DEEP RANDOM FEATURES FOR SCALABLE INTERPOLATION OF SPATIOTEMPORAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

The rapid growth of earth observation systems calls for a scalable approach to interpolate remote-sensing observations. These methods in principle, should acquire more information about the observed field as data grows. Gaussian processes (GPs) are candidate model choices for interpolation. However, due to their poor scalability, they usually rely on inducing points for inference, which restricts their expressivity. Moreover, commonly imposed assumptions such as stationarity prevents them from capturing complex patterns in the data. While deep GPs can overcome this issue, training and making inference with them are difficult, again requiring crude approximations via inducing points. In this work, we instead approach the problem through Bayesian deep learning, where spatiotemporal fields are represented by deep neural networks, whose layers share the inductive bias of stationary GPs on the plane/sphere via random feature expansions. This allows one to (1) capture high frequency patterns in the data, and (2) use mini-batched gradient descent for large scale training. We experiment on various remote sensing data at local/global scales, showing that our approach produce competitive or superior results to existing methods, with well-calibrated uncertainties.

1 INTRODUCTION

The advent of earth observation systems have made it possible to monitor virtually all of earth's atmosphere and the ocean at unprecedented scales. This development has been pivotal to the understanding of anthropogenic impact on the environment, including global warming and rise in sea level. Hence, it is crucial that we are able to process the voluminous data effectively and extract maximal information from it to make better informed decisions in our path to achieving sustainable development goals.

However, observations from satellite products are inherently sparse in space-time, requiring methods to effectively fill in the gap at unobserved locations (Le Traon et al., 1998). This typically relies on data assimilation techniques such as the ensemble Kalman filter (Evensen, 2003), which requires one to have access to a physical model that describes the evolution of the field. While this can produce detailed and accurate reconstructions of the field, the physical models are typically expensive to run at high resolutions, often requiring access to high performance compute clusters. This can be challenging when one does not have the expertise nor the resources to gain access and/or run the models. On the other hand, statistical methods such as Gaussian process regression (GPR, Williams & Rasmussen (2006)) can be deployed. However, GPR scales poorly to large data sets, necessitating approximate inference schemes such as sparse Gaussian processes (Titsias, 2009), which may result in crude approximations if the underlying process does not have sufficiently large lengthscale or smoothness (Burt et al., 2019). Moreover, kernels used for GPR are often too simplistic, which can prevent learning of detailed fluctuations in the underlying non-stationary and multi-scale field. Deep Gaussian processes (DGPs) (Damianou & Lawrence, 2013) have emerged as an attractive solution to the latter problem. However, they still suffer from the difficulty of computing the posterior, again requiring variational inference to learn only a crude approximation to the true posterior.

In recent years, Bayesian deep learning (BDL) have emerged as an alternative paradigm for statistical modelling, which combines the flexibility and scalability of deep learning methods with Bayesian modelling principles (Papamarkou et al., 2024). In our current setting, we can approach spatiotemporal interpolation using BDL, by representing the ground truth underlying field $f^\dagger$ by a

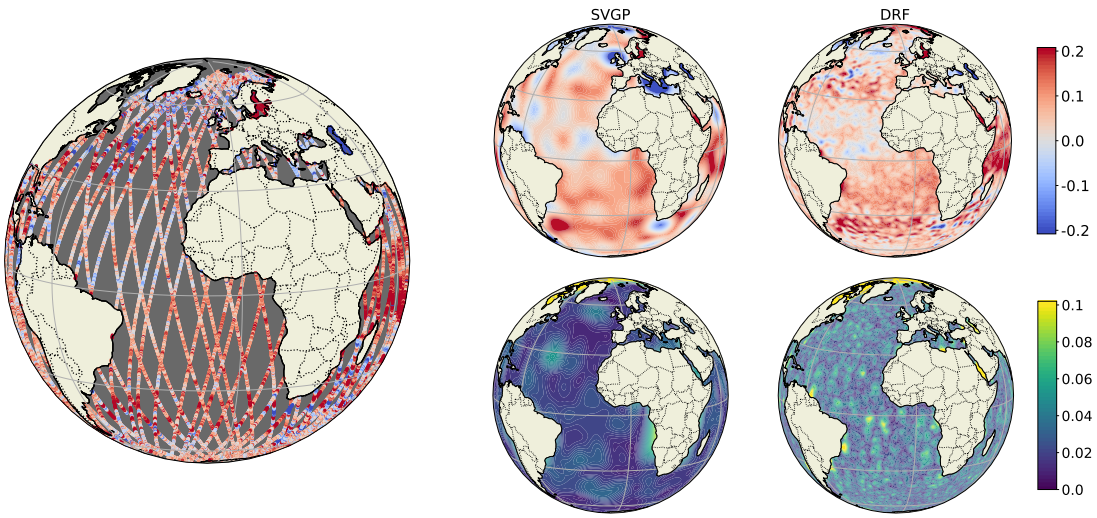

Figure 1: We propose deep random features (DRF) for accurate and flexible interpolation of satellite measurements of the earth's surface (Left). Compared to sparse variational GPs (SVGP, Centre), an ensemble of DRFs is able to achieve more detailed reconstructions of the field with sensible uncertainty estimates (Right).

Bayesian neural network $f_\theta : \mathcal{X} \to \mathcal{Y}$, (here, $\mathcal{X}$ denotes the spatiotemporal input space and $\mathcal{Y}$ the output signals) and training on input-output pairs $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$ for $x_n \in \mathcal{X}$ and $y_n \in \mathcal{Y}$, corresponding to earth observations. However, naïve design choices for f_θ can lead to poor reconstructions of $f^\dagger$; for example, a vanilla deep ReLU network is bound to perform poorly as it fails to learn high frequency features Tancik et al. (2020). On the other hand, deep neural networks with trigonometric activations (Sitzmann et al., 2020; Lu & Shafto, 2022) have emerged as an effective model for representing high-frequency spatiotemporal signals. However, they are not designed for interpolation of sparse data in mind and are therefore prone to overfitting.

Taken altogether, we propose to design f_θ inspired by DGPs, such that it retains the learning capacity of DNNs, while having the interpretability and inductive biases of GPs. Our main contributions are as follows: We propose the use of kernel-derived random features (Rahimi & Recht, 2007) as building blocks for BNNs to model spatiotemporal fields. We demonstrate through extensive experiments that they are capable of capturing fine-scale information in data, while being able to quantify uncertainty accurately by considering deep ensembles. Furthermore, motivated by recent developments in geometric probabilistic modelling (Borovitskiy et al., 2020), we also consider analogous random features on the sphere, leading to a novel DNN architecture with Gegenbauer polynomial activation functions that can model *global* weather fields that are adapted to the sphere. Our models are easily implementable in modern deep learning frameworks such as `PyTorch` and scale up to large datasets exceeding millions of data points through mini-batched gradient-based optimisation, pushing the boundary of what is currently possible with statistical interpolation.

1.1 Related works

In Cutajar et al. (2017), trigonometric feature expansion of DGPs similar to ours have been considered, with the intent of proposing a tractable variational inference (VI) scheme for DGPs. They show superior performance to mean-field VI (Damianou & Lawrence, 2013), however, have been largely overlooked due in part to the adoption of doubly stochastic VI (Salimbeni & Deisenroth, 2017), as the de facto standard method for DGP inference. DNNs with trigonometric activations have resurfaced as an object of interest more recently, with the emergence of neural radiance fields (Mildenhall et al., 2021) and subsequent work on implicit neural representations Tancik et al. (2020); Sitzmann et al. (2020). Rigorous study of trigonometric networks and their connection to DGPs have been considered in Lu & Shafto (2022), and more general investigation of wide DNNs with bottlenecks in relation to DGPs have been considered in Agrawal et al. (2020); Pleiss & Cunningham (2021). Other closely related works include Meronen et al. (2020; 2021), who study calibration of shallow

networks with periodic activations, Garnelo et al. (2018) proposes a different approach to combining aspects of GPs with DNNs, and the works Sun et al. (2020); Dutordoir et al. (2021) establish connections between neural network layers and inducing points for GPs/DGPs.

2 BACKGROUND

2.1 GAUSSIAN PROCESSES AND DEEP GAUSSIAN PROCESSES

A Gaussian process (GP) is a random function $f : \mathbb{R}^I \to \mathbb{R}$ such that for any $N > 0$ and any set of points $\boldsymbol{x}_n \in \mathbb{R}^I$ for $n = 1, \ldots, N$, we have that $(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))^\top \in \mathbb{R}^N$ is Gaussian. GPs are characterised by a mean function $m : \mathbb{R}^I \to \mathbb{R}$ and a kernel $k : \mathbb{R}^I \times \mathbb{R}^I \to \mathbb{R}$, such that $\mathbb{E}\big[f(\boldsymbol{x})\big] = m(\boldsymbol{x})$ and $\mathrm{Cov}\left[f(\boldsymbol{x}), f(\boldsymbol{x}')\right] = k(\boldsymbol{x}, \boldsymbol{x}')$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^I$ (Williams & Rasmussen, 2006). Extending these to have vector outputs $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$ is made possible by considering vector-valued means $\boldsymbol{m} : \mathbb{R}^I \to \mathbb{R}^O$ and matrix-valued kernels $\boldsymbol{k} : \mathbb{R}^I \times \mathbb{R}^I \to \mathbb{R}^{O \times O}$, satisfying $\mathbb{E}\big[f_i(\boldsymbol{x})\big] = m_i(\boldsymbol{x})$ and $\mathrm{Cov}\left[f_i(\boldsymbol{x}), f_i(\boldsymbol{x}')\right] = k_{ij}(\boldsymbol{x}, \boldsymbol{x}')$, $\forall i, j = 1, \ldots, O$. We write $\boldsymbol{f} \sim \mathcal{GP}(\boldsymbol{m}, \boldsymbol{k})$ to denote that $\boldsymbol{f}$ is a GP with mean $\boldsymbol{m}$ and kernel $\boldsymbol{k}$. A deep GP (DGP) $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$ extends GPs by considering compositions $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}^L \circ \cdots \circ \boldsymbol{f}^1(\boldsymbol{x})$, where $\boldsymbol{f}^1 : \mathbb{R}^I \to \mathbb{R}^B$, $\boldsymbol{f}^\ell : \mathbb{R}^B \to \mathbb{R}^B$ for $\ell = 2, \ldots, L-1$ and $\boldsymbol{f}^L : \mathbb{R}^B \to \mathbb{R}^O$ are vector-GPs. The intermediate states $\mathbb{R}^B$ are referred to as the *bottlenecks*. We note that DGPs are more flexible class of models than GPs. However, due to their compositional structure, they are no longer Gaussian and therefore require approximate methods for inference, e.g. using variational Bayes.

2.2 RANDOM FOURIER FEATURES

Consider a zero-mean scalar GP $f \sim \mathcal{GP}(0, k)$ for some kernel k. We say that k is *stationary* if there exists a function $\kappa : \mathbb{R}^I \to \mathbb{R}$ such that $k(\boldsymbol{x}, \boldsymbol{x}') = \kappa(\boldsymbol{x} - \boldsymbol{x}')$. In Rahimi & Recht (2007), it is shown that any stationary kernel on $\mathbb{R}^I$ can be expressed as an expectation

$$k(\boldsymbol{x}, \boldsymbol{x}') = 2\sigma^2 \mathbb{E}_{\boldsymbol{\omega}, b}\left[\cos(\boldsymbol{\omega}^\top \boldsymbol{x} + b)\cos(\boldsymbol{\omega}^\top \boldsymbol{x}' + b)\right] \tag{1}$$

$$\approx \frac{2\sigma^2}{M} \sum_{m=1}^{M} \cos(\boldsymbol{\omega}_m^\top \boldsymbol{x} + b_m)\cos(\boldsymbol{\omega}_m^\top \boldsymbol{x}' + b_m), \quad \boldsymbol{\omega}_m \sim p(\boldsymbol{\omega}), \quad b_m \sim U([0, 2\pi]) \tag{2}$$

for some $\sigma > 0$, where $p(\boldsymbol{\omega})$ is the normalised Fourier transform of the function κ and $U([0, 2\pi])$ denotes the uniform distribution in the interval $[0, 2\pi]$. From the weight-space viewpoint of GPs, equation 2 implies that we have

$$f(\boldsymbol{x}) \approx \sum_{m=1}^{M} \theta_m \phi_m(\boldsymbol{x}), \quad \theta_m \sim \mathcal{N}(0, 1), \tag{3}$$

$$\text{where} \quad \phi_m(\boldsymbol{x}) = \sqrt{2\sigma^2/M}\cos(\boldsymbol{\omega}_m^\top \boldsymbol{x} + b_m), \quad m = 1, \ldots, M, \tag{4}$$

with $\boldsymbol{\omega}_m \sim p(\boldsymbol{\omega})$ and $b_m \sim U([0, 2\pi]$. This parametric representation of f in terms of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$ is useful as it enables one to make statistical inference with cost $\mathcal{O}(NM^2)$, where N is the number of data; for $M \ll N$, this is much cheaper than the usual $\mathcal{O}(N^3)$ cost of GP inference. For more details and examples of random Fourier features, see Appendix A. Extension to vector-valued GPs $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$ is straightforward, leading to a random Fourier feature representation of the form $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{\Theta}\boldsymbol{\phi}(\boldsymbol{x})$ for $\boldsymbol{\Theta} \in \mathbb{R}^{O \times M}$ with $\Theta_{ij} \sim \mathcal{N}(0, 1), \forall i, j$ (see Appendix A).

3 DEEP RANDOM FEATURES FOR SPATIOTEMPORAL MODELLING

GPs are commonly used in spatiotemporal modelling due to their interpretability and smoothness inductive biases that are appealing to many geostatistical applications (Wikle et al., 2019). However, they are limited by their poor scalability and Gaussian assumptions. On the other hand, deep neural networks (DNN) offer a scalable, flexible modelling framework, however, do not have the desirable inductive bias of GPs. Motivated by this, we consider *deep random features* (Figure 2), which use random features corresponding to stationary GPs as building blocks for a larger neural network model tailored for spatiotemporal modelling, combining the benefits of both GPs and DNNs.

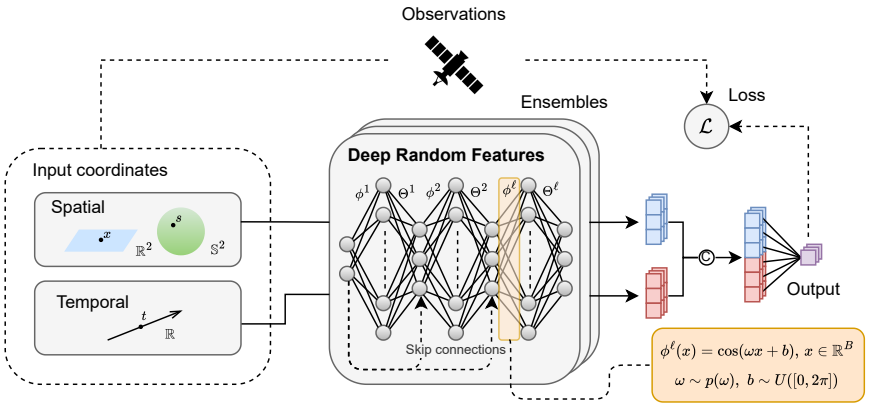

Figure 2: Illustration of spatiotemporal modelling with deep random features.

3.1 Deep random features

In Section 2.2, we have seen that a shallow GP can be approximated by a combination of random features according to equation 3. In a similar fashion, Cutajar et al. (2017) propose a random feature expansion of deep GPs, by replacing each GP layer by their corresponding random features, yielding a DNN architecture that mimic the behaviour of the original DGP. In further details, let $\boldsymbol{\phi}^1 : \mathbb{R}^I \to \mathbb{R}^H$ be a random feature (equation 4), which may be viewed equivalently as a single layer of a neural network with weights $\{\boldsymbol{\omega}_m\}_{m=1}^H$, biases $\{b_m\}_{m=1}^H$ and cosine activation. The first hidden layer in a deep GP, which itself is a vector-valued GP $\boldsymbol{f}^1 : \mathbb{R}^I \to \mathbb{R}^B$, can be approximated by a linear model

$$\boldsymbol{h}^1(\boldsymbol{x}) = \boldsymbol{\Theta}^1 \boldsymbol{\phi}^1(\boldsymbol{x}), \tag{5}$$

where $\boldsymbol{\Theta}^1 \in \mathbb{R}^{B \times H}$ with $\Theta^1_{ij} \sim \mathcal{N}(0, 1)$, $i = 1, \ldots, B$, $j = 1, \ldots, H$. Similarly, given random features $\boldsymbol{\phi}^\ell : \mathbb{R}^B \to \mathbb{R}^H$ for $\ell = 2, \ldots, L$, Gaussian weights $\boldsymbol{\Theta}^\ell \in \mathbb{R}^{B \times H}$ for $\ell = 2, \ldots, L-1$ and $\boldsymbol{\Theta}^L \in \mathbb{R}^{B \times O}$, we may consider a DNN $\boldsymbol{f}_{\boldsymbol{\Theta}} : \mathbb{R}^I \to \mathbb{R}^O$ of the form

$$\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \boldsymbol{h}^L \circ \cdots \circ \boldsymbol{h}^2 \circ \boldsymbol{h}^1(\boldsymbol{x}), \quad \text{where} \quad \boldsymbol{h}^\ell(\boldsymbol{x}) := \boldsymbol{\Theta}^\ell \boldsymbol{\phi}^\ell(\boldsymbol{x}), \quad \ell = 1, \ldots, L, \tag{6}$$

which we refer to as the random feature expansion of a DGP $\boldsymbol{f} : \mathbb{R}^I \to \mathbb{R}^O$. More generally, we may consider building DNNs independently of a DGP by using layers of the form $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{\Theta}\boldsymbol{\phi}(\boldsymbol{x})$ as building blocks for a neural network. We refer to such models as *deep random features*.

3.1.1 Training

In contrast to standard neural networks, when we train deep random features, we opt to alternate between trainable and fixed layers, where the parameters $\boldsymbol{\omega}_m^\ell, b_m^\ell$ in the layers $\boldsymbol{\phi}_\ell(\cdot)$ are fixed upon initialisation, but the parameters $\boldsymbol{\Theta} := \{\boldsymbol{\Theta}^1, \ldots, \boldsymbol{\Theta}^L\}$ are trained. This is to mimic training of DGPs from the weight-space perspective, where inference should only be made with respect to $\boldsymbol{\Theta}$. Given a dataset $\mathcal{D} = \{(\boldsymbol{X}_n, \boldsymbol{y}_n)\}_{n=1}^N$, and an arbitrary loss $\ell : \mathbb{R}^O \times \mathbb{R}^O \to \mathbb{R}$, we minimise

$$\mathcal{L}_{\text{train}}(\boldsymbol{\Theta}; \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} \ell(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{X}_n), \boldsymbol{y}_n) + \beta \|\boldsymbol{\Theta}\|^2, \tag{7}$$

for some regularisation parameter $\beta > 0$. From a generalised Bayes' perspective, this is equivalent to maximum a priori estimation with the generalised posterior $p(\boldsymbol{\Theta}|\mathcal{D}) \propto \exp(-\ell(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{X}), \boldsymbol{y}))p(\boldsymbol{\Theta})$ (Bissiri et al., 2016). Using mean-squared error as the loss and considering a shallow network, minimising equation 7 via gradient descent can be seen as sampling from the GP posterior in the infinite width limit (Lee et al., 2019). Using other losses such as the Huber loss, this can be seen as a BNN analogue to robust GP regression (Algikar & Mili, 2023; Altamirano et al., 2024).

3.1.2 SPHERICAL RANDOM FEATURES

When modelling signals over the sphere, which arises when we need to interpolate global satellite measurements (see Figure 1), we require an analogous notion of random features defined over the sphere. In Borovitskiy et al. (2020), commonly used kernels such as the Matérn kernels are extended to be defined over general Riemannian manifolds, including the two-sphere $\mathbb{S}^2$. In general, such kernels can be approximated by the Mercer sum

$$k(s, s') \approx \frac{1}{C_\Phi} \sum_{j=0}^{J} \Phi(\lambda_j) \varphi_j(s) \varphi_j(s'), \quad s, s' \in \mathbb{S}^2, \tag{8}$$

for some $\Phi : \mathbb{R} \to \mathbb{R}$ and constant C_Φ determined from the kernel, and $\{\lambda_j\}_{j=0}^J$, $\{\varphi_j\}_{j=0}^J$ are the J top eigenvalues and eigenfunctions respectively of the negative Laplace-Beltrami operator; on $\mathbb{S}^2$, the latter is precisely the spherical harmonics. Furthermore, on $\mathbb{S}^2$, by making use of the addition theorem for spherical harmonics and the result (Azangulov et al., 2024, Proposition 7), we get an alternative expression for the kernel (see Appendix A.2 for the derivation)

$$k(s, s') \approx \mathbb{E}_{\omega, b}\left[c_\omega \, \mathcal{G}_\omega^{1/2}(d_{\mathbb{S}^2}(s, b)) \, \mathcal{G}_\omega^{1/2}(d_{\mathbb{S}^2}(s', b)) \right], \quad s, s' \in \mathbb{S}^2, \tag{9}$$

where $\mathcal{G}_n^\alpha(\cdot)$ are the Gegenbauer polynomials of order n and weight parameter α, $d_{\mathbb{S}^2}(\cdot, \cdot)$ denotes the geodesic distance on $\mathbb{S}^2$, c_ω is an appropriate scaling constant, and the expectation is taken over $b \sim U(\mathbb{S}^2)$, the uniform distribution over the sphere, and $\omega \sim \mathrm{Multinomial}(C_\Phi^{-1}\Phi(\lambda_1), \ldots, C_\Phi^{-1}\Phi(\lambda_J))$. Then, by considering Monte Carlo approximation of the expectation in equation 9, this implies random feature maps of the form

$$\phi_{\mathbb{S}^2}^m(s) = \sqrt{M^{-1} c_{\omega_m}} \, \mathcal{G}_{\omega_m}^{1/2}(d_{\mathbb{S}^2}(s, b_m)), \quad s \in \mathbb{S}^2, \quad m = 1, \ldots, M, \tag{10}$$

$$\text{where} \quad \omega_m \sim \mathrm{Multinomial}(C_\Phi^{-1}\Phi(\lambda_1), \ldots, C_\Phi^{-1}\Phi(\lambda_J)), \quad b_m \sim U(\mathbb{S}^2). \tag{11}$$

This gives us an analogous notion of random features (equation 4) on the sphere, which we can use as a component in our deep random feature model when our input is spherical.

Remark 1 *We may also consider the deterministic features $\phi^m(s) = \sqrt{C_\Phi^{-1}\Phi(\lambda_m)}\varphi_m(s)$, derived from equation 8, which is analogous to the regular Fourier features (Hensman et al., 2018; Solin & Särkkä, 2020) in the planar case. However in practice, we find that working with random features (equation 10) produce more stable results.*

3.1.3 SPATIOTEMPORAL MODELLING WITH DEEP RANDOM FEATURES

So far, we have only discussed how to process spatial inputs. In order to deal with the temporal components in our data, we first consider deep random features in the spatial domain $\boldsymbol{h}_x^{(L_x)} : \mathcal{X} \to \mathbb{R}^B$ ($\mathcal{X} = \mathbb{R}^I$ or $\mathbb{S}^2$), and temporal domain $\boldsymbol{h}_t^{(L_t)} : \mathbb{R} \to \mathbb{R}^B$ separately, before combining them as

$$\boldsymbol{f}(x, t) = \boldsymbol{\Theta}\big(\texttt{concat}[\boldsymbol{h}_x^{(L_x)}(x), \boldsymbol{h}_t^{(L_t)}(t)]\big), \tag{12}$$

where $\boldsymbol{\Theta} \in \mathbb{R}^{O \times 2B}$ are learnable weights initialised with i.i.d. standard Gaussians. At short timescales, geospatial fields are approximately stationary, hence we can use a single layer network to model the temporal component $\boldsymbol{h}_t^{(L_t)}$ (i.e., $L_t = 1$). To introduce more complex spatiotemporal dependence, we can replace the linear output layer in equation 12 with deep random features. However we find that in most applications this is unnecessary, only introducing extra cost.

3.1.4 SKIP CONNECTIONS

To prevent pathological behaviour from emerging as we increase the network depth, we add skip connections to the inputs (Duvenaud et al., 2014; Dunlop et al., 2018). In the planar case, this takes

$$\boldsymbol{h}^{(\ell+1)}(\boldsymbol{x}) = \boldsymbol{\Theta}^{\ell+1} \boldsymbol{\phi}^{\ell+1}\big(\texttt{concat}[\boldsymbol{h}^{(\ell)}(\boldsymbol{x}), \boldsymbol{x}]\big) \tag{13}$$

in the $(\ell + 1)$-th layer of the network, where $\boldsymbol{\phi}^{\ell+1} : \mathbb{R}^{B+I} \to \mathbb{R}^M$. In the spherical case, this is not straightforward as the outputs of each layer will be Euclidean while the input is spherical. To this end, we consider $\boldsymbol{\phi}^{\ell+1}$ to be additive random features corresponding to the sum kernel $\boldsymbol{k}((\boldsymbol{x}, s), (\boldsymbol{x}', s')) = \boldsymbol{k}_{\mathbb{R}^B}(\boldsymbol{x}, \boldsymbol{x}') + \boldsymbol{k}_{\mathbb{S}^2}(s, s')$, for $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^B$, $s, s' \in \mathbb{S}^2$, where $k_{\mathbb{R}^B}(\cdot, \cdot)$, $k_{\mathbb{S}^2}(\cdot, \cdot)$ are stationary kernels on their respective spaces (see Appendix B.1 for details).

3.2 UNCERTAINTY QUANTIFICATION

Uncertainty quantification (UQ) with deep random features is achieved using standard Bayesian deep learning techniques. In particular, we consider the following methods in our experiments:

Variational inference. This considers a Gaussian approximation to the posterior $p(\boldsymbol{\Theta}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{\Theta}|\boldsymbol{m}, \boldsymbol{C})$. Here, the moments of the variational distribution $q(\boldsymbol{\Theta}) := \mathcal{N}(\boldsymbol{\Theta}|\boldsymbol{m}, \boldsymbol{C})$ are learned by maximising the evidence lower bound (ELBO)

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{f}_{\boldsymbol{\Theta}}; \mathcal{D}) = \mathbb{E}_q \left[-\ell(\boldsymbol{f}_{\boldsymbol{\Theta}}(\boldsymbol{X}), \boldsymbol{y})\right] - \mathcal{KL}(q||p_{\boldsymbol{\Theta}}), \tag{14}$$

where $p_{\boldsymbol{\Theta}}(\boldsymbol{\Theta}) = \mathcal{N}(\boldsymbol{\Theta}|\boldsymbol{0}, \boldsymbol{I})$ is the prior on $\boldsymbol{\Theta}$ and $\mathcal{KL}(\cdot||\cdot)$ is the Kullback-Leibler divergence.

Dropout at test time. A simple heuristic for obtaining uncertainty estimates is to apply dropout not only at training time but also at test time. This yields an ensemble of random outputs, whose empirical distribution informs us of the model uncertainty. In fact, one can understand this as a form of variational inference, as shown in Gal & Ghahramani (2016).

Deep ensembles. Deep ensembles (Lakshminarayanan et al., 2017) obtain uncertainty estimates by training an ensemble of models, initialised from different random seeds. The ensemble of outputs is then used to estimate uncertainty, similar to the dropout method for UQ. While being a simple method, this has been shown to be surprisingly effective at obtaining uncertainty estimates. Moreover, provided the model is small enough (which is often the case for deep random features), the ensembles can be trained in parallel on a single GPU.

3.3 HYPERPARAMETER SELECTION

Our deep random features model contain several hyperparameters $\boldsymbol{\lambda}$, including those of the kernel (e.g. lengthscales) that we use to derive our random features. If variational inference is used for UQ, we can take the ELBO (equation 14) for model comparison, as it may be viewed as a surrogate for the log model evidence $\log p(\mathcal{D}|\boldsymbol{\lambda})$, being its lower bound. When using ensemble based methods, we rely on performance on a held-out validation set $\mathcal{D}^* = \{(\boldsymbol{x}_n^*, \boldsymbol{y}_n^*)\}_{n=1}^{N^*}$ to select our hyperparameters. In particular, we consider the following validation loss to select $\boldsymbol{\lambda}$

$$\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) = \frac{1}{N^*} \sum_{n=1}^{N^*} \frac{1}{J} \sum_{j=1}^{J} \ell(\boldsymbol{f}_j(\boldsymbol{x}_n^*; \mathcal{D}, \boldsymbol{\lambda}); \boldsymbol{y}_n^*), \tag{15}$$

where $\{\boldsymbol{f}_j\}_{j=1}^J$ are the ensembles. Note that this can be viewed as approximating the negative log-predictive density (see Remark 3, Appendix C.1). We minimise this loss using Bayesian optimisation. In practice, we find that learning $\boldsymbol{\lambda}$ from equation 15 alone may still lead to overfitting models. Therefore to prevent this, we may opt to add an extra *functional regularisation* term

$$\mathcal{L}_{\text{val+reg}}(\boldsymbol{\lambda}) = (1-\alpha)\mathcal{L}_{\text{val}}(\boldsymbol{\lambda}) + \alpha \|\nabla \bar{\boldsymbol{f}}(\,\cdot\,; \mathcal{D}, \boldsymbol{\lambda})\|_{L^2}^2, \quad \bar{\boldsymbol{f}} := \frac{1}{J} \sum_{j=1}^{J} \boldsymbol{f}_j \tag{16}$$

for $\alpha \in [0, 1)$, which helps to penalise those $\boldsymbol{\lambda}$ that give rise to functions $\boldsymbol{f}$ with sharp gradients (see Remark 2 in Appendix B.2 on why we do not consider hyperpriors for regularisation). Here, $\|\cdot\|_{L^2}^2$ denotes the appropriate L^2 norm depending on the input space and ∇ the gradient. For spherical inputs, we refer the readers to Appendix B.2 for more details.

4 EXPERIMENTS

We evaluate the spatiotemporal deep random features (DRF) model on various remote sensing datasets and compare against various baselines to assess its ability to make predictions and quantify uncertainty. In our first experiment, we consider interpolation of synthetic data, and evaluate our model's ability to recover the ground truth. In our second and third experiments, we consider interpolation of real satellite data at local and global scales to test the robustness of our method. Details can be found in Appendix C. All experiments are performed using the NVIDIA L4 GPU.

4.1 BASELINE MODELS

Throughout this section, we consider several models as baselines to compare our model against. We consider both GP-based baselines and DNN-based baselines. In the former category, we consider the sparse variational GP (SVGP) model in Hensman et al. (2013), deep GPs (DGP) using doubly stochastic variational inference (Salimbeni & Deisenroth, 2017) and a mixture model of local GPs using the `GPSat` library (Gregory et al., 2024b). In the latter caregory, we consider deep ensembles of multilayer perceptrons (MLP) with ReLU activations, MLP with sinusoidal activations (SIREN, Sitzmann et al. (2020)), and conditional neural processes Garnelo et al. (2018).

4.2 EVALUATION ON SYNTHETIC DATA

The purpose of our first experiment is to use synthetic observations from a ground truth field to evaluate our model's ability to reconstruct the field. We use mean sea surface height (MSS) in the arctic as our ground truth, synthesised from 12 years of altimetry readings of Sentinel-3A, 3B (S3A, 3B) and CryoSat-2 (CS2) satellites. We then generate artificial measurements along S3A, 3B and CS2 tracks between the dates March 1st–10th 2020, taking the MSS values along the tracks and adding i.i.d. Gaussian noise to mimic measurement noise. Our final dataset comprise 1,158,505 datapoints; we select 80% of these randomly for training and the remaining 20% for validation. We train all models using the mean-squared error loss (for GP baselines, this corresponds to a Gaussian likelihood) with fixed weight decay parameter matching the observation noise variance. Visual comparison of predictions from all models can be found in Appendix C.5.1.

4.2.1 EFFECT OF DEPTH

In Figure 3, we show the effect of depth on our model's ability to reconstruct the true MSS field (in terms of RMSE) and corresponding computation time of the entire wokflow, including the time to tune the kernel hyperparameters (see Appendix C.2.3). Generally, we find that deep networks outperform the shallow network on the RMSE, with the four layer model performing the best on this example. With > 4 layers, we start to see some overfitting, which explains the higher RMSE for the 10 and 20 layer models. In Figure 4, we display mean results for models with two and four layers. We see that the deeper network is able to capture higher frequency details, resulting in the improved RMSE. The time it takes to train and tune models with 1-4 layers are not significantly different.

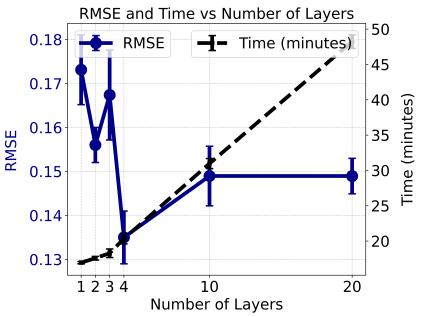

Figure 3: Comparison of RMSE and computation time vs. number of layers.

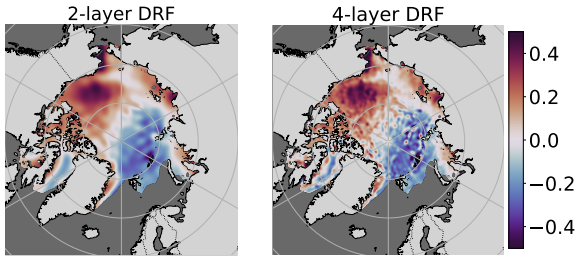

Figure 4: Comparison of predictions from DRF with two layers (left) and four layers (right). The four layer model is able to capture finer details compared to the two layer model.

4.2.2 UQ COMPARISONS

Next, we compare various UQ methods applied to a DRF model with four layers. In Table 1, we display comparisons with respect to the root mean squared error (RMSE), the negative log-likelihood (NLL), and the continuous ranked probability score (CRPS) (see Appendix C.1 for details on our evaluation method). The latter two evaluate the quality of uncertainties produced. Overall, we find that deep ensembles produce the best result in the CRPS and the RMSE, whereas variational inference (VI) yielded the best NLL. The lower NLL using VI may be due to the fact that its predictions are typically underconfident (see Figure 8, Appendix C.5.1) and NLL penalises them more lightly

Model	CRPS	NLL	RMSE	Time (minutes)
DRF (Ensembles)	$\mathbf{0.046 \pm 0.005}$	13.590 ± 4.899	0.135 ± 0.006	19.7 ± 0.4
DRF (VI)	0.071 ± 0.019	$\mathbf{-0.407 \pm 0.756}$	0.166 ± 0.021	6.40 ± 0.06
DRF (Dropout)	0.174 ± 0.001	425.987 ± 208.969	0.238 ± 0.001	48.6 ± 2.5
SVGP	0.230 ± 0.001	320.811 ± 52.960	0.155 ± 0.002	14.6 ± 0.0
DGP	0.058 ± 0.001	1614.069 ± 328.517	0.135 ± 0.002	42.3 ± 0.03
GPSat	$\mathbf{0.045 \pm 0.007}$	74.738 ± 15.622	$\mathbf{0.126 \pm 0.001}$	63.6 ± 0.2
ReLU MLP	0.062 ± 0.000	30.504 ± 7.877	0.146 ± 0.000	10.07 ± 0.01
SIREN	0.066 ± 0.000	13.974 ± 0.393	0.155 ± 0.000	2.55 ± 0.002
CNP	0.238 ± 0.070	2.525 ± 0.459	0.202 ± 0.010	21.0 ± 0.1

Table 1: Comparison of the CRPS, NLL and RMSE scores for a four-layer DRF (with different UQ methods) against various baselines on the synthetic experiment. Best performing model in **bold** and second best performing in blue. We display the mean and standard deviation over five experiments.

than overconfident ones. However, the results in Figure 8 suggest that the results from deep ensembles are better calibrated to the observations, which explains the lower CRPS. Dropout does not perform well in neither the mean prediction nor uncertainty estimation.

4.2.3 Baseline comparisons

In Table 1, we also display results for the other baseline models described in Section 4.1. Regarding computation times, for DRF, we include the time to tune the kernel hyperparameters using Bayesian optimisation (Section 3.3) to make a fair comparison with the GP-based baselines, where the total time for training and inference are recorded. However, we assume other hyperparameters, such as number of layers and hidden units to be fixed ($L = 4$, $B = 128$, $H = 1000$). For the other DNN-based baselines, we assume the architecture is tuned ahead of time and fixed.

Comparing with the GP baselines, we find that DGP and the `GPSat` model to be closest competitors to the DRF deep ensembles, with `GPSat` surpassing its performance on the RMSE. However, the time taken to train the DGP and `GPSat` model are two to three times longer than the time taken to train and tune the ensemble DRF. For example, `GPSat` trains 1225 local GP models on this example, which makes computation heavy. DGP has overall low predictive variances (see Figure 9, Appendix C.5.1), which results in high NLL values. In contrast, the uncertainty estimates of DRF and `GPSat` are well-calibrated to the satellite tracks. Qualitatively, all three models recover the ground truth well, with `GPSat` and DRF reconstructing it almost perfectly.

Now, comparing to other DNN baselines, SIREN's performance is noteworthy, being similar to DRF in that it both uses trigonometric activations and differing only in the way the weights are initialised and whether it has bottleneck layers with fixed preactivations. The DRF ensemble is better able to capture the spatiotemporal patterns of the field, reinforcing the importance of the subtle architectural differences. The ReLU network and CNP both lead to oversmoothing results, similar to SVGP.

4.3 Freeboard estimation from satellite altimetry data

Here, we consider the interpolation of real altimeter readings of sea-ice freeboard taken along the Sentinel-3A, 3B and CryoSat-2 satellites (Gregory et al., 2024a). Real satellite altimetry measurements are typically noisy with heavy-tailed statistics (see Figure 7, Appendix C.3), hence they present a more challenging setting than our previous synthetic experiment. Our goal here is to test the robustness of our approach in comparison to other methods in this more realistic setting. Experimental details can be found in Appendix C.3 and visual comparison of all results can be found in Figure 10, Appendix C.5.2.

We compare a two-layer DRF against the same baselines as before and display the root mean absolute error (RMAE), CRPS and negative log-predictive density (NLPD) on a separately held out test data comprising 15% of the entire data in Table 2. We use the RMAE instead of the RMSE here as it is more robust to the heavy-tailed statistics of the measurement error, and therefore provides a more reliable performance metric in this setting. The hyperparameter search for DRF was performed

with functional regularisation (see Section 3.3) as we found that without it, optimising on only the validation loss lead to overfitting models (see Figure 5). Here, we used a penalty weight of $\alpha = 0.9$.

We find that out of our GP-based baselines, SVGP and the `GPSat` model give comparable performance to DRF ensembles, with `GPSat` giving the best performance quantitatively. However, when we examine the outputs from all three models (Figure 5), we find that the results from `GPSat` contain spurious checkerboard-like patterns resulting from unstable hyperparameter optimisation at several local expert locations. This issue occurs since the `GPSat` local experts only see data in local regions, making them more sensitive to the heavy noise present in the data. On the other hand, DRF sees data globally, which helps them to identify the larger structures in the data, while simultaneously capturing the finer details owing to their deep architecture. We find that qualitatively, SVGP performs well on this example, due to the larger prominence of low frequency features in the underlying field that extend across the basin. We see that DRF provides a middle ground between the two, being neither "too local" as we see in `GPSat`, nor "too global", demonstrating its ability to adapt to the characteristics of the field via its ability to tune per-layer lengthscales during hyperparameter optimisation. The other DNN baselines do not have this capability and therefore perform poorly, either severely overfitting (SIREN) or underfitting (ReLU and CNP) on the noisy data.

Model	CRPS	NLPD	RMAE
DRF (Ensembles)	$\mathbf{0.077 \pm 0.000}$	-0.944 ± 0.020	0.322 ± 0.000
SVGP	0.079 ± 0.001	$\mathbf{-1.30 \pm 0.004}$	0.322 ± 0.001
DGP	0.208 ± 0.000	-0.159 ± 0.000	0.339 ± 0.001
GPSat	$\mathbf{0.076 \pm 0.001}$	-1.167 ± 0.025	$\mathbf{0.318 \pm 0.000}$
ReLU MLP	0.714 ± 0.936	0.076 ± 1.578	0.751 ± 0.554
SIREN	0.088 ± 0.000	-1.109 ± 0.01	0.345 ± 0.001
CNP	0.101 ± 0.001	-0.898 ± 0.03	0.328 ± 0.002

Table 2: Comparison of the CRPS, NLPD and RMAE scores for a two-layer DRF against various baselines on sea-ice freeboard interpolation from S3A, 3B and CS2 satellite altimetry readings. Best performing model in **bold** and second best performing in blue.

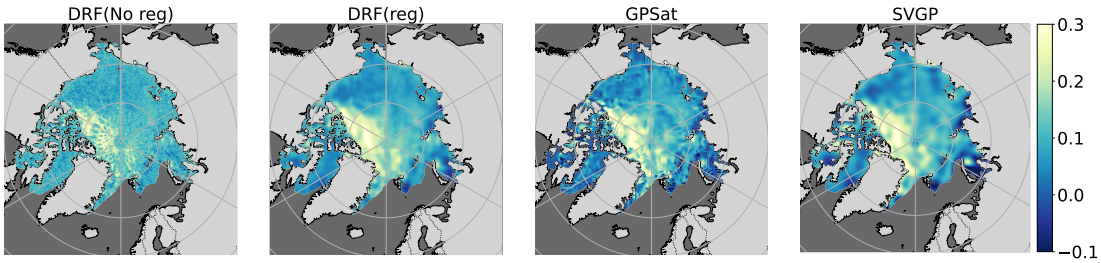

Figure 5: Mean results of DRF, GPSat and SVGP on freeboard interpolation. For DRF, we plot results obtained both with and without functional regularisation during hyperparameter search.

4.4 Large scale interpolation of global sea level anomaly

In this final experiment, we investigate the potential of DRF to interpolate *global* fields using spherical random features (Section 3.1.2) in the spatial inputs. For this experiment, we use real data of sea level anomaly measurements collected from the Sentinel 3A, 3B satellites (Copernicus Data Space Ecosystem, 2024). By considering four days of measurements, our final data consists of 8,094,569 datapoints. We use 80% for training, and 20% for validation. Similar to our previous data obtained from real satellite measurements, this data contains many outliers, making it a challenge to interpolate the data robustly, let alone whilst being consistent with the geometry of the sphere.

Our goal is to fit a spatiotemporal field $f : \mathbb{S}^2 \times \mathbb{R} \to \mathbb{R}$. To this end, we consider a DRF model whose first layer in the spatial component is given by the spherical random feature $\boldsymbol{\phi}_{\mathbb{S}^2} : \mathbb{S}^2 \to \mathbb{R}^H$ (equation 10). The subsequent layers are given by Euclidean random feautures. To train our model, we opted to use the Huber loss instead of MSE, which gave rise to slightly more robust results, likely

due to the large number of outliers. We also used functional regularisation with a penalty weight of $\alpha = 0.95$ when tuning hyperparameters.

In Figure 6, we compare the mean predictions of the spherical DRF model with predictions from (1) SVGP using the spherical Matérn kernel of Borovitskiy et al. (2020), and (2) the Euclidean DRF model, taking the longitude and latitude coordinates of the satellite tracks as spatial inputs in $\mathbb{R}^2$. We use the spherical Matérn kernel implementation in the `geometric-kernels` package Mostowsky et al. (2024) to model the spatial component of our SVGP baseline. The temporal component is included by modelling the GP with a product kernel $k((x,t),(x',t')) = k_{\mathbb{S}^2}(x,x')k_{\mathbb{R}}(t,t')$. Comparing the SVGP output with DRF, we see that they are both able to capture the larger patterns in the data. However, SVGP fails to capture some of the finer fluctuations, for instance those around the Antarctic circumpolar current, known for its intense ocean activities.

For the Euclidean DRF, while it admits a deep structure, we find that it is not flexible enough to adapt to the spherical geometry of the input space. For example, there are spurious distortions around the poles cause by the stereographic projection, in addition to a discontinuity at longitude $= 0°$ (see Figure 11 in Appendix C.5.3). Perhaps more interestingly, the Euclidean DRF is not able to learn the fine scale fluctuations that the spherical DRF is able to pick up, only being able to learn large scale trends in the data, similar to SVGP. This highlights the importance of explicitly incorporating the spherical inductive bias into the model when modelling global fields.

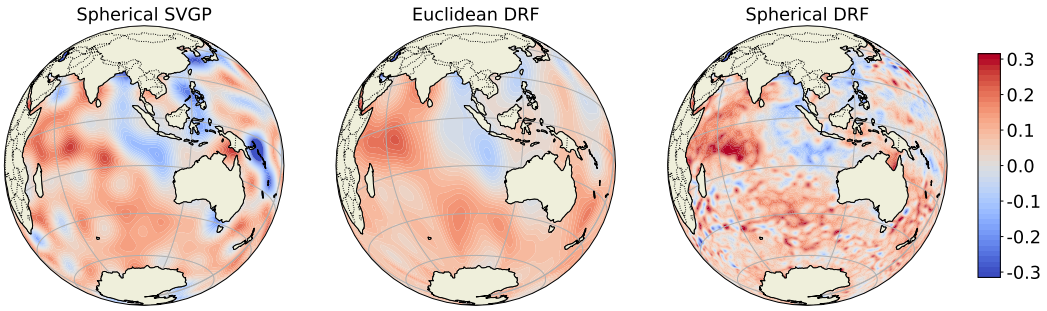

Figure 6: Mean results for global sea level anomaly interpolation. From left to right: SVGP using the spherical Matérn kernel, Euclidean DRF, and Spherical DRF. Spherical DRF is able to learn more intricate details compared to the other two baselines.

5 Conclusion

In this paper, we propose to model spatiotemporal fields using deep neural networks, whose layers are derived from random feature expansions of stationary kernels. This neural representation can be trained on observations to effectively fill in the gaps between remote sensing observations of the earth's surface. By using random features as neural network layers, we retain some of the inductive biases of Gaussian processes that helps to generalise across unobserved areas, while the deep architecture allows them to learn complex patterns in the data. We derive models both on the plane and the sphere to reconstruct local and global fields. Our experiments on various remote sensing data demonstrate that our deep ensemble model is able to flexibly adapt to the data, being able to learn both low and high-frequency structures that exist in the underlying field. A current limitation of our approach is the difficulty of tuning kernel hyperparameters; we use Bayesian optimisation (BO) on the validation loss to achieve this, which require knowledge of the ranges each hyperparameter may take. This is not clear due to the deep architecture, making the hyperparameters less interpretable than the shallow case. Using ranges that are too narrow may lead to model misspecification and when too wide, it will lead to longer BO iterations. Additionally, we observe that it is sometimes necessary to add functional regularisation to reduce BO variance, necessitating hand tuning of the penalty weight α, a hyper-hyperparameter. Despite this, our promising results suggest the potential for deep learning methods to pave the way for more accurate and flexible reconstructions of spatiotemporal fields from remote sensing data.

REFERENCES

Devanshu Agrawal, Theodore Papamarkou, and Jacob Hinkle. Wide neural networks with bottlenecks are deep Gaussian processes. *Journal of Machine Learning Research*, 2020.

Pooja Algikar and Lamine Mili. Robust Gaussian process regression with Huber likelihood. *arXiv preprint arXiv:2301.07858*, 2023.

Matias Altamirano, Francois-Xavier Briol, and Jeremias Knoblauch. Robust and conjugate Gaussian process regression. In *International Conference on Machine Learning*. PMLR, 2024.

Iskander Azangulov, Andrei Smolensky, Alexander Terenin, and Viacheslav Borovitskiy. Stationary kernels and Gaussian processes on Lie groups and their homogeneous spaces I: the compact case. *Journal of Machine Learning Research*, 2024.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Pier Giovanni Bissiri, Chris C Holmes, and Stephen G Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, et al. Matérn Gaussian processes on Riemannian manifolds. In *Advances in Neural Information Processing Systems*, 2020.

David Burt, Carl Edward Rasmussen, and Mark Van Der Wilk. Rates of convergence for sparse variational Gaussian process regression. In *International Conference on Machine Learning*. PMLR, 2019.

Copernicus Data Space Ecosystem. Sentinel-3 sea level anomaly data, 2024. URL `https://dataspace.copernicus.eu`.

Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *International Conference on Machine Learning*. PMLR, 2017.

Andreas Damianou and Neil D Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*. PMLR, 2013.

Matthew M Dunlop, Mark A Girolami, Andrew M Stuart, and Aretha L Teckentrup. How deep are deep Gaussian processes? *Journal of Machine Learning Research*, 2018.

Vincent Dutordoir, James Hensman, Mark van der Wilk, Carl Henrik Ek, Zoubin Ghahramani, and Nicolas Durrande. Deep neural networks as point estimates for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2021.

David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*. PMLR, 2014.

David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. Additive Gaussian processes. In *Advances in Neural Information Processing Systems*, 2011.

Geir Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 2003. doi: https://doi.org/10.1007/s10236-003-0036-9.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*. PMLR, 2016.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*. PMLR, 2018.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007. doi: https://doi.org/10.1198/016214506000001437.

William Gregory, Isobel R Lawrence, and Michel Tsamados. A Bayesian approach towards daily pan-Arctic sea ice freeboard estimates from combined CryoSat-2 and Sentinel-3 satellite observations. *The Cryosphere*, 2021. doi: https://doi.org/10.5194/tc-15-2857-2021.

William Gregory, Ronald MacEachern, So Takao, Isobel R Lawrence, Carmen Nab, Marc Peter Deisenroth, and Michel Tsamados. Datasets for "Scalable interpolation of satellite altimetry data with probabilistic machine learning", 2024a. URL https://zenodo.org/doi/10.5281/zenodo.13218448.

William Gregory, Ronald MacEachern, So Takao, Isobel R Lawrence, Carmen Nab, Marc Peter Deisenroth, and Michel Tsamados. Scalable interpolation of satellite altimetry data with probabilistic machine learning. *Nature Communications*, 2024b. doi: https://doi.org/10.1038/s41467-024-51900-x.

James Hensman, Nicolo Fusi, and Neil D Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, 2013.

James Hensman, Nicolas Durrande, and Arno Solin. Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, 2018.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, 2017.

PY Le Traon, F Nadal, and N Ducet. An improved mapping method of multisatellite altimeter data. *Journal of Atmospheric and Oceanic Technology*, 1998. doi: https://doi.org/10.1175/1520-0426(1998)015%3C0522:AIMMOM%3E2.0.CO;2.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, 2019.

Nils Lehmann. Lightning UQ Box, 2024. URL https://lightning-uq-box.readthedocs.io/en/latest/.

Chi-Ken Lu and Patrick Shafto. On connecting deep trigonometric networks with deep Gaussian processes: Covariance, Expressivity, and Neural Tangent Kernel. *arXiv preprint arXiv:2203.07411*, 2022. doi: https://doi.org/10.48550/arXiv.2203.07411.

Lassi Meronen, Christabella Irwanto, and Arno Solin. Stationary activations for uncertainty calibration in deep learning. In *Advances in Neural Information Processing Systems*, 2020.

Lassi Meronen, Martin Trapp, and Arno Solin. Periodic activation functions induce stationarity. In *Advances in Neural Information Processing Systems*, 2021.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. doi: https://doi.org/10.1145/3503250.

Peter Mostowsky, Vincent Dutordoir, Iskander Azangulov, Noémie Jaquier, Michael John Hutchinson, Aditya Ravuri, Leonel Rozo, Alexander Terenin, and Viacheslav Borovitskiy. The GeometricKernels package: Heat and Matérn kernels for geometric learning on manifolds, meshes, and graphs. *arXiv:2407.08086*, 2024.

Theodore Papamarkou, Maria Skoularidou, Konstantina Palla, Laurence Aitchison, Julyan Arbel, David Dunson, Maurizio Filippone, Vincent Fortuin, Philipp Hennig, José Miguel Hernández-Lobato, et al. Position: Bayesian deep learning is needed in the age of large-scale AI. In *International Conference on Machine Learning*, 2024.

Geoff Pleiss and John P Cunningham. The limitations of large width in neural networks: A deep Gaussian process perspective. In *Advances in Neural Information Processing Systems*, 2021.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, 2017.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, 2020.

Henriette Skourup, Sinéad Louise Farrell, Stefan Hendricks, Robert Ricker, Thomas WK Armitage, Andy Ridout, Ole Baltazar Andersen, Christian Haas, and Steven Baker. An assessment of state-of-the-art mean sea surface and geoid models of the Arctic ocean: Implications for sea ice freeboard retrieval. *Journal of Geophysical Research: Oceans*, 2017. doi: https://doi.org/10.1002/2017JC013176.

Arno Solin and Simo Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *Statistics and Computing*, 2020. doi: https://doi.org/10.1007/s11222-019-09886-w.

Shengyang Sun, Jiaxin Shi, and Roger Baker Grosse. Neural networks as inter-domain inducing points. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems*, 2020.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*. PMLR, 2009.

Christopher K Wikle, Andrew Zammit-Mangion, and Noel Cressie. *Spatio-temporal statistics with R*. Chapman and Hall/CRC, 2019.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.