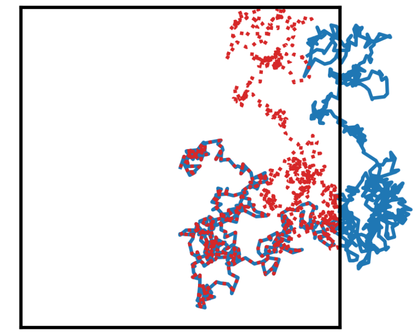
Talk for MSR Cambridge

Geometry
in
Score-Based Generative Modelling

Research Outline

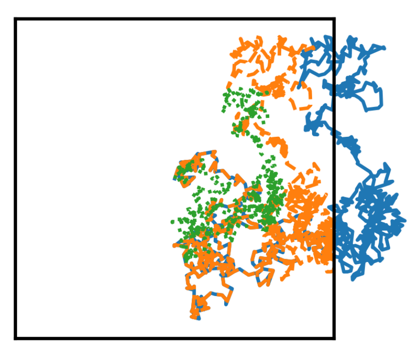
Manifolds

Neurips 2022



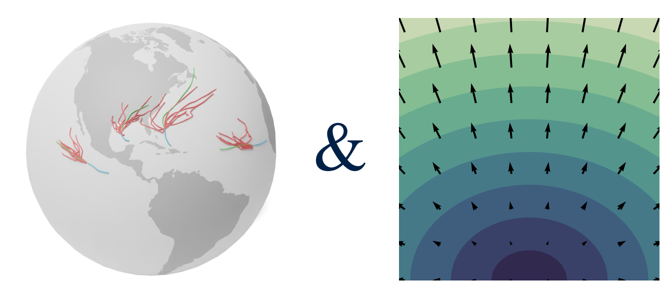
Better Manifolds with Boundary

Arxiv 2023



Manifolds with Boundary

TMLR 2023



Fields and Paths on Manifolds

Arxiv 2023

Riemannian Score-Based Generative Modelling

Outstanding Paper Award, Neurips 2022






Valentin De Bortoli*

Émile Mathieu*

Michael Hutchinson*

James Thornton

Yee Whye Teh

Arnaud Doucet

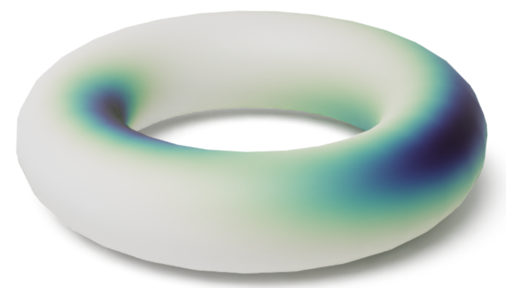
*equal contribution

Geometry

What do I mean by geometry in this context?

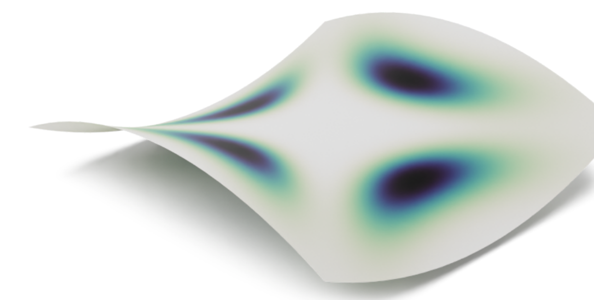
Euclidean space

Torus



Sphere

Hyperbolic space

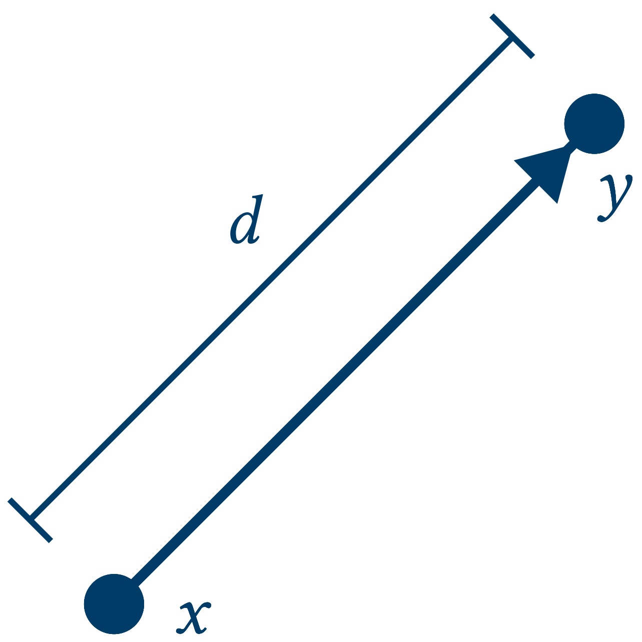


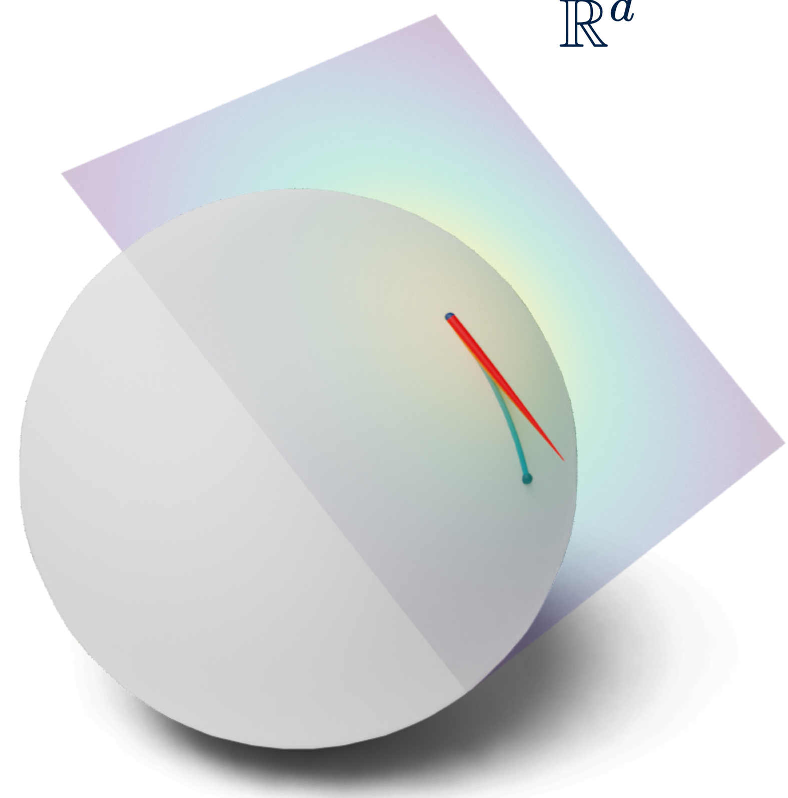
Locally: manifolds look Euclidean (flat); Globally: they look very different

Many common concepts are different in non-Euclidean space!

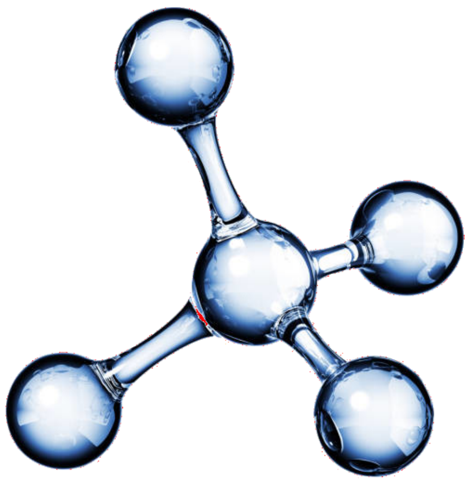
Geometry

	Euclidean Space	**Manifolds**
Straight lines	$x + t(y - x)$	Geodesics
Distances	$\|x - y\| = \sqrt{\sum (x_i - y_i)^2}$	$d(x, y)$
Getting between points	$x + (y - x)$	$\exp(x, \log(x, y))$
Tangent space	$\mathbb{R}^d$	$\mathbb{R}^d$

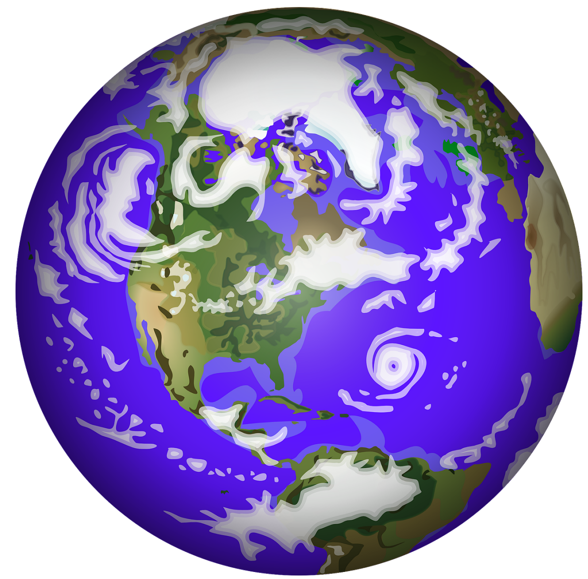




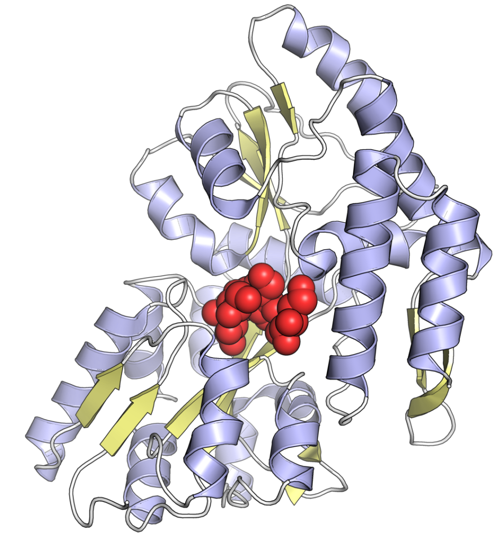
Geometry in Generative Modelling



Molecules

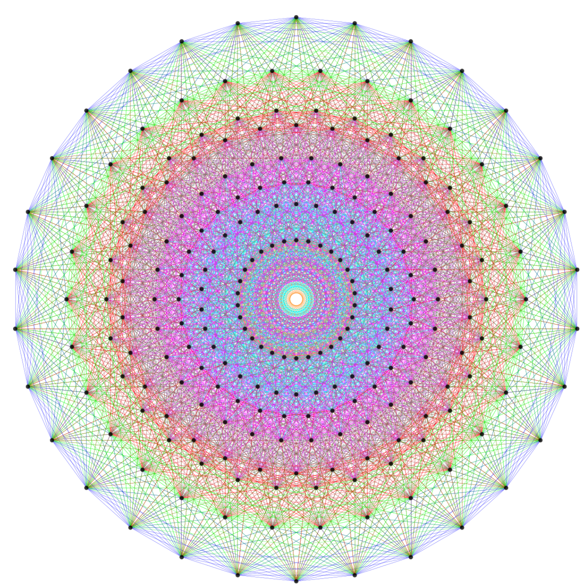


Climate data

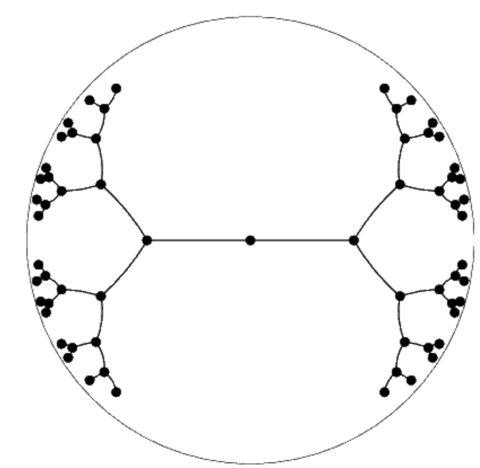


Proteins

Robotics



Lie groups

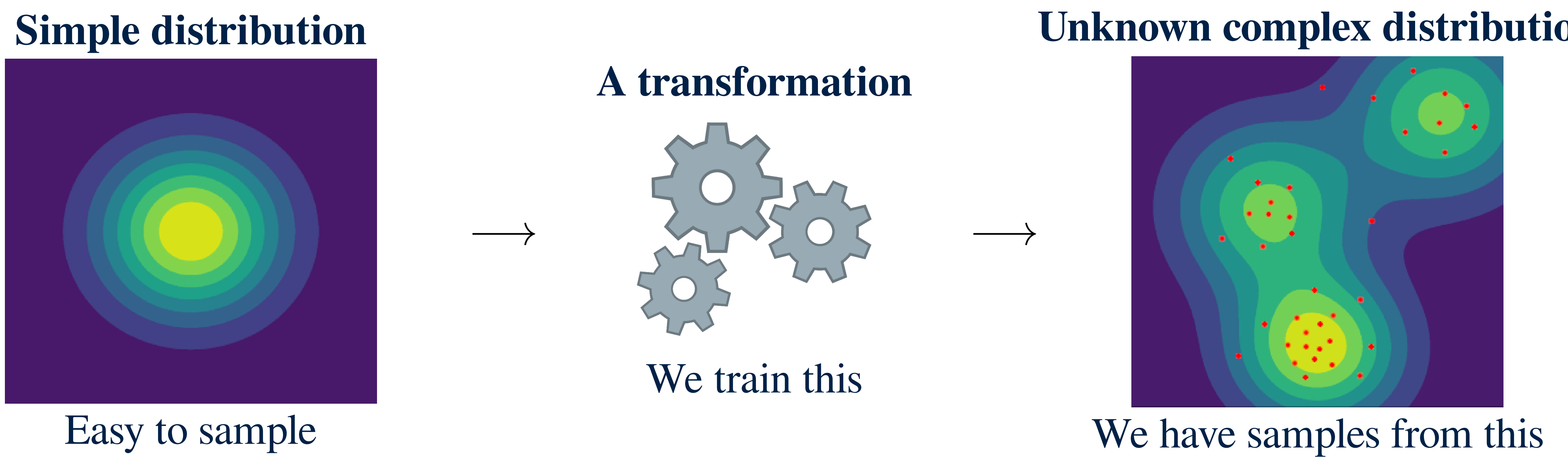


Trees

Generative Modelling

Generically, in generative modelling we are looking to parmetrise an unknown density. Typically we have access to *samples* from that density. We may want to:

- Sample more items like them.
- Produce a density estimator for the density.



Generative Modelling

Likelihood based models

- VAEs
- Normalizing flows
- Autoregressive models
- Energy based models

These typically have restricted forms on the models, or are trained via surrogate ELBOs.

Implicit models

- GANs

The adversarial losses of these models can be very tricky to train, and we have no access to likelihoods from the models.

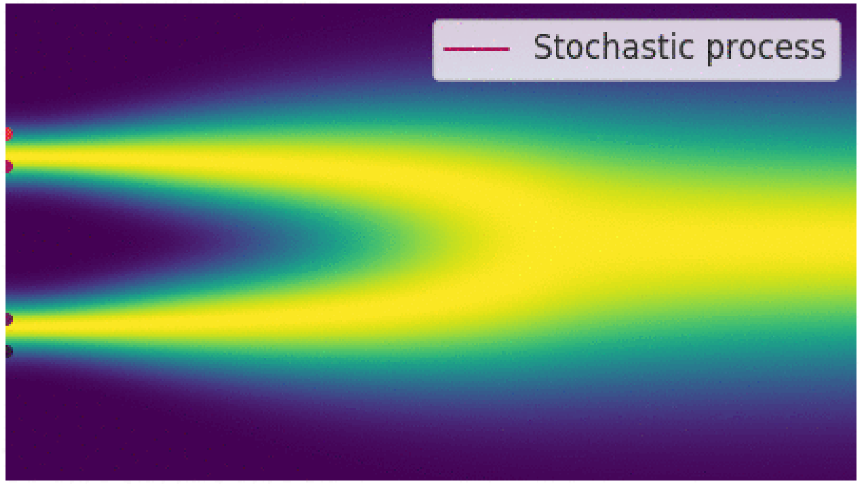
What benefit do score based models bring?

- Simulation-free training → Much faster than normalsing flows 👍
- Stationary, regression, objective → Much more stable than GANs 👍
- Empirically exceptional results with minimal tricks 👍

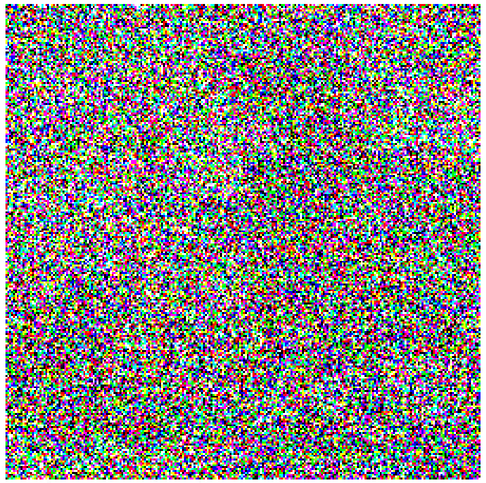
Score-Based Generative Modelling

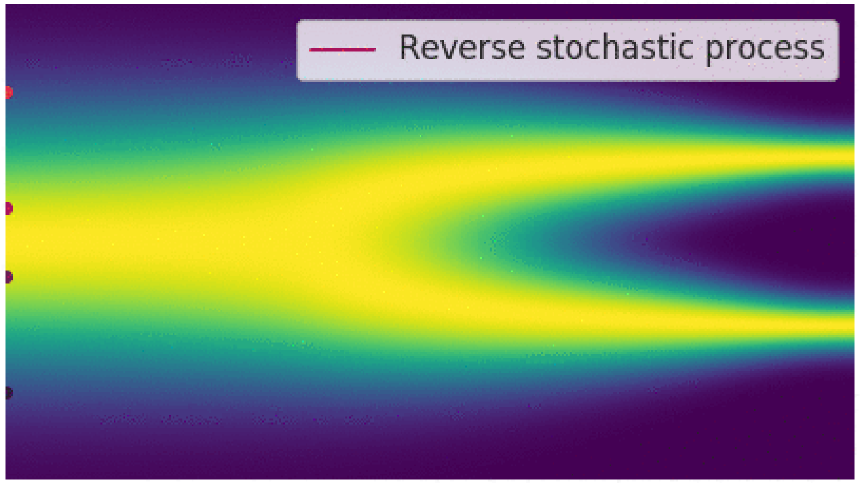
How do score-based generative models work?

A forward process...



...which we then reverse





GIFs reproduced from https://yang-song.net/blog/2021/score/

Score-Based Generative Modelling

How do score-based generative models work?

The forward noising process is a *Stochastic differential equation (SDE)*

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

which should in the limit $t \to \infty$ converge to a stable analytic distribution. Typical score matching uses the *Ornstein-Uhlenbeck* process:

$$\mathrm{d}\mathbf{X}_t = -\mathbf{X}_t\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

which converges to a Gaussian. Other options exist.

The reverse can be proved to be defined by:

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

Where $p_t(\mathbf{X})$ is the evolved density of the SDE at time t.

so our deep learning challenge is *learning the score*, $\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t)$.

Learning the score

Ideally, we would train the score function $\mathbf{s}$ to match the score directly.

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \operatorname*{argmin}_{\mathbf{s} \in L^2} \mathbb{E}_{\mathbf{X}_t \sim p_t} \left[\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) \|^2 \right]$$

Clearly ths won't work... We can introduce a *conditional expectation* with the same minimiser:

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \operatorname*{argmin}_{\mathbf{s} \in L^2} \mathbb{E}_{\mathbf{X}_t \sim p_t} \left[\mathbb{E}_{\mathbf{X}_0 | \mathbf{X}_t \sim p_{0|t}} [\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0) \|]^2 \right]$$

We can compute $\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0)$! But sampling $p_{0|t}$ is hard.

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \operatorname*{argmin}_{\mathbf{s} \in L^2} \mathbb{E}_{\mathbf{X}_0 \sim p_0} \mathbb{E}_{\mathbf{X}_t | \mathbf{X}_0 \sim p_{t|0}} \left[\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0) \|^2 \right]$$

Using usual probability rules we can flip the time indices!

Learning the score

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \underset{\mathbf{s} \in L^2}{\operatorname{argmin}}\, \mathbb{E}_{\mathbf{X}_0 \sim p_0} \mathbb{E}_{\mathbf{X}_t | \mathbf{X}_0 \sim p_{t|0}} \left[\left\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0) \right\|^2 \right]$$

Why is this useful?

- p_0 is our data distribution.
- $p_{t|0}$ is analytic for the OU process.

Now we just integrate over the time variable with some weighting $\lambda(t)$

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \underset{\mathbf{s} \in L^2}{\operatorname{argmin}} \int \lambda(t) \mathbb{E}_{\mathbf{X}_0 \sim p_0} \mathbb{E}_{\mathbf{X}_t | \mathbf{X}_0 \sim p_{t|0}} \left[\left\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0) \right\|^2 \right] \mathrm{d}t$$

And with this we can learn the score, *simulation free!*

N.B. This objective is *high variance*, and requires us to take a running average of the parameters at test time.

Sampling the model: via SDEs

Given an SDE of the form

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

We can discretise this with steps of the form

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma b(t(k), \mathbf{X}_k) + \sqrt{\gamma}\sigma(t(k))\mathbf{Z}_{k+1} \qquad \mathbf{Z}_{k+1} \sim \mathcal{N}(0, \mathrm{Id})$$

You can get error bounds on the convergence to the true SDE, and you can use this to sample the forward and backwards SDE.

You can use *Langevin correction steps* to help sampling as well.

$$\mathrm{d}\mathbf{X}_t = \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$$

Targets exactly the density p_t when discretised.

Sampling the model: via ODEs

Given an SDE of the form

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

The following ODE has the same time-marginals

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

With this ODE we can:

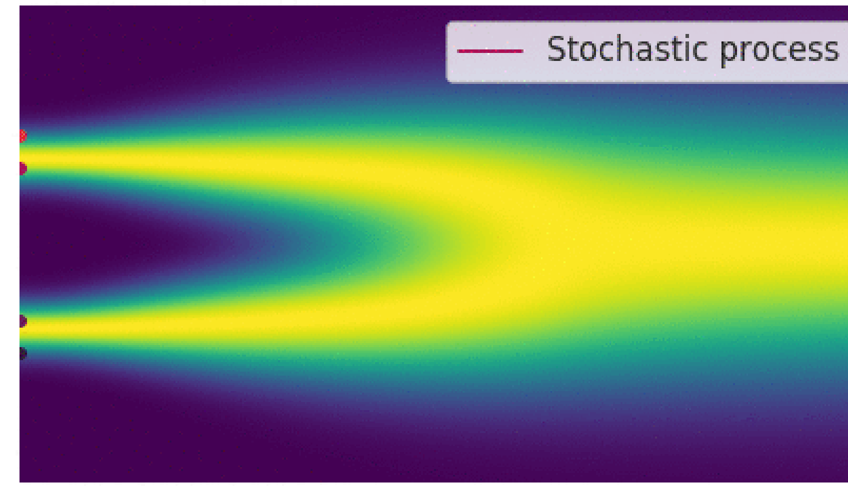
- Use *error-tolerant* ODE solvers.
- Apply the same methods as *Continuous Normalising Flows* to get a *change in likelihood* for the flow, and therefore for the datapoint.

Score-Based Generative Modelling

How do score-based generative models work?

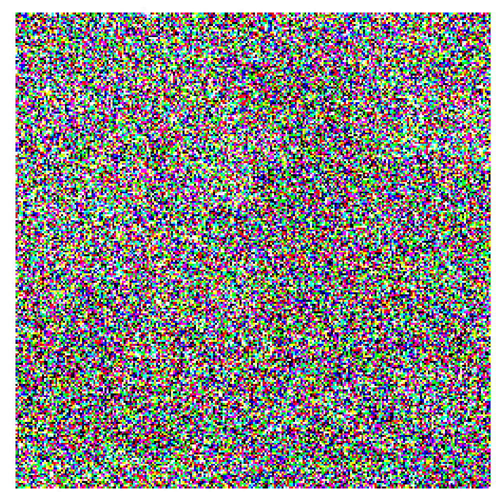
A forward process...

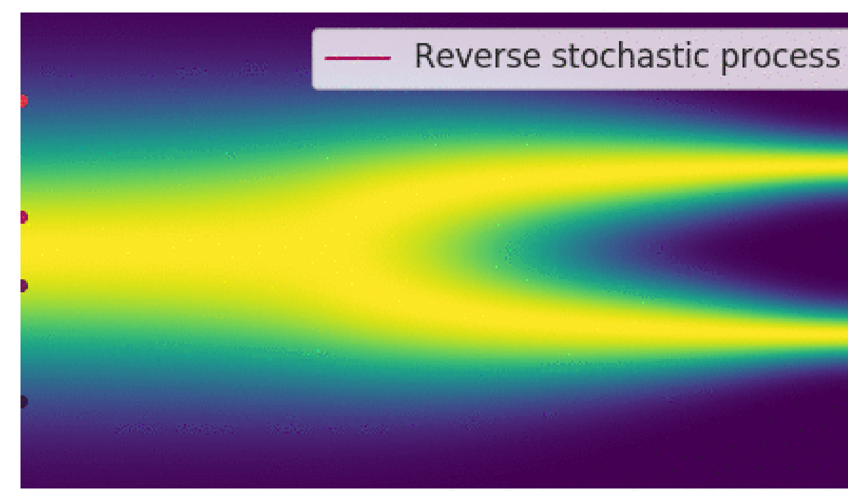
- Defined by an SDE
- that converges to a nicely
- with an analytic reversal



...which we then reverse

- By learning the score
- and discretising the SDE
- or solving the ODE.





GIFs reproduced from https://yang-song.net/blog/2021/score/

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact Manifold
Forward Process	OU	✗	✗
Base distribution	Gaussian	✗	✗
Time reversal	Cattiaux, 2021	✗	✗
SDE Discretisation	Eular-Maruyama	✗	✗
Score-matching	Denoising	✗	✗
Sample $p_{t\|s}(\mathbf{X}_s)$	Analytic	✗	✗
$\nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t \| \mathbf{X}_0)$	Analytic	✗	✗

Forward Process

The typical forward SDE is in fact a specific form of *Langevin dynamics*

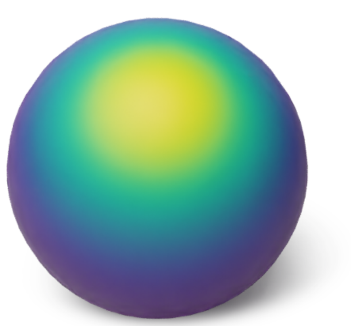
$$\mathrm{d}\mathbf{X}_t = -\nabla_{\mathbf{X}} U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t \xrightarrow[t\to\infty]{\text{converges to}} p(\mathbf{X}) \propto e^{-U(\mathbf{X})}$$

Where you have $U(\mathbf{X}) = \mathbf{X}^2$, this gives a Gaussian

As it turns out, Langevin dynamics still hold on most manifolds

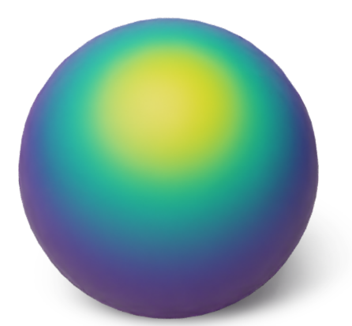
$$\mathrm{d}\mathbf{X}_t = -\nabla_{\mathbf{X}} U(\mathbf{X}_t)\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t^{\mathcal{M}} \xrightarrow[t\to\infty]{\text{converges to}} \frac{\mathrm{d}p}{\mathrm{d}\,\mathrm{Vol}_{\mathcal{M}}}(\mathbf{X}) \propto e^{-U(\mathbf{X})}$$

Riemannian normal



$U(\mathbf{X}) = d_{\mathcal{M}}(\mathbf{X}, \mu)^2$

Wrapped normal



$U(\mathbf{X}) = d_{\mathcal{M}}(\mathbf{X}, \mu)^2 + \log |D \exp_{\mu}^{-1}(\mathbf{X})|$

Uniform

$U(\mathbf{X}) = 0$

Geometry in Score-Based Generative Modelling

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact Manifold
Forward Process	OU	Langevin dynamics	Langevin dynamics
Base distribution	Gaussian	Wrapped normal	Uniform
Time reversal	Cattiaux, 2021	✗	✗
SDE Discretisation	Eular-Maruyama	✗	✗
Score-matching	Denoising	✗	✗
Sample $p_{t\|s}(\mathbf{X}_s)$	Analytic	✗	✗
$\nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0)$	Analytic	✗	✗

Time reversal on Euclidean space

Theorem (Time-reversal of linear SDEs on $\mathbb{R}^n$):
Let $(\mathbf{X}_t)_{t\in[0,T]}$ with be associated with the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{B}_t$.
Then the time-reversal $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ is associated with

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

This result has been proved in a number of ways with increasingly modern tools, some examples:

- Anderson 1982 (light on rigour, stochastic control point of view)
- Haussmann and Pardoux 1986 (PDE point of view)
- Cattiaux et al. 2021, Theorem 4.9 (rigorous Anderson)

but none of these results apply outside the Euclidean setting $\rightarrow$ we will need to generalise this.

Time reversal on Manifolds

Theorem 1 (Time-reversal of linear SDEs on manifolds)
Let $(\mathbf{X}_t)_{t\in[0,T]}$ with be associated with the SDE $\mathrm{d}\mathbf{X}_t = b(t, \mathbf{X}_t)\mathrm{d}t + \sigma(t)\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$.
Then the time-reversal $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ is associated with

$$\mathrm{d}\mathbf{Y}_t = \left\{-b(T-t, \mathbf{Y}_t) + \sigma(T-t)^2 \nabla_{\mathbf{X}} \log p_{T-t}(\mathbf{Y}_t)\right\} \mathrm{d}t + \sigma(T-t)\mathrm{d}\mathbf{B}_t^{\mathcal{M}}$$

Why is this hard? $\rightarrow$ Geometry $\cap$ Stochastic processes throws up technical difficulties with regularity of functions.

How do we solve this in the end?

- Following the spirit of Cattieux's proof.
- State a simplified version of the theorem for Markov processes.
- Verify the regularity conditions by adapting Girsanov theory to manifolds, utilising the Nash embedding theorem.

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact Manifold
Forward Process	OU	Langevin dynamics	Langevin dynamics
Base distribution	Gaussian	Wrapped normal	Uniform
Time reversal	Cattiaux, 2021	Theorem 1	Theorem 1
SDE Discretisation	Eular-Maruyama	✗	✗
Score-matching	Denoising	✗	✗
Sample $p_{t\|s}(\mathbf{X}_s)$	Analytic	✗	✗
$\nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0)$	Analytic	✗	✗

Discretising SDEs on Euclidean space

Given an SDE of the form

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

We would discretise this with steps of the form

$$\mathbf{X}_{k+1} = \mathbf{X}_k + \gamma b(t(k), \mathbf{X}_k) + \sqrt{\gamma}\sigma(t(k))\mathbf{Z}_{k+1} \qquad \mathbf{Z}_{k+1} \sim \mathcal{N}(0, \mathrm{Id})$$

On manifolds we need to generalise this a little bit

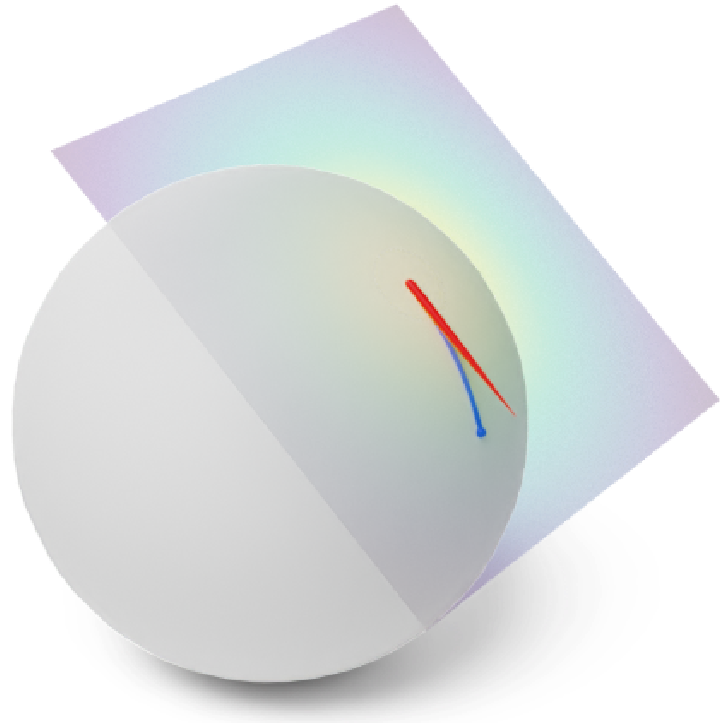
Discretising SDEs on Manifolds

Given an SDE of the form

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

We would discretise this with steps of the form

$$\mathbf{X}_{k+1} = \exp\left(\mathbf{X}_k, \gamma b(t(k), \mathbf{X}_k) + \sqrt{\gamma}\sigma(t(k))\mathbf{Z}_{k+1}\right) \qquad \mathbf{Z}_{k+1} \sim \mathcal{N}(0, \mathrm{Id})$$




Discretising SDEs on Manifolds

Given an SDE of the form

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

We would discretise this with steps of the form

$$\mathbf{X}_{k+1} = \exp\left(\mathbf{X}_k, \gamma b(t(k), \mathbf{X}_k) + \sqrt{\gamma}\sigma(t(k))\mathbf{Z}_{k+1}\right) \qquad \mathbf{Z}_{k+1} \sim \mathcal{N}(0, \mathrm{Id})$$

These are known as *Geodesic Random Walks*

These we well known, but we produce a new *error control* theorem for time-inhomogenous SDEs.

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact Manifold
Forward Process	OU	Langevin dynamics	Langevin dynamics
Base distribution	Gaussian	Wrapped normal	Uniform
Time reversal	Cattiaux, 2021	Theorem 1	Theorem 1
SDE Discretisation	Eular-Maruyama	GRW	GRW
Score-matching	Denoising	✗	✗
Sample $p_{t\|s}(\mathbf{X}_s)$	Analytic	✗	✗
$\nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0)$	Analytic	✗	✗

Denoising Score-matchings on Manifolds

Fortunately the denoising score-matching objective carries over with no trouble to manifolds. That is

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \underset{\mathbf{s} \in L^2}{\operatorname{argmin}} \int \lambda(t) \mathbb{E}_{\mathbf{X}_0 \sim p_0} \mathbb{E}_{\mathbf{X}_t | \mathbf{X}_0 \sim p_{t|0}} \left[\left\| \mathbf{s}(t, \mathbf{X}_t) - \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t | \mathbf{X}_0) \right\|^2 \right] \mathrm{d}t$$

Our issue comes with evaluating $\nabla_{\mathbf{X}_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ and sampling $p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$

- For the wrapped Gaussian SDE, we don't have a closed form for sampling or evaluation.
- For Brownian motion SDE, this is the *heat kernel*
  - $\nabla_{\mathbf{X}_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ has approximations.
  - Sampling analytically is still difficult.

Approximating $\nabla_{\mathbf{X}_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)$ on Manifolds

Strum-Louiville (compact only)

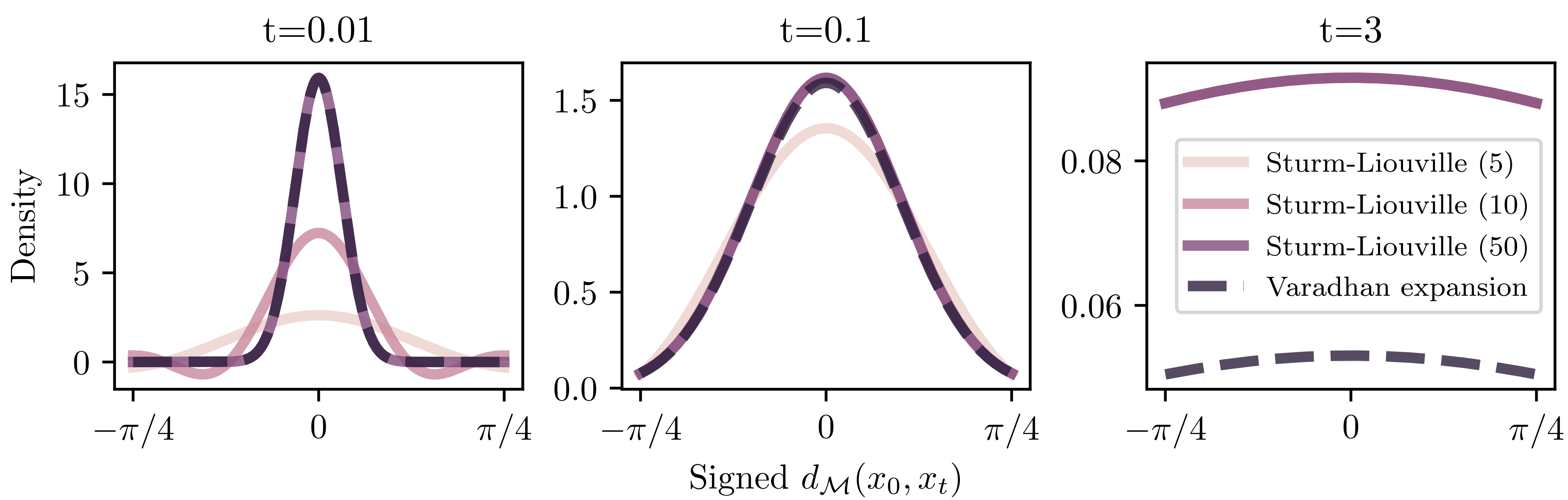
If we have the *eigenpairs* (λ_j, ϕ_j) of the *Laplace-Beltrami operator* $\Delta_\mathcal{M}$ then

$$p_{t|0}(\mathbf{X}_t|\mathbf{X}_0) = \sum_{j\in\mathbb{N}} e^{-\lambda_i t}\phi_j(\mathbf{X}_0)\phi_j(\mathbf{X}_t)$$

Varadhan

Alternatively we have in the small time limit:

$$\lim_{t\to 0} \nabla_{\mathbf{X}_t} \log p_{t|0}(\mathbf{X}_t|\mathbf{X}_0) = \exp^{-1}(\mathbf{X}_t, \mathbf{X}_0)/t$$



Implicit Score Matching on Manifolds [Hyvärinen 2005]

What if we can't approximate the conditional score?

$$\mathbb{E}_{\mathbf{X}_t \sim p_t}\left[\left\|\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) - \mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{X}_t \sim p_t}\left[C - 2\left\langle \nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t), \mathbf{s}(\mathbf{X}_t)\right\rangle + \left\|\mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{X}_t \sim p_t}\left[C - 2/p_t(\mathbf{X}_t) * \left\langle \nabla_{\mathbf{X}_t} p_t(\mathbf{X}_t), \mathbf{s}(\mathbf{X}_t)\right\rangle + \left\|\mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{X}_t \sim p_t}\left[C + 2/p_t(\mathbf{X}_t) * \left\langle p_t(\mathbf{X}_t), \operatorname{div}(\mathbf{s})(\mathbf{X}_t)\right\rangle + \left\|\mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{X}_t \sim p_t}\left[C + 2\operatorname{div}(\mathbf{s})(\mathbf{X}_t) + \left\|\mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

$$= \mathbb{E}_{\mathbf{X}_0 \sim p_0, \mathbf{X}_t \sim p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)}\left[C + 2\operatorname{div}(\mathbf{s})(\mathbf{X}_t) + \left\|\mathbf{s}(\mathbf{X}_t)\right\|^2\right]$$

Implicit Score Matching on Manifolds [Hyvärinen 2005]

$$= \mathbb{E}_{\mathbf{X}_0 \sim p_0, \mathbf{X}_t \sim p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)} \left[C + 2\,\mathrm{div}\,(\mathbf{s})(\mathbf{X}_t) + \|\mathbf{s}(\mathbf{X}_t)\|^2 \right]$$

Using a divergence theorem for non-compact manifolds (e.g. Gaffney 1954) we can show an identical result. with some regularity conditions...

That is:

$$\nabla_{\mathbf{X}} \log p_t(\mathbf{X}_t) = \underset{\mathbf{s} \in L^2}{\operatorname{argmin}} \int_0^T \lambda(t) \mathbb{E}_{\mathbf{X}_0 \sim p_0, \mathbf{X}_t \sim p_{t|0}(\mathbf{X}_t|\mathbf{X}_0)} \left[2\,\mathrm{div}\,(\mathbf{s})(\mathbf{X}_t) + \|\mathbf{s}(\mathbf{X}_t)\|^2 \right] \mathrm{d}t$$

And the usual Hutchinson trace trick estimator can be used [Song et al. 2019].

Loss	Approximation	Loss function	Requirements		Complexity
			$p_{t\|0}$	$\exp^{-1}$	
DSM	None	$\mathbb{E}\left[\left\|\mathbf{s}(\mathbf{X}_t) - \nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0)\right\|^2\right]$	✔	✘	$\mathcal{O}(1)$
	Truncation	$\mathbb{E}\left[\left\|\mathbf{s}(\mathbf{X}_t) - \nabla_{\mathbf{X}_t} \log S_J(\mathbf{X}_t, \mathbf{X}_0)\right\|^2\right]$	Expansion	✘	$\mathcal{O}(1)$
	Varhardan	$\mathbb{E}\left[\left\|\mathbf{s}(\mathbf{X}_t) - \exp^{-1}_{\mathbf{X}_t}(\mathbf{X}_s)/t\right\|\right]$	✘	✔	$\mathcal{O}(1)$
ISM	Deterministic	$\mathbb{E}\left[\left\|\mathbf{s}(t, \mathbf{X}_t)\right\|^2 + 2\,\mathrm{div}(t, \cdot)(\mathbf{X}_t)\right]$	✘	✘	$\mathcal{O}(d)$
	Stochastic	$\mathbb{E}\left[\left\|\mathbf{s}(t, \mathbf{X}_t)\right\|^2 + 2\varepsilon^\top\,\mathrm{div}(t, \cdot)(\mathbf{X}_t)\varepsilon\right]$	✘	✘	$\mathcal{O}(1)$

Ingredient \ Space	Euclidean	'Generic' Manifold	Compact Manifold
Forward Process	OU	Langevin dynamics	Langevin dynamics
Base distribution	Gaussian	Wrapped normal	Uniform
Time reversal	Cattiaux, 2021	Theorem 1	Theorem 1
SDE Discretisation	Eular-Maruyama	GRW	GRW
Score-matching	Denoising/Implicit	Denoising/Implicit	Denoising/Implicit
Sample $p_{t\|s}(\mathbf{X}_s)$	Analytic	Discretise SDE	Discretise SDE
$\nabla_{\mathbf{X}_t} \log p_{t\|0}(\mathbf{X}_t\|\mathbf{X}_0)$	Analytic	✗ - Use ISM	Strum-Louiville & Varadhan Approx.

Experimental Validation

Baseline Methods

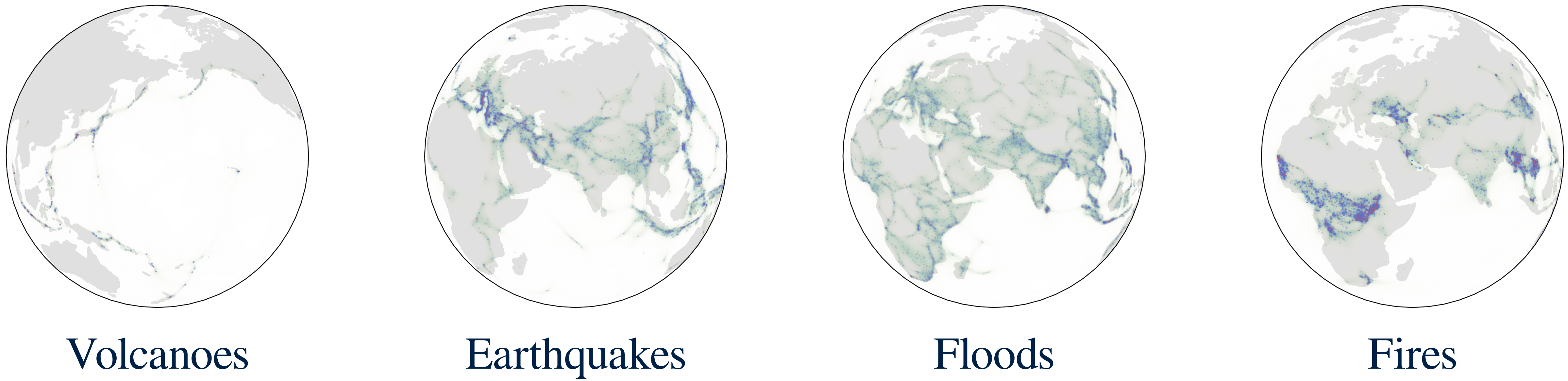
Riemannian Continuous Normalising Flows (RCNFs) *[Mathiue & Nickel 2020]*

- Map a simple density under a vector field flow to a complex density.
- Compute the change in density via the log-det-Jacobian of this flow.
- Train with maximum likelihood.
- Requires full forward/backward simulation to train.

Moser Flows *[Rozen et al. 2020]*

- Specify the *form* for the vector field flow as the linear interpolation of the start/end distributions.
- 👍 Exploit a property to get simulation-free likelihoods for training.
- 👎 Require a regulariser that integrates over the whole domain.

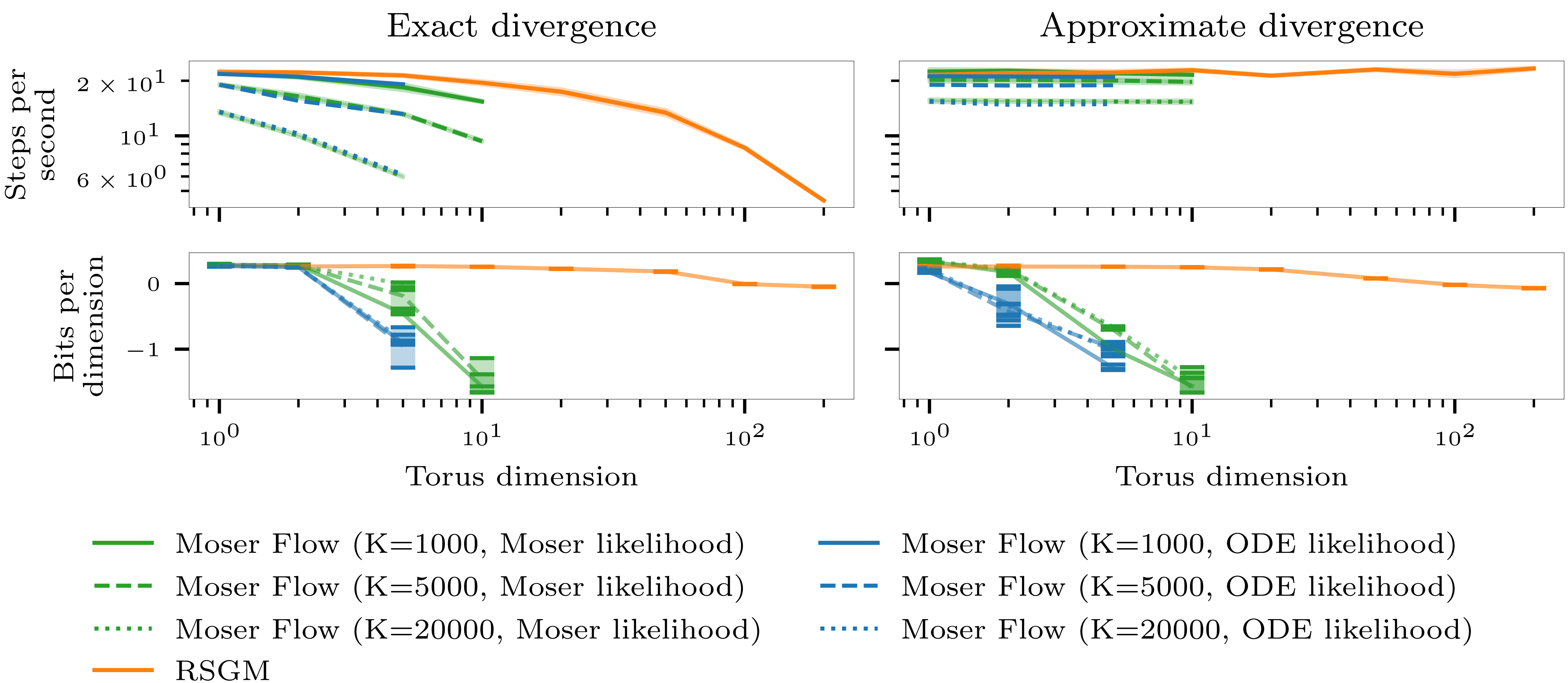
Earth-Science Datasets



Method	Volcanoes	Earthquakes	Floods	Fires
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	$-4.21_{\pm 0.17}$	$\mathbf{-0.16_{\pm 0.06}}$	$\mathbf{0.57_{\pm 0.10}}$	$\mathbf{-1.28_{\pm 0.05}}$
Sterorgraphic SGM	$-3.80_{\pm 0.27}$	$\mathbf{-0.19_{\pm 0.05}}$	$\mathbf{0.59_{\pm 0.07}}$	$\mathbf{-1.28_{\pm 0.12}}$
Riemannian SGM	$-4.92_{\pm 0.25}$	$\mathbf{-0.19_{\pm 0.07}}$	$\mathbf{0.45_{\pm 0.17}}$	$\mathbf{-1.33_{\pm 0.06}}$
Dataset Size	827	6120	4875	12809

High-Dimension Torii

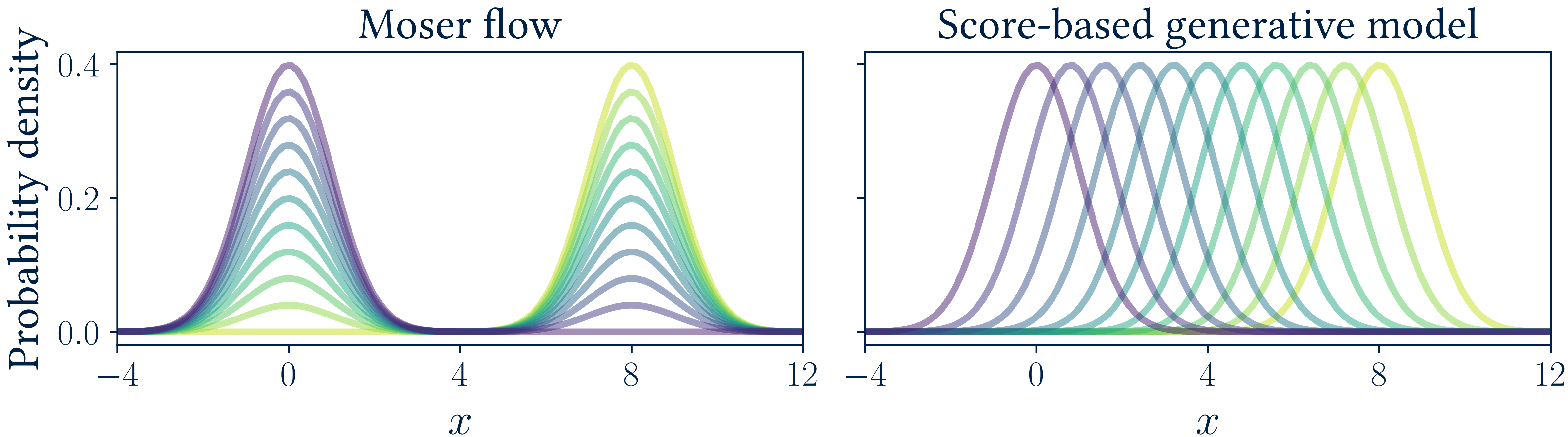
Place a 2-mode mixture-of-Gaussian ditribution on $\mathcal{S}_1^n$.



Synthetic $SO(3)$ data

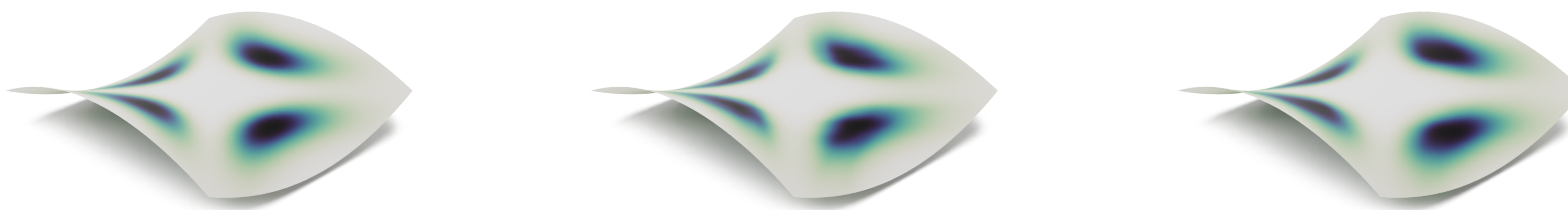
Place a M-mode mixture-of-Gaussian ditribution on $SO(3)$.

Method	$M = 16$		$M = 32$		$M = 64$	
	LL	**NFE** ($\times 10^3$)	**LL**	**NFE** ($\times 10^3$)	**LL**	**NFE** ($\times 10^3$)
Moser Flow	$0.85_{\pm 0.03}$	$2.3_{\pm 0.5}$	$0.17_{\pm 0.03}$	$2.3_{\pm 0.9}$	$-0.49_{\pm 0.02}$	$7.3_{\pm 1.4}$
Exp-wrapped SGMs	$0.87_{\pm 0.04}$	$0.5_{\pm 0.1}$	$0.16_{\pm 0.03}$	$0.5_{\pm 0.0}$	$-0.58_{\pm 0.04}$	$0.5_{\pm 0.0}$
RSGM	$0.89_{\pm 0.03}$	$0.1_{\pm 0.0}$	$0.20_{\pm 0.03}$	$0.1_{\pm 0.0}$	$-0.49_{\pm 0.02}$	$0.1_{\pm 0.0}$



Moser flow
Score-based generative model
Probability density
0.4
0.2
0.0
−4
0
4
8
12
x

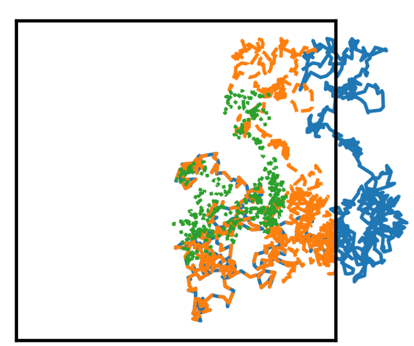
Synthetic Hyperbolic Distributions



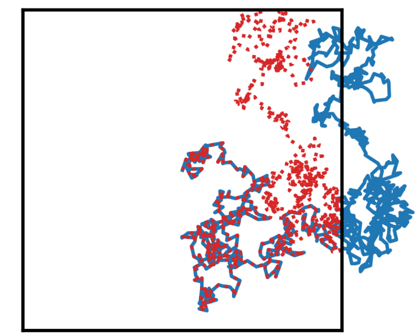
Target Exp-wrapped SGM RSGM

Research Outline

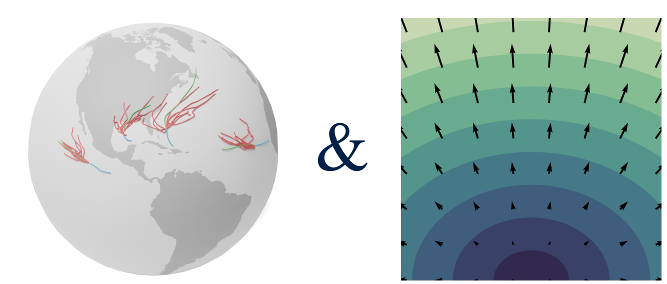
Manifolds
Neurips 2022



Manifolds with Boundary
TMLR 2023

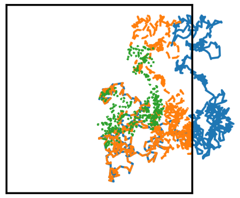


Better Manifolds with Boundary
Arxiv 2023



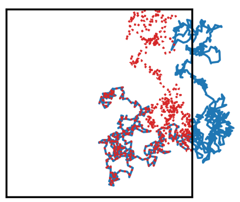
Fields and Paths on Manifolds
Arxiv 2023

Research Outline



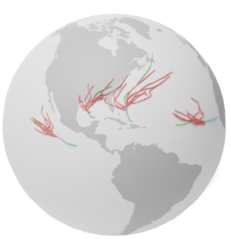
Manifolds with Boundary
TMLR 2023

- When our spaces have boundaries, normal SDEs will escape.
- Replace these with *log-barrier* and *reflected* SDEs.
- Investigate these and show how to make them work in practise.

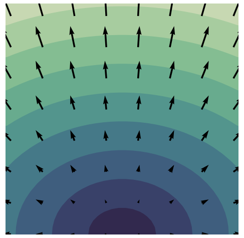


Better Manifolds with Boundary
Arxiv 2023

- Reflected SDEs are expensive to discretise in practise.
- Introduce a new sampling scheme based on Metropolis sampling.
- Show that this scheme works effectivly and very fast in practise.



&



Fields and Paths on Manifolds
Arxiv 2023

- What of we want to think not about distribution on manifolds but:
  - Distributions on *functions on manifolds*.
  - Distributions on *paths on manifolds*.

Thanks for listening!

Thanks for listening!

MJHUTCHINSON.INFO @MHUTCHINSON141

Riemannian Score-Based Generative Modelling.
V. D. Bortoli*, E. Mathieu*, M. Hutchinson*, J. Thornton, Y. W. Teh, A. Doucet. *Neurips*, 2022.

Diffusion Models for Constrained Domains.
N. Fishman, L. Klarner, V. D. Bortoli, E. Mathieu, M. Hutchinson. *TMLR*, 2023.

Metropolis Sampling for Constrained Diffusion Models .
N. Fishman, L. Klarner, E. Mathieu, M. Hutchinson, V. D. Bortoli. *arXiv:2307.05439*, 2023.

Geometric Neural Diffusion Processes.
E. Mathieu*, V. Dutordoir*, M. Hutchinson*, V. D. Bortoli, Y. W. Teh, R. Turner. *arXiv:2307.05431*, 2023.