# Diffusion models

Ernesto Mordecki

Facultad de Ciencias, Universidad de la República. Montevideo, Uruguay

Procesos estocásticos y simulación - FCIEN - 2023

# Contenidos

# ¿Qué es una difusión?

- "... flujo de átomos, iones ... que se mueven de una región de alta concentración a un área de baja concentración hasta obtener una distribución uniforme."[1]
- Los modelos matemáticos de difusiones pueden ser
  - *macroscópicos* y se modelan por ecuaciones en derivadas parciales (PDE)
  - *microscópicos* y se modelan por ecuaciones diferenciales estocásticas (SDE)
- tenemos:
  - tendencia o drift $f(t, \mathbf{x})$ que indica el sentido del movimiento
  - coeficiente de difusión $g(t, \mathbf{x})$, un ruido o error[2].

[1] Wikipedia

[2] La *volatilidad*

# Modelo en tiempo discreto

- Tenemos un intervalo de tiempo $[0, T]$, lo partimos en $N$ intervalos iguales de largo $\Delta t = T/N$
- Tenemos dos funciones

$f(t, \mathbf{x}) \colon \mathbb{R}^d \to \mathbb{R}^d$, el drift o tendencia,

$g(t, \mathbf{x}) \colon \mathbb{R}^d \to \mathbb{R}^{d \times d}$, la volatilidad o desvío estándar.

- Tenemos nodos

$$0 = t_0, \frac{T}{N}, \ldots, t_k = \frac{kT}{N}, \ldots, t_N = T.$$

# Simulación

- *Simulamos* la difusión partiendo de un punto $X(0) = \mathbf{x}_0$ y

$$X(t_{k+1}) = X(t_k) + f(t_k, X(t_k))\Delta t + g(t_k, X(t_k))\sqrt{\Delta t}\mathcal{N}(0, Id_d).$$

- En cada paso sumamos:
  - un desplazamiento diferencial en la dirección de $f(t_k, X(t_k))$
  - un ruido diferencial gaussiano con correlaciones dadas por $g(t_k, X(t_k))$
  - Es importante la densidad

$$p_t(x)\, dx = \mathbf{P}(X(t) \in\ dx).$$

# Modelo general en tiempo continuo

- Tenemos un intervalo de tiempo $[0, T]$,
- Tenemos dos funciones $f(t, \mathbf{x}) \colon \mathbb{R} \to \mathbb{R}$, $g(t, \mathbf{x}) \colon \mathbb{R} \to \mathbb{R}$
- *Resolvemos* la ecuación diferencial estocástica partiendo de un punto $X(0) = \mathbf{x}_0$ y

$$X(t) = \mathbf{x}_0 + \int_0^t f(s, X(s))\, ds + \int_0^t g(s, X(s))\, dW(s)$$

- Aquí la primer integral es usual, y la segunda es una *integral estocástica*, donde $W(t)$ es un movimiento Browniano
- La solución $X(t)$ es una *difusión*

# Un poco de historia

- En 1931 Andrei Kolmogorov da una descripción macroscópica estableciendo entre otras cosas dos EDP: básicas de la teoría:

(B) Ecuación backward, para $t \leq T$, con $p(T, x) = u(T, x)$:

$$-\frac{\partial}{\partial t}p(t,x) = f(t,x)\frac{\partial}{\partial x}p(t,x) + \frac{1}{2}g^2(t,x)\frac{\partial^2}{\partial x^2}p(t,x)$$

(F) Ecuación forward, con $t \leq s$, con $p(t, x) = p_t(x)$:

$$\frac{\partial}{\partial s}p(s,x) = -\frac{\partial}{\partial x}[\mu(s,x)p(s,x)] + \frac{1}{2}\frac{\partial^2}{\partial x^2}[\sigma^2(s,x)p(s,x)],$$

- En 1946 Kiyosi Itō publica sus ecuaciones diferenciales estocásticas. Teorema de existencia y unicidad de soluciones de

$$X(t) = \mathbf{x}_0 + \int_0^t f(s, X(s))\, ds + \int_0^t g(s, X(s))\, dW(s)$$

- La integral estocástica verifica la isometría de Itō:

$$\mathbf{E}\left(\int_0^t g(s, X(s))\, dW(s)\right)^2 = \int_0^t \mathbf{E}(g(s, X(s)))^2\, ds.$$

# Fórmula de Itō

- En 1951 obtiene la siguiente fórmula: Si $H(t,x)$ es $C^{1,2}$,

$$H(t,X(t)) - H(0,X(0)) = \int_0^t (H_t + \mathcal{L}H)(s,X(s))\,ds + \int_0^t (gH_x)(s,X(s))\,dW_s.$$

- El generador infinitesimal $\mathcal{L}$ es

$$\mathcal{L}H(t,x) = f(t,x)H_x(t,x) + \frac{1}{2}g(t,x)^2 H_{xx}(t,x).$$

- Esto relaciona las difusiones con las ecuaciones en derivadas parciales (EDP)

# Una explosión de aplicaciones: matemática financiera

- En 1973 Black y Scholes publican su fórmula de valuación de opciones[3]. El modelo era

$$dB_t = rB_t\,dt$$
$$dS_t = S_t(\mu\,dt + \sigma\,dW_t)$$

- En los 90 las empresas financieras contrataban matemáticos formados en la academia
- A partir de 2010 aparecen una serie de artículos aplicando difusiones a Machine Learning (imágenes)
- Hoy las empresas de ML producen sus resultados con matemáticos que contratan

[3] dando origen a una migración de matemáticos hacia la matemática financiera

# Una segunda aplicación: Machine learning

## Deep Unsupervised Learning using Nonequilibrium Thermodynamics

**Jascha Sohl-Dickstein** JASCHA@STANFORD.EDU
Stanford University

**Eric A. Weiss** EWEISS@BERKELEY.EDU
University of California, Berkeley

**Niru Maheswaranathan** NIRUM@STANFORD.EDU
Stanford University

**Surya Ganguli** SGANGULI@STANFORD.EDU
Stanford University

⋮

### 1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However,

*Proceedings of the $32^{nd}$ International Conference on Machine Learning*, Lille, France, 2015. JMLR: W&CP volume 37. 

### 1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

[1]Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

# Backward diffusions: un teorema importante

**Teoerma**[4]**:** Sea $X(t)$ solución de una EDE

$$dX(t) = f(t, X(t))dt + g(t)dW$$

Entonces, el mismo proceso también es solución, con tiempo revertido (empieza en $t = T$ y termina en $t = 0$) de la EDE

$$dX(t) = [f(t, X(t)) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)d\bar{W}(t)$$

Donde

- $p_t(x)$ es la densidad de $X(t)$ (solución de la PDE forward de 1931!)
- $s(t, x) = \nabla_x \log p_t(x)$ es el score
- $\{\bar{W}_t\}$ es un Browniano con tiempo revertido.

[4] B. D. A. Andersen. Reverse-time diffusion equation models, SPA (1982)

# Diffusion models en Machine Learning

- Nuestro objetivo es *simular* una imagen, por ejemplo el rostro de una persona, por ejemplo en blanco y negro
- Tenemos una gran base de datos con rostros de personas. Cada foto consiste en:
  - 1920 x 1080 = 2 073 600 pixels
  - De 0 (negro) a 255 (blanco) en grises, se lleva a $[-1, 1]$
  - Aproximadamente 2 MB
- Cada foto es un punto $\mathbf{x}$ en $\mathbb{R}^{2073600}$, proveniente de una distribución $p_\theta(\mathbf{x})$
- Nos proponemos simular $\mathbf{x}^*$ con densidad $p_\theta(\mathbf{x})$.

# Algoritmo

- Partimos de un punto $\mathbf{x}_0$ (una foto)
- Le *agregamos* ruido mediante una difusion forward hasta llegar[5] a "ruido blanco" $\mathbf{x}_T$
- Entrenamos una red neuronal (RN) para *sacar* ruido (denoising) mediante la difusión backward
- Este es un paso clave del algoritmo (próxima diapo)
- Una vez entrenada la RN, simulamos un ruido blanco y lo backwardeamos por la RN y obtenemos $\mathbf{x}^*$, nuestra imagen simulada

[5] Si bien es ruido blanco, tiene información de la foto

# Toy model[6]

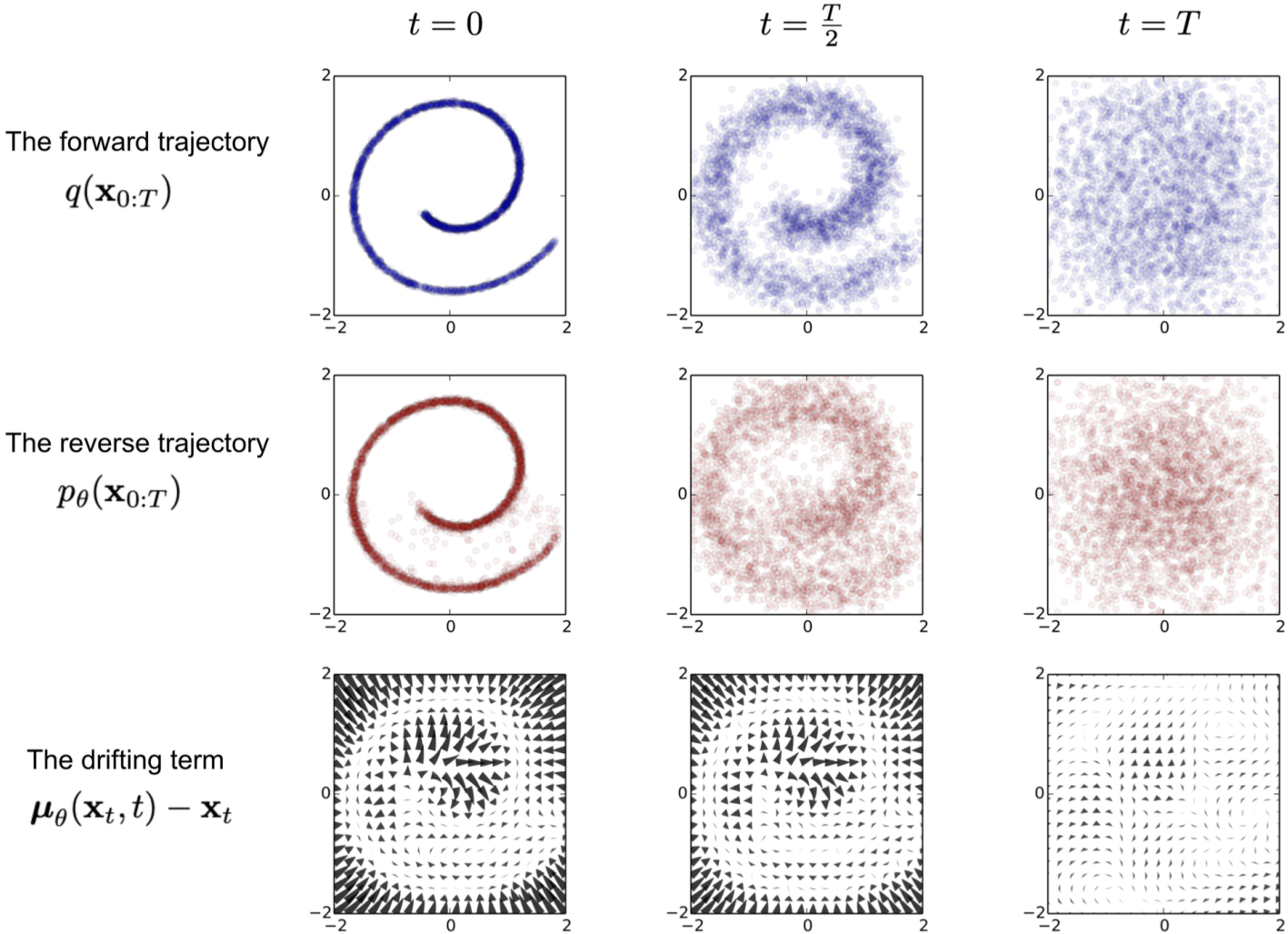


Fig. 3. An example of training a diffusion model for modeling a 2D swiss roll

[6]Deep Unsupervised Learning using Nonequilibrium Thermodynamics. Sohl-Dickstein et al. 2015

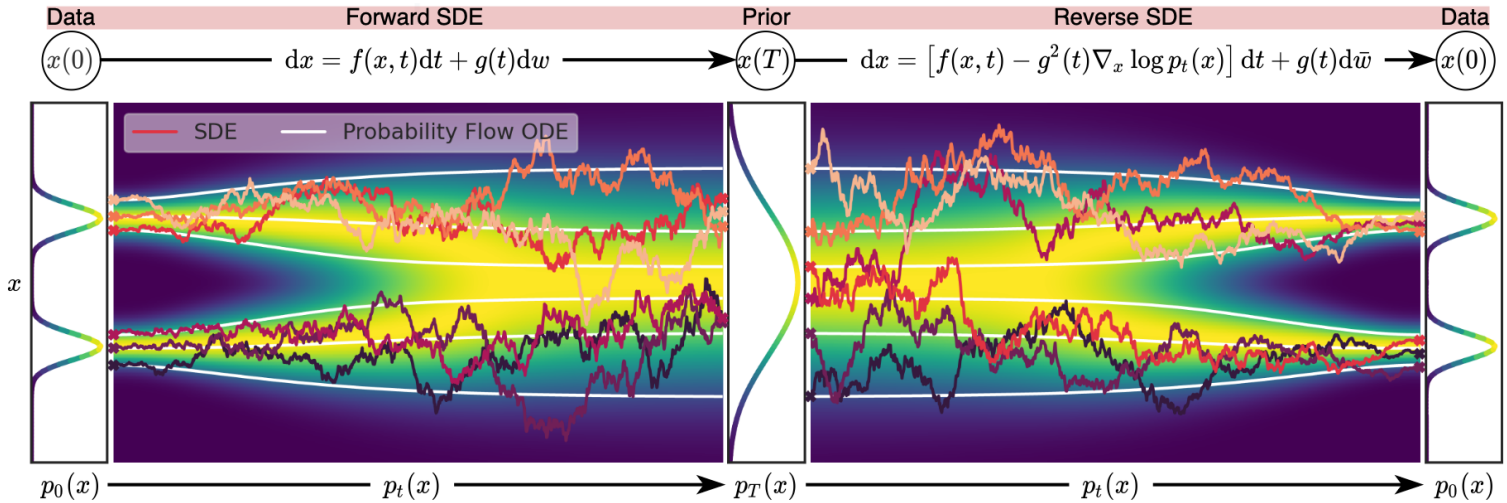


Figure 2: **Overview of score-based generative modeling through SDEs**. We can map data to a noise distribution (the prior) with an SDE (Section 3.1), and reverse this SDE for generative modeling (Section 3.2). We can also reverse the associated probability flow ODE (Section 4.3), which yields a deterministic process that samples from the same distribution as the SDE. Both the reverse-time SDE and probability flow ODE can be obtained by estimating the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ (Section 3.3).

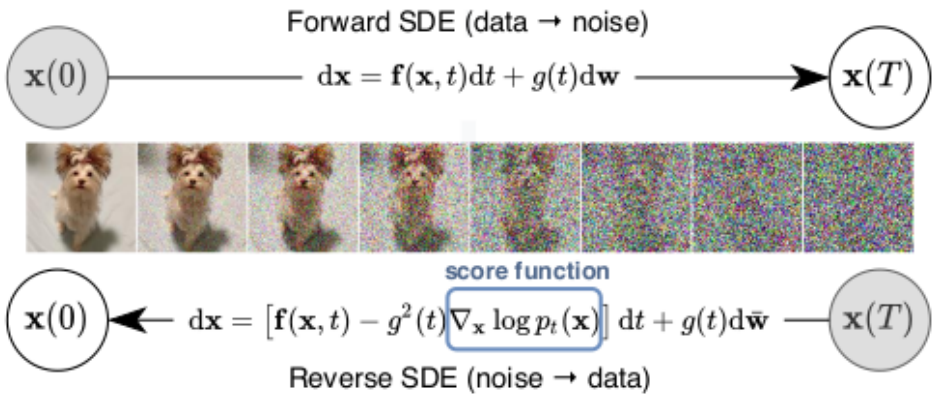


Figure 1: **Solving a reverse-time SDE yields a score-based generative model.** Transforming data to a simple noise distribution can be accomplished with a continuous-time SDE. This SDE can be reversed if we know the score of the distribution at each intermediate time step, $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

# Entrenamiento de la Red Neuronal

- El problema es estimar

$$s_\theta(t, \mathbf{x}) = \nabla \log p_\theta(t, \mathbf{x})$$

- $p_\theta(t, \mathbf{x})$ es la densidad de las imágenes en el espacio, algo del tipo $\mathbb{R}^{\text{dos millones}}$
- La fotos viven en un subconjunto de mucho menor dimensión

# Función de **pérdida** $L$ (tiempo continuo)

- Se estima $s(t,x)$ mediante $s_\theta(t,x)$, una red neuronal.
- La función de pérdida (teórica) es

$$L(\theta) = \int_0^T \mathbf{E}_{p_t(x)} \left( w(t) \| s_\theta(t, x_t) - \nabla p_t(x) \| \right) \, dt.$$

- Se aproxima $p_t(x)$ por

$$\begin{aligned} \hat{p}_t(x) = \mathbf{E}_{\hat{\mu}_{data}(x_0)}(p_{t|0}(x|x_0)) &= \frac{1}{N} \sum_{i=1}^{N} p_{t|0}(x|x_i) \\ &\approx p_{t|0}(x|x_0) = \mathbf{E}_{\mu_{data}(x_0)}(p_{t|0}(x|x_0)). \end{aligned}$$

# Función de **pérdida** para el entrenamiento

- Luego la pérdida aproximada (surrogate loss) es

$$\hat{L}(\theta) = \int_0^T \mathbf{E}_{\hat{p}_t(x)} \left( w(t) \| s_\theta(t, x_t) - \nabla \hat{p}_t(x) \| \right) \, dt.$$

- $w(t)$ es un ponderador a elegir.

# Función de **pérdida** $L$ (tiempo discreto)

Se demuestra que

$$\mathbf{E}(-\log p_\theta(\mathbf{x}_0)) \leq \mathbf{E}\left[-\log p(\mathbf{x}_T) - \sum_{t=1}^{t=T} \log\left(\frac{p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)}{q(\mathbf{x}_t \mid \mathbf{x}_{t-1})}\right)\right] =: L$$

Cálculos mediante, se llega a minimizar el modelo

$$\begin{aligned} L(\theta) &= \mathbf{E}_{t,\mathbf{x}_0,\epsilon} \left\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, t)\right\|^2 \\ &= \frac{1}{N}\sum_{t=1}^{N} \mathbf{E}_{\mathbf{x}_0,\epsilon} \left\|\epsilon - \epsilon_\theta(\sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon, t)\right\|^2 \end{aligned}$$

donde

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\epsilon.$$

- Se entrena la RN a partir del ruido $\epsilon$ para producir $\epsilon_\theta$ a partir de una CL de la imagen y el ruido

**Algorithm 1** Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$$
6: **until** converged

**Algorithm 2** Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

Figura: De “Denoising Diffusion Probabilistic Models. Ho el al. (2020)”

# Modelo de Sohl-Dickstein (2015)

Con la notación del libro[7] en el caso discreto:

$$\begin{aligned} x_{t+1} &= \sqrt{1-\beta_t}x_t + \sqrt{\beta_t}\mathcal{N}(0, Id) \\ &= \sqrt{1-\beta(t)\Delta t}x_k + \sqrt{\beta(t)\Delta t}\mathcal{N}(0, Id) \end{aligned}$$

Para pasar al caso continuo, aproximando

$$x_{t+1} \sim x_t - \frac{1}{2}\beta(t)x_t\Delta t + \sqrt{\beta(t)\Delta t}\mathcal{N}(0, Id),$$

que nos lleva a la ecuación continua

$$dx(t) = -\frac{1}{2}\beta(t)x(t)\,dt + \sqrt{\beta(t)}\,dW(t),$$

donde $\{W(t)\}$ es un proceso de Wiener o movimiento Browniano.

[7] Murphy, Probabilistic Machine Learning, Vol 2.

# Comentarios

- La ecuación
$$dx = -\beta x_t dt + \sigma dW_t$$
define un proceso de Ornstein-Uhlenbeck (1930)
- Tenemos $f(t, x) = -\frac{1}{2}\beta(t)x$
- Tenemos $g(t, x) = diag(\sqrt{\beta(t)}, \ldots, \sqrt{\beta(t)})$
- Todas las coordenadas tienen el mismo ruido, que es no correlacionado.
- Tenemos reversión a la media ($x = 0$) y distribución estacionaria

# Solución

Si $\beta(t)$ es una funcion determinística, la ecuación es

$$dx(t) = -\frac{1}{2}\beta(t)x(t)dt + \sqrt{\beta(t)}dW_t,$$

la solución viene dada por

$$\boxed{x_t = x_0 e^{-\frac{1}{2}\int_0^t \beta(u)\,du} + \int_0^t e^{-\frac{1}{2}\int_s^t \beta(u)\,du}\sqrt{\beta(s)}dW_s}$$

# Solución

En efecto, multiplicando

$$e^{\frac{1}{2}\int_0^t \beta(u)\,du} x_t = x_0 + \int_0^t e^{\frac{1}{2}\int_0^s \beta(u)\,du} \sqrt{\beta(s)}\, dW_s$$

usamos la fórmula de Itō con $H(t,x) = e^{\frac{1}{2}\int_0^t \beta(u)\,du} x$:

$$e^{\frac{1}{2}\int_0^t \beta(u)\,du} \left[ dx_t + \frac{1}{2}\beta(t) x_t\, dt \right] = e^{\frac{1}{2}\int_0^t \beta(u)\,du} \sqrt{\beta(t)}\, dW_t$$

Ahora cancelamos el factor exponencial, y reordenamos

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

# Tres modelos

- El modelo de Ornstein-Uhlenbeck de 2015[8]:

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

- En 2019-2020[9] se propuso

$$dx_t = \sigma_t\,dW_t,$$

- En 2021[10] se propuso un modelo con dos ecuaciones: $\dot{V}_t = X_t$ (velocidad) y

$$dV_t = -X_t - 2V_t + 2\,dW_t.$$

- Los tres modelos son gaussianos (condicional a $x_0$)

[8] Deep Unsupervised . . . Sohl-Dickstein et al. 2015

[9] Song-Ermon. Improved techniques . . . 2020.

[10] Dockhorn et al. Score-based generative . . . arxiv 2021.

# Densidad del primer modelo[11]:

La densidad $p_t(x)$ de $x_t$ juega un papel importante en este problema. Como

$$x_t = x_0 e^{-\frac{1}{2}\int_0^t \beta(u)\,du} + \int_0^t e^{-\frac{1}{2}\int_s^t \beta(u)\,du} \sqrt{\beta(s)} dW_s$$

tenemos que $x_t$ es una convolución. La integral estocástica es normal centrada, con matriz diagonal de varianzas

$$\begin{aligned}\Sigma(t) &= \int_0^t e^{-\int_s^t \beta(u)\,du} \beta(s)\,ds \\ &= \left[-e^{-\int_s^t \beta(u)\,du}\right]_0^t \\ &= 1 - e^{-\int_0^t \beta(u)\,du}.\end{aligned}$$

[11] Score-Based Generative Models Detect Manifolds, Pidstrigach, arxiv:2022

# Conclusiones

- La esperanza condicional

$$\mathbf{E}(x_t|x_0) = x_0 e^{-\frac{1}{2}\int_0^t \beta(u)\,du} \to 0, \quad \text{si } t \to \infty$$

- La varianza

$$\Sigma(t) = 1 - e^{-\int_0^t \beta(u)\,du} \to 1 - e^{-\int_0^\infty \beta(u)\,du} = 1,$$

- Cuando $t \to \infty$

$$X_t \to \mathcal{N}(0, Id),$$

- Pero si $t \to 0$, tenemos $\Sigma(t) \to 0$.

# Densidad:

- Calculamos

$$m_t(x_0) = \mathbf{E}(x_t \mid x_0) = x_0 e^{-\frac{1}{2}\beta t},$$
$$p_{t|0}(x|x_0) = \mathcal{N}(x; m_t(x_0), \Sigma_t).$$

- Luego la densidad se escribe

$$p_t(x) = \int_{\mathbb{R}} \mathcal{N}(x; m_t(x_0), \Sigma_t) p(x_0)\, dx_0.$$

# Singularidad en $t = 0$:

- Además

$$\nabla \log p_t(x) = \frac{\nabla p_t(x)}{p_t(x)} = (\Sigma_t)^{-1} \left(x - \mathbf{E}(m_t(x_0)|X_t = x)\right) \to \infty$$

con $t \to 0$.

- Eso da divergencia del método cuando $t \sim 0$.
- Recordando la pérdida

$$L(\theta) = \int_0^T \mathbf{E}_{p_t(x)} \left(w(t) \| s_\theta(t, x_t) - \nabla p_t(x) \|\right) \, dt.$$

elegimos $w(t) \sim (\Sigma_t)^{-1}$.

# Stable diffusions[12]

- Por último, para acelerar el proceso y hacerlo posible en celulares se combina con encoders.
- Se utiliza una red neuronal tipo encoder (estrangulada) para las imágenes
- Se corre la difusión en el espacio codificado (de mucho menor diemensión) y se decodifica

[12] High-Resolution Image Synthesis with Latent Diffusion Models. Rombach et al. 2022

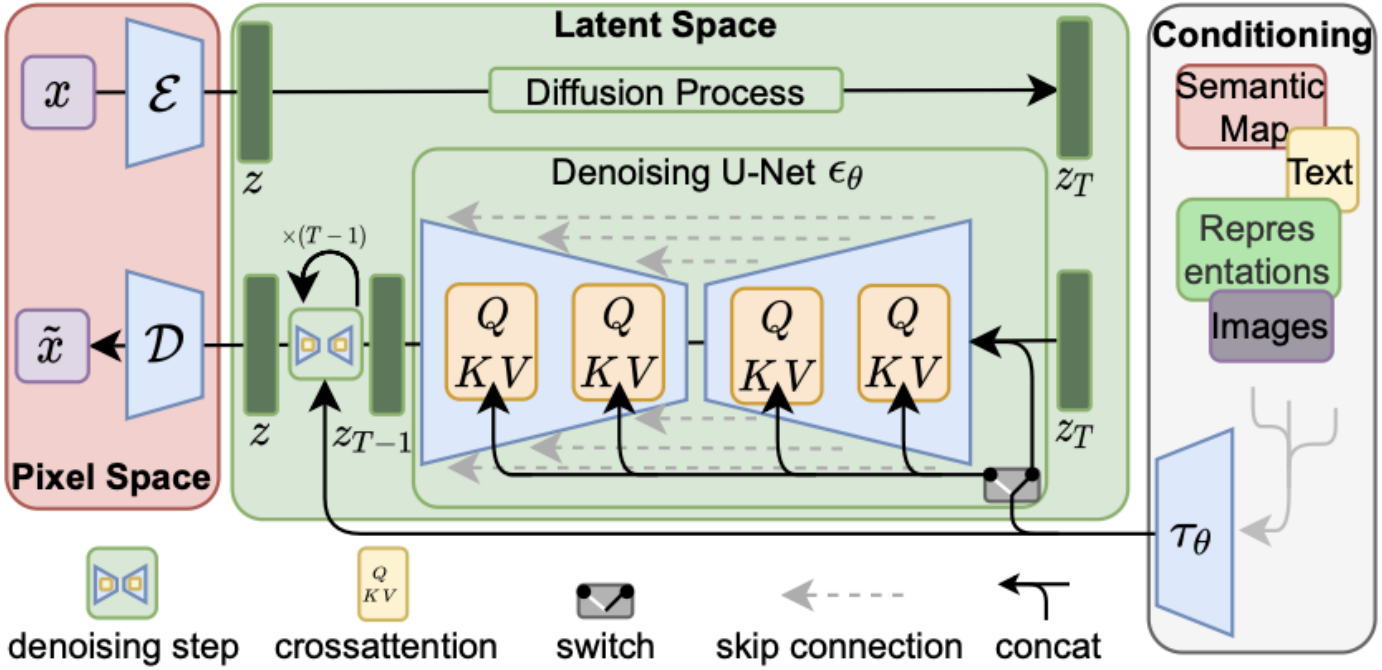


Figure 3. We condition LDMs either via concatenation or by a more general cross-attention mechanism. See Sec. 3.3

Figura: De “Latent Diffusion Models. Rombach et al. 2022”

# Referencias


- Kolmogorov, A.N. Ober die analytischen Methoden in der Wahrscheinlichkeitsrechnung. Math. Ann. 104415-458. (1931)
- K. Itō. On a stochastic integral equation. Proc. Imp. Acad. Tokyo 22 (1946), 32-35
- K. Itō. On a formula concerning stochastic differentials. Nagoya Math. J. Vol.3, 1951