Introduction to generative modelling
Session 2: Score-based Generative Models

Émile Mathieu, University of Cambridge

November 14, 2022

Muframex.

What are we going to talk about?

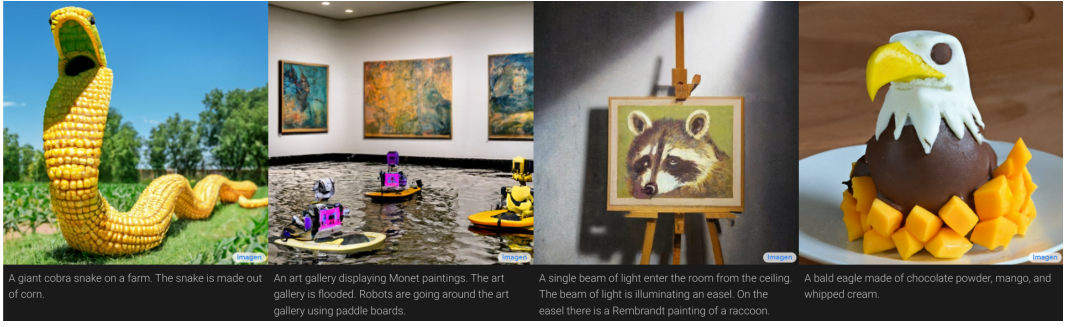
1. Score-based generative models:
    - Energy-based models
    - Score matching
    - Discrete-time diffusion process
2. Continuous score-based models:
    - Defining the model
    - Important tricks for training
    - Connection with continuous normalising flows
3. Variational perspective
4. Active research directions

We acknowledge Yang Song for using some images from his blogpost, Valentin De Bortoli's for some material from his generative modelling course and help from Michael Hutchinson in making these slides.

Score-based generative models

Motivating examples

Motivating examples



Energy-based models

Energy-based models (EBMs)

Parameterise a density via an *energy function* $U_\theta : \mathbb{R}^d \to \mathbb{R}_+$

$$p_\theta(\boldsymbol{x}) = \frac{\exp\left(-U_\theta(\boldsymbol{x})\right)}{Z_\theta}. \tag{1}$$

We can fit this energy function by maximising the log likelihood of the data under the density.

$$\theta^* = \max_\theta \mathbb{E}_{\boldsymbol{x}\sim p_{data}}[\log p_\theta(\boldsymbol{x})] = \max_\theta \sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}_i) \tag{2}$$

Note however we need to compute Z_θ to do this, and

$$Z_\theta = \int \exp(-U_\theta(\boldsymbol{x}))\mathrm{d}\boldsymbol{x} \tag{3}$$

Which we cannot tractably integrate in general, and leads to a nasty estimation problem.

Langevin Dynamics

Langevin Dynamics however give us an easy way to sample from such a distribution.

Theorem 1

The density of $\mathbf{X}_t$ as $t \to \infty$ for the SDE

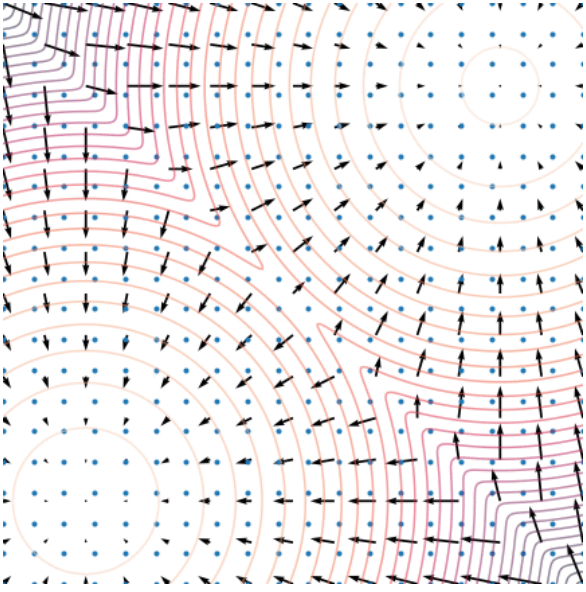
$$\mathrm{d}\mathbf{X}_t = -\nabla U(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t \tag{4}$$

is proportional to $\exp(-U(\mathbf{X}))$, where $\mathbf{B}_t$ is a suitable Brownian motion.

We can simulate this in discrete steps by iterating

$$\mathbf{X}_{t+1} = \mathbf{X}_t + \gamma \nabla U(\mathbf{X}_t) + \sqrt{2\gamma}\boldsymbol{z}_k, \quad \boldsymbol{z}_t \sim \mathcal{N}(0, I) \tag{5}$$

Langevin Dynamics



Why score based models?

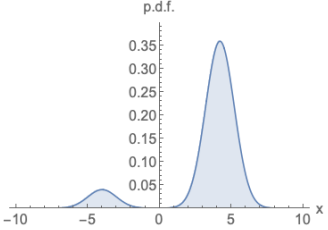
The (Stein) **score** of a distribution is the gradient w.r.t. the support of the log density.

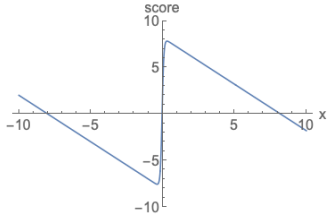
$$\mathbf{s}_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_\theta(\boldsymbol{x}) \tag{6}$$

This is useful as it is *independent* of the normalisation!

For example if we take the score of an energy-based model:

$$\nabla_{\boldsymbol{x}} \log p_\theta(x) = -\nabla_{\boldsymbol{x}} U_\theta(\boldsymbol{x}) - \underbrace{\nabla_{\boldsymbol{x}} \log Z_\theta}_{=0} = -\nabla_{\boldsymbol{x}} U_\theta(\boldsymbol{x}) \tag{7}$$





How do we learn a score?

We would like to **explicitly** match a parametric score to the (true) score, minimising

$$\ell_{\text{esm}}(\,\mathbf{s}_\theta\,) \triangleq \mathbb{E}_{p(\boldsymbol{x})}\left[\left\|\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}) - \mathbf{s}_\theta(\boldsymbol{x})\right\|^2\right] \tag{8}$$

which is referred as the *Fisher divergence*, or as the **explicit score matching** (ESM) loss.

Then we have $\mathbf{s}_\theta = \nabla \log p \Leftrightarrow p_\theta = p$. Yet the true score is *unavailable* to us...

Theorem 2: Implicit score matching (ISM), (Hyvärinen, 2005)

The Fisher divergence can be rewritten in a form free of the true score:

$$\ell_{\text{ism}}(\,\mathbf{s}_\theta\,) \triangleq \mathbb{E}_{p(\boldsymbol{x})}\left[\nabla_{\boldsymbol{x}} \cdot \mathbf{s}_\theta(\boldsymbol{x}) + \frac{1}{2}\left\|\mathbf{s}_\theta(\boldsymbol{x})\right\|^2\right] = \frac{1}{2}\ell_{\text{esm}}(\,\mathbf{s}_\theta\,) + C \tag{9}$$

What is good is that $\mathbf{s}_\theta$ is a *completely unconstrained function* $\mathbf{s}_\theta : \mathbb{R}^d \to \mathbb{R}^d$!

Proof of the Implicit score matching loss

$$\begin{aligned}
\ell_{\text{esm}}(\,\mathbf{s}_\theta\,) &= \mathbb{E}_{p(\boldsymbol{x})}\left[\left\|\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}) - \mathbf{s}_\theta(\boldsymbol{x})\right\|^2\right] \\
&= \mathbb{E}_{p(\boldsymbol{x})}\left[\left\|\mathbf{s}_\theta(\boldsymbol{x})\right\|^2 + \|\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x})\|^2 - 2\left\langle\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}),\, \mathbf{s}_\theta(\boldsymbol{x})\right\rangle\right] \\
&= \mathbb{E}_{p(\boldsymbol{x})}\left[\left\|\mathbf{s}_\theta(\boldsymbol{x})\right\|^2 - 2\left\langle\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}),\, \mathbf{s}_\theta(\boldsymbol{x})\right\rangle\right] + C
\end{aligned}$$

Looking at the second term

$$\begin{aligned}
\int\left\langle\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}),\, \mathbf{s}_\theta(\boldsymbol{x})\right\rangle p(\boldsymbol{x})\mathrm{d}\boldsymbol{x} &= \int\left\langle\nabla_{\boldsymbol{x}} p(\boldsymbol{x}),\, \mathbf{s}_\theta(\boldsymbol{x})\right\rangle\mathrm{d}\boldsymbol{x} \\
\text{via the divergence theorem} \quad &= -\int p(\boldsymbol{x})\left[\nabla_{\boldsymbol{x}}\cdot\mathbf{s}_\theta(\boldsymbol{x})\right]\mathrm{d}\boldsymbol{x} \\
&= -\,\mathbb{E}_{p(\boldsymbol{x})}\left[\nabla_{\boldsymbol{x}}\cdot\mathbf{s}_\theta(\boldsymbol{x})\right]
\end{aligned}$$

Assuming that $\|p(\boldsymbol{x})\boldsymbol{s}_\theta(\boldsymbol{x})\| \to 0$ as $\|\boldsymbol{x}\| \to \infty$

Learning Stein score in high dimensions

Taking the *divergence* of the score $\nabla_{\boldsymbol{x}} \cdot \mathbf{s}_\theta(x)$ cost grows with $\mathcal{O}(d)$.

Sliced score matching (SSM) (Song, Garg, et al., 2019) alleviates this with random projections:

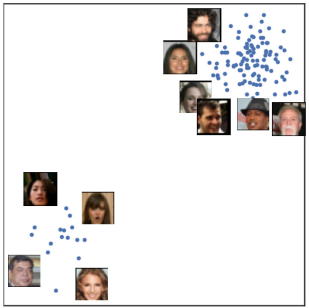
$$\ell_{\text{ssm}}(\mathbf{s}_\theta) \triangleq \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\, \mathbb{E}_{\boldsymbol{v}\sim p(\boldsymbol{v})}\left[\left|\boldsymbol{v}^\top \nabla \log p(\boldsymbol{x}) - \boldsymbol{v}^\top \mathbf{s}_\theta(\boldsymbol{x})\right|^2\right]. \tag{10}$$

We can show this has an equivalent form to the implicit score matching objective.

$$\ell_{\text{ssm}}(\mathbf{s}_\theta) = \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\, \mathbb{E}_{\boldsymbol{v}\sim p(\boldsymbol{v})}\left[\boldsymbol{v}^\top \mathrm{D}\, \mathbf{s}_\theta(\boldsymbol{x})\boldsymbol{v} + \tfrac{1}{2}\left\|\boldsymbol{v}^\top \mathbf{s}_\theta(\boldsymbol{x})\right\|^2\right] + C = \ell_{\text{ism}}(\mathbf{s}_\theta) + C \tag{11}$$

where $\mathbb{E}_{\boldsymbol{v}\sim p(\boldsymbol{v})}\left[\boldsymbol{v}^\top\left[\nabla \cdot \mathbf{s}_\theta(\boldsymbol{x})\right]\boldsymbol{v}\right] = \mathrm{Tr}(\mathrm{D}\,\mathbf{s}_\theta)(\boldsymbol{x}) = \nabla \cdot \mathbf{s}_\theta(\boldsymbol{x})$ is just Hutchinson's trace trick.

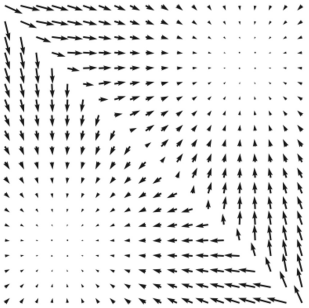
The picture so far



Data samples

$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$

score matching



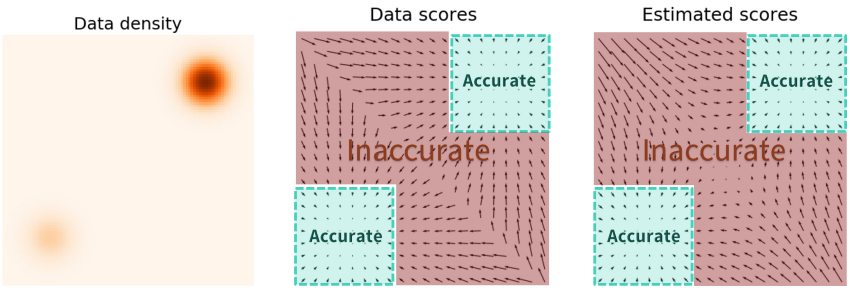
Scores

$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x})$

Langevin dynamics

New samples

But this does not quite work...



Issues:

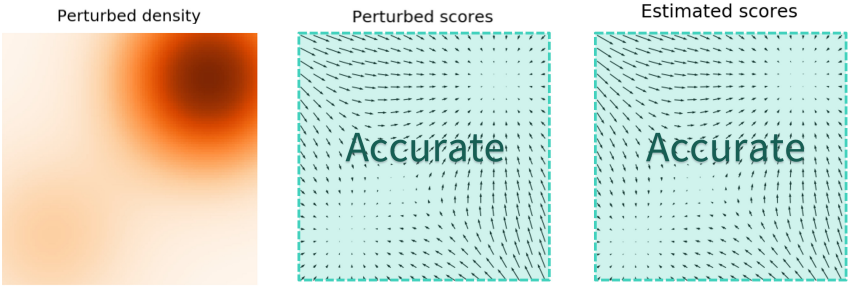
- *Poor score approximation* outside the support of p.
- *Slow mixing* with Langevin algorithm (non-convex (Eberle, 2016)).

Noising and denoising

Denoising Score Matching (Vincent, 2011)

Solution: Smoothing data density / spreading samples by adding *noise* to the data!

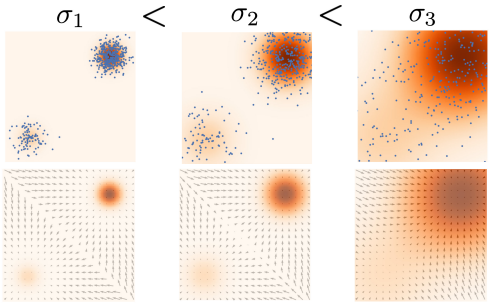
$$p_\sigma(\tilde{\boldsymbol{x}}) \triangleq \int p_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x};\sigma)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}, \quad \ell_{\mathrm{dsm}}(\mathbf{s}_\theta) \triangleq \mathbb{E}_{\boldsymbol{x}\sim p_\sigma}\left[\left\|\nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}) - \mathbf{s}_\theta(\boldsymbol{x})\right\|^2\right].$$



Typically $p_\sigma(\tilde{\boldsymbol{x}}|\boldsymbol{x}) = \mathcal{N}(\tilde{\boldsymbol{x}}|\boldsymbol{x},\sigma^2)$. Unfortunately, targeting wrong density as $p_\sigma(\tilde{\boldsymbol{x}}) \neq p(\boldsymbol{x})$, trade-off between small and large value of σ.

Multiple noise perturbations (Song and Ermon, 2019; Song and Ermon, 2020)

Solution: Using *multiple* scales $\sigma_0 < \cdots < \sigma_T$: $p_{\sigma_t}(\tilde{\boldsymbol{x}}) \triangleq \int p_{\sigma_t}(\tilde{\boldsymbol{x}}|\boldsymbol{x};\sigma_t)p(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$.



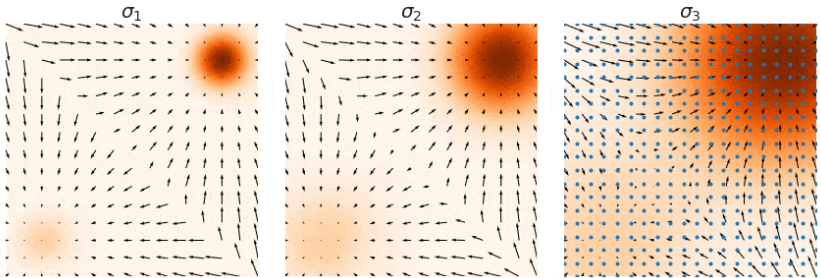
Score matching with Langevin dynamics (SMLD): Parametrise a score network $s_\theta(\sigma_t, \boldsymbol{x})$ indexed by the noise scale σ_t:

$$\ell_{\mathrm{smld}}(\mathbf{s}_\theta) \triangleq \sum_{t=0}^{T} \lambda(t)\, \mathbb{E}_{\boldsymbol{x}\sim p_{\sigma_t}(\boldsymbol{x})}\left[\left\|\nabla_{\boldsymbol{x}} \log p_{\sigma_t}(\boldsymbol{x}) - \mathbf{s}_\theta(\sigma_t, \boldsymbol{x})\right\|^2\right]. \tag{12}$$

Sampling from multiple noise scales

Sample with **annealed Langevin** dynamics:

1. Start with large σ_T and target p_{σ_T} with Langevin dynamics.
2. Decrease noise $\sigma_{T-1} < \sigma_T$ and *warm-start* with previous samples:
3. Repeat procedure until σ_0 is very small so that $p_{\sigma_0} \approx p$.

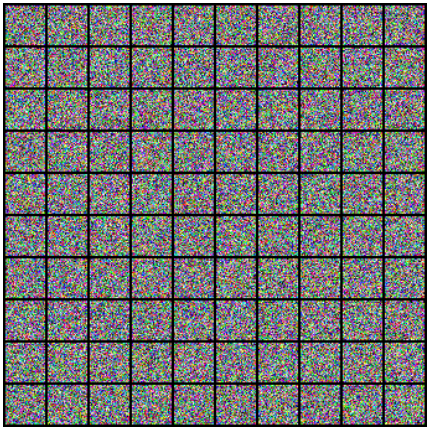


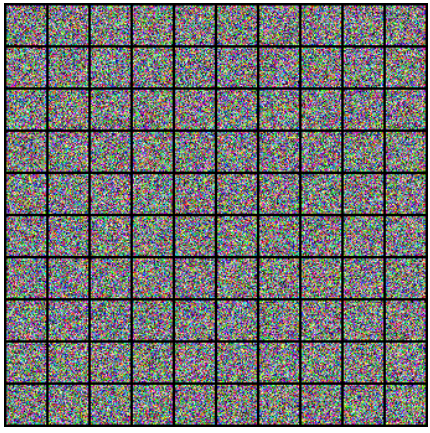
Annealing Langevin dynamics

Algorithm 1 Sampling of annealing Langevin dynamics

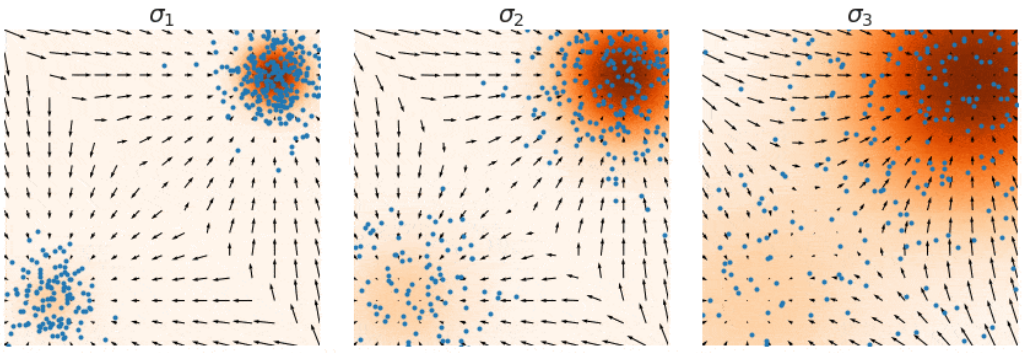
Input: $\{\sigma_t\}_{t=1}^T$, $\{\gamma_t\}_{t=1}^T$, K
Initialize $\mathbf{X}_T^0 \sim \mathcal{N}(0, \sigma_T \operatorname{Id})$.
for $t = T$ to 1 **do**
    for $k = 0$ to $K - 1$ **do**
        Sample $\mathbf{X}_t^{k+1} = \mathbf{X}_t^k + \gamma_t \, \mathbf{s}_\theta \, (\sigma_t, \mathbf{X}_t^k) + \sqrt{2\gamma_t} Z_t^{k+1}$
    $X_{t-1}^0 = X_t^K$

This really works!





Recap: SMLD



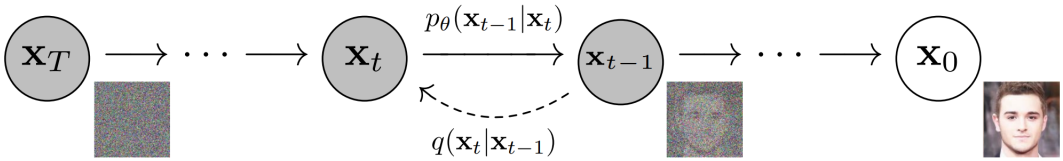
- Choose increasing sequence of **noise** σ_t.
- Construct noised distribution p_{σ_t}.
- Fit amortised score network $\mathbf{s}_\theta(\sigma_t, x_t)$ to approximate true score $\nabla \log p_{\sigma_t}(x_t)$.
- Sample with **annealed Langevin** dynamics.

Discrete SGMs

Discrete SGMs: Noising process

Discrete SGMs (Sohl-Dickstein et al., 2015; Ho et al., 2020; De Bortoli et al., 2021) have a similar principle to the multi-scale score matching.

A **forward process**, $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ (pictured, $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$), takes data and turns it into noise.



We then want to **reverse** this process to be able to turn noise back into data! We need to *learn* $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$.

Choosing a forward process

First, we need to choose a **forward transition** $p(x_t|x_{t-1})$

$$p(x_{0:N}) = p(x_0) \prod_{k=1}^{N} p(x_t|x_{t-1}). \tag{13}$$

Example of transition:

- Set the transition to be $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t|(1-\gamma_t)\boldsymbol{x}_{t-1}, 2\gamma_t\mathbf{I})$.
- This form lets us sample $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-k})$ analytically.
- If we take $t \to \infty$, $p_t \to \mathcal{N}(0, \mathbf{I})$ geometrically quickly.
- This is important! It means we can approximately sample $p(\boldsymbol{x}_T)$ for large T.

Reversing the forward process

We now want to be able to **invert** this transition, so we can sampling the reverse process.

Unfortunately $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1})/p(\boldsymbol{x}_t)$ is intractable!

But using some Taylor expansions and approximations, we can show

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx C_{\gamma_t} \exp\left[-\frac{\left\|\boldsymbol{x}_{t-1} - (1+\gamma_t)\boldsymbol{x}_t - 2\gamma_t \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)\right\|^2}{4\gamma_t^2}\right].$$

So we have

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx \mathcal{N}\Big(\boldsymbol{x}_{t-1}\Big|(1+\gamma_t)\boldsymbol{x}_t + 2\gamma_t \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t), 2\gamma_t \mathbf{I}\Big).$$

As $\gamma_t \to 0$, this becomes exact.

Discrete SGMs: Denoising Score Matching (Vincent, 2011)

The score $\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t)$ is unfortunately **intractable**. By using the identity $p(\boldsymbol{x}_t) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_0)p(\boldsymbol{x}_0)\mathrm{d}\boldsymbol{x}_0$ we can get (Efron, 2011)

$$\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t) = \int \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\ p(\boldsymbol{x}_0|\boldsymbol{x}_t)\,\mathrm{d}\boldsymbol{x}_0 = \underbrace{\mathbb{E}_{p(\boldsymbol{x}_0|\boldsymbol{x}_t)}}_{\text{non tractable}}[\underbrace{\nabla \log p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0)}_{\text{tractable}}].$$

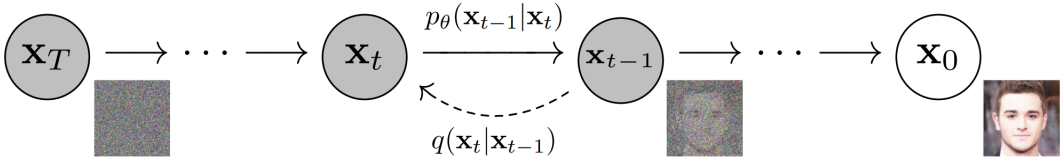
$\nabla \log p_t(\boldsymbol{x}_t) = \mathbb{E}_{\boldsymbol{x}_0|\boldsymbol{x}_t}\Big[\nabla \log p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0)|\boldsymbol{x}_t\Big]$ is a **conditional expectation** hence by definition

$$\nabla \log p_k = \arg\min\Big\{\mathbb{E}_{\boldsymbol{x}_k,\boldsymbol{x}_0}\Big[\|\ \mathbf{s}\ (\boldsymbol{x}_k) - \log p_{k|0}(\boldsymbol{x}_k|\boldsymbol{x}_0)\|^2\Big] :\ \mathbf{s}\ \in \mathrm{L}^2(p_k)\Big\}. \tag{14}$$

We then need to estimate the score over all the steps, so amortise $\boldsymbol{s}_\theta(\boldsymbol{x}_t) \to \boldsymbol{s}_\theta(t, \boldsymbol{x}_t)$ and take a weighted loss over all steps:

$$\ell_{\mathrm{dsm}}(\ s_\theta\ , \lambda) = \mathbb{E}_{p(t)}\Big[\lambda(t)\,\mathbb{E}_{p(\boldsymbol{x}_0)p(\boldsymbol{x}_t|\boldsymbol{x}_0)}\Big[\|\boldsymbol{s}_\theta(t, \boldsymbol{x}_t) - \ \nabla \log p_{t|0}(\boldsymbol{x}_t|\boldsymbol{x}_0)\ \|^2\Big]\Big].$$

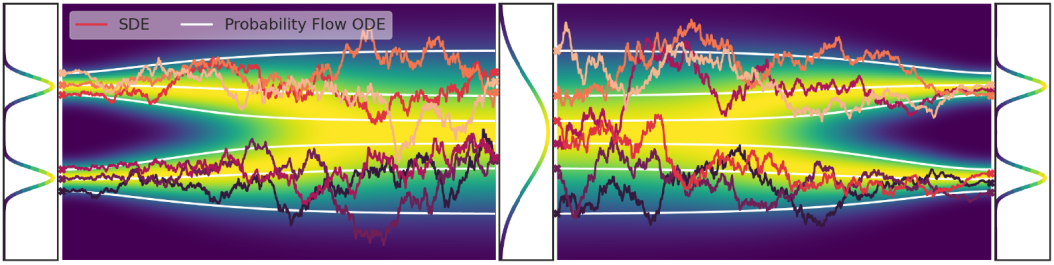
Recap: Discrete SGMs



- Choose forward **Markov kernel** $\Rightarrow$ induces noising **Markov process** $(\mathbf{X}_t)_{t\in[1,\dots,T]}$.
- Aim: Sample from **time-reverse** process $(\mathbf{Y}_t)_{t\in[1,\dots,T]} = (\mathbf{X}_{T-t})_{t\in[1,\dots,T]}$.
  - Requires $\Rightarrow$ approximating **backward Markov kernel**.
  - Its variance is approximately the same as the forward kernel's.
  - Its mean depends on the forward kernel's mean and the **Stein score**.
  - The score is parametrised and trained by **learning to 'denoise'** samples.
  - Generate samples via **ancestral sampling** on the backward process.

Continuous score-based models

Principles of continuous diffusion models



Why going to the continuous setting? 1/ shed a new light on discrete SGMs 2/ easier quantitative bounds 3/ likelihood evaluation.

- Idea: Use a *continuous* series of noise scales!
- Do this by constructing an **SDE** forward noising process $(\mathbf{X}_t)_{t\in[0,T]}$.
- Have this noising converge to a **known distribution**.
- **Invert** this SDE noising process to get $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_t$.

Continuous noising processes

A wide class of SDEs can be written as:

$$\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t, \mathbf{X}_t)\,\mathrm{d}\mathbf{B}_t. \tag{15}$$

A couple of common examples:

	Brownian Motion	Ornstein-Ulhenbeck process
$b(t, \mathbf{X}_t)$	0	$-\mathbf{X}_t$
$\sigma(t, \mathbf{X}_t)$	1	$\sqrt{2}$
Invariant Measure	Lebesgue	$\mathcal{N}(0, 1)$ (w.r.t Leb.)
Conditional $\mathbf{X}_t \vert \mathbf{X}_0$	$\mathbf{X}_0 + \mathbf{B}_t$	$e^{-t}\mathbf{X}_0 + \mathbf{B}_{1-e^{-2t}}$

Euler–Maruyama discretisation with time step γ_t yields a Markov kernel:

$$p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}) \approx \mathcal{N}(\boldsymbol{x}_t | \boldsymbol{x}_{t-1} + \gamma_t\, b(t, \boldsymbol{x}_{t-1})\,, \gamma_t\, \sigma(t, \boldsymbol{x}_t)^2 \mathbf{I}).$$

Discrete SGMs as discretised continuous SGMs (Song, Sohl-Dickstein, et al., 2021)

These things look oddly familiar!

	SMLD (Song and Ermon, 2019)	DDPM (Ho et al., 2020)
$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$	$\mathcal{N}(\boldsymbol{x}_{t-1}, \sigma_t^2 - \sigma_{t-1}^2)$	$\mathcal{N}((1-\gamma_t)\boldsymbol{x}_{t-1}, 2\gamma_t \mathbf{I})$
Cont. $b(t, \mathbf{X}_t)$	0	$-\mathbf{X}_t$
Cont. $\sigma(t, \mathbf{X}_t)$	$\sqrt{\frac{\mathrm{d}\sigma^2(t)}{\mathrm{d}t}}$	$\sqrt{2}$
$p(\boldsymbol{x}_T)$, T is 'big'	$\mathcal{N}\left(0, \int_0^T \sigma^2(t)\mathrm{d}t\mathbf{I}\right)$	$\mathcal{N}(0, \mathbf{I})$

Continuous score-based models: Time reversal process

Theorem 3: (Cattiaux et al., 2021; Haussmann and Pardoux, 1986)

Under mild conditions on p_0, the time-reversed process $(\mathbf{Y}_t)_{t\geq 0} = (\mathbf{X}_{T-t})_{t\in[0,T]}$, with forward process $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t$, also satisfies an SDE given by

$$\mathrm{d}\mathbf{Y}_t = \Big[-b(T-t, \mathbf{Y}_t) + \sigma(T-t)^2 \nabla \log p_{T-t}(\mathbf{Y}_t) \Big]\,\mathrm{d}t + \sigma(T-t)\,\mathrm{d}\mathbf{B}_t,$$

assuming $\mathbf{Y}_0$ is distributed the same as $\mathbf{X}_T$.

Training the score network $\mathbf{s}_\theta : \mathbb{R}_+ \times \mathbb{R}^d \to \mathbb{R}^d$ is the same as in the discrete setting, with the (denoising) score matching loss, but with continuous time.

$$\ell_{\mathrm{dsm},\lambda}(\mathbf{s}_\theta) = \mathbb{E}_{p(t)p(\mathbf{X}_0)p(\mathbf{X}_t|\mathbf{X}_0)} \Big[\lambda(t) \Big\| \mathbf{s}_\theta(t, \mathbf{X}_t) - \nabla_{\mathbf{X}_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0) \Big\|^2 \Big].$$

Sampling from SDEs

Euler–Maruyama discretisation with some time step γ_t gives an approximate trajectory of the SDE

$$\mathbf{Y}_{t+1} \approx \mathbf{Y}_t + \gamma_t \big[b(T-t, \mathbf{Y}_t) + \sigma(T-t, \mathbf{Y}_t)^2 \underbrace{\nabla \log p_{T-t}(\mathbf{Y}_t)}_{\approx \mathbf{s}_\theta(T-t, \mathbf{Y}_t)} \big] + \sqrt{\gamma_t}\, \sigma(T-t, \mathbf{Y}_t)\, \mathbf{Z}_t.$$

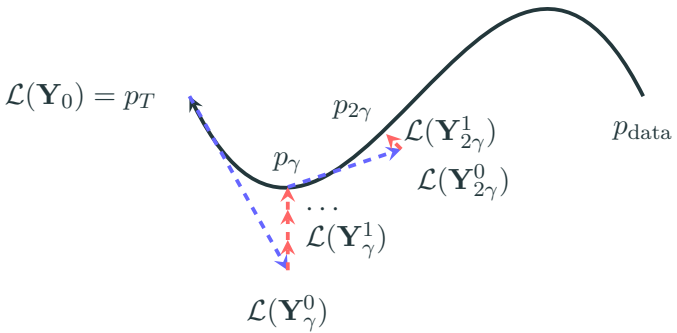
This discretisation however leads to some error in the sampling from the SDE. We can **correct** this error by running Langevin dynamics after this step targeting the distribution $\mathcal{L}(\mathbf{X}_t)$. We run these dynamics along another time axis s with the (time-reversal) SDE

$$\mathrm{d}\mathbf{Y}_t^s = \nabla \log p(\mathbf{Y}_t^s)\, \mathrm{d}s + \sqrt{2}\mathrm{d}\mathbf{B}_s$$

which can be discretised as

$$\mathbf{Y}_t^s \approx \mathbf{Y}_t^{s-1} + \gamma_{t,s} \nabla \log p(\mathbf{Y}_t^{s-1}) + \sqrt{2\gamma_{t,s}}\mathbf{Z}_t^s, \quad \mathbf{Z}_t^s \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

Predictor-Corrector sampling



The black line corresponds to the dynamics of the noising process $(p_t)_{t\in[0,T]}$. The blue dashed lines correspond to the predictor step (going backward in time) and the red dashed lines correspond to the corrector step (projecting back onto the forward dynamics).

Predictor-Corrector sampling from the backwards process

Algorithm 2 Predictor-Corrector

Require: $\mathbf{Y}_0$, T, N, S, $\gamma = T/N$, γ_s

for $k \in \{0, \dots, N-1\}$ **do**
  /// PREDICTOR STEP
  $\mathbf{Z}_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Standard Gaussian noise
  $\mathbf{Y}_{k+1} = \mathbf{Y}_k + \gamma[-b(T - k\gamma, \mathbf{Y}_k) + \nabla \log p_{T-k\gamma}(\mathbf{Y}_k)] + \sqrt{\gamma}\mathbf{Z}_{k+1}$ ▷ E-M step
  /// CORRECTOR STEP
  $\mathbf{Y}^0_{k+1} = \mathbf{Y}_{k+1}$
  for $s \in \{0, \dots, S-1\}$ **do**
    $\mathbf{Z}^s_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Standard Gaussian noise
    $\mathbf{Y}^{s+1}_{k+1} = \mathbf{Y}^s_{k+1} + \gamma_s \nabla \log p_{T-k\gamma}(\mathbf{Y}^s_{k+1}) + \sqrt{2\gamma_s}\mathbf{Z}^s_{k+1}$ ▷ Langevin step
  $\mathbf{Y}_{k+1} = \mathbf{Y}^S_{k+1}$
return $\mathbf{Y}_N$

Important 'tricks' - The SDE

Noise scheduling: For whatever SDE we are looking at, remap

- $b(t, \mathbf{X}_t) \rightarrow \beta(t)\, b(t, \mathbf{X}_t)$
- $\sigma(t) \rightarrow \sqrt{\beta(t)}\, \sigma(t)$

Doing this is equivalent to *re-scaling time* such that $t \mapsto \int_0^t \beta(s) ds$. We can set $\beta(t)$ to reduce the step size where the score norm is high.

Process truncation: As $t \rightarrow 0$, the score blows up, i.e. $\| \nabla \log p(x_t) \| \rightarrow \infty$, with the manifold hypothesis (De Bortoli, 2022), or we only have access to finite data.

Truncating the process so that $t \in [\epsilon, T]$ prevents this issue. This is equivalent to smoothing the data with some small noise, effectively extending the support of the data distribution to $\mathbb{R}^d$.

Important 'tricks' - Learning the score

Parametrising the score: Learning the score naively can prove tricky. There is a principled way to parameterise the score however!

$$\mathrm{d}\mathbf{Y}_t = \Big[-b(T-t, \mathbf{Y}_t) + \sigma(T-t)^2 \boldsymbol{s}_\theta(T-t, \mathbf{Y}_t) \Big] \mathrm{d}t + \sigma(T-t)\, \mathrm{d}\mathbf{B}_t$$

We set

$$\boldsymbol{s}_\theta(t, \mathbf{Y}_t) = \boldsymbol{h}_\theta(t, \mathbf{Y}_t)/\sigma_t + 2\, b(t, \mathbf{Y}_t) \,/\, \sigma(t)^2, \quad \sigma_t = \mathbb{E}\Big[\|\nabla \log p(\mathbf{X}_t | \mathbf{X}_0)\|^2 \Big]^{1/2}.$$

1. This sets the dynamics of the backwards SDE to be the same as the forward if $\boldsymbol{h}_\theta(t, \mathbf{Y}_t) = 0$.
2. This normalises $\boldsymbol{h}_\theta(t, \mathbf{Y}_t)$ to have expected norm of 1.

We can show that $\mathbb{E}\big[\|\nabla \log p(\mathbf{X}_t | \mathbf{X}_0)\|^2 \big]^{1/2} = \mathrm{Std}[\mathbf{X}_t | \mathbf{X}_0] = \sigma_t$.

Important 'tricks' - Learning the score

The denoising score matching loss has high variance when approximated with Monte Carlo.

$\implies$ Using a exponential moving average of the parameters at test time is very impactful.

Likelihood evaluation and connection with continuous normalising flows

Fokker-Planck equation

Given the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \Sigma^{1/2}(t, \mathbf{X}_t)\,\mathrm{d}\mathbf{B}_t$, the **Fokker-Planck** equation describes the evolution of the density

$$\frac{\partial}{\partial t} p_t(x) = -\mathrm{div}\left(b(t,\cdot)\, p_t(\cdot)\right)(x) + \frac{1}{2}\sum_{i,j} \frac{\partial^2}{\partial_i \partial_j}\left(\Sigma_{i,j}(t,\cdot)\, p_t(\cdot)\right)(x). \tag{16}$$

- If $\Sigma = 0$ (deterministic dynamics):

$$\frac{\partial}{\partial t} p_t(x) = -\mathrm{div}\left(b(t,\cdot)\, p_t(\cdot)\right)(x).$$

- If $\Sigma = \sigma^2(t)\mathbf{I}$ (Langevin dynamics):

$$\begin{aligned}\frac{\partial}{\partial t} p_t(x) &= -\mathrm{div}\left(b(t,\cdot)\, p_t(\cdot)\right)(x) + \tfrac{1}{2}\,\sigma(t)^2 \Delta p_t(x) \\ &= -\mathrm{div}\left(\left[b(t,\cdot) - \tfrac{1}{2}\,\sigma(t)^2\, \nabla \log p_t(\cdot)\right] p_t(x)\right).\end{aligned}$$

Fokker-Planck equation (Cont'd)

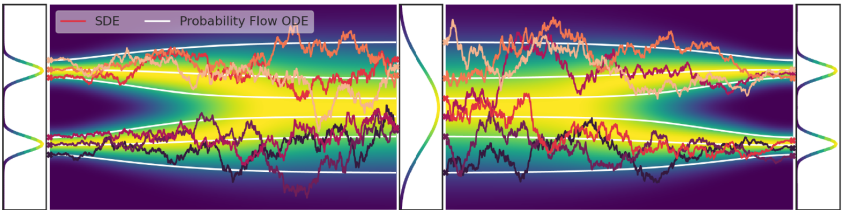
Then both of the following dynamics

1. $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t$ (stochastic).
2. $\mathrm{d}\mathbf{X}_t = \left[b(t, \mathbf{X}_t) - \frac{1}{2}\sigma(t)^2 \nabla \log p_t(\mathbf{X}_t) \right] \mathrm{d}t$ (deterministic).

have the same marginal density $\mathbb{P}_t \triangleq \mathcal{L}(\mathbf{X}_t)$ which evolution is given by

$$\frac{\partial}{\partial t} p_t(x) = -\mathrm{div}\left(\left[b(t, \cdot) - \frac{1}{2}\sigma(t)^2 \nabla \log p_t(\cdot) \right] p_t(\cdot) \right)(x).$$

This gives us a *deterministic* ODE with the same marginal density as the SDE.



Log-likelihood evolution of ODEs

Assume a **deterministic** evolution of $\mathbf{X}_t$ given by the ODE

$$\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t. \tag{17}$$

The evolution of the log-density is given by (Chen et al., 2018)

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(x) = -\mathrm{div}\left(b(t, \cdot)\right)(x). \tag{18}$$

Assuming that $\mathbf{X}_T \sim p_T$, the **log-likelihood** can be computed as

$$\log p_0(\mathbf{X}_0) = \log p_T(\mathbf{X}_T) + \int_0^T \mathrm{div}\left(b(t, \cdot)\right)(\mathbf{X}_t)\mathrm{d}t. \tag{19}$$

Log-likelihood evaluation of ODEs

The following **augmented ODE** allows to solve at once the trajectory of $\mathbf{X}_t$ and the change in log-likelihood

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\mathbf{X}_t \\ \log p(\mathbf{X}_t)\end{bmatrix} = \begin{bmatrix} b_\theta(t,\cdot) \\ -\mathrm{div}\left(b_\theta(t,\cdot)\right)\end{bmatrix}(\mathbf{X}_t). \tag{20}$$

Which can be estimated numerically with a myriad of (adaptive) ODE solvers.

This is exactly how **continuous normalising flows** (Chen et al., 2018; Grathwohl et al., 2019) are trained. Maximising the likelihood ($\mathcal{O}(Nd^2)$ or $\mathcal{O}(Nd)$ with div estimator)

$$\mathbb{E}\left[\log p_0(\mathbf{X}_0)\right] = \mathbb{E}[\log p_T(\mathbf{X}_T) - \int_0^T \mathrm{div}(\, b_\theta(t,\mathbf{X}_t)\,)\mathrm{d}t]. \tag{21}$$

Probability flow (Song, Sohl-Dickstein, et al., 2021)

We can apply this likelihood evaluation method for continuous SGMs induced by

$$\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t \tag{22}$$

since it has the same marginal density has the ODE

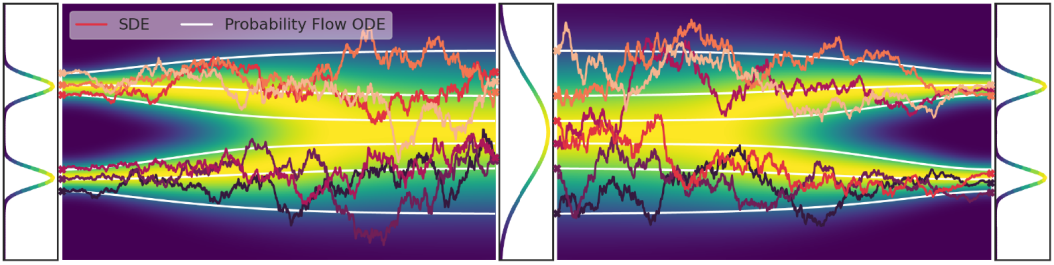
$$\mathrm{d}\mathbf{X}_t = \left[b(t, \mathbf{X}_t) - \tfrac{1}{2}\,\sigma(t)^2\,\nabla \log p_t(\mathbf{X}_t) \right] \mathrm{d}t.$$

We have the associated augmented ODE

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \mathbf{X}_t \\ \log p(\mathbf{X}_t) \end{bmatrix} = \begin{bmatrix} b(t, \mathbf{X}_t) - \tfrac{1}{2}\,\sigma(t)^2\,\nabla \log p_t(\mathbf{X}_t) \\ -\mathrm{div}\left(b(t, \mathbf{X}_t) - \tfrac{1}{2}\,\sigma(t)^2\,\nabla \log p_t(\mathbf{X}_t) \right) \end{bmatrix} (\mathbf{X}_t). \tag{23}$$

We then just have to add on the log likelihood of the reference density, $\log p_T(\mathbf{X}_T)$.

Recap: Continuous SGMs



- Continuously **noise** data samples with forward SDE
- Aim: time-reverse this process $\Rightarrow$ **denoising** process
  - Same variance as the forward process
  - Its mean depends on the forward process's mean and the **Stein score**
  - The score is parametrised and trained by learning to 'denoise' samples
  - Generate samples by discretising the (approximate) backward process with Euler-Maruyama

A variational perspective

Discrete SGMs: ELBO (Sohl-Dickstein et al., 2015; Ho et al., 2020)

We assume a discrete process induced by the **forward transition** $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$:

$$q(\boldsymbol{x}_{0:T}) = q(\boldsymbol{x}_0)\prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}). \tag{24}$$

The (intractable) **backward transition** $p_{t-1|t}(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ induces a backward process

$$p(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T)\prod_{t=T}^{1} p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t). \tag{25}$$

A bound on the (negative) **log-likelihood** can be derived as

$$\mathbb{E}[-\log p_\theta(\boldsymbol{x}_0)] \leq \mathbb{E}_q\left[-\log \frac{p_\theta(\boldsymbol{x}_{0:T})}{q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0)}\right] = \mathbb{E}_q\left[-\log p(\boldsymbol{x}_T) - \sum_{t\geq 1}\log \frac{p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)}{q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})}\right] \triangleq \mathcal{E}$$

which can straightforwardly be estimated via Monte Carlo sampling.

A more efficient EBLO

$$\mathcal{E} = \mathbb{E}_q \left[\underbrace{D_{KL}(q(\boldsymbol{x}_T|\boldsymbol{x}_0)||p(\boldsymbol{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t))}_{L_{t-1}} \underbrace{- \log p_\theta(\boldsymbol{x}_0|\boldsymbol{x}_1)}_{L_0} \right].$$

L_T *prior matching*: constant if we fix q

L_0 *reconstruction*: can compute directly

L_{t-1} *denoising matching*: requires access to $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0)$

Denoting $\boldsymbol{\mu}_q(x_t, x_0) = \mathbb{E}[\mathbf{X}_t|\mathbf{X}_0]$ and $\sigma_q(t) = \text{Std}[\mathbf{X}_t|\mathbf{X}_0]$

Choosing $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \approx p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \triangleq \mathcal{N}\left(\boldsymbol{x}_{t-1} \middle| \boldsymbol{\mu}_\theta(t, \boldsymbol{x}_t), \sigma_q^2(t)\,\text{Id}\right)$ we have

$$\Rightarrow \arg\min_\theta D_{KL}(q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)||p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)) = \arg\min_\theta \tfrac{1}{2}\tfrac{1}{\sigma_q^2(t)} \left[\|\boldsymbol{\mu}_\theta(t, \boldsymbol{x}_t) - \boldsymbol{\mu}_q(\boldsymbol{x}_t, \boldsymbol{x}_0)\|^2 \right]$$

$$= \arg\min_\theta \tfrac{1}{2}\tfrac{1}{\sigma_q^2(t)} C(t) \left[\| \mathbf{s}_\theta(t, \boldsymbol{x}_t) - \nabla \log q(\boldsymbol{x}_t|\boldsymbol{x}_0) \|^2 \right].$$

Continuous SGMs: Score matching as ELBO

We now assume a continuous process $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t$ with reversal

$$\mathrm{d}\mathbf{Y}_t = \Big[-b(T-t, \mathbf{Y}_t) + \sigma^2(T-t)\,\nabla \log p_{T-t}(\mathbf{Y}_t) \Big]\,\mathrm{d}t + \sigma(T-t)\,\mathrm{d}\mathbf{B}_t. \tag{26}$$

Theorem 4: (Song, Durkan, et al., 2021)

Under some regularity assumptions, setting the weighting to $\lambda(t) = \sigma^2(t)$:

$$\mathrm{D}_{\mathsf{KL}}(p|p_\theta) \leq \ell_{\mathrm{sm}}(\,\mathbf{s}_\theta\,; \sigma^2(\cdot)) + \mathrm{D}_{\mathsf{KL}}(p_T|p_{\mathsf{ref}}) = \ell_{\mathrm{dsm}}(\,\mathbf{s}_\theta\,; \sigma^2(\cdot)) + C + \mathrm{D}_{\mathsf{KL}}(p_T|p_{\mathsf{ref}}). \tag{27}$$

$$\mathbb{E}_{p_0(\boldsymbol{x})}[-\log p_\theta(\boldsymbol{x})] \leq \ell_{\mathrm{sm}}(\,\mathbf{s}_\theta\,; \sigma^2(\cdot)) + C_1 = \ell_{\mathrm{dsm}}(\,\mathbf{s}_\theta\,; \sigma^2(\cdot)) + C_2.$$

ELBOs for single datapoints

Theorem 5: (Song, Durkan, et al., 2021; Huang et al., 2021)

$$-\log p_\theta(\boldsymbol{x}) \leq \mathcal{L}_\theta^{SM}(\boldsymbol{x}) = \mathcal{L}_\theta^{DSM}(\boldsymbol{x})$$

$$\mathcal{L}_\theta^{SM}(\boldsymbol{x}_0) = \overbrace{-\mathbb{E}_{p(\boldsymbol{x}_T|\boldsymbol{x}_0)}[\log p_{ref}(\boldsymbol{x}_T)]}^{\text{const.}} + \frac{1}{2}\int_0^T \mathbb{E}_{p(\boldsymbol{x}_t|\boldsymbol{x}_0)}\Bigg[\underbrace{2\,\sigma(t)^2 \nabla_{\boldsymbol{x}_t} \cdot s_\theta(t, \boldsymbol{x}_t) + \sigma(t)^2 \left\| s_\theta(t, \boldsymbol{x}_t) \right\|^2}_{\text{implicit score matching}} \underbrace{- 2\nabla_{\boldsymbol{x}_t} \cdot b(t, \boldsymbol{x}_t)}_{\text{const.}}\Bigg] \mathrm{d}t$$

ELBOs for single datapoints

Theorem 6: (Song, Durkan, et al., 2021; Huang et al., 2021)

$$-\log p_\theta(\boldsymbol{x}) \leq \mathcal{L}_\theta^{SM}(\boldsymbol{x}) = \mathcal{L}_\theta^{DSM}(\boldsymbol{x})$$

$$\mathcal{L}_\theta^{DSM}(\boldsymbol{x}_0) = \overbrace{-\mathbb{E}_{p(\boldsymbol{x}_T|\boldsymbol{x}_0)}[\log p_{ref}(\boldsymbol{x}_T)]}^{\text{const.}}$$

$$+ \quad \frac{1}{2}\int_0^T \mathbb{E}_{p(\boldsymbol{x}_t|\boldsymbol{x}_0)}\Big[\underbrace{\sigma(t)^2 \left\| s_\theta(t, \boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\right\|^2}_{\text{denoising score matching}}\Big]\,\mathrm{d}t$$

$$- \quad \frac{1}{2}\int_0^T \mathbb{E}_{p(\boldsymbol{x}_t|\boldsymbol{x}_0)}\Big[\underbrace{\sigma(t)^2 \|\nabla_{\boldsymbol{x}_t} \log p(\boldsymbol{x}_t|\boldsymbol{x}_0)\|^2 + 2\nabla_{\boldsymbol{x}_t} \cdot b(t, \boldsymbol{x}_t)}_{\text{const.}}\Big]\,\mathrm{d}t$$

Active research directions

Active research directions

- *Accelerate* reverse sampling
- *Structured data*: text, graph, discrete, manifold, protein, functions etc
- *Latent* manipulation
- *Scaling* for large models
- *Theory*: how come it works so well?

References

P. Cattiaux, G. Conforti, I. Gentil, and C. Léonard. Time reversal of diffusion processes under a finite entropy condition. *arXiv preprint arXiv:2104.07708*, 2021. Cited on page 33.

R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018. Cited on pages 43, 44.

V. De Bortoli. Convergence of Denoising Diffusion Models under the Manifold Hypothesis. Aug. 10, 2022. DOI: 10.48550/arXiv.2208.05314. Cited on page 37.

V. De Bortoli, J. Thornton, J. Heng, and A. Doucet. Diffusion Schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on page 24.

A. Eberle. Reflection couplings and contraction rates for diffusions. *Probability theory and related fields*, 166(3):851–886, 2016. Cited on page 15.

B. Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. Cited on page 27.

W. Grathwohl, R. T. Q. Chen, J. Bettencourt, and D. Duvenaud. Scalable Reversible Generative Models with Free-Form Continuous Dynamics. In *International Conference on Learning Representations*, 2019. URL: https://openreview.net/forum?id=rJxgknCcK7. Cited on page 44.

U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986. Cited on page 33.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. Cited on pages 24, 32, 48.

C.-W. Huang, J. H. Lim, and A. C. Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on pages 51, 52.

A. Hyvärinen. Estimation of Non-Normalized Statistical Models by Score Matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL: `http://jmlr.org/papers/v6/hyvarinen05a.html`. Cited on page 11.

J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. Cited on pages 24, 48.

Y. Song, C. Durkan, I. Murray, and S. Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34, 2021. Cited on pages 50–52.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on pages 18, 32.

Y. Song and S. Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. Cited on page 18.

Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced Score Matching: A Scalable Approach to Density and Score Estimation. May 2019. Cited on page 13.

Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, 2021. Cited on pages 32, 45.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. Cited on pages 17, 27.