Riemannian Score-Based Generative Modelling

Valentin De Bortoli*, Emile Mathieu*, Michael Hutchinson*, James Thornton, Yee Whye Teh, Arnaud Doucet

Department of Statistics, University of Oxford, United Kingdom. *Equal contribution.

Motivation and overview

- **Score-based generative models** (SGMs) [6, 2] have shown great success for **modelling flexible distributions**.
- **Data** is often naturally described on **Riemannian manifolds** such as spheres, torii, and Lie groups, whereas standard SGMs assume a **flat geometry**, making them ill-suited.
- We introduce **Riemannian SGMs**, a model which admits the parametrization of **flexible distributions on manifolds** by simulating the time reversal of continuous diffusion process.

Contributions

- We establish that the corresponding **time-reversal process** is also a diffusion whose drift includes the Stein score.
- Rely on Geodesic Random Walk for **sampling** processes [3].
- We provide theoretical **convergence bounds** for RSGMs.
- We **empirically demonstrate** that RGSMs perform and scale better than recent baselines [4, 5].

Ingredient \ Space	Euclidean	Manifolds / Compact manifolds
Forward process	Ornstein–Uhlenbeck	Langevin Dynamics / Brownian
Base distribution p_{ref}	Gaussian	Wrapped Normal / Uniform
Time reversal	[1, Theorem 4.9]	Theorem 1
Sampling of the forward	Direct	Geodesic Random Walk
Sampling of the backward	Euler–Maruyama	Geodesic Random Walk

Table 1: SGM on Euclidean spaces vs RSGM on Riemannian manifolds.

Noising and denoising processes

We rely on **Langevin dynamics** for the forward noising process

$$\mathrm{d}\mathbf{X}_t = -\tfrac{1}{2}\nabla_{\mathbf{X}_t} U(\mathbf{X}_t)\mathrm{d}t + \mathrm{d}\mathbf{B}_t^{\mathcal{M}}, \tag{1}$$

which admits the invariant density $\mathrm{d}p_{\text{ref}}/\mathrm{dVol}_{\mathcal{M}}(x) \propto \mathrm{e}^{-U(x)}$. A convenient choice for p_{ref} is the Wrapped normal distribution, and on compact manifolds we choose the uniform $p_{\text{ref}} = 1/\mathrm{Vol}_{\mathcal{M}}$.

Theorem 1: Time-reversed diffusion

Let $(\mathbf{Y}_t)_{t\in[0,T]} = (\mathbf{X}_{T-t})_{t\in[0,T]}$ 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

$$\mathrm{d}\mathbf{Y}_t = \{\tfrac{1}{2}\nabla_{\mathbf{Y}_t} U(\mathbf{Y}_t) + \nabla \log p_{T-t}(\mathbf{Y}_t)\}\mathrm{d}t + \mathrm{d}\mathbf{B}_t^{\mathcal{M}}. \tag{2}$$

(a) GRW vs Brownian motion density (b) GRW step (c) Trajectory

Algorithm 1 GRW (Geodesic Random Walk)

Require: $T, N, X_0^\gamma, b, \sigma, \mathrm{P}, \gamma = T/N$

- **for** $k \in \{0,\dots,N-1\}$ **do**
  - $Z_{k+1} \sim \mathrm{N}(0, \mathrm{Id})$ ▷ Gaussian on tangent space $\mathrm{T}_x\mathcal{M}$
  - $W_{k+1} = \gamma b(k\gamma, X_k^\gamma) + \sqrt{\gamma}\sigma(k\gamma, X_k^\gamma)Z_{k+1}$ ▷ Euler–Maruyama step
  - $X_{k+1}^\gamma = \exp_{X_k^\gamma}[W_{k+1}]$ ▷ Geodesic projection onto $\mathcal{M}$

Score-based generative modelling

Algorithm 2 RSGM (Riemannian Score-Based Generative Model)

Require: $\varepsilon, T, N, \{X_0^m\}_{m=1}^M, \mathrm{loss}, \mathbf{s}, \theta_0, N_{\text{iter}}, p_{\text{ref}}, \mathrm{P}$

- **for** $n \in \{0,\dots,N_{\text{iter}}-1\}$ **do** /// TRAINING ///
  - $X_0 \sim (1/M)\sum_{m=1}^M \delta_{X_0^m}$ ▷ Random mini-batch from dataset
  - $t \sim U([\varepsilon, T])$ ▷ Uniform sampling between ε and T
  - $\mathbf{X}_t = \mathrm{GRW}(t, N, X_0, 0, \mathrm{Id}, \mathrm{P})$ ▷ Approximate forward diffusion with Algorithm 1
  - $\ell(\theta_n) = \ell_t(T, N, X_0, \mathbf{X}_t, \mathrm{loss}, \mathbf{s}_{\theta_n})$ ▷ Compute score matching loss from Table 2
  - $\theta_{n+1} = \texttt{optimizer\_update}(\theta_n, \ell(\theta_n))$ ▷ ADAM optimizer step
- $\theta^\star = \theta_{N_{\text{epoch}}}$
- $Y_0 \sim p_{\text{ref}}$ /// SAMPLING /// ▷ Sample from uniform distribution
- $b_\theta^\star(t,x) = \frac{1}{2}\nabla_x U(x) + \mathbf{s}_{\theta^\star}(T-t, x)$ for any $t \in [0,T]$, $x \in \mathcal{M}$ ▷ Reverse process drift
- $\{Y_k\}_{k=0}^N = \mathrm{GRW}(T, N, Y_0, b_{\theta^\star}, \mathrm{Id}, \mathrm{P})$ ▷ Approximate reverse diffusion with Algorithm 1

- The following result ensures that RSGM generates samples which are close to p_0.

Theorem 2: Quantitative bounds for RSGM

Under mild assumption over p_0, assuming that $\mathcal{M}$ is compact and that there exists $\mathtt{M} \geq 0$ such that for any $t \in [0,T]$ and $x \in \mathcal{M}$, $\|\mathbf{s}_{\theta^\star}(t,x) - \nabla \log p_t(x)\| \leq \mathtt{M}$, 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.

$$\mathbf{W}_1(\mathcal{L}(Y_N), p_0) = C(\mathrm{e}^{-\lambda_1 T} + \sqrt{T/2}\mathtt{M} + \mathrm{e}^T\gamma^{1/2}), \tag{3}$$

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

Score matching on compact manifolds

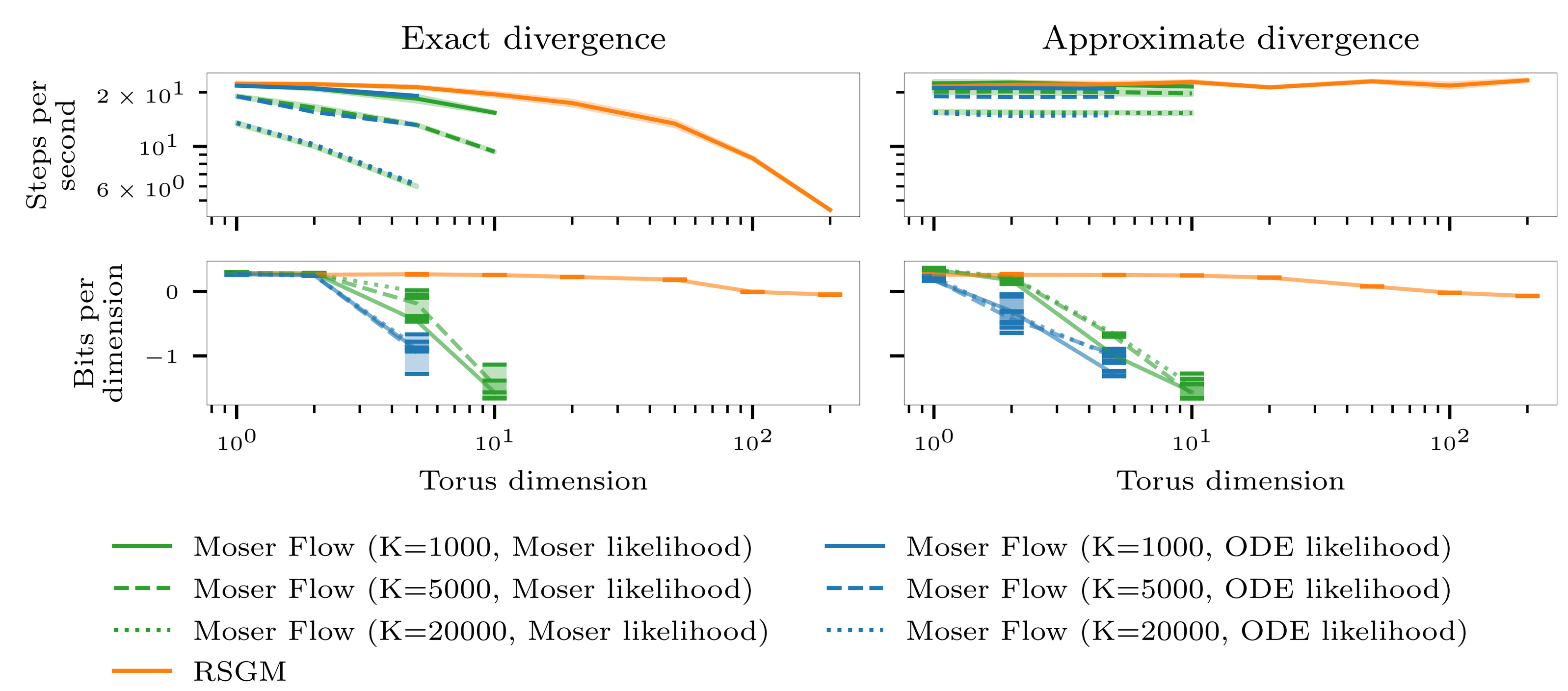
- The heat kernel is given by $p_{t|0}(x_t|x_0) = \sum_{j\in\mathbb{N}} \mathrm{e}^{-\lambda_j t}\phi_j(x_0)\phi_j(x_t)$ (Sturm–Liouville).
- Truncation: $S_{J,t}(x_0, x_t) \triangleq \nabla_{x_t} \log \sum_{j=0}^J \mathrm{e}^{-\lambda_j t}\phi_j(x_0)\phi_j(x_t) \approx \nabla_{x_t} \log p_t(x_t|x_0)$.

Loss	Approximation	Loss function	Requirements $p_{t\|0}$	Requirements $\exp_{\mathbf{X}_t}^{-1}$	Complexity
$\ell_{t\|0}$ (DSM)	None	$\frac{1}{2}\mathbb{E}\left[\|\mathbf{s}(\mathbf{X}_t) - \nabla_{\mathbf{X}_t}\log p_t(\mathbf{X}_t\|\mathbf{X}_0)\|^2\right]$	–	–	–
	Truncation	$\frac{1}{2}\mathbb{E}\left[\|\mathbf{s}(\mathbf{X}_t) - S_{J,t}(\mathbf{X}_0, \mathbf{X}_t)\|^2\right]$	✓	✗	$\mathcal{O}(1)$
	Varhadan	$\frac{1}{2}\mathbb{E}\left[\|\mathbf{s}(\mathbf{X}_t) - \exp_{\mathbf{X}_t}^{-1}(\mathbf{X}_0)/t\|^2\right]$	✗	✓	$\mathcal{O}(1)$
ℓ_t^{im} (ISM)	Deterministic	$\mathbb{E}\left[\frac{1}{2}\|\mathbf{s}(\mathbf{X}_t)\|^2 + \mathrm{div}(\mathbf{s})(\mathbf{X}_t)\right]$	✗	✗	$\mathcal{O}(d)$
	Hutchinson	$\mathbb{E}\left[\frac{1}{2}\|\mathbf{s}(\mathbf{X}_t)\|^2 + \varepsilon^\top\partial\mathbf{s}(\mathbf{X}_t)\varepsilon\right]$	✗	✗	$\mathcal{O}(1)$

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

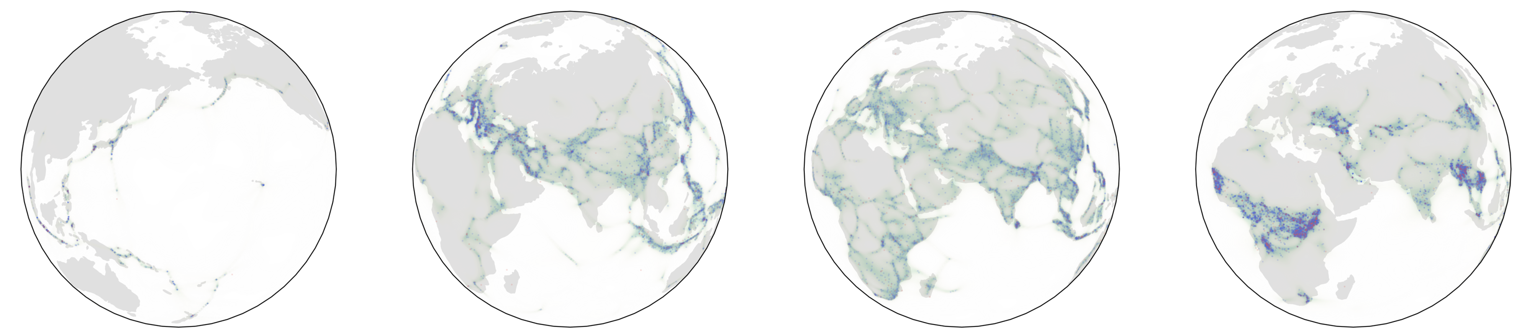
Synthetic data on torii

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



Earth science datasets on the sphere

- We evaluate RSGMs on occurrences of earth and climate science events distributed on the surface of the earth.



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

Figure 2: Trained RSGM on earth sciences data, with density in green-blue.

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

Table 3: Negative log-likelihood. Confidence intervals computed over 5 runs.

Method	Training	Likelihood evaluation	Sampling
RCNF [4]	Solve ODE $\mathcal{O}(dN)$	Solve ODE $\mathcal{O}(dN)$	Solve ODE $\mathcal{O}(N)$
Moser [5]	Computing div $\mathcal{O}(dk)$ or $\mathcal{O}(k)$	Solve ODE $\mathcal{O}(dN)$	Solve ODE $\mathcal{O}(N)$
RSGM	Score matching $\mathcal{O}(d)$ or $\mathcal{O}(1)$	Solve ODE $\mathcal{O}(dN)$	Solve SDE $\mathcal{O}(N^*)$

Table 4: Computational complexity w.r.t. neural network passes.

Synthetic data on SO(3)

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

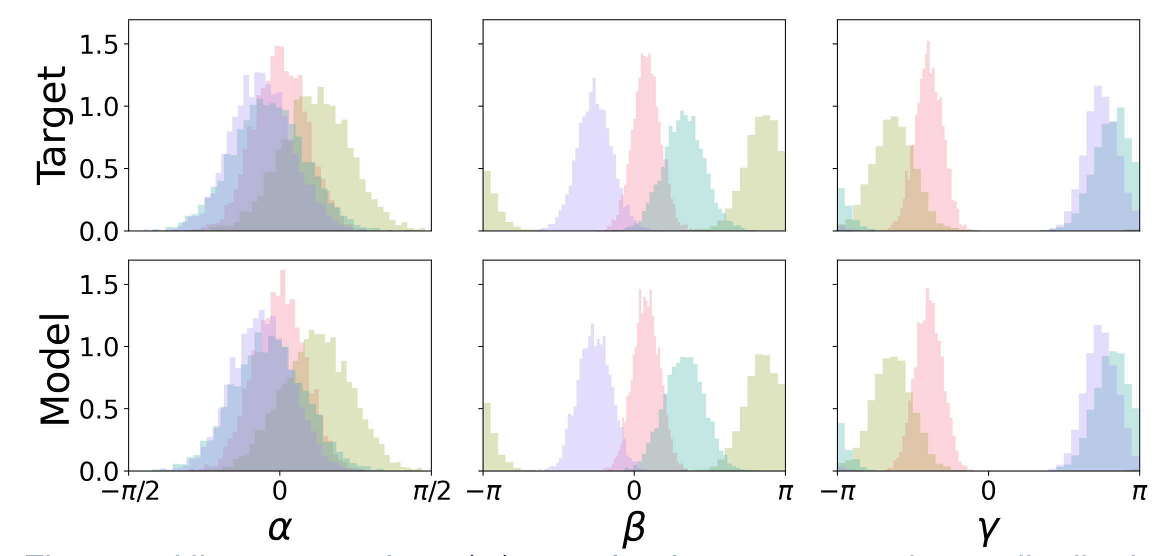


Figure 3: Histograms of $\mathrm{SO}_3(\mathbb{R})$ samples from a target mixture distribution.

Method	$M=16$ log p	$M=16$ NFE	$M=32$ log p	$M=32$ NFE	$M=64$ log p	$M=64$ 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.

References

[1] 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.
[2] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 2020.
[3] E. Jørgensen. The central limit problem for geodesic random walks. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 32(1-2):1–64, 1975.
[4] E. Mathieu and M. Nickel. Riemannian continuous normalizing flows. *arXiv preprint arXiv:2006.10605*, 2020.
[5] N. Rozen, A. Grover, M. Nickel, and Y. Lipman. Moser flow: Divergence-based generative modeling on manifolds. *Advances in Neural Information Processing Systems*, 2021.
[6] 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.