Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Multi-Resolution Filters for Massive Spatio-Temporal Data

Marcin Jurek & Matthias Katzfuss

To cite this article: Marcin Jurek & Matthias Katzfuss (2021): Multi-Resolution Filters for Massive Spatio-Temporal Data, Journal of Computational and Graphical Statistics, DOI: 10.1080/10618600.2021.1886938

To link to this article: https://doi.org/10.1080/10618600.2021.1886938

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2021, VOL. 00, NO. 0, 1–16
https://doi.org/10.1080/10618600.2021.1886938

Multi-Resolution Filters for Massive Spatio-Temporal Data

Marcin Jurek[a] and Matthias Katzfuss[b]

[a]Department of Statistics and Data Science, University of Texas at Austin, Austin, TX; [b]Department of Statistics, Texas A&M University, College Station, TX

ABSTRACT
Spatio-temporal datasets are rapidly growing in size. For example, environmental variables are measured with increasing resolution by increasing numbers of automated sensors mounted on satellites and aircraft. Using such data, which are typically noisy and incomplete, the goal is to obtain complete maps of the spatio-temporal process, together with uncertainty quantification. We focus here on real-time filtering inference in linear Gaussian state-space models. At each time point, the state is a spatial field evaluated on a very large spatial grid, making exact inference using the Kalman filter computationally infeasible. Instead, we propose a multi-resolution filter (MRF), a highly scalable and fully probabilistic filtering method that resolves spatial features at all scales. We prove that the MRF matrices exhibit a particular block-sparse multi-resolution structure that is preserved under filtering operations through time. We describe connections to existing methods, including hierarchical matrices from numerical mathematics. We also discuss inference on time-varying parameters using an approximate Rao-Blackwellized particle filter, in which the integrated likelihood is computed using the MRF. Using a simulation study and a real satellite-data application, we show that the MRF strongly outperforms competing approaches. Supplementary materials include Python code for reproducing the simulations, some detailed properties of the MRF and auxiliary theoretical results.

ARTICLE HISTORY
Received May 2020
Revised January 2021

KEYWORDS
Filtering inference; Gaussian process approximations; Satellite data; Spatio-temporal statistics; State-space models

1. Introduction

Massive spatio-temporal data have become ubiquitous in the environmental sciences, which is largely due to Earth-observing satellites providing high-resolution measurements of environmental variables on a continental or even global scale. Accounting for spatial and temporal dependence is very important for satellite data, as atmospheric variables vary over space and time, and measurements from different orbits are often complementary in their coverage.

When time and space are discretized, spatio-temporal data are typically modeled using a dynamical state-space model (SSM), which describes how the state (i.e., the spatial field evaluated at a spatial grid) evolves over time and how the state is related to the observations. Dynamical SSMs can include information from other sources and sophisticated temporal dynamics in terms of partial differential equations; for example, the effect of wind on atmospheric variables can be captured by an advection term. Such informative, physical evolution models are crucial for producing meaningful forecasts.

We focus here on real-time or online filtering inference in linear Gaussian SSMs, which means that at each time point t, we are interested in the posterior distribution of the spatial field at time t given all data obtained up to time t. The filtering distributions in this setting are Gaussian and can in principle be determined exactly by the Kalman filter (Kalman 1960), but this technique is not computationally feasible for large grids. Particle filter methods such as sequential importance (re)sampling (e.g., Gordon, Salmond, and Smith 1993) are asymptotically exact as the number of particles increases, but suffer from particle collapse for finite particle size in high dimensions (e.g., Snyder et al. 2008).

In the geosciences, filtering inference in SSMs is referred to as data assimilation (see, e.g., Nychka and Anderson 2010, for a review), especially when the evolution is described by a complex computer model. Data assimilation is typically carried out via variational methods (e.g., Talagrand and Courtier 1987) or the ensemble Kalman filter (EnKF; e.g., Evensen 1994, 2007; Katzfuss, Stroud, and Wikle 2016; Houtekamer and Zhang 2016). The EnKF represents distributions by an ensemble, which is propagated using the temporal evolution model and updated via a linear shift based on new observations. In practice, only small ensemble sizes are computationally feasible, resulting in a low-dimensional representation and substantial sampling error.

In the statistics literature, computationally feasible filtering approaches for dynamical spatio-temporal SSMs often rely on low-rank assumptions (e.g., Verlaan and Heemink 1995; Pham, Verron, and Christine Roubaud 1998; Wikle and Cressie 1999; Wikle et al. 2001; Cressie, Shi, and Kang 2010; Katzfuss and Cressie 2011; Bradley, Holan, and Wikle 2018), but such approaches cannot fully resolve fine-scale variation (Stein 2014). Therefore, recent methods for large spatial-only data have instead achieved fast computation through sparsity assumptions (e.g., Lindgren, Rue, and Lindström 2011; Nychka et al. 2015; Datta et al. 2016a; Katzfuss and Guinness 2021), and idea that can also be used in the context of retrospective, "off-line"

CONTACT Matthias Katzfuss katzfuss@gmail.com Department of Statistics, Texas A&M University, College Station, TX 77843.

Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

spatio-temporal analysis, in which time is essentially treated as an additional spatial dimension and the resulting spatio-temporal covariance function is modeled or approximated (e.g., Zhang, Sang, and Huang 2015; Datta et al. 2016b). However, most sparsity-based methods cannot be easily extended to the filtering perspective of interest here, because the sparsity structure is lost when propagating forward in time.

Here, we propose a novel multi-resolution filter (MRF) for big streaming spatio-temporal data based on linear Gaussian SSMs. The MRF is a highly scalable, fully probabilistic procedure that results in joint posterior predictive distributions for the spatio-temporal field of interest. In contrast to the EnKF, MRF computations are deterministic and do not suffer from sampling variability. In contrast to low-rank approaches, the MRF does not rely on dimension reduction. Similar to wavelet-based filtering methods (e.g., Chui 1992; Cristi and Tummala 2000; Renaud, Starck, and Murtagh 2005; Beezley, Mandel, and Cobb 2011; Hickmann and Godinez 2015), the MRF can be viewed as employing a large number of basis functions at multiple levels of spatial resolution, which can capture spatial structure from very fine to very large scales. However, as opposed to wavelets, the MRF basis functions automatically adapt to approximate the covariance structure implied by the assumed SSM. These features allowed the MRF to strongly outperform existing approaches in our numerical comparisons.

The MRF relies on a new approximate covariance-matrix decomposition, for which the resulting matrix factors exhibit a particular block-sparse multi-resolution structure. This decomposition is based on the multi-resolution approximation (Katzfuss 2017; Katzfuss and Gong 2020) of spatial processes, which performed very well in a recent comparison of different methods for large spatial-only data (Heaton et al. 2017). Using advanced concepts from graph theory, we prove the perhaps surprising property that the block-sparse structure of the MRF matrices can be maintained under filtering operations through time, which in turn is crucial for allowing us to show that the MRF exhibits linear computational complexity for fixed tuning parameters. Note that this is in contrast to other sparse-matrix approximations, as even matrices with simple sparsity patterns (e.g., tridiagonal matrices) do not preserve sparsity under inversion. In fact, we suspect that the multi-resolution decomposition (MRD) and its special cases are unique in terms of preserving matrix sparsity.

We also establish a close connection between our MRD and hierarchical matrices. Despite being relatively unknown in statistics, hierarchical matrices (e.g., Hackbusch 2015) are a highly popular and widely studied class of matrix approximations in numerical mathematics. We introduce this matrix class into the statistical literature, and describe how hierarchical matrices can be applied to SSMs based on second-order partial differential equations, including those describing advection and diffusion. This marks a major step forward with respect to multiple previous hierarchical-matrix approaches for fast high-dimensional Kalman filtering (e.g., Li et al. 2014; Saibaba, Miller, and Kitanidis 2015; Ambikasaran et al. 2016), which were only applicable in the simple case of a random walk. The MRF is able to preserve important aspects of the physical model, such as interactions between the cells of the discretization grid, which previous approaches were forced to ignore for computational reasons. For example, this makes the MRF suitable for transport models (e.g., advection, human and animal mobility, flow), which are ubiquitous in the geosciences.

Finally, we discuss extensions for inference on time-varying parameters that are not part of the spatial field, using a Rao-Blackwellized particle filter, in which the integrated likelihood is approximated using the MRF.

The remainder of this article is organized as follows. Section 2 describes the linear Gaussian SSM and reviews the Kalman filter. In Section 3, we present the MRF. Section 4 details key properties of the MRF, and Section 5 discusses connections to existing approaches. Section 6 shows how the MRF can be extended when the model includes unknown parameters. In Section 7, we present a numerical comparison of the MRF to existing methods. Section 8 demonstrates a practical application of the MRF to inferring sediment concentration in Lake Michigan based on satellite data. We conclude in Section 9. Proofs are given in Appendix A.

A separate supplementary material document contains Sections S1–S8 with further properties, details, and proofs. At *http://spatial.stat.tamu.edu*, we provide additional illustrations. All code will be provided upon publication.

2. Spatio-Temporal SSMs and Filtering Inference

2.1. Spatio-Temporal SSM

Let $\mathbf{x}_t$ be the $n_{\mathcal{G}}$-dimensional latent state vector of interest, representing a (mean-corrected) spatio-temporal process $x_t(\cdot)$ at time t evaluated at a fine grid $\mathcal{G} = \{\mathbf{g}_1, \ldots, \mathbf{g}_{n_\mathcal{G}}\}$ on a spatial domain or region $\mathcal{D}$. Further, let $\mathbf{y}_t$ denote the observed n_t-dimensional data vector at time t. We assume a linear Gaussian spatio-temporal SSM given by an observation equation and an evolution equation,

$$\begin{aligned} \mathbf{y}_t &= \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t, \quad \mathbf{v}_t \sim \mathcal{N}_{n_t}(\mathbf{0}, \mathbf{R}_t), & (1)\\ \mathbf{x}_t &= \mathbf{A}_t\mathbf{x}_{t-1} + \mathbf{w}_t, \ \mathbf{w}_t \sim \mathcal{N}_{n_\mathcal{G}}(\mathbf{0}, \mathbf{Q}_t), & (2) \end{aligned}$$

respectively, for time $t = 1, 2, \ldots$. The initial state also follows a Gaussian distribution: $\mathbf{x}_0 \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Sigma}_{0|0})$. The noise covariance matrix $\mathbf{R}_t$ will be assumed to be diagonal or block-diagonal here for simplicity (see Assumption 1 in Section 4.2.1). No computationally convenient structure is assumed for the innovation covariance matrix $\mathbf{Q}_t$. The observation noise $\mathbf{v}_t$ and the innovation $\mathbf{w}_t$ are mutually and serially independent, and independent of the state $\mathbf{x}_{t-1}$. We assume that all matrices in (1)–(2) (and $\boldsymbol{\mu}_{0|0}$ and $\boldsymbol{\Sigma}_{0|0}$) are known. The case of unknown parameters is discussed in Section 6.

The observation matrix $\mathbf{H}_t$ relates the state to the observations. This enables combining observations from different instruments or modeling areal observations given by averaging over certain elements of the state vector. Here, we usually assume point-level measurements for simplicity, although a block-diagonal form for $\mathbf{H}_t$ is possible (see Assumption 1).

The evolution matrix $\mathbf{A}_t$ determines how the process evolves over time. It can be specified in terms of a system of partial differential equations (PDEs), may depend on other variables, or—in the absence of further information—could simply be a scaled identity operator indicating a random walk over time. We assume that the evolution is local and $\mathbf{A}_t$ is sparse (Assumption 2 in Section 4.2.2).

Note that the SSM in (1)–(2), which is a latent Markov model of order 1, is very general and can describe a broad class of systems. Higher-order Markov models can also be written in the form (1)–(2) by expanding the state space. Non-Gaussian observations can often be transformed to be approximately Gaussian. Other extensions are also straightforward, such as letting the grid $\mathcal{G}$ vary over time.

2.2. Filtering Inference Using the Kalman Filter (KF)

We are interested in filtering inference on the state $\mathbf{x}_t$. That is, at each time t, the goal is to find the conditional distribution of $\mathbf{x}_t$ given all observations up to and including time t, denoted by $\mathbf{x}_t|\mathbf{y}_{1:t}$, where $\mathbf{y}_{1:t} = (\mathbf{y}_1', \dots, \mathbf{y}_t')'$.

For the linear Gaussian SSM in (1)–(2), the filtering distributions are Gaussian. These filtering distributions can be obtained recursively for $t = 1, 2, \dots$ using the Kalman filter (Kalman 1960), which consists of a forecast step and an update step at each time point. Denote the filtering distribution at time $t-1$ by $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1} \sim \mathcal{N}_n(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1})$. The forecast step obtains the forecast or prior distribution of $\mathbf{x}_t$ based on the previous filtering distribution and the evolution model (2) as

$$\mathbf{x}_t|\mathbf{y}_{1:t-1} \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}), \qquad \boldsymbol{\mu}_{t|t-1} := \mathbf{A}_t\boldsymbol{\mu}_{t-1|t-1},$$
$$\boldsymbol{\Sigma}_{t|t-1} := \mathbf{A}_t\boldsymbol{\Sigma}_{t-1|t-1}\mathbf{A}_t' + \mathbf{Q}_t.$$

Then, the update step modifies this forecast distribution based on the observation vector $\mathbf{y}_t$ and the observation Equation (1), to obtain the filtering distribution of $\mathbf{x}_t$:

$$\begin{aligned} &\mathbf{x}_t|\mathbf{y}_{1:t} \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t}),\\ &\quad\boldsymbol{\mu}_{t|t} := \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1}),\\ &\quad\boldsymbol{\Sigma}_{t|t} := (\mathbf{I}_{n_\mathcal{G}} - \mathbf{K}_t\mathbf{H}_t)\boldsymbol{\Sigma}_{t|t-1}, \end{aligned} \tag{3}$$

where $\mathbf{K}_t := \boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t'(\mathbf{H}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t' + \mathbf{R}_t)^{-1}$ is the $n_\mathcal{G} \times n_t$ Kalman gain matrix.

While the Kalman filter provides the exact solution to our filtering problem, it requires computing and propagating the $n_\mathcal{G} \times n_\mathcal{G}$ covariance matrix $\boldsymbol{\Sigma}_{t|t}$ and decomposing the $n_t \times n_t$ matrix $(\mathbf{H}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}_t' + \mathbf{R}_t)$ in $\mathbf{K}_t$, and is thus computationally infeasible for large $n_\mathcal{G}$ or large n_t. Therefore, approximations are required for large spatio-temporal data.

3. The MRF

3.1. Overview

We now propose an MRF for spatio-temporal SSMs of the form (1)–(2) when the grid size $n_\mathcal{G}$ or the data sizes n_t are large, roughly between 10^4 and 10^9. The MRF can be viewed as an approximation of the Kalman filter in Section 2.2.

The most important ingredient of the MRF is a novel MRD. Given a spatial covariance matrix $\boldsymbol{\Sigma}$, the MRD computes $\mathbf{B} = \text{MRD}(\boldsymbol{\Sigma})$ such that $\boldsymbol{\Sigma} \approx \mathbf{B}\mathbf{B}'$. We will describe the MRD in detail in Section 3.4. For now, we merely note that the MRD algorithm is fast, and the resulting multi-resolution factor $\mathbf{B}$ is of the same dimensions as $\boldsymbol{\Sigma}$ but exhibits a particular block-sparse structure (see Figure 2(a)).

The MRF algorithm proceeds as follows:

Algorithm 1: Multi-resolution filter (MRF).
At the initial time $t = 0$, compute $\mathbf{B}_{0|0} = \text{MRD}(\boldsymbol{\Sigma}_{0|0})$.
Then, for each $t = 1, 2, \dots$, do:

1. **Forecast step:** Apply the evolution matrix $\mathbf{A}_t$ to obtain $\boldsymbol{\mu}_{t|t-1} = \mathbf{A}_t\boldsymbol{\mu}_{t-1|t-1}$ and $\mathbf{B}^F_{t|t-1} = \mathbf{A}_t\mathbf{B}_{t-1|t-1}$. Carry out an MRD $\mathbf{B}_{t|t-1} = \text{MRD}(\mathbf{S}_{t|t-1})$, where $\mathbf{S}_{t|t-1} = \mathbf{B}^F_{t|t-1}(\mathbf{B}^F_{t|t-1})' + \mathbf{Q}_t$, to obtain $\mathbf{x}_t|\mathbf{y}_{1:t-1} \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$ with $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}_{t|t-1}'$.
2. **Update step:** Compute $\mathbf{B}_{t|t} := \mathbf{B}_{t|t-1}(\mathbf{L}_t^{-1})'$, where $\mathbf{L}_t$ is the lower Cholesky triangle of $\boldsymbol{\Lambda}_t := \mathbf{I}_{n_\mathcal{G}} + \mathbf{B}_{t|t-1}'\mathbf{H}_t'\mathbf{R}_t^{-1}\mathbf{H}_t\mathbf{B}_{t|t-1}$, to obtain $\mathbf{x}_t|\mathbf{y}_{1:t} \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$ with $\boldsymbol{\Sigma}_{t|t} = \mathbf{B}_{t|t}\mathbf{B}_{t|t}'$ and $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1} + \mathbf{B}_{t|t}\mathbf{B}_{t|t}'\mathbf{H}_t'\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1})$.

The terms related to the forecast and update distributions are denoted with subscripts $t|t-1$ and $t|t$, respectively. The key to the scalability of this algorithm is that while $\boldsymbol{\Sigma}_{t|t-1}$ and $\boldsymbol{\Sigma}_{t|t}$ are large and dense matrices, they are never explicitly calculated and instead represented by the block-sparse matrices $\mathbf{B}_{t|t-1}$ and $\mathbf{B}_{t|t}$, respectively. Also, as shown in Section 4.2, $\mathbf{B}_{t|t}$ has the same sparsity structure as $\mathbf{B}_{t|t-1}$, which allows the cycle to start over for the next time point $t + 1$. The forecast step and update step will be discussed in more detail in Sections 3.2 and 3.3, respectively.

3.2. Details of the MRF Forecast Step

Assume that we have obtained the filtering distribution $\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1} \sim \mathcal{N}_n(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\Sigma}_{t-1|t-1})$, where $\boldsymbol{\Sigma}_{t-1|t-1} = \mathbf{B}_{t-1|t-1}\mathbf{B}_{t-1|t-1}'$ and $\mathbf{B}_{t-1|t-1}$ is a block-sparse matrix. Following the forecast step of the standard Kalman filter, we want to obtain the prior distribution at time t, $\mathbf{x}_t|\mathbf{y}_{1:t-1} \sim \mathcal{N}_n(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})$.

Because of the sparsity of $\mathbf{A}_t$ (see Assumption 2 in Section 4.2.2), computing the forecast mean $\boldsymbol{\mu}_{t|t-1} = \mathbf{A}_t\boldsymbol{\mu}_{t-1|t-1}$ and the forecast basis matrix $\mathbf{B}^F_{t|t-1} = \mathbf{A}_t\mathbf{B}_{t-1|t-1}$ is fast. Then, rather than calculating the dense $n_\mathcal{G} \times n_\mathcal{G}$ forecast covariance matrix $\mathbf{S}_{t|t-1} = \mathbf{B}^F_{t|t-1}(\mathbf{B}^F_{t|t-1})' + \mathbf{Q}_t$ explicitly, we obtain its MRD $\mathbf{B}_{t|t-1} = \text{MRD}(\mathbf{S}_{t|t-1})$ as described in Section 3.4. This implies an approximation to the prior covariance matrix as $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}_{t|t-1}'$. Again, $\boldsymbol{\Sigma}_{t|t-1}$ does not need to be computed explicitly, because only $\mathbf{B}_{t|t-1}$ is used in the update step below.

3.3. Details of the MRF Update Step

The objective of the update step is to compute the posterior distribution $\mathbf{x}_t|\mathbf{y}_{1:t} \sim \mathcal{N}_{n_\mathcal{G}}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})$ given the prior quantities $\boldsymbol{\mu}_{t|t-1}$ and $\mathbf{B}_{t|t-1}$ (such that $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}_{t|t-1}'$) obtained in the forecast step.

Following the Kalman filter update in (3), we have

$$\begin{aligned}\boldsymbol{\Sigma}_{t|t} &= (\mathbf{I}_{n_\mathcal{G}} - \mathbf{K}_t\mathbf{H}_t)\boldsymbol{\Sigma}_{t|t-1}\\ &= \mathbf{B}_{t|t-1}\big(\mathbf{I}_{n_\mathcal{G}} - \mathbf{B}'_{t|t-1}\mathbf{H}'_t(\mathbf{H}_t\mathbf{B}_{t|t-1}\mathbf{B}'_{t|t-1}\mathbf{H}'_t + \mathbf{R}_t)^{-1}\\ &\qquad \mathbf{H}_t\mathbf{B}_{t|t-1}\big)\mathbf{B}'_{t|t-1}\\ &= \mathbf{B}_{t|t-1}\big(\mathbf{I}_{n_\mathcal{G}} + \mathbf{B}'_{t|t-1}\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{H}_t\mathbf{B}_{t|t-1}\big)^{-1}\mathbf{B}'_{t|t-1}\\ &= \mathbf{B}_{t|t-1}\boldsymbol{\Lambda}_t^{-1}\mathbf{B}'_{t|t-1} = \mathbf{B}_{t|t}\mathbf{B}'_{t|t},\end{aligned}$$

where $\mathbf{B}_{t|t} := \mathbf{B}_{t|t-1}(\mathbf{L}_t^{-1})'$, $\mathbf{L}_t$ is the lower Cholesky triangle of $\boldsymbol{\Lambda}_t := \mathbf{I}_{n_\mathcal{G}} + \mathbf{B}'_{t|t-1}\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{H}_t\mathbf{B}_{t|t-1}$, and we have applied the Sherman–Morrison–Woodbury formula (e.g., Henderson and Searle 1981) to $\boldsymbol{\Lambda}_t$. As we show in Section 4.2.2, $\boldsymbol{\Lambda}_t$ exhibits a special sparsity pattern that ensures fast Cholesky factorization.

To obtain the filtering mean, we use the Searle set of identities (Searle 1982, p. 151), to write the Kalman gain as

$$\begin{aligned}\mathbf{K}_t &= \boldsymbol{\Sigma}_{t|t-1}\mathbf{H}'_t(\mathbf{H}_t\boldsymbol{\Sigma}_{t|t-1}\mathbf{H}'_t + \mathbf{R}_t)^{-1}\\ &= \mathbf{B}_{t|t-1}\mathbf{B}'_{t|t-1}\mathbf{H}'_t(\mathbf{H}_t\mathbf{B}_{t|t-1}\mathbf{B}'_{t|t-1}\mathbf{H}'_t + \mathbf{R}_t)^{-1}\\ &= \mathbf{B}_{t|t-1}(\mathbf{I}_{n_\mathcal{G}} + \mathbf{B}'_{t|t-1}\mathbf{H}'_t\mathbf{R}_t^{-1}\mathbf{H}_t\mathbf{B}_{t|t-1})^{-1}\mathbf{B}'_{t|t-1}\mathbf{H}'_t\mathbf{R}_t^{-1}\\ &= \mathbf{B}_{t|t-1}\boldsymbol{\Lambda}_t^{-1}\mathbf{B}'_{t|t-1}\mathbf{H}'_t\mathbf{R}^{-1} = \mathbf{B}_{t|t}\mathbf{B}'_{t|t}\mathbf{H}'_t\mathbf{R}_t^{-1},\end{aligned}$$

and so we have

$$\begin{aligned}\boldsymbol{\mu}_{t|t} &= \boldsymbol{\mu}_{t|t-1} + \mathbf{K}_t(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1})\\ &= \boldsymbol{\mu}_{t|t-1} + \mathbf{B}_{t|t}\mathbf{B}'_{t|t}\mathbf{H}'_t\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1}).\end{aligned}$$

Thus, the MRF update step in Algorithm 1 is exact for given $\boldsymbol{\mu}_{t|t-1}$ and $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}'_{t|t-1}$. Crucially, we will show in Proposition 3 that $\mathbf{B}_{t|t}$ has the same sparsity structure as $\mathbf{B}_{t|t-1}$, and hence it satisfies the block-sparsity assumption at the beginning of Section 3.2.

3.4. The MRD

We now propose an approximate MRD of a generic spatial covariance matrix $\boldsymbol{\Sigma}$, which is used in the forecast step of the MRF in Algorithm 2. Specifically, we consider a vector $\mathbf{x} = \big(x(\mathbf{g}_1), \dots, x(\mathbf{g}_{n_\mathcal{G}})\big)' \sim \mathcal{N}_{n_\mathcal{G}}(\mathbf{0}, \boldsymbol{\Sigma})$, evaluated at a grid $\mathcal{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_{n_\mathcal{G}}\}$ over the spatial domain $\mathcal{D}$. The MRD is based on a multi-resolution approximation of Gaussian processes (Katzfuss 2017)—see Section 5.2 for more details.

3.4.1. Partitioning and Knots

The MRD requires a domain partitioning and selection of knots at M resolutions. Consider a recursive partitioning of $\mathcal{D}$ into J regions, $\mathcal{D}_1, \dots, \mathcal{D}_J$, each of which is again divided into J smaller subregions (e.g., $\mathcal{D}_2$ is split into subregions $\mathcal{D}_{2,1}, \dots, \mathcal{D}_{2,J}$), and so forth, up to resolution M. We write this as

$$\begin{gathered}\mathcal{D}_{j_1,\dots,j_m} = \dot{\bigcup}_{j_{m+1}=1}^{J} \mathcal{D}_{j_1,\dots,j_{m+1}},\\ (j_1, \dots, j_m) \in \{1, \dots, J\}^m,\\ m = 1, \dots, M-1.\end{gathered}$$

Let $\mathcal{G}_{j_1,\dots,j_m} = \mathcal{G} \cap \mathcal{D}_{j_1,\dots,j_m}$ be the grid points that lie in region $\mathcal{D}_{j_1,\dots,j_m}$, and let $\mathcal{I}_{j_1,\dots,j_m} = \{i : \mathbf{g}_i \in \mathcal{D}_{j_1,\dots,j_m}\}$ be the corresponding indices, and so $\mathcal{I} = \{1, \dots, n_\mathcal{G}\}$.

Further, we require a hierarchy of "knot" indices, such that $\mathcal{K}_{j_1,\dots,j_m}$ is a small set of r_m indices chosen from $\mathcal{I}_{j_1,\dots,j_m}$. It is assumed that for each resolution m, the number of knots is roughly the same in each subregion (i.e., $|\mathcal{K}_{j_1,\dots,j_m}| = r_m$), while it may change across resolutions. We denote the knots at the coarsest resolution by $\mathcal{K}^0$, and we use $\mathcal{K}^m = \bigcup_{j_1,\dots,j_m} \mathcal{K}_{j_1,\dots,j_m}$ to denote the set of all knots at each resolution $m = 1, \dots, M$. Further, we define $\mathcal{K}^{0:m} = \bigcup_{l=0}^{m} \mathcal{K}^l$ as the set of all knots at resolutions 0 through m. To ensure that $\mathcal{K}^0 \cup \{\mathcal{K}_{j_1,\dots,j_m} : (j_1, \dots, j_m) \in \{1, \dots, J\}^m; m = 1, \dots, M\}$ is a partition of $\mathcal{I}$, we sequentially choose $\mathcal{K}_{j_1,\dots,j_m} \subset (\mathcal{I}_{j_1,\dots,j_m} \setminus \mathcal{K}^{0:m-1})$ for $m = 1, \dots, M$.

The partitioning and knot selection is illustrated in a toy example in Figure 1. In general, the accuracy of the approximation increases with r_m and decreases with J. Each r_{m-1} should be sufficiently large (subject to computational constraints) to capture the dependence between the $\mathcal{D}_{j_1,\dots,j_m}$ that is not already captured at lower resolutions, which often means that r_m can decrease as a function of m.

While the optimal strategy for domain partitioning and knot selection is unknown, we found the following approach to work well in practice. We set $J = 2$ for one-dimensional problems and $J = 4$ for two-dimensional problems, always splitting the domain along one of the coordinates. We set $N = \sum_{m=0}^{M} r_m$ according to the available computational budget (see Section 4.2.2 for details); then, we split N into r_m that are roughly equal for most resolutions, except that we use a larger r_0 at the coarsest resolution and smaller r_m for very high resolutions m. Given the domain partitioning and the r_m, each set of knots

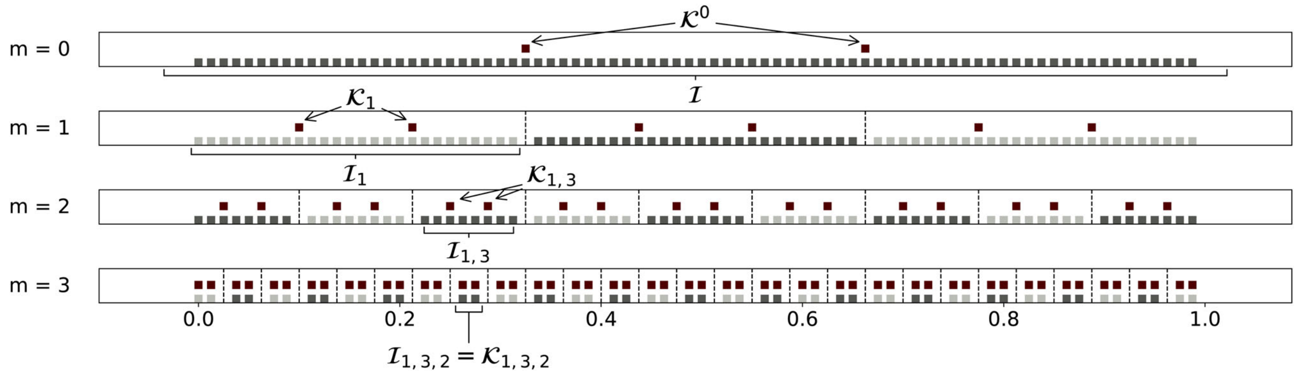


Figure 1. Illustration of knot placement for a regular grid of $n_\mathcal{G} = 80$ points on a one-dimensional domain $\mathcal{D} = [0, 1]$ (x-axis). Recursively for $m = 0, 1, \dots, M$ (with $M = 3$ here), each region is split into $J = 3$ subregions (dashed lines), with $r_m = 2$ knots per region (maroon dots). Thus, for example, $\mathcal{D}_1 = [0, 1/3)$, $\mathcal{D}_{1,3} = [2/9, 1/3)$, and $\mathcal{D}_{1,3,2} = [7/27, 8/27)$. Grid points used as knots at resolutions m are not plotted on finer ($> m$) resolutions for clarity.

$\mathcal{K}_{j_1,\dots,j_m}$ can be chosen as a roughly uniform grid over the subregion $\mathcal{D}_{j_1,\dots,j_m}$. We used this strategy for our numerical examples in Sections 7 and 8. Knot selection can also be accomplished by picking points ordered earlier using a maximum-minimum-distance ordering (e.g., Guinness 2016).

Note that because $\mathcal{G}$ is assumed constant over time here, we only need to do the domain partitioning and knot selection once (not at each time point). We assume that the elements in $\mathbf{x}_t$ are ordered such that if $(j_1,\dots,j_M) \succ_L (i_1,\dots i_M)$, where $\succ_L$ stands for lexicographic ordering, then $\min\left(\mathcal{I}_{j_1,\dots,j_M}\right) > \max\left(\mathcal{I}_{i_1,\dots,i_M}\right)$.

3.4.2. The MRD Algorithm

For index sets $\mathcal{J}_1$ and $\mathcal{J}_2$, denote by $\mathbf{\Sigma}[\mathcal{J}_1,\mathcal{J}_2]$ the submatrix of $\mathbf{\Sigma}$ obtained by selecting the $\mathcal{J}_1$ rows and $\mathcal{J}_2$ columns, and $\mathbf{\Sigma}[\mathcal{J}_1, :]$ is the submatrix of the $\mathcal{J}_1$ rows and all columns. Based on grid indices $\{\mathcal{I}_{j_1,\dots,j_m}\}$ and knot indices $\{\mathcal{K}_{j_1,\dots,j_m}\}$ selected as described in Section 3.4.1, the MRD of a spatial covariance matrix $\mathbf{\Sigma}$ proceeds as follows:

```
Algorithm 2: Multi-resolution decomposition of $\mathbf{\Sigma}$.
1. Compute $\mathbf{W}^0 = \mathbf{\Sigma}[\mathcal{I},\mathcal{K}^0]$ and $\mathbf{V}^0 = \mathbf{\Sigma}[\mathcal{K}^0,\mathcal{K}^0]$.
2. For $m = 0,1,\dots,M$ and $(j_1,\dots,j_m) \in \{1,\dots,J\}^m$:
    (a) For $\ell = 1,\dots,m$, compute
        $$\mathbf{W}^\ell_{j_1,\dots,j_m} = \mathbf{\Sigma}[\mathcal{I}_{j_1,\dots,j_m},\mathcal{K}_{j_1,\dots,j_\ell}] - \sum_{k=0}^{\ell-1} \mathbf{W}^k_{j_1,\dots,j_m}(\mathbf{V}^k_{j_1,\dots,j_k})^{-1}(\mathbf{V}^k_{j_1,\dots,j_\ell})', \quad (4)$$
        $$\mathbf{V}^\ell_{j_1,\dots,j_m} = \mathbf{\Sigma}[\mathcal{K}_{j_1,\dots,j_m},\mathcal{K}_{j_1,\dots,j_\ell}] - \sum_{k=0}^{\ell-1} \mathbf{V}^k_{j_1,\dots,j_m}(\mathbf{V}^k_{j_1,\dots,j_k})^{-1}(\mathbf{V}^k_{j_1,\dots,j_\ell})'. \quad (5)$$
    (b) Set $\mathbf{B}_{j_1,\dots,j_m} = \mathbf{W}^m_{j_1,\dots,j_m}(\mathbf{V}^m_{j_1,\dots,j_m})^{-1/2}$.
3. Return $\mathbf{B} = \mathrm{MRD}(\mathbf{\Sigma})$, where $\mathbf{B} = \left(\mathbf{B}^M,\mathbf{B}^{M-1},\dots,\mathbf{B}^0\right)$ with $\mathbf{B}^m = \mathit{blockdiag}(\{\mathbf{B}_{j_1,\dots,j_m} : (j_1,\dots,j_m) \in \{1,\dots,J\}^m\})$.
```

The resulting matrix $\mathbf{B}$ is of the same size as $\mathbf{\Sigma}$ but has a block-sparse structure, which is illustrated in Figure 2(a).

4. Properties of the MRF

4.1. Approximation Accuracy

The only difference between the MRF (Algorithm 1) and the exact Kalman filter (Section 2.2) is the MRD approximation of the forecast covariance matrix at each time point; that is, instead of taking $\mathbf{\Sigma}_{t|t-1} = \mathbf{S}_{t|t-1}$, the MRF assumes $\mathbf{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}'_{t|t-1}$ with $\mathbf{B}_{t|t-1} = \mathrm{MRD}(\mathbf{S}_{t|t-1})$. Hence, the MRF is exact when the MRD at each time point is exact.

However, the MRD is only exact in certain special cases. One trivial example is given by $M = 0$ and $r_0 = n_{\mathcal{G}}$ (see Section S1). Thus, the MRF converges to the exact Kalman filter as $r_0 \to n_{\mathcal{G}}$, but computational feasibility for large $n_{\mathcal{G}}$ relies on $r_0 \ll n_{\mathcal{G}}$. More generally, the accuracy of the MRD is highest for settings and covariance functions with a strong screening effect (Stein 2002, 2011). Hence, another instance of exactness is when $\mathbf{S}_{t|t-1}$ is based on an exponential covariance function on a one-dimensional domain, $\mathcal{D} \subset \mathbb{R}$, and we place a total of $r_m = J - 1$ knots, one at each subregion boundary (Katzfuss and Gong 2020, Prop. 6). Figure 1 provides an example of such knot placement.

Finally, approximation error can also be avoided when $\mathbf{A}_t = c_t\mathbf{I}_{n_{\mathcal{G}}}$ with $c_t \in \mathbb{R}^+$ and $\mathbf{Q}_t = \mathbf{0}$. In this case we can set $\mathbf{B}_{t|t-1} := \sqrt{c_t}\,\mathbf{B}_{t-1|t-1}$, rather than employing the MRD in the forecast step. In data assimilation, the assumption $\mathbf{Q}_t = \mathbf{0}$ is quite common, when model error is incorporated through multiplicative inflation of the forecast covariance matrix (e.g., Pham, Verron, and Christine Roubaud 1998; Anderson and Anderson 1999).

Aside from these special cases, the MRD and hence the MRF are approximate. However, the MRA, which is the technique underlying the MRD (see Section 5.2), can provide excellent accuracy, as has been shown, for example, by Katzfuss (2017), Katzfuss and Gong (2020), and in a recent comparison of different methods for large spatial data (Heaton et al. 2017). In applications where accuracy is crucial, one could successively increase the number of knots r_m used at low resolutions until the inference "converges." This allows the user to trade off computational cost for the quality of the approximation.

We demonstrate the MRF's accuracy numerically in Sections 7 and 8. In practice, the SSM in (1)–(2) will usually be an approximation to the true system, and we expect the MRD approximation error to often be negligible relative to the error due to model misspecification.

4.2. Computational Complexity

We now determine the computational complexity of the MRF algorithm as a function of $N = \sum_{m=0}^{M} r_m$ under the assumption that $n = \mathcal{O}(n_{\mathcal{G}}) = \mathcal{O}(n_t)$ for all $t = 1,2,\dots$. The results in this section will culminate in precise statements of the memory complexity (end of Section 4.2.1) and the time complexity (Proposition 6) of the MRF, which indicate that the algorithm is highly scalable. Some of the details revealed by the intermediate results are also useful for the design of an efficient implementation of the MRF (cf. Section 3.4.1).

4.2.1. Sparsity and Memory Requirements

As can be seen in Algorithm 2, a multi-resolution factor is composed of block-diagonal submatrices by construction. This leads to a sparse structure and the following proposition quantifies the number of its nonzero elements.

Proposition 1. For a covariance matrix $\mathbf{\Sigma}$, each row of $\mathbf{B} = \mathrm{MRD}(\mathbf{\Sigma})$ has N nonzero elements.

Thus, if $r_m = r$ for $m = 0,\dots,M$, then each row of $\mathbf{B}$ has exactly $(M+1)r$ nonzero elements. Figure 2(a) illustrates this case for $M = 3, J = 3$ and $r = 2$. The MRD results

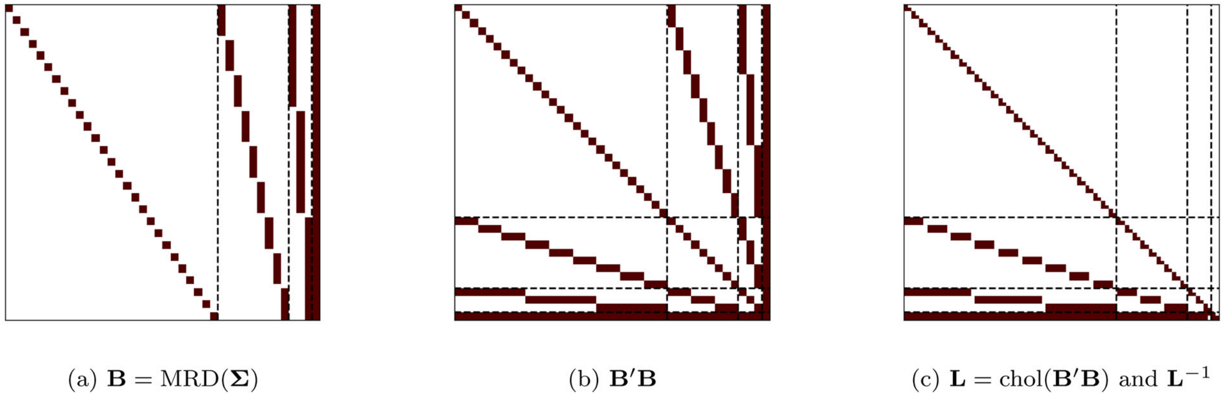

(a) $\mathbf{B} = \text{MRD}(\mathbf{\Sigma})$ (b) $\mathbf{B}'\mathbf{B}$ (c) $\mathbf{L} = \text{chol}(\mathbf{B}'\mathbf{B})$ and $\mathbf{L}^{-1}$

Figure 2. Sparsity patterns for $n_{\mathcal{G}} = 80$, $M = 3$, $J = 3$, and $r_m = 2$ for $m = 0, \ldots, 3$. Rows and columns correspond to hierarchically arranged elements of the grid $\mathcal{G}$ in Figure 1 from resolution $m = 3$ down to $m = 0$. Blocks corresponding to different resolutions are separated by dashed lines. Relating Figure 2(a) to Figure 3, we see the transition from many sparse columns corresponding to many compactly supported basis functions ($m = 3$), to only two dense columns corresponding to two basis functions supported over the entire domain ($m = 0$).

in a convenient structure of the inner product of the multi-resolution factor. The following proposition formally describes the sparsity pattern of this inner product (see Figure 2(b)), while Proposition 3 shows its usefulness in applications to filtering problems.

Proposition 2. Let $\mathbf{B} = \text{MRD}(\mathbf{\Sigma})$ for some covariance matrix $\mathbf{\Sigma}$. Then $\mathbf{B}'\mathbf{B}$ is a block matrix with $M + 1$ row blocks and $M + 1$ column blocks. For $k, l = 0, \ldots, M$ with $k \geq l$, the $(M + 1 - k, M + 1 - l)$th block is of dimension $|\mathcal{K}^k| \times |\mathcal{K}^l|$ and is itself block-diagonal with blocks that are r_l columns wide.

The following technical assumption ensures that both $\mathbf{H}_t$ and $\mathbf{R}_t$ are block-diagonal with blocks corresponding to indices $\mathcal{I}_{j_1,\ldots,j_M}$ within each of the finest subregions:

Assumption 1. Let $i \in \mathcal{I}_{i_1,\ldots,i_M}$ and $j \in \mathcal{I}_{j_1,\ldots,j_M}$. Assume $\mathbf{R}_t[i,j] = 0$, unless $(i_1, \ldots, i_M) = (j_1, \ldots, j_M)$. Further, if $\mathbf{H}_t[i,j] \neq 0$, then $\mathbf{H}_t[i,k] = 0$ for all $k \notin \mathcal{I}_{j_1,\ldots,j_M}$. Finally, if $i_1, i_2 \in \mathcal{I}_{j_1,\ldots,j_M}$ and $i_1 < i_2$, then for all i_3 with $i_1 < i_3 < i_2$, we have $i_3 \in \mathcal{I}_{j_1,\ldots,j_M}$.

This assumption guarantees the key property of the MRF: The sparsity pattern of the multi-resolution factor is preserved in the update step. To formalize this idea, we introduce two new symbols. First, let $\mathcal{S}(\mathbf{G})$ denote the set of matrices whose set of structural zeros is the same or a superset of the structural zeros in some matrix $\mathbf{G}$. Second, we write $\mathbf{G}^L$ to denote the lower triangle of $\mathbf{G}$, meaning that $\mathbf{G}^L[i,j] = \mathbf{G}[i,j]$ if $i \geq j$, and $\mathbf{G}^L[i,j] = 0$ otherwise. We use this notation to formulate Proposition 3, which captures three properties of the MRF algorithm that are later used in the proofs of Propositions 1 and 6.

Proposition 3. Let $\mathbf{B}_{t|t-1}, \mathbf{B}_{t|t}, \mathbf{\Lambda}_t, \mathbf{L}_t$ be defined as in Algorithm 1. Under Assumption 1, we have

1. $\mathbf{\Lambda}_t \in \mathcal{S}(\mathbf{B}'_{t|t-1}\mathbf{B}_{t|t-1})$;
2. $\mathbf{L}_t \in \mathcal{S}(\mathbf{\Lambda}_t^L)$ and $\mathbf{L}_t^{-1} \in \mathcal{S}(\mathbf{\Lambda}_t^L)$;
3. $\mathbf{B}_{t|t} \in \mathcal{S}(\mathbf{B}_{t|t-1})$.

The first part of the result says that the sparsity pattern of the precision matrix $\mathbf{\Lambda}_t$ is equivalent to the sparsity pattern of the inner product of the square roots of the approximate forecast covariance matrix $\mathbf{\Sigma}_{t|t-1}$. This is important because this inner product has a particular sparsity structure shown in Figure 2(b), which in turn results in a Cholesky factor and its inverse with the same sparsity structure (see part two), which can hence be computed cheaply. The last part of the proposition then shows that the sparsity in the forecast basis-function matrix $\mathbf{B}_{t|t-1}$ is preserved in the posterior or filtering matrix $\mathbf{B}_{t|t}$, which allows the MRF to proceed forward through time while preserving the sparsity as detailed here.

We finish analyzing the sparsity properties with a proposition that quantifies the number of nonzero elements in the Cholesky factors of the precision matrix.

Proposition 4. If $\mathbf{L}_t$ is the lower Cholesky factor of $\mathbf{\Lambda}_t$, then each column of $\mathbf{L}_t$ has at most $\mathcal{O}(N)$ nonzero elements.

Figure 2(c) illustrates the sparsity structure of $\mathbf{L}$, of which we take advantage in proving Proposition 6.

The overall takeaway from the sparsity properties discussed above is that all matrices computed in the MRF Algorithm 1 are very sparse, with only $\mathcal{O}(nN)$ nonzero entries. The update step preserves the sparsity, so that $\mathbf{B}_{t|t} \in \mathcal{S}(\mathbf{B}_{t|t-1})$. Due to the Markov structure of order 1 implied by our SSM, there is no need to store matrices from previous time points, and so the memory complexity of the entire MRF algorithm is $\mathcal{O}(nN)$.

4.2.2. Computation Time

For determining the time complexity of the MRF, we assume that the number of knots within each subregion is constant across resolutions (i.e., $r_m = r$ for $m = 0, \ldots, M$) and so $N = (M + 1)r$. While the efficacy of our method does not depend on this assumption, it greatly simplifies the complexity calculations and helps to develop an intuition regarding its computational benefits.

Proposition 5. Given a covariance matrix $\mathbf{\Sigma}$, $\mathbf{B} = \text{MRD}(\mathbf{\Sigma})$ can be computed in $\mathcal{O}(nN^2)$ time using Algorithm 2.

We further assume that the evolution is local, in the sense that the nonzero elements in any given row of $\mathbf{A}_t$ only correspond to grid points in a small number of regions at the finest resolution of the domain partitioning. This assumption often holds when constructing $\mathbf{A}_t$ based on partial differential equations (e.g., see the advection-diffusion in Section 4).

Assumption 2. Assume that the evolution matrix $\mathbf{A}_t$ is sparse with at most $\mathcal{O}(r)$ nonzero elements per row, which must only correspond to a small, constant number of subregions,

$$|\{\mathcal{I}_{j_1,\ldots,j_M} : \exists j \in \mathcal{I}_{j_1,\ldots,j_M} \text{ such that } \mathbf{A}_t[i,j] \neq 0\}| \leq c, \quad i = 1,\ldots,n.$$

For example, for local evolution in two-dimensional space, we have $c \leq 4$.

Proposition 6. Under Assumptions 1 and 2, the MRF in Algorithm 1 requires $\mathcal{O}(nN^2)$ operations at each time step t.

In practice, $N = (M+1)r$ is chosen by the user depending on the required approximation accuracy and the available computational resources. For fixed N, the time and memory complexity of Algorithm 1 are linear in n. If M increases as $M = \mathcal{O}(\log n)$ for increasing n (e.g., Katzfuss 2017) and r is held constant, the resulting complexity is quasilinear.

4.3. Distributed Computation

For truly massive dimensions (i.e., $n_{\mathcal{G}} = \mathcal{O}(10^7)$ or more), memory limitations will typically require distributing the analysis across several computational nodes. The MRF is well suited for such a distributed environment, as information pertaining to different subregions of the domain can be stored and processed in separate nodes, with limited communication overhead required between nodes. We plan to leverage these properties of the MRF by designing an implementation of Algorithm 1 that can be deployed in a high-performance-computation environment. We include further details in Section S2.

4.4. Forecasting and Smoothing

Forecasting is straightforward using the MRF. Given the filtering distribution $\mathbf{x}_T|\mathbf{y}_{1:T}$ as obtained by Algorithm 1, we can compute the k-step-ahead forecast $\mathbf{x}_{T+k}|\mathbf{y}_{1:T}$ by simply carrying out the forecast step in Algorithm 1 k times, while skipping the update step. More precisely, we carry out Algorithm 1 for $t = T+1,\ldots,T+k$, but at each time point t, we replace Step 2 by simply setting $\boldsymbol{\mu}_{t|t} = \boldsymbol{\mu}_{t|t-1}$ and $\mathbf{B}_{t|t} = \mathbf{B}_{t|t-1}$. The accuracy of such forecasts will depend heavily on the quality of the evolution matrices $\mathbf{A}_t$, and so a physics-informed evolution can result in much better forecasts than simple models such as random walks.

In some applications, one might also be interested in obtaining retrospective smoothing distributions $\mathbf{x}_t|\mathbf{y}_{1:T}$ for $t < T$. These can be computed exactly by carrying out the Kalman filter up to time T, and then carrying out recursive backward smoothing (e.g., Rauch, Striebel, and Tung 1965), but this is not feasible for large grids. It is challenging to extend the MRF by deterministic backward-smoothing operations that preserve sparsity, but it may be possible to devise a scalable MRF-based forward-filter-backward-sampler algorithm. We intend to investigate this modification in future work.

5. Connections to Existing Methods

In this section, we discuss in some detail the connections between our MRF and hierarchical matrix decompositions and basis-function approximations. Further, in Section S3, we discuss connections to multi-resolution autoregressive models, which demonstrate that the MRF can also be interpreted as a nested Kalman filter that proceeds over resolutions within each outer filtering step over time.

5.1. MRD as Hierarchical Low-Rank Decomposition

Hierarchical off-diagonal low-rank (HODLR) matrices are a popular tool in numerical analysis, and they have recently also been applied to Gaussian processes (e.g., Ambikasaran and Darve 2013; Ambikasaran et al. 2016). In HODLR matrices, the off-diagonal blocks are recursively specified or approximated as low-rank matrices. In this section, we show the connection between the HODLR format and the MRD when $J = 2$.

Definition 1 (Ambikasaran et al. 2016). A matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ is termed a 1-level hierarchical off-diagonal low-rank (HODLR) matrix of rank p, if it can be written as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_1^{(1)} & \mathbf{U}_1^{(1)}(\mathbf{V}_1^{(1)})' \\ \mathbf{U}_2^{(1)}(\mathbf{V}_2^{(1)})' & \mathbf{K}_2^{(1)} \end{bmatrix},$$

where $\mathbf{K}_i^{(1)} \in \mathbb{R}^{N/2\times N/2}$, and $\mathbf{U}_i^{(1)}, \mathbf{V}_i^{(1)} \in \mathbb{R}^{N/2\times p}$. We call $\mathbf{K}$ an m-level HODLR matrix of rank p if both diagonal blocks are $(m-1)$-level HODLR matrices of rank p.

If we use H_m^p to denote the set of all m-level HODLR matrices of rank p, then it follows that $H_m^p \subset H_{m-1}^p$. The optimal low-rank representation is obtained by specifying the matrices $\mathbf{U}_i^{(j)}$ and $\mathbf{V}_i^{(j)}$ as the first p singular vectors of the corresponding off-diagonal submatrix (Hogben 2006, Item 5.6.13), but this is prohibitively expensive. Ambikasaran et al. (2016) discuss multiple ways of approximating this low-rank representation.

We now show that the outer product of an MRD factor is a HODLR matrix, specifically one in which the low-rank approximations are obtained as skeleton factorizations.

Proposition 7. Let $\mathbf{B} = \text{MRD}(\boldsymbol{\Sigma})$, where the decomposition is based on a partitioning scheme with $J = 2$ and $r_m = r$ for $m = 0,\ldots,M$. Then, $\mathbf{B}\mathbf{B}' \in H_M^r$.

The proof is given in Appendix A. It can easily be extended to r_m varying by resolution. Thus, the MRF approximation of the prior covariance matrix, $\boldsymbol{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}_{t|t-1}'$, is a HODLR matrix (Ambikasaran et al. 2016). In contrast to previous approaches using HODLR matrices for spatio-temporal models (e.g., Li et al. 2014; Saibaba, Miller, and Kitanidis 2015), the block-sparse MRD matrices allow the MRF to handle nondiagonal evolution matrices $\mathbf{A}_t$ and full-rank model-error matrices $\mathbf{Q}_t$.

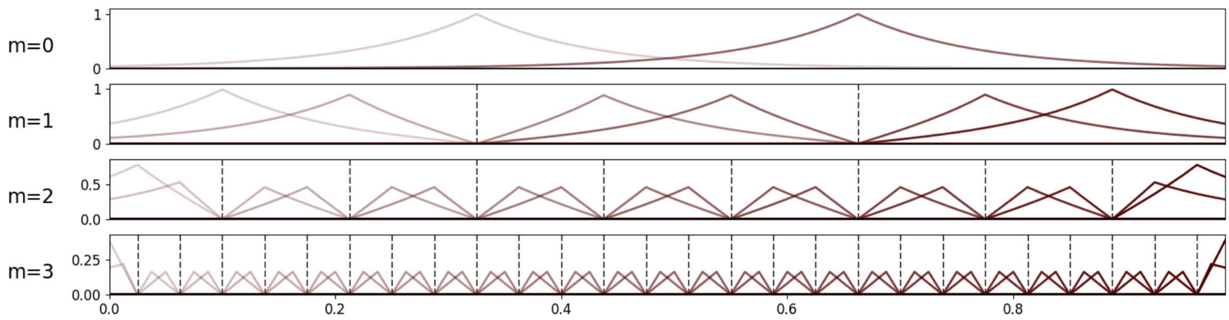


Figure 3. Basis functions obtained by interpolating the entries in each column of $\mathbf{B} = \text{MRD}(\mathbf{\Sigma})$ in Figure 2(a) using the grid from Figure 1, with $\mathbf{\Sigma}$ based on an exponential covariance with range 0.3. Each basis function's support is restricted to one of the subregions (dashed lines) at each resolution.

5.2. *MRD as Basis-Function Approximation*

The MRD is related to the multi-resolution approximation (MRA; Katzfuss 2017) of a Gaussian process as a weighted sum of increasingly compactly supported basis functions at M resolutions. While the MRD adapts the MRA to an approximate decomposition of a covariance matrix evaluated at a spatial grid, $\mathbf{\Sigma} = \mathbf{BB}'$, we can similarly interpret each column of $\mathbf{B}$ as a basis vector over the grid. In other words, the spatial field $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is approximated as $\mathbf{x} \approx \mathbf{B}\boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the vector of independent standard normal weights. By interpolating the values of the basis vectors between grid points, we can visualize the columns of $\mathbf{B}$ as basis functions, which is illustrated in Figure 3.

The basis functions obtained in this way exhibit interesting properties. Their support is increasingly compact as the resolution increases, and basis functions at low resolution capture the large-scale structure. There are strong connections between the MRD and stochastic wavelets, with the major difference that the shape of the basis functions in the MRD adapts to the covariance structure in $\mathbf{\Sigma}$. This adaptation is especially useful in the spatio-temporal context here, which requires approximation of the forecast covariance matrix $\mathbf{S}_{t|t-1}$ that depends on the data at previous time points and is hence highly nonstationary. The compact support stems from the assumption of a block-sparse structure at each resolution in the MRD, which is equivalent to assuming that the remainder at each resolution is conditionally independent between subregions at that resolution, given the terms at lower resolutions. In general, this assumption is not satisfied and thus produces an approximation error, although the MRD is exact in some special cases (see Section 4.1).

6. Parameter Inference

If there are random, time-varying parameters $\boldsymbol{\theta}_t$ in any of the matrices in (1)–(2), that are distinct from the Gaussian state $\mathbf{x}_t$, we can make inference on these parameters using an approximate Rao-Blackwellized particle filter (Doucet et al. 2000), in which we use the MRF algorithm to approximately integrate out the high-dimensional state $\mathbf{x}_t$ at each time point. An alternative approach, based on including the unknown parameters in the state vector, otherwise known as data augmentation, tends to work poorly for certain parameters and thus is less general (e.g., DelSole and Yang 2010; Katzfuss, Stroud, and Wikle 2020).

To derive our filter, note that we have

$$p(\mathbf{y}_{1:T}|\boldsymbol{\theta}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{1:t}) =: \prod_{t=1}^{T} \mathcal{L}_t(\boldsymbol{\theta}_{1:t}),$$

where, after integrating out $\mathbf{x}_t$, we have $\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{1:t} \sim \mathcal{N}_{n_t}(\mathbf{H}_t\boldsymbol{\mu}_{t|t-1}, \mathbf{H}_t\mathbf{\Sigma}_{t|t-1}\mathbf{H}_t' + \mathbf{R}_t)$ with $\mathbf{\Sigma}_{t|t-1} = \mathbf{B}_{t|t-1}\mathbf{B}_{t|t-1}'$. By applying a matrix determinant lemma (e.g., Harville 1997, Thm. 18.1.1) to the determinant and the Sherman–Morrison–Woodbury formula to the quadratic form in the multivariate normal density, it is straightforward to show that the integrated filtering likelihood at time t can be written, up to a constant, as

$$\begin{aligned} -2\log\mathcal{L}_t(\boldsymbol{\theta}_{1:t}) = 2\log|\mathbf{L}_t| + \log|\mathbf{R}_t| + (\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1})'\mathbf{R}_t^{-1} \\ (\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1}) - \tilde{\mathbf{y}}_t'\tilde{\mathbf{y}}_t, \end{aligned} \tag{6}$$

where $\tilde{\mathbf{y}}_t := \mathbf{B}_{t|t}'\mathbf{H}_t'\mathbf{R}_t^{-1}(\mathbf{y}_t - \mathbf{H}_t\boldsymbol{\mu}_{t|t-1})$, and we have omitted dependence on the parameters $\boldsymbol{\theta}_{1:t}$ for the terms on the right-hand side.

Assuming that the priors for the $\boldsymbol{\theta}_t$ are given by $p_0(\boldsymbol{\theta}_0)$ for $t = 0$, and recursively by $p_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ for $t = 1, 2, \ldots$, the particle MRF proceeds as follows:

Algorithm 3: Particle MRF.
At time $t = 0$, for $i = 1, \ldots, N_p$, draw $\boldsymbol{\theta}_0^{(i)} \sim p_0(\boldsymbol{\theta}_0)$ with equal weights $w_0^{(i)} = 1/N_p$, and compute $\boldsymbol{\mu}_{0|0}^{(i)} = \boldsymbol{\mu}_{0|0}(\boldsymbol{\theta}^{(i)})$ and $\mathbf{B}_{0|0}^{(i)} = \text{MRD}(\mathbf{\Sigma}_{0|0}(\boldsymbol{\theta}_0^{(i)}))$. Then, for each $t = 1, 2, \ldots$, do:

- For $i = 1, \ldots, N_p$:
    - Sample $\boldsymbol{\theta}_t^{(i)}$ from a proposal distribution $q_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1}^{(i)})$.
    - Forecast step: Compute $\boldsymbol{\mu}_{t|t-1}^{(i)} = \mathbf{A}_t(\boldsymbol{\theta}_t^{(i)})\boldsymbol{\mu}_{t-1|t-1}^{(i)}$, ${\mathbf{B}_{t|t-1}^{F}}^{(i)} = \mathbf{A}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{B}_{t-1|t-1}^{(i)}$, and $\mathbf{B}_{t|t-1}^{(i)} = \text{MRD}(\mathbf{\Sigma}_{t|t-1}^{(i)})$, where ${\mathbf{B}_{t|t-1}^{F}}^{(i)}({\mathbf{B}_{t|t-1}^{F}}^{(i)})' + \mathbf{Q}_t(\boldsymbol{\theta}_t^{(i)}))$.
    - Update step: Compute $\mathbf{\Lambda}_t^{(i)} = \mathbf{I}_{n_{\mathcal{G}}} + \mathbf{B}_{t|t-1}^{(i)}{}'\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})'\mathbf{R}_t(\boldsymbol{\theta}_t^{(i)})^{-1}\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\mathbf{B}_{t|t-1}^{(i)}$, $\mathbf{L}_t^{(i)}$ as the lower Cholesky triangle of $\mathbf{\Lambda}_t^{(i)}$, $\mathbf{B}_{t|t}^{(i)} = \mathbf{B}_{t|t-1}^{(i)}((\mathbf{L}_t^{(i)})^{-1})'$, and $\boldsymbol{\mu}_{t|t}^{(i)} = \boldsymbol{\mu}_{t|t-1}^{(i)} + \mathbf{B}_{t|t}^{(i)}\mathbf{B}_{t|t}^{(i)}{}'\mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})'\mathbf{R}_t(\boldsymbol{\theta}_t^{(i)})^{-1}(\mathbf{y}_t - \mathbf{H}_t(\boldsymbol{\theta}_t^{(i)})\boldsymbol{\mu}_{t|t-1}^{(i)})$.

  - Using the quantities just computed for $\boldsymbol{\theta}_t^{(i)}$, calculate $\mathcal{L}_t(\boldsymbol{\theta}_{1:t}^{(i)})$ as in (6), and update the particle weight $w_t^{(i)} \propto w_{t-1}^{(i)} \mathcal{L}_t(\boldsymbol{\theta}_{1:t}^{(i)}) p_t(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{t-1}^{(i)}) / q_t(\boldsymbol{\theta}_t^{(i)} | \boldsymbol{\theta}_{t-1}^{(i)})$.

- The filtering distribution is $p(\boldsymbol{\theta}_t, \mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{i=1}^{N_p} w_t^{(i)} \delta_{\boldsymbol{\theta}_t^{(i)}}(\boldsymbol{\theta}_t) \mathcal{N}_{n_\mathcal{G}}(\mathbf{x}_t | \boldsymbol{\mu}_{t|t}^{(i)}, \mathbf{B}_{t|t}^{(i)} \mathbf{B}_{t|t}^{(i)\prime})$.
- If desired, resample the triplets $\{(\boldsymbol{\theta}_t^{(i)}, \boldsymbol{\mu}_{t|t}^{(i)}, \mathbf{B}_{t|t}^{(i)}) : i = 1, \ldots, N_p\}$ with weights $w_t^{(1)}, \ldots, w_t^{(N_p)}$, respectively, to obtain an equally weighted sample (see, e.g., Douc, Cappé, and Moulines 2005, for a comparison of resampling schemes).

Section S5 presents numerical experiments demonstrating the accuracy of Algorithm 3 and its advantage over a low-rank particle filter. While a general discussion of the many aspects of prior choice and particle filtering is difficult, our particle MRF can often provide a close approximation to a Rao-Blackwellized particle filter, for which there exists a big body of literature describing its application in different settings. For example, Martín-Fernández et al. (2014) demonstrated how such a filter can be used to analyze ecological systems, Martín-Fernández and Lanzarone (2015) employed it in the context of a heat-transfer model, and Doucet et al. (2013) discussed its extension to dynamic Bayesian networks.

7. Simulation Study

We used simulated data to compare the performance of the MRF with several filtering methods:

KF: The Kalman filter (see Section 2.2) provides the exact filtering distributions, but has $\mathcal{O}(n^3)$ time complexity at each time point.

MRF: The multi-resolution filter proposed here in Section 3, with $\mathcal{O}(nN^2)$ time complexity, where $N = \sum_{m=0}^{M} r_m$.

EnKF: An ensemble Kalman filter with stochastic updates (e.g., Katzfuss, Stroud, and Wikle 2016, sec. 3.1). We set the ensemble size to N and use Kanter's function (Kanter 1997) for tapering such that the resulting matrix has N nonzero entries per row. This results roughly in $\mathcal{O}(nN^2)$ time complexity (e.g., Tippett et al. 2003).

LRF: A low-rank-plus-diagonal filter that can be viewed as a spatio-temporal extension of the modified predictive process (Finley et al. 2009) and as a special case of a fixed-rank filter (Cressie, Shi, and Kang 2010). Moreover, it can be viewed as a special case of the MRF (hence allowing for ease of comparison) with $M = 1$ resolutions and N knots at resolution 0, where each grid point is in its own partition at resolution 1, resulting in a time complexity of $\mathcal{O}(nN^2)$.

MRA: The MRA (Katzfuss 2017) is a spatial-only method. It can essentially be viewed as a special case of the MRF, for which the filtering distribution at each time t is obtained by assuming that only $\mathbf{y}_t$ and no data at previous time points are available. It has the same $\mathcal{O}(nN^2)$ complexity as the MRF.

While the KF provides the exact filtering distributions, it is only computationally feasible due to the deliberately small problem size chosen here. All other methods attempt to approximate the exact KF solution, but have the advantage of being scalable to much larger grid sizes. For a fair comparison, all approximate methods used the same N, which trades off approximation accuracy and computational complexity. Further, we acknowledge that the EnKF was designed for nonlinear evolution in operational data assimilation, and it is thus more widely applicable than the other methods.

We used two criteria to compare the performance of the approximate filters: the Kullback–Leibler (KL) divergence between the true and approximated filtering distribution of the state vector (i.e., the joint distribution for the entire spatial field), and the ratio of the root mean squared prediction error (RMSPE) achieved by each approximate method relative to the RMSPE of the KF. Detailed definitions of the criteria can be found in Section S4.1. Lower is better for both criteria, with optimal values of 0 for the KL divergence and 1 for the RMSPE ratio. In addition, Section S4.4 examines the performance of all methods in terms of the confidence-interval coverage. All quantities were averaged over 50 simulated datasets.

7.1. One-Dimensional Circular Domain

In our first simulation scenario, we considered a diffusion-advection model on a one-dimensional domain consisting of a circle with a unit circumference. After discretizing both the spatial and the temporal dimensions using $n_\mathcal{G} = 80$ and $T = 20$ regularly spaced points, respectively, we obtained a linear model as in (1)–(2), where $\mathbf{A}_t$ was a tri-diagonal matrix and $\mathbf{Q}_t = \sigma_w^2 \left[\mathcal{M}_{\nu,\lambda}(s_i, s_j)\right]_{i,j=1,\ldots n_\mathcal{G}}$ was based on a Matérn correlation function $\mathcal{M}_{\nu,\lambda}(\cdot,\cdot)$ with smoothness ν and range λ. At each time point, we randomly selected n_t observed locations, so that $\mathbf{H}_t$ is a subset of the identity, and we set $\mathbf{R}_t = \sigma_v^2 \mathbf{I}_{n_t}$. A detailed description of the simulation, including examples of process realizations, is given in Section S4.2.

Because of the many possible choices of parameters, we first established baseline settings that we considered relevant for practical applications, and then examined the effects of changing them one by one. The resulting simulation scenarios are detailed in Table 1. For the MRD, we set $M = 3$, $J = 3$, and $r_m = 2$ for all m, and so we used $N = (3 + 1)2 = 8$ for EnKF, LRF, and MRA.

As shown in Figure 4, the MRF performed best in all four scenarios, both in terms of the KL divergence and the RMSPE ratio.

Table 1. Settings used in the one-dimensional simulation.

	$n_t/n_\mathcal{G}$	ν	λ	σ_w^2	σ_v^2
Baseline	0.3	0.5	0.1	0.5	0.05
Smooth	0.3	**1.5**	0.1	0.5	0.05
Dense obs.	**0.8**	0.5	0.1	0.5	0.05
Low noise	0.3	0.5	0.1	0.5	**0.01**

NOTE: Bold values indicate changes with respect to the baseline.

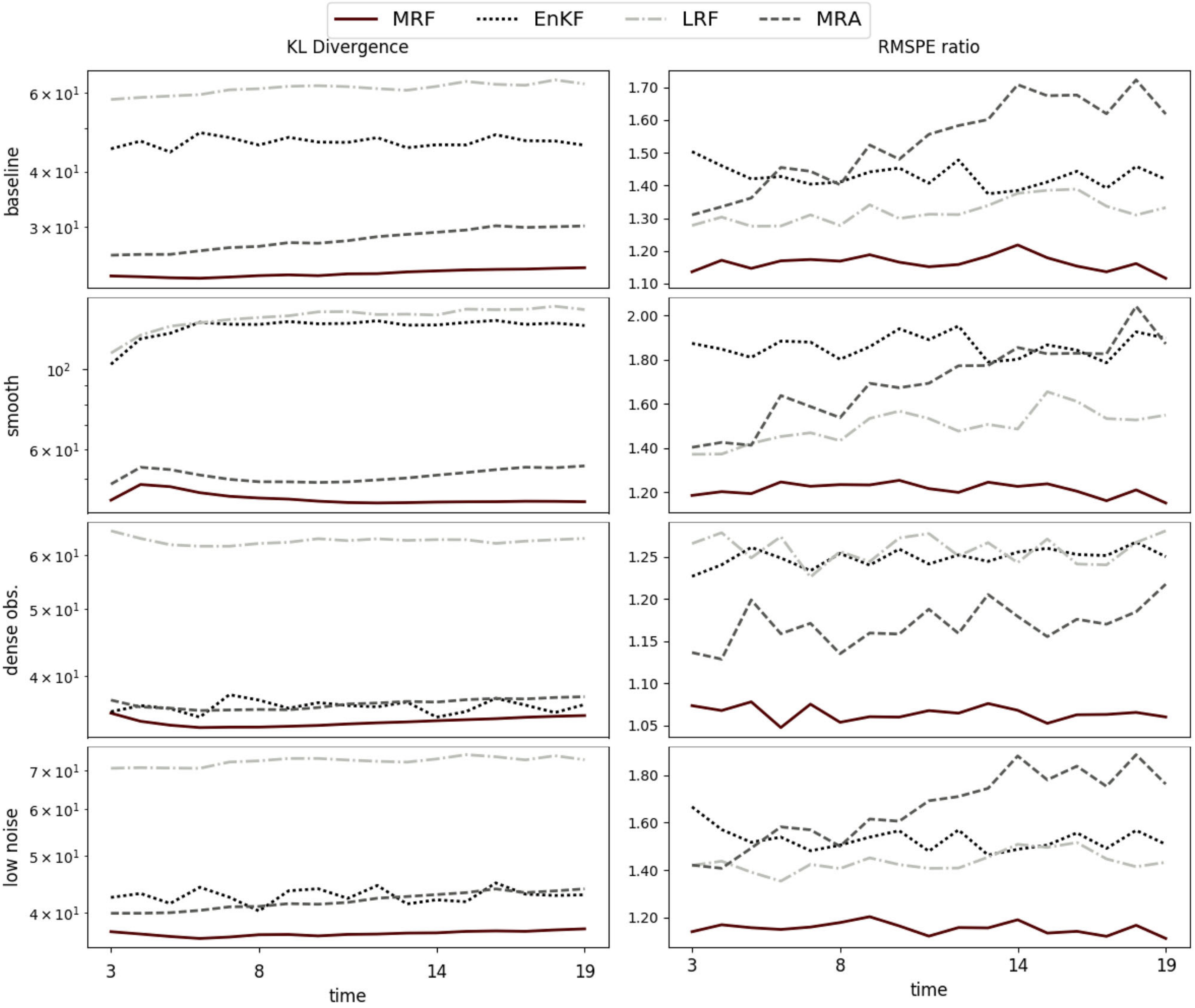


Figure 4. Filter scores for different parameter settings; one-dimensional domain. Note that we used different scales on the vertical axis for each plot, with a logarithmic scale for the KL divergence.

7.2. Two-Dimensional Domain

We also considered a diffusion-advection model on a unit square, and we discretized it on a regular grid of size $n_{\mathcal{G}} = 34 \times 34 = 1156$. As before, we used $T = 20$ evenly spaced time points. Writing the model in the linear form (1)–(2), $\mathbf{A}_t$ was a sparse matrix with nonzero entries corresponding to interactions between neighboring grid points to the right, left, top, and bottom. A detailed description of the simulation, including examples of process realizations, is given in Section S4.3.

Similar to the one-dimensional case, we first considered baseline parameter settings and then we changed some of them, one at a time (see Table 2). The MRD used $M = 4$ and, similar to Katzfuss (2017) we changed J_m across resolutions m: $(J_1, \dots, J_4) = (2, 4, 4, 4)$. We also varied the numbers of knots r_m used at each resolution: $(r_0, \dots, r_4) = (16, 8, 6, 6, 6)$. Thus, to achieve a fair comparison, we used $N = 42$ for EnKF, LRF, and MRA. As shown in Figure 5, MRF again performed best in all four scenarios.

Table 2. Settings used in the two-dimensional simulation.

	$n_t/n_{\mathcal{G}}$	ν	λ	σ_w^2	σ_v^2
Baseline	0.1	0.5	0.15	0.5	0.25
Smooth	0.1	**1.5**	0.15	0.5	0.25
Dense obs.	**0.3**	0.5	0.15	0.5	0.25
Low noise	0.1	0.5	0.15	0.5	**0.1**

NOTE: Bold values indicate changes with respect to the baseline.

8. Sediment Movements in Lake Michigan

We also conducted filtering inference on sediment concentration in Lake Michigan over a period of one month, March 1998, based on satellite data. Such inference can be used by hydrologists to increase their understanding of sediment transport mechanisms and fine-tune existing domain-specific models. We used data from Stroud et al. (2010), which are publicly available online at the journal's page, and closely followed their earlier study of this problem in the context of spatio-temporal smoothing. Unless specified otherwise, we used the same model and parameter settings that were carefully determined for this

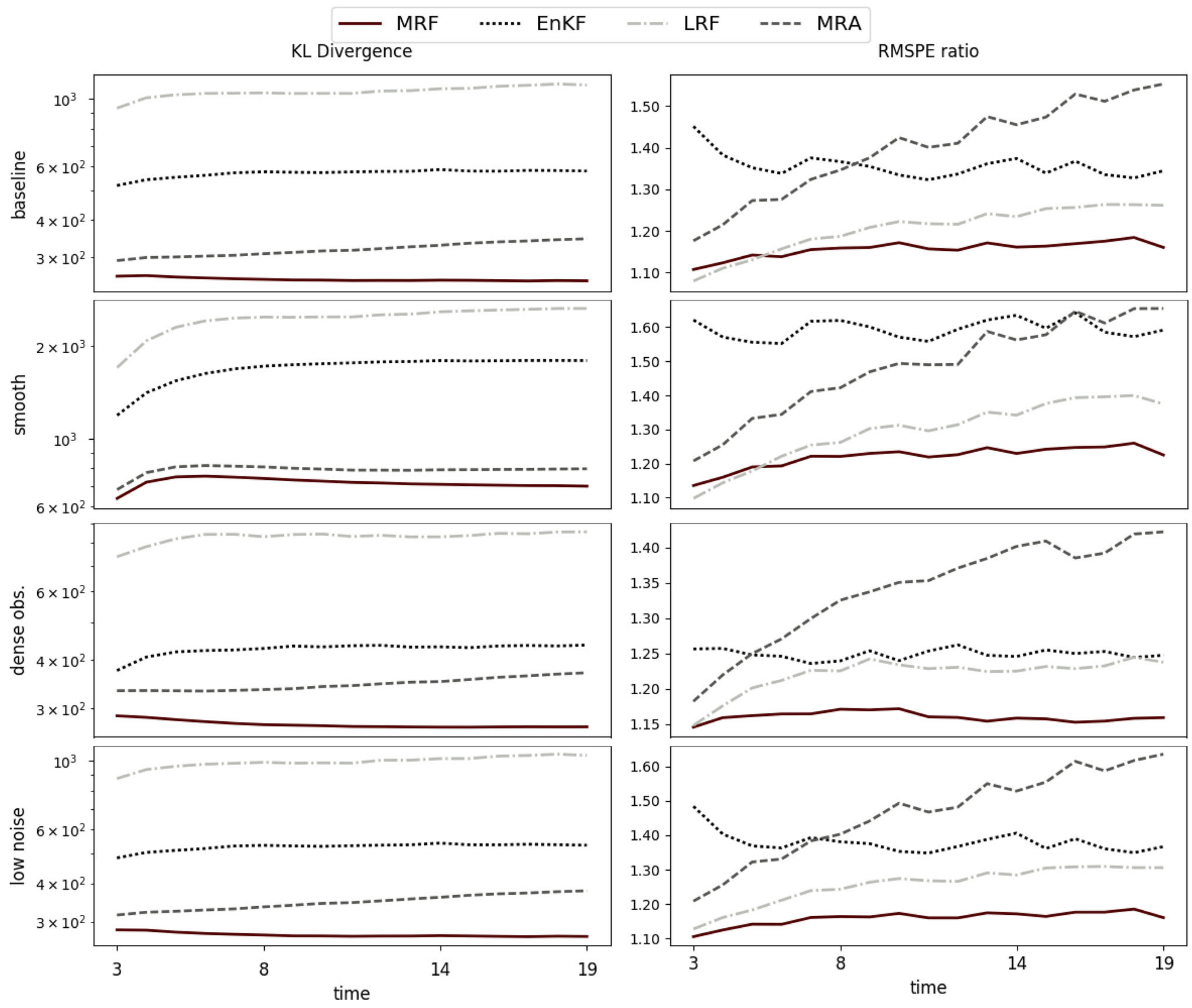


Figure 5. Filter scores for different parameter settings; two-dimensional domain. Note that we used different scales on the vertical axis for each plot, with a logarithmic scale for the KL divergence.

application in collaboration with domain experts in Stroud et al. (2010). Hence, we focused on inferring the state of the spatial field based on Algorithm 1, and we did not need the particle MRF in Algorithm 3. We briefly summarize the general framework below and indicate the few modifications we introduced.

The lake area was divided into $n_{\mathcal{G}} = 14{,}558$ grid cells of size $2\,\text{km} \times 2\,\text{km}$ each. We use $\mathbf{x}_t$ to denote the sediment concentrations at the $n_{\mathcal{G}}$ cells at time t. The time dimension was discretized into 409 intervals. The sediment transport model was assumed to be $\mathbf{x}_t = \mathbf{A}_t\mathbf{x}_{t-1} + \boldsymbol{\rho}_t + \mathbf{w}_t$, where $\mathbf{A}_t$ describes the temporal evolution based on a hydrological PDE model provided to us by Stroud et al. (2010), $\boldsymbol{\rho}_t$ is a vector with external inputs representing the influence of water velocity and bottom sheer stress, and the model error $\mathbf{w}_t$ is assumed to follow a $\mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ distribution with covariance matrix $\mathbf{Q}_t = (\sigma_\omega^2 \boldsymbol{\Omega}_t \boldsymbol{\Omega}_t') \circ \mathbf{T}$, where $\circ$ denotes element-wise multiplication. All matrices $\boldsymbol{\Omega}_t$ have dimensions $n_{\mathcal{G}} \times 5$ and reflect the spatial structure of the error in the original study, while $\mathbf{T}$ is taken to be a tapering matrix based on a Kanter covariance function with a tapering radius that leaves about 200 nonzero elements in each row. We used $\sigma_\omega = 0.063$, as in the original study.

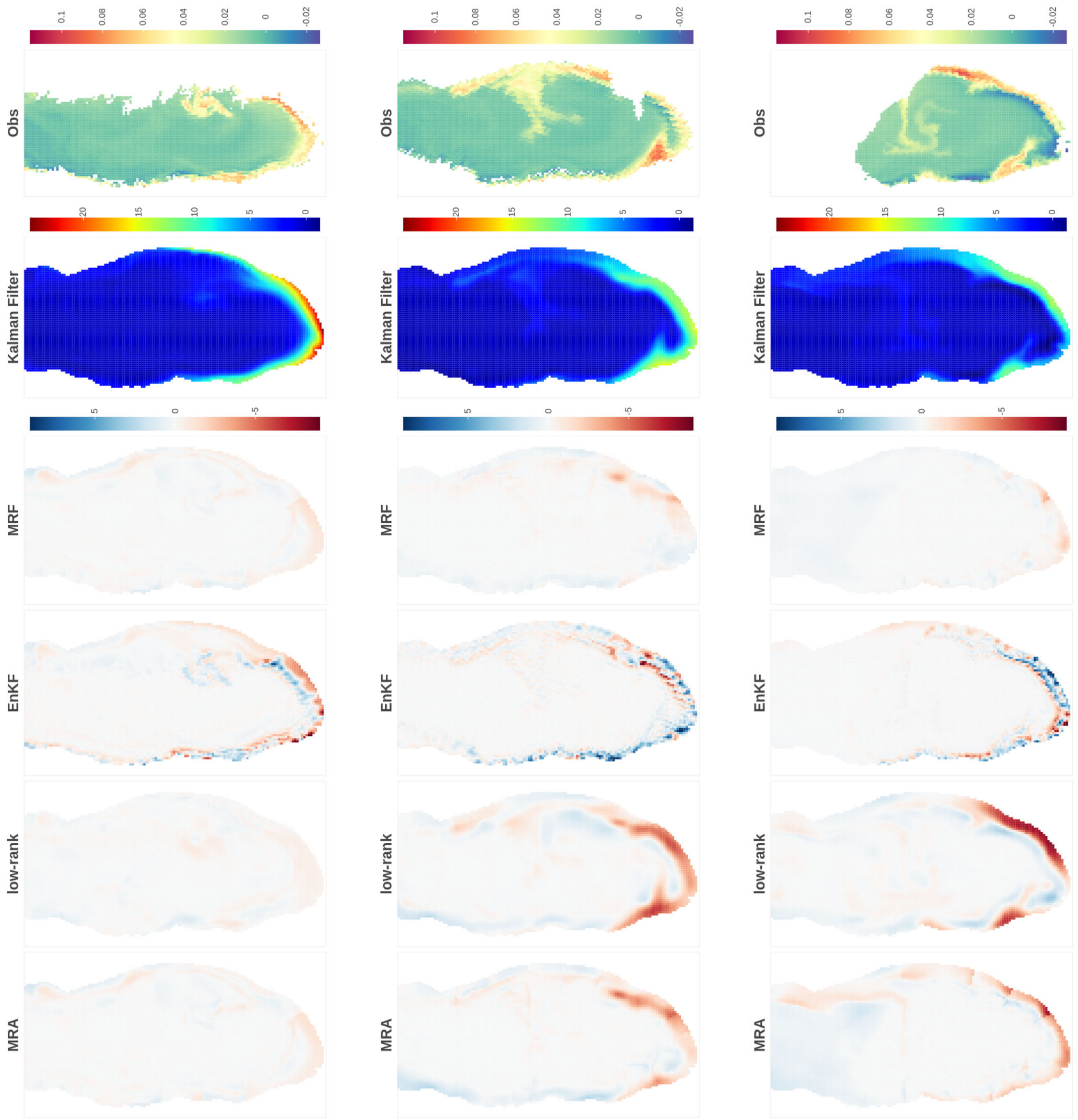
The data comprise 10 satellite measurements of remote-sensing reflectance (RSR) at the frequency of 555 nm taken over the southern basin of Lake Michigan, modified in a way that accounts for the effects of the cloud cover. The observed value at each grid point was assumed to be the first-order Taylor expansion of $h(c) = \theta_0 + \theta_1 \log(1 + \theta_2(c + \theta_3))$ taken around the initial mean of the sediment concentration at time $t = 0$. We took the parameter vector $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$ to be $(0.003, 0.054, 0.474, 0.55)$, the same as in Stroud et al. (2010). Using $\mathbf{y}_t$ to denote the vector of observations at time t after removing a time-varying instrument bias and accounting for constant terms in the Taylor expansion, we assumed $\mathbf{y}_t = \mathbf{H}_t\mathbf{x}_t + \mathbf{v}_t$ as in (1), where $\mathbf{H}_t$ had only one nonzero element in each row, $\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$, and $\mathbf{R}_t = \sigma_v^2\mathbf{I}$ was diagonal, where $\sigma_v = 0.007$, as in Stroud et al. (2010).

Because of the moderate size of the spatial grid, we were able to compute the exact Kalman filter solution. We set $M = 5$, $J = 4$, and $(r_0, \dots, r_5) = (16, 8, 8, 8, 4, 4)$ for the MRF, which implied that $N = \sum_m r_m = 48$ for the other approximation methods in Section 7. The tapering range used in EnKF was selected such that the tapering matrix had only 5 nonzero elements per row, which corresponds to the setting used by Stroud et al. (2010). While this is inconsistent with the comparison principles outlined in Section 7, it made the EnKF perform better in this case.

Table 3. Root average squared difference (RASD) between approximate and exact filtering means for sediment concentration.

	MRF	EnKF	LRF	MRA
RASD	0.08	0.22	0.42	0.72

As the true concentrations were unknown, we compared the approximate filtering means to the exact means obtained by the Kalman filter, using the root average squared difference $\left(\sum_t \sum_i (\hat{\boldsymbol{\mu}}_t[i] - \boldsymbol{\mu}_t^{KF}[i])^2\right)^{1/2}$ between the approximate filtering means $\hat{\boldsymbol{\mu}}_t[i]$ and the KF means $\boldsymbol{\mu}_t^{KF}[i]$, averaged over all times t and grid points i. The results, reported in Table 3, show the MRF



(a) t = 2 (b) t= 267 (c) t=409

Figure 6. Satellite data (in log RSR) and exact Kalman filtering means of sediment concentrations (in mg/L), along with differences of approximate filtering means to the Kalman filter. We display the results for the southern basin of the lake only, where differences between the methods are most pronounced.

outperforming all other approximate methods. To visually verify these results, we also present satellite data and sediment concentration estimates for three selected time points in Figure 6. A video with all time points can be found at *http://spatial.stat.tamu.edu*.

For a grid of the size $n_\mathcal{G}$ considered here, a single step of the MRF took roughly 5% of the time required by the exact Kalman filter (on a laptop with 8GB of memory and Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz). More importantly, the MRF scales well to even larger grid sizes (see Section 4.2), while exact calculations will quickly become infeasible due to memory constraints. Exact computation times and memory limitations will, of course, depend heavily on the computational environment.

9. Conclusions and Future Work

We introduced the MRF, a new filtering method for linear Gaussian spatio-temporal SSMs, which relies on a block-sparse multi-resolution matrix decomposition. We proved that the sparsity can be preserved under filtering through time, ensuring scalability of the MRF to very large spatial grids. In our comparisons, the MRF substantially outperformed existing methods that can be used to approximate the Kalman filter. We also successfully applied the MRF to inferring sediment concentration in Lake Michigan.

Spatio-temporal processes typically exhibit highly complicated structures that make exact inference intractable, especially in high dimensions. We believe that it is often better to conduct approximate inference for a realistic, intractable model, rather than carrying out “exact” inference for a crude simplification (e.g., a low-rank version) of the model. While it might be challenging to precisely quantify the approximation accuracy in the former case (e.g., for the MRF), approximate inference can give better results than exact inference in a simplified model, which often completely ignores the error incurred by simplifying the model.

While we have focused on spatio-temporal data here, our methods are also applicable to general SSMs of the form (1)–(2) that do not correspond to physical space and time, as long as some distance between the elements of each state vector can be specified.

Potential future work includes extensions to non-Gaussian data, nonlinear evolution, and smoothing inference. We are also developing a user-friendly implementation of the MRF with sensible default settings for the number of knots and domain partitioning.

Appendix A. Proofs

We now provide proofs for the propositions stated throughout the article. We simplify notation by dropping most time subscripts; to avoid confusion, we denote $\mathbf{B}_{t|t-1}$ by $\mathbf{B}$, and $\mathbf{B}_{t|t}$ by $\tilde{\mathbf{B}}$. In Section S8, we provide lemmas with proofs that are used in the proof of Proposition 3 here. Sections S6 and S7 contain additional technical concepts used in the lemmas, including a review of basic ideas from graph theory, hierarchical-matrix theory, and some illustrative figures. Finally, throughout this appendix, if $\mathbf{G}$ is a square matrix, we use $\mathbf{G}^L$ and $\mathbf{G}^U$ to denote its lower and upper triangles, respectively.

Proof of Proposition 1. Recall that $\mathbf{B} = \left(\mathbf{B}^M, \mathbf{B}^{M-1}, \dots, \mathbf{B}^0\right)$. This lets us write the ith row of $\mathbf{B}$ as $\mathbf{B}[i,:] = \left(\mathbf{B}^M[i,:], \mathbf{B}^{M-1}[i,:], \dots, \mathbf{B}^0[i,:]\right)$. By construction, each block $\mathbf{B}^m$ is block-diagonal and such that for $m \leq M$, each segment $\mathbf{B}^m[i,:]$ has only r_m nonzero elements. Because each row of $\mathbf{B}$ is composed of $M+1$ blocks $\mathbf{B}^m[i,:]$ for $m = 0, \dots, M$, this ends the proof. □

Proof of Proposition 2. Direct calculation shows that $\mathbf{B}'\mathbf{B}$ is a block matrix consisting of $(M+1) \times (M+1)$ blocks with $(M-k+1, M-l+1)$th block $(\mathbf{B}^k)'\mathbf{B}^l$. Since for each j the matrix $\mathbf{B}^j$ has dimensions $n_\mathcal{G} \times |\mathcal{K}^j|$ it follows that $(\mathbf{B}^k)'\mathbf{B}^l$ is of size $|\mathcal{K}^k| \times |\mathcal{K}^l|$. Note that $\mathbf{B}^k$ and $\mathbf{B}^l$ are block-diagonal with blocks of size $|\mathcal{I}_{j_1,\dots,j_k}| \times r_k$ and $|\mathcal{I}_{j_1,\dots,j_l}| \times r_l$, respectively. Assuming without loss of generality that $k \leq l$, we have that

$$\begin{aligned} \mathcal{I}_{j_1,\dots,j_k} &= \bigcup_{j_{k+1}=1}^{J} \cdots \bigcup_{j_l=1}^{J} \mathcal{I}_{j_1,\dots,j_\ell}, \\ \implies \quad |\mathcal{I}_{j_1,\dots,j_k}| &= \sum_{j_{k+1}=1}^{J} \cdots \sum_{j_l=1}^{J} |\mathcal{I}_{j_1,\dots,j_\ell}|. \end{aligned} \tag{7}$$

Thus, $\mathbf{B}^l$ can be viewed as a block-diagonal matrix with blocks of height $|\mathcal{I}_{j_1,\dots,j_k}|$. We can also determine their width to be $w_{j_1,\dots,j_k} = \sum_{j_{k+1}=1}^{J} \cdots \sum_{j_l=1}^{J} |\mathcal{K}_{j_1,\dots,j_k,j_{k+1},\dots j_l}|$. This means $(\mathbf{B}^k)'\mathbf{B}^l$ is the product of two block-diagonal matrices with matching block sizes. Therefore, the product will be also block-diagonal with blocks of dimensions $w_{j_1,\dots,j_k} \times r_k$. □

Proof of Proposition 3.

1. Observe that under Assumption 1, $\mathbf{R}^{-1}$ and $\mathbf{H}$ are block-diagonal with blocks of matching dimensions. Since $\mathbf{R}^{-1}$ has square blocks, we conclude that $\mathbf{H}'\mathbf{R}^{-1} \in \mathcal{S}(\mathbf{H}')$. Thus, if $\tilde{\mathbf{R}}^{-1} := \mathbf{H}'\mathbf{R}^{-1}\mathbf{H}$, then $\tilde{\mathbf{R}}^{-1} \in \mathcal{S}(\mathbf{H}'\mathbf{H})$. The latter is a block-diagonal matrix with square blocks of size $|\mathcal{I}_{j_1,\dots,j_M}|$.

   Next, we demonstrate that $\mathbf{B}'\tilde{\mathbf{R}}^{-1}\mathbf{B} \in \mathcal{S}(\mathbf{B}'\mathbf{B})$. First, as $\tilde{\mathbf{R}}^{-1}$ is block-diagonal, the $(M+1-k, M+1-l)$th block of $\mathbf{B}'\tilde{\mathbf{R}}^{-1}\mathbf{B}$ is given by $(\mathbf{B}^k)'\tilde{\mathbf{R}}^{-1}\mathbf{B}^l$.

   Now, for each $0 \leq k \leq M$, $\mathbf{B}^k$ is a block-diagonal matrix with blocks of size $|\mathcal{I}_{j_1,\dots,j_k}| \times r_k$, but $\tilde{\mathbf{R}}^{-1}$ has blocks of size $|\mathcal{I}_{j_1,\dots,j_M}| \times |\mathcal{I}_{j_1,\dots,j_M}|$. However, recalling (7), blocks of $\tilde{\mathbf{R}}^{-1}$ can also be viewed as having dimensions $|\mathcal{I}_{j_1,\dots,j_k}| \times r_k$. Because this implies that $(\mathbf{B}^k)'\tilde{\mathbf{R}}^{-1} \in \mathcal{S}(\mathbf{B}^k)'$, we have $(\mathbf{B}^k)'\tilde{\mathbf{R}}^{-1}\mathbf{B}^l \in \mathcal{S}((\mathbf{B}^k)'\mathbf{B}^l)$ and hence $(\mathbf{B})'\tilde{\mathbf{R}}^{-1}\mathbf{B} \in \mathcal{S}(\mathbf{B}'\mathbf{B})$. Finally, we conclude that $\mathbf{\Lambda} \in \mathcal{S}(\mathbf{B}'\mathbf{B})$, because $\mathbf{\Lambda} = \mathbf{I}_{n_\mathcal{G}} + \mathbf{B}'\tilde{\mathbf{R}}^{-1}\mathbf{B}$ and all diagonal elements of $\mathbf{B}'\mathbf{B}$ are nonzero.
2. According to Khare and Rajaratnam (2012, Thm. 1), for any positive definite matrix $\mathbf{S}$, the sparsity pattern in the Cholesky factor and its inverse are the same as that of the lower triangle of $\mathbf{S}$, if (a) the pattern of zeros in $\mathbf{S}$ corresponds to a homogeneous graph, and (b) the order of the vertices of the graph implied by the order of the rows is a Hasse-tree-based elimination scheme. Lemmas S1 and S2 in Section S8 show that these two conditions are met for $\mathbf{B}'\mathbf{B}$. These lemmas, together with Part 1 above, imply that $\mathbf{L} \in \mathcal{S}(\mathbf{\Lambda}^L)$ and $\mathbf{L}^{-1} \in \mathcal{S}(\mathbf{\Lambda}^L)$.
3. Observe that $\mathbf{\Lambda}^{-1} = (\mathbf{L}\mathbf{L}')^{-1} = (\mathbf{L}^{-1})'\mathbf{L}^{-1}$. Thus, $(\mathbf{L}^{-1})'$ is the Cholesky factor of $\mathbf{\Lambda}^{-1}$. Moreover, by Part 2, $(\mathbf{L}^{-1})' \in \mathcal{S}((\mathbf{B}'\mathbf{B})^U)$. This allows us to define blocks $\tilde{\mathbf{L}}^{m,k}$ such that

$$(\mathbf{L}^{-1})' = \begin{bmatrix} \tilde{\mathbf{L}}^{M,M} & \dots & \tilde{\mathbf{L}}^{M,1} & \tilde{\mathbf{L}}^{M,0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \dots & \tilde{\mathbf{L}}^{1,1} & \tilde{\mathbf{L}}^{1,0} \\ \mathbf{0} & \dots & \mathbf{0} & \tilde{\mathbf{L}}^{0,0} \end{bmatrix} = \left[\tilde{\mathbf{L}}^{m,k}\right]_{m,k=M,\dots,0},$$

where each $\tilde{\mathbf{L}}^{m,k} \in \mathcal{S}((\mathbf{B}^m)'\mathbf{B}^k)$ for $m \geq k$ and is zero when $m < k$. This means that for each m, k with $m \geq k$, we can consider the sparsity of $\mathbf{B}^m(\mathbf{B}^m)'\mathbf{B}^k$ instead of $\mathbf{B}^m\tilde{\mathbf{L}}^{m,k}$.

Recall that $\mathbf{B}^k$ is block-diagonal with blocks of size $|\mathcal{I}_{j_1,\ldots,j_k}| \times r_k$. Similarly, $\mathbf{B}^m$ has blocks that are $|\mathcal{I}_{j_1,\ldots,j_m}| \times r_m$. However, since $k \leq m$, using (7) we can also see $\mathbf{B}^m$ as a block-diagonal matrix whose blocks have dimensions $|\mathcal{I}_{j_1,\ldots,j_k}| \times r_k$ (cf. proof of Proposition 2). This implies that $\mathbf{B}^m(\mathbf{B}^m)' \in \mathcal{S}(\mathbf{B}^k(\mathbf{B}^k)')$, which means that $\mathbf{B}^m(\mathbf{B}^m)'\mathbf{B}^k \in \mathcal{S}(\mathbf{B}^k)$ and hence $\mathbf{B}^m\tilde{\mathbf{L}}^{m,k} \in \mathcal{S}(\mathbf{B}^k)$.

Finally, we observe that

$$\begin{aligned}\mathbf{B} \cdot (\mathbf{L}^{-1})' &= \left[\mathbf{B}^M \ \mathbf{B}^{M-1} \ \ldots \ \mathbf{B}^0\right] \cdot \begin{bmatrix} \tilde{\mathbf{L}}^{M,M} & \ldots & \tilde{\mathbf{L}}^{M,1} & \tilde{\mathbf{L}}^{M,0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \ldots & \tilde{\mathbf{L}}^{1,1} & \tilde{\mathbf{L}}^{1,0} \\ \mathbf{0} & \ldots & \mathbf{0} & \tilde{\mathbf{L}}^{0,0} \end{bmatrix} \\ &= \left[\tilde{\mathbf{B}}^M \ \tilde{\mathbf{B}}^{M-1} \ \ldots \ \tilde{\mathbf{B}}^0\right],\end{aligned}$$

where $\tilde{\mathbf{B}}^k = \sum_{m=k}^M \mathbf{B}^m\tilde{\mathbf{L}}^{m,k}$. Since we showed that $\mathbf{B}^m\tilde{\mathbf{L}}^{m,k} \in \mathcal{S}(\mathbf{B}^k)$, this means that $\tilde{\mathbf{B}}^k \in \mathcal{S}(\mathbf{B}^k)$.

□

Proof of Proposition 4. By Proposition 3, Parts 1 and 2, $\mathbf{L} \in \mathcal{S}((\mathbf{B}'\mathbf{B})^L)$. Therefore, it suffices to show that $\mathbf{c}_j = (\mathbf{B}'\mathbf{B})^L[:, j]$, the jth column of $(\mathbf{B}'\mathbf{B})^L$, has $\mathcal{O}(N)$ nonzero elements for each j. Notice that $\mathbf{c}_j = (0, \ldots, 0, \mathbf{c}_j^{k,k}, \ldots \mathbf{c}_j^{M,k})'$ where $\mathbf{c}_j^{k,l} = ((\mathbf{B}^k)'\mathbf{B}^l)[:, j]$, the jth column of $(\mathbf{B}^k)'\mathbf{B}^l$. Because $l \geq k$, each of the $\mathbf{B}^k(\mathbf{B}^l)'$ matrices is block-diagonal with blocks of height $|\mathcal{K}_{j_1,\ldots,j_k}|$. The vector $\mathbf{c}_j^{k,l}$ intersects exactly one of such diagonal blocks, and so the total number of nonzero elements in $\mathbf{c}$ is at most $N = \sum_m r_m$. □

Proof of Proposition 5. Observe that it is enough to consider only the complexity of operations in (4) because $\mathbf{V}^l_{j_1,\ldots,j_m}$ can be obtained by selecting appropriate rows from $\mathbf{W}^l_{j_1,\ldots,j_m}$. Given matrix $\boldsymbol{\Sigma}$, we only need to calculate the second term in (4). First, note that calculating $\mathbf{W}^l_{j_1,\ldots,j_m}$ for all $(j_1, \ldots, j_m)$ is the same as computing $\mathbf{W}^l_{j_1,\ldots,j_\ell}$ for all l, and then, for each $(j_1, \ldots, j_m)$, selecting the rows corresponding to $\mathcal{I}_{j_1,\ldots,j_m}$. Thus, we show the complexity of calculating all $\mathbf{W}^l_{j_1,\ldots,j_\ell}$.

Assume that all $\mathbf{W}^k_{j_1,\ldots,j_\ell}$ for $k < l$ are already given and consider the summation term. Each of its components takes $\mathcal{O}(|\mathcal{I}_{j_1,\ldots,j_\ell}|r^2 + r^3 + r^3 + r^3) = \mathcal{O}(|\mathcal{I}_{j_1,\ldots,j_\ell}|r^2)$ to compute. Because for any given l, there are at most M terms under the summation, their joint computation time is $\mathcal{O}(M \cdot |\mathcal{I}_{j_1,\ldots,j_\ell}|r^2)$. For a given l, these calculations have to be performed for each set of indices $\mathcal{I}_{j_1,\ldots,j_\ell}$. Thus, obtaining all $\mathbf{W}^l_{j_1,\ldots,j_\ell}$ requires $\mathcal{O}(M \cdot \sum_{j_1,\ldots,j_\ell} |\mathcal{I}_{j_1,\ldots,j_\ell}|r^2) = \mathcal{O}(M \cdot nr^2)$ time. Now notice that $\mathcal{I}_{j_1,\ldots,j_m} \subset \mathcal{I}_{j_1,\ldots,j_\ell}$. Therefore, once we have $\mathbf{W}^l_{j_1,\ldots,j_\ell}$, we obtain $\mathbf{W}^l_{j_1,\ldots,j_m}$ by selecting appropriate rows from $\mathbf{W}^l_{j_1,\ldots,j_\ell}$. Finally, iterating over $l = 0, \ldots, M$ means that the total cost of Algorithm 2 is $\mathcal{O}(M^2nr^2) = \mathcal{O}(nN^2)$. □

Proof of Proposition 6. The forecast step requires calculating $\boldsymbol{\mu}_{t|t-1} = \mathbf{A}_t\boldsymbol{\mu}_{t-1|t-1}$ and $\mathbf{B}^F_{t|t-1} = \mathbf{A}_t\mathbf{B}_{t-1|t-1}$, which can be obtained in $\mathcal{O}(nr)$ and $\mathcal{O}(nrN)$ time, respectively, due to the sparsity structures of $\mathbf{B}_{t-1|t-1}$ (see Proposition 1) and $\mathbf{A}_t$ (Assumption 2).

By Proposition 5, the MRD of a given covariance matrix $\boldsymbol{\Sigma}$ requires $\mathcal{O}(nN^2)$ operations. Here, $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_{t|t-1}$ is not given, but each (i, j) element must be computed as

$$\boldsymbol{\Sigma}_{t|t-1}[i, j] = (\mathbf{B}^F_{t|t-1}[i, :])(\mathbf{B}^F_{t|t-1}[j, :])' + \mathbf{Q}_t[i, j].$$

This does not increase the complexity of the MRD, because the MRD requires only $\mathcal{O}(nN)$ elements of $\boldsymbol{\Sigma}_{t|t-1}$, each of which can be computed in $\mathcal{O}(N)$ time due to the sparsity structure of $\mathbf{B}^F_{t|t-1}$. Thus, the entire forecast step can be performed in $\mathcal{O}(nN^2)$ time.

In the update step, we must compute $\widetilde{\boldsymbol{\Lambda}}$, $\mathbf{L}^{-1} = \widetilde{\boldsymbol{\Lambda}}^{-1/2}$, and $\mathbf{B}_{t|t} = \mathbf{B}_{t|t-1}(\mathbf{L}^{-1})'$. Under Assumption 1, $\mathbf{H}$ and $\mathbf{R}$ are block-diagonal matrices with at most J^M blocks of size $\mathcal{O}(r \times r)$ each. Thus, calculating $\tilde{\mathbf{R}} := \mathbf{H}'\mathbf{R}^{-1}\mathbf{H}$ requires $\mathcal{O}(J^Mr^3) = \mathcal{O}(nr^2)$ operations. The resulting matrix is block-diagonal with blocks of size $\mathcal{O}(r \times r)$, conformable with the blocks of $\mathbf{B}_{t|t-1}$. Given $\tilde{\mathbf{R}}$, the cost of calculating $\widetilde{\boldsymbol{\Lambda}}$ is dominated by multiplying $\mathbf{B}_{t|t-1}$ by $\tilde{\mathbf{R}}$. By Proposition 1, each row of $\mathbf{B}_{t|t-1}$ has N nonzero elements, so in view of the structure of $\tilde{\mathbf{R}}$ determined above, it takes $\mathcal{O}(nN^2)$ operations to obtain the product $\mathbf{B}_{t|t-1}\tilde{\mathbf{R}}$ and, consequently, to calculate $\widetilde{\boldsymbol{\Lambda}}$.

The complexity of computing a Cholesky factor is on the order of the sum of the squared number of nonzero elements per column (e.g., Toledo 2007, Thm. 2.2). Thus, computing $\mathbf{L}$ requires $\mathcal{O}(nN^2)$ time, because $\mathbf{L}$ has $\mathcal{O}(N)$ elements in each of its n columns (Proposition 3). Computing $\mathbf{L}^{-1}$ can be accomplished by solving a triangular system of equations for each column of $\mathbf{L}^{-1}$. Using Proposition 4, we conclude that each of these systems will have only $\mathcal{O}(N)$ equations and thus can be solved in $\mathcal{O}(N^2)$ time (Kincaid and Cheney 2002, chap. 4.2). As we need to compute n columns, the total effort required for obtaining $\mathbf{L}^{-1}$ is $\mathcal{O}(nN^2)$.

Finally, recall that both $\mathbf{B}_{t|t-1}$ and $\mathbf{L}^{-1}$ have $\mathcal{O}(N)$ elements in each row and that, by Proposition 3, their product, $\mathbf{B}_{t|t}$, has only $\mathcal{O}(nN)$ nonzero elements. Because each of these elements can be computed in $\mathcal{O}(N)$ time, the total computation cost of this step is $\mathcal{O}(nN^2)$.

To summarize, all three matrices necessary in the update step can be obtained in $\mathcal{O}(nN^2)$ time. Thus, we showed that both steps of Algorithm 1 require $\mathcal{O}(nN^2)$ time, which completes the proof. □

Proof of Proposition 7. For $m = 1, \ldots, M$, define $\mathbf{B}^{0:m} = (\mathbf{B}^m, \ldots, \mathbf{B}^0)$ as the submatrix of $\mathbf{B}$ consisting of the column blocks corresponding to resolutions $0, \ldots, m$. To show that $\mathbf{BB}' \in H^r_M$, we prove by induction over $m = 1, \ldots, M$ that $(\mathbf{B}^{0:m}\mathbf{B}^{0:m})' \in H^r_m$. For $m = 1$, we have $\mathbf{B}^{0:1} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \mathbf{B}_{01} \\ \mathbf{0} & \mathbf{B}_2 & \mathbf{B}_{02} \end{bmatrix}$, where $\mathbf{B}_{01}$ and $\mathbf{B}_{02}$ are each r columns wide. Thus,

$$\mathbf{B}^{0:1}(\mathbf{B}^{0:1})' = \begin{bmatrix} \mathbf{B}_{01}\mathbf{B}_{01}' + \mathbf{B}_1\mathbf{B}_1' & \mathbf{B}_{01}\mathbf{B}_{02}' \\ \mathbf{B}_{02}\mathbf{B}_{01}' & \mathbf{B}_{02}\mathbf{B}_{02}' + \mathbf{B}_2\mathbf{B}_2' \end{bmatrix}$$

and so $\mathbf{B}^{0:1}(\mathbf{B}^{0:1})' \in H^r_1$.

Now, assume that $\mathbf{B}^{0:m-1}(\mathbf{B}^{0:m-1})' \in H^r_{m-1}$. We have

$$\begin{aligned}\mathbf{B}^{0:m}(\mathbf{B}^{0:m})' &= \sum_{j=0}^{m} \mathbf{B}^j(\mathbf{B}^j)' = \sum_{j=0}^{m-1} \mathbf{B}^j(\mathbf{B}^j)' + \mathbf{B}^m(\mathbf{B}^m)' \\ &= \mathbf{B}^{0:m-1}(\mathbf{B}^{0:m-1})' + \mathbf{B}^m(\mathbf{B}^m)'.\end{aligned}$$

Next observe that for any k, the matrix $\mathbf{B}^k$ is block-diagonal, which means that $\mathbf{B}^k(\mathbf{B}^k)'$ is also block-diagonal with dense blocks $\mathbf{B}^k(\mathbf{B}^k)'[\mathcal{I}_{j_1,\ldots,j_k}, \mathcal{I}_{j_1,\ldots,j_k}]$. However, recursive partitioning of the domain means that $\mathcal{I}_{j_1,\ldots,j_{k-1}} \supset \mathcal{I}_{j_1,\ldots,j_k}$. Therefore, if $k > j$, then blocks of $\mathbf{B}^k(\mathbf{B}^k)'$ are nested within the blocks of $\mathbf{B}^j(\mathbf{B}^j)'$. Since this holds also for $k = m - 1$ and $j = m$, it means that $\mathbf{B}^{1:m}(\mathbf{B}^{1:m})' \in H^r_m$. □

Supplementary Materials

Extensions and further details referenced in the article using the "S" prefix. (supplement.pdf)

Python code which can be used to reproduce numerical simulations. (MRFcode.zip)

Acknowledgments

We would like to thank Jacob Bien, Wenlong Gong, Joseph Guinness, Dorit Hammerling, Ephrahim Hanks, Peter Kuchement, Mohsen Pourahmadi, Ramalingam Saravanan, Michael Stein, Jonathan Stroud, Istvan Szunyogh, Christopher Wikle, Catherine Yan, and several reviewers for helpful comments and discussions. Special thanks to Jonathan Stroud for providing code and data for the Lake Michigan example.

Funding

The authors were partially supported by National Science Foundation (NSF) grants DMS-1521676 and DMS-1654083.

ORCID

Marcin Jurek http://orcid.org/0000-0002-2331-8236

References

Ambikasaran, S., and Darve, E. (2013), "An $O(n \log n)$ Fast Direct Solver for Partial Hierarchically Semi-Separable Matrices," *Journal of Scientific Computing*, 57, 477–501. [7]

Ambikasaran, S., Foreman-Mackey, D., Greengard, L., Hogg, D. W., and O'Neil, M. (2016), "Fast Direct Methods for Gaussian Processes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38, 252–265. [2,7]

Anderson, J. L., and Anderson, S. L. (1999), "A Monte Carlo Implementation of the Nonlinear Filtering Problem to Produce Ensemble Assimilations and Forecasts," *Monthly Weather Review*, 127, 2741–2758. [5]

Beezley, J. D., Mandel, J., and Cobb, L. (2011), "Wavelet Ensemble Kalman Filters," in *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems* (Vol. 2), pp. 514–517. [2]

Bradley, J. R., Holan, S. H., and Wikle, C. K. (2018), "Computationally Efficient Multivariate Spatio-Temporal Models for High-Dimensional Count-Valued Data" (with discussion), *Bayesian Analysis*, 13, 253–310. [1]

Chui, C. (1992), *An Introduction to Wavelets*, San Diego, CA: Academic Press. [2]

Cressie, N., Shi, T., and Kang, E. L. (2010), "Fixed Rank Filtering for Spatio-Temporal Data," *Journal of Computational and Graphical Statistics*, 19, 724–745. [1,9]

Cristi, R., and Tummala, M. (2000), "Multirate, Multiresolution, Recursive Kalman Filter," *Signal Processing*, 80:1945–1958. [2]

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E. (2016a), "Hierarchical Nearest-Neighbor Gaussian Process Models for Large Geostatistical Datasets," *Journal of the American Statistical Association*, 111, 800–812. [1]

Datta, A., Banerjee, S., Finley, A. O., Hamm, N. A. S., and Schaap, M. (2016b), "Non-Separable Dynamic Nearest-Neighbor Gaussian Process Models for Large Spatio-Temporal Data With an Application to Particulate Matter Analysis," *The Annals of Applied Statistics*, 10, 1286–1316. [2]

DelSole, T., and Yang, X. (2010), "State and Parameter Estimation in Stochastic Dynamical Models," *Physica D*, 239, 1781–1788. [8]

Douc, R., Cappé, O., and Moulines, E. (2005), "Comparison of Resampling Schemes for Particle Filtering," in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005*, pp. 64–69. [9]

Doucet, A., de Freitas, N., Murphy, K., and Russell, S. (2000), "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," in *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 176–183. [8]

——— (2013), "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," arXiv no. 1301.3853. [9]

Evensen, G. (1994), "Sequential Data Assimilation With a Nonlinear Quasi-Geostrophic Model Using Monte Carlo Methods to Forecast Error Statistics," *Journal of Geophysical Research*, 99, 10143–10162. [1]

——— (2007), *Data Assimilation: The Ensemble Kalman Filter*, Berlin, Heidelberg: Springer. [1]

Finley, A. O., Sang, H., Banerjee, S., and Gelfand, A. E. (2009), "Improving the Performance of Predictive Process Modeling for Large Datasets," *Computational Statistics & Data Analysis*, 53, 2873–2884. [9]

Gordon, N., Salmond, D., and Smith, A. (1993), "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings F (Radar and Signal Processing)*, 140, 107–113. [1]

Guinness, J. (2016), "Permutation Methods for Sharpening Gaussian Process Approximations," arXiv no. 1609.05372. [5]

Hackbusch, W. (2015), *Hierarchical Matrices: Algorithms and Analysis* (Vol. 49), Berlin, Heidelberg: Springer. [2]

Harville, D. A. (1997), *Matrix Algebra From a Statistician's Perspective*, New York, NY: Springer. [8]

Heaton, M. J., Datta, A., Finley, A., Furrer, R., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2017), "Methods for Analyzing Large Spatial Data: A Review and Comparison," arXiv no. 1710.05013. [2,5]

Henderson, H., and Searle, S. (1981), "On Deriving the Inverse of a Sum of Matrices," *SIAM Review*, 23, 53–60. [4]

Hickmann, K. S., and Godinez, H. C. (2015), "A Multiresolution Ensemble Kalman Filter Using Wavelet Decomposition," arXiv no. 1511.01935. [2]

Hogben, L. (2006), *Handbook of Linear Algebra*, Discrete Mathematics and Its Applications, Boca Raton, FL: CRC Press. [7]

Houtekamer, P. L., and Zhang, F. (2016), "Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation," *Monthly Weather Review*, 144, 4489–4532. [1]

Kalman, R. (1960), "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, 82, 35–45. [1,3]

Kanter, M. (1997), "Unimodal Spectral Window," *Statistics & Probability Letters*, 34, 403–411. [9]

Katzfuss, M. (2017), "A Multi-Resolution Approximation for Massive Spatial Datasets," *Journal of the American Statistical Association*, 112, 201–214. [2,4,5,7,8,9,10]

Katzfuss, M., and Cressie, N. (2011), "Spatio-Temporal Smoothing and EM Estimation for Massive Remote-Sensing Data Sets," *Journal of Time Series Analysis*, 32, 430–446. [1]

Katzfuss, M., and Gong, W. (2020), "A Class of Multi-Resolution Approximations for Large Spatial Datasets," *Statistica Sinica*, 30, 2203–2226. [2,5]

Katzfuss, M., and Guinness, J. (2021), "A General Framework for Vecchia Approximations of Gaussian Processes," *Statistical Science*, 36, 124–141. [1]

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2016), "Understanding the Ensemble Kalman Filter," *The American Statistician*, 70, 350–357. [1,9]

——— (2020), "Ensemble Kalman Methods for High-Dimensional Hierarchical Dynamic Space-Time Models," *Journal of the American Statistical Association*, 115, 866–885. [8]

Khare, K., and Rajaratnam, B. (2012), "Sparse Matrix Decompositions and Graph Characterizations," *Linear Algebra and Its Applications*, 437, 932–947. [13]

Kincaid, D., and Cheney, E. (2002), *Numerical Analysis: Mathematics of Scientific Computing*, Pure and Applied Undergraduate Texts, Providence, RI: American Mathematical Society. [14]

Li, J. Y., Ambikasaran, S., Darve, E. F., and Kitanidis, P. K. (2014), "A Kalman Filter Powered by H-Matrices for Quasi-Continuous Data Assimilation Problems," *Water Resources Research*, 50, 3734–3749. [2,7]

Lindgren, F., Rue, H., and Lindström, J. (2011), "An Explicit Link Between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach," *Journal of the Royal Statistical Society*, Series B, 73, 423–498. [1]

Martín-Fernández, L., Gilioli, G., Lanzarone, E., Míguez, J., Pasquali, S., Ruggeri, F., and Ruiz, D. P. (2014), "A Rao-Blackwellized Particle Filter for Joint Parameter Estimation and Biomass Tracking in a Stochastic Predator-Prey System," *Mathematical Biosciences & Engineering*, 11, 573. [9]

Martín-Fernández, L., and Lanzarone, E. (2015), "A Particle-Filtering Approach for Real-Time Estimation of Thermal Conductivity and Temperature Tracking in Homogeneous Masses," *Numerical Heat Transfer, Part B: Fundamentals*, 67, 507–530. [9]

Nychka, D. W., and Anderson, J. L. (2010), "Data Assimilation" (Chapter 27) in *Handbook of Spatial Statistics*, eds. A. E. Gelfand, P. J. Diggle, M. Fuentes, and P. Guttorp, Boca Raton, FL: CRC Press, pp. 477–494. [1]

Nychka, D. W., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. R. (2015), "A Multi-Resolution Gaussian Process Model for the Analysis of Large Spatial Data Sets," *Journal of Computational and Graphical Statistics*, 24, 579–599. [1]

Pham, D. T., Verron, J., and Christine Roubaud, M. (1998), "A Singular Evolutive Extended Kalman Filter for Data Assimilation in Oceanography," *Journal of Marine Systems*, 16, 323–340. [1,5]

Rauch, H. E., Striebel, C., and Tung, F. (1965), "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA Journal*, 3, 1445–1450. [7]

Renaud, O., Starck, J. L., and Murtagh, F. (2005), "Wavelet-Based Combined Signal Filtering and Prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* , 35, 1241–1251. [2]

Saibaba, A. K., Miller, E. L., and Kitanidis, P. K. (2015), "Fast Kalman Filter Using Hierarchical Matrices and a Low-Rank Perturbative Approach," *Inverse Problems*, 31, 015009. [2,7]

Searle, S. (1982), *Matrix Algebra Useful for Statistics*, New York: Wiley. [4]

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. L. (2008), "Obstacles to High-Dimensional Particle Filtering," *Monthly Weather Review*, 136, 4629–4640. [1]

Stein, M. L. (2002), "The Screening Effect in Kriging," *The Annals of Statistics*, 30, 298–323. [5]

——— (2011), "2010 Rietz Lecture: When Does the Screening Effect Hold?," *The Annals of Statistics*, 39, 2795–2819. [5]

——— (2014), "Limitations on Low Rank Approximations for Covariance Matrices of Spatial Data," *Spatial Statistics*, 8, 1–19. [1]

Stroud, J. R., Stein, M. L., Lesht, B. M., Schwab, D. J., and Beletsky, D. (2010), "An Ensemble Kalman Filter and Smoother for Satellite Data Assimilation," *Journal of the American Statistical Association*, 105, 978–990. [10,11,12]

Talagrand, O., and Courtier, P. (1987), "Variational Assimilation of Meteorological Observations With the Adjoint Vorticity Equation. I: Theory," *Quarterly Journal of the Royal Meteorological Society*, 113, 1311–1328. [1]

Tippett, M. K., Anderson, J. L., Bishop, C. H., Hamill, T. M., and Whitaker, J. S. (2003), "Ensemble Square-Root Filters," *Monthly Weather Review*, 131, 1485–1490. [9]

Toledo, S. (2007), "Lecture Notes on Combinatorial Preconditioners, Chapter 3," available at *http://www.tau.ac.il/~{}stoledo/Support/chapter-direct.pdf* . [14]

Verlaan, M., and Heemink, A. (1995), "Reduced Rank Square Root Filters for Large Scale Data Assimilation Problems," in *Proceedings of the 2nd International Symposium on Assimilation in Meteorology and Oceanography*, World Meteorological Organization, pp. 247–252. [1]

Wikle, C. K., and Cressie, N. (1999), "A Dimension-Reduced Approach to Space-Time Kalman Filtering," *Biometrika*, 86, 815–829. [1]

Wikle, C. K., Milliff, R. F., Nychka, D., and Berliner, L. M. (2001), "Spatiotemporal Hierarchical Bayesian Modeling Tropical Ocean Surface Winds," *Journal of the American Statistical Association*, 96, 382–397. [1]

Zhang, B., Sang, H., and Huang, J. Z. (2015), "Full-Scale Approximations of Spatio-Temporal Covariance Models for Large Datasets," *Statistica Sinica*, 25, 99–114. [2]