Active Learning with Convolutional Gaussian Neural Processes for Environmental Sensor Placement

Tom R. Andersson
British Antarctic Survey
tomand@bas.ac.uk

Wessel P. Bruinsma
Microsoft Research AI4Science
wbruinsma@microsoft.com

Stratis Markou
University of Cambridge
em626@cam.ac.uk

James Requeima
University of Cambridge
Invenia Labs
jrr41@cam.ac.uk

Alejandro Coca-Castro
The Alan Turing Institute
acoca@turing.ac.uk

Anna Vaughan
University of Cambridge
av555@cam.ac.uk

Anna-Louise Ellis
Met Office
anna-louise.ellis@metoffice.gov.uk

Matthew Lazzara
University of Wisconsin-Madison
mattl@ssec.wisc.edu

Daniel C. Jones
British Antarctic Survey
dannes@bas.ac.uk

J. Scott Hosking
British Antarctic Survey
The Alan Turing Institute
jask@bas.ac.uk

Richard E. Turner
University of Cambridge
ret26@cam.ac.uk

Abstract

Deploying environmental measurement stations can be a costly and time-consuming procedure, especially in remote regions that are difficult to access, such as Antarctica. Therefore, it is crucial that sensors are placed as efficiently as possible, maximising the informativeness of their measurements. This can be tackled by fitting a probabilistic model to existing data and identifying placements that would maximally reduce the model's uncertainty. The models most widely used for this purpose are Gaussian processes (GPs; Williams and Rasmussen, 2006). However, designing a GP covariance which captures the complex behaviour of non-stationary spatiotemporal data is a difficult task. Further, the computational cost of GPs makes them challenging to scale to large environmental datasets. In this work, we explore using a convolutional Gaussian neural process (ConvGNP; Bruinsma et al., 2021; Markou et al., 2022) to address these issues. A ConvGNP is a *meta-learning model* that uses neural networks to parameterise a GP predictive. Our model is data-driven, flexible, efficient, and permits multiple input predictors of gridded or scattered modalities. Using simulated surface air temperature fields over Antarctica as ground truth, we show that a ConvGNP significantly outperforms a non-stationary GP baseline in terms of predictive performance. We then use the ConvGNP in an Antarctic sensor placement toy experiment, yielding promising results.

1 Introduction

The problem of sensor placement can be posed as follows: *given an initial set of observations of some system, where should the next K observations be taken?* In environmental sciences, the goal is to better understand and predict a variable across time and space. Machine learning methods

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

can be used to tackle this problem in a data-driven fashion using techniques from active learning. In active learning, the first step is to fit a *probabilistic model* to noisy observations of an unknown function $f(x)$, specifying a predictive distribution over possible functions, and identify new observation locations that are informative for the model.

The above task is extremely challenging with complex spatiotemporal variables. For instance, atmospheric surface temperature will be highly non-stationary in space (e.g. a mountainous region versus a plain) and time (e.g. summer versus winter). Furthermore, temperature depends heavily on other predictor variables, which may be observed at point-based locations (such as winds) or on a dense grid, such as surface elevation and sea ice coverage.

GPs are the go-to models for sensor placement and the related task of Bayesian optimisation[1] in the large majority of cases (Krause et al., 2008; Shahriari et al., 2016) due to their flexibility, ease of implementation, and closed-form posteriors. However, with dynamic environmental data, it is difficult to specify GP mean and covariance functions that faithfully capture the complex behaviour of the data. Furthermore, vanilla GPs cannot leverage other predictor variables (such as satellite data), which may be crucial for the prediction task.

In this work, using simulated daily-average Antarctic surface temperature as a motivating example, we demonstrate the ability to address the above problems using a convolutional Gaussian neural process (ConvGNP; Markou et al., 2022), a model which has recently shown significant promise in modelling environmental data (Vaughan et al., 2021; Markou et al., 2022).

1.1 Problem setup and meta-learning

Suppose the data lies on a regular time grid, indexed by τ. For each τ, we can consider each time slice to contain (1) a *context set* $(\boldsymbol{x}^{(\mathrm{c})}, \boldsymbol{y}^{(\mathrm{c})})$ where $\boldsymbol{x}^{(\mathrm{c})}$ are the input observation locations and $\boldsymbol{y}^{(\mathrm{c})}$ are the output observations, and (2) a *target set* $(\boldsymbol{x}^{(\mathrm{t})}, \boldsymbol{y}^{(\mathrm{t})})$. The context and target sets are aggregated into a *task* $\mathcal{D}_\tau = (\boldsymbol{x}_\tau^{(\mathrm{c})}, \boldsymbol{y}_\tau^{(\mathrm{c})}, \boldsymbol{x}_\tau^{(\mathrm{t})}, \boldsymbol{y}_\tau^{(\mathrm{t})})$. The goal is to build a model that takes the context set as input and makes probabilistic predictions for the target outputs $\boldsymbol{y}_\tau^{(\mathrm{t})}$ given the target inputs.

1.2 ConvGNP model

Neural processes (NPs; Garnelo et al., 2018a,b) are a recent class of meta-learning models which aim to combine the modelling flexibility of neural networks with the prediction uncertainties of Gaussian processes. An NP can be framed as a *prediction map* π (Foong et al., 2020), which is a function that directly maps from the context set to a predictive distribution over the corresponding target outputs $\boldsymbol{y}^{(\mathrm{t})}$. The ConvGNP is a particular NP variant that outputs a GP predictive:

$$\pi(\boldsymbol{y}^{(\mathrm{t})}; \boldsymbol{x}^{(\mathrm{c})}, \boldsymbol{y}^{(\mathrm{c})}, \boldsymbol{x}^{(\mathrm{t})}) = \mathcal{N}(\boldsymbol{y}^{(\mathrm{t})}; \boldsymbol{m}, \boldsymbol{K}), \tag{1}$$

where r, a CNN, produces an *encoding* of the context set with $\boldsymbol{r} = r(\boldsymbol{x}^{(\mathrm{c})}, \boldsymbol{y}^{(\mathrm{c})})$, and multilayer perceptrons f, g parameterise the mean and covariance with $\boldsymbol{m} = f(\boldsymbol{x}^{(\mathrm{t})}, \boldsymbol{r})$ and $\boldsymbol{K}_{ij} = g(\boldsymbol{x}_i^{(\mathrm{t})}, \boldsymbol{r})^{\mathrm{T}} g(\boldsymbol{x}_j^{(\mathrm{t})}, \boldsymbol{r})$. This formulation allows training the ConvGNP with a simple maximum likelihood procedure, learning to output arbitrary mean and covariance functions with a single forward pass, rather than explicitly specifying a prior over the data (as with vanilla GPs). Further, designing the covariance to be explicitly low-rank enables fast predictions, with a cost that is linear in the number of target points. This out-of-the-box scalability allows the model to run over tens of thousands of target points, permitting the placement algorithm to scale to vast regions such as Antarctica. We refer the reader to Appendix B for further model details, including an architecture schematic (Fig. B1).

1.3 The dataset

We use a reanalysis of daily-averaged atmospheric surface temperature over Antarctica as ground truth. Reanalysis data are produced by fitting a climate model simulation to observations (Gettelman et al., 2022), which capture the complex spatiotemporal dynamics of the Earth system on a regular grid. We further provide the ConvGNP with another context set of gridded elevation, a land mask, and space/time coordinate variables to enable learning non-stationarities (Appendix B.4). Real Antarctic station locations are used for the sensor placement experiment and generation of Figure 1. For details on the data sources and processing stages see Appendix A.

[1] Bayesian optimisation differs slightly from sensor placement in that the task is to optimise a black-box function f rather than reduce uncertainty about f.

2 Results

In this section we perform two experiments. First, we compare the ConvGNP with a stationary and non-stationary GP benchmark on a temperature regression dataset. Second, we perform sensor placement with a range of placement criteria, analysing the placement quality against the ERA5 ground truth data.

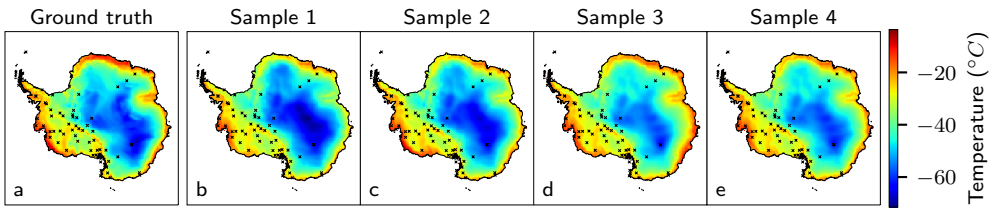

Figure 1: **a**, ERA5 temperature on $\tau = 2018/6/25$; **b-e**, ConvGNP samples. Crosses denote $\boldsymbol{x}_\tau^{(\mathrm{c})}$.

2.1 Performance on unseen data

Figure 1 compares ground truth ERA5 temperature with ConvGNP samples after running the model with $(\boldsymbol{x}_\tau^{(\mathrm{c})}, \boldsymbol{y}_\tau^{(\mathrm{c})})$ interpolated at real Antarctic temperature station locations at time τ for an unseen test date. The samples closely interpolate the context observations at station locations. Away from the stations, over remote and sparsely monitored regions, they display significant variation with interesting non-stationary spatial correlation structure.

ConvGNP performance metrics are reported in Table 1. We use two vanilla GP benchmarks: one with a non-stationary Gibbs kernel where the length scale varies over input space (Fig. C1), and one with a stationary exponentiated quadratic (EQ) kernel (Appendix C). The ConvGNP significantly outperforms both GP benchmarks in terms of negative log likelihood (NLL) and mean absolute error (MAE). See Appendix D for discussion on the differences between these three models.

Table 1: Test results on the temperature regression task on unseen data from the period 2018–2019. Errors indicate standard errors. Significantly best results in bold.

Metric	ConvGNP	Gibbs GP	EQ GP
Normalised NLL	$\mathbf{-1.76}_{\pm 0.03}$	$-1.15_{\pm 0.04}$	$-0.72_{\pm 0.01}$
MAE (°C)	$\mathbf{0.93}_{\pm 0.03}$	$1.34_{\pm 0.04}$	$2.10_{\pm 0.06}$

2.2 Sensor placement toy experiment

Following previous works (Krause et al., 2008), we pose sensor placement as a discrete optimisation problem using a greedy algorithm with a placement criterion to propose sensor placements one at a time. We choose $K = 10$ sensor placements $\boldsymbol{x}^*$ out of a set of $S = 863$ possible placement locations $\boldsymbol{x}^{(\mathrm{s})}$ on a regular grid over Antarctica. For each placement criterion, we compute the acquisition function at each time index τ and discrete query location $x_i^{(\mathrm{s})}$, returning a *ranking score*. The final ranking score at $x_i^{(\mathrm{s})}$ is the mean over time:

$$\alpha_{\text{avg}}(x_i^{(\mathrm{s})}) = \frac{1}{T} \sum_{\tau=1}^{T} \alpha(x_i^{(\mathrm{s})}, \tau). \tag{2}$$

In a greedy iteration, we compute $\alpha_{\text{avg}}(x_i^{(\mathrm{s})})$ for $i = (1, \ldots, S)$ and select the i^* corresponding to the maximum ranking score. We then append the corresponding input $x_{i^*}^{(\mathrm{s})}$ to the context set $\boldsymbol{x}_\tau^{(\mathrm{c})} \to \{\boldsymbol{x}_\tau^{(\mathrm{c})}, x_{i^*}^{(\mathrm{s})}\}$ and reveal ground truth to the model with $\boldsymbol{y}_\tau^{(\mathrm{c})} \to \{\boldsymbol{y}_\tau^{(\mathrm{c})}, \hat{y}_{\tau,i^*}^{(\mathrm{s})}\}$. This process is repeated until K placements have been made. See Appendix E for further experiment details.

2.2.1 Acquisition functions

Here we describe the acquisition functions used for each placement criterion. Mathematical definitions are provided in Appendix E.3.

`OracleRMSE`: decrease in RMSE after conditioning the model on the ground truth observation, $\hat{y}_{\tau,i}^{(\mathrm{s})}$ at $x_{\tau,i}^{(\mathrm{s})}$, using the mean of the predictive distribution as the model's point-based estimate.

`MaxMI`: mutual information (MI) between $\boldsymbol{y}_\tau^{(\mathrm{t})}$ and a query sensor observation $(x_i^{(\mathrm{s})}, \bar{y}_{\tau,i}^{(\mathrm{s})})$, where $\bar{y}_{\tau,i}^{(\mathrm{s})}$ is the model's mean prediction at $x_i^{(\mathrm{s})}$ for time τ. The MI has been used frequently in past work to identify locations that maximally reduce the model's predictive uncertainty (Lindley, 1956; MacKay, 1992; Krause et al., 2008; Schmidt et al., 2019).

`MaxStddev`: marginal standard deviation of the predictive distribution at query location $x_i^{(\mathrm{s})}$.

`MaxDist`: distance to the closest sensor. This is a simple heuristic which proposes placements as far away from the observed data as possible.

`Random`: uniform white noise function (equivalent to placing sensors randomly). The performance of this criterion reflects the average benefit of adding new observations to the model.

2.2.2 Toy experiment results

We find that the `MaxMI` criterion proposes very similar sensor placements to `OracleRMSE` (Fig. 2f,g, Fig. E1), despite not having access to ground truth (see Appendix E.4 for a detailed comparison of the placements from `MaxMI` and `OracleRMSE`). Furthermore, the initial $k = 1$ greedy iteration $\alpha_{\mathrm{avg}}(x_i^{(\mathrm{s})})$ field of `MaxMI` shows similar structure to `OracleRMSE`, with three clear modes of ranking score in similar locations (Fig. 2a,b). This is not picked up by the simpler `MaxStddev` criterion that does not leverage output dependencies in the predictive (Fig. 2c).

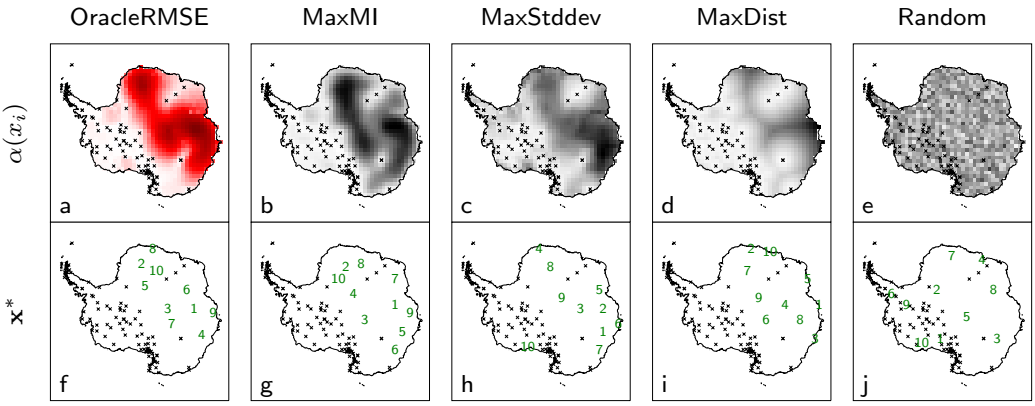

Figure 2: **a-e**, Maps of acquisition function values $\alpha_{\mathrm{avg}}(x_i^{(\mathrm{s})})$ for the first greedy iteration. **f-j**, Sensor placements indicated with green integers for the greedy iteration they were chosen. Black crosses denote the initial $\boldsymbol{x}_\tau^{(\mathrm{c})}$ (shown for $\tau = 2018/1/1$).

Figure 3 shows the quality of the placements based on the model's ability to predict simulated temperature over Antarctica as observations are added at the proposed locations. The difference in RMSE is small between `MaxMI` and the simpler `MaxStddev` and `MaxDist` criteria by $k = 10$. However, the `MaxMI` criterion increases the normalised joint log density of held-out points the fastest (even faster than `OracleRMSE`, which doesn't optimise for the log density). This shows that `MaxMI` identifies highly informative locations for improving the model's probabilistic predictions.

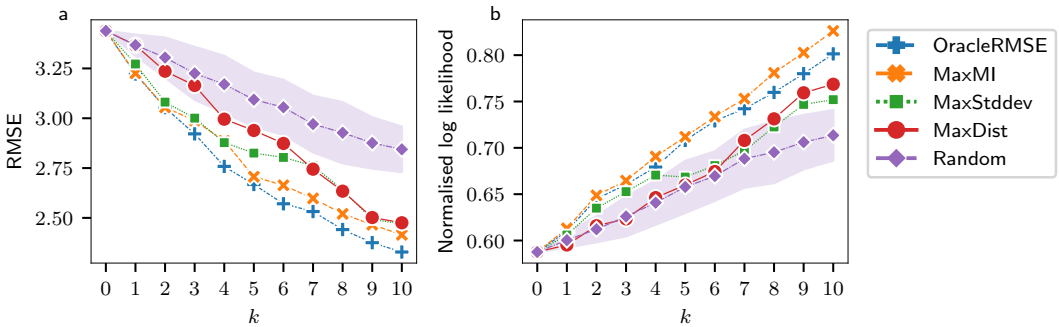

Figure 3: Sensor placement results averaged over 2019 data. **a**, RMSE. **b**, log likelihood normalised by the number of targets. The `Random` error bar shows the standard deviation over 10 random seeds.

3 Discussion

Using simulated Antarctic surface temperature data, we have shown that ConvGNPs have a range of properties that enable modelling complex spatiotemporal climate variables. These properties include: ability to learn arbitrary covariance and mean functions from raw data; meta-learning across multiple time steps; ingestion of multiple predictors of various modalities (gridded and off-grid); and out-of-the-box scalability. Notably, the ConvGNP makes significantly better probabilistic predictions than a non-stationary GP benchmark on unseen test data. In a sensor placement toy experiment using the simulated temperature fields, we show that these modelling benefits enable highly informative sensor placements by leveraging a mutual information placement criterion.

The results of our toy experiments are promising but preliminary: here the ConvGNP was trained on a large dataset with observations across the entire input space, but real observations may be sparse and constrained to finite locations. Future work will test modelling scenarios that leverage both observational data and simulated data. We will explore the fusion of in-situ station observations with satellite data to fill missing gaps in station coverage and assess uncertainty quantification in this setting.

By leveraging modelling advances that can better tackle the complexities of environmental data, our work goes beyond the traditional GP-based geostatistics Kriging approach (Cressie, 1993) and brings us closer to answering the demands of modern model-data fusion methods (Gettelman et al., 2022) and digital twins of the natural environment (Blair, 2021).

Data and Code

Code to reproduce the results in this paper will be released in the near future. The ConvGNP was implemented using the Python package *neuralprocesses*, available at `https://github.com/wesselb/neuralprocesses`.

Acknowledgements

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 Research Council (studentship number 10436152). RET is supported by Google, Amazon, ARM, Improbable and EPSRC grant EP/T005386/1. We thank Markus Kaiser and the anonymous reviewer for feedback on the paper.

References

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

Bruinsma, W. P., Requeima, J., Foong, A. Y., Gordon, J., and Turner, R. E. (2021). The gaussian neural process. *arXiv preprint arXiv:2101.03606*.

Cressie, N. (1993). *Statistics for Spatial Data*. John Wiley & Sons. Google-Books-ID: MzN_BwAAQBAJ.

Foong, A. Y. K., Bruinsma, W. P., Gordon, J., Dubois, Y., Requeima, J., and Turner, R. E. (2020). Meta-Learning Stationary Stochastic Process Prediction with Convolutional Neural Processes. arXiv:2007.01332 [cs, stat].

Garnelo, M., Rosenbaum, D., Maddison, C. J., Ramalho, T., Saxton, D., Shanahan, M., Teh, Y. W., Rezende, D. J., and Eslami, S. M. A. (2018a). Conditional neural processes. Number: arXiv:1807.01613.

Garnelo, M., Schwarz, J., Rosenbaum, D., Viola, F., Rezende, D. J., Eslami, S. M. A., and Teh, Y. W. (2018b). Neural Processes. 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. Publisher: American Association for the Advancement of Science.

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

Gordon, J., Bruinsma, W. P., Foong, A. Y. K., Requeima, J., Dubois, Y., and Turner, R. E. (2020). Convolutional Conditional Neural Processes. arXiv:1910.13556 [cs, stat].

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. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/qj.3803.

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.

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

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

Markou, S., Requeima, J., Bruinsma, W., and Turner, R. (2021). Efficient Gaussian Neural Processes for Regression. Number: arXiv:2108.09676 arXiv:2108.09676 [cs, stat].

Markou, S., Requeima, J., Bruinsma, W. P., Vaughan, A., and Turner, R. E. (2022). Practical Conditional Neural Processes Via Tractable Dependent Predictions. arXiv:2203.08775 [cs, stat].

Morlighem, M. (2020). Measures bedmachine antarctica, version 2.

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

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. 12(6). Institution: Lawrence Livermore National Lab. (LLNL), Livermore, CA (United States) Number: LLNL-JRNL-753008 Publisher: Wiley.

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. Conference Name: Proceedings of the IEEE.

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. Publisher: Copernicus GmbH.

Williams, C. K. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Appendix

A Data considerations

In this section we describe details behind 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). The Antarctic temperature observations from crewed and automatic weather stations were downloaded from ftp.bas.ac.uk/src/.

A.2 Data preprocessing

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

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 x were normalised from metres to take values in $[-1, 1]$.

B ConvGNP

Here we provide details on the ConvGNP training procedure and architecture. A high-level schematic of the ConvGNP forward-pass is shown in Figure B1. We refer the reader to (Markou et al., 2021) for further model details.

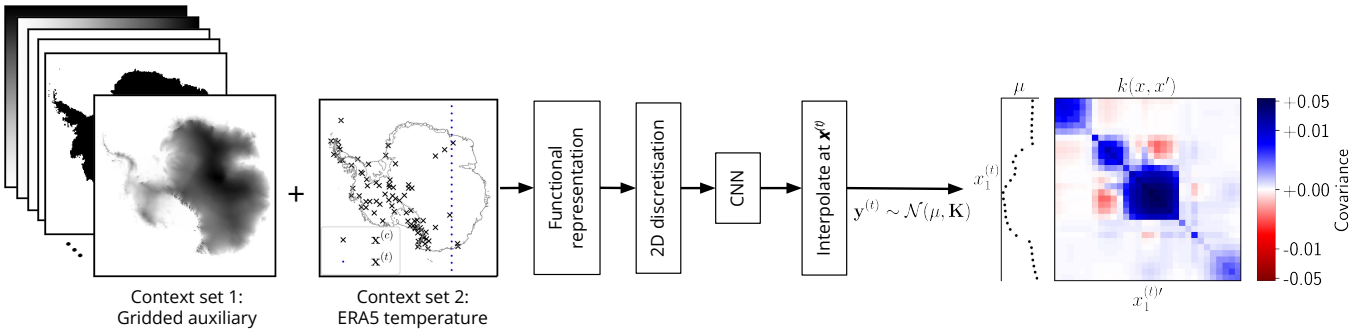

Figure B1: Schematic depicting the ConvGNP forward pass. As input, the model receives two context sets: 6 gridded auxiliary fields, and temperature observations. These context sets are converted to a functional representation (with a density channel for each context set indicating where data is observed), followed by a 2D discretisation procedure (Gordon et al., 2020). A CNN takes these channels as input and outputs a gridded representation, r, which is interpolated at target points $\boldsymbol{x}^{(t)}$ and used to parameterise the mean and covariance of the multivariate Gaussian distribution over $\boldsymbol{y}^{(t)}$. The output mean vector μ and covariance matrix $\boldsymbol{K}$ are shown after running the model with a vertical line of target points (overlaid on context set 2).

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 context points $N_c \sim \text{Unif}\{1, 2, \ldots 500\}$. Allowing for randomness in N_c encourages the model to learn to deal with both data-sparse and data-rich scenarios. We held the number of target points N_t fixed at a fairly high value of 2,000 to provide sufficient signal for learning the covariance structure of the data while not incurring the cost of a very large N_t.

Next, the input locations $\boldsymbol{x}_\tau^{(\mathrm{c})}$ and $\boldsymbol{x}_\tau^{(\mathrm{t})}$ were sampled uniformly at random across the entire 280×280 input space.

For the training dates, the random seed used for generating $\mathcal{D}_\tau$ is changed every epoch, allowing for an infinitely growing simulated 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 shown in Table 1, we increase the lower bound on N_c from 1 to 50 to allow the GP benchmarks more context data for deviating from their constant prior mean behaviour.

B.2 Antarctic surface temperature ConvGNP training procedure

The model was trained on data from 1980-2013. An Adam optimiser was used with a learning rate of 5×10^{-5} and a negative log likelihood loss function. Gradients with respect to the loss were averaged over batches of two datasets.

Validation data from 2014-2017 was used for checkpointing the model weights using the normalised loss (i.e. loss normalised by number of target points). 2018-2019 data was reserved for the test set.

B.3 ConvGNP architecture

For the ConvGNP model we use the same architecture as described in Markou et al. (2021), except for a few modifications. The UNet 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): $16 \xrightarrow{\text{d.s.}} 32 \xrightarrow{\text{d.s.}} 64 \xrightarrow{\text{d.s.}} 128 \xrightarrow{\text{u.s.}} 64 \xrightarrow{\text{u.s.}} 32 \xrightarrow{\text{u.s.}} 16$. We use 128 basis functions for the covariance parameterisation g. In total, the ConvGNP has 620,853 learnable parameters. 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 200 points-per-unit (i.e., a 1x1 square of input space contains 200x200 internal discretisation points).

The above choices for the UNet filter size and internal discretisation density results in a receptive field of almost 790 km. In other words, context observations can influence target predictions in the Gaussian predictive distribution up to 395 km in either direction of the x_1- or 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. The second contains a set of 6 gridded auxiliary and metadata variables. These are: elevation, land mask, cos(day of year), sin(day of year), x_1, and x_2. The cos(day of year) and sin(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 CNN's translation equivariance property.

C Gaussian Process benchmarks

Here we provide details on the GP benchmarks included in Table 1.

C.1 Gibbs kernel

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:

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \prod_{i=1}^{2} \left(\frac{2 l_i(\boldsymbol{x}) l_i(\boldsymbol{x}')}{l_i(\boldsymbol{x})^2 + l_i(\boldsymbol{x}')^2} \right)^{1/2} \exp\left(- \sum_{i=1}^{2} \frac{(x_i - x_i')^2}{l_i(\boldsymbol{x})^2 + l_i(\boldsymbol{x}')^2} \right), \tag{3}$$

where 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,

$$l_i(\boldsymbol{x}) = \sum_{m=1}^{M} \theta_{i,m} \exp\left(- \frac{(\boldsymbol{x} - \boldsymbol{\mu}_m)^2}{2\lambda^2} \right), \tag{4}$$

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 $\boldsymbol{\mu}_m$ on a 100 x 100 grid spanning the input space. The basis function length scale λ is kept fixed and equal to the spacing between basis functions.

We train the parameters $\{\theta_i, \sigma\}$, as well as the GP prior mean and noise variance, using gradient descent on the negative log marginal likelihood (NLML) using an Adam optimiser with learning rate 1×10^{-2} and a batch size of 5. We used 1980-2013 as a training period, subsampling by a factor of 7, amounting to 1,825 days in total, with the context and target locations generated randomly in each epoch as described in Appendix B.1. Training was halted after the NLML 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.

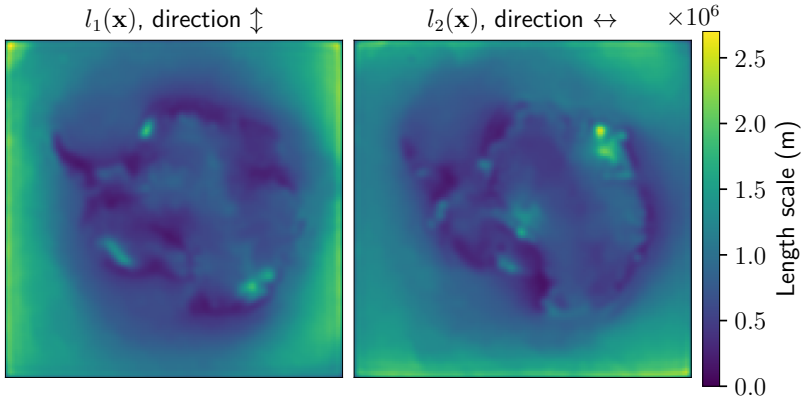

Figure C1: Learned Gibbs GP length scale functions, $l_1(\boldsymbol{x})$ and $l_2(\boldsymbol{x})$.

C.2 Exponentiated quadratic kernel

We also include a much more basic GP benchmark using a non-isotropic exponentiated quadratic (EQ) kernel, which has a stationary covariance function (unlike the Gibbs kernel).

We fit the EQ GP hyperparameters using the L-BFGS-B algorithm on random batches of 60 datasets in 1980-2013, with convergence occurring very rapidly.

D Comparison of covariance functions

Figure D1 illustrates the differences in the covariance behaviour for the ConvGNP, the Gibbs GP and the EQ GP, prior to conditioning on temperature observations. The ConvGNP leverages the gridded auxiliary fields (context set 1) to learn highly non-stationary spatial dependencies in surface temperature. For example, Figure D1a shows decorrelation over the Transantarctic Mountains, and Figure D1a-c all show sudden decorrelation over the coastline. Furthermore, the covariance is not constrained to be positive, unlike the GP benchmarks.

The Gibbs GP also learns non-stationary behaviour (Figure D1d-f), but the covariance function is constrained to the form in Equation (3), restricting the kind of behaviour it can model, unlike the ConvGNP which can learn arbitrary covariance functions.

The stationarity assumption of the EQ GP is evident in (Figure D1g-i), with the covariance behaviour being independent of input location.

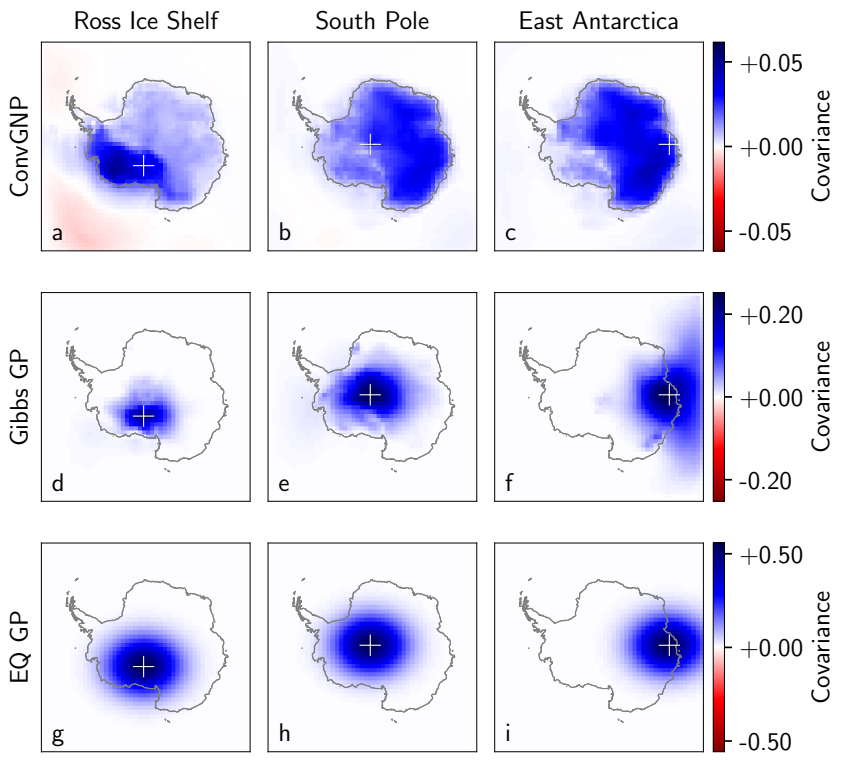

Figure D1: Covariance function heatmaps showing $k(\boldsymbol{x}_1, \boldsymbol{x}_2)$, with $\boldsymbol{x}_1$ fixed at the white plus location and $\boldsymbol{x}_2$ varying over the grid. Plots are shown for three different $\boldsymbol{x}_1$-locations (the Ross Ice Shelf, the South Pole, and East Antarctica) and the three models (ConvGNP, Gibbs GP, and EQ GP).

E Sensor placement toy experiment details

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

E.1 Experiment design choices

To emulate a non-uniform sensor network to be optimised, we initialise $\boldsymbol{x}_\tau^{(\mathrm{c})}$ at the locations of real Antarctic temperature station observations at time τ, and interpolate the gridded ERA5 temperature at $\boldsymbol{x}_\tau^{(\mathrm{c})}$ to compute $\boldsymbol{y}_\tau^{(\mathrm{c})}$.

We place $K = 10$ sensors using 52 uniformly spaced dates in 2018 for computing $\alpha_{\mathrm{avg}}(x_i^{(\mathrm{s})})$ in Equation (2) at each query location $x_i^{(\mathrm{s})}$. A regular spatial grid is used for the search space $\boldsymbol{x}^{(\mathrm{s})}$, with one query location every 125 km. The $x_i^{(\mathrm{s})}$ were masked out over ocean to focus on land stations. These choices result in a search size of $S = 863$. The proposed placements $\boldsymbol{x}^*$ were assessed by analysing model performance over 121 uniformly spaced dates in 2019. Note that both the sensor placement search and analysis periods are in the ConGNP's test years.

For the `OracleRMSE` and `MaxMI` placement criteria, as well as for the analysis of the placements, the target locations $\boldsymbol{x}_\tau^{(\mathrm{t})}$ were defined over the original 25 km ERA5 grid. Like the query placement locations $\boldsymbol{x}^{(\mathrm{s})}$, all target locations $\boldsymbol{x}_\tau^{(\mathrm{t})}$ were masked out over ocean to focus on predicting land surface temperature.

E.2 Imputed y-values after each greedy placement

We reveal the ground truth observation to the model at proposed placements $x_{i^*}^{(\mathrm{s})}$ after each greedy iteration: $\boldsymbol{y}_\tau^{(\mathrm{c})} \mapsto \{\boldsymbol{y}_\tau^{(\mathrm{c})}, \hat{y}_{\tau,i^*}^{(\mathrm{s})}\}$. In practice, this is not plausible for spatiotemporal data because it would require us to go back in time to observe the function at $x_{i^*}^{(\mathrm{s})}$. An alternative approach to

propose multiple placements using real observational data would be to approximate $\alpha_{\text{avg}}(x_i^{(\text{s})})$ by infilling with the model mean prediction at $x_{i^*}^{(\text{s})}$, e.g. $\boldsymbol{y}_{\boldsymbol{\tau}}^{(\text{c})} \mapsto \{\boldsymbol{y}_{\boldsymbol{\tau}}^{(\text{c})}, \bar{y}_{\tau,i^*}^{(\text{s})}\}$. Another option would be to draw MCMC samples from the model's distribution over $y_{\tau,i^*}^{(\text{s})}$. This would likely degrade the placements to some degree. Future work will quantify the impact of placing multiple sensors at a time.

E.3 Acquisition functions

Here we expand upon and mathematically define each acquisition function.

OracleRMSE:

$$\alpha(x_i^{(\text{s})}, \tau) = \text{RMSE}(\hat{\boldsymbol{y}}_\tau^{(\text{t})}, \mathbb{E}[\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}]) - \text{RMSE}(\hat{\boldsymbol{y}}_\tau^{(\text{t})}, \mathbb{E}[\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, \hat{y}_{\tau,i}^{(\text{s})}]), \tag{5}$$

where for a ConvGNP the expectation over target outputs is simply the mean parameter $\boldsymbol{m}$ output by the model in (1).

MaxMI:

$$\alpha(x_i^{(\text{s})}, \tau) = H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}) - \int H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, y_{\tau,i}^{(\text{s})}) \, \mathrm{d}y_{\tau,i}^{(\text{s})} \tag{6}$$

$$\overset{(a)}{\approx} H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}) - H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, \bar{y}_{\tau,i}^{(\text{s})}) \tag{7}$$

$$\overset{(b)}{\approx} c_\tau - H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, \bar{y}_{\tau,i}^{(\text{s})}), \tag{8}$$

where c_τ is a constant. In (a) we approximate the intractable expectation integral over the entropy term $H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, y_{\tau,i}^{(\text{s})})$ with a simple substitution of the model's mean prediction $\bar{y}_{\tau,i}^{(\text{s})}$ at query location $x_i^{(\text{s})}$. In (b) we use the fact that $H(\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})})$ depends only on τ and not $x_i^{(\text{s})}$. Thus, this placement criterion is equivalent to minimising the entropy over $\boldsymbol{y}_\tau^{(\text{t})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}, x_{\tau,i}^{(\text{s})}, \bar{y}_{\tau,i}^{(\text{s})}$.

Future work could improve the approximation in (a) using MCMC sampling over $y_{\tau,i}^{(\text{s})}$. However, this would bring a linear increase in the cost of evaluating the acquisition function.

Note, in the equation for the posterior covariance of a vanilla GP, there is only dependence on the input coordinates $\boldsymbol{x}^{(\text{c})}$ and $\boldsymbol{x}^{(\text{t})}$, not the observed values $\boldsymbol{y}^{(\text{c})}$. This could be seen as an inflexible limitation of GPs; they cannot augment the posterior entropy based on the values observed at the x-locations. For example, if an extreme $y-$value is observed in the context set, a GP posterior cannot become more uncertain or augment its correlation structure. A ConvGNP does not have this restriction, necessitating the expectation integral in (6).

MaxStddev:

$$\alpha(x_i^{(\text{s})}, \tau) = \sqrt{\text{Var}\big(y_{\tau,i}^{(\text{s})} | \boldsymbol{x}_\tau^{(\text{c})}, \boldsymbol{y}_\tau^{(\text{c})}\big)}, \tag{9}$$

where for a ConvGNP the variance over target outputs is simply the diagonal entries of the covariance matrix output by the model in (1).

Note, for a given τ, the $\alpha(x_i^{(\text{s})}, \tau)$ above can be computed over all $x_i^{(\text{s})}$ in parallel with a single forward pass of the model, so this criterion can be seen as a very computationally efficient benchmark that makes no use of the dependencies between target points.

MaxDist:

$$\alpha(x_i^{(\text{s})}, \tau) = \min\{||x_i^{(\text{s})} - x_{\tau,1}^{(\text{c})}||_2, ..., ||x_i^{(\text{s})} - x_{\tau,N_c}^{(\text{c})}||_2\}, \tag{10}$$

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

$$\alpha(x_i^{(\text{s})}, \tau) = u_{\tau,i} \quad \text{where} \quad u_{\tau,i} \sim \text{Unif}(0, 1). \tag{11}$$

E.4 Comparison of `OracleRMSE` and `MaxMI` placements

Figure E1 shows an expanded view of the toy experiment placements from `OracleRMSE` and `MaxMI`.

`MaxMI` proposes placements very close in x-space and ranking to the RMSE-optimal criterion of `OracleRMSE`. The first 3 placements are strikingly similar, appearing to correspond to the three peaks of high $\alpha_{\text{avg}}(x_i^{(s)})$ in Figure 2a and 2b. After this, the placements at iteration $k = 4$ and $k = 5$ highlight very similar regions of Antarctica but swapped between the two criteria. The `MaxMI` placement at $k = 6$ does not correspond clearly to any placement from `OracleRMSE`, but the placement at $k = 7$ is fairly close to `OracleRMSE`'s placement at $k = 6$. The $k = 8, 9, 10$ placements are then all in similar locations.

The qualitative analysis above would benefit from a quantitative comparison between sets of placements with corresponding rankings. Future work will explore the use of a distance metric that operates on pairs of $(\boldsymbol{k}, \boldsymbol{x})$ to quantify the difference between the greedy placements. Furthermore, while the findings from this single run of the sensor placement algorithm shows the promise of the `MaxMI` criterion, future work will explore the robustness of this result.

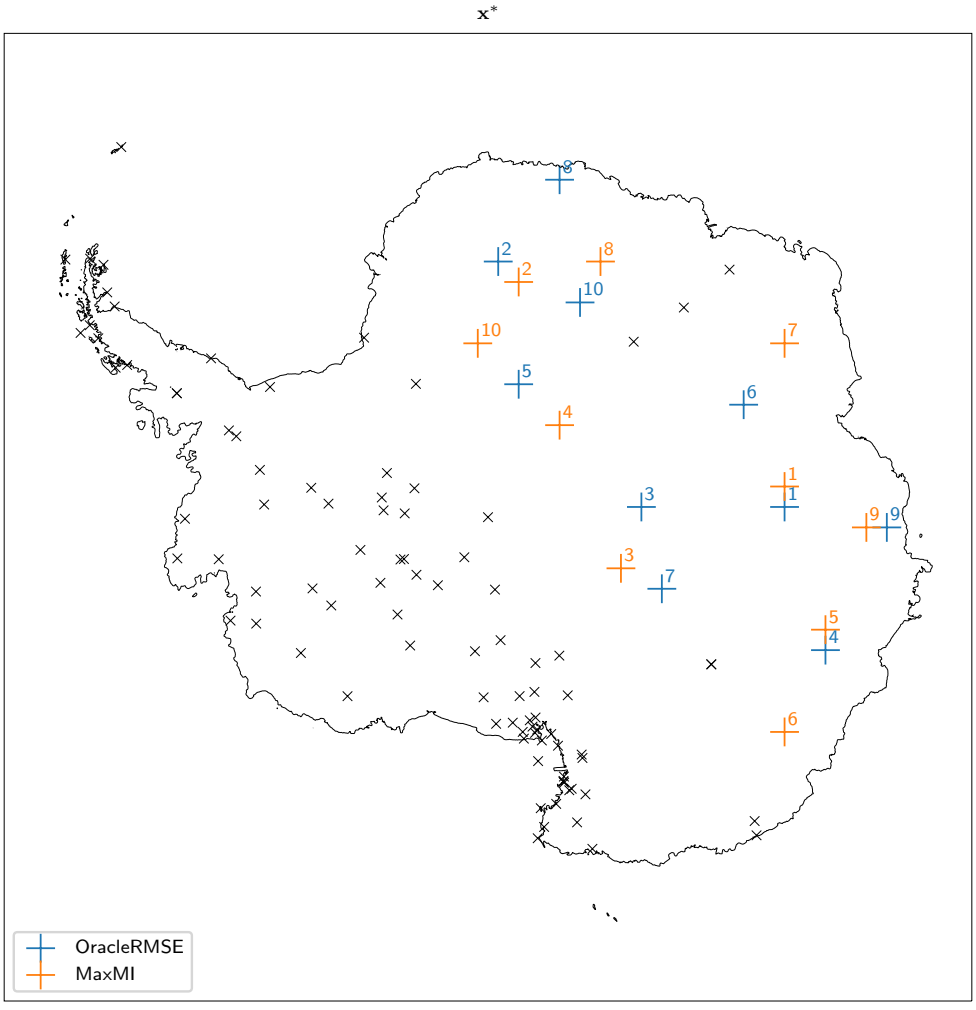

Figure E1: Expanded view of placements from `OracleRMSE` and `MaxMI`. Initial context stations from 2018/01/01 are shown for demonstration.