Score-based Generative Models on Euclidean space and Riemannian manifolds

Michael Hutchinson & Émile Mathieu

October 30, 2022

What are we going to talk about?

This will be a talk of three parts:

1. A brief introduction on deep generative models.
2. A tutorial and exposition of *score-based generative models*, a new (since 2019, although the ideas are older) method of generative modelling behind many of the State of the Art results today.
3. A talk on work done by speakers (+others!) on extending the formulation of score-based generative modelling to data living on *Riemannian manifolds.*

Deep generative models

What is generative modelling?

Given some samples from a density, we would like to fit the distribution of these samples, and make more samples from the distribution.[Do this in pictures with some images + some generated examples]

Note fitting a Gaussian distribution to samples is *technically* generative modelling, albeit a very simple version.

What is generative modelling?

Slide in the style of Valentin, although EBMs do not fit this!

Motivating examples

1.

Deep generative models

A number of model type innovate on this to restrict the model to be normalised.

- VAEs - a probablistic auto encoder
- Normalizing flows - series of invertable transforms with computable log det jacobians to we can compute the lieklihood through the chain of chain rules.

The restrictions placed on these models are quite strong, and attaining tractable maximum likelihood objectives is hard, and often relies on ELBOs.

[expand into more slides]

A narrative of generative model development

We also have *implicit models* such as GANs which involve adviserial training.
Notoriously fiddly to train and can just collapse

Score-based generative models

Motivating examples

1. StableDiffusion
2. Dallie-2
3. Molecular science related task

Energy-based models (EBMs)

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

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

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

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

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

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

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

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

Langevin Dynamics

Why score based models?

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

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

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

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

$$\nabla_x \log p_\theta(x) = -\nabla_x f_\theta(x) - \underbrace{\nabla_x \log Z_\theta}_{=0} = -\nabla_x f_\theta(x)/ \tag{7}$$

How do we learn a score?

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

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

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:

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

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

Learning Stein score in high dimensions

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

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

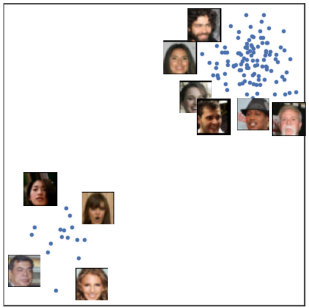
$$\ell_{\text{ssm}}(\mathbf{s}_\theta) \triangleq \mathbb{E}_{x\sim p(x)}\,\mathbb{E}_{\boldsymbol{v}\sim p(\boldsymbol{v})}\left[\left|\boldsymbol{v}^\top \nabla \log p(x) - \boldsymbol{v}^\top \mathbf{s}_\theta(x)\right|^2\right]. \tag{10}$$

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

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

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

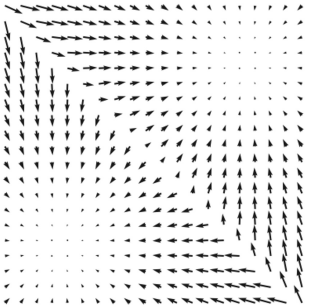
The picture so far



Data samples

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

score matching



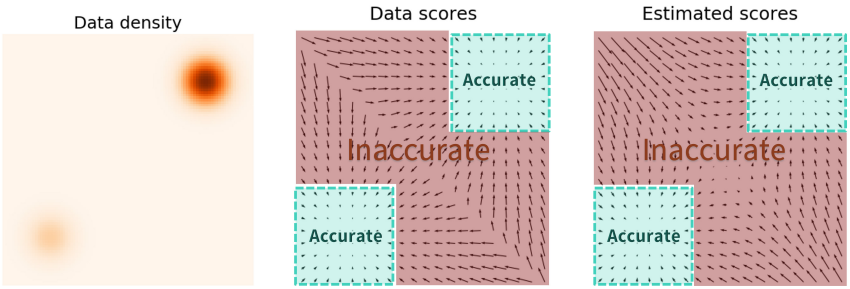
Scores

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

Langevin dynamics

New samples

But this does not quite work...



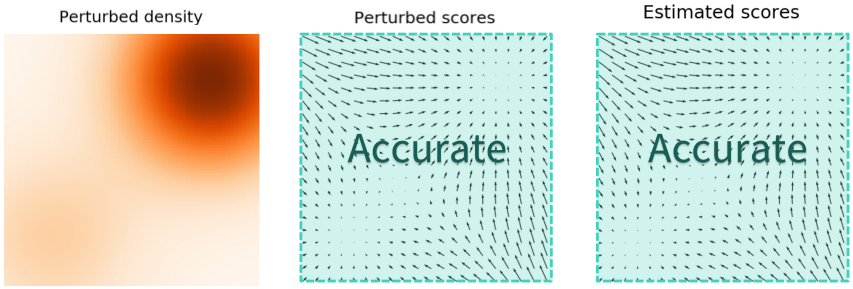
Issues:

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

Denoising Score Matching (Vincent, 2011)

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

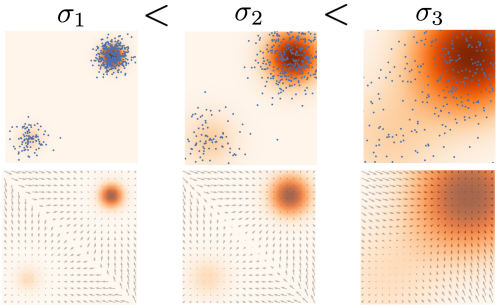
$$p_\sigma(\tilde{x}) \triangleq \int p_\sigma(\tilde{x}|x;\sigma)p(x)\mathrm{d}x, \quad \ell_{\mathrm{dsm}}(\mathbf{s}_\theta) \triangleq \mathbb{E}_{x\sim p_\sigma}\left[\left\|\nabla_x \log p(x) - \mathbf{s}_\theta(x)\right\|^2\right] \quad (12)$$



Typically $p_\sigma(\tilde{x}|x) = \mathcal{N}(0, \sigma^2)$. Unfortunately, targeting wrong density as $p_\sigma(\tilde{x}) \neq p(x)$, trade-off between small and large value of σ.

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

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



Now we have to learn a score function indexed by the noise scale, $s_\theta(\sigma_t, x)$:

$$\ell_{\mathrm{dsm}}(\mathbf{s}_\theta) \triangleq \sum_{t=0}^{T} \lambda(t)\, \mathbb{E}_{x\sim p_{\sigma_t}(x)}\left[\left\|\nabla_x \log p_{\sigma_t}(x) - \mathbf{s}_\theta(\sigma_t, x)\right\|^2\right]. \tag{13}$$

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: $\mathbf{X}_t^{k+1} = \mathbf{X}_t^k + \gamma_t \, \mathbf{s}_\theta\left(\sigma_t, \mathbf{X}_t^k\right) + \sqrt{2\gamma_t}\mathbf{Z}_t^{k+1}$ and $\mathbf{X}_{t-1}^0 = \mathbf{X}_t^K$.
3. Repeat procedure until σ_0 is very small so that $p_{\sigma_0} \approx p$.

This really works!

Relation with time-reversal

Algorithm 1 Sampling of annealing Langevin dynamics

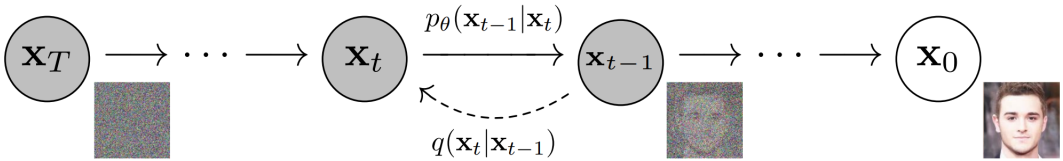
Require: $\{\sigma_t\}_{t=1}^T$, $\{\gamma\}_{t=1}^T$, K

1: Initialize $X_T^0 \sim \mathcal{N}(0, \sigma_T \,\mathrm{Id})$.
2: **for** $t = T$ to 1 **do**
3: **for** $k = 0$ to $K - 1$ **do**
4: Sample $X_t^{k+1} = X_t^k + \gamma_t \mathbf{s}_\theta(\sigma_t, X_t^k) + \sqrt{2\gamma_t} Z_t^{k+1}$
5: $X_{t-1}^0 = X_t^K$
6: **Return** X_0^0.

to update and move later?

- If $K = 1$ then it is *equivalent to the time-reversal* except that:
  - $\{\gamma_t\}_{t=1}^T$ is *a priori* unrelated to $\{\sigma_t\}_{t=1}^T$ contrary to the time-reversal approach where we would have $\gamma_t = \gamma$ and $\sigma_t^2 = t\gamma$.
  - Main difference is that the *forward* process is the discretization of a *Brownian motion*

Discrete Diffusion Probabilistic Models (Sohl-Dickstein et al., 2015; Ho et al., 2020)



Forward process — choose a **forward transition** $p_{k+1|k}(x_{k+1}|x_k)$

$$p(x_{0:N}) = p(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k). \tag{14}$$

Typically $p_{k+1|k}(x_{k+1}|x_k) = \mathcal{N}(\mu_k(x_k), \sigma_k)$

Reversed process — For Gaussian forward, $p_{k|k+1}(x_k|x_{k+1})$ is also Gaussian with intractable mean and computable variance.

$$p(x_{0:N}) = p(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}). \tag{15}$$

Discrete SGMs: Noising process

Forward process

Choose a **forward transition** $p_{k+1|k}(x_{k+1}|x_k)$ (typically gaussian kernel)

$$p(x_{0:N}) = p(x_0) \prod_{k=0}^{N-1} p_{k+1|k}(x_{k+1}|x_k). \tag{16}$$

marginal $p_{k+1}(x_{k+1}) = \int p_{k+1|k}(x_{k+1}|x_k) p_k(x_k) \mathrm{d}x_k.$

Time reversal process

$$p(x_{0:N}) = p(x_N) \prod_{k=0}^{N-1} p_{k|k+1}(x_k|x_{k+1}). \tag{17}$$

Discrete SGMs: Denoising Score Matching

Backward transition

$p_{k|k+1}(x_k|x_{k+1}) = p_{k+1|k}(x_{k+1}|x_k)p_k(x_k)/p_{k+1}(x_{k+1})$ is intractable.

$$p_{k|k+1}(x_k|x_{k+1}) \approx C_\gamma^{-1} \exp\left[-\|x_k - x_{k+1} - \gamma\left(x_{k+1} + 2\gamma \log p_k(x_{k+1})\right)\|^2/(4\gamma)\right] \tag{18}$$

up to a term of order γ in the exponential and $C_\gamma = (4\pi\gamma)^{-d/2}$.

Stein **Score** $\nabla p_k(x_k)$ is intractable, but (Efron, 2011)

$$\nabla_{x_k} \log p_k(x_k) = \int \nabla_{x_k} \log p_{k|0}(x_k|x_0)\, \mathbb{P}_{k|0}(x_k|x_0) \mathrm{d}x_0 = \mathbb{E}_{p_{0|k}(\cdot|x_k)}\left[\nabla \log p_{k|0}(x_k|x_0)\right] \tag{19}$$

This is a **conditional expectation**, hence l2 minimiser -> DSM loss

$$\nabla_{x_k} \log p_k = \arg\min\{\mathbb{E}\left[\|s(\mathbf{X}_k) - \log p_{k|0}(\mathbf{X}_k|\mathbf{X}_0)\|^2\right]\} \tag{20}$$

Continuous score-based models

Taking the limit $\gamma \to 0$...

$$\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t \,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{X}_0 \sim p_0, \tag{21}$$

$$\mathbf{X}_t | \mathbf{X}_0 = e^{-1}\mathbf{X}_0 + \mathbf{B}_{1-e^{-2t}} \tag{22}$$

Under mild conditions on p_0, the time-reversed process $(\mathbf{Y}_t)_{t\geq 0} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ also satisfies an SDE (Cattiaux et al., 2021; Haussmann and Pardoux, 1986) given by

$$\mathrm{d}\mathbf{Y}_t = \{\mathbf{Y}_t + 2\nabla \log p_{T-t}(\mathbf{Y}_t)\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \quad \mathbf{Y}_0 \sim p_T, \tag{23}$$

DSM Loss

$$\ell_t(\mathbf{s}) = \mathbb{E}_{x_0,t,x_t}\left[\lambda(t)\|\mathbf{s}(x_t) - \nabla_{x_t}\log p_{t|0}(x_t|x_0)\|^2\right] \tag{24}$$

Fokker-Planck

Fokker-Planck equation describes the evolution of the density

$$\frac{\partial}{\partial t} p_t(x) = \operatorname{div}\left(b(t,\cdot)\right)(x) + \frac{1}{2}\sum_{i,j}\frac{\partial}{\partial_{i,j}}\left(\Sigma_{i,j}(t,\cdot)p_t\right)(x). \tag{25}$$

If $\Sigma = 0$

$$\frac{\partial}{\partial t} p_t(x) = \operatorname{div}\left(b(t,\cdot)\right)(x). \tag{26}$$

If $\Sigma = c^{1/2}\,\mathrm{Id}$

$$\frac{\partial}{\partial t} p_t(x) = \operatorname{div}\left(b(t,\cdot)\right)(x) + (c/2)\Delta p_t(x) \tag{27}$$

The following dynamics have the same marginal densities:

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

Log-likelihood evolution

To fill

Probability flow

To fill

Equivalence between SM and ELBO loss

To fill

- discrete setting (Ho et al., 2020)
- Continuous setting (Huang et al., 2021; Song, Durkan, et al., 2021)

Active research directions

To fill

- Accelerate reverse sampling
- Structured data (graph, discrete, manifold, functions etc)
- ??

Riemannian score-based generative models

Motivation

Manifold-valued data:

- intrinsic coordinates of molecules: torsional angles $\mathbb{T}^d$ (Jing et al., 2022)
- Amino-acid or protein-ligand binding $\mathrm{SE}_3(\mathbb{R})$ (Corso et al., 2022)
- Hurricane modelling $\mathbb{S}^2$
- Graph embedding in hyperbolic $\mathbb{H}^d$ space? cell development?
- $\mathrm{SU}_n(\mathbb{C})$?

Refine examples + Illustration?

Noising processes on manifolds

Stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t, \mathbf{X}_t)\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}. \tag{28}$$

Langevin dynamics:

$$\mathrm{d}\mathbf{X}_t = -\nabla_{\mathbf{X}_t} U(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}, \tag{29}$$

admits **invariant** density: $\mathrm{d}p_{\mathsf{ref}}/\mathrm{dVol}_{\mathcal{M}}(x) \propto \mathrm{e}^{-U(x)}$ (Durmus, 2016, Section 2.4).

Generalisations of the Gaussian distribution:

- $U(x) = d_{\mathcal{M}}(x, \mu)^2/(2\gamma^2) \Rightarrow\ b(t, x) = -\exp_x^{-1}(\mu)/\gamma^2$ (*Riemannian* normal)
- $U(x) = d_{\mathcal{M}}(x, \mu)^2/(2\gamma^2) + \log|D\exp_\mu^{-1}(x)|$ (*Exp-wrapped* normal)

If $\mathcal{M} = \mathbb{R}^d$ and $\gamma = 1 \Rightarrow\ b(t, x) = \exp_x^{-1}(0) = -x \Leftrightarrow$ **Ornstein-Uhlenbeck**.

Noising processes on manifolds (Cont'd)

If manifold is **compact**:

- $U(x) = \text{constant} \Rightarrow \boxed{b(t,x) = \mathbf{0}} \Leftrightarrow \mathrm{d}\mathbf{X}_t = \mathrm{d}\mathbf{B}_t^{\mathcal{M}}$.

Sampling $\mathbf{X}_t | \mathbf{X}_0$:

- Sometimes available closed form e.g. $\mathbb{T}^n$, $\mathrm{SO}_3(\mathbb{R})$.
- Discretise forward SDE eq. (29).
- Converge to p_{ref} with geometric rate.



Time reversal process

Theorem 3: Time-reversed diffusion

Let $(\mathbf{X}_t)_{t\in[0,T]}$ associated with the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$ and $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ the time-reversal. Under mild assumptions on p_0 and on p_t the density of $\mathbb{P}_t = \mathcal{L}(\mathbf{X}_t)$, then $(\mathbf{Y}_t)_{t\in[0,T]}$ is associated with

$$\mathrm{d}\mathbf{Y}_t = \{-b(t, \mathbf{Y}_t) + \sigma(t)^2 \nabla \log p_{T-t}(\mathbf{Y}_t)\}\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}. \tag{30}$$

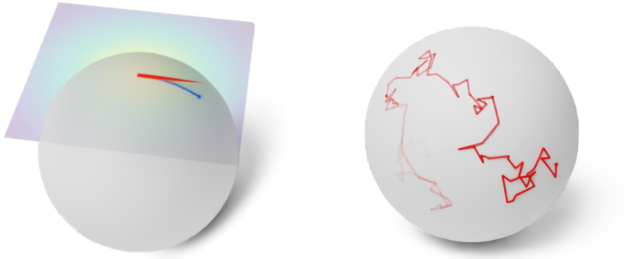
Time reversal of **Langevin dynamics**:

$$\mathrm{d}\mathbf{Y}_t = \{\nabla_{\mathbf{X}_t} U(\mathbf{X}_t) + 2\,\nabla \log p_{T-t}(\mathbf{Y}_t)\}\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}. \tag{31}$$

Algorithm 2 GRW (Geodesic Random Walk)

Require: $T, N, \gamma = T/N, X_0^\gamma, b, \sigma$

1: **for** $k \in \{0, \dots, N-1\}$ **do**
2: $\quad Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Sample a Gaussian in the tangent space of X_k^γ
3: $\quad W_{k+1} = \gamma\, b(k\gamma, X_k^\gamma) + \sqrt{\gamma}\, \sigma(k\gamma, X_k^\gamma)\, Z_{k+1}$ ▷ Euler step on tangent space
4: $\quad X_{k+1}^\gamma = \exp_{X_k^\gamma}[W_{k+1}]$ ▷ Move along the geodesic defined by W_{k+1} and X_k^γ on $\mathcal{M}$
5: **return** $\{X_k^\gamma\}_{k=0}^N$



Summary of SGM on different spaces

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact
Forward process $\mathrm{d}\mathbf{X}_t =$	$-\mathbf{X}_t\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$	$-\nabla_{\mathbf{X}_t} U(\mathbf{X}_t)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$	$\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$
Base distribution	Gaussian	Wrapped Gaussian	Uniform
Time reversal	Cattiaux, 2021	Sec. 3	
Sampling forward	Direct	Geodesic Random Walk (Alg. 2)	
Sampling backward	Euler–Maruyama	Geodesic Random Walk (Alg. 2)	

Table 1: Differences between SGM on Euclidean spaces and RSGM on Riemannian manifolds.

Score approximation: Denoising score matching (DSM)

refactor with 1st part

$$\nabla_{x_t} \log p_t(x_t) = \int_{\mathcal{M}} \nabla_{x_t} \log p_{t|s}(x_t|x_s)\, \mathbb{P}_{s|t}(x_t, \mathrm{d}x_s). \tag{32}$$

Denoising score matching (DSM)

$$\ell_{t|s}(\mathbf{s}_t) = \int_{\mathcal{M}^2} \left\| \nabla_x \log p_{t|s}(x_t|x_s) - \mathbf{s}_t(x_t) \right\|^2 \mathrm{d}\mathbb{P}_{s,t}(x_s, x_t) \tag{33}$$

Sturm–Liouville decomposition (Chavel, 1984) (assumes compactness)

$$p_{t|0}(x_t|x_0) = \textstyle\sum_{j\in\mathbb{N}} \mathrm{e}^{-\lambda_j t} \phi_j(x_0)\phi_j(x_t), \tag{34}$$

Truncation approximation to DSM

$$\nabla_{x_t} \log p_{t|0}(x_t|x_0) \approx S_{J,t}(x_0, x_t) \triangleq \nabla_{x_t} \log \textstyle\sum_{j=0}^{J} \mathrm{e}^{-\lambda_j t} \phi_j(x_0)\phi_j(x_t). \tag{35}$$

Varadhan approximation to DSM

Score approximation: Implicit Score Matching (ISM)

Proposition 1

Let $t, s \in (0, T]$ with $t > s$. Then, for any $\mathbf{s}_t \in \mathrm{C}^\infty(\mathcal{M})$, $\ell_{t|s}(\mathbf{s}_t) = 2\ell_t^{\mathrm{im}}(\mathbf{s}_t) + \int_{\mathcal{M}^2} \left\| \nabla_{x_t} \log p_{t|s}(x_t|x_s) \right\|^2 \mathrm{d}\mathbb{P}_{s,t}(x_s, x_t)$, where

$$\ell_t^{\mathrm{im}}(\mathbf{s}_t) = \int_{\mathcal{M}} \{\tfrac{1}{2} \left\| \mathbf{s}_t(x_t) \right\|^2 + \mathrm{div}(\mathbf{s}_t)(x_t)\} \mathrm{d}\mathbb{P}_t(x_t). \tag{37}$$

add stochastic estimator?

Score approximation: Summary

Loss	Approx	Loss function	Requirements $p_{t\|0}$	Requirements $\exp_{\mathbf{X}_t}^{-1}$	Complexity
$\ell_{t\|0}$ (DSM)	None	$\frac{1}{2}\mathbb{E}\left[\| \mathbf{s}(\mathbf{X}_t) - \nabla \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0) \|^2\right]$	✓	✗	$\mathcal{O}(1)$
	Truncation (35)	$\frac{1}{2}\mathbb{E}\left[\| \mathbf{s}(\mathbf{X}_t) - S_{J,t}(\mathbf{X}_0, \mathbf{X}_t)\|^2\right]$	eigen system	✗	$\mathcal{O}(1)$
	Varhadan (36)	$\frac{1}{2}\mathbb{E}\left[\| \mathbf{s}(\mathbf{X}_t) - \exp_{\mathbf{X}_t}^{-1}(\mathbf{X}_0)/t\|^2\right]$	✗	✓	$\mathcal{O}(1)$
$\ell_{t\|s}$ (DSM)	Varhadan (36)	$\frac{1}{2}\mathbb{E}\left[\| \mathbf{s}(\mathbf{X}_t) - \exp_{\mathbf{X}_t}^{-1}(\mathbf{X}_s)/(t-s)\|^2\right]$	✗	✓	$\mathcal{O}(1)$
ℓ_t^{im} (ISM)	Deterministic	$\mathbb{E}\left[\frac{1}{2}\| \mathbf{s}(\mathbf{X}_t)\|^2 + \operatorname{div}(\mathbf{s})(\mathbf{X}_t)\right]$	✗	✗	$\mathcal{O}(d)$
	Stochastic	$\mathbb{E}\left[\frac{1}{2}\| \mathbf{s}(\mathbf{X}_t)\|^2 + \varepsilon^\top \partial \mathbf{s}(\mathbf{X}_t)\varepsilon\right]$	✗	✗	$\mathcal{O}(1)$

Table 2: Computational complexity of score matching losses w.r.t. score network passes.

Parametrisation of score network

Approximate **Stein score** $(\nabla \log p_t)_{t\in[0,T]} \approx \mathbf{s}_\theta(t,\cdot)$ and $\mathbf{s}_\theta : [0,T] \to \mathcal{X}(\mathcal{M})$.

Generators of vector fields:

- *Definition*: Smooth vector fields $\{E_i(x)\}_{i=1}^n$ such that span $\left(\{E_i(x)\}_{i=1}^n\right) = \mathrm{T}_x\mathcal{M}$.
- $\mathbf{s}_\theta(t,x) \triangleq \sum_{i=1}^n \mathbf{s}_\theta^i(t,x)\, E_i(x)$.
- $\mathcal{M}$ is parallelisable $\Leftrightarrow$ there exists generators $\{E_i\}_{i=1}^n$ with $n = d$.
- If $\mathcal{M} = \mathbb{R}^d$, can choose $E_i(x) = e_i$ for $i = 1, \dots, d$.
- If $\mathcal{M} = G$ is a Lie group, can choose $E_i(g) = g \cdot e_i$ with $\{e_i\}_i$ basis of Lie algebra.
- If $\mathcal{M} \subset \mathbb{R}^n$ is submersion, can choose $E_i(x) = P_i(x)$ for $i = 1, \dots, n$ with P_i the ith column of the tangent projection matrix operator.

Important 'tricks'

Move non manifold specific tricks to 1st part?

Also: Loss function weighting, corrector and more

- **Exponential moving average** of the weights $\Leftarrow$ due to high stochasticity of the training loss.
- **Noise scheduling** $\beta(t)$
  - $\mathrm{d}\mathbf{X}_t = -\beta(t)\, \nabla_{\mathbf{X}_t} U(\mathbf{X}_t)\, \mathrm{d}t + \sqrt{2}\sqrt{\beta(t)}\, \mathrm{d}\mathbf{B}_t^{\mathcal{M}}$.
  - 'Rescale time': $t \mapsto \int_0^t \beta(s) ds$.
  - Spend more time when the score has high norm, i.e. when t is small.
  - Aim: $\mathbb{W}(\mathcal{L}(\mathbf{X}_t), p_{\text{ref}}) \approx \mathbb{W}(\mathcal{L}(\mathbf{X}_t), p_0) * (1-t)_+$.
- **Score network parametrisation**
  - $\mathbf{s}_\theta(t, x_t) = \Big(h_\theta(t, x_t)/\sigma_t + 2 * b(t, x_t)/\beta(t)\Big)$.
  - with $\mathbb{E}\big[\|\nabla \log p(\mathbf{X}_t|\mathbf{X}_0)\|^2\big]^{1/2} = \mathrm{Std}[\mathbf{X}_t|\mathbf{X}_0] \triangleq \sigma_t = 1 - e^{-\int_0^1 \beta(s)ds}$.
  - If $h_\theta(t, x_t) = 0$, **forward=backward** since $\bar{b}(t, y_t) = b(t, y_t)$.

Theoretical guaranties on time-reversal

Theorem 4

Under mild assumption over p_0 and assuming that there exists $\mathtt{M} \geq 0$ such that for any $t \in [0,T]$ and $x \in \mathcal{M}$, $\left\| \mathbf{s}_{\theta^\star}(t,x) - \nabla \log p_t(x) \right\| \leq \mathtt{M}$, with $\mathbf{s}_{\theta^\star} \in \mathrm{C}([0,T], \mathcal{X}(\mathcal{M}))$. Then if $T > 1/2$, there exists $C \geq 0$ independent on T s.t.

$$\mathbb{W}_1(\mathcal{L}(Y_N), p_0) = C(\mathrm{e}^{-\lambda_1 T} + \sqrt{T/2}\mathtt{M} + \mathrm{e}^T \gamma^{1/2}), \tag{38}$$

where $\mathbb{W}_1$ is the Wasserstein distance of order one on the probability measures on $\mathcal{M}$.

Experimental results

Prior work

To fill

Continuous normalising flows (CNFs)

-

Moser flows

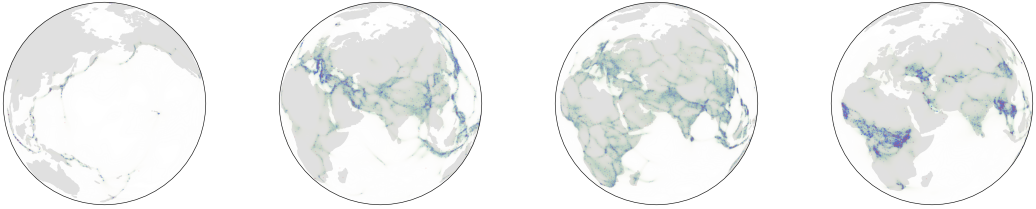
-

Prior work: Summary

Method	Training	Likelihood evaluation	Sampling
RCNF	ODE $\mathcal{O}(dN)$	Augmented ODE $\mathcal{O}(dN)$	ODE $\mathcal{O}(N)$
Moser flow	div $\mathcal{O}(dk)$ or $\mathcal{O}(k)$	Augmented ODE $\mathcal{O}(dN)$	ODE $\mathcal{O}(N)$
RSGM	Score matching $\mathcal{O}(d)$ or $\mathcal{O}(1)$	Augmented ODE $\mathcal{O}(dN)$	SDE $\mathcal{O}(N^*)$

Table 3: Summary of computational complexity (w.r.t. neural network forward and backward passes) for different methods. d is the manifold dimension, k the number of Monte Carlo batches in Moser flow's regularizer, N is the number of steps in the (adaptive) ODE solver, whereas N^* is the number of steps in the SDE Euler-Maruyama solver.

Earth science data

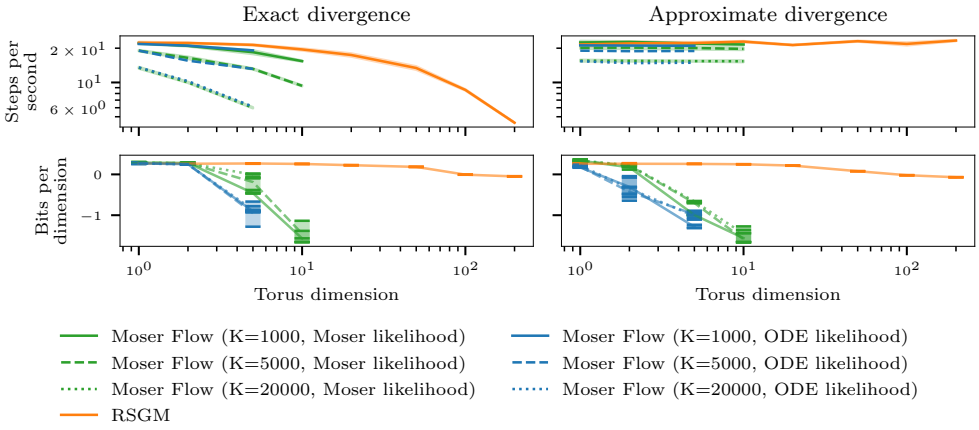


(a) Volcano **(b)** Earthquake **(c)** Flood **(d)** Fire

Method	Volcano	Earthquake	Flood	Fire
Mixture of Kent	$-0.80_{\pm 0.47}$	$0.33_{\pm 0.05}$	$0.73_{\pm 0.07}$	$-1.18_{\pm 0.06}$
Riemannian CNF	$\mathbf{-6.05_{\pm 0.61}}$	$0.14_{\pm 0.23}$	$1.11_{\pm 0.19}$	$\mathbf{-0.80_{\pm 0.54}}$
Moser Flow	$-4.21_{\pm 0.17}$	$\mathbf{-0.16_{\pm 0.06}}$	$\mathbf{0.57_{\pm 0.10}}$	$\mathbf{-1.28_{\pm 0.05}}$
Stereographic Score-Based	$-3.80_{\pm 0.27}$	$\mathbf{-0.19_{\pm 0.05}}$	$\mathbf{0.59_{\pm 0.07}}$	$\mathbf{-1.28_{\pm 0.12}}$
Riemannian Score-Based	$-4.92_{\pm 0.25}$	$\mathbf{-0.19_{\pm 0.07}}$	$\mathbf{0.45_{\pm 0.17}}$	$\mathbf{-1.33_{\pm 0.06}}$
Dataset size	827	6120	4875	12809

High dimensional torus

- We consider a wrapped Gaussian target distribution on $\mathbb{T}^d = \mathbb{S}^1 \times \cdots \times \mathbb{S}^1$.



Synthetic data on Lie groups

- We consider a mixture of wrapped Gaussian target distribution on $\mathrm{SO}_3(\mathbb{R}) = \{\mathrm{Q} \in \mathrm{M}_3(\mathbb{R}) \,:\, \mathrm{Q}\mathrm{Q}^\top = I_3,\ \det(\mathrm{Q}) = 1\}$.

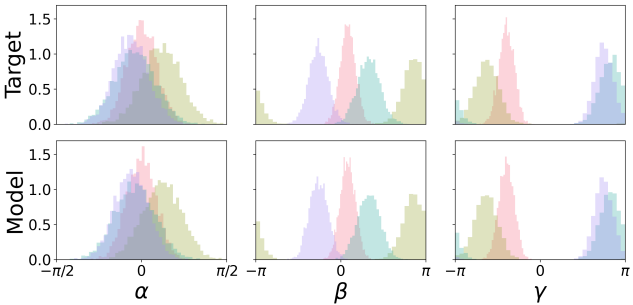


Figure 4: Histograms of $\mathrm{SO}_3(\mathbb{R})$ samples from a target mixture distribution.

Synthetic data on Lie groups (Cont'd)

Method	$M = 16$		$M = 32$		$M = 64$	
	$\log p$	NFE	$\log p$	NFE	$\log p$	NFE
Moser Flow	$0.85_{\pm 0.03}$	$2.3_{\pm 0.5}$	$0.17_{\pm 0.03}$	$2.3_{\pm 0.9}$	$\mathbf{-0.49_{\pm 0.02}}$	$7.3_{\pm 1.4}$
Exp-wrapped SGM	$\mathbf{0.87_{\pm 0.04}}$	$0.5_{\pm 0.1}$	$0.16_{\pm 0.03}$	$0.5_{\pm 0.0}$	$-0.58_{\pm 0.04}$	$0.5_{\pm 0.0}$
RSGM	$\mathbf{0.89_{\pm 0.03}}$	$\mathbf{0.1_{\pm 0.0}}$	$\mathbf{0.20_{\pm 0.03}}$	$\mathbf{0.1_{\pm 0.0}}$	$\mathbf{-0.49_{\pm 0.02}}$	$\mathbf{0.1_{\pm 0.0}}$

Table 5: Log-likelihood and neural function evaluations (NFE) in 10^3.

Recap!

-

Future directions

- Remove boundaryless assumption $\Rightarrow$ processes on manifolds with boundary
  - Reflected Brownian motion
  - Log-barrier Langevin dynamics
- Faster sampling, e.g. closed form for $\mathbb{S}^d$
- Stochastic processes (Phillips et al., 2022)
- Incorporate symmetries

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 28.

I. Chavel. *Eigenvalues in Riemannian Geometry*. Academic press, 1984. Cited on page 41.

G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. Jaakkola. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. Oct. 2022. URL: http://arxiv.org/abs/2210.01776. Cited on page 35.

G. Daras, M. Delbracio, H. Talebi, A. G. Dimakis, and P. Milanfar. Soft Diffusion: Score Matching for General Corruptions. Sept. 2022. DOI: 10.48550/arXiv.2209.05442. Cited on page 45.

A. Durmus. *High Dimensional Markov Chain Monte Carlo Methods: Theory, Methods and Application*. PhD thesis, Paris-Sud XI, 2016. Cited on page 36.

A. Eberle. Reflection couplings and contraction rates for diffusions. *Probability theory and related fields*, 166(3):851–886, 2016. Cited on page 18.

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

U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986. Cited on page 28.

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.

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 page 32.

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 15.

B. Jing, G. Corso, J. Chang, R. Barzilay, and T. Jaakkola. Torsional Diffusion for Molecular Conformer Generation. June 2022. URL: http://arxiv.org/abs/2206.01729. Cited on page 35.

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 page 24.

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 page 32.

Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019. Cited on page 20.

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 20.

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 16.

P. Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. Cited on page 19.