Supplementary material

Proof of Proposition 2

The transform matrix $\mathbf{G}_+$ can be written as

$$\mathbf{G}_+ = \mathbf{T}_1 \oplus \mathbf{T}_2 = \mathbf{T}_1 \otimes \mathbf{I}_H + \mathbf{I}_W \otimes \mathbf{T}_2,$$

where $\oplus$ denotes the Kronecker sum and $\otimes$ the Kronecker product, $\mathbf{T}_1$ is a $W \times W$ tridiagonal Toeplitz matrix, denoted $\mathbf{T}_1 = (W; a_2, a_1/2, a_4)$, meaning that

$$\mathbf{T}_1 = \begin{bmatrix} a_1/2 & a_4 & & & 0 \\ a_2 & a_1/2 & a_4 & & \\ & a_2 & \ddots & \ddots & \\ & & \ddots & & a_4 \\ 0 & & & a_2 & a_1/2 \end{bmatrix},$$

and similarly $\mathbf{T}_2 = (H; a_3, a_1/2, a_5)$. For the eigenvalues of a Kronecker sum, it holds that if λ_1 is an eigenvalue of $\mathbf{T}_1$ and λ_2 is an eigenvalue of $\mathbf{T}_2$, then $\lambda_1 + \lambda_2$ is an eigenvalue of $\mathbf{T}_1 \oplus \mathbf{T}_2$ (Graham, 1981). Moreover, the eigenvalues of a tridiagonal Toeplitz matrix $\mathbf{T} = (n; b, a, c)$ have a simple formula

$$\lambda_i(\mathbf{T}) = a + 2\sqrt{bc}\cos\left(\pi \frac{i}{n+1}\right), \quad \text{for } i = 1, \ldots, n,$$

which holds for real and complex a, b and c (Smith, 1985). Substituting this formula into the expression

$$\det(\mathbf{G}_+) = \prod_{i=1}^{H}\prod_{j=1}^{W} \left(\lambda_i(\mathbf{T}_2) + \lambda_j(\mathbf{T}_1)\right)$$

gives the result in Proposition 2.

$+$-Filter Reparameterization

The following reparameterization is used to ensure that $\mathbf{G}_+$ has real positive eigenvalues

$$\begin{aligned} a_1 &= \text{softplus}(\rho_1) + \text{softplus}(\rho_2) \\ a_2 a_4 &= \left(\text{softplus}(\rho_1)\tanh(\rho_3)/2\right)^2, \quad a_4/a_2 = \exp(\rho_4) \\ a_3 a_5 &= \left(\text{softplus}(\rho_2)\tanh(\rho_5)/2\right)^2, \quad a_5/a_3 = \exp(\rho_6), \end{aligned}$$

where $\rho_1, \ldots, \rho_6$ are real numbers.

Implementation Details

The model parameters $\boldsymbol{\theta}$ and variational parameters $\boldsymbol{\phi}$ are trained with respect to the negative ELBO (Eq. (10)) divided by N as loss function, using Adam optimization (Kingma & Ba, 2014) with default settings, learning rate 0.01 and 100k iterations. The parameters with the lowest loss value are then saved and conditioned on by our implementation of the CG algorithm, for computing the posterior mean and standard deviation of $\mathbf{x}$. We use $N_q = 10$ samples from variational approximation to compute the expectation in each iteration. We can train the measurement error σ together with the other parameters $\boldsymbol{\theta}$, but we have used a fixed $\sigma = 0.001$, which seems to give very similar results, but with somewhat faster convergence. For the DGMRFs with seq-filters, we randomly select among the eight possible orientations of the filters in each layer. As the toy data is centered around 0, the bias in each layer was fixed to 0 for this experiment. The satellite data was normalized to have maximum pixel value 1.

Competing Methods

We here briefly describe the methods that are compared against in Table 1, except DIP that is mentioned in Section 5.3. For more details we refer to Heaton et al. (2018).

FRK (Fixed rank kriging) (Zammit-Mangion & Cressie, 2017) approximates a spatial process using a linear combination of K spatial basis functions with $K \ll N$.

Gapfill (Gerber et al., 2018) is an algorithmic, distribution free method that makes predictions using sorting and quantile regression based on closeby pixels.

LatticeKrig (Nychka et al., 2015) approximates a GP with a linear combination of multi-resolution basis function with weights that follow a certain GMRF.

LAGP (Local approximate Gaussian process) (Gramacy & Apley, 2015) fits a GP, but only uses the subset of points in the training data that are closest to the points in the test data.

MetaKriging (Guhaniyogi et al., 2017) is an approximate Bayesian method that splits the training data into subsets, fits one model to each subset, and combines them all into a meta-posterior, here using GPs.

MRA (Multi-resolution approximation) (Katzfuss, 2017) uses a multi-resolution approximation of a GP, similar to LatticeKrig, but uses compactly supported basis functions.

NNGP (Nearest-neighbor Gaussian process) (Datta et al., 2016) approximates a GP by rewriting the joint density of the data points as a product of conditional densities, and truncating the conditioning sets to only contain the nearest neighbors.

Partition makes a spatial partitioning (splits the domain into disjoint subsets) and fits spatial basis functions to each partition, similar to FRK, but with some parameters shared between partitions.

Pred. Proc. (Predictive processes) (Finley et al., 2009) approximates a GP using a set of K knot locations, also known as inducing points, with $K \ll N$ which reduces the

size of the covariance matrix that needs to be inverted.
SPDE (Stochastic partial differential equation) (Lindgren et al., 2011) represents a GP with a GMRF (see Section 2.3).
Tapering (Furrer et al., 2006) obtains an approximation of a GP with sparse covariance matrix by truncating small covariances in a way that preserves positive definiteness.
Peri. Embe. (Periodic embedding) (Guinness & Fuentes, 2017) approximates a GP using the fast Fourier transform on a regular grid.

Linear Trend Model

For inference with the linear trend model in Eq. (11), we extend the vector of latents $\mathbf{x}$ to include also the regression coefficients $\boldsymbol{\beta}$, and use (for linear DGMRFs) the prior

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{z}' \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & v\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \boldsymbol{\beta} \end{bmatrix} \Leftrightarrow \bar{\mathbf{z}} = \bar{\mathbf{G}}\bar{\mathbf{x}}, \quad \bar{\mathbf{z}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),$$

where v can be interpreted as the prior inverse standard deviation of the elements of $\boldsymbol{\beta}$, which we fix at $v = 0.0001$. The posterior for $\bar{\mathbf{x}}$ is a GMRF, similar to Eq. (5), with

$$\tilde{\mathbf{Q}} = \bar{\mathbf{G}}^\top \bar{\mathbf{G}} + \frac{1}{\sigma^2} \begin{bmatrix} \mathbf{I} \\ \mathbf{F}^\top \end{bmatrix} \mathbf{I}_\mathbf{m} \begin{bmatrix} \mathbf{I} & \mathbf{F} \end{bmatrix},$$
$$\tilde{\boldsymbol{\mu}} = \tilde{\mathbf{Q}}^{-1} \left(-\bar{\mathbf{G}}^\top \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} + \frac{1}{\sigma^2} \begin{bmatrix} \mathbf{I} \\ \mathbf{F}^\top \end{bmatrix} \mathbf{y} \right),$$

and thus we can proceed with inference as before, with $\bar{\mathbf{x}}$ instead of $\mathbf{x}$, with slight modifications to the ELBO and to the CG method. We use an independent variational approximation $q_{\boldsymbol{\phi}_\beta}(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta}|\boldsymbol{\nu}_\beta, \mathbf{S}_\beta)$ for $\boldsymbol{\beta}$. Integrating out $\boldsymbol{\beta}$ is important for the predictive performance. For reference, if linear trends are instead removed using the ordinary least squares estimates of $\boldsymbol{\beta}$ in a preprocessing step, the row in Table 1 corresponding to $\text{seq}_{5\times5, L=5}$ instead reads (1.25, 1.74, 0.90, 8.45, 0.89). When the linear trend model is used, we compute posterior standard deviations using standard Monte Carlo estimates, instead of simple RBMC, using $N_s = 100$ samples.

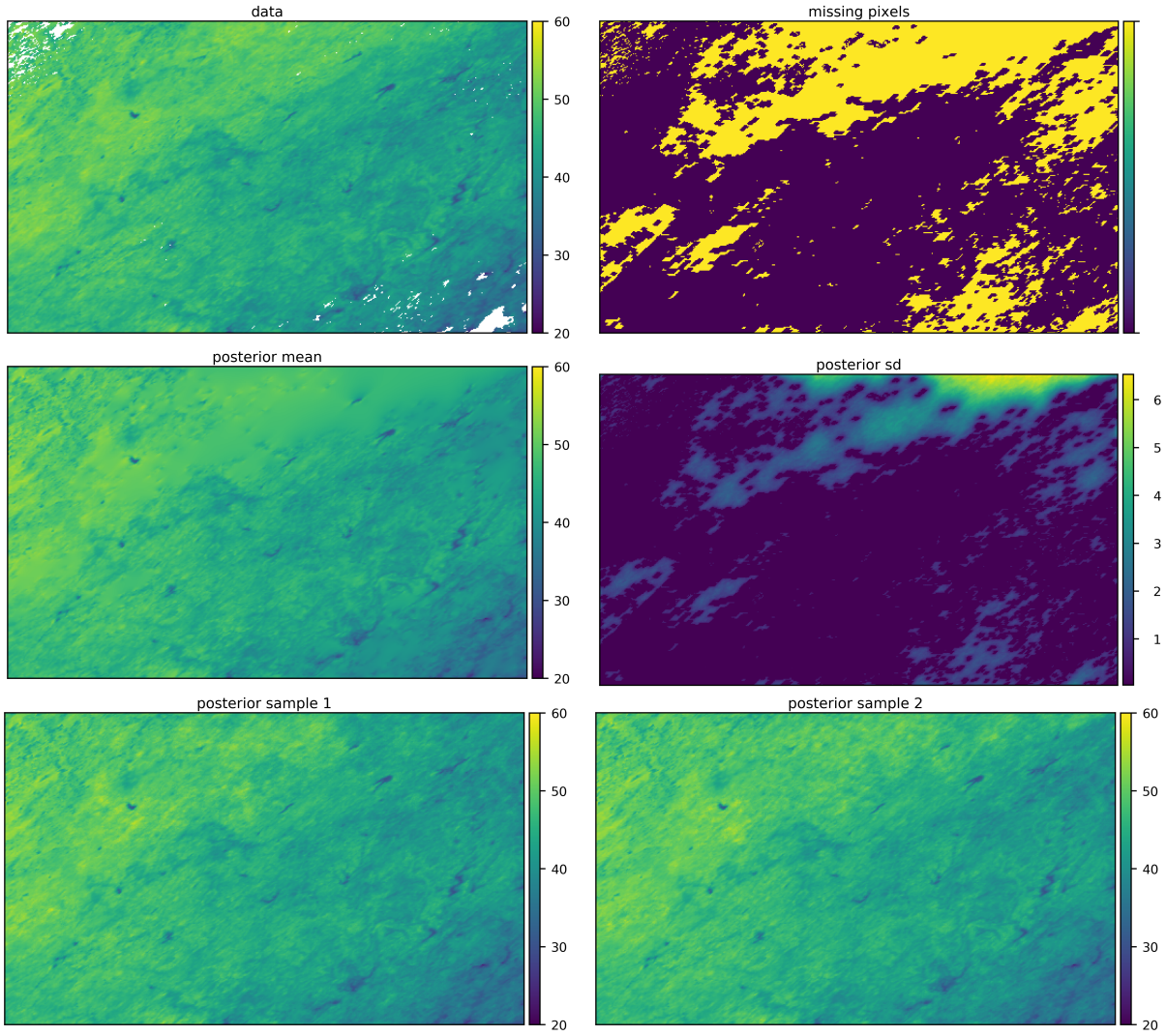

Figure 4. Satellite data inpainting by a linear DGMRF with 5 layers of 5×5 seq-filters.

Table 2. Standard deviations across seeds for the results in Table 1.

Method	MAE	RMSE	CRPS	INT	CVG
$\text{seq}_{5\times5,L=1}$	0.029	0.040	0.011	0.216	0.000
$\text{seq}_{5\times5,L=3}$	0.022	0.042	0.019	0.462	0.001
$\text{seq}_{5\times5,L=5}$	0.037	0.051	0.012	0.461	0.001
$\text{seq}_{3\times3,L=5}$	0.066	0.097	0.039	0.171	0.003
$+_{L=5}$	0.039	0.056	0.018	0.221	0.001
$\text{seq}_{5\times5,L=5,\text{NL}}$	0.066	0.092	-	-	-

Supplement References

[S1] Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association*, 111(514):800–812, 2016.

[S2] Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. Improving the performance of predictive process modeling for large datasets. *Computational statistics & data analysis*, 53(8):2873–2884, 2009.

[S3] Furrer, R., Genton, M. G., and Nychka, D. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3):502–523, 2006.

[S4] Gerber, F., de Jong, R., Schaepman, M. E., Schaepman-Strub, G., and Furrer, R. Predicting missing values in spatio-temporal remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 56:2841—-2853, 2018.

[S5] Graham, A. *Kronecker products and matrix calculus with applications*. Ellis Horwood Limited, Chichester, 1981.

[S6] Gramacy, R. B. and Apley, D. W. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

[S7] Guhaniyogi, R., Li, C., Savitsky, T. D., and Srivastava, S. A divide-and-conquer bayesian approach to large-scale kriging. *arXiv preprint arXiv:1712.09767*, 2017.

[S8] Katzfuss, M. A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association*, 112(517):201–214, 2017.

[S9] Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[S10] Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. A multiresolution gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics*, 24(2):579–599, 2015.

[S11] Smith, G. D. *Numerical solution of partial differential equations: finite difference methods*. Oxford University Press, 1985.

[S12] Zammit-Mangion, A. and Cressie, N. Frk: An r package for spatial and spatio-temporal prediction with large datasets. *arXiv preprint arXiv:1705.08105*, 2017.