Environmental Data Science (2023), xx: 1–37
doi:10.1017/xxxx

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Environmental Sensor Placement with Convolutional Gaussian Neural Processes

Tom R. Andersson[1*], Wessel P. Bruinsma[2], Stratis Markou[3], James Requeima[3,4], Alejandro Coca-Castro[5], Anna Vaughan[3], Anna-Louise Ellis[6], Matthew A. Lazzara[7,8], Daniel C. Jones[†1], J. Scott Hosking[†1,5] and Richard E. Turner[†2,3]

[1]British Antarctic Survey, NERC, UKRI
[2]Microsoft Research AI4Science
[3]University of Cambridge
[4]Invenia Labs
[5]The Alan Turing Institute
[6]Met Office
[7]University of Wisconsin-Madison
[8]Madison Area Technical College
[†] Joint senior authors
*Corresponding author. Email: tomand@bas.ac.uk

Received: 1 February 2023

Keywords: sensor placement, neural processes, active learning, meta-learning

Abstract

Environmental sensors are crucial for monitoring weather conditions and the impacts of climate change. However, it is challenging to maximise measurement informativeness and place sensors efficiently, particularly in remote regions like Antarctica. Probabilistic machine learning models can evaluate placement informativeness by predicting the uncertainty reduction provided by a new sensor. Gaussian process (GP) models are widely used for this purpose, but they struggle with capturing complex non-stationary behaviour and scaling to large datasets. This paper proposes using a convolutional Gaussian neural process (ConvGNP) to address these issues. A ConvGNP uses neural networks to parameterise a joint Gaussian distribution at arbitrary target locations, enabling flexibility and scalability. Using simulated surface air temperature anomaly over Antarctica as ground truth, the ConvGNP learns spatial and seasonal non-stationarities, outperforming a non-stationary GP baseline. In a simulated sensor placement experiment, the ConvGNP better predicts the performance boost obtained from new observations than GP baselines, leading to more informative sensor placements. We contrast our approach with physics-based sensor placement methods and propose future work towards an operational sensor placement recommendation system. This system could help to realise environmental digital twins that actively direct measurement sampling to improve the digital representation of reality.

Impact Statement

This paper addresses the challenge of identifying intelligent sensor placements for monitoring environmental phenomena, using Antarctic air temperature anomaly as an example. The authors propose using a recent machine learning model—a convolutional Gaussian neural process (ConvGNP)—which can capture complex non-stationary behaviour and scale to large datasets. The ConvGNP outperforms previous data-driven approaches in simulated experiments, finding more informative and cost-effective sensor placements. This could lead to improved decision-making for monitoring weather conditions and climate change impacts.

1. Introduction

Selecting optimal locations for placing environmental sensors is an important scientific challenge. For example, improved environmental monitoring can lead to more accurate weather forecasting (Weissmann et al., 2011; Jung et al., 2016). Further, better observation coverage can improve the representation of extreme events, climate variability, and long-term trends in reanalysis models (Bromwich and Fogt, 2004) and aid their validation (Bracegirdle and Marshall, 2012). This is particularly important in remote regions like Antarctica, where observations are sparse (Jung et al., 2016) and the cost of deploying weather stations is high (Lazzara et al., 2012), motivating an objective model-based approach that provides an accurate notion of the informativeness of new observation locations. This informativeness can then guide decision-making so that scientific goals are achieved with as few sensors as possible.

The above sensor placement problem has been studied extensively from a physics-based numerical modelling perspective (Majumdar, 2016). Multiple approaches exist for estimating the value of current or new observation locations for a physical model. Examples include observing system simulation experiments (Hoffman and Atlas 2016), adjoint methods (Langland and Baker, 2004), and ensemble sensitivity analysis (ESA; Torn and Hakim, 2008). Using a numerical model for sensor placement comes with benefits and limitations. One drawback is that physical models can be biased, and this can degrade sensor placements. This suggests that physics-based approaches could be supplemented by data-driven methods that learn statistical relationships directly from the data.

Machine learning (ML) methods also have a long history of use for experimental design and sensor placement (MacKay, 1992; Cohn, 1993; Seo et al., 2000; Krause et al., 2008). First, a *probabilistic model* is fit to noisy observations of an unknown function $f(\boldsymbol{x})$. Then, *active learning* is used to identify new $\boldsymbol{x}$-locations that are expected to maximally reduce the model's uncertainty about some aspect of $f(\boldsymbol{x})$. The Gaussian process (GP; Rasmussen 2004) has so far been the go-to class of probabilistic model for sensor placement and the related task of Bayesian optimisation[1] (Singh et al., 2007; Krause et al., 2008; Marchant and Ramos, 2012; Shahriari et al., 2016). Setting up a GP requires the user to specify a mean function (describing the expected value of the function) and a covariance function (describing how similar the $f(\boldsymbol{x})$ values are at different $\boldsymbol{x}$-locations). Once a GP has been initialised, conditioning it on observed data and evaluating at target locations produces a multivariate Gaussian distribution, which can be queried to search for informative sensor placements.

GPs have several compelling strengths which make them particularly amenable to small data regimes and simple target functions. However, modelling a climate variable with a GP is challenging due to spatiotemporal non-stationarity and large volumes of data corresponding to multiple predictor variables. While non-stationary GP covariance functions are available (and improve sensor placement in Krause et al. 2008 and Singh et al. 2010), this still comes with the task of choosing the right functional form and introduces a risk of overfitting (Fortuin et al., 2020). Further, conditioning GPs on supplementary predictor variables (such as satellite data) is non-trivial and their computational cost scales cubically with dataset size, which becomes prohibitive with large environmental datasets. Approximations allow GPs to scale to large data (Titsias, 2009; Hensman et al., 2013), but these also harm prediction quality. The above model misspecifications can lead to uninformative or degraded sensor placements, motivating a new approach which can more faithfully capture the behaviour of complex environmental data.

Convolutional neural processes (ConvNPs) are a recent class of ML models that have shown promise in modelling environmental variables. For example, ConvNPs can outperform a large ensemble of climate downscaling approaches (Vaughan et al., 2021; Markou et al., 2022) and integrate data of gridded and point-based modalities (Bruinsma et al., 2023). One variant, the convolutional Gaussian neural process (ConvGNP; Markou et al., 2022; Bruinsma et al., 2021), uses neural networks to parameterise a

[1] Bayesian optimisation differs slightly from sensor placement in that the task is to find the maximum (of minimum) a black-box function f rather than reduce overall uncertainty about f.

joint Gaussian distribution at target locations, allowing them to scale linearly with dataset size while learning mean and covariance functions directly from the data.

In this paper, simulated atmospheric data is used to assess the ability of the ConvGNP to model a complex environmental variable and find informative sensor placements. The paper is laid out as follows. Section 2 introduces the data and describes the ConvGNP model. Section 3 compares the ConvGNP with GP baselines with three experiments: predicting unseen data, predicting the benefit of new observations, and a sensor placement toy experiment. It is then shown how placement informativeness can be traded-off with cost using multi-objective optimisation to enable a human-in-the-loop decision-support tool. Section 4 discusses limitations and possible extensions to our approach, contrasting ML-based and physics-based sensor placements. Concluding remarks are provided in Section 5.

2. Methods

In this section we define the goal and data, formalise the problem tackled, and introduce the ConvGNP.

2.1. Goal and source data

The simulated target variable we use to analyse sensor placement abilities is $25\,\mathrm{km}$-resolution ERA5 daily-averaged $2\,\mathrm{m}$ temperature anomaly over Antarctica (Figure 2a; Hersbach et al. 2020). Reanalysis data are produced by fitting a numerical climate model to observations using data assimilation (Gettelman et al., 2022), capturing the complex dynamics of the Earth system on a regular grid. This yields a strong testbed for assessing sensor placement abilities. We train a ConvGNP and a set of GP baselines to produce probabilistic predictions for ERA5 temperature anomalies, assessing performance on a range of metrics. We then perform simulated sensor placement experiments to quantitatively compare the ConvGNP's estimates of observation informativeness with that of the GP baselines and simple heuristic placement methods. The locations of 79 Antarctic temperature stations that existed on February 15th, 2009, are used as the starting point for the sensor placement experiment (black crosses in left-most panel of Figure 1), simulating a realistic sensor network design scenario.

Alongside inputs of ERA5 temperature anomaly observations, we also provide the ConvGNP with a second data stream on a $25\,\mathrm{km}$ grid, containing surface elevation and a land mask (obtained from the BedMachine dataset; Morlighem 2020), as well as space/time coordinate variables. For further details on the data sources and processing stages see Appendix A.

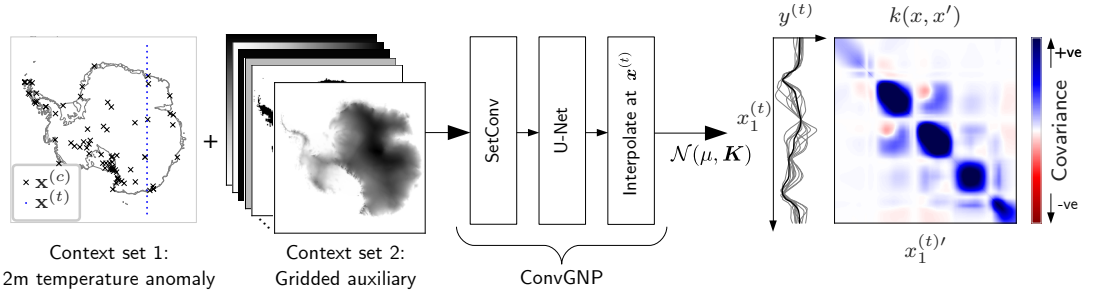

Figure 1. The ConvGNP data pipeline. *We have two context sets: ERA5 temperature anomaly observations and 6 gridded auxiliary fields, and we wish to make probabilistic predictions for temperature anomaly over a vertical line of target points (blue dotted line in left-most panel). In the ConvGNP, a SetConv layer fuses the context sets into a single gridded encoding (Figure B1, Gordon et al. 2020). A U-Net (Ronneberger et al., 2015) takes this encoded tensor as input and outputs a gridded representation, which is interpolated at target points $\boldsymbol{X}^{(t)}$ and used to parameterise the mean and covariance of a multivariate Gaussian distribution over $\boldsymbol{y}^{(t)}$. The output mean vector μ is shown as a black line, with 10 Gaussian samples overlaid in grey. The heatmap of the covariance matrix $\boldsymbol{K}$ shows the magnitude of spatial covariances, with covariance decreasing close to temperature anomaly context points.*

2.2. Formal problem set-up

We now formalise the problem set-up tackled in this study. First, we make some simplifying assumptions about the data to be modelled. We assume that data from different time steps, τ, are independent, and so we will not model temporal dependencies in the data. Further, we only consider variables that live in a 2D input space, as opposed to variables with a third input spatial dimension (e.g. altitude or depth). This simplifies the 3D or 4D modelling problem into a 2D one. Models built with these assumptions can learn correlations across 2D space, but not across time and/or height, which could be important in forecasting or oceanographic applications.

At each τ, there will be particular target locations $\boldsymbol{X}_\tau^{(\mathrm{t})} \in \mathbb{R}^{N_t \times 2}$ where we wish to predict an environmental variable $\boldsymbol{y}_\tau^{(\mathrm{t})} \in \mathbb{R}^{N_t}$ (we assume that the target variable is a 1D scalar for simplicity, but this need not be the case). Our target may be surface temperature anomaly along a line of points over Antarctica (blue dotted line in left-most panel of Figure 1). We call this a *target set* $T_\tau = (\boldsymbol{X}_\tau^{(\mathrm{t})}, \boldsymbol{y}_\tau^{(\mathrm{t})})$. The target set predictions will be made using several data streams, containing N-D observations $\boldsymbol{Y}_\tau^{(\mathrm{c})} \in \mathbb{R}^{N_c \times N}$ at particular locations $\boldsymbol{X}_\tau^{(\mathrm{c})} \in \mathbb{R}^{N_c \times 2}$. We call these data streams *context sets* $(\boldsymbol{X}_\tau^{(\mathrm{c})}, \boldsymbol{Y}_\tau^{(\mathrm{c})})$, and write the collection of all N_C context sets as $C_\tau = \{(\boldsymbol{X}_\tau^{(\mathrm{c})}, \boldsymbol{Y}_\tau^{(\mathrm{c})})_i\}_{i=1}^{N_C}$. Context sets may lie on scattered, off-grid locations (e.g. temperature anomaly observations at black crosses in left-most panel in Figure 1) or on a regular grid (e.g. elevation and other auxiliary fields in the second panel of Figure 1). We call the collection of context sets and the target set a *task* $\mathcal{D}_\tau = (C_\tau, T_\tau)$. The goal is to build a ML model that takes the context sets as input and maps to probabilistic predictions for the target values $\boldsymbol{y}_\tau^{(\mathrm{t})}$ given the target locations $\boldsymbol{X}_\tau^{(\mathrm{t})}$. Following Foong et al. 2020, we refer to this model as a *prediction map*, π. Once π is set up, a sensor placement algorithm $\mathcal{S}$ will use π to propose K new placement locations $\boldsymbol{X}^* \in \mathbb{R}^{K \times 2}$ based on query locations $\boldsymbol{X}^{(\mathrm{s})} \in \mathbb{R}^{S \times 2}$ and a set of tasks $\{\mathcal{D}_{\tau_j}\}_{j=1}^{J}$. Section 3.2 provides details on how we implement $\mathcal{S}$ in practice.

Physics-based numerical models could be framed as hard-coded prediction maps, ingesting context sets through data assimilation schemes and using physical laws to predict targets on a regular grid over space and time. These model outputs are deterministic by default, but applying stochastic perturbations to initial conditions or model parameters induces an intractable distribution over model outputs, $p(\boldsymbol{y}^{(\mathrm{t})})$, which can be sampled from to generate an ensemble of reanalyses or forecasts. However, current numerical models do not learn directly from data. In contrast, ML-based prediction maps will be trained from scratch to directly output a distribution over targets based on the context data.

2.3. ConvGNP model

Most ML methods are ill-suited to the problem described in Section 2.2. Typical deep learning approaches used in environmental applications, such as convolutional neural networks, require the data to lie on a regular grid, and thus cannot handle non-gridded data (e.g. Andersson et al. 2021; Ravuri et al. 2021). Recent emerging architectures such as transformers can handle off-the-grid data in principle, but in practice have used gridded data in environmental applications (e.g. Bi et al. 2022). Moreover, they also need architectural changes to make predictions at previously unseen input locations. On the other hand, Bayesian probabilistic models based on stochastic processes (such as GPs), can ingest data at arbitrary locations, but it is difficult to integrate more than one input data stream, especially when those streams are high dimensional (e.g. supplementary satellite data which aids the prediction task). Neural processes (NPs; Garnelo et al., 2018a,b) are prediction maps that address these problems by combining the modelling flexibility and scalability of neural networks with the uncertainty quantification benefits of GPs. The ConvGNP is a particular prediction map π whose output distribution is a correlated (joint) Gaussian with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$:

$$\pi(\boldsymbol{y}^{(\mathrm{t})}; C, \boldsymbol{X}^{(\mathrm{t})}) = \mathcal{N}(\boldsymbol{y}^{(\mathrm{t})}; \boldsymbol{\mu}(C, \boldsymbol{X}^{(\mathrm{t})}), \boldsymbol{K}(C, \boldsymbol{X}^{(\mathrm{t})})). \tag{1}$$

The ConvGNP takes in the context sets C and outputs a mean and non-stationary covariance function of a GP predictive, which can be queried at arbitrary target locations (Figure 1). It does this by first fusing the context sets into a gridded encoding using a SetConv layer (Gordon et al., 2020). The SetConv encoder interpolates context observations onto an internal grid with the density of observations captured

by a 'density channel' for each context set (example encoding shown in Figure B1). This endows the model with the ability to ingest multiple predictors of various modalities (gridded and point-based) and handle missing data (Appendix B.5). The gridded encoding is passed to a U-Net (Ronneberger et al., 2015), which produces a *representation* of the context sets with $\boldsymbol{R} = \text{U-Net}(\text{SetConv}(C))$. The tensor $\boldsymbol{R}$ is then spatially interpolated at each target location $\boldsymbol{x}_i^{(\mathrm{t})}$, yielding a vector $\boldsymbol{r}_i$ and enabling the model to predict at arbitrary locations. Finally, $\boldsymbol{r}_i$ is passed to multilayer perceptrons f and $\boldsymbol{g}$, parameterising the mean and covariance respectively with $\mu_i = f(\boldsymbol{r}_i)$ and $k_{ij} = \boldsymbol{g}(\boldsymbol{r}_i)^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{r}_j)$. This architecture results in a mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$ that are *functions* of C and $\boldsymbol{X}^{(\mathrm{t})}$ (Equation 1).

Constructing the covariances via a dot product leads to a low-rank covariance matrix structure, which is exploited to reduce the computational cost of predictions from cubic to linear in the number of targets. Furthermore, the use of a SetConv to encode the context sets results in linear scaling with the number of context points. This out-of-the-box scalability allows the ConvGNP to process 100,000 context points and predict over 100,000 target points in less than a second on a single GPU[2].

NPs can be considered *meta-learning* models (Foong et al., 2020) which *learn how to learn*, mapping directly from context sets to predictions without requiring retraining when presented with new tasks. This is useful in environmental sciences because it enables learning statistical relationships (such as correlations) that depend on the context observations. In contrast, conventional supervised learning models, such as GPs, instead learn fixed statistical relationships which do not depend on the context observations.

2.4. Training the ConvGNP

Training tasks $\mathcal{D}_\tau$ are generated by first sampling the day τ randomly from the training period, 1950–2013. Then, ERA5 grid cells are sampled uniformly at random across the entire 280×280 input space, with the number of ERA5 temperature anomaly context and target points drawn uniformly at random from $N_c \in \{5, 6, \ldots, 500\}$ and $N_t \in \{3000, 3001, \ldots, 4000\}$. The ConvGNP is trained to minimise the negative log-likelihood (NLL) of target values $\boldsymbol{y}_\tau^{(\mathrm{t})}$ under its output Gaussian distribution using the Adam optimiser. After each training epoch, the model is checkpointed if an improvement is made to the mean NLL on validation tasks from 2014–2017. For further model and training details see Appendix B.

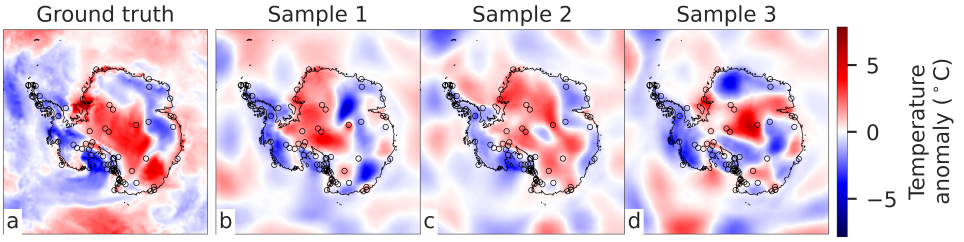

***Figure 2. The ConvGNP extrapolates plausible scenarios away from data. a**, ERA5 2 m temperature anomaly on January 1st 2018; **b-d**, ConvGNP samples after conditioning on ERA5 temperature anomaly observations at context locations, denoted by black circles, at Antarctic station locations. Comparing colours within the black circles across plots shows that the ConvGNP interpolates ground truth at context points while extrapolating expressive, plausible scenarios away from them.*

Once trained in this manner, the ConvGNP outputs expressive, non-stationary mean and covariance functions. When conditioning the ConvGNP on ERA5 temperature anomaly observations and drawing Gaussian samples on a regular grid, the samples interpolate ground truth at the context points and extrapolate plausible scenarios away from them (Figure 2). Running the ConvGNP on a regular grid with no temperature anomaly observations reveals the prior covariance structure learned by the model

[2]Our ConvGNP (with 4.16M parameters) takes 0.88 s to process a total of 100,000 context points (21,600 temperature points and 78,400 gridded auxiliary points) and predict over 100,000 target points on a 16 GB NVIDIA A4 GPU using TensorFlow's eager mode.

(Figure 3). The ConvGNP leverages the gridded auxiliary fields and day-of-year inputs from the second context set to output highly non-stationary spatial dependencies in surface temperature, such as sharp drops in covariance over the coastline (Figure 3a-c), anticorrelation (Figure 3a), and decorrelation over the Transantarctic Mountains (Figure 3b). In Appendix D we contrast this with GP prior covariances and further show that the ConvGNP learns seasonally-varying spatial correlation (Figure D1–Figure D3).

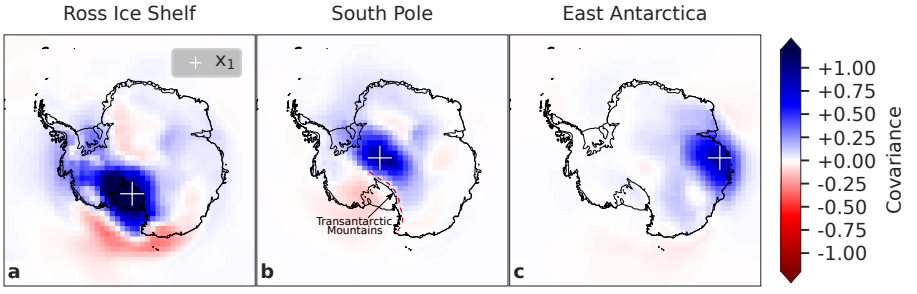

Figure 3. The ConvGNP learns spatially-varying covariance structure. *Prior covariance function, $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with $\boldsymbol{x}_1$ fixed at the white plus location and $\boldsymbol{x}_2$ varying over the grid. Plots are shown for three different $\boldsymbol{x}_1$-locations (the Ross Ice Shelf, the South Pole, and East Antarctica) for the 1st of June. The most prominent section of the Transantarctic Mountains is indicated by the red dashed line in **b**.*

3. Results

We evaluate the ConvGNP's ability to model 2 m daily-average surface temperature anomaly in a range of experiments, comparing the ConvGNP with GP baselines with both non-stationary and stationary covariance functions. We use three GP baselines with different non-isotropic covariance functions: the exponentiated quadratic (EQ), the rational quadratic (RQ), and the Gibbs kernel. The EQ and RQ are stationary because the covariance depends only on the distance between two input points, $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x} - \boldsymbol{x}')$. The Gibbs covariance function is a more sophisticated, non-stationary baseline, where the correlation length scale is allowed to vary over space (Fig. C1). As noted in Section 2.3, there is no simple way to condition vanilla GP models on multiple context sets, and so the GP baselines only ingest the context set containing the ERA5 observations. For more details on the GPs, including their covariance functions and training procedure, see Appendix C.

3.1. Performance on unseen data

To assess the models' abilities to predict unseen data, we set the number of targets $N_t = 2,000$ and iterate over context set sizes $N_c \in \{0, 25, 50, \ldots, 500\}$, generating 1,458 tasks from unseen test years 2018–2019 for each setting of N_c (Appendix B.1). For each task, we compute three performance metrics of increasing complexity. The first, the root mean squared error (RMSE), simply measures the difference between the model's mean prediction and ground truth. The second, the mean marginal NLL, includes the variances of the model's point-wise Gaussian distributions, measuring how confident and well-calibrated the marginal distributions are. The third metric, the joint NLL, uses the model's full joint Gaussian distribution, measuring how likely the true $\boldsymbol{y}^{(\mathrm{t})}$ vector is under the model. This quantifies the reliability of the model's off-diagonal spatial correlations as well as its marginal variances.

In general, the ConvGNP performs best, followed by the Gibbs GP, the RQ GP, and finally the EQ GP (Figure 4). There are some exceptions to this trend. For example, the models produce similar RMSEs for $N_c < 100$. This is likely because for small N_c the models revert to zero-mean predictions away from context points (matching the zero-mean of temperature anomaly over the training period) because there are insufficient observations. Another exception is that the Gibbs GP outperforms the ConvGNP on joint NLL for small N_c. This may be because the ConvGNP's training process biases learning towards 'easier' tasks (where N_c is larger). Alternatively, the ConvGNP's low-rank covariance

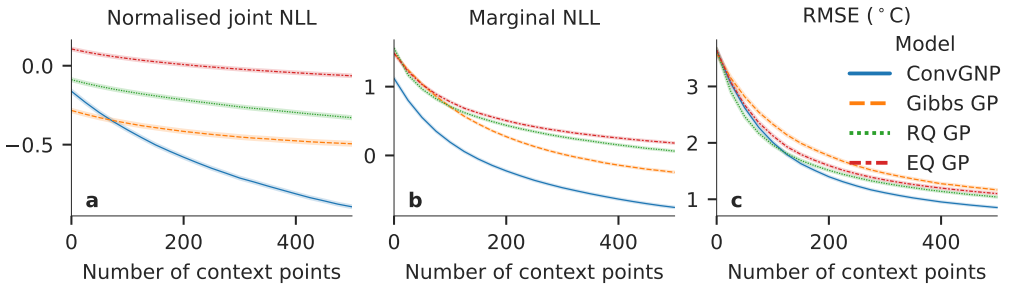

Figure 4. Test set results. *Mean metric values versus number of context points on the test set. The joint negative log-likelihood (NLL) is normalised by the number of targets. Error bars are standard errors.*

parameterisation could be poorly suited to small N_c. However, with increasing N_c, the ConvGNP significantly outperforms all three GP baselines across all three metrics, with its performance improving at a faster rate with added data. When averaging the results across N_c, the ConvGNP significantly outperforms all three GP baselines for each metric (Table E1). We further find that the ConvGNP's marginal distributions are substantially sharper and better calibrated than the GP baselines (Figure E1 and Figure E2), which is an important goal for probabilistic models (Gneiting et al., 2007). Well-calibrated uncertainties are also key for active learning, which is explored below in Section 3.2.

3.2. Sensor placement

Following previous works (Krause et al., 2008), we pose sensor placement as a discrete optimisation problem. The task is to propose a subset of K sensor placement locations, $\boldsymbol{X}^*$, from a set of S search locations, $\boldsymbol{X}^{(\mathrm{s})}$. In practice, to avoid the infeasible combinatorial cost of searching over multiple placements jointly, a *greedy* approximation is made by selecting one sensor placement at a time. Within a greedy iteration, a value is assigned to each query location $\boldsymbol{x}_i^{(\mathrm{s})}$ using an *acquisition function*, $\alpha(\boldsymbol{x}_i^{(\mathrm{s})}, \tau)$, specifying the utility of a new observation at $\boldsymbol{x}_i^{(\mathrm{s})}$ for time τ, which we average over J dates:

$$\alpha(\boldsymbol{x}_i^{(\mathrm{s})}) = \frac{1}{J} \sum_{j=1}^{J} \alpha(\boldsymbol{x}_i^{(\mathrm{s})}, \tau_j). \tag{2}$$

We use five acquisition functions which are to be maximised, defining a set of placement criteria (mathematical definitions are provided in Appendix F):

`JointMI`: mutual information (MI) between the model's prediction and the query sensor observation, imputing the missing value with the model's mean at the query location, $\bar{y}_{\tau,i}^{(\mathrm{s})}$.[3] This criterion attempts to minimise the model's joint entropy by minimising the log-determinant of the output covariance matrix, balancing minimising marginal variances with maximising correlation magnitude, which can be viewed as minimising uncertainty about the spatial patterns (MacKay, 1992). The joint MI has been used frequently in past work (Lindley, 1956; Krause et al., 2008; Schmidt et al., 2019).

`MarginalMI`: as above, but ignoring the off-diagonal elements in the models' Gaussian distributions and considering only the diagonal (marginal) entries. This criterion attempts to minimise the model's marginal entropy by minimising the log-variances in the output distribution.

`DeltaVar`: decrease in average marginal variance in the output distribution (similar to `MarginalMI` but using absolute variances rather than log-variances). Previous works have used this criterion for active learning both with neural networks (Cohn, 1993) and GPs (Seo et al., 2000).

`Remoteness`: distance to the closest sensor. This is a simple heuristic which proposes placements as far away as possible from the current observations. While this is a strong baseline, non-stationarities in the ground truth data will mean that it is sub-optimal. For example, a high density of sensors will be needed in areas where correlation length scales are short, and a low density where they are large. Therefore, the optimal sensor placement strategy should differ from and outperform this approach.

[3] A better approach would be to draw Monte Carlo samples over $y_{\tau,i}^{(\mathrm{s})}$, although this would be more costly – see Appendix F.

`Random`: uniform white noise function (i.e. placing sensors randomly). The performance of this criterion reflects the average benefit of adding new observations for a given model and context set.

The search locations $\boldsymbol{X}^{(\mathrm{s})}$ and target locations $\boldsymbol{X}_\tau^{(\mathrm{t})}$ are both defined on a $100\,\mathrm{km}$ grid over Antarctica, resulting in $S = N_t = 1,365$ targets and possible placement locations. The context set locations $\boldsymbol{X}_\tau^{(\mathrm{c})}$ are fixed at Antarctic temperature station locations (black circles in Figure 5) to simulate a realistic network design scenario. We use $J = 105$ dates from the validation period (2014–2017), sampled at a 14-day interval, to compute the acquisition functions. Heatmaps showing the above five acquisition functions on the $\boldsymbol{X}^{(\mathrm{s})}$ grid, using the ConvGNP for the model underlying the three uncertainty-reduction acquisition functions, are shown in Figure 5. There are interesting differences between the model-based acquisition functions of the ConvGNP, the Gibbs GP, and the EQ GP (Figure F1).

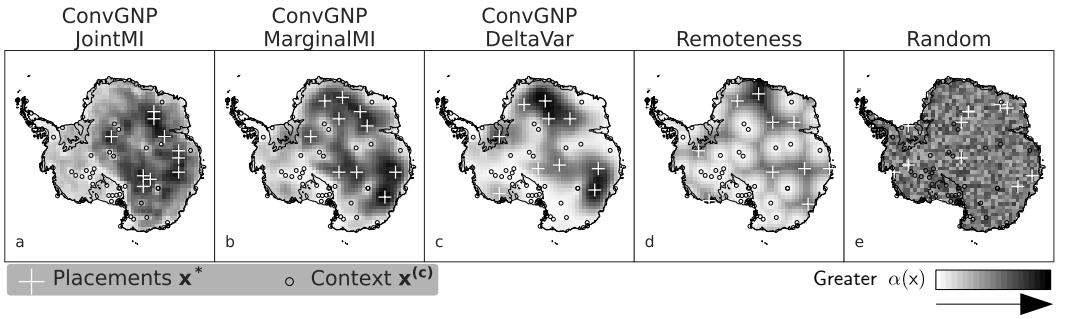

Figure 5. Acquisition functions and sensor placements for the ConvGNP and heuristic baselines. *Maps of acquisition function values $\alpha(\boldsymbol{x}_i^{(s)})$ for the initial $k = 1$ greedy iteration. The initial context set $\boldsymbol{X}^{(c)}$ is derived from real Antarctic station locations (black circles). Running the sensor placement algorithm for $K = 10$ sensor placements results in the proposed sensor placements $\boldsymbol{X}^*$ (white pluses). Each pixel is* $100\times100\,\mathrm{km}$.

3.2.1. Oracle acquisition function experiment

By using sensor placement criteria that reduce uncertainty in the model's predictions, one hopes that predictions also become more accurate in some way. For example, the entropy of the model's predictive distribution is the expected NLL of the data under the model, so the decrease in entropy from a new observation (i.e. the MI) should relate to the NLL improvement – assuming the model is well-specified for the data. Further, if the model's marginal distributions are well-calibrated, marginal variance relates to expected squared error. Therefore, the `JointMI`, `MarginalMI`, and `DeltaVar` acquisition functions should relate to improvements in joint NLL, marginal NLL, and RMSE, respectively. However, in general, the strength of these relationships are unknown. In the toy setting of this study, where ERA5 is treated as ground truth and is known everywhere, these relationships can be examined empirically.

We compare the ability of the ConvGNP, Gibbs GP, and EQ GP to predict the benefit of new observations based on the `JointMI`, `MarginalMI`, and `DeltaVar` acquisition functions, using `Remoteness` as a naïve baseline. The true benefit of observations is determined using *oracle* acquisition functions, α_{oracle}, where the ground truth ERA5 value is revealed at $\boldsymbol{x}_i^{(\mathrm{s})}$ and the average performance gain on the target set is measured for each metric: joint NLL, marginal NLL, and RMSE (Appendix F.3). Computing non-oracle and oracle acquisition functions at all S query locations produces vectors, $\boldsymbol{\alpha}(\boldsymbol{X}^{(\mathrm{s})})$ and $\boldsymbol{\alpha}_{\text{oracle}}(\boldsymbol{X}^{(\mathrm{s})})$. The Pearson correlation $r = \mathrm{corr}(\boldsymbol{\alpha}(\boldsymbol{X}^{(\mathrm{s})}), \boldsymbol{\alpha}_{\text{oracle}}(\boldsymbol{X}^{(\mathrm{s})}))$ between these vectors quantifies how strong the relationship is for a given model, acquisition function, and metric. With the context set initialised at Antarctic station locations (Section 3.2), the ConvGNP's joint MI achieves the best correlation with its joint NLL improvement ($r = 0.90$), as for its marginal MI with its marginal NLL improvement ($r = 0.93$) and its change in variance with its RMSE improvement ($r = 0.93$) (Figure 6a), substantially outperforming the `Remoteness` baseline in each case. The Gibbs GP's acquisition functions are less robust at predicting performance gain, with the joint MI being

particularly poor at predicting joint NLL improvement (Figure 6b). The EQ GP's model-based acquisition functions all perform similarly to `Remoteness` for each metric (Figure 6c), which is likely an artifact of its stationary covariance function.

We repeat the above analysis using the Kendall rank correlation coefficient, κ, which measures the similarity between the rankings of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}_{\text{oracle}}$ by computing the fraction of all pairs of search points $(\boldsymbol{x}_i^{(\text{s})}, \boldsymbol{x}_j^{(\text{s})})$ that are ordered the same way in the two rankings and normalising this fraction to lie in $(-1, 1)$ (Equation G.3). The findings are very similar to the Pearson correlation results above: only the ConvGNP has good alignment between acquisition functions and metrics, with `JointMI`, `MarginalMI`, and `DeltaVar` obtaining the best κ-values for joint NLL ($\kappa = 0.74$), marginal NLL ($\kappa = 0.82$), and RMSE ($\kappa = 0.84$), respectively (Figure G1).

These results indicate that the ConvGNP can robustly predict performance gain, unlike the GP baselines. Appendix G provides more detailed plots from this experiment, including the acquisition function heatmaps (Figure G2–Figure G4) and scatter plots for all the oracle/non-oracle pairs underlying Figure 6 (Figure G5–Figure G7).

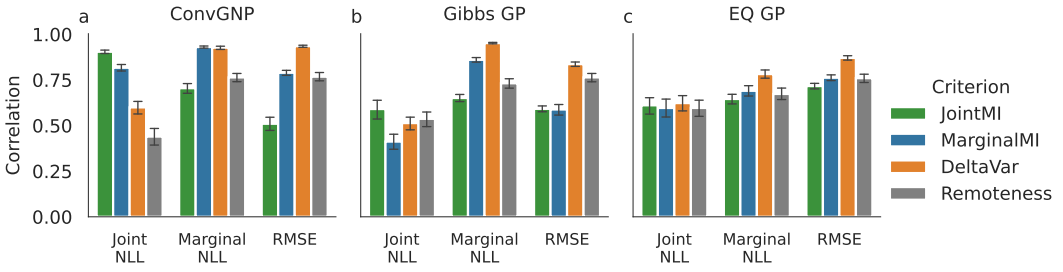

Figure 6. The ConvGNP reliably predicts the value of new observations. *Correlation between model-based and oracle acquisition functions, $\boldsymbol{\alpha}(\boldsymbol{X}^{(s)})$ and $\boldsymbol{\alpha}_{oracle}(\boldsymbol{X}^{(s)})$. Error bars indicate the 2.5%–97.5% quantiles from 5000 bootstrapped correlation values, computed by resampling the 1365 pairs of points with replacement, measuring how spatially consistent the correlation is across the search space $\boldsymbol{X}^{(s)}$.*

3.2.2. Sensor placement experiment

We now run a simulated greedy sensor placement experiment. After $\alpha(\boldsymbol{x}_i^{(\text{s})})$ is computed for $i = (1, \ldots, S)$, the i^* corresponding to the maximum value is selected. The corresponding input $x_{i^*}^{(\text{s})}$ is then appended to its context set, $\boldsymbol{X}_\tau^{(\text{c})} \to \{\boldsymbol{X}_\tau^{(\text{c})}, x_{i^*}^{(\text{s})}\}$. If α depends on the context y-values, we fill the missing observation with the model mean, $\boldsymbol{y}_\tau^{(\text{c})} \to \{\boldsymbol{y}_\tau^{(\text{c})}, \bar{y}_{\tau,i^*}^{(\text{s})}\}$, where $\bar{y}_{\tau,i}^{(\text{s})}$ is the model's mean at $\boldsymbol{x}_i^{(\text{s})}$ for time τ. This process is repeated until $K = 10$ placements have been made. To evaluate placement quality, we reveal ground truth to the models in the order they are proposed, computing performance metrics over 2018–2019 on the 100 km target grid. See Appendix H for full experiment details.

The ConvGNP's `JointMI`, `MarginalMI`, and `DeltaVar` placements substantially outperform `Remoteness` for the metrics they target by the 5th placement onwards (Figure 7a–c; Figure H1a,d,g), and lead to greater performance improvements by the $K = 10$th placement than both of the GP baselines (Figure H2).[4] This is despite the ConvGNP starting off with better performance than both of the GP baselines for each metric. Furthermore, the proposed locations from the ConvGNP model-based criteria differ greatly from `Remoteness` (Figure 5a–d), with the `JointMI` placements being notably clustered together (Figure 5a). In contrast, the EQ GP's model-based criteria propose diffuse placements (Figure F1g–i) which are strikingly similar to those of `Remoteness` and perform very similarly to `Remoteness` on the test data (Figure H1c,f,i), which is likely an artifact of the naïve stationary covariance. Future work should repeat these experiments with different initial sensor network configurations to assess the robustness of these results.

[4]The only exception to this is the Gibbs GP's `DeltaVar`, which improves its RMSE by 0.79 °C compared with 0.77 °C for the ConvGNP. However, the Gibbs GP starts off with an RMSE that is 0.60 °C worse than the ConvGNP (Figure H2).

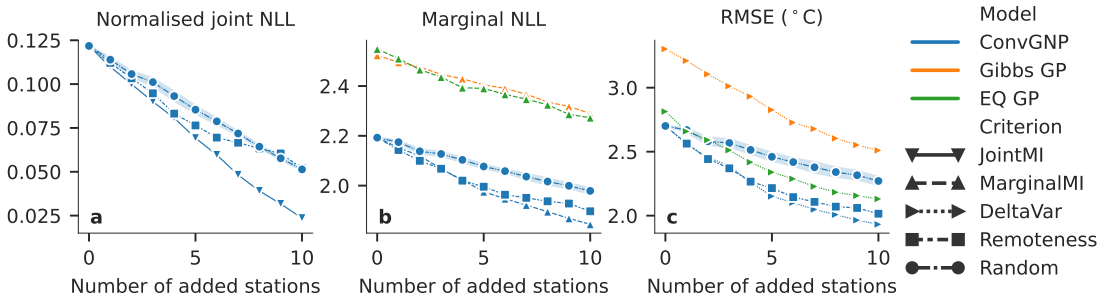

Figure 7. Sensor placement results. *Performance metrics on the sensor placement test data versus number of stations revealed to the models. Results are averaged over 243 dates in 2018–2019, with targets defined on a* 100 km *grid over Antarctica. For simplicity, we only plot the model-based criterion that targets the plotted metric. The GP baselines are shown on the marginal negative log-likelihood (NLL) and RMSE panels. For the joint NLL, the GP baselines perform far worse than the ConvGNP and are not shown. The confidence interval of* `Random` *is the standard error from 5 random placements.*

3.2.3. Multi-objective optimisation for finding cost-effective sensor placements

In practice, the scientific goals of sensor placement must be reconciled with cost and safety considerations, which are key concerns in Antarctic fieldwork (Lazzara et al., 2012) and will likely override the model's optimal siting recommendations $\boldsymbol{X}^*$. In this case, it is crucial that the model can faithfully predict observation informativeness across the entire search space $\boldsymbol{X}^{(s)}$, not just at the optimal sites $\boldsymbol{X}^*$, so that informativeness can be traded-off with cost. Leveraging our findings from Section 3.2.1 that the ConvGNP's `DeltaVar` is a robust indicator of RMSE and marginal NLL improvement (Figure 6a), we demonstrate a toy example of multi-objective optimisation with `DeltaVar` as a proxy for informativeness and `Remoteness` as a proxy for cost. One way of integrating cost in the optimisation is to constrain the search such that the total cost is within a pre-defined budget (Sviridenko, 2004; Krause et al., 2006). Alternatively, cost can be traded off with informativeness in the objective, allowing for unconstrained optimisation. We use Pareto optimisation for this purpose, which identifies a set of 'Pareto optimal' sites corresponding to points where the informativeness cannot be improved without an increase to the cost. These rank-1 points can then be removed, the Pareto optimal set computed again, and so on until all sites have been assigned a Pareto rank (Figure 8). This procedure trivially generalises to multiple objectives and could underlie a future operational, human-in-the-loop sensor placement recommendation system that leverages an accurate cost model to guide expert decision-making.

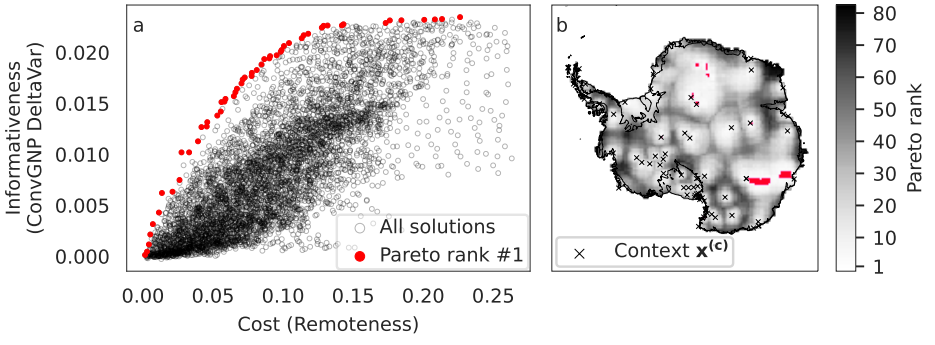

Figure 8. Trading off informativeness with cost*. Accounting for sensor placement cost using multi-objective Pareto optimisation, maximising the ConvGNP's* `DeltaVar` *(a proxy for informativeness) and minimising* `Remoteness` *(a proxy for cost).* ***a****, Scatter plot showing all pairs of informativeness and cost values.* ***b****, Heatmap of Pareto rank. The rank-1 Pareto set is highlighted in red for both plots.*

4. Discussion

In this study, we trained a ConvGNP regression model to spatially interpolate ERA5 Antarctic $2\,\mathrm{m}$ temperature anomaly. The ConvGNP learned seasonally-varying non-stationary spatial covariance by leveraging a second data stream ('context set' in meta-learning language) containing auxiliary predictor variables, such as the day of year and surface elevation. The more flexible architecture and second data stream allow the ConvGNP to make substantially better probabilistic predictions on test data than those of GP baselines, including a GP with a non-stationary covariance function. A simulated sensor placement experiment was devised with context ERA5 observations initialised at real Antarctic station locations. New sensor placements were evaluated via the reduction in model prediction uncertainty over the Antarctic continent, with different measures of uncertainty targeting different performance metrics. For each of these uncertainty-based acquisition functions, the ConvGNP predicts its true performance metric gain from new observations substantially more accurately than GP baselines. This leads to informative new sensor placements that improve the ConvGNP's performance metrics on test data by a wider margin than the GP baselines, despite the ConvGNP starting off with more performant predictions and thus having less room for improvement. These findings are notable given that GPs have a long history of use in geostatistics under the term 'kriging' (Cressie, 1993) and are frequently used for sensor placement. Equipped with a robust measure of placement informativeness from the ConvGNP, multi-objective Pareto optimisation could be used to account for sensor placement cost, pruning a large search space of possible locations into a smaller set of cost-effective sites which can be considered by human experts. Our approach can readily be applied to other geographies and climate variables by fitting a ConvGNP to existing reanalysis data and running a greedy sensor placement algorithm, such as the ones outlined in this work. However, there are some limitations to this approach, which we highlight below alongside recommendations for future work.

4.1. Limitations

Not accounting for real-world observations. The main limitation of the current approach is that by training the ConvGNP to spatially interpolate noise-free reanalysis output instead of real-world observations, the model measures the informativeness of reanalysis data and not of real-world observations. Two consequences arise from this shortcoming. First, the model does not account for real-world sensor noise. A simple way to alleviate this issue would be to simulate sensor noise by training the ConvGNP with varying levels of i.i.d. Gaussian noise added to the ERA5 context points, which could be explored in future work. The second consequence is that bias and coarse spatial resolution in the reanalysis data are reflected in the ConvGNP's predictions. One way to deal with this would be to train with observational data. However, real *in-situ* environmental sensor observations can be sparse in space or time, which brings a risk of spatial overfitting when used as training data for highly flexible models like the ConvGNP. An interesting potential solution is to pre-train the ConvGNP on simulated data and fine-tune it on observational data. Provided sufficient observational data for training, the fine-tuning phase would correct some of the simulator biases and lead to a better representation of the target variable.

The ConvGNP must learn how to condition on observations. The ConvGNP is directly trained to output a GP predictive, which is different from specifying a GP prior and then conditioning that prior on context observations using Bayes' rule. The ConvGNP's neural networks can learn non-Bayesian conditioning mechanics from the data, which brings greater flexibility at the cost of increased training requirements. Nevertheless, provided sufficient training data and an appropriate training scheme, the ConvGNP's conditioning flexibility is better-suited to complex environmental data than similar approaches like deep kernel learning (Wilson et al. 2015), where neural networks learn non-stationary prior GP covariance functions from data and then use standard Bayes' rule conditioning to compute posterior predictives (Appendix I). However, if insufficient data is available to train a flexible model like the ConvGNP, a more appropriate choice would be a less data-hungry model with stronger inductive biases and better-quantified epistemic uncertainty, such as a latent GP (Patel et al., 2022).

4.2. Future work

Possible extensions Going forwards, there are several possible extensions to this work with simple modifications to our approach. For example, our model can be used to rank the value of current stations (Tardif et al., 2022), which could identify redundant stations that can be moved to more valuable locations. Alternatively, the ConvGNP could be set up as a *forecasting* model, with the target data being some number of discrete time steps ahead of the context data. The same greedy sensor placement algorithms can then be used to find station sites that minimise forecast uncertainty, which is important for supporting safety-critical operations in remote regions like Antarctica that depend on reliable weather forecasts (Lazzara et al., 2012; Hakim et al., 2020). Another exciting avenue is to build a ConvGNP that can propose optimal trajectories for a fleet of moving robots (e.g. autonomous underwater vehicles) (Singh et al., 2007; Marchant and Ramos, 2012). One way to do this is to have two context sets of the target variable: one for the current time step ($\tau = 0$) and another for the next time step ($\tau = +1$). This model can propose perturbations to the robot locations from $\tau = 0$ to $\tau = +1$ (within speed limits) that minimise prediction uncertainty at $\tau = +1$. Trajectories can then be formed by running this model autoregressively. To extend our approach to non-Gaussian variables, our analysis could be repeated with models that output non-Gaussian stochastic processes, such as convolutional latent neural processes (Foong et al., 2020), normalising flows (Durkan et al., 2019), or autoregressive ConvCNPs (Bruinsma et al., 2023). In general, future work should explore training and architecture schemes that enable learning from multiple heterogeneous data sources, such as simulated data, satellite observations, and *in situ* stations. Foundation modelling approaches have recently shown substantial promise in this area (Nguyen et al., 2023) and could be explored with ConvNPs.

Comparison and integration with physics-based sensor placement methods As with any model-based sensor placement approach, the ConvGNP's measure of informativeness depends on the model itself. In general, an observation with high impact on the uncertainty of one model may have little impact on the uncertainty of another model. This raises interesting questions about which model should be trusted, particularly for models based on very different principles such as data-driven and physics-based numerical models. It would be insightful to examine the level of agreement or disagreement between the informativeness estimates of physics-based and ML models. Agreement would suggest that the informativeness predicted by the causal dynamics of the numerical model is also statistically evident in the training data of the ML model. However, a blocker to such intercomparison studies is the minimal overlap between the physics-based and ML sensor placement literatures. Future work should trace explicit links between these distinct research worlds to translate differing terminologies and facilitate the cross-pollination of ideas. For example, we identified potential ML analogues for several physics-based observing system design approaches: ablation-based variable importance methods (Fisher et al., 2019) for observing system experiments (OSEs; Boullot et al., 2016); gradient-based saliency methods (Bach et al., 2015) for adjoint modelling (Langland and Baker, 2004); and uncertainty-based active learning (Krause et al., 2008) for ESA (Torn and Hakim, 2008). Here we remark only on the latter, where we note a striking similarity. In ESA, sensor placement informativeness is measured by assimilating query observations into a numerical model and computing the reduction in ensemble member variance for a target quantity. This approach has been used for Antarctic temperature sensor placement in previous studies (Hakim et al., 2020; Tardif et al., 2022) with a goal of minimising the total marginal variance of Antarctic surface temperature in ensemble member samples from a numerical model, which can be seen as a Monte Carlo estimate of the `DeltaVar` acquisition function used in this study. This similarity makes ESA a ripe starting point for comparing the sensor informativeness estimates of numerical and ML models in future work. Other than simply comparing ML-based and physics-based sensor placement methods, future work could also integrate the two. For example, although the ConvGNP lacks the causal grounding of dynamical models, it can run orders of magnitude faster. Thus, the ConvGNP could nominate a few observation locations from a large search space to be analysed by more expensive physics-based techniques such as adjoint sensitivity (Loose et al., 2020; Loose and Heimbach, 2021).

5. Conclusion

The ConvGNP is a novel, flexible ML model with a range of capabilities that aid modelling complex spatiotemporal climate variables. These include an ability to ingest multiple predictors of various modalities (gridded and off-grid) and learn arbitrary mean and covariance functions from raw data. Using simulated Antarctic air temperature anomaly as a proof-of-concept, this study found that the ConvGNP can robustly evaluate the informativeness of new sites for environmental sensors, unlike GP baselines. By providing a faithful notion of observation informativeness, the ConvGNP has the potential to underlie an operational, scalable environmental sensor placement recommendation tool which can find cost-effective locations for new measurements that substantially reduce uncertainty. We see our approach as complementary to existing physics-based methods, with interesting avenues for comparison and integration in future. By better tackling the complexities of environmental data, our work could help answer calls for hybrid 'digital twin' models that support real-world decision-making by leveraging the benefits of both dynamical understanding and ML (Blair, 2021; Gettelman et al., 2022).

Acknowledgements. We thank Markus Kaiser, Marta Garnelo, Samantha Adams, Kevin Murphy, Anton Geraschenko, Michael Brenner, and Elre Oldewage for insightful early discussions and feedback on this work. We also thank Steve Colwell for assistance with accessing the Antarctic station data. Finally, we thank the two anonymous Climate Informatics 2023 reviewers and the anonymous reviewer from the NeurIPS 2022 Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems for suggestions that improved this manuscript.

Funding Statement. This work was supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/W006022/1, particularly the AI for Science theme within that grant & The Alan Turing Institute. This research was conducted while WPB was a student at the University of Cambridge, where he was supported by the Engineering and Physical Sciences Research Council (studentship number 10436152). RET is supported by Google, Amazon, ARM, Improbable and EPSRC grant EP/T005386/1. DJ is supported by a UKRI Future Leaders Fellowship (MR/T020822/1). MAL is supported is via the US National Science Foundation, grant number 1924730.

Competing Interests. None

Code and Data Availability Statement. The code to reproduce this paper's results will be released at https://github.com/tom-andersson/EDS2022-convgnp-sensor-placement. We are also currently developing a generic Python package for modelling environmental data with neural processes, which we aim to release in future at https://github.com/tom-andersson/deepsensor. The ConvGNP was implemented using the Python package `neuralprocesses` (https://github.com/wesselb/neuralprocesses). All GPs were implemented using the Python package `stheno` (https://github.com/wesselb/stheno) and optimised using the Python package `varz` (https://github.com/wesselb/varz). Pareto optimisation for the multi-objective optimisation example was performed using `paretoset` (https://github.com/tommyod/paretoset). Antarctic station data, containing the station locations used in this study, is available from ftp.bas.ac.uk/src/. The ERA5 data was downloaded from https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview. The Antarctic land mask and elevation field was obtained from version 2 of the BedMachine dataset from https://nsidc.org/data/nsidc-0756/versions/2.

Ethical Standards. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

Author Contributions. Contributions are listed in the order of the author list. Conceptualisation: T.R.A, W.P.B, S.M, D.J.C, J.S.H, R.E.T. Methodology: T.R.A, W.P.B, S.M, J.R, A.V, D.C.J, R.E.T. Software: T.R.A, W.P.B. Data curation: T.R.A. Visualisation: T.R.A. Supervision: M.A.L, D.C.J, J.S.H, R.E.T. Project administration: D.C.J, J.S.H. Funding acquisition: J.S.H. Writing original draft: T.R.A, W.P.B, S.M, J.R. Writing – review & editing: T.R.A, W.P.B, S.M, J.R, A.C-C, A.V, A-L.E, M.A.L, D.C.J, R.E.T.

Supplementary Material. An Appendix is included with this submission.

References

Addison, H., Kendon, E., Ravuri, S., Aitchison, L., and Watson, P. A. (2022). Machine learning emulation of a local-scale UK climate model. In *Tackling Climate Change with Machine Learning workshop at NeurIPS 2022*. arXiv. arXiv:2211.16116 [physics].

Andersson, T. R., Hosking, J. S., PÃ©rez-Ortiz, M., Paige, B., Elliott, A., Russell, C., Law, S., Jones, D. C., Wilkinson, J., Phillips, T., Byrne, J., Tietsche, S., Sarojini, B. B., Blanchard-Wrigglesworth, E., Aksenov, Y., Downie, R., and Shuckburgh, E. (2021). Seasonal Arctic sea ice forecasting with probabilistic deep learning. *Nature Communications*, 12(1):5124.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation. *PLOS ONE*, 10(7):e0130140.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., and Tian, Q. (2022). Pangu-Weather: A 3D High-Resolution Model for Fast and Accurate Global Weather Forecast. arXiv:2211.02556 [physics].

Blair, G. S. (2021). Digital twins of the natural environment. *Patterns*, 2(10):100359.

Boullot, N., Rabier, F., Langland, R., Gelaro, R., Cardinali, C., Guidard, V., Bauer, P., and Doerenbecher, A. (2016). Observation impact over the southern polar area during the Concordiasi field campaign. *Quarterly Journal of the Royal Meteorological Society*, 142(695):597–610.

Bracegirdle, T. J. and Marshall, G. J. (2012). The Reliability of Antarctic Tropospheric Pressure and Temperature in the Latest Global Reanalyses. *Journal of Climate*, 25(20):7138–7146.

Bromwich, D. H. and Fogt, R. L. (2004). Strong Trends in the Skill of the ERA-40 and NCEP–NCAR Reanalyses in the High and Midlatitudes of the Southern Hemisphere, 1958–2001. *Journal of Climate*, 17(23):4603–4619.

Bruinsma, W., Markou, S., Requeima, J., Foong, A. Y. K., Andersson, T., Vaughan, A., Buonomo, A., Hosking, S., and Turner, R. E. (2023). Autoregressive Conditional Neural Processes. In *The Eleventh International Conference on Learning Representations*.

Bruinsma, W., Requeima, J., Foong, A. Y. K., Gordon, J., and Turner, R. E. (2021). The Gaussian Neural Process. In *Third Symposium on Advances in Approximate Bayesian Inference*.

Cohn, D. (1993). Neural Network Exploration Using Optimal Experiment Design. In *Advances in Neural Information Processing Systems*, volume 6.

Cressie, N. (1993). *Statistics for Spatial Data*. John Wiley & Sons.

Durkan, C., Bekasovs, A., Murray, I., and Papamakarios, G. (2019). Neural Spline Flows. In *Advances in Neural Information Processing Systems*, pages 7511–7522.

Fisher, A., Rudin, C., and Dominici, F. (2019). All Models are Wrong, but *Many* are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of machine learning research*, 20:177.

Foong, A., Bruinsma, W., Gordon, J., Dubois, Y., Requeima, J., and Turner, R. (2020). Meta-Learning Stationary Stochastic Process Prediction with Convolutional Neural Processes. In *Advances in Neural Information Processing Systems*, volume 33, pages 8284–8295. Curran Associates, Inc.

Fortuin, V., Strathmann, H., and Rätsch, G. (2020). Meta-Learning Mean Functions for Gaussian Processes. arXiv:1901.08098 [cs, stat].

Garnelo, M., Rosenbaum, D., Maddison, C., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D., and Eslami, S. M. A. (2018a). Conditional Neural Processes. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1704–1713. PMLR. ISSN: 2640-3498.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018b). Neural Processes. In *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*. arXiv:1807.01622 [cs, stat].

Gettelman, A., Geer, A. J., Forbes, R. M., Carmichael, G. R., Feingold, G., Posselt, D. J., Stephens, G. L., van den Heever, S. C., Varble, A. C., and Zuidema, P. (2022). The future of Earth system prediction: Advances in model-data fusion. *Science Advances*, 8(14):eabn3488.

Gibbs, M. (1997). Bayesian gaussian processes for regression and classication. *PhD Thesis*.

Gneiting, T., Balabdaoui, F., and Raftery, A. E. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268.

Gordon, J., Bruinsma, W. P., Foong, A. Y. K., Requeima, J., Dubois, Y., and Turner, R. E. (2020). Convolutional Conditional Neural Processes. In *International Conference on Learning Representations*.

Hakim, G. J., Bumbaco, K. A., Tardif, R., and Powers, J. G. (2020). Optimal Network Design Applied to Monitoring and Forecasting Surface Temperature in Antarctica. *Monthly Weather Review*, 148(2):857–873.

Hensman, J., Fusi, N., and Lawrence, N. D. (2013). Gaussian processes for Big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, UAI'13, pages 282–290.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N. (2020). The ERA5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049.

Hoffman, R. N. and Atlas, R. (2016). Future Observing System Simulation Experiments. *Bulletin of the American Meteorological Society*, 97(9):1601–1616.

Jung, T., Gordon, N. D., Bauer, P., Bromwich, D. H., Chevallier, M., Day, J. J., Dawson, J., Doblas-Reyes, F., Fairall, C., Goessling, H. F., Holland, M., Inoue, J., Iversen, T., Klebe, S., Lemke, P., Losch, M., Makshtas, A., Mills, B., Nurmi, P., Perovich, D., Reid, P., Renfrew, I. A., Smith, G., Svensson, G., Tolstykh, M., and Yang, Q. (2016). Advancing Polar Prediction Capabilities on Daily to Seasonal Time Scales. *Bulletin of the American Meteorological Society*, 97(9):1631–1647.

Krause, A., Guestrin, C., Gupta, A., and Kleinberg, J. (2006). Near-optimal sensor placements: maximizing information while minimizing communication cost. In *2006 5th International Conference on Information Processing in Sensor Networks*, pages 2–10.

Krause, A., Singh, A., and Guestrin, C. (2008). Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, 9(8):235–284.

Langland, R. H. and Baker, N. L. (2004). Estimation of observation impact using the NRL atmospheric variational data assimilation adjoint system. *Tellus A: Dynamic Meteorology and Oceanography*, 56(3):189–201.

Lazzara, M. A., Weidner, G. A., Keller, L. M., Thom, J. E., and Cassano, J. J. (2012). Antarctic Automatic Weather Station Program: 30 Years of Polar Observation. *Bulletin of the American Meteorological Society*, 93(10):1519–1537.

Lindley, D. V. (1956). On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

Loose, N. and Heimbach, P. (2021). Leveraging Uncertainty Quantification to Design Ocean Climate Observing Systems. *Journal of Advances in Modeling Earth Systems*, 13(4).

Loose, N., Heimbach, P., Pillar, H. R., and Nisancioglu, K. H. (2020). Quantifying Dynamical Proxy Potential Through Shared Adjustment Physics in the North Atlantic. *Journal of Geophysical Research: Oceans*, 125(9):e2020JC016112.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604.

Majumdar, S. J. (2016). A Review of Targeted Observations. *Bulletin of the American Meteorological Society*, 97(12):2287–2303.

Marchant, R. and Ramos, F. (2012). Bayesian optimisation for Intelligent Environmental Monitoring. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2242–2249. ISSN: 2153-0866.

Markou, S., Requeima, J., Bruinsma, W. P., Vaughan, A., and Turner, R. E. (2022). Practical Conditional Neural Processes Via Tractable Dependent Predictions. In *The Tenth International Conference on Learning Representations*.

Mitra, R., McGough, S. F., Chakraborti, T., Holmes, C., Copping, R., Hagenbuch, N., Biedermann, S., Noonan, J., Lehmann, B., Shenvi, A., Doan, X. V., Leslie, D., Bianconi, G., Sanchez-Garcia, R., Davies, A., Mackintosh, M., Andrinopoulou, E.-R., Basiri, A., Harbron, C., and MacArthur, B. D. (2023). Learning from data with structured missingness. *Nature Machine Intelligence*, 5(1):13–23.

Morlighem, M. (2020). MEaSUREs BedMachine Antarctica, Version 2. *NASA National Snow and Ice Data Center DAAC*.

Nguyen, T., Brandstetter, J., Kapoor, A., Gupta, J. K., and Grover, A. (2023). ClimaX: A foundation model for weather and climate. arXiv:2301.10343 [cs].

Ober, S. W., Rasmussen, C. E., and Wilk, M. v. d. (2021). The promises and pitfalls of deep kernel learning. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 1206–1216. PMLR. ISSN: 2640-3498.

Odena, A., Dumoulin, V., and Olah, C. (2016). Deconvolution and Checkerboard Artifacts. *Distill*, 1(10):e3.

Patacchiola, M., Turner, J., Crowley, E. J., O'Boyle, M., and Storkey, A. (2020). Bayesian Meta-Learning for the Few-Shot Setting via Deep Kernels. In *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*. arXiv:1910.05199 [cs, stat].

Patel, Z. B., Batra, N., and Murphy, K. (2022). Uncertainty Disentanglement with Non-stationary Heteroscedastic Gaussian Processes for Active Learning. In *NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems, 2022*. arXiv:2210.10964 [cs, stat].

Rasmussen, C. E. (2004). Gaussian Processes in Machine Learning. Springer.

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., Fitzsimons, M., Athanassiadou, M., Kashem, S., Madge, S., Prudden, R., Mandhane, A., Clark, A., Brock, A., Simonyan, K., Hadsell, R., Robinson, N., Clancy, E., Arribas, A., and Mohamed, S. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241.

Schmidt, K., Smith, R. C., Hite, J., Mattingly, J., Azmy, Y., Rajan, D., and Goldhahn, R. (2019). Sequential optimal positioning of mobile sensors using mutual information. *Statistical Analysis and Data Mining*, 12(6).

Seo, S., Wallat, M., Graepel, T., and Obermayer, K. (2000). Gaussian process regression: active data selection and test point rejection. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, volume 3, pages 241–246 vol.3. ISSN: 1098-7576.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, 104(1):148–175.

Singh, A., Krause, A., Guestrin, C., Kaiser, W., and Batalin, M. (2007). Efficient planning of informative paths for multiple robots. In *Proceedings of the 20th international joint conference on Artifical intelligence*, pages 2204–2211.

Singh, A., Ramos, F., Whyte, H. D., and Kaiser, W. J. (2010). Modeling and decision making in spatio-temporal processes for environmental surveillance. In *2010 IEEE International Conference on Robotics and Automation*, pages 5490–5497. ISSN: 1050-4729.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32:41–43.

Tardif, R., Hakim, G. J., Bumbaco, K. A., Lazzara, M. A., Manning, K. W., Mikolajczyk, D. E., and Powers, J. G. (2022). Assessing observation network design predictions for monitoring Antarctic surface temperature. *Quarterly Journal of the Royal Meteorological Society*, 148(743):727–746.

Titsias, M. (2009). Variational Learning of Inducing Variables in Sparse Gaussian Processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, pages 567–574. PMLR. ISSN: 1938-7228.

Torn, R. D. and Hakim, G. J. (2008). Ensemble-Based Sensitivity Analysis. *Monthly Weather Review*, 136(2):663–677.

Vaughan, A., Tebbutt, W., Hosking, J. S., and Turner, R. E. (2021). Convolutional conditional neural processes for local climate downscaling. *Geoscientific Model Development Discussions*, pages 1–25.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). Exact Gaussian Processes on a Million Data Points. In *Advances in Neural Information Processing Systems*, volume 32.

Weissmann, M., Harnisch, F., Wu, C.-C., Lin, P.-H., Ohta, Y., Yamashita, K., Kim, Y.-H., Jeon, E.-H., Nakazawa, T., and Aberson, S. (2011). The Influence of Assimilating Dropsonde Data on Typhoon Track and Midlatitude Forecasts. *Monthly Weather Review*, 139(3):908–920.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. (2015). Deep Kernel Learning. In *Artificial Intelligence and Statistics (AISTATS)*. arXiv:1511.02222 [cs, stat].

A. Data considerations

In this section we provide details on the data sources, preprocessing, and normalisation.

A.1. Data sources

The daily-averaged temperature reanalysis data was obtained from ERA5 (Hersbach et al., 2020). The land mask and elevation field was obtained from the BedMachine dataset (Morlighem, 2020). Antarctic temperature locations from staffed and automatic weather stations were downloaded from `ftp.bas.ac.uk/src/`.

A.2. Data preprocessing

The temperature anomaly data and land/elevation auxiliary data were regridded from lat/lon to a Southern Hemisphere Equal Area Scalable Earth 2 (EASE2) grid at $25\,\mathrm{km}$ resolution and cropping to a size of 280×280. This centres the data on the South Pole.

Temperature anomalies were computed from the absolute temperature values by first computing the daily-average climatology across 1950-2013 (i.e. averaging the absolute temperature over time for each day of year, returning a $366 \times 280 \times 280$ tensor). Then, for each day of year, the climatological average was subtracted from the absolute temperature values, returning anomaly values.

A.3. Data normalisation

To aid the training process, we normalised the data before passing it to the ConvGNP and GP models. The temperature data was normalised from Celsius to a mean of 0 and standard deviation of 1. The elevation field was normalised from metres to values in $[0, 1]$. The land mask already took appropriate normalised values in $\{0, 1\}$. The input coordinates were normalised from metres to take values in $[-1, 1]$.

B. The ConvGNP model

Here we provide details on the ConvGNP training procedure and architecture. A high-level schematic of the ConvGNP forward-pass is shown in Figure 1. We refer the reader to Markou et al. 2022 for further model details.

B.1. Generation of $\mathcal{D}_\tau$ for the training, validation, and test datasets

Each daily-average training dataset $\mathcal{D}_\tau$ was generated by first drawing the integer number of simulated temperature anomaly context points $N_c \sim \mathrm{Unif}\{5, 6, \ldots 500\}$ and target points $N_t \sim \mathrm{Unif}\{3000, 3001, \ldots 4000\}$. Allowing for randomness in N_c encourages the model to learn to deal with both data-sparse and data-rich scenarios. Using a fairly large number of target points ensures there is sufficient signal for learning the covariance structure of the data while not incurring the computational cost of a very large N_t. Next, given the randomly sampled N_c and N_t, the 280×280 ERA5 grid cells were sampled uniformly at random to generate the ERA5 context and target locations, $\boldsymbol{X}_\tau^{(\mathrm{c})}$ and $\boldsymbol{X}_\tau^{(\mathrm{t})}$.

For the training dates, the random seed used for generating $\mathcal{D}_\tau$ is changed every epoch, allowing for an infinitely growing training dataset. In contrast, for the validation and test dates, fixed random seeds are used so that the $\mathcal{D}_\tau$ are deterministically random. This ensures the validation and test metrics are deterministic during and after training.

For the test results given in Table E1, we loop over $N_c \in \{0, 25, 50, \ldots 500\}$ and fix N_t to a value of 2,000. For each setting of N_c, we generate test tasks $\mathcal{D}_\tau$ by looping twice over each day in 2018-2019, resulting in 1,458 test tasks per N_c.

B.2. Antarctic surface temperature anomaly ConvGNP training procedure

The model was trained on data from 1950-2013. An Adam optimiser was used with a learning rate of 8×10^{-5} and an NLL loss function. Gradients with respect to the loss were averaged over batches of two datasets. Validation tasks from 2014-2017 were used for checkpointing the model weights based on the mean NLL over the validation tasks. In total, the ConvGNP was trained for 170 epochs. Training took 11.5 days on a Tesla V100 GPU with a simple implementation of the training pipeline. This could

be improved with better computational practices, such as using TensorFlow's graph mode rather than eager mode, which was not supported by the ConvGNP implementation at the time of the experiments.

B.3. ConvGNP architecture

For the ConvGNP model we use the same architecture as described in Markou et al. 2022, except for a few modifications. The U-Net component of the encoder uses 5x5 convolutional kernels with the following sequence of channel numbers (d.s. = 2x2 downsample layer, u.s. = 2x2 upsample layer):

$$128 \xrightarrow{\text{d.s.}} 128 \xrightarrow{\text{d.s.}} 128 \xrightarrow{\text{d.s.}} 128 \xrightarrow{\text{u.s.}} 128 \xrightarrow{\text{u.s.}} 128 \xrightarrow{\text{u.s.}} 128.$$

We use 128 basis functions for the covariance-parameterising neural network, $\boldsymbol{g}$. Using 128 channels for each layer in the U-Net means there are no dimensionality bottlenecks that could reduce the actual rank of the output lowrank covariance matrix. We use bilinear resize operators for the upsampling layers to fix checkerboard artifacts that we encountered when using standard zero-padding upsampling (Odena et al., 2016). For the internal discretisation density of the model, we used 150 points-per-unit (i.e., a 1×1 square of input space contains 150×150 internal discretisation points). We chose 150 points-per-unit to be close to the density of the ERA5 data in normalised coordinates, which is 140×140 in a 1×1 square of input space.[5]

The hyperparameter settings above construct a ConvGNP with 4.16 million learnable parameters. In addition, the choices for the U-Net filter size and internal discretisation density results in a receptive field of over $1500\,\mathrm{km}$. In other words, context observations can influence target predictions in the Gaussian predictive distribution up to roughly $750\,\mathrm{km}$ in either direction of the $\boldsymbol{x}_1$- or $\boldsymbol{x}_2$-dimensions.

B.4. ConvGNP input data

The ConvGNP receives two context sets as input. The first contains observations of the simulated ERA5 daily-average temperature anomaly. The second contains a set of 6 gridded auxiliary and meta-data variables. These are: elevation, land mask, $\cos(\text{day of year})$, $\sin(\text{day of year})$, x_1, and x_2. The $\cos(\text{day of year})$ and $\sin(\text{day of year})$ inputs, where the day of year is normalised between 0 and 2π, together define a circular variable that rotates once per year. This informs the model at what time of year $\mathcal{D}_\tau$ corresponds to, helping with learning seasonal variations in the data. The x_1 and x_2 gridded fields inform the model where in input space the data corresponds to. The gridded auxiliary fields that vary over input space are crucial for allowing the ConvGNP to model spatial non-stationarity. This is because they break the U-Net's translation equivariance property. Future work could explore using *learnable* input auxiliary channels, as in Addison et al. 2022, which could lead to richer non-stationarity at the cost of a greater spatial overfitting risk.

B.5. The SetConv encoder enables fusing data sources with multiple modalities and missing values

The SetConv encoder (Gordon et al., 2020), which fuses the context sets into a single gridded tensor, enables the ConvGNP to model 1) missing data, 2) multiple data streams, and 3) both gridded and off-grid data modalities. This is achieved, in part, through the use of a 'density channel' for each context set (Gordon et al., 2020). The density channel measures the density of context points by placing a small Gaussian basis function of unit amplitude at each observation location, such that the density channel takes values close to zero away from observations. Output y-values of context points are encoded in a similar fashion, but the amplitude of the Gaussian basis function is weighted by the value of y. These intermediate functional representations are then discretised onto the model's internal grid to yield the density channels and (N-dimensional) 'data channels' in the gridded encoding. This set-up allows the model to distinguish between the case where an observation is made with a y-value of 0 (data channel is zero but density channel is non-zero), and the case where there is no observation available (both the data and density channels are zero).

[5] Note, finer resolution context data would motivate the use of higher internal discretisation densities.

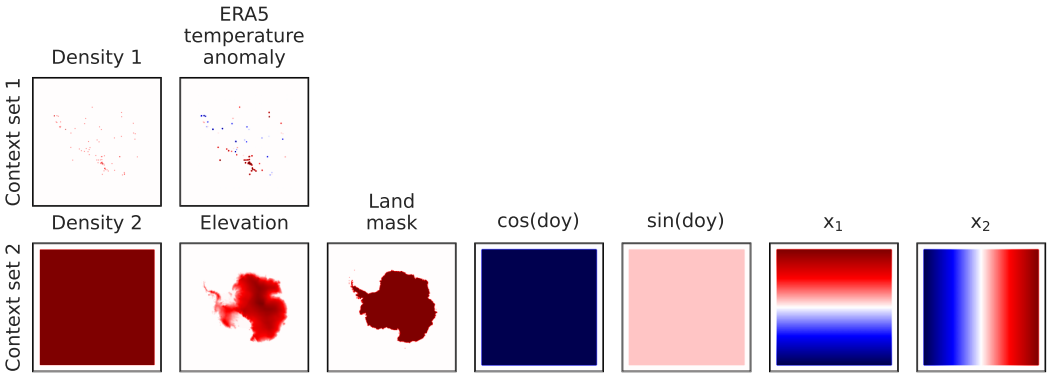

Figure B1. *Example output of the ConvGNP's SetConv encoder. The SetConv was passed a context set with ERA5 temperature anomaly at Antarctic station locations.*

Figure B1 shows the channels of a gridded encoding, which are splayed out to highlight the two input context sets provided to the ConvGNP in this study. The density channel for the first context set pinpoints the scattered, point-based temperature anomaly observation locations. The second density channel pertains to the gridded auxiliary context set and thus takes a constant value within the region of data. While in this setting the second gridded context set contained auxiliary variables with no missing data, the SetConv can represent missing data with gridded variables as well as non-gridded variables. For example, missing satellite observations due to cloud cover would be captured by patterns of zeros in the density channel. The density channel can thus be seen as a kind of *missing data channel*, where missing data (e.g. due to sensor malfunction, clouds, or the absence of sensing equipment) is captured with density values close to zero. Therefore, the SetConv encoder equips geospatial deep learning models with an ability to handle missing data, which is an important problem in many application areas (Mitra et al., 2023). However, the degree to which the model can learn to respond to missing data appropriately depends on a training scheme with sufficient examples of missing data, as discussed in Section 4.

C. Gaussian Process benchmarks

Here we provide details on the GP baseline kernels and hyperparameter fitting procedure. All GPs were implemented using the Python package `stheno` (https://github.com/wesselb/stheno) and optimised using the Python package `varz` (https://github.com/wesselb/varz).

C.1. Gibbs GP

The Gibbs kernel (Gibbs, 1997) is a non-stationary generalisation of the EQ kernel. In the $\boldsymbol{x} \in \mathbb{R}^2$ case, the covariance function is given by:

$$k_{\text{Gibbs}}(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \prod_{i=1}^{2} \left(\frac{2 l_i(\boldsymbol{x}) l_i(\boldsymbol{x}')}{l_i(\boldsymbol{x})^2 + l_i(\boldsymbol{x}')^2} \right)^{1/2} \exp\left(- \sum_{i=1}^{2} \frac{(x_i - x_i')^2}{l_i(\boldsymbol{x})^2 + l_i(\boldsymbol{x}')^2} \right), \tag{C.1}$$

where σ^2 is the variance, and length scale functions $l_1(\boldsymbol{x})$ and $l_2(\boldsymbol{x})$ dictate the length scales in the x_1- and x_2-directions. We parameterise the length scale functions $l_i(\boldsymbol{x})$ as a weighted sum of M regularly placed Gaussian basis functions,

$$l_i(\boldsymbol{x}) = \sum_{m=1}^{M} \theta_{i,m} \exp\left(- \frac{(\boldsymbol{x} - \boldsymbol{x}_m^{(\mu)})^T (\boldsymbol{x} - \boldsymbol{x}_m^{(\mu)})}{2\lambda^2} \right), \tag{C.2}$$

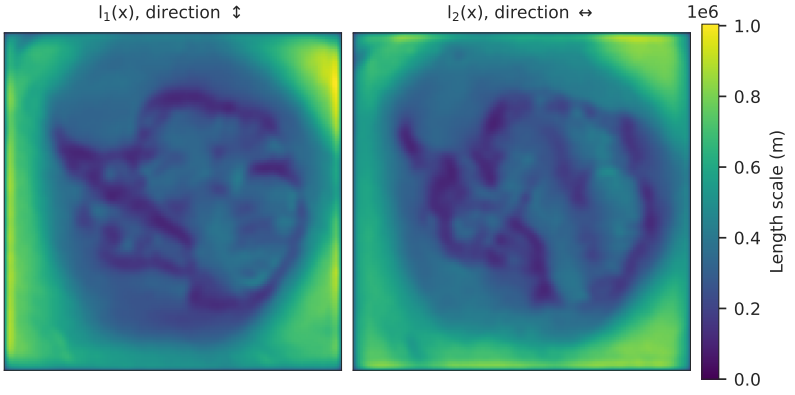

Figure C1. *Learned Gibbs GP length scale functions, $l_1(\boldsymbol{x})$ and $l_2(\boldsymbol{x})$.*

where the $\theta_{i,m}$ are the constrained-positive weights of basis function m for input dimension i, and the basis functions are placed with the $\boldsymbol{x}_m^{(\mu)}$ on a 100×100 grid spanning the input space. The basis function length scale λ is kept fixed and equal to the spacing between basis functions. We note that the basis function spacing controls how quickly the length scale functions can vary, and is a fixed (untrainable) hyperparameter. Too many basis functions can lead to overfitting, while too few can lead to underfitting. We tried different settings and chose a 100×100 grid as the most performant.

We train the parameters $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \sigma\}$, along with the other hyperparameters, using gradient descent on the negative log marginal likelihood (NLML) using an Adam optimiser with learning rate of 5×10^{-3} and a batch size of 10. We used 1950-2013 as a training period, subsampling the dates by a factor of 3, and sampling 500 random context locations for each of the training tasks Appendix B.1. Training was halted after the NLL on validation data spanning 2014–2017 did not improve for 5 epochs.

Figure C1 shows the trained length scale functions $l_1(\boldsymbol{x})$ and $l_2(\boldsymbol{x})$, revealing interesting detail such as very low correlation length scales perpendicular to the coastline.

C.2. Exponentiated quadratic and rational quadratic GPs

We also include more simplistic GP baselines using non-isotropic exponentiated quadratic (EQ) and rational quadratic (RQ) kernels, which are both stationary prior covariance functions (unlike the Gibbs kernel). The non-isotropic EQ kernel is:

$$k_{\text{EQ}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{(x_1 - x_1')^2}{2\ell_1^2} - \frac{(x_2 - x_2')^2}{2\ell_2^2}\right), \tag{C.3}$$

where σ^2 is the variance, and ℓ_1 and ℓ_2 are the length scales in each input dimension. The non-isotropic RQ kernel is:

$$k_{\text{RQ}}(\mathbf{x}, \mathbf{x}') = \sigma^2 \left(1 + \frac{(x_1 - x_1')^2}{2\alpha\ell_1^2} + \frac{(x_2 - x_2')^2}{2\alpha\ell_2^2}\right)^{-\alpha}, \tag{C.4}$$

where α is the shape parameter, controlling the smoothness of the kernel. The RQ kernel can be seen as an infinite sum of EQ kernels with different length scales (Rasmussen, 2004).

We fit the EQ and RQ GP hyperparameters using the L-BFGS-B algorithm on a batch of 730 dates sampled randomly from 1950-2013. The EQ and RQ GPs are thus exposed to fewer training tasks. However, these models only have a few parameters each. We found that increasing the training set size did not yield improved performance.

D. Non-stationarity in the ConvGNP

The ConvGNP learns richer spatial covariance structure than the GP baselines (Figure D1). Further, while our implementation of the ConvGNP only models correlations over 2D space (i.e. modelling time independently), the model can leverage the day of year auxiliary inputs to learn seasonal non-stationarity in the data (Figure D2). We note that the main changes in the ConvGNP's covariance from summer to winter are caused by changes in the magnitudes of the marginal variances (temperature anomalies take more extreme values in winter). However, the spatial correlations also change (Figure D3). For example, the Ross Ice Shelf site becomes less correlated with the Southern Ocean (Figure D3a), the South Pole becomes more correlated with the surrounding region (Figure D3b), and the East Antarctica site becomes more correlated with the Southern Ocean (Figure D3c).

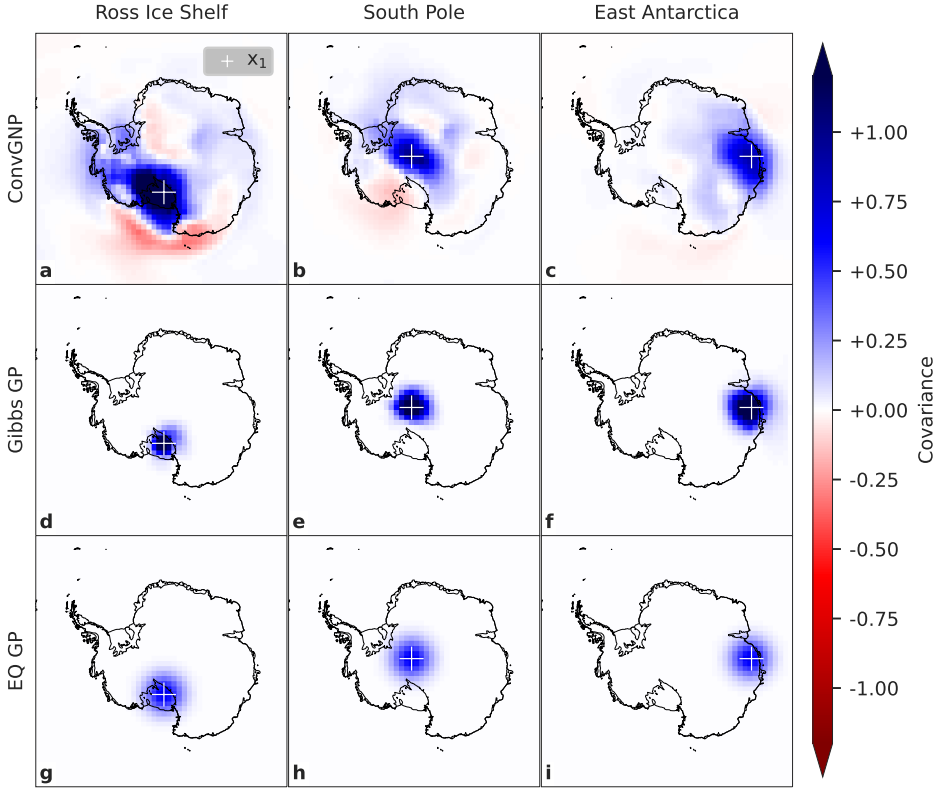

Figure D1. The ConvGNP learns spatially-varying covariance structure. *Heatmaps showing the prior covariance function, $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with $\boldsymbol{x}_1$ fixed at the white plus location and $\boldsymbol{x}_2$ varying over the grid. Plots are shown for three different $\boldsymbol{x}_1$-locations (the Ross Ice Shelf, the South Pole, and East Antarctica) and the three models (ConvGNP, Gibbs GP, and EQ GP). The ConvGNP's day of year input was the 1st of June.*

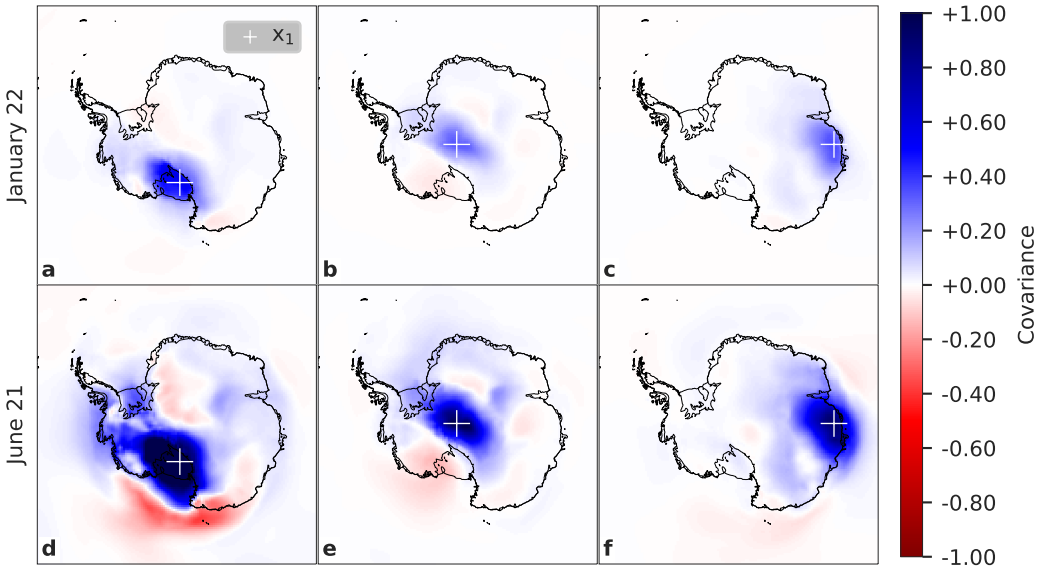

Figure D2. The ConvGNP learns seasonally-varying covariance. *Heatmaps showing the ConvGNP's prior covariance function $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$, as in Figure 3a-c, but for two times of the year: midsummer (Jan 22nd) and midwinter (June 21st). This shows that the ConvGNP has learned a prior covariance function with non-stationarity over day of year.*

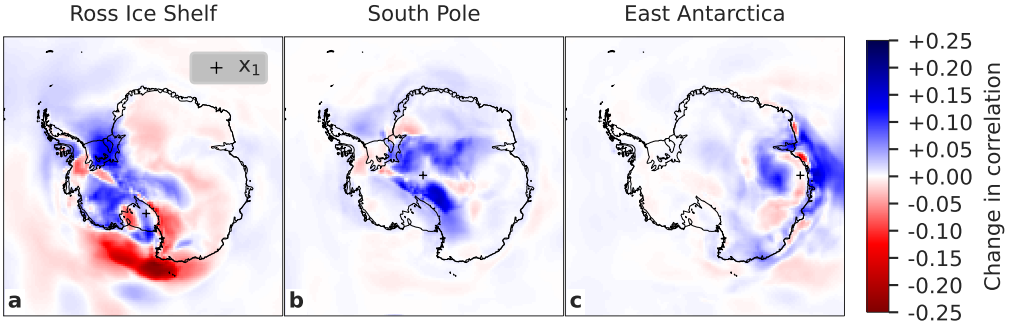

Figure D3. The ConvGNP learns seasonal changes in spatial correlations. *Change in prior correlation from the ConvGNP from the Austral Midsummer (Jan 22nd) to Midwinter (Jun 21st). The correlation ρ was computed from the covariance using $\rho = k(\boldsymbol{x}_1, \boldsymbol{x}_2)/\sqrt{k(\boldsymbol{x}_1, \boldsymbol{x}_1)k(\boldsymbol{x}_2, \boldsymbol{x}_2)}$, with $\boldsymbol{x}_1$ fixed at the black plus location and $\boldsymbol{x}_2$ varying over the grid. Plot shows summer minus winter.*

E. Additional test set results

In this section we provide further details on the models' test set performance.

E.1. Overall performance metrics

Table E1 shows the test set results averaged over $N_c \in \{0, 25, 50, \ldots 500\}$. The ConvGNP outperforms the three GP baselines with statistical significance across all three metrics (the normalised joint NLL, the marginal NLL, and the RMSE).

Table E1. *Performance on test tasks from the period* 2018–2019*, using* $N_c \in \{0, 25, 50, \ldots 500\}$*. Errors indicate standard errors. For each metric, lower is better. Significantly best results in bold.*

METRIC	CONVGNP	GIBBS GP	RQ GP	EQ GP
NORMALISED JOINT NLL	$\mathbf{-0.61}$±0.00	-0.42±0.00	-0.23±0.00	0.00±0.00
MARGINAL NLL	$\mathbf{-0.19}$±0.01	0.30±0.01	0.47±0.01	0.54±0.01
RMSE (°C)	**1.54**±0.01	1.84±0.01	1.63±0.01	1.72±0.01

E.2. Marginal calibration and sharpness

The calibration and sharpness of probabilistic prediction systems are key performance indicators (Gneiting et al., 2007). To assess these two quantities, we generated test tasks by subsampling the test dates (2018–2019) by a factor of 30 to obtain 25 dates. We then followed the same procedure to generate test tasks as in Appendix B.1–looping over $N_c \in \{0, 25, 50, \ldots 500\}$ with $N_t = 2000$, resulting in 50,000 marginal predictions per N_c for each model.

The calibration of a model's marginal distributions can be assessed using the probability integral transform (PIT). The PIT is defined as the cumulative distribution function (CDF) of the model's marginal distribution evaluated at the true observed value of a particular target point. If a model has perfect calibration, the histogram of PIT values is uniform (Gneiting et al., 2007). When aggregating across all test tasks (with varying N_c values), the ConvGNP's marginal distributions are much better calibrated than the GP baselines, coming closer to the ideal uniform distribution of PIT values (Figure E1).

The sharpness (i.e. degree of certainty) of probabilistic predictions must also be considered alongside calibration; a key goal for probabilistic prediction systems is to maximise sharpness subject to good calibration (Gneiting et al., 2007). We assess marginal distribution sharpness by plotting the standard deviation of the univariate Gaussian marginals against N_c. The ConvGNP makes substantially more confident predictions than the GP baselines across all values of N_c Figure E2. The GP baselines tend to make uninformative predictions with large uncertainty, explaining why their PIT values cluster around 0.5 (Figure E1b–c).

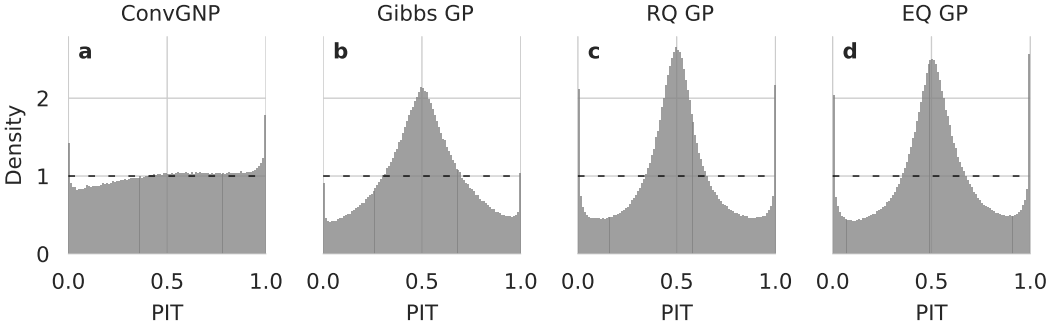

Figure E1. The ConvGNP produces the most well-calibrated marginal predictions. *Probability integral transform (PIT) histograms evaluated on 25 dates from the test years (2018–2019). The PIT is defined as the cumulative distribution function (CDF) of the model's marginal distribution evaluated at the true y-values. A black dashed line shows the ideal uniform distribution (corresponding to perfect calibration).*

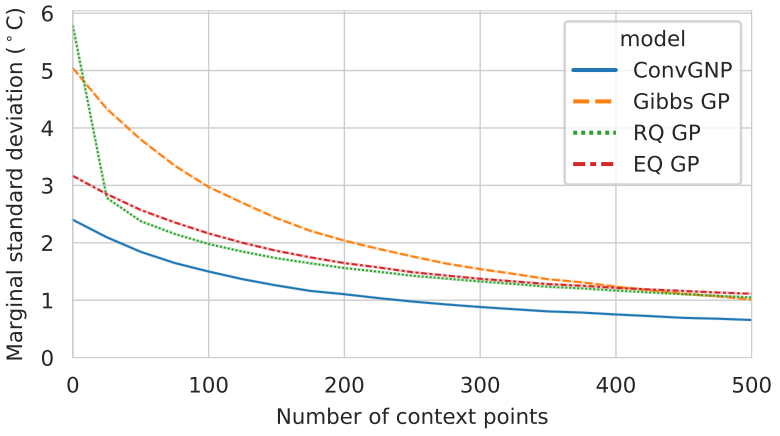

Figure E2. The ConvGNP produces the sharpest marginal predictions. *Mean standard deviation of the models' marginal Gaussian distributions versus number of context points, N_c. 50,000 standard deviation values were used per N_c, derived from 25 dates in the test years (2018–2019). Error bars are standard errors.*

F. Sensor placement acquisition functions

Here we expand upon and mathematically define each acquisition function used for sensor placement.

F.1. Model-based uncertainty reduction acquisition functions

JointMI: Expected reduction in joint entropy over targets after appending the query sensor to the context set C.

$$\alpha(\boldsymbol{x}_i^{(\mathrm{s})}, \tau) = H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau) - \mathop{\mathbb{E}}_{\pi(y_{\tau,i}^{(\mathrm{s})})}\left[H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, y_{\tau,i}^{(\mathrm{s})})\right] \tag{F.1}$$

$$= H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau) - \int \pi(y_{\tau,i}^{(\mathrm{s})}; C_\tau) H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, y_{\tau,i}^{(\mathrm{s})})\, \mathrm{d}y_{\tau,i}^{(\mathrm{s})} \tag{F.2}$$

$$\overset{(a)}{\approx} H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau) - H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, \bar{y}_{\tau,i}^{(\mathrm{s})}) \tag{F.3}$$

$$\overset{(b)}{\approx} c_\tau - H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, \bar{y}_{\tau,i}^{(\mathrm{s})}), \tag{F.4}$$

$$\approx c_\tau - \frac{1}{2}\log((2\pi e)^2|\boldsymbol{K}|), \tag{F.5}$$

$$\approx c_\tau - \frac{1}{2}\log|\boldsymbol{K}|, \tag{F.6}$$

where c_τ is a constant and $\boldsymbol{K}$ is the model's covariance matrix at the target locations after conditioning on the imputed query sensor observation, $(\boldsymbol{x}_i^{(\mathrm{s})}, \bar{y}_{\tau,i}^{(\mathrm{s})})$. In (a) we approximate the intractable expectation integral over the entropy term $H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, y_{\tau,i}^{(\mathrm{s})})$ with a simple substitution of the model's mean prediction $\bar{y}_{\tau,i}^{(\mathrm{s})}$ at query location $\boldsymbol{x}_i^{(\mathrm{s})}$. In (b) we use the fact that $H(\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau)$ depends only on τ and not $\boldsymbol{x}_i^{(\mathrm{s})}$. Thus, this placement criterion is equivalent to minimising the entropy over $\boldsymbol{y}_\tau^{(\mathrm{t})}|C_\tau, \boldsymbol{x}_i^{(\mathrm{s})}, \bar{y}_{\tau,i}^{(\mathrm{s})}$ by minimising the determinant of the covariance matrix $\boldsymbol{K}$.

This placement criterion may be hindered by approximating the expectation over the query observation $y_{\tau,i}^{(\mathrm{s})}$ by imputing with the model's mean prediction $\bar{y}_{\tau,i}^{(\mathrm{s})}$ in Equation F.3. Instead, a better scheme

would draw samples from the model's marginal distribution over $y^{(\mathrm{s})}_{\tau,i}$ to estimate the expectation using Monte Carlo sampling, which may better predict the actual reduction in entropy upon conditioning on the true observation. However, Monte Carlo sampling linearly increases the cost of evaluating the acquisition function. Future work should quantify the performance boost from switching to this sampling procedure.

MarginalMI: Expected reduction in marginal entropy over targets upon conditioning on the query sensor. Equivalent to that of Equation F.6, but setting the off-diagonal covariances in the model's output Gaussian distribution to zero when evaluating the entropy term $H(\boldsymbol{y}^{(\mathrm{t})}_{\tau}|C_\tau, \boldsymbol{x}^{(\mathrm{s})}_i, y^{(\mathrm{s})}_{\tau,i})$. Using the resulting independence of the individual marginal distributions after step (b) in Equation A.5 leads to:

$$\alpha(\boldsymbol{x}^{(\mathrm{s})}_i, \tau) \approx c_\tau - \sum_{j=1}^{N_t} H(y^{(\mathrm{t})}_{j,\tau}|C_\tau, \boldsymbol{x}^{(\mathrm{s})}_i, \bar{y}^{(\mathrm{s})}_{\tau,i}), \tag{F.7}$$

$$\approx c_\tau - \sum_{j=1}^{N_t} \frac{1}{2}\log(2\pi e \sigma^2_{\tau,j}), \tag{F.8}$$

$$\approx c_\tau - \sum_{j=1}^{N_t} \log(\sigma^2_{\tau,j}), \tag{F.9}$$

where $\sigma^2_{\tau,j}$ is the variance of the model's marginal Gaussian distribution of target point j at time τ after conditioning on the imputed query sensor observation, $(\boldsymbol{x}^{(\mathrm{s})}_i, \bar{y}^{(\mathrm{s})}_{\tau,i})$. In other words, the acquisition function is the negative sum of the marginal Gaussian entropies at each target location after adding the query sensor to the context set. After this acquisition function is averaged over time steps in Equation 2, the placement criterion amounts to minimising the mean log-variance over time and target locations.

DeltaVar: Expected reduction in mean marginal variance over targets upon conditioning on the query sensor. Following the same expectation approximation as `JointMI` and `MarginalMI` we arrive at:

$$\alpha(\boldsymbol{x}^{(\mathrm{s})}_i, \tau) \approx c_\tau - \frac{1}{N_t}\sum_{j=1}^{N_t} \sigma^2_{\tau,j}. \tag{F.10}$$

The main difference with the `MarginalMI` acquisition function is that `DeltaVar` minimises the absolute marginal variances rather than the log marginal variances.

F.1.1. Model-based uncertainty reduction acquisition functions from the sensor placement experiment

Figure F1 plots heatmaps of the three model-based acquisition functions (`JointMI`, `MarginalMI`, and `DeltaVar`) at the first greedy iteration for the ConvGNP, Gibbs GP, and EQ GP. Also shown are the $K = 10$ proposed sensor placements with the criterion of maximising these acquisition functions.

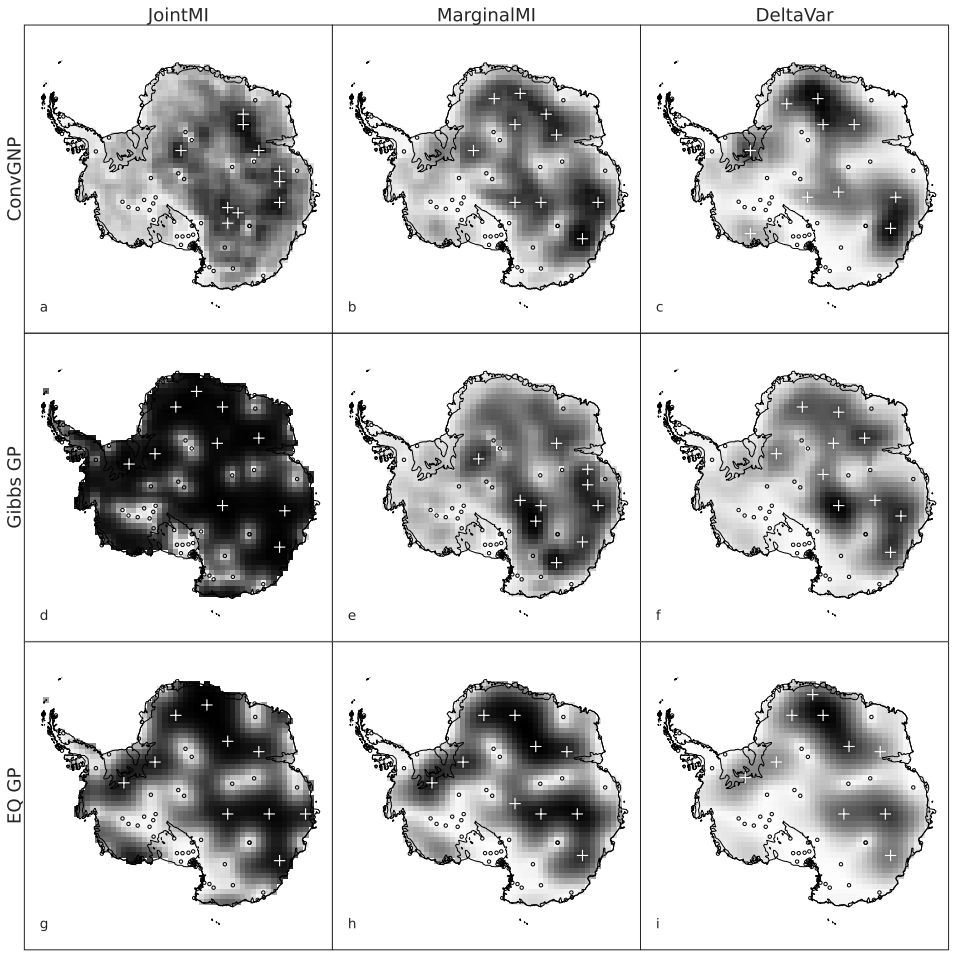

Figure F1. Acquisition functions and sensor placements for all three models. *Maps of acquisition function values $\alpha(\boldsymbol{x}_i^{(s)})$ for the initial $k = 1$ greedy iteration. The initial context set $\boldsymbol{X}^{(c)}$ is derived from real Antarctic station locations (indicated by black crosses). Running the sensor placement algorithm for $K = 10$ sensor placements results in the proposed sensor placements $\boldsymbol{X}^*$ (indicated by pluses).*

F.2. Baseline acquisition functions

Remoteness: Euclidean distance from the closest context point,

$$\alpha(\boldsymbol{x}_i^{(\mathrm{s})}, \tau) = \min\{||\boldsymbol{x}_i^{(\mathrm{s})} - \boldsymbol{x}_{\tau,1}^{(\mathrm{c})}||_2, ..., ||\boldsymbol{x}_i^{(\mathrm{s})} - \boldsymbol{x}_{\tau,N_c}^{(\mathrm{c})}||_2\}, \tag{F.11}$$

Random: Uniform at random in [0, 1],

$$\alpha(\boldsymbol{x}_i^{(\mathrm{s})}, \tau) = u_{\tau,i} \quad \text{where} \quad u_{\tau,i} \sim \mathrm{Unif}(0, 1). \tag{F.12}$$

Maximising this random acquisition function results in placements that are sampled uniformly from the search points $\boldsymbol{X}^{(\mathrm{s})}$.

F.3. Oracle acquisition functions

Let a performance metric γ take in the probability distribution over targets output by prediction map π and the ground truth target values $\boldsymbol{y}_\tau^{(\mathrm{t})}$. Considering only the 1D context set corresponding to

observations of the target variable for notational simplicity, the oracle acquisition functions are:

$$\alpha_{\text{oracle}}(\boldsymbol{x}_i^{(\text{s})}, \tau) = \gamma(\pi(\boldsymbol{y}_\tau^{(\text{t})}; \boldsymbol{X}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, \boldsymbol{X}_\tau^{(\text{t})}), \boldsymbol{y}_\tau^{(\text{t})}) - \gamma(\pi(\boldsymbol{y}_\tau^{(\text{t})}; \{\boldsymbol{X}_\tau^{(\text{c})}, \boldsymbol{x}_i^{(\text{s})}\}, \{\boldsymbol{y}_\tau^{(\text{c})}, y_{\tau,i}^{(\text{s})}\}, \boldsymbol{X}_\tau^{(\text{t})}), \boldsymbol{y}_\tau^{(\text{t})}), \tag{F.13}$$

$$= c_\tau - \gamma(\pi(\boldsymbol{y}_\tau^{(\text{t})}; \{\boldsymbol{X}_\tau^{(\text{c})}, \boldsymbol{x}_i^{(\text{s})}\}, \{\boldsymbol{y}_\tau^{(\text{c})}, y_{\tau,i}^{(\text{s})}\}, \boldsymbol{X}_\tau^{(\text{t})}), \boldsymbol{y}_\tau^{(\text{t})}), \tag{F.14}$$

which is the decrease in performance metric (assuming lower is better) induced by concatenating the ground truth observation $(\boldsymbol{x}_i^{(\text{s})}, y_{\tau,i}^{(\text{s})})$ to the context set. $\alpha_{\text{oracle}}(\boldsymbol{x}_i^{(\text{s})}, \tau)$ is then averaged over τ as in Equation (2) to obtain $\alpha_{\text{oracle}}(\boldsymbol{x}_i^{(\text{s})})$.

F.4. Comment on the dependence on context observations in the acquisition functions

The posterior covariance function of a vanilla GP depends only on the input locations $\boldsymbol{X}^{(\text{c})}$ of the context set, not the observed values $\boldsymbol{y}^{(\text{c})}$. Consequently, the `JointMI`, `MarginalMI`, and `DeltaVar` placement methods will depend only on the input locations. This behaviour is noted by MacKay 1992 for a Bayesian linear regression model with a Gaussian prior, which is a special case of a GP (Rasmussen, 2004). This could be seen as an inflexible limitation of GPs; they cannot augment their posterior correlation structure based on the y-values observed at the $\boldsymbol{x}$-locations. For example, if an extreme y-value is observed in the context set, a GP posterior cannot become more uncertain, which may be a desirable characteristic. By construction, the ConvGNP is a non-linear map from context sets to GPs, which means that the whole GP, including the covariance, can depend on every aspect of the context set, including the y-values. The ConvGNP's y-dependence necessitates the expectation integral over the unobserved query y-value in Equation F.1, as well as the averaging over multiple time steps for the uncertainty-based acquisition functions in Equation 2. Neither of these steps are necessary for the GP baselines since their covariance is independent of the y-values.

G. Oracle sensor placement results

Here we provide more detailed plots from the oracle acquisition function experiment described in Section 3.2.1. Heatmaps of the temporally-averaged $\alpha(\boldsymbol{x}_i^{(s)})$ acquisition functions for the non-oracle and oracle acquisition functions for the ConvGNP, Gibbs GP, and EQ GP are shown in Figure G2, Figure G3, and Figure G4, respectively. Figure G5, Figure G6, and Figure G7 show scatter plots of non-oracle acquisition functions against `OracleJointNLL`, `OracleMarginalNLL`, and `OracleRMSE`, respectively.

We repeat the Pearson correlation analysis of Figure 6 but using a correlation coefficient specifically suited to rankings, the Kendall rank correlation coefficient:

$$\kappa = \frac{1}{N_{\text{pairs}}} \sum_{i<j} \text{sgn}(\alpha_i - \alpha_j)\text{sgn}(\alpha_{\text{oracle},i} - \alpha_{\text{oracle},j}), \tag{G.1}$$

where $N_{\text{pairs}} = S(S-1)/2$ is the total number of pairs, $\alpha_i = \alpha(x_i^{(s)})$, and sgn is the sign function which is $+1$ if the argument is positive and -1 if the argument is negative.. This loops over all pairs of $(\alpha_i, \alpha_{\text{oracle},i})$ and $(\alpha_j, \alpha_{\text{oracle},j})$, checking whether the α values are ranked in the same order as the α_{oracle} values. If so, the pair is 'concordant' and contributes a $+1$ to the sum in Equation (G.1). Otherwise, it is 'discordant' and contributes a -1. Defining the total number of concordant pairs as N_{con}, Equation G.1 can be rewritten as:

$$\kappa = \frac{1}{N_{\text{pairs}}}\big(N_{\text{con}} - (N_{\text{pairs}} - N_{\text{con}})\big), \tag{G.2}$$

$$= 2 \times \frac{N_{\text{con}}}{N_{\text{pairs}}} - 1, \tag{G.3}$$

which we see as the fraction of pairs that are concordant, $N_{\text{con}}/N_{\text{pairs}}$, normalised to lie in $(-1, 1)$.

Identical rankings yield $\kappa = 1$, exactly opposite rankings yield $\kappa = -1$, and if the two rankings are independent the expected value of κ is zero. We computed κ in Python using the `scipy.stats.kendalltau` function. The results are shown in Figure G1.

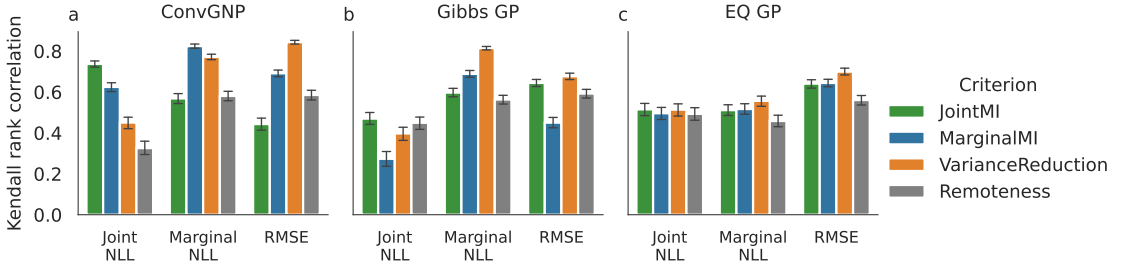

Figure G1. The ConvGNP can reliably rank the value of new observations. *Kendall rank correlation coefficient κ between model-based and oracle acquisition functions. Error bars indicate the 95% percentile interval over 5000 bootstrapped correlation values by resampling the 1365 pairs of points with replacement, measuring how spatially consistent κ is across space.*

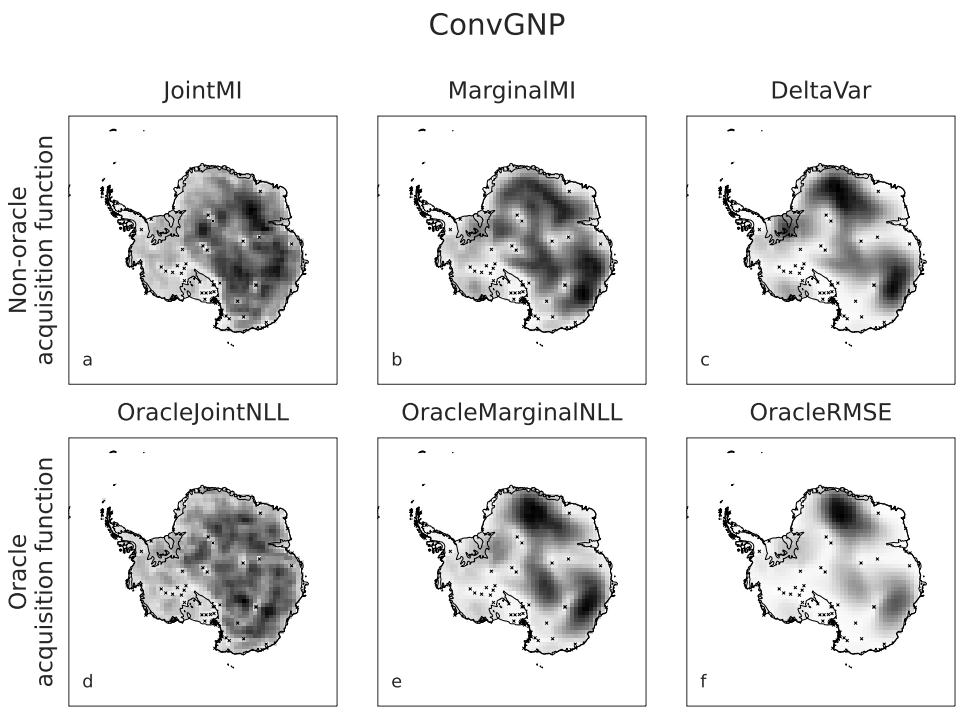

Figure G2. *Non-oracle and oracle acquisition functions for the ConvGNP.*

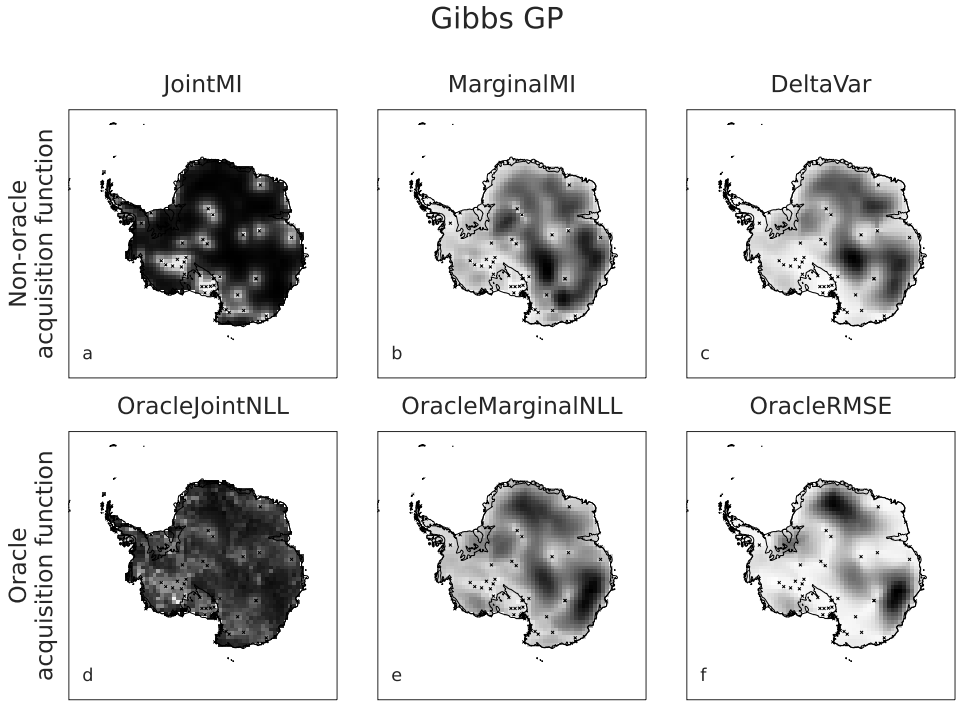

Figure G3. *Non-oracle and oracle acquisition functions for the Gibbs GP.*

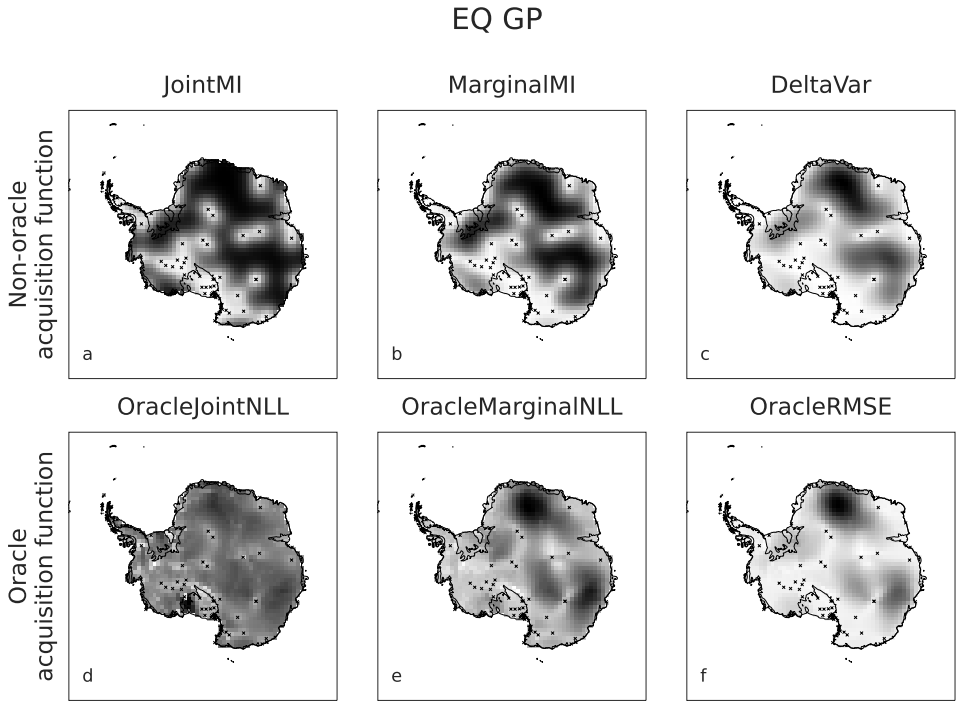

Figure G4. *Non-oracle and oracle acquisition functions for the EQ GP.*

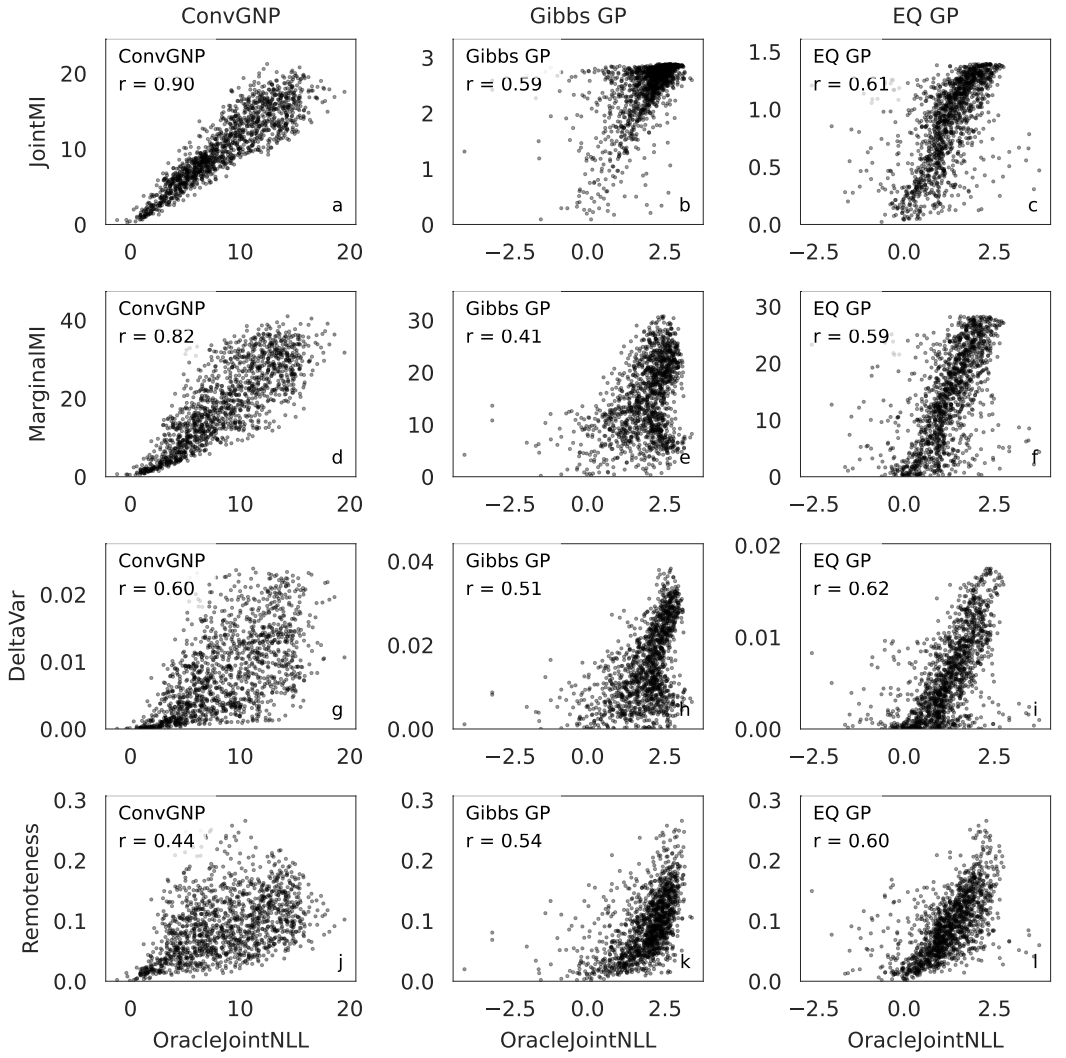

Figure G5. *Scatter plots and correlations between non-oracle acquisition functions and* `OracleJointNLL`.

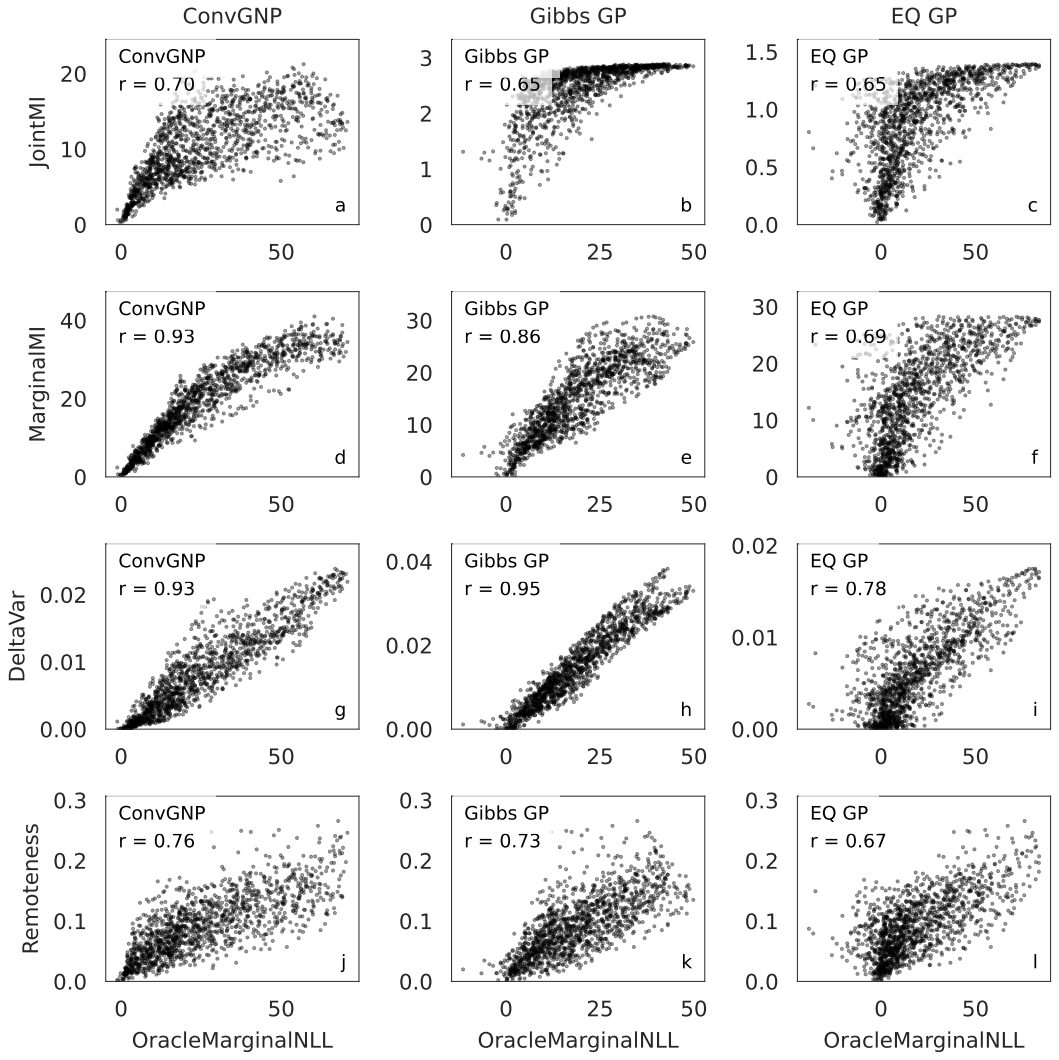

Figure G6. *Scatter plots and correlations between non-oracle acquisition functions and* `OracleMarginalNLL`.

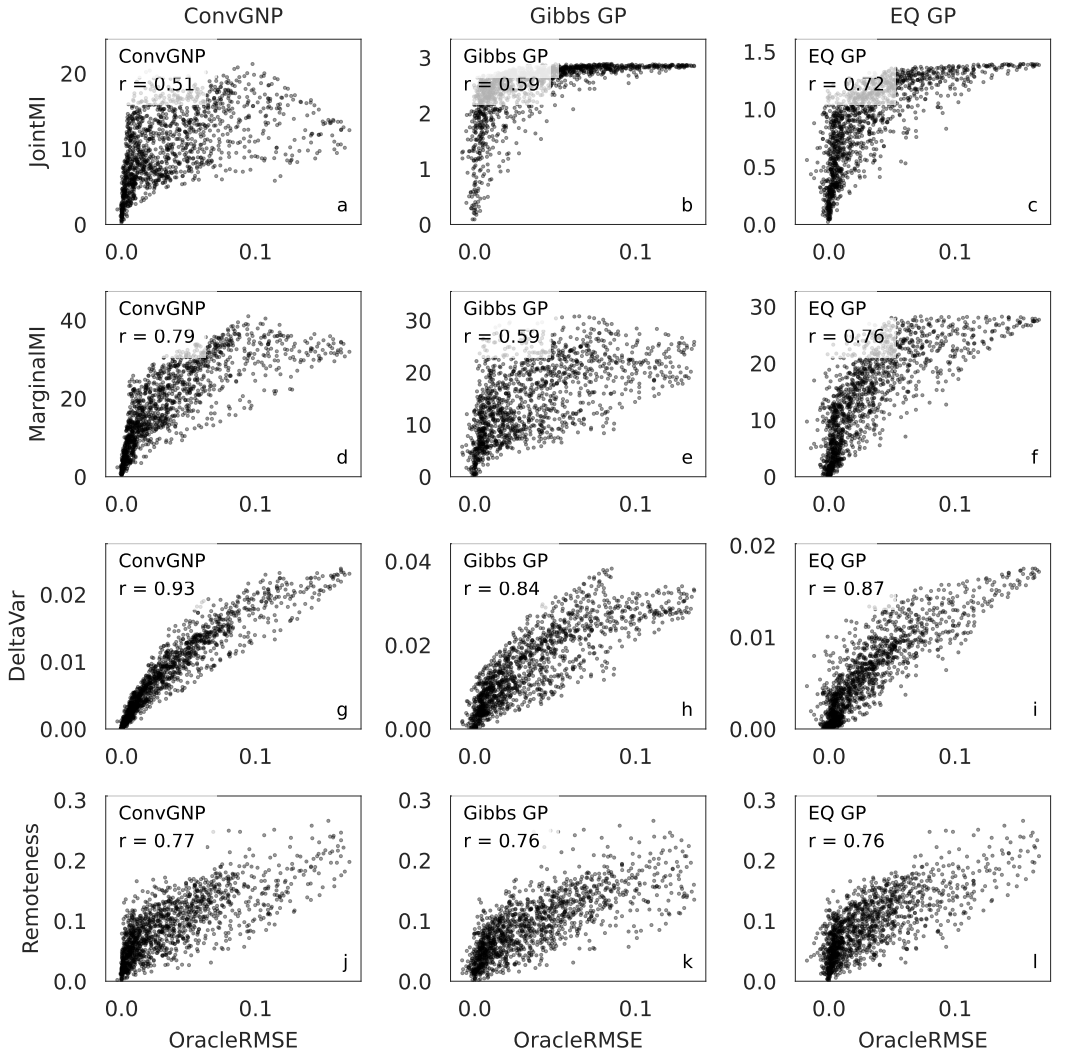

Figure G7. *Scatter plots and correlations between non-oracle acquisition functions and* `OracleRMSE`.

H. Sensor placement toy experiment details

Here we provide more details on the sensor placement toy experiment.

H.1. Experiment design choices

To emulate a non-uniform, real-world sensor network to be optimised, we initialise $\boldsymbol{X}_\tau^{(\mathrm{c})}$ at the locations of real Antarctic temperature station observations at $\tau = 2009/02/15$, and interpolate the gridded ERA5 temperature anomaly at $\boldsymbol{X}_\tau^{(\mathrm{c})}$ to compute $\boldsymbol{y}_\tau^{(\mathrm{c})}$.

The `JointMI`, `MarginalMI`, and `DeltaVar` criteria for the GP models and the `Remoteness` criterion only depend on the context set locations, $\boldsymbol{X}_\tau^{(\mathrm{c})}$, not the observed values $\boldsymbol{y}_\tau^{(\mathrm{t})}$ (Appendix F.4). Since we have a non-varying $\boldsymbol{X}_\tau^{(\mathrm{c})}$ in this toy experiment, the model-based acquisition functions do not depend on time τ for the GPs, and so the sensor placements for these criteria can be run using a single task. In contrast, each oracle acquisition function and the ConvGNP's model-based criteria *do* depend on the observed values $\boldsymbol{y}_\tau^{(\mathrm{t})}$. For these acquisition functions we compute the average $\alpha(\boldsymbol{x}_i^{(\mathrm{s})})$ values in Equation 2 using dates in 2014–2017, subsampled by a factor of 14, to yield $J = 105$ sensor placement search tasks. A regular spatial grid is used for the search space $\boldsymbol{X}^{(\mathrm{s})}$, with one query location every $100\,\mathrm{km}$. The $\boldsymbol{x}_i^{(\mathrm{s})}$ were masked out over the ocean to focus on land stations. The target locations $\boldsymbol{X}_\tau^{(\mathrm{t})}$ were defined on the same grid with points over ocean masked out to focus on predicting land surface temperature. These choices result in a search size and target set size of $S = N_t = 1,365$. Note, limiting the target set size to $N_t = 1,365$ was due to the cubic computational cost of the GP baselines–the ConvGNP could use a much denser target grid due to its linear scaling with number of target points. For the ConvGNP, sequentially computing one of the acquisition functions over these $J = 105$ dates and $S = 1,365$ search points (totalling $143,325$ forward passes) took roughly 3 hours on a 32 GB NVIDIA V100 GPU using TensorFlow's eager mode.

The proposed placements $\boldsymbol{X}^*$ were assessed by analysing model performance over 243 uniformly spaced dates in 2018–2019 (sampling every 3rd day). The sensor placement search period aligns with the model validation period, while the sensor placement analysis period aligns with the model test period.

H.2. Full sensor placement results

Figure F1 plots the $K = 10$ proposed sensor placements for each model and placement criterion. The full breakdown of the sensor placement results for each model, metric, and criterion is shown in Figure H1, using independent y-axes to highlight differences in placement criterion performance for a given model. However, this visually obscures two other differences: initial model performance and the scale of improvement with added stations. Plotting the results with the y-axes shared across models highlights these differences (Figure H2).

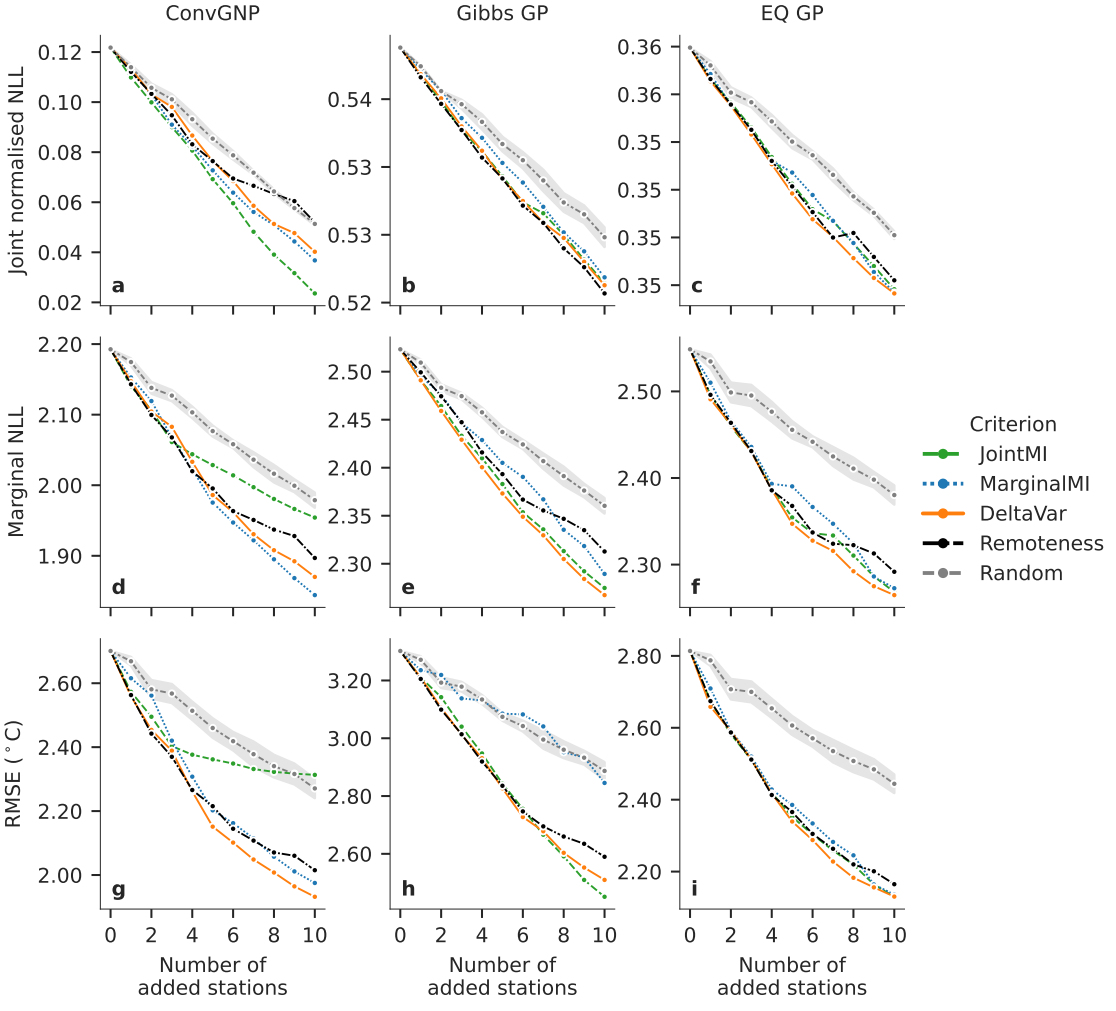

Figure H1. Sensor placement results. *Performance metrics on the 2018-2019 sensor placement ERA5 test data versus the number of stations revealed to the models. Results are averaged over tasks with targets defined on a regular grid over Antarctica. For the* `Random` *placement criterion, the confidence interval shows the standard error based on 5 random placements.* ***a-c****, joint normalised negative log-likelihood (NLL).* ***d-f****, mean marginal NLL.* ***g-i****, root mean squared error (RMSE).*

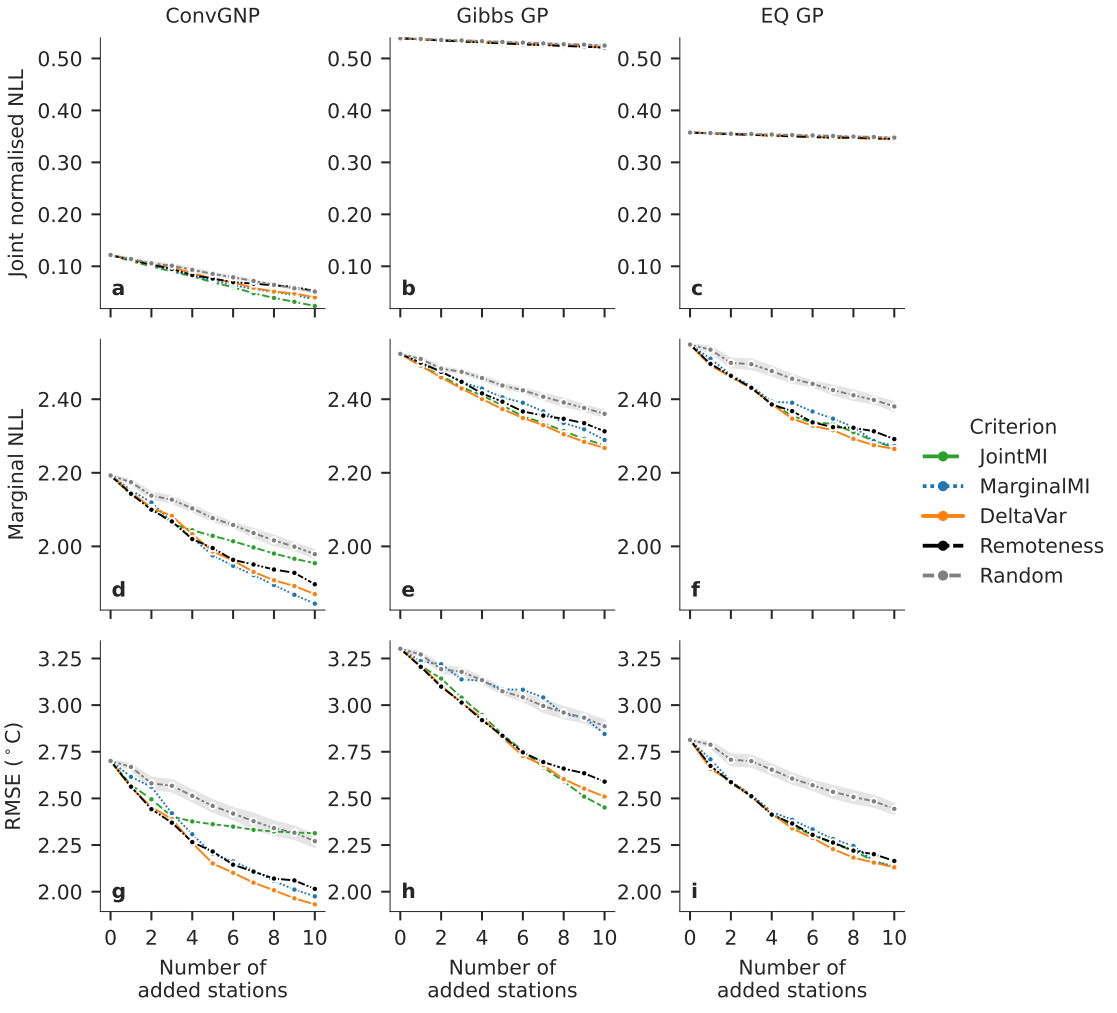

Figure H2. Sensor placement results with shared y-axes. *Performance metrics on the 2018-2019 sensor placement ERA5 test data versus the number of stations revealed to the models. Placements are revealed in the order of placement for each criterion. Results are averaged over tasks. For the* `Random` *placement criterion, the confidence interval shows the standard error based on 5 random placements.* ***a-c****, joint normalised negative log-likelihood (NLL).* ***d-f****, mean marginal NLL.* ***g-i****, root mean squared error (RMSE).*

I. Comparison of the ConvGNP with deep kernel learning

The ConvGNP is perhaps most comparable to deep kernel learning (DKL; Wilson et al. 2015). In DKL, neural networks parameterise a non-stationary covariance function and are trained to optimise a GP prior over the context data. Then, as with vanilla GPs, standard Bayes' rule conditioning is used with that prior to output posterior GP predictives. In contrast, the ConvGNP learns to directly output the GP predictive during training. This allows for outputting GPs that are not in the class of conditioned GP priors, which is much more flexible and aids modelling complex environmental data. However, since robust conditioning is not built in to the ConvGNP, it must learn appropriate conditioning mechanics from the data. This necessitates a novel training scheme where the model is provided with a range of context scenarios, expanding the training design space and likely making the ConvGNP more data-hungry than DKL.

The ConvGNP scales linearly with the number of context points due to the SetConv encoder and neural network architecture used to output the GP predictive. Predictions with the ConvGNP's GP predictive are made scalable by directly learning to output a low-rank approximation of the covariance, reducing the computational cost from cubic to linear. In contrast, DKL methods are by default cubic in the number of context and target points and must use approximate inference on the exact GP to make predictions scalable (Wilson et al., 2015), resulting in an unknown penalty to prediction quality (Wang et al., 2019). It is not obvious which is the best approach, although the out-of-the-box nature of the ConvGNP's scalability is convenient from a practitioner's point of view. Computational cost at inference time is important in the context of environmental applications because observations and target predictions locations may lie on dense grids.

DKL can also be deployed in a meta-learning fashion (Patacchiola et al., 2020), which mitigates the risk of overfitting that comes with heavily-parameterised covariance functions (Ober et al., 2021). A direct comparison between the meta-learning abilities of the ConvGNP and DKL has not been performed and would be a valuable addition to the literature.