference". In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64.

[6] Yang Song et al. "Score-based generative modeling through stochastic differential equations". In: *arXiv preprint arXiv:2011.13456* (2020).