Geometric Neural Diffusion Processes

Vincent Dutordoir

15th September 2023

Alan Turing Institute
Uncertainty Quantification for Generative Modelling

UNIVERSITY OF CAMBRIDGE

Papers of Reference and Collaborators

Neural Diffusion Processes. ICML 2023.

Vincent Dutordoir

Alan Saul

Zoubin Ghahramani

Fergus Simpson

Geometric Neural Diffusion Processes. Under submission.

Émile Mathieu[*]

Vincent Dutordoir[*]

Michael Hutchinson[*]

Valentin De Bortoli

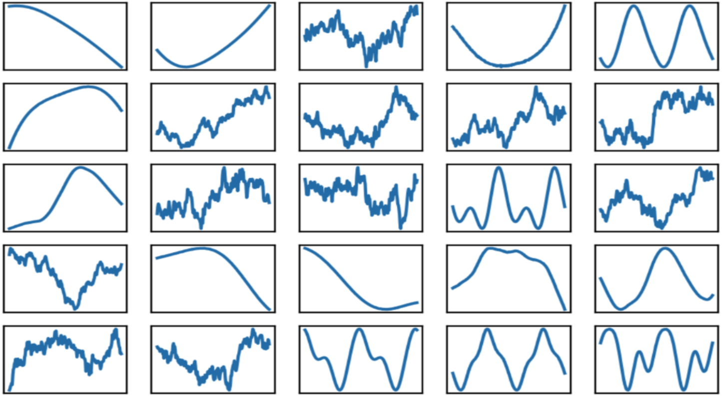
Yee Whye Teh

Richard E. Turner

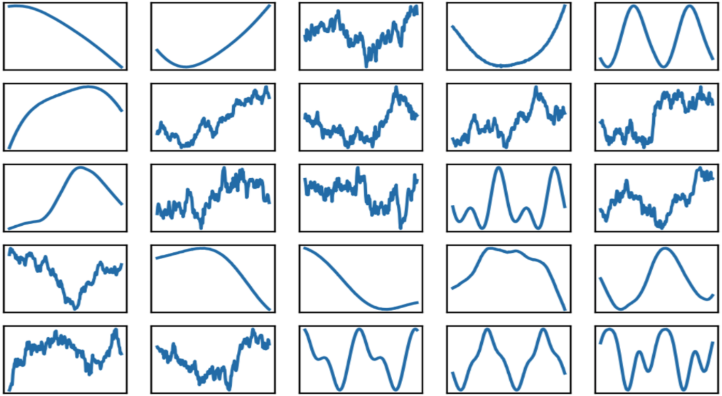
Rise of Diffusion Models

Figure 1: Samples from stable diffusion

Goal



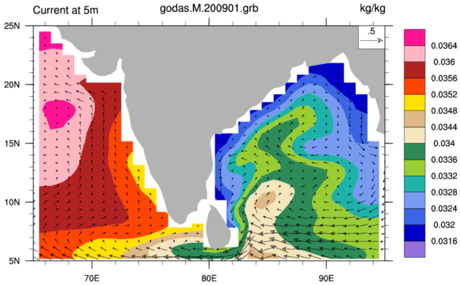
Goal



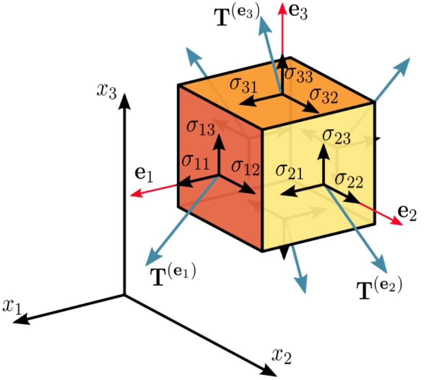
Why

- Many physical and natural phenomena are better characterised as functions.
- Meta-learn and treat limited data as originating from a function.

- Mathematical framework for modelling natural phenomena.
- Examples: Temperature $f : \mathcal{X} \to \mathbb{R}$, and wind direction on globe $f : \mathcal{S}^2 \to T\mathcal{S}^2$.



(a) Temperature map and wind vector fields.



(b) 3D stress tensor (type-2) diagram.

Prior invariances

Encode invariances w.r.t. group transformations. For a group G, we want $\forall g \in G$

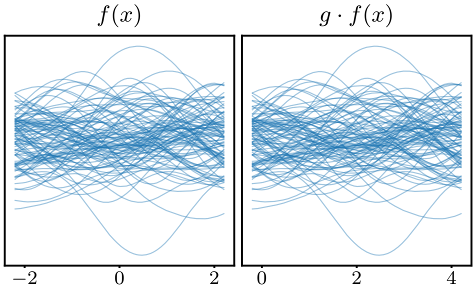
$$p(f) = p(g \cdot f) \quad \text{with} \quad g \cdot f = \rho(g) f(g^{-1}x).$$

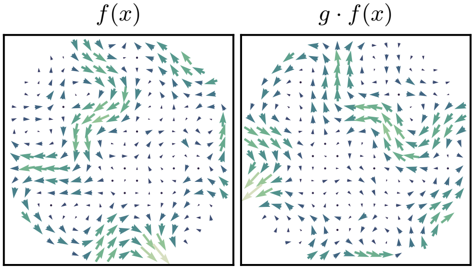
Prior invariances

Encode invariances w.r.t. group transformations. For a group G, we want $\forall g \in G$

$$p(f) = p(g \cdot f) \quad \text{with} \quad g \cdot f = \rho(g) f(g^{-1} x).$$

Examples translation invariance (stationarity) and rotational invariance.

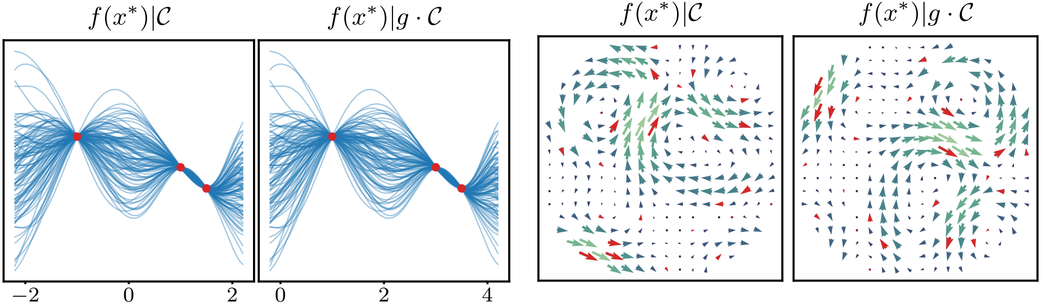




Conditional process

- Interested in the conditional process given a set of observations $\mathcal{C} = \{(x_n, y_n)\}_{n=1}$.
- If the prior is G-invariant, then the conditional is G-equivariant:

$$p(f \mid \mathcal{C}) = p(g \cdot f | g \cdot \mathcal{C}) \quad \text{where} \quad g \cdot \mathcal{C} = \{(g \cdot x_n, \rho(g) y_n)\}.$$



Continuous diffusion models

Principles of continuous diffusion models

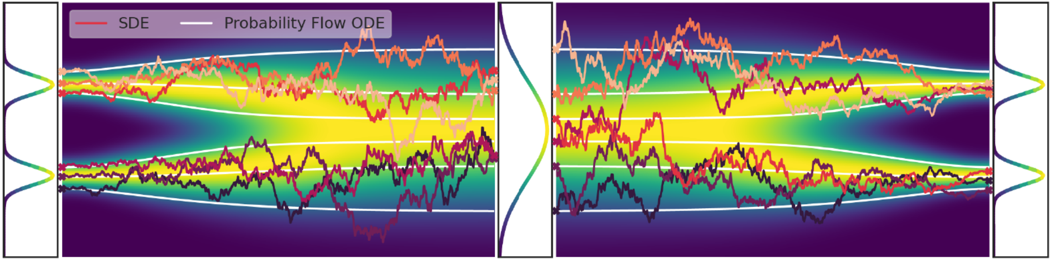


Figure 5: (Song et al., 2021)

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

Continuous noising processes

The **Forward process** progressively perturbs the data following a SDE

$$\mathrm{d}\mathbf{Y}_t = -\mathbf{Y}_t \,\mathrm{d}t + \sqrt{2}\,\mathrm{d}\mathbf{B}_t, \tag{1}$$

where $\mathbf{B}_t$ is Brownian motion (think of it conceptually as $\mathrm{d}\mathbf{B}_t/\mathrm{d}t \sim \mathcal{N}(0, \mathrm{d}t)$).

Example: 2D Gaussian data

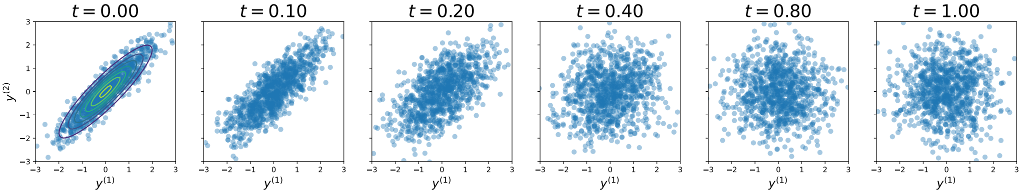


Figure 6: Forward process

Continuous score-based models: Time reversal process

Theorem 1: (Cattiaux et al., 2021; Haussmann and Pardoux, 1986)

The time-reversed process $(\bar{\mathbf{Y}}_t)_{t\geq 0} = (\mathbf{Y}_{T-t})_{t\in[0,T]}$, with forward process $\mathrm{d}\mathbf{Y}_t = -\mathbf{Y}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, also satisfies an SDE given by

$$\mathrm{d}\bar{\mathbf{Y}}_t = \left[-\bar{\mathbf{Y}}_t + 2\,\nabla \log p_t(\bar{\mathbf{Y}}_t)\right]\mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t,$$

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

Continuous score-based models: Time reversal process

Theorem 2: (Cattiaux et al., 2021; Haussmann and Pardoux, 1986)

The time-reversed process $(\bar{\mathbf{Y}}_t)_{t\geq 0} = (\mathbf{Y}_{T-t})_{t\in[0,T]}$, with forward process $\mathrm{d}\mathbf{Y}_t = -\mathbf{Y}_t \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t$, also satisfies an SDE given by

$$\mathrm{d}\bar{\mathbf{Y}}_t = \left[-\bar{\mathbf{Y}}_t + 2\,\nabla \log p_t(\bar{\mathbf{Y}}_t)\right] \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t,$$

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

Problem The Stein score $\nabla \log p_t$ $= \nabla \log \int p_{data}(\mathbf{Y}_0) p_{t|0}(\mathbf{Y}_t \mid \mathbf{Y}_0) \mathrm{d}\mathbf{Y}_0$ is intractable.

Denoising Score Matching

Parameterise score using neural network $\mathbf{s}_\theta : [0,T] \times \mathbb{R}^d \to \mathbb{R}^d$ and learn score using the Denoising Score Matching objective

$$\mathcal{L}(\theta) = \mathbb{E}[\| \mathrm{s}_\theta(t, \mathbf{Y}_t) - \nabla \log p_t(\mathbf{Y}_t|\mathbf{Y}_0)\|^2]. \tag{2}$$

Example

$$\mathrm{d}\bar{\mathbf{Y}}_t = \left[-\bar{\mathbf{Y}}_t + 2\, s_\theta(t, \bar{\mathbf{Y}}_t)\right] \mathrm{d}t + \sqrt{2}\mathrm{d}\mathbf{B}_t,$$

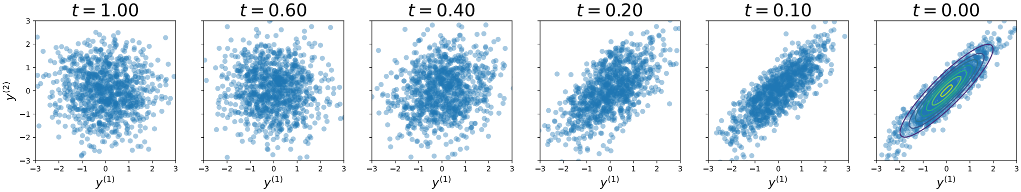
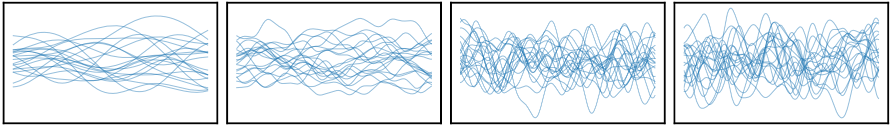


Figure 7: Reverse process

Diffusion on Function Spaces

Continuous noising process

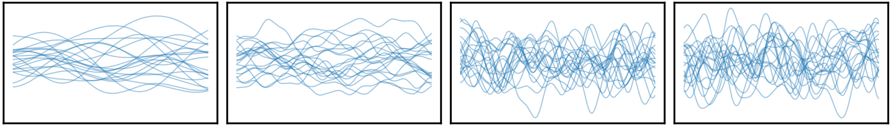


We construct the forward **noising process** $(\mathbf{Y}_t(x))_{t\geq 0} \triangleq (\mathbf{Y}_t(x^1), \ldots, \mathbf{Y}_t(x^n))_{t\geq 0}$ defined by the multivariate SDE (multivariate Ornstein-Uhlenbeck process)

$$\mathrm{d}\mathbf{Y}_t(x) = \tfrac{1}{2}\left\{m(x) - \mathbf{Y}_t(x)\right\}\beta_t \,\mathrm{d}t + \beta_t^{1/2}\mathrm{K}(x,x)^{1/2}\,\mathrm{d}\mathbf{B}_t, \tag{3}$$

where $\mathrm{K}(x,x)_{i,j} = k(x^i, x^j)$ with $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a kernel and $m : \mathcal{X} \to \mathcal{Y}$.

Continuous noising process



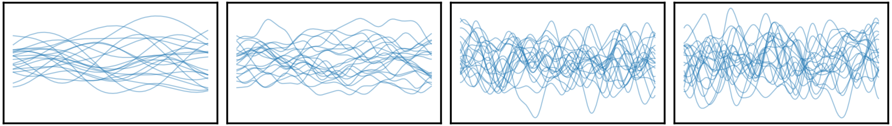
We construct the forward **noising process** $(\mathbf{Y}_t(x))_{t \geq 0} \triangleq (\mathbf{Y}_t(x^1), \ldots, \mathbf{Y}_t(x^n))_{t \geq 0}$ defined by the multivariate SDE (multivariate Ornstein-Uhlenbeck process)

$$\mathrm{d}\mathbf{Y}_t(x) = \frac{1}{2}\left\{m(x) - \mathbf{Y}_t(x)\right\}\beta_t \,\mathrm{d}t + \beta_t^{1/2}\mathrm{K}(x,x)^{1/2}\,\mathrm{d}\mathbf{B}_t, \tag{3}$$

where $\mathrm{K}(x,x)_{i,j} = k(x^i, x^j)$ with $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a kernel and $m : \mathcal{X} \to \mathcal{Y}$.

- $\mathbf{Y}_t(x) \to \mathrm{N}(m(x), \mathrm{K}(x,x))$ with geometric rate, for any $x \in \mathcal{X}^n$.
- $\mathbf{Y}_t \to \mathrm{GP}(m,k) \triangleq \mathbf{Y}_\infty$ (Phillips et al., 2022).

Continuous noising process



We construct the forward **noising process** $(\mathbf{Y}_t(x))_{t\geq 0} \triangleq (\mathbf{Y}_t(x^1), \dots, \mathbf{Y}_t(x^n))_{t\geq 0}$ defined by the multivariate SDE (multivariate Ornstein-Uhlenbeck process)

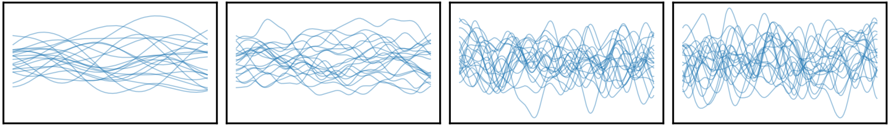
$$\mathrm{d}\mathbf{Y}_t(x) = \tfrac{1}{2}\left\{m(x) - \mathbf{Y}_t(x)\right\}\beta_t \, \mathrm{d}t + \beta_t^{1/2}\mathrm{K}(x,x)^{1/2}\,\mathrm{d}\mathbf{B}_t, \qquad (3)$$

where $\mathrm{K}(x,x)_{i,j} = k(x^i, x^j)$ with $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a kernel and $m : \mathcal{X} \to \mathcal{Y}$.

- $\mathbf{Y}_t(x) \to \mathrm{N}(m(x), \mathrm{K}(x,x))$ with geometric rate, for any $x \in \mathcal{X}^n$.
- $\mathbf{Y}_t \to \mathrm{GP}(m,k) \triangleq \mathbf{Y}_\infty$ (Phillips et al., 2022).
- $\mathbf{Y}_t$ interpolates between $\mathbf{Y}_0$ and $\mathbf{Y}_\infty$.

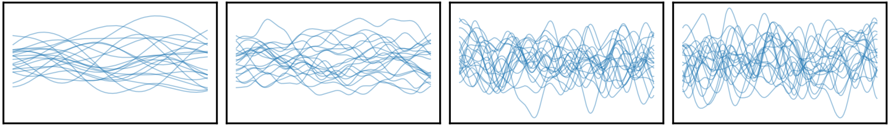
Continuous noising process

$$k(x, x') = k_{\mathrm{rbf}}(x, x') = \sigma^2 \exp\left(\frac{\|x - x'\|^2}{2l^2}\right), \text{ with } l = 1.$$

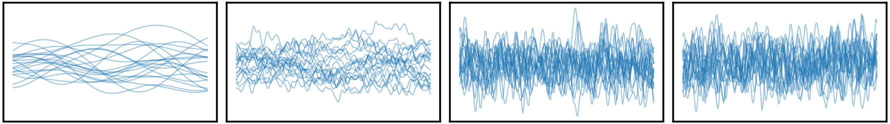


Continuous noising process

$$k(x, x') = k_{\mathrm{rbf}}(x, x') = \sigma^2 \exp\left(\frac{\|x - x'\|^2}{2l^2}\right),$$ with $l = 1$.

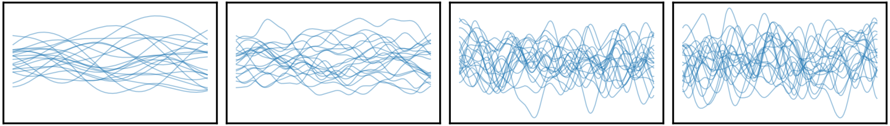


$$k(x, x') = k_{\mathrm{rbf}}(x, x'),$$ with $l = 0.2$.

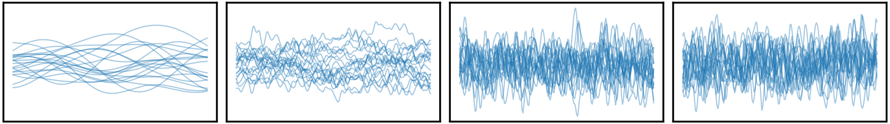


Continuous noising process

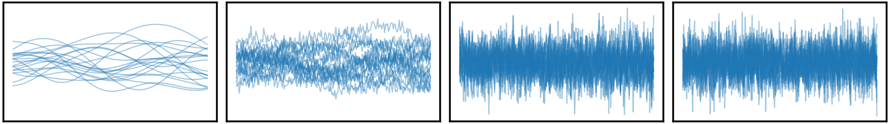
$k(x, x') = k_{\mathrm{rbf}}(x, x') = \sigma^2 \exp\left(\frac{\|x - x'\|^2}{2l^2}\right)$, with $l = 1$.



$k(x, x') = k_{\mathrm{rbf}}(x, x')$, with $l = 0.2$.

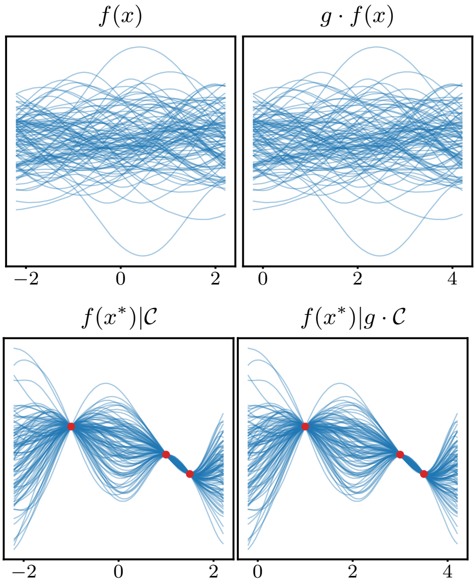


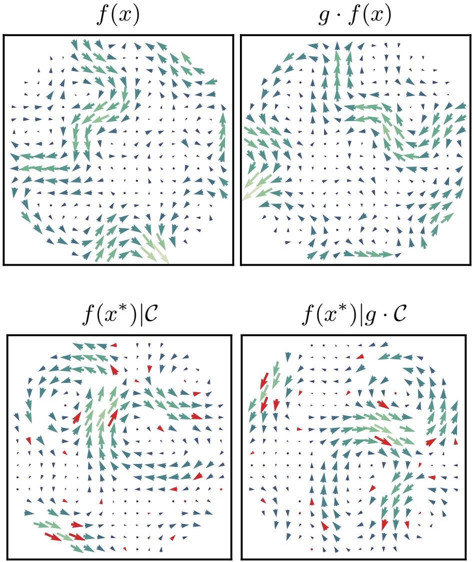
$k(x, x') = \delta_x(x')$ (The tranditional DDPM settings).



Encoding Invariances

Prior and Conditional Symmetries





Invariant neural diffusion processes

Proposition 1: Invariant Neural Diffusion Processes

The denoising process on functions as defined above and with initial sample given by $p(\bar{\mathbf{Y}}_0) = \mathrm{GP}(m, k)$ is G-invariant if

1. m and k are both G-equivariant (i.e. G-invariant Gaussian process), i.e.

$$m(g \cdot x) = \rho(g) m(x) \quad \text{and} \quad k(g \cdot x, g \cdot x') = \rho(g) k(x, x') \rho(g)^\top,$$

2. the score network is G-equivariant vector field, i.e.

$$\mathbf{s}_\theta(t, g \cdot x, \rho(g) y) = \rho(g) \mathbf{s}_\theta(t, x, y),$$

for all $x \in \mathcal{X}, g \in G$.

Conditional Process

Conditional sampling in diffusion models

Goal: Sample from $y \sim p(\cdot \mid \mathcal{C})$ given a condition $\mathcal{C}$.

Conditional sampling in diffusion models

Goal: Sample from $y \sim p(\cdot \mid \mathcal{C})$ given a condition $\mathcal{C}$.

"a hedgehog using a calculator" | "a corgi wearing a red bowtie and a purple party hat" | "robots meditating in a vipassana retreat" | "a fall landscape with a small cottage next to a lake"

Figure 10: $p(image \mid text)$

Often the condition is a property (e.g., caption).

Conditional sampling in Neural Diffusion Processes

Condition is a subspace of the state space: $\mathbf{Y}^{\mathcal{C}} = (y^{(1)}, \ldots, y^{(m)})$.

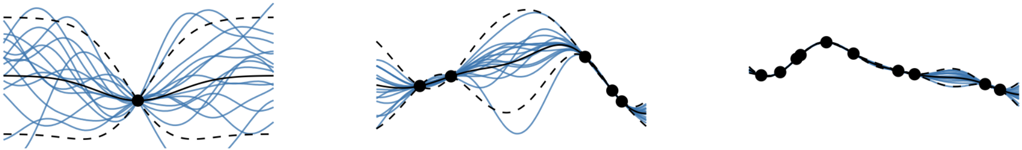


Figure 11: Conditional samples $p(\cdot \mid \mathbf{Y}^{\mathcal{C}})$.

Conditional sampling in Neural Diffusion Processes

- We need the **conditional score** $\nabla \log p_t(\mathbf{Y}_t) \implies \nabla \log p_t(\mathbf{Y}_t \mid \mathbf{Y}^{\mathcal{C}})$
- Applying Bayes rule to the conditional score

$$\nabla_{\mathbf{Y}_t} \log p_t(\mathbf{Y}_t \mid \mathbf{Y}^{\mathcal{C}}) = \nabla_{\mathbf{Y}_t} \log p_t(\mathbf{Y}_t, \mathbf{Y}^{\mathcal{C}}) - \nabla_{\mathbf{Y}_t} \log p_t(\mathbf{Y}^{\mathcal{C}}) = \nabla_{\mathbf{Y}_t} \log p_t(\mathbf{Y}_t, \mathbf{Y}^{\mathcal{C}})$$

- Use standard reverse process with score $s_\theta(t, [\mathbf{Y}_t, \mathbf{Y}^{\mathcal{C}}])$.

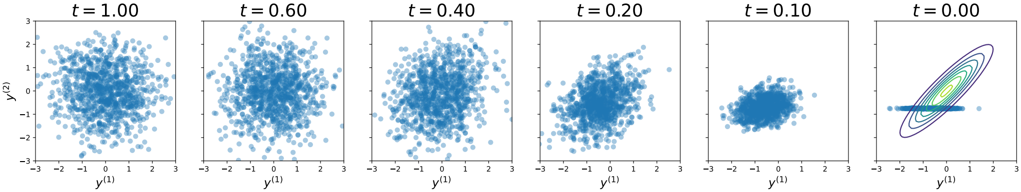
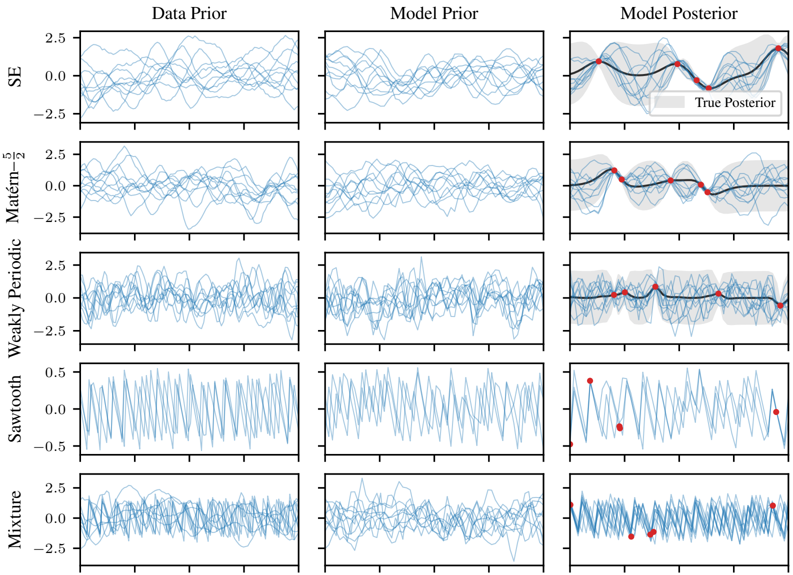


Figure 12: Conditional reverse process $p(\mathbf{Y}_0 \mid y^{(2)} = -1)$

Experimental results

1D regression: Datasets



1D regression: Predictive log-likelihood (Cont'd)

Table 1: Mean test log-likelihood (TLL) (higher is better)

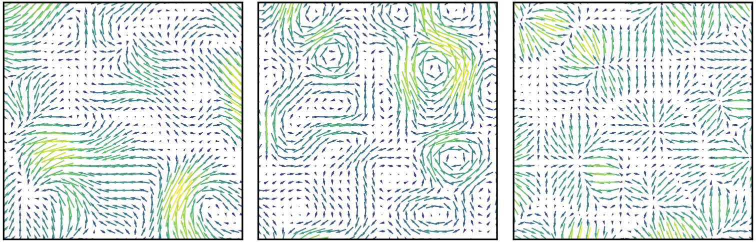
		SE	Matérn($\frac{5}{2}$)	Weakly Per.	Sawtooth	Mixture
Interpolat.	GP (optimum)	0.70 ± 0.00	0.31 ± 0.00	-0.32 ± 0.00	-	-
	T(1)−GeomNDP	$\mathbf{0.72} \pm 0.03$	$\mathbf{0.32} \pm 0.03$	$\mathbf{-0.38} \pm 0.03$	$\mathbf{3.39} \pm 0.04$	$\mathbf{0.64} \pm 0.08$
	NDP*	$\mathbf{0.71} \pm 0.03$	$\mathbf{0.30} \pm 0.03$	$\mathbf{-0.37} \pm 0.03$	$\mathbf{3.39} \pm 0.04$	$\mathbf{0.64} \pm 0.08$
	GNP	$\mathbf{0.70} \pm 0.01$	$\mathbf{0.30} \pm 0.01$	-0.47 ± 0.01	0.42 ± 0.01	0.10 ± 0.02
	ConvNP	-0.46 ± 0.01	-0.67 ± 0.01	-1.02 ± 0.01	1.20 ± 0.01	-0.50 ± 0.02

1D regression: Predictive log-likelihood (Cont'd)

Table 2: Mean test log-likelihood (TLL) (higher is better)

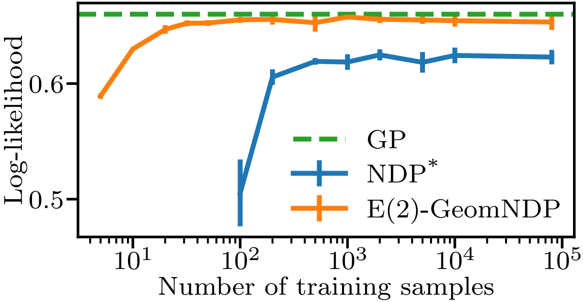
		SE	Matérn($\frac{5}{2}$)	Weakly Per.	Sawtooth	Mixture
Interpolat.	GP (optimum)	0.70 ± 0.00	0.31 ± 0.00	-0.32 ± 0.00	-	-
	T(1)−GeomNDP	$\mathbf{0.72} \pm 0.03$	$\mathbf{0.32} \pm 0.03$	$\mathbf{-0.38} \pm 0.03$	$\mathbf{3.39} \pm 0.04$	$\mathbf{0.64} \pm 0.08$
	NDP*	$\mathbf{0.71} \pm 0.03$	$\mathbf{0.30} \pm 0.03$	$\mathbf{-0.37} \pm 0.03$	$\mathbf{3.39} \pm 0.04$	$\mathbf{0.64} \pm 0.08$
	GNP	$\mathbf{0.70} \pm 0.01$	$\mathbf{0.30} \pm 0.01$	-0.47 ± 0.01	0.42 ± 0.01	0.10 ± 0.02
	ConvNP	-0.46 ± 0.01	-0.67 ± 0.01	-1.02 ± 0.01	1.20 ± 0.01	-0.50 ± 0.02
Generalisat.	GP (optimum)	0.70 ± 0.00	0.31 ± 0.00	-0.32 ± 0.00	-	-
	T(1)−GeomNDP	$\mathbf{0.70} \pm 0.02$	$\mathbf{0.31} \pm 0.02$	$\mathbf{-0.38} \pm 0.03$	$\mathbf{3.39} \pm 0.03$	$\mathbf{0.62} \pm 0.02$
	NDP*	*	*	*	*	*
	GNP	$\mathbf{0.69} \pm 0.01$	$\mathbf{0.30} \pm 0.01$	-0.47 ± 0.01	0.42 ± 0.01	0.10 ± 0.02
	ConvNP	-0.46 ± 0.01	-0.67 ± 0.01	-1.02 ± 0.01	1.19 ± 0.01	-0.53 ± 0.02

2D invariant Gaussian vector fields



Model	SE	Curl-free	Div-free
GP	$0.56_{\pm 0.00}$	$0.66_{\pm 0.00}$	$0.66_{\pm 0.00}$
NDP	$0.55_{\pm 0.00}$	$0.62_{\pm 0.01}$	$0.62_{\pm 0.01}$
E(2)-GeomNDP	$\mathbf{0.56_{\pm 0.01}}$	$\mathbf{0.65_{\pm 0.01}}$	$\mathbf{0.66_{\pm 0.01}}$

2D invariant Gaussian vector fields (Cont'd)



Global tropical cyclone trajectory prediction

Learn $f : \mathbb{R} \to \mathcal{S}^2$ from hurricane trajectory data (Knapp et al., 2018).

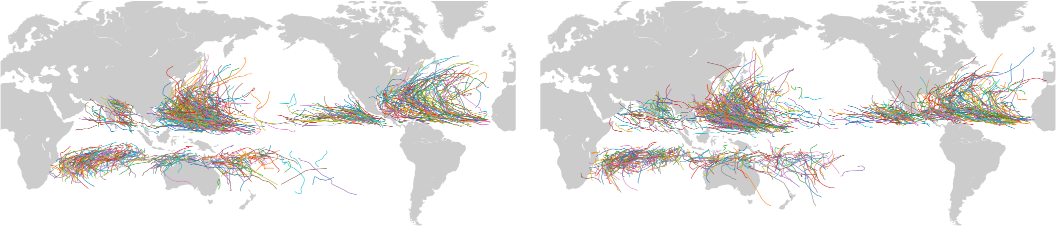
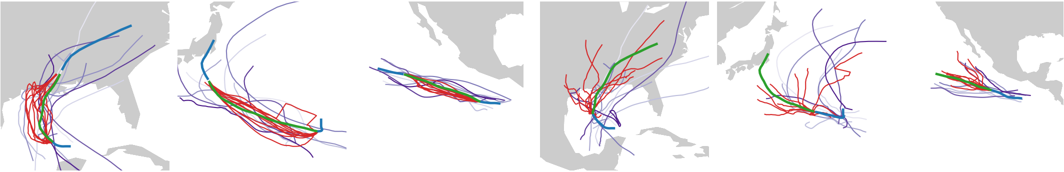


Figure 13: *Left:* 1000 samples from the training data. *Right:* 1000 samples from trained model.

Global tropical cyclone trajectory prediction (Cont'd)



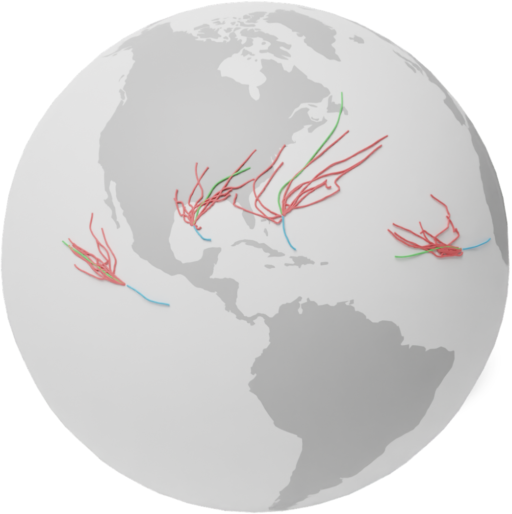
(a) Interpolation **(b)** Extrapolation

Model	Test data	Interpolation		Extrapolation	
	Likelihood	Likelihood	MSE (km)	Likelihood	MSE (km)
GeomNDP($\mathbb{R} \to \mathcal{S}^2$)	$\mathbf{802}_{\pm 5}$	$\mathbf{535}_{\pm 4}$	$\mathbf{162}_{\pm 6}$	$\mathbf{536}_{\pm 4}$	$\mathbf{496}_{\pm 14}$
Stereo GP ($\mathbb{R} \to \mathbb{R}^2/\{0\}$)	$393_{\pm 3}$	$266_{\pm 3}$	$2619_{\pm 13}$	$245_{\pm 2}$	$6587_{\pm 55}$
NDP ($\mathbb{R} \to \mathbb{R}^2$)	-	-	$166_{\pm 22}$	-	$769_{\pm 48}$
GP ($\mathbb{R} \to \mathbb{R}^2$)	-	-	$6852_{\pm 41}$	-	$8138_{\pm 87}$

Recap: Geometric diffusion neural processes

- Aim: probabilistic model over features fields.
- Constructed diffusion models over function space by correlating finite marginals
- Incorporating group invariance by
  - targetting invariant Gaussian processes and
  - parameterising the score with an equivariant neural network
- Sampling from the conditional process
- Empirically demonstrated modelling capacity on scalar and vector fields, with Euclidean and spherical output space

Thank you for your attention. Questions?



Credits to Michael Hutchinson for this 3D render.

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 pages 13, 14.

U. G. Haussmann and E. Pardoux. Time reversal of diffusions. *The Annals of Probability*, 14(4):1188–1205, 1986. Cited on pages 13, 14.

H. J. Knapp Kenneth R. Diamond, J. P. Kossin, M. C. Kruk, and C. J. I. Schreck. International Best Track Archive for Climate Stewardship (IBTrACS) Project, Version 4. Technical report, NOAA National Centers for Environmental Information, 2018. DOI: https://doi.org/10.25921/82ty-9e16. Cited on page 37.

A. Phillips, T. Seror, M. Hutchinson, V. De Bortoli, A. Doucet, and E. Mathieu. Spectral Diffusion Processes. Nov. 2022. URL: http://arxiv.org/abs/2209.14125. Cited on pages 17–19.

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 page 11.