Hierarchical sparse Cholesky decomposition with applications to high-dimensional spatio-temporal filtering

Marcin Jurek* Matthias Katzfuss*†

Abstract

Spatial statistics often involves Cholesky decomposition of covariance matrices. To ensure scalability to high dimensions, several recent approximations have assumed a sparse Cholesky factor of the precision matrix. We propose a hierarchical Vecchia approximation, whose conditional-independence assumptions imply sparsity in the Cholesky factors of both the precision and the covariance matrix. This remarkable property is crucial for applications to high-dimensional spatio-temporal filtering. We present a fast and simple algorithm to compute our hierarchical Vecchia approximation, and we provide extensions to non-linear data assimilation with non-Gaussian data based on the Laplace approximation. In several numerical comparisons, our methods strongly outperformed alternative approaches.

Keywords: state-space model, spatio-temporal statistics, data assimilation, Vecchia approximation, hierarchical matrix, incomplete Cholesky decomposition

1 Introduction

Symmetric positive-definite matrices arise in spatial statistics, Gaussian-process inference, and spatio-temporal filtering, with a wealth of application areas, including geoscience (e.g., Cressie, 1993; Banerjee et al., 2004), machine learning (e.g., Rasmussen and Williams, 2006), data assimilation (e.g., Nychka and Anderson, 2010; Katzfuss et al., 2016), and the analysis of computer experiments (e.g., Sacks et al., 1989; Kennedy and O'Hagan, 2001). Inference in these areas typically relies on Cholesky decomposition of the positive-definite matrices. However, this operation scales cubically in the dimension of the matrix, and it is thus computationally infeasible for many modern problems and applications, which are increasingly high-dimensional.

Countless approaches have been proposed to address these computational challenges. Heaton et al. (2019) provide a recent review from a spatial-statistics perspective, and Liu et al. (2020) review approaches in machine learning. In high-dimensional filtering, proposed

*Department of Statistics, Texas A&M University
†Corresponding author: katzfuss@gmail.com

solutions include low-dimensional approximations (e.g., Verlaan and Heemink, 1995; Pham et al., 1998; Wikle and Cressie, 1999; Katzfuss and Cressie, 2011), spectral methods (e.g. Wikle and Cressie, 1999; Sigrist et al., 2015), and hierarchical approaches (e.g., Johannesson et al., 2003; Li et al., 2014; Saibaba et al., 2015; Jurek and Katzfuss, 2018). Operational data assimilation often relies on ensemble Kalman filters (e.g., Evensen, 1994; Burgers et al., 1998; Anderson, 2001; Evensen, 2007; Katzfuss et al., 2016, 2020c), which represent distributions by samples or ensembles.

Maybe the most promising approximations for spatial data and Gaussian processes implicitly or explicitly rely on sparse Cholesky factors. The assumption of ordered conditional independence in the popular Vecchia approximation (Vecchia, 1988) and its extensions (e.g., Stein et al., 2004; Datta et al., 2016; Guinness, 2018; Katzfuss and Guinness, 2019; Katzfuss et al., 2020a,b; Schäfer et al., 2020) implies sparsity in the Cholesky factor of the precision matrix. Schäfer et al. (2017) uses an incomplete Cholesky decomposition to construct a sparse approximate Cholesky factor of the covariance matrix. However, these methods are not generally applicable to spatio-temporal filtering, because the assumed sparsity is not preserved under filtering operations.

Here, we relate the sparsity of the Cholesky factors of the covariance matrix and the precision matrix to specific assumptions regarding ordered conditional independence. We show that these assumptions are simultaneously satisfied for a particular Gaussian-process approximation that we call hierarchical Vecchia (HV), which is a special case of the general Vecchia approximation (Katzfuss and Guinness, 2019) based on hierarchical domain partitioning (e.g., Katzfuss, 2017; Katzfuss and Gong, 2019). We show that the HV approximation can be computed using a simple and fast incomplete Cholesky decomposition.

Due to its remarkable property of implying a sparse Cholesky factor whose inverse has equivalent sparsity structure, HV is well suited for extensions to spatio-temporal filtering; this is in contrast to other Vecchia approximations and other spatial approximations relying on sparsity. We provide a scalable HV-based filter for linear Gaussian spatio-temporal state-space models, which is related to the multi-resolution filter of Jurek and Katzfuss (2018). Further, by combining HV with a Laplace approximation (cf. Zilber and Katzfuss, 2019), our method can be used for the analysis of non-Gaussian data. Finally, by combining the methods with the extended Kalman filter (e.g., Grewal and Andrews, 1993, Ch. 5), we obtain fast filters for high-dimensional, non-linear, and non-Gaussian spatio-temporal models. For a given formulation of HV, the computational cost of all of our algorithms scales linearly in the state dimension, assuming sufficiently sparse temporal evolution.

The remainder of this document is organized as follows. In Section 2, we specify the relationship between ordered conditional independence and sparse (inverse) Cholesky factors. Then, we build up increasingly complex and general methods, culminating in non-linear and non-Gaussian spatio-temporal filters: in Section 3, we introduce hierarchical Vecchia for a linear Gaussian spatial field at a single time point; in Section 4, we extend this to non-Gaussian data; and in Section 5, we consider the general spatio-temporal filtering case, including non-linear evolution and parameter inference on unknown parameters in the model. Section 6 contains numerical comparisons to existing approaches. Section 7 concludes. Appendices A–B contain proofs and further details. Code implementing our methods and numerical comparisons is available at `https://github.com/katzfuss-group/vecchiaFilter`.

2 Sparsity of Cholesky factors

We begin by specifying the connections between ordered conditional independence and sparsity of the Cholesky factor of the covariance and precision matrix.

CLAIM 1. *Let* $\mathbf{w}$ *be a normal random vector with variance-covariance matrix* $\mathbf{K}$.

1. *Let* $\mathbf{L} = \text{chol}(\mathbf{K})$ *be the lower-triangular Cholesky factor of the covariance matrix* $\mathbf{K}$. *For* $i > j$:
$$\mathbf{L}_{i,j} = 0 \iff w_i \perp w_j \,|\, \mathbf{w}_{1:j-1}$$
2. *Let* $\mathbf{U} = \text{rchol}(\mathbf{K}^{-1}) = \mathbf{P}\,\text{chol}(\mathbf{P}\mathbf{K}^{-1}\mathbf{P})\,\mathbf{P}$ *be the Cholesky factor of the precision matrix under reverse ordering, where* $\mathbf{P}$ *is the reverse-ordering permutation matrix. Then* $\mathbf{U}$ *is upper-triangular, and for* $i > j$:
$$\mathbf{U}_{j,i} = 0 \iff w_i \perp w_j \,|\, \mathbf{w}_{1:j-1}, \mathbf{w}_{j+1:i-1}$$

The connection between ordered conditional independence and the Cholesky factor of the precision matrix is well known (e.g., Rue and Held, 2010); Part 2 of our claim states this connection under reverse ordering (e.g., Katzfuss and Guinness, 2019, Prop. 3.3). In Part 1, we consider the lesser-known relationship between ordered conditional independence and sparsity of the Cholesky factor of the covariance matrix, which was recently discussed in Schäfer et al. (2017, Sect. 1.4.2). For completeness, we provide a proof of Claim 1 in Appendix B.

Claim 1 is crucial for our later developments and proofs. In Section 3, we specify a hierarchical Vecchia approximation of Gaussian processes that satisfies both types of conditional independence in Claim 1; the resulting sparsity of the Cholesky factor and its inverse allows extensions to spatio-temporal filtering in Section 5.

3 Hierarchical Vecchia for large Gaussian spatial data

Consider a Gaussian process $x(\cdot)$ and a vector $\mathbf{x} = (x_1, \ldots, x_n)^\top$ representing $x(\cdot)$ evaluated on a grid $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ with $\mathbf{s}_i \in \mathcal{D} \subset \mathbb{R}^d$ and $x_i = x(\mathbf{s}_i)$ for $\mathbf{s}_i \in \mathcal{D}$, $i = 1, \ldots, n$. We assume the following model:

$$y_i \,|\, \mathbf{x} \overset{ind}{\sim} \mathcal{N}(x_i, \tau_i^2), \qquad i \in \mathcal{I}, \tag{1}$$
$$\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2}$$

where $\mathcal{I} \subset \{1, \ldots, n\}$ are the indices of grid points at which observations are available, and $\mathbf{y}$ is the data vector consisting of these observations $\{y_i : i \in \mathcal{I}\}$. Note that we can equivalently express (1) using matrix notation as $\mathbf{y} \,|\, \mathbf{x} \sim \mathcal{N}(\mathbf{H}\mathbf{x}, \mathbf{R})$, where $\mathbf{H}$ is obtained by selecting only the rows with indices $i \in \mathcal{I}$ from an identity matrix, and $\mathbf{R}$ is a diagonal matrix with entries $\{\tau_i^2 : i \in \mathcal{I}\}$.

Our interest is in computing the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$, which requires inverting or decomposing an $n \times n$ matrix at a cost of $\mathcal{O}(n^3)$ if $|\mathcal{I}| = \mathcal{O}(n)$. This is computationally infeasible for large n.

3.1 The hierarchical Vecchia approximation

We now describe a hierarchical Vecchia approximation with unique sparsity and computational properties, which enable fast computation for spatial models as in (1)–(2) and also allow extensions to spatio-temporal filtering as explained later.

Assume that the elements of the vector $\mathbf{x}$ are hierarchically partitioned into a set $\mathcal{X}^{0:M} = \bigcup_{m=0}^{M} \mathcal{X}^m$, where $\mathcal{X}^m = \bigcup_{k=1}^{m} \bigcup_{j_k=1}^{J_k} \mathcal{X}_{j_1,\ldots,j_m}$, and $\mathcal{X}_{j_1,\ldots,j_m}$ is a set consisting of $|\mathcal{X}_{j_1,\ldots,j_m}|$ elements of $\mathbf{x}$, such that there is no overlap between any two sets, $\mathcal{X}_{j_1,\ldots,j_m} \cap \mathcal{X}_{i_1,\ldots,i_l} = \emptyset$ for $(j_1,\ldots,j_m) \neq (i_1,\ldots,i_l)$. We assume that $\mathbf{x}$ is ordered according to $\mathcal{X}^{0:M}$, in the sense that if $i > j$, then $x_i \in \mathcal{X}^{m_1}$ and $x_j \in \mathcal{X}^{m_2}$ with $m_1 \geq m_2$. As a toy example with $n = 6$, the vector $\mathbf{x} = (x_1,\ldots,x_6)$ might be partitioned with $M = 1$, $J_1 = 2$ as $\mathcal{X}^{0:1} = \mathcal{X}^0 \cup \mathcal{X}^1$, $\mathcal{X}^0 = \mathcal{X} = \{x_1, x_2\}$, and $\mathcal{X}^1 = \mathcal{X}_{1,1} \cup \mathcal{X}_{1,2}$, where $\mathcal{X}_{1,1} = \{x_3, x_4\}$, and $\mathcal{X}_{1,2} = \{x_5, x_6\}$. Another toy example is illustrated in Figure 1.

The exact distribution of $\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ can be written as

$$p(\mathbf{x}) = \prod_{m=0}^{M} \prod_{j_1,\ldots,j_m} p(\mathcal{X}_{j_1,\ldots,j_m} | \mathcal{X}^{0:m-1}, \mathcal{X}_{j_1,\ldots,j_{m-1},1:j_m-1}),$$

where the conditioning set of $\mathcal{X}_{j_1,\ldots,j_m}$ consists of all sets $\mathcal{X}^{0:m-1}$ at lower resolution, plus those at the same resolution that are previous in lexicographic ordering. The idea of Vecchia (1988) was to remove many of these variables in the conditioning set, which for geostatistical applications often incurs only small approximation error due to the so-called screening effect (e.g., Stein, 2002, 2011).

Here we consider a hierarchical Vecchia (HV) approximation of the form

$$\hat{p}(\mathbf{x}) = \prod_{m=0}^{M} \prod_{j_1,\ldots,j_m} p(\mathcal{X}_{j_1,\ldots,j_m} | \mathcal{A}_{j_1,\ldots,j_m}), \tag{3}$$

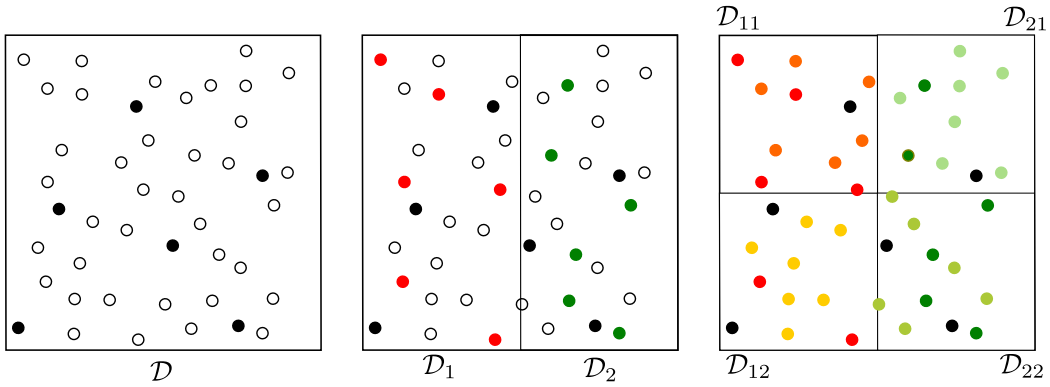
where $\mathcal{A}_{j_1,\ldots,j_m} = \mathcal{X} \cup \mathcal{X}_{j_1} \cup \ldots \cup \mathcal{X}_{j_1,\ldots,j_{m-1}}$ is the set of ancestors of $\mathcal{X}_{j_1,\ldots,j_m}$. For example, the set of ancestors of $\mathcal{X}_{2,1,2}$ is $\mathcal{A}_{2,1,2} = \mathcal{X} \cup \mathcal{X}_2 \cup \mathcal{X}_{2,1}$. Thus, $\mathcal{A}_{j_1,\ldots,j_m} = \mathcal{A}_{j_1,\ldots,j_{m-1}} \cup \mathcal{X}_{j_1,\ldots,j_{m-1}}$, and the ancestor sets are nested: $\mathcal{A}_{j_1,\ldots,j_{m-1}} \subset \mathcal{A}_{j_1,\ldots,j_m}$. We can equivalently write (3) in terms of individual variables as

$$\hat{p}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i | \mathcal{C}_i), \tag{4}$$

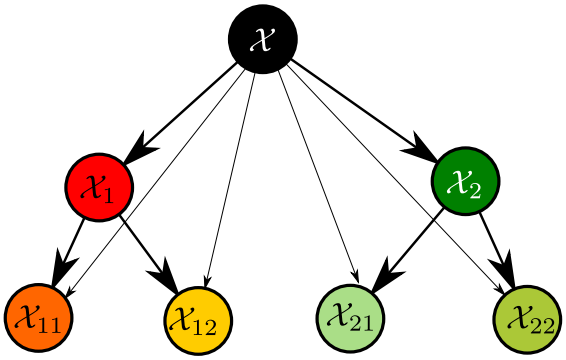
where $\mathcal{C}_i = \mathcal{A}_{j_1,\ldots,j_m} \cup \{x_k \in \mathcal{X}_{j_1,\ldots,j_m} : k < i\}$ for $x_i \in \mathcal{X}_{j_1,\ldots,j_m}$. The choice of the $\mathcal{C}_i$ involves a trade-off: generally, the larger the $\mathcal{C}_i$, the higher the computational cost (see Proposition 4 below), but the smaller the approximation error; HV is exact when all $\mathcal{C}_i = \{x_1,\ldots,x_{i-1}\}$.

Vecchia approximations and their conditional-independence assumptions are closely connected to directed acyclic graphs (DAGs; Datta et al., 2016; Katzfuss and Guinness, 2019). Summarizing briefly, as illustrated in Figure 1b, we associate a vertex with each set $\mathcal{X}_{j_1,\ldots,j_m}$, and we draw an arrow from the vertex corresponding to $\mathcal{X}_{i_1,\ldots,i_l}$ to the vertex corresponding to $\mathcal{X}_{j_1,\ldots,j_m}$ if and only if $\mathcal{X}_{i_1,\ldots,i_l}$ is in the conditioning set of $\mathcal{X}_{j_1,\ldots,j_m}$ (i.e., $\mathcal{X}_{i_1,\ldots,i_l} \subset \mathcal{A}_{j_1,\ldots,j_m}$). DAGs corresponding to HV approximations always have a tree structure, due to the nested ancestor sets. Necessary terminology and notation from graph theory is reviewed in Appendix A.

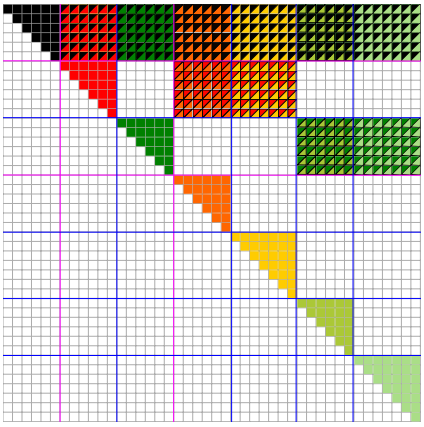
In practice, as illustrated in Figure 1a, we partition the spatial field $\mathbf{x}$ into the hierarchical set $\mathcal{X}^{0:M}$ based on a recursive partitioning of the spatial domain $\mathcal{D}$ into J_1 regions $\mathcal{D}_1,\ldots,\mathcal{D}_{J_1}$, each of which is again split into J_2 regions, and so forth, up to resolution M (Katzfuss, 2017):



(a) Iterative domain partitioning for $n = 35$ locations



(b) Directed acyclic graph (DAG)



(c) Sparsity pattern of the $\mathbf{U}$ matrix

Figure 1: Toy example with $n = 35$ of the hierarchical Vecchia approximation in (3) with $M = 2$ and $J_1 = J_2 = 2$; the color for each set $\mathcal{X}_{j_1,\ldots,j_m}$ is consistent across (a)–(c). (a) Partitioning of the spatial domain $\mathcal{D}$ and the locations $\mathcal{S}$; for resolution $m = 0, 1, 2$, locations of $\mathcal{X}^{0:m}$ (solid dots) and locations of points at finer resolutions ($\circ$). (b) DAG illustrating the conditional-dependence structure, with bigger arrows for connections between vertices at neighboring levels of the hierarchy, to emphasize the tree structure. (c) Corresponding sparsity pattern of $\mathbf{U}$ (see Proposition 1), with groups of columns/rows corresponding to different resolutions separated by pink lines, and groups of columns/rows corresponding to different $\mathcal{X}_{j_1,\ldots,j_m}$ at the same resolution separated by blue lines.

$\mathcal{D}_{j_1,\ldots,j_{m-1}} = \bigcup_{j_m=1}^{J_m} \mathcal{D}_{j_1,\ldots,j_m}$, $m = 1, \ldots, M$. We then set each $\mathcal{X}_{j_1,\ldots,j_m}$ to be a subset of the variables in $\mathbf{x}$ whose location is in $\mathcal{D}_{j_1,\ldots,j_m}$: $\mathcal{X}_{j_1,\ldots,j_m} \subset \{x_i : \mathbf{s}_i \in \mathcal{D}_{j_1,\ldots,j_m}\}$. This implies that the ancestors $\mathcal{A}_{j_1,\ldots,j_m}$ of each set $\mathcal{X}_{j_1,\ldots,j_m}$ consist of the variables associated with regions at lower resolutions $m = 0, \ldots, m-1$ that contain $\mathcal{D}_{j_1,\ldots,j_m}$. Specifically, for all our numerical examples, we set $J_1 = \ldots = J_M = 2$, and we select each $\mathcal{X}_{j_1,\ldots,j_m}$ corresponding to the first $|\mathcal{X}_{j_1,\ldots,j_m}|$ locations in a maximum-distance ordering (Guinness, 2018; Schäfer et al., 2017) of $\mathcal{S}$ that are contained in $\mathcal{D}_{j_1,\ldots,j_m}$ but are not already in $\mathcal{A}_{j_1,\ldots,j_m}$.

The HV approximation (3) is closely related to the multi-resolution approximation (Katzfuss, 2017; Katzfuss and Gong, 2019), as noted in Katzfuss and Guinness (2019, Sec. 2.5), which in turn is closely related to hierarchical off-diagonal low-rank (HODLR) matrices (e.g. Hackbusch, 2015; Ambikasaran et al., 2016; Saibaba et al., 2015; Geoga et al., 2018), as noted

in Jurek and Katzfuss (2018). However, the definition, exposition, and details provided here enable our later proofs, simple incomplete-Cholesky-based computation, and extensions to non-Gaussian data and to nonlinear space-time filtering.

3.2 Sparsity of the hierarchical Vecchia approximation

For all Vecchia approximations, the assumed conditional independence implies a sparse Cholesky factor of the precision matrix (e.g., Datta et al., 2016; Katzfuss and Guinness, 2019, Prop. 3.3). The conditional-independence assumption made in our HV approximation also implies a sparse Cholesky factor of the covariance matrix, which is in contrast to many other formulations of the Vecchia approximation:

PROPOSITION 1. *For the HV approximation in* (3)*, we have* $\hat{p}(\mathbf{x}) = \mathcal{N}_n(\mathbf{x}|\boldsymbol{\mu}, \hat{\boldsymbol{\Sigma}})$*. Define* $\mathbf{L} = \text{chol}(\hat{\boldsymbol{\Sigma}})$ *and* $\mathbf{U} = \text{rchol}(\hat{\boldsymbol{\Sigma}}^{-1}) = \mathbf{P}\,\text{chol}(\mathbf{P}\hat{\boldsymbol{\Sigma}}^{-1}\mathbf{P})\,\mathbf{P}$*, where* $\mathbf{P}$ *is the reverse-ordering permutation matrix.*

1. *For* $i \neq j$*:*
   - (a) $\mathbf{L}_{i,j} = 0$ *unless* $x_j \in \mathcal{C}_i$
   - (b) $\mathbf{U}_{j,i} = 0$ *unless* $x_j \in \mathcal{C}_i$
2. $\mathbf{U} = \mathbf{L}^{-\top}$

The proof relies on Claim 1. All proofs can be found in Appendix B. Proposition 1 says that the Cholesky factors of the covariance and precision matrix implied by a HV approximation are both sparse, and $\mathbf{U}$ has the same sparsity pattern as $\mathbf{L}^\top$. An example of this pattern is shown in Figure 1c. Furthermore, because $\mathbf{L} = \mathbf{U}^{-\top}$, we can quickly compute one of these factors given the other, as described in Section 3.3 below.

For other Vecchia approximations, the sparsity of the prior Cholesky factor $\mathbf{U}$ does not necessarily imply the same sparsity for the Cholesky factor of the posterior precision matrix, and in fact there can be substantial in-fill (Katzfuss and Guinness, 2019). However, this is not the case for the particular case of HV, for which the posterior sparsity is exactly the same as the prior sparsity:

PROPOSITION 2. *Assume that* $\mathbf{x}$ *has the distribution* $\hat{p}(\mathbf{x})$ *given by the HV approximation in* (3)*. Let* $\widetilde{\boldsymbol{\Sigma}} = \text{Var}(\mathbf{x}|\mathbf{y})$ *be the posterior covariance matrix of* $\mathbf{x}$ *given data* $y_i \,|\, \mathbf{x} \overset{ind}{\sim} \mathcal{N}(x_i, \tau_i^2)$*,* $i \in \mathcal{I} \subset \{1, \ldots, n\}$ *as in* (1)*. Then:*

1. $\widetilde{\mathbf{U}} = \text{rchol}(\widetilde{\boldsymbol{\Sigma}}^{-1})$ *has the same sparsity pattern as* $\mathbf{U} = \text{rchol}(\hat{\boldsymbol{\Sigma}}^{-1})$*.*
2. $\widetilde{\mathbf{L}} = \text{chol}(\widetilde{\boldsymbol{\Sigma}})$ *has the same sparsity pattern as* $\mathbf{L} = \text{chol}(\hat{\boldsymbol{\Sigma}})$*.*

3.3 Fast computation using incomplete Cholesky factorization

For notational and computational convenience, we assume now that each conditioning set $\mathcal{C}_i$ consists of at most N elements of $\mathbf{x}$. For example, this can be achieved by setting $|\mathcal{X}_{j_1,\ldots,j_m}| \leq r$ with $r = N/(M+1)$. Then $\mathbf{U}$ can be computed using general expressions for the

Vecchia approximation in $\mathcal{O}(nN^3)$ time (e.g., Katzfuss and Guinness, 2019). Alternatively, inference can be carried out using multi-resolution decompositions (Katzfuss, 2017; Katzfuss and Gong, 2019; Jurek and Katzfuss, 2018) in $\mathcal{O}(nN^2)$, but these algorithms are fairly involved.

Instead, we show here how HV inference can be carried out in $\mathcal{O}(nN^2)$ time using standard sparse-matrix algorithms, including the incomplete Cholesky factorization, based on at most nN entries of $\mathbf{\Sigma}$. Our algorithm, which is based on ideas in Schäfer et al. (2017), is much simpler than multi-resolution decompositions.

Algorithm 1: Incomplete Cholesky decomposition: ichol$(\mathbf{A}, \mathbf{S})$

Input: positive-definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, sparsity matrix $\mathbf{S} \in \{0,1\}^{n \times n}$
Result: lower-triangular $n \times n$ matrix $\mathbf{L}$

for $i = 1$ to n **do**
  for $j = 1$ to $i - 1$ **do**
    if $\mathbf{S}_{i,j} = 1$ **then**
      $\mathbf{L}_{i,j} = (\mathbf{A}_{i,j} - \sum_{k=1}^{j-1} \mathbf{L}_{i,k}\mathbf{L}_{j,k})/\mathbf{L}_{j,j}$
    end if
  end for
  $\mathbf{L}_{i,i} = (\mathbf{A}_{i,i} - \sum_{k=1}^{i-1} \mathbf{L}_{k,k})^{1/2}$
end for

The incomplete Cholesky factorization (e.g., Golub and Van Loan, 2012), denoted by ichol$(\mathbf{A}, \mathbf{S})$ and given in Algorithm 1, is identical to the standard Cholesky factorization of the matrix $\mathbf{A}$, except that we skip all operations that involve elements that are not in the sparsity pattern represented by the zero-one matrix $\mathbf{S}$. It is important to note that to compute $\mathbf{L} = \text{ichol}(\mathbf{A}, \mathbf{S})$ for a large dense matrix $\mathbf{A}$, we do not actually need to form or access the entire $\mathbf{A}$; instead, to reduce memory usage and computational cost, we simply compute $\mathbf{L} = \text{ichol}(\mathbf{A} \circ \mathbf{S}, \mathbf{S})$ based on the sparse matrix $\mathbf{A} \circ \mathbf{S}$, where $\circ$ denotes element-wise multiplication. Thus, while we write expressions like $\mathbf{L} = \text{ichol}(\mathbf{A}, \mathbf{S})$ for notational simplicity below, this should always be read as $\mathbf{L} = \text{ichol}(\mathbf{A} \circ \mathbf{S}, \mathbf{S})$.

For our HV approximation in (3), we henceforth set $\mathbf{S}$ to be a sparse lower-triangular matrix with $\mathbf{S}_{i,j} = 1$ if $x_j \in \mathcal{C}_i$, and 0 otherwise. Thus, the sparsity pattern of $\mathbf{S}$ is the same as that of $\mathbf{L}$, and its transpose is that of $\mathbf{U}$ shown in Figure 1c.

PROPOSITION 3. *Assuming* (3)*, denote* $\text{Var}(\mathbf{x}) = \hat{\mathbf{\Sigma}}$ *and* $\mathbf{L} = \text{chol}(\hat{\mathbf{\Sigma}})$*. Then,* $\mathbf{L} = \textit{ichol}(\mathbf{\Sigma}, \mathbf{S})$.

Hence, the Cholesky factor of the covariance matrix $\hat{\mathbf{\Sigma}}$ implied by the HV approximation can be computed using the incomplete Cholesky algorithm based on the (at most) nN entries of the exact covariance $\mathbf{\Sigma}$ indicated by $\mathbf{S}$. Using this result, we propose Algorithm 2 for posterior inference on $\mathbf{x}$ given $\mathbf{y}$.

By combining the incomplete Cholesky factorization with the results in Propositions 1 and 2 (saying that all involved Cholesky factors are sparse), we can perform fast posterior inference:

PROPOSITION 4. *Algorithm 2 can be carried out in* $\mathcal{O}(nN^2)$ *time and* $\mathcal{O}(nN)$ *space, assuming that* $|\mathcal{C}_i| \leq N$ *for all* $i = 1, \ldots, n$.

Algorithm 2: Posterior inference using hierarchical Vecchia: $\texttt{HV}(\mathbf{y}, \mathbf{S}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{H}, \mathbf{R})$

Input: data $\mathbf{y}$; sparsity $\mathbf{S}$; $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ s.t. $\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$; obs. matrix $\mathbf{H}$; noise variances $\mathbf{R}$
Result: $\widetilde{\boldsymbol{\mu}}$ and $\widetilde{\mathbf{L}}$ such that $\hat{p}(\mathbf{x}|\mathbf{y}) = \mathcal{N}_n(\mathbf{x}|\widetilde{\boldsymbol{\mu}}, \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top)$

1: $\mathbf{L} = \text{ichol}(\boldsymbol{\Sigma}, \mathbf{S})$, using Algorithm 1
2: $\mathbf{U} = \mathbf{L}^{-\top}$
3: $\boldsymbol{\Lambda} = \mathbf{U}\mathbf{U}^\top + \mathbf{H}^\top \mathbf{R}^{-1} \mathbf{H}$
4: $\widetilde{\mathbf{U}} = \mathbf{P}\left(\text{chol}(\mathbf{P}\boldsymbol{\Lambda}\mathbf{P})\right)\mathbf{P}$, where $\mathbf{P}$ is the order-reversing permutation matrix
5: $\widetilde{\mathbf{L}} = \widetilde{\mathbf{U}}^{-\top}$
6: $\widetilde{\boldsymbol{\mu}} = \boldsymbol{\mu} + \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top \mathbf{H}^\top \mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\boldsymbol{\mu})$

4 Extensions to non-Gaussian spatial data using the Laplace approximation

Now consider the model

$$y_i \,|\, \mathbf{x} \overset{ind}{\sim} g_i(y_i|x_i), \qquad i \in \mathcal{I}, \tag{5}$$

$$\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{6}$$

where g_i is a distribution from an exponential family. Using the HV approximation in (3)–(4) for $\mathbf{x}$, the implied posterior can be written as:

$$\hat{p}(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{x})\hat{p}(\mathbf{x})}{\int p(\mathbf{y}|\mathbf{x})\hat{p}(\mathbf{x})d\mathbf{x}} = \frac{(\prod_{i\in\mathcal{I}} g_i(y_i|x_i))\hat{p}(\mathbf{x})}{\int (\prod_{i\in\mathcal{I}} g_i(y_i|x_i))\hat{p}(\mathbf{x})d\mathbf{x}}. \tag{7}$$

Unlike in the Gaussian case as in (1), the integral in the denominator cannot generally be evaluated in closed form, and Markov Chain Monte Carlo methods are often used to numerically approximate the posterior. Instead, Zilber and Katzfuss (2019) proposed a much faster method that combines a general Vecchia approximation with the Laplace approximation (e.g. Tierney and Kadane, 1986; Rasmussen and Williams, 2006, Sect. 3.4). The Laplace approximation is combined with a Gaussian approximation of the posterior, obtained by carrying out a second-order Taylor expansion of the posterior log-density around its mode. Although the mode cannot generally be obtained in closed form, it can be computed straightforwardly using a Newton-Raphson procedure, because $\log \hat{p}(\mathbf{x}|\mathbf{y}) = \log p(\mathbf{y}|\mathbf{x}) + \log \hat{p}(\mathbf{x}) + c$ is a sum of two concave functions and hence also concave (as a function of $\mathbf{x}$, under appropriate parametrization of the g_i).

While each Newton-Raphson update requires the computation and decomposition of the $n \times n$ Hessian matrix, the update can be carried out quickly by making use of the sparsity implied by the Vecchia approximation. To do so, we follow Zilber and Katzfuss (2019) in exploiting the fact that the Newton-Raphson update is equivalent to computing the conditional mean of $\mathbf{x}$ given pseudo-data. Specifically, at the ℓ-th iteration of the algorithm, given the current state value $\mathbf{x}^{(\ell)}$, let us define

$$\mathbf{u}^{(\ell)} = \left[u_i^{(\ell)}\right]_{i\in\mathcal{I}}, \qquad \text{where} \quad u_i^{(\ell)} = \tfrac{\partial}{\partial x} \log g_i(y_i|x)\big|_{x=x_i^{(\ell)}}, \tag{8}$$

and

$$\mathbf{D}^{(\ell)} = \operatorname{diag}\left(\{d_i^{(\ell)} : i \in \mathcal{I}\}\right), \qquad \text{where} \quad d_i^{(\ell)} = -\left(\tfrac{\partial^2}{\partial x^2} \log g_i(y_i|x)\right)^{-1}\Big|_{x=x_i^{(\ell)}}. \tag{9}$$

Then, we compute the next iteration's state value $\mathbf{x}^{(\ell+1)} = \mathbb{E}(\mathbf{x}|\mathbf{t}^{(\ell)})$ as the conditional mean of $\mathbf{x}$ given pseudo-data $\mathbf{t}^{(\ell)} = \mathbf{x}^{(\ell)} + \mathbf{D}^{(\ell)}\mathbf{u}^{(\ell)}$ assuming Gaussian noise, $t_i^{(\ell)}|\mathbf{x} \overset{ind.}{\sim} \mathcal{N}(x_i, d_i^{(\ell)})$, $i \in \mathcal{I}$. Zilber and Katzfuss (2019) recommend computing the conditional mean $\mathbb{E}(\mathbf{x}|\mathbf{t}^{(\ell)})$ based on a general-Vecchia-prediction approach proposed in Katzfuss et al. (2020a). Here, we instead compute the posterior mean using Algorithm 2 based on the HV method described in Section 3, due to its sparsity-preserving properties. In contrast to the approach recommended in Zilber and Katzfuss (2019), our algorithm is guaranteed to converge, because it is equivalent to Newton-Raphson optimization of the log of the posterior density in (7), which is concave. Once the algorithm converges to the posterior mode $\widetilde{\boldsymbol{\mu}}$, we obtain a Gaussian HV-Laplace approximation of the posterior as

$$\hat{p}_L(\mathbf{x}|\mathbf{y}) = \mathcal{N}_n(\mathbf{x}|\widetilde{\boldsymbol{\mu}}, \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top),$$

where $\widetilde{\mathbf{L}}$ is the Cholesky factor of the negative Hessian of the log-posterior at $\widetilde{\boldsymbol{\mu}}$. Our approach is described in Algorithm 3. The main computational expense for each iteration of the `for` loop is carrying out Algorithm 2, and so each iteration requires only $\mathcal{O}(nN^2)$ time.

Algorithm 3: Hierarchical-Vecchia-Laplace inference: $\texttt{HVL}(\mathbf{y}, \mathbf{S}, \boldsymbol{\mu}, \boldsymbol{\Sigma}, \{g_i\})$

Input: data $\mathbf{y}$; sparsity $\mathbf{S}$; $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ such that $\mathbf{x} \sim \mathcal{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$; likelihoods $\{g_i : i \in \mathcal{I}\}$
Result: $\widetilde{\boldsymbol{\mu}}$ and $\widetilde{\mathbf{L}}$ such that $\hat{p}_L(\mathbf{x}|\mathbf{y}) = \mathcal{N}_n(\mathbf{x}|\widetilde{\boldsymbol{\mu}}, \widetilde{\mathbf{L}}\widetilde{\mathbf{L}}^\top)$

```
Initialize x^(0) = μ
Set H = I_{I,:} as the rows I of the n × n identity matrix I
for ℓ = 0, 1, 2, ... do
    Calculate u^(ℓ) as in (8), D^(ℓ) as in (9), and t^(ℓ) = x^(ℓ) + D^(ℓ) u^(ℓ)
    Calculate [x^(ℓ+1), L̃] = HV(t^(ℓ), S, μ, Σ, H, D^(ℓ)) using Algorithm 2
    if ||x^(ℓ+1) − x^(ℓ)|| / ||x^(ℓ)|| < ϵ then
        break
    end if
end for
return μ̃ = x^(ℓ+1) and L̃
```

In the Gaussian case, when $g_i(y_i|x_i) = \mathcal{N}(y_i|a_i x_i, \tau_i^2)$ for some $a_i \in \mathbb{R}$, it can be shown using straightforward calculations that the pseudo-data $t_i = y_i/a_i$ and pseudo-variances $d_i = \tau_i^2$ do not depend on $\mathbf{x}$, and so Algorithm 3 converges in a single iteration. If, in addition, $a_i = 1$ for all $i = 1, \ldots, n$, then (5) becomes equivalent to (1), and Algorithm 3 simplifies to Algorithm 2. For non-Gaussian data, our Laplace and Gaussian approximations introduce an additional source of error. While this error is difficult to quantify theoretically, empirical studies (e.g., Bonat and Ribeiro Jr, 2016; Zilber and Katzfuss, 2019) have shown that Laplace-type approximations can be very accurate and can strongly outperform sampling-based approaches such as Markov Chain Monte Carlo.

5 Fast filters for spatio-temporal models

5.1 Linear evolution

We now turn to a spatio-temporal state-space model (SSM), which adds a temporal evolution model to the spatial model (5) considered in Section 4. For now, assume that the evolution is linear. Starting with an initial distribution $\mathbf{x}_0 \sim \mathcal{N}_n(\boldsymbol{\mu}_{0|0}, \boldsymbol{\Sigma}_{0|0})$, we consider the following SSM for discrete time $t = 1, 2, \ldots$:

$$y_{ti} \,|\, \mathbf{x}_t \overset{ind}{\sim} g_{ti}(y_{ti}|x_{ti}), \qquad i \in \mathcal{I}_t \tag{10}$$

$$\mathbf{x}_t \,|\, \mathbf{x}_{t-1} \sim \mathcal{N}_n(\mathbf{E}_t \mathbf{x}_{t-1}, \mathbf{Q}_t), \tag{11}$$

where $\mathbf{y}_t$ is the data vector consisting of $n_t \leq n$ observations $\{y_{ti} : i \in \mathcal{I}_t\}$, $\mathcal{I}_t \subset \{1, \ldots, n\}$ contains the observation indices at time t, g_{ti} is a distribution from the exponential family, $\mathbf{x}_t = (x_1, \ldots, x_n)^\top$ is the latent spatial field of interest at time t observed at a spatial grid $\mathcal{S}$, and $\mathbf{E}_t$ is a sparse $n \times n$ evolution matrix.

At time t, our goal is to obtain or approximate the filtering distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ of $\mathbf{x}_t$ given data $\mathbf{y}_{1:t}$ up to the current time t. This task, also referred to as data assimilation or online inference, is commonly encountered in many fields of science whenever one is interested in quantifying the uncertainty in the current state or in obtaining forecasts into the future. If the observation equations g_{ti} are all Gaussian, the filtering distribution can be derived using the Kalman filter (Kalman, 1960) for small to moderate n. At each time t, the Kalman filter consist of a forecast step that computes $p(\mathbf{x}_t|\mathbf{y}_{1:t-1})$, and an update step which then obtains $p(\mathbf{x}_t|\mathbf{y}_{1:t})$. For linear Gaussian SSMs, both of these distributions are multivariate normal.

Our Kalman-Vecchia-Laplace (KVL) filter extends the Kalman filter to high-dimensional SSMs (i.e., large n) with non-Gaussian data, as in (10)–(11). Its update step is very similar to the inference problem in Section 4, and hence it essentially consists of the HVL in Algorithm 3. We complement this update with a forecast step, in which the moment estimates are propagated forward using the temporal evolution model. This forecast step is exact, and so the KVL approximation error is solely due to the HVL approximation at each update step. The KVL filter is given in Algorithm 4.

Algorithm 4: Kalman-Vecchia-Laplace (KVL) filter

Input: $\mathbf{S}$, $\boldsymbol{\mu}_{0|0}$, $\boldsymbol{\Sigma}_{0|0}$, $\{(\mathbf{y}_t, \mathbf{E}_t, \mathbf{Q}_t, \{g_{t,i}\}) : t = 1, 2, \ldots\}$
Result: $\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}$, such that $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}_n(\mathbf{x}_t|\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}\mathbf{L}_{t|t}^\top)$

1: Compute $\mathbf{U}_{0|0} = \text{ichol}(\boldsymbol{\Sigma}_{0|0}, \mathbf{S})$ and $\mathbf{L}_{0|0} = \mathbf{U}_{0|0}^{-\top}$
2: **for** $t = 1, 2, \ldots$ **do**
3: Forecast: $\boldsymbol{\mu}_{t|t-1} = \mathbf{E}_t \boldsymbol{\mu}_{t-1|t-1}$ and $\mathbf{L}_{t|t-1} = \mathbf{E}_t \mathbf{L}_{t-1|t-1}$
4: For all (i, j) with $\mathbf{S}_{i,j} = 1$: $\boldsymbol{\Sigma}_{t|t-1;i,j} = \mathbf{L}_{t|t-1;i,:}\mathbf{L}_{t|t-1;j,:}^\top + \mathbf{Q}_{t;i,j}$
5: Update: $[\boldsymbol{\mu}_t, \mathbf{L}_{t|t}] = \texttt{HVL}(\mathbf{y}_t, \mathbf{S}, \boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1}, \{g_{t,i}\})$ using Algorithm 3
6: **return** $\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}$
7: **end for**

In Line 4, $\mathbf{L}_{t|t-1;i,:}$ denotes the ith row of $\mathbf{L}_{t|t-1}$. The KVL filter scales well with the state dimension n. The evolution matrix $\mathbf{E}_t$, which is often derived using a forward-finite-difference scheme and thus has only a few nonzero elements in each row, can be quickly

multiplied with $\mathbf{L}_{t-1|t-1}$ in Line 3, as the latter is sparse (see Section 3.3). The $\mathcal{O}(nN)$ necessary entries of $\mathbf{\Sigma}_{t|t-1}$ in Line 4 can also be calculated quickly due to the sparsity of $\mathbf{L}_{t|t-1;i,:}$. The low computational cost of the `HVL` algorithm has already been discussed in Section 4. Thus, assuming sufficiently sparse $\mathbf{E}_t$, the KVL filter scales approximately as $\mathcal{O}(nN^2)$ per iteration. In the case of Gaussian data (i.e., all g_{ti} in (10) are Gaussian), our KVL filter will produce essentially equivalent filtering distributions as the more complicated multi-resolution filter of Jurek and Katzfuss (2018).

5.2 An extended filter for nonlinear evolution

Finally, we consider a nonlinear and non-Gaussian model, which extends (10)–(11) by allowing nonlinear evolution operators, $\mathcal{E}_t : \mathbb{R}^n \to \mathbb{R}^n$. This results in the model

$$y_{ti} \,|\, \mathbf{x}_t \overset{ind}{\sim} g_{ti}(y_{ti}|x_{ti}), \qquad i \in \mathcal{I}_t \tag{12}$$

$$\mathbf{x}_t \,|\, \mathbf{x}_{t-1} \sim \mathcal{N}_n(\mathcal{E}_t(\mathbf{x}_{t-1}), \mathbf{Q}_t). \tag{13}$$

Due to the nonlinearity of the evolution operator $\mathcal{E}_t$, the KVL filter in Algorithm 4 is not directly applicable anymore. However, similar inference is still possible as long as the evolution is not too far from linear. Approximating the evolution as linear is generally reasonable if the time steps are short, or if the measurements are highly informative. In this case, we propose the extended Kalman-Vecchia-Laplace filter (EKVL) in Algorithm 5, which approximates the extended Kalman filter (e.g., Grewal and Andrews, 1993, Ch. 5) and extends it to non-Gaussian data using the Vecchia-Laplace aproach. For the forecast step, EKVL computes the forecast mean as $\boldsymbol{\mu}_{t|t-1} = \mathcal{E}_t(\boldsymbol{\mu}_{t-1|t-1})$. The forecast covariance matrix $\mathbf{\Sigma}_{t|t-1}$ is obtained as before, after approximating the evolution using the Jacobian as $\mathbf{E}_t = \frac{\partial \mathcal{E}_t(\mathbf{y}_{t-1})}{\partial \mathbf{y}_{t-1}}\Big|_{\mathbf{y}_{t-1}=\boldsymbol{\mu}_{t-1|t-1}}$. Errors in the forecast covariance matrix due to this linear approximation can be captured in the innovation covariance, $\mathbf{Q}_t$. If the Jacobian matrix cannot be computed, it is sometimes possible to build a statistical emulator (e.g., Kaufman et al., 2011) instead, which approximates the true evolution operator.

Once $\boldsymbol{\mu}_{t|t-1}$ and $\mathbf{\Sigma}_{t|t-1}$ have been obtained, the update step of the EKVL proceeds exactly as in the KVL filter by approximating the forecast distribution as Gaussian.

Algorithm 5: Extended Kalman-Vecchia-Laplace (EKVL) filter

Input: $\mathbf{S}$, $\boldsymbol{\mu}_{0|0}$, $\mathbf{\Sigma}_{0|0}$, $\{(\mathbf{y}_t, \mathcal{E}_t, \mathbf{Q}_t, \{g_{t,i}\}) : t = 1, 2, \ldots\}$

Result: $\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}$, such that $\hat{p}(\mathbf{x}_t|\mathbf{y}_{1:t}) = \mathcal{N}_n(\mathbf{x}_t|\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}\mathbf{L}_{t|t}^\top)$

1: Compute $\mathbf{U}_{0|0} = \text{ichol}(\mathbf{\Sigma}_{0|0}, \mathbf{S})$ and $\mathbf{L}_{0|0} = \mathbf{U}_{0|0}^{-\top}$

2: **for** $t = 1, 2, \ldots$ **do**

3: Calculate $\mathbf{E}_t = \frac{\partial \mathcal{E}_t(\mathbf{x}_{t-1})}{\partial \mathbf{x}_{t-1}}\Big|_{\mathbf{x}_{t-1}=\boldsymbol{\mu}_{t-1|t-1}}$

4: Forecast: $\boldsymbol{\mu}_{t|t-1} = \mathcal{E}_t(\boldsymbol{\mu}_{t-1|t-1})$ and $\mathbf{L}_{t|t-1} = \mathbf{E}_t\mathbf{L}_{t-1|t-1}$

5: For all (i, j) with $\mathbf{S}_{i,j} = 1$: $\mathbf{\Sigma}_{t|t-1;i,j} = \mathbf{L}_{t|t-1;i,:}\mathbf{L}_{t|t-1;j,:}^\top + \mathbf{Q}_{t;i,j}$

6: Update: $[\boldsymbol{\mu}_t, \mathbf{L}_{t|t}] = \texttt{HVL}(\mathbf{y}_t, \mathbf{S}, \boldsymbol{\mu}_{t|t-1}, \mathbf{\Sigma}_{t|t-1}, \{g_{t,i}\})$ using Algorithm 3

7: **return** $\boldsymbol{\mu}_{t|t}, \mathbf{L}_{t|t}$

8: **end for**

Similarly to Algorithm 4, EKVL scales very well with the dimension of $\mathbf{x}$, the only difference being the additional operation of calculating the Jacobian in Line 3, whose cost is problem dependent. Only those entries of $\mathbf{E}_t$ need to be calculated that are multiplied with non-zero entries of $\mathbf{L}_{t-1|t-1}$, whose sparsity structure is known ahead of time.

5.3 A particle-EKVL filter in case of unknown parameters

The distributions and matrices in model (12)–(13) may depend on parameters $\boldsymbol{\theta}_t$ at each time t, which we have implicitly assumed to be known thus far. We now discuss the case of a (small) number of unknown parameters $\boldsymbol{\theta}_t$. Specifically, $\boldsymbol{\mu}_{0|0}$ and $\boldsymbol{\Sigma}_{0|0}$ may depend on $\boldsymbol{\theta}_0$, and the quantities $\{g_{t,i}\}$, $\mathcal{E}_t$, and $\mathbf{Q}_t$ at each time t may depend on $\boldsymbol{\theta}_t$. There are two main approaches to simultaneous filtering for the state $\mathbf{x}_t$ and the parameters $\boldsymbol{\theta}_t$: state augmentation and Rao-Blackwellized filters (Doucet and Johansen, 2011). The main idea behind the former is to include $\boldsymbol{\theta}_t$ in the state vector $\mathbf{x}_t$ and to modify the evolution and the model error matrices accordingly, but this approach is known to work poorly in certain cases (e.g., DelSole and Yang, 2010; Katzfuss et al., 2020c). Thus, following Jurek and Katzfuss (2018), we now present a Rao-Blackwellized filter in which integration over $\mathbf{x}_t$ is performed based on our HVL approximation.

Writing $\boldsymbol{\theta}_{0:t} = (\boldsymbol{\theta}_0, \ldots, \boldsymbol{\theta}_t)$, the integrated likelihood at time t is given by

$$p(\mathbf{y}_{1:t}|\boldsymbol{\theta}_{0:t}) = p(\mathbf{y}_1|\boldsymbol{\theta}_{0:1}) \prod_{k=2}^{t} p(\mathbf{y}_k|\mathbf{y}_{1:k-1}, \boldsymbol{\theta}_{0:k}).$$

It is well known that

$$p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{0:t}) = \frac{p(\mathbf{y}_t, \mathbf{x}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{0:t})}{p(\mathbf{x}_t|\mathbf{y}_{1:t}, \boldsymbol{\theta}_{0:t})} = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t) p(\mathbf{x}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{0:t})}{p(\mathbf{x}_t|\mathbf{y}_{1:t}, \boldsymbol{\theta}_{0:t})},$$

where $p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t)$ is available in closed form from (12), and the forecast and filtering distributions can be approximated using the EKVL, to obtain

$$\mathcal{L}_t(\boldsymbol{\theta}_{0:t}) := \hat{p}(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{0:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t, \boldsymbol{\theta}_t) \mathcal{N}(\boldsymbol{\mu}_{t|t-1}, \boldsymbol{\Sigma}_{t|t-1})}{\mathcal{N}(\boldsymbol{\mu}_{t|t}, \boldsymbol{\Sigma}_{t|t})}. \tag{14}$$

The normal densities can be quickly evaluated for given parameter values $\boldsymbol{\theta}_{0:t}$, because Algorithm 5 calculates sparse Cholesky factors of their precision matrices. For $t = 1$, the term $\mathcal{L}_1(\boldsymbol{\theta}_{0:1}) := \hat{p}(\mathbf{y}_1|\boldsymbol{\theta}_{0:1})$ can be approximated in a similar way using $\boldsymbol{\mu}_{0|0}$ and $\boldsymbol{\Sigma}_{0|0}$.

The particle-EKVL filter is given by Algorithm 6, assuming that the parameter priors are given by $f_0(\boldsymbol{\theta}_0)$ and then recursively by $f_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$.

6 Numerical comparison

6.1 Methods and criteria

We considered and compared the following methods:

Hierarchical Vecchia (HV): Our methods as described in this paper.

Algorithm 6: Particle-EKVL filter

Input: $\mathbf{S}$, $\boldsymbol{\mu}_{0|0}$, $\boldsymbol{\Sigma}_{0|0}$, $\{(\mathbf{y}_t, \mathcal{E}_t, \mathbf{Q}_t, \{g_{t,i}\}) : t = 1, 2, \ldots\}$, priors $\{f_t\}$, proposal distributions $\{q_t\}$, desired number of particles N_p

Result: $\{(\boldsymbol{\theta}_t^{(l)}, w_t^{(l)}, \boldsymbol{\mu}_{t|t}^{(l)}, \mathbf{L}_{t|t}^{(l)}) : l = 1, \ldots, N_p\}$, such that $\hat{p}(\boldsymbol{\theta}_t, \mathbf{x}_t | \mathbf{y}_{1:t}) = \sum_{l=1}^{N_p} w_t^{(l)} \delta_{\boldsymbol{\theta}_t^{(l)}}(\boldsymbol{\theta}_t) \mathcal{N}_n(\mathbf{x}_t | \boldsymbol{\mu}_{t|t}^{(l)}, \mathbf{L}_{t|t}^{(l)} \mathbf{L}_{t|t}^{(l)\top})$

for $l = 1, 2, \ldots, N_p$ **do**
  Draw $\boldsymbol{\theta}_0^{(l)} \sim f_0(\boldsymbol{\theta}_0)$ and set weight $w_0^{(l)} = 1/N_p$
  Compute $\mathbf{L}_{0|0}(\boldsymbol{\theta}_0^{(l)}) = \text{ichol}(\boldsymbol{\Sigma}_{0|0}(\boldsymbol{\theta}_0^{(l)}), \mathbf{S})$
  Compute $\boldsymbol{\mu}_{0|0}^{(l)}(\boldsymbol{\theta}_0^{(l)})$ and $\mathbf{U}_{0|0}(\boldsymbol{\theta}_0^{(l)}) = \mathbf{L}_{0|0}^{-\top}(\boldsymbol{\theta}_0^{(l)})$
end for
for $t = 1, 2, \ldots$ **do**
  for $l = 1, 2, \ldots, N_p$ **do**
    Draw $\boldsymbol{\theta}_t^{(l)} \sim q_t(\boldsymbol{\theta}_t^{(l)} | \boldsymbol{\theta}_{t-1}^{(l)})$
    Calculate $\mathbf{E}_t^{(l)} = \frac{\partial \mathcal{E}_t(\mathbf{y}_{t-1}, \boldsymbol{\theta}_t^{(l)})}{\partial \mathbf{y}_{t-1}}\Big|_{\mathbf{y}_{t-1} = \boldsymbol{\mu}_{t-1|t-1}(\boldsymbol{\theta}_{t-1}^{(l)})}$
    Forecast: $\boldsymbol{\mu}_{t|t-1}^{(l)} = \mathcal{E}_t(\boldsymbol{\mu}_{t-1|t-1}, \boldsymbol{\theta}_{t-1}^{(l)})$ and $\mathbf{L}_{t|t-1}^{(l)} = \mathbf{E}_t^{(l)} \mathbf{L}_{t-1|t-1}^{(l)}$
    For (i, j) s.t. $\mathbf{S}_{i,j} = 1$: $\boldsymbol{\Sigma}_{t|t-1;i,j}^{(l)} = \mathbf{L}_{t|t-1;i,:}^{(l)} (\mathbf{L}_{t|t-1;j,:}^{(l)})^\top + \mathbf{Q}_{t;i,j}(\boldsymbol{\theta}_t^{(l)})$
    Update: $[\boldsymbol{\mu}_t^{(l)}, \mathbf{L}_{t|t}^{(l)}] = \texttt{HVL}(\mathbf{y}_t, \mathbf{S}, \boldsymbol{\mu}_{t|t-1}^{(l)}, \boldsymbol{\Sigma}_{t|t-1}^{(l)}, \{g_{t,i}(\boldsymbol{\theta}_t^{(l)})\})$
    Calculate $\mathcal{L}_t(\boldsymbol{\theta}_{0:t}^{(l)})$ as in (14)
    Update particle weight $w_t^{(l)} \propto w_{t-1}^{(l)} \mathcal{L}_t(\boldsymbol{\theta}_{0:t}^{(l)}) f_t(\boldsymbol{\theta}_t^{(l)} | \boldsymbol{\theta}_{t-1}^{(l)}) / q_t(\boldsymbol{\theta}_t^{(l)} | \boldsymbol{\theta}_{t-1}^{(l)})$
    return $\boldsymbol{\mu}_{t|t}^{(l)}, \mathbf{L}_{t|t}^{(l)}, \boldsymbol{\theta}_t^{(l)}, w_t^{(l)}$
  end for
  Resample $\{(\boldsymbol{\theta}_t^{(l)}, \boldsymbol{\mu}_{t|t}^{(l)}, \mathbf{L}_{t|t}^{(l)})\}_{l=1}^{N_p}$ with weights $\{w_t^{(l)}\}_{l=1}^{N_p}$ to obtain equally weighted particles (e.g., Douc et al., 2005)
end for

Low rank (LR): A special case of HV with $M = 1$, in which the diagonal and the first N columns of $\mathbf{S}$ are nonzero, and all other entries are zero. This results in a matrix approximation $\hat{\boldsymbol{\Sigma}}$ that is of rank N plus diagonal, known as the modified predictive process (Banerjee et al., 2008; Finley et al., 2009) in spatial statistics. LR has the same computational complexity as HV.

Dense Laplace (DL): A further special case of HV with $M = 0$, in which $\mathbf{S}$ is a fully dense matrix of ones. Thus, there is no error due to the Vecchia approximation, and so in the non-Gaussian spatial-only setting, this is equivalent to a Gaussian Laplace approximation. DL will generally be more accurate than HV and low-rank, but it scales as $\mathcal{O}(n^3)$ and is thus not feasible for high dimension n.

For each scenario below, we simulated observations using (12), taking $g_{t,i}$ to be each of four exponential-family distributions: Gaussian, $\mathcal{N}(x, \tau^2)$; logistic Bernoulli, $\mathcal{B}(1/(1+e^{-x}))$; Poisson, $\mathcal{P}(e^x)$; and gamma, $\mathcal{G}(a, ae^{-x})$, with shape parameter $a = 2$. For most scenarios, we assumed a moderate state dimension n, so that DL remained feasible; a large n was

considered in Section 6.4.

The main metric to compare HV and LR was the difference in KL divergence between their posterior or filtering distributions and those generated by DL; as the exact distributions were not known here, we approximated this metric by the average difference in log scores (dLS; e.g., Gneiting and Katzfuss, 2014) over several simulations. We also calculated the relative root mean square prediction error (RRMSPE), defined as the root mean square prediction error of HV and LR, respectively, divided by the root mean square prediction error of DL. For both criteria, lower values are better.

6.2 Spatial-only data

In our first scenario, we considered spatial-only data according to (5)–(6) on a grid $\mathcal{S}$ of size $n = 34 \times 34 = 1{,}156$ on the unit square, $\mathcal{D} = [0,1]^2$. We set $\boldsymbol{\mu} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{i,j} = \exp(-\|\mathbf{s}_i - \mathbf{s}_j\|/0.15)$. For the Gaussian likelihood, we assumed variance $\tau^2 = 0.2$.

The comparison scores averaged over 100 simulations for the posteriors obtained using Algorithm 3 are shown as a function of N in Figure 2. HV (Algorithm 2) was much more accurate than LR for each value of N.

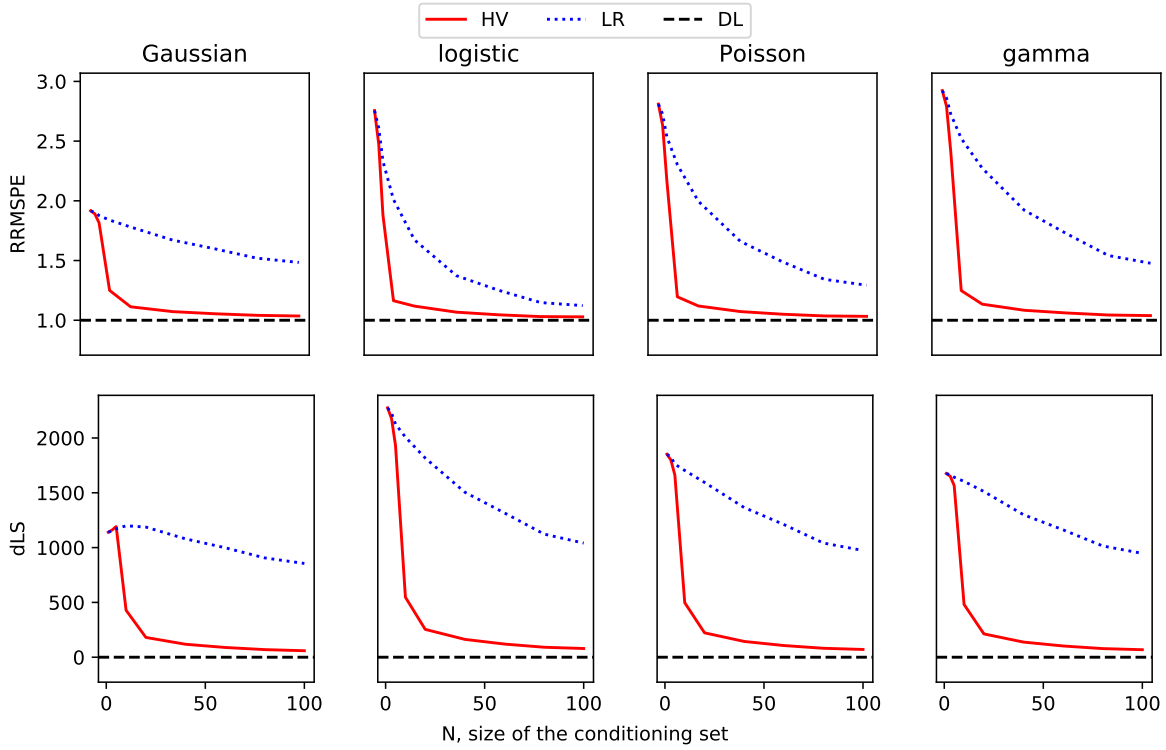


Figure 2: Approximation accuracy for the posterior distribution $\mathbf{x}|\mathbf{y}$ for spatial data (see Section 6.2)

6.3 Linear temporal evolution

Next, we considered a linear spatio-temporal advection-diffusion process with diffusion parameter $\alpha = 4 \times 10^{-5}$ and advection parameter $\beta = 10^{-2}$ as in Jurek and Katzfuss (2018).

The spatial domain $\mathcal{D} = [0,1]^2$ was discretized on a grid of size $n = 34 \times 34 = 1{,}156$ using the centered finite differences, and we considered discrete time points $t = 1, \ldots, T$ with $T = 20$. After this discretization, our model was of the form (10)–(11), where $\mathbf{\Sigma}_{0|0} = \mathbf{Q}_1 = \ldots = \mathbf{Q}_T$ with (i,j)th entry $\exp(-\|\mathbf{s}_i - \mathbf{s}_j\|/0.15)$, and $\mathbf{E}_t$ was a sparse matrix with nonzero entries corresponding to interactions between neighboring grid points to the right, left, top and bottom. See the supplementary material of Jurek and Katzfuss (2018) for details. At each time t, we generated $n_t = 0.1n$ observations with indices $\mathcal{I}_t$ sampled randomly from $\{1, \ldots, n\}$. For the Gaussian case, we assumed variance $\tau^2 = 0.25$. We used conditioning sets of size at most $N = 41$ for both HV and LR; specifically, for HV, we used $J = 2$ partitions at $M = 7$ resolutions, with set sizes $|\mathcal{X}_{j_1,\ldots,j_m}|$ of $5, 5, 5, 5, 6, 6, 6$, respectively, for $m = 0, 1, \ldots, M-1$, and $|\mathcal{X}_{j_1,\ldots,j_M}| \leq 3$.

Figure 3 compares the scores for the filtering distributions $\mathbf{x}_t|\mathbf{y}_{1:t}$ obtained using Algorithm 4, averaged over 80 simulations. Again, HV was much more accurate than LR. Importantly, while the accuracy of HV was relatively stable over time, LR became less accurate over time, with the approximation error accumulating.

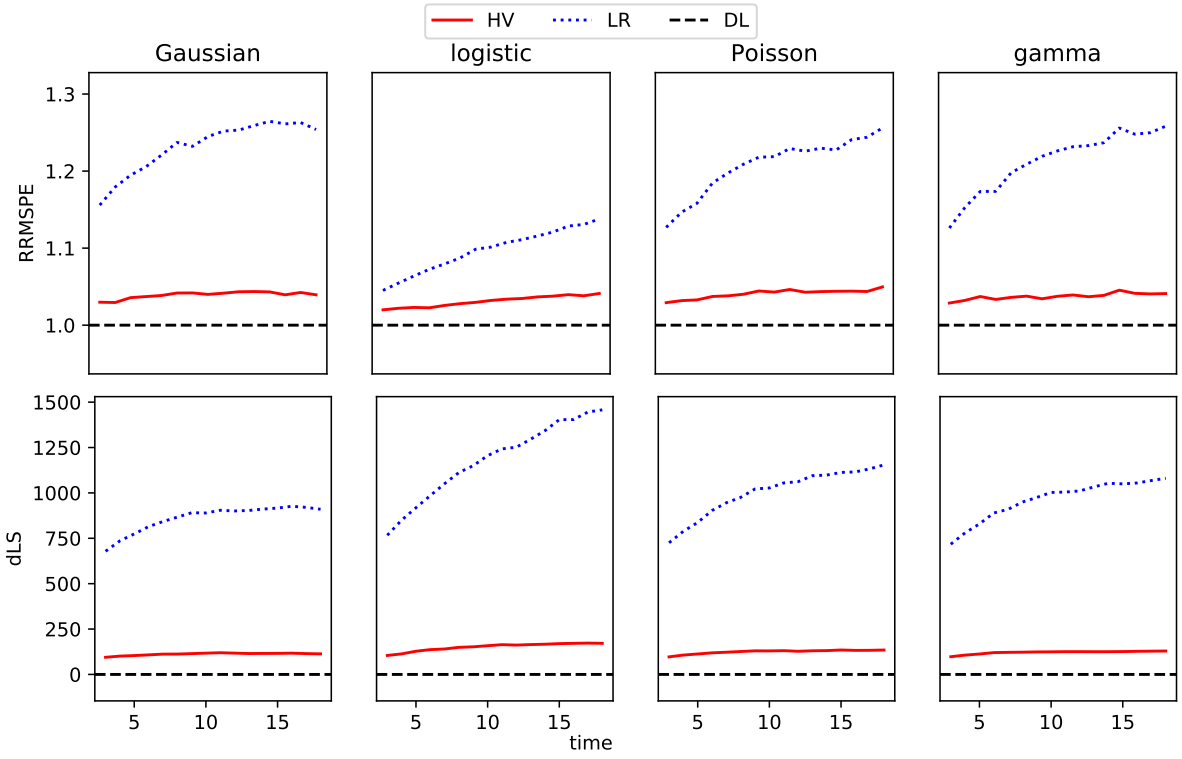


Figure 3: Accuracy of filtering distributions $\mathbf{x}_t|\mathbf{y}_{1:t}$ for the advection-diffusion model in Section 6.3

6.4 Simulations using a very large n

We repeated the advection-diffusion experiment from Section 6.3 on a high-resolution grid of size $n = 300 \times 300 = 90{,}000$, with $n_t = 9{,}000$ observations corresponding to 10% of the grid points. In order to avoid numerical artifacts related to the finite differencing scheme, we reduced the advection and diffusion coefficients to $\alpha = 10^{-7}$ and $\beta = 10^{-3}$, respectively. We

set $N = 44$, $M = 14$, $J = 2$, and $|\mathcal{X}_{j_1,\ldots,j_m}| = 3$ for $m = 0, 1, \ldots, M-1$, and $|\mathcal{X}_{j_1,\ldots,j_M}| \leq 2$. DL was too computationally expensive due to the high dimension n, and so we simply compared HV and LR based on the root mean square prediction error (RMSPE) between the true state and their respective filtering means, averaged over 10 simulations.

As shown in Figure 4, HV was again much more accurate than LR. Comparing to Figure 3, we see that the relative improvement of HV to LR increased even further; taking the Gaussian case as an example, the ratio of the RMSPE for HV and LR was around 1.2 in the small-n setting, and greater than 2 in the large-n setting.

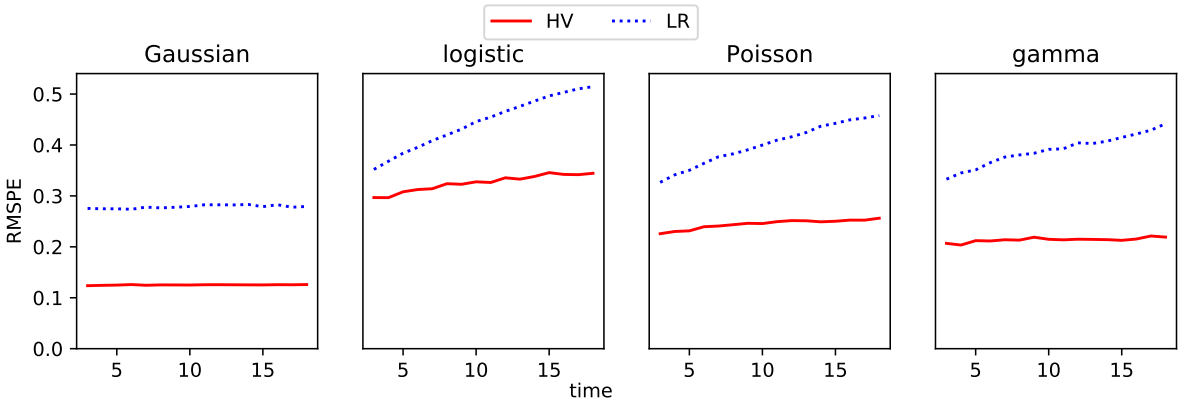


Figure 4: Root mean square prediction error (RMSPE) for the filtering mean in the high-dimensional advection-diffusion model with $n = 90{,}000$ in Section 6.4

6.5 Nonlinear evolution with non-Gaussian data

Our final set of our simulations involves the most general model (12)-(13) with nonlinear evolution $\mathcal{E}_t$. Specifically, we considered a complicated model (Lorenz, 2005, Sect. 3) that realistically replicates many features of atmospheric variables along a latitudinal band. A special case of this model (Lorenz, 1996) is an important benchmark for data-assimilation techniques. The model dynamics are described by

$$\frac{\partial}{\partial t}\tilde{x}_i = \frac{1}{K^2}\sum_{l=-K/2}^{K/2}\sum_{j=-K/2}^{K/2} -\tilde{x}_{i-2K-l}\tilde{x}_{i-K-j} + \tilde{x}_{i-K+j-l}\tilde{x}_{i+K+j} - \tilde{x}_i + F, \qquad (15)$$

where $\tilde{x}_{-i} = \tilde{x}_{n-i}$, and we used $K = 32$ and $F = 10$. By solving (15) on a regular grid of size $n = 960$ on a circle with unit circumference using a 4-th order Runge-Kutta scheme, with five internal steps of size $dt = 0.005$, and setting $x_i = b\tilde{x}_i$ with $b = 0.2$, we obtained the evolution operator $\mathcal{E}_t$. We also calculated an analytic expression for its derivative $\nabla\mathcal{E}_t$, which is necessary for Algorithm 5.

To complete our state-space model (12)-(13), we assumed $\mathbf{Q}_{t;i,j} = 0.2\exp(-\|\mathbf{s}_i - \mathbf{s}_j\|/0.15)$, we randomly selected $n_t = n/10 = 96$ observation indices at each t, and we took the initial moments $\boldsymbol{\mu}_{0|0}$ and $\boldsymbol{\Sigma}_{0|0}$ to be the corresponding sample moments from a long simulation from (15). We simulated 40 datasets from the state-space model, each at $T = 20$ time steps, and for each of the four exponential-family likelihoods, using $\tau^2 = 0.2$ in the Gaussian case.

For each data set obtained in this way, we applied the three filtering methods described in Section 6.1. We used $N = 39$, and for the EKVL filter (Algorithm 5) we set $J = 2$,

$M = 7$, and $|\mathcal{X}_{j_1,\ldots,j_m}|$ equal to 5, 5, 5, 5, 6, 6, 6, respectively, for $m = 0, 1, \ldots, M-1$, and $|\mathcal{X}_{j_1,\ldots,j_M}| \leq 1$. The average scores over the 40 simulations are shown in Figure 5. Our method (HV) compared favorably to the low-rank filter and provided excellent approximation accuracy as evidenced by very low RRMSPE and dLS scores.

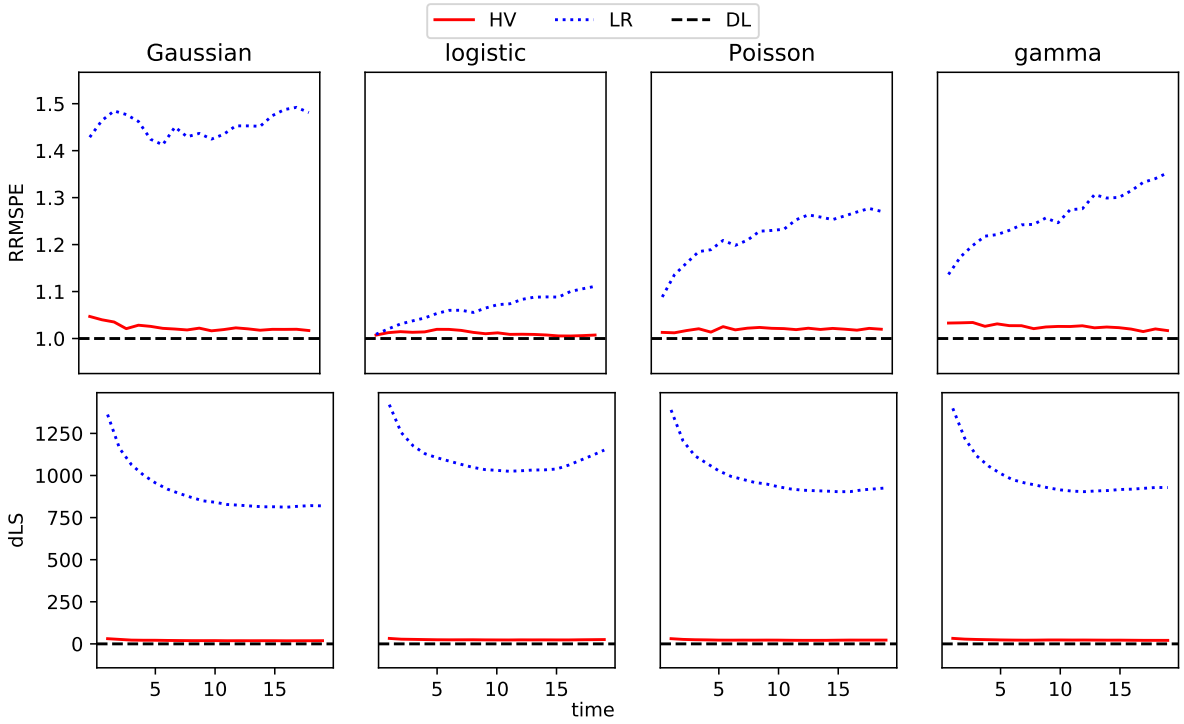


Figure 5: Accuracy of filtering distribution $\mathbf{x}_t|\mathbf{y}_{1:t}$ for the Lorenz model in Section 6.5

7 Conclusions

We specified the relationship between ordered conditional independence and sparse (inverse) Cholesky factors. Next, we described a hierarchical Vecchia approximation and showed that it exhibits equivalent sparsity in the Cholesky factors of both its precision and covariance matrices. Due to this remarkable property, the approximation is suitable for high-dimensional spatio-temporal filtering. The hierarchical Vecchia approximation can be computed using a simple and fast incomplete Cholesky decomposition. Further, by combining the approach with a Laplace approximation and the extended Kalman filter, we obtained scalable filters for non-Gaussian and non-linear spatio-temporal state-space models.

Our methods can also be directly applied to spatio-temporal point patterns modeled using log-Gaussian Cox processes, which can be viewed as Poisson data after discretization of the spatial domain, resulting in accurate Vecchia-Laplace-type approximations (Zilber and Katzfuss, 2019). We plan on investigating an extension of our methods to retrospective smoothing over a fixed time period. Another interesting extension would be to combine our methodology with the unscented Kalman filter (Julier and Uhlmann, 1997) for strongly nonlinear evolution. Finally, while we focused our attention on spatio-temporal data, our

work can be extended to other applications, as long as a sensible hierarchical partitioning of the state vector can be obtained as in Section 3.1.

Code implementing our methods and reproducing our numerical experiments is available at `https://github.com/katzfuss-group/vecchiaFilter`.

Acknowledgments

The authors were partially supported by National Science Foundation (NSF) Grant DMS–1654083. Katzfuss' research was also partially supported by NSF Grant DMS–1953005. We would like to thank Yang Ni, Florian Schäfer, and Mohsen Pourahmadi for helpful comments and discussions. We are also grateful to Edward Ott and Seung-Jong Baek for code, and Phil Taffet for advice for our implementation of the Lorenz model.

A Glossary of graph theory terms

We briefly review here some graph terminology necessary for our exposition and proofs, following Lauritzen (1996).

If $a \to b$ then we say that a is a *parent* of b and, conversely, that b is a *child* of a. Moreover, if there is a sequence of distinct vertices $h_1, \ldots, h_k$ such that $h_i \to h_{i+1}$ or $h_i \leftarrow h_{i+1}$ for all $i < k$, then we say that $h_1, \ldots, h_k$ is a *path*. If all arrows point to the right, we say that h_k is a *descendant* of h_i for $i < k$, while each h_i is an *ancestor* of h_k.

A *moral graph* is an undirected graph obtained from a DAG by first finding the pairs of parents of a common child that are not connected, adding an edge between them, and then by removing the directionality of all edges. If no edges need to be added to a DAG to make it moral, we call it a *perfect graph*.

Let $G = (V, E)$ be a directed graph with vertices V and edges E. If V_1 is a subset of V, then the *ancestral set* of V_1, denoted $\mathrm{An}(V_1)$, is the smallest subset of V that contains V_1 and such that for each $v \in \mathrm{An}(V_1)$ all ancestors of v are also in $\mathrm{An}(V_1)$.

Finally, consider three disjoint sets of vertices A, B, C in an undirected graph. We say that C *separates* A and B if, for every pair of vertices $a \in A$ and $b \in B$, every path connecting a and b passes through C.

B Proofs

Proof of Claim 1.

1. This proof is based on Schäfer et al. (2017, Sect. 3.2). Split $\mathbf{w} = (w_1, \ldots, w_n)^\top$ into two vectors, $\mathbf{u} = \mathbf{w}_{1:j-1}$ and $\mathbf{v} = \mathbf{w}_{j:n}$. Then,

$$\begin{aligned}
\mathbf{L} &= \mathbf{K}^{\frac{1}{2}} = \mathrm{chol}\begin{pmatrix} \mathbf{K}_{uu} & \mathbf{K}_{uv} \\ \mathbf{K}_{vu} & \mathbf{K}_{vv} \end{pmatrix} \\
&= \mathrm{chol}\left(\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{K}_{uu} & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_{vv} - \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{uv} \end{pmatrix}\begin{pmatrix} \mathbf{I} & \mathbf{K}_{uu}^{-1}\mathbf{K}_{uv} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}\right) \\
&= \begin{pmatrix} \mathbf{K}_{uu}^{\frac{1}{2}} & \mathbf{0} \\ \mathbf{K}_{vu}\mathbf{K}_{uu}^{-\frac{1}{2}} & \left(\mathbf{K}_{vv} - \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{vu}\right)^{\frac{1}{2}} \end{pmatrix}.
\end{aligned}$$

   Note that $\mathbf{L}_{i,j}$ is the $(i-j+1)$-th element in the first column of $\left(\mathbf{K}_{vv} - \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{vu}\right)^{\frac{1}{2}}$, which is the Cholesky factor of $\mathrm{Var}(\mathbf{v}|\mathbf{u}) = \mathbf{K}_{vv} - \mathbf{K}_{vu}\mathbf{K}_{uu}^{-1}\mathbf{K}_{vu}$. Careful examination of the Cholesky factorization of a generic matrix $\mathbf{A}$, which is described in Algorithm 1 when setting $s_{i,j} = 0$ for all (i, j), shows that the computations applied to the first column of $\mathbf{A}$ are fairly simple. In particular, this implies that

$$\mathbf{L}_{i,j} = \left(\mathrm{chol}(\mathrm{Var}(\mathbf{v}|\mathbf{u}))\right)_{i-j+1,1} = \frac{\mathrm{Cov}(w_i, w_j|\mathbf{w}_{1:j-1})}{\sqrt{\mathrm{Var}(w_j|\mathbf{w}_{1:j-1})}},$$

because $w_j = \mathbf{v}_1$ and $w_i = \mathbf{v}_{i-j+1}$. Thus, $\mathbf{L}_{i,j} = 0 \iff \mathrm{Cov}(w_i, w_j|\mathbf{w}_{1:j-1}) = 0 \iff w_i \perp w_j \,|\, \mathbf{w}_{1:j-1}$ because $\mathbf{w}$ was assumed to be jointly normal.

2. Thm. 12.5 in Rue and Held (2010) implies that for a Cholesky factor $\breve{\mathbf{U}}$ of a precision matrix $\mathbf{P}\mathbf{K}^{-1}\mathbf{P}$ of a normal random vector $\breve{\mathbf{w}} = \mathbf{P}\mathbf{w}$, we have $\breve{\mathbf{U}}_{i,j} = 0 \iff \breve{w}_i \perp \breve{w}_j \,|\, \{\breve{\mathbf{w}}_{j+1:i-1}, \breve{\mathbf{w}}_{i+1:n}\}$. Equivalently, because $\mathbf{U} = \mathbf{P}\breve{\mathbf{U}}\mathbf{P}$, we conclude that $\mathbf{U}_{j,i} = 0 \iff w_i \perp w_j \,|\, \{\mathbf{w}_{1:j-1}, \mathbf{w}_{j+1:i}\}$.

□

Proof of Proposition 1. The fact that $\hat{p}(\mathbf{x})$ is jointly normal holds for any Vecchia approximation (e.g., Datta et al., 2016; Katzfuss and Guinness, 2019, Prop. 1).

1. First, note that $\mathbf{L}$ and $\mathbf{U}$ are lower- and upper-triangular matrices, respectively. Hence, we assume in the following that $j < i$ but $x_j \notin \mathcal{C}_i$, and then show the appropriate conditional-independence results.

   (a) By Claim 1, we only need to show that $x_i \perp x_j|\mathbf{x}_{1:j-1}$. Let G be the graph corresponding to factorization (3) and denote by $G^m_{\mathrm{An}(A)}$ the moral graph of the ancestral set of A. By Corollary 3.23 in Lauritzen (1996), it is enough to show that $\{x_1, \ldots, x_{j-1}\}$ separates x_i and x_j in $G^m_{An(\{x_1,\ldots,x_j,x_i\})}$. In the rest of the proof we label each vertex by its index, to simplify notation.

   We make three observations which can be easily verified. [1] $\mathrm{An}(\{1, \ldots, j, i\}) \subset \{1, \ldots, i\}$; [2] Given the ordering of variables described in Section 3.1, if $k \to l$ then $k < l$; [3] G is a perfect graph, so $G^m_{An(\{1,\ldots,j,i\})}$ is a subgraph of G after all edges are turned into undirected ones.

   We now prove Proposition 1.1.a by contradiction. Assume that $\{x_1, \ldots, x_{j-1}\}$ does not separate x_i and x_j, which means that there exists a path $(h_1, \ldots, h_k)$ in $\{x_1, \ldots, x_i\}$ connecting x_i and x_j such that $h_k \in \mathrm{An}(\{1, \ldots, j, i\})$ and $j + 1 \le h_k \le i - 1$.

   There are four cases we need to consider and we show that each one of them leads to a contradiction. First, assume that the last edge in the path is $h_k \to j$. This violates observation [2]. Second, assume that the first edge is $i \leftarrow h_1$. But because of [1] we know that $h_1 < i$, and by [2] we get a contradiction again. Third, let the path be of the form $i \to h_1 \leftarrow \cdots \leftarrow h_k \leftarrow j$ (i.e., all edges are of the form $h_r \leftarrow h_{r+1}$). However, this would mean that $\mathcal{X}_{j_1,\ldots,j_\ell} \subset \mathcal{A}_{i_1,\ldots,i_m}$, for $x_i \in \mathcal{X}_{i_1,\ldots,i_m}$ and $x_j \in \mathcal{X}_{j_1,\ldots,j_\ell}$. This implies that $j \in \mathcal{C}_i$, which in turn contradicts the assumption of the proposition. Finally, the only possibility we have not excluded yet is a path such that $i \leftarrow h_1 \ldots h_k \leftarrow j$ with some edges of the form $h_r \to h_{r+1}$. Consider the largest r for which this is true. Then by [3] there has to exist an edge $h_r \leftarrow h_p$ where $h_p \in \{h_{r+2}, \ldots, h_k, j\}$. But this means that j is an ancestor of h_r so the path can be reduced to $i \to h_1, \ldots h_r \to j$. We continue in this way for each edge "$\leftarrow$" which reduces this path to case 3 and leads to a contradiction.

   Thus we showed that all paths in $G^m_{An(\{1,\ldots,j,i\})}$ connecting i and j necessarily have be contained in $\{1, \ldots, j-1\}$, which proves Proposition 1.1.a.

   (b) Like in part (a), we note that by Claim 1 it it enough to show that $x_i \perp x_j|\mathbf{x}_{1:j-1,j+1:i-1}$. Therefore, proceeding in a way similar to the previous case, we need to show that $\{1, \ldots, j-1, j+1, \ldots, i\}$ separates i and j in $G^m_{An(\{1,\ldots,i\})}$. However, notice that it can be easily verified that $\mathrm{An}(\{1, \ldots, i\}) \subset \{1, \ldots, i\})$, which means that $\mathrm{An}(\{1, \ldots, i\}) = \{1, \ldots, i\})$. Moreover, observe that the subgraph of G generated by $\mathrm{An}(\{1, \ldots, i\})$ is already moral, which means that if two vertices did not have a connecting edge in the original DAG, they also do not share an edge in $G^m_{1,\ldots,i}$. Thus i and j are separated by $\{1, \ldots, j-1, j+1, \ldots, i-1\}$ in $G^m_{1,\ldots,i}$, which by Corollary 3.23 in (Lauritzen, 1996) proves part (b).

2. Let $\mathbf{P}$ be the reverse-ordering permutation matrix. Let $\mathbf{B} = \mathrm{chol}(\mathbf{P}\hat{\mathbf{\Sigma}}^{-1}\mathbf{P})$. Then $\mathbf{U} = \mathbf{P}\mathbf{B}\mathbf{P}$. By the definition of $\mathbf{B}$, we know that $\mathbf{B}\mathbf{B}^\top = \mathbf{P}\hat{\mathbf{\Sigma}}^{-1}\mathbf{P}$, and consequently $\mathbf{P}\mathbf{B}\mathbf{B}^\top\mathbf{P} = \hat{\mathbf{\Sigma}}^{-1}$. Therefore, $\hat{\mathbf{\Sigma}} = (\mathbf{P}\mathbf{B}\mathbf{B}'\mathbf{P})^{-1}$. However, we have $\mathbf{P}\mathbf{P} = \mathbf{I}$ and $\mathbf{P} = \mathbf{P}^\top$, and hence $\hat{\mathbf{\Sigma}} = \big((\mathbf{P}\mathbf{B}\mathbf{P})(\mathbf{P}\mathbf{B}^\top\mathbf{P})\big)^{-1}$. So we conclude that $\hat{\mathbf{\Sigma}} = (\mathbf{U}\mathbf{U}^\top)^{-1} = (\mathbf{U}^\top)^{-1}\mathbf{U}^{-1} = (\mathbf{U}^{-1})^\top\mathbf{U}^{-1}$ and $(\mathbf{U}^{-1})^\top = \mathbf{L}$, or alternatively $\mathbf{L}^{-\top} = \mathbf{U}$.

□

Proof of Proposition 2.

1. We observe that hierarchical Vecchia satisfies the sparse general Vecchia requirement specified in Katzfuss and Guinness (2019, Sect. 4), because the nested ancestor sets imply that $\mathcal{C}_j \subset \mathcal{C}_i$ for all $j < i$ with $i, j \in \mathcal{C}_k$. Hence, reasoning presented in Katzfuss and Guinness (2019, proof of Prop. 6) allow us to conclude that $\widetilde{\mathbf{U}}_{j,i} = 0$ if $\mathbf{U}_{j,i} = 0$.

2. As the observations in $\mathbf{y}$ are conditionally independent given $\mathbf{x}$, we have

$$\hat{p}(\mathbf{x}|\mathbf{y}) \propto \hat{p}(\mathbf{x})p(\mathbf{y}|\mathbf{x}) = \big(\textstyle\prod_{i=1}^{n} p(x_i|\mathcal{C}_i)\big)\big(\prod_{i\in\mathcal{I}} p(y_i|x_i)\big), \tag{16}$$

   Let G be the graph representing factorization (3), and let $\tilde{G}$ be the DAG corresponding to (16). We order vertices in $\tilde{G}$ such that vertices corresponding to $\mathbf{y}$ have numbers $n+1, n+2, \ldots, n+|\mathcal{I}|$. For easier notation we also define $\tilde{\mathcal{I}} = \{i+n : i \in \mathcal{I}\}$. Similar to the proof of Proposition 1.1, we suppress the names of variables and use the numbers of vertices instead (i.e., we refer to the vertex x_k as k and y_j as $n+j$). Using this notation, and following the proof of Proposition 1, it is enough to show that $\{1, \ldots, j-1\} \cup \tilde{\mathcal{I}}$ separate i and j in $\tilde{\mathcal{G}}^m$, where $\tilde{\mathcal{G}} := G_{\mathrm{An}(\{1,\ldots,j,i\}\cup\tilde{\mathcal{I}})}$.

   We first show that $1, \ldots, j-1$ separate i and j in G. Assume the opposite, that there exists a path $(h_1, \ldots, h_k)$ in $\{j+1, \ldots, i-1, i+1, \ldots, n\}$ connecting x_i and x_j. Let us start with two observations. First, note that the last arrow has to go toward j (i.e., $h_k \to j$), because $h_k > j$. Second, let $p_0 = \max\{p < k : h_p \to h_{p+1}\}$, the index of the last vertex with an arrow pointing toward j that is not h_k. If p_0 exists, then $(h_1, \ldots, h_{p_0})$ is also a path connecting i and j. This is because h_{p_0} and j are parents of h_{p_0+1}, and so $h_{p_0} \to j$, because G is perfect and $h_p > j$.

   Now notice that a path (h_1) (i.e., one consisting of a single vertex) cannot exist, because we would either have $i \to h_1 \to j$ or $i \leftarrow h_1 \leftarrow j$. The first case implies that $i \to j$, because in G a node is a direct parent of all its descendants. Similarly in the second case, because G is perfect and $j < i$, we also have that $i \leftarrow j$. In either case the assumption $j \notin \mathcal{C}_i$ is violated.

   Now consider the general case of a path $(h_1, \ldots, h_k)$ and recall that by observation 1 also $h_k \leftarrow j$. But then the path (h_1) also exists because by 3 all descendants are also direct children of their ancestors. As shown in the previous paragraph, we thus have a contradiction.

   Finally, consider the remaining case such that $p_0 = \max\{p : h_p \to h_{p+1}\}$ exists. But then $(h_1, \ldots, h_{p_0})$ is also a path connecting i and j. If all arrows in this reduced paths are to the left, we already showed it produces a contradiction. If not, we can again find $\max\{p : h_p \to h_{p+1}\}$ and continue the reduction until all arrows are in the same direction, which leads to a contradiction again.

   Thus we show that i and j are separated by $\{1, \ldots, j-1\}$ in G. This implies that they are separated by this set in every subgraph of G that contains vertices $\{1, \ldots, j, i\}$ and in particular in $\mathcal{G} := G_{An(\{1,\ldots,j-1\}\cup\{j\}\cup\{i\}\cup\tilde{\mathcal{I}})}$. Recall that we showed in Proposition 1 that G is perfect, which means that $\mathcal{G} = \mathcal{G}^m$.

   Next, for a directed graph $\mathcal{F} = (V, E)$ define the operation of *adding a child* as extending $\mathcal{F}$ to $\bar{\mathcal{F}} = (V \cup \{w\}, E \cup \{v \to w\})$ where $v \in V$. In other words, we add one vertex and one edge such that one of the old vertices is a parent and the new vertex is a child. Note that a perfect graph with an added child is still perfect. Moreover, because the new vertex is connected to only a single existing one, adding a child does not create any new connections between the old vertices. It follows that if C separates A and B in $\mathcal{F}$, then $C \cup \{w\}$ does so in $\bar{\mathcal{F}}$ as well.

   Finally, notice that $\tilde{\mathcal{G}}$, the graph we are ultimately interested in, can be obtained from $\mathcal{G}$ using a series of child additions. Because these operations preserve separation even after adding the child to the separating set, we conclude that i and j are separated by $\{1, \ldots, j-1\} \cup \tilde{\mathcal{I}}$ in $\tilde{\mathcal{G}}$. Moreover, because $\mathcal{G}$ was perfect and because graph perfection is preserved under child addition, we have that $\tilde{\mathcal{G}} = \tilde{\mathcal{G}}^m$.

□

CLAIM 2. *Assuming the joint distribution $\hat{p}(\mathbf{x})$ as in (4), we have $\hat{p}(x_i, x_j) = p(x_i, x_j)$ if $x_j \in \mathcal{C}_i$; that is, the marginal bivariate distribution of a pair of variables is exact if one of the variables is in the conditioning set of the other.*

Proof. First, consider the case where $x_i, x_j \in \mathcal{X}_{j_1,\ldots,j_m}$. Then note that $\hat{p}(\mathcal{X}_{j_1,\ldots,j_m}) = \int \hat{p}(\mathbf{x}) d\mathbf{x}_{-\mathcal{X}_{j_1,\ldots,j_m}}$. Furthermore, notice that given the decomposition (3) and combining appropriate terms we can write

$$\hat{p}(\mathbf{x}) = p\left(\mathcal{X}_{j_1,\ldots,j_m} | \mathcal{A}_{j_1,\ldots,j_m}\right) p\left(\mathcal{A}_{j_1,\ldots,j_m}\right) p\left(\mathbf{x}_{-\{\mathcal{X}_{j_1,\ldots,j_m} \cup \mathcal{A}_{j_1,\ldots,j_m}\}}\right).$$

Using these two observations, we conclude that

$$\hat{p}(\mathcal{X}_{j_1,\ldots,j_m}) = \int \hat{p}(\mathbf{x}) d\mathbf{x}_{-\mathcal{X}_{j_1,\ldots,j_m}} = \int \prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_m} | \mathcal{A}_{j_1,\ldots,j_m}) d(\mathcal{X}, \mathcal{X}_{j_1}, \ldots, \mathcal{X}_{j_1,\ldots,j_{m-1}}) = p(\mathcal{X}_{j_1,\ldots,j_m}),$$

which proves that $\hat{p}(x_i, x_j) = p(x_i, x_j)$ if $x_i, x_j \in \mathcal{X}_{j_1,\ldots,j_m}$.

Now let $x_i \in \mathcal{X}_{j_1,\ldots,j_m}$ and $x_j \in \mathcal{X}_{j_1,\ldots,j_\ell}$ with $\ell < m$, because $x_j \in \mathcal{C}_i$ implies that $j < i$. Then,

$$\begin{aligned}
\hat{p}(\mathcal{X}_{j_1,\ldots,j_m}, \mathcal{X}_{j_1,\ldots,j_\ell}) &= \int \hat{p}(\mathbf{x}) d\mathbf{x}_{-\{\mathcal{X}_{j_1,\ldots,j_m} \cup \mathcal{X}_{j_1,\ldots,j_\ell}\}} = \\
&= \int \prod_{k=0}^{M} \prod_{j_1,\ldots,j_k} p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k}) d\mathbf{x}_{-\{\mathcal{X}_{j_1,\ldots,j_m} \cup \mathcal{X}_{j_1,\ldots,j_\ell}\}} \\
&= \int \prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k}) d(A_{j_1,\ldots,j_\ell} \cup \mathcal{X}_{j_1,\ldots,j_{\ell+1}} \cup \cdots \cup \mathcal{X}_{j_1,\ldots,j_{m-1}}).
\end{aligned}$$

The second equality uses (3), the definition of $\hat{p}$; the last equation is obtained by integrating out $\mathcal{X}^{0:M} \setminus \bigcup_{k=0}^{m} \mathcal{X}_{j_1,\ldots,j_k}$. Note that $\mathcal{A}_{j_1,\ldots,j_m} = \mathcal{A}_{j_1,\ldots,j_\ell} \cup \bigcup_{k=\ell}^{m-1} \mathcal{X}_{j_1,\ldots,j_k}$. Therefore, by Bayes law, for any $k > \ell$:

$$\begin{aligned}
p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k}) &= p\left(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_\ell} \cup (\mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})\right) \\
&= \frac{p\left(\mathcal{A}_{j_1,\ldots,j_\ell} | \mathcal{X}_{j_1,\ldots,j_k} \cup (\mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})\right) p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})}{p(\mathcal{A}_{j_1,\ldots,j_\ell} | \mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})} \\
&= \frac{p(\mathcal{A}_{j_1,\ldots,j_\ell} | \mathcal{A}_{j_1,\ldots,j_{k+1}} \setminus \mathcal{A}_{j_1,\ldots,j_\ell}) p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})}{p(\mathcal{A}_{j_1,\ldots,j_\ell} | \mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell})} = (*)
\end{aligned}$$

The last equality holds because $\mathcal{X}_{j_1,\ldots,j_m} \cup \mathcal{A}_{j_1,\ldots,j_m} = \mathcal{A}_{j_1,\ldots,j_{m+1}}$. As a consequence

$$\begin{aligned}
\prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k}) &= \prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_k} | \mathcal{A}_{j_1,\ldots,j_k} \setminus \mathcal{A}_{j_1,\ldots,j_\ell}) p(\mathcal{A}_{j_1,\ldots,j_\ell}) \\
&= \prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_k} | \bigcup_{s=k-1}^{\ell} \mathcal{X}_{j_1,\ldots,j_s}) p(\mathcal{A}_{j_1,\ldots,j_\ell})
\end{aligned}$$

and

$$\begin{aligned}
(*) &= \int \prod_{k=0}^{m} p(\mathcal{X}_{j_1,\ldots,j_m} | \mathcal{A}_{j_1,\ldots,j_m}) p(\mathcal{A}_{j_1,\ldots,j_\ell}) d(\mathcal{A}_{j_1,\ldots,j_\ell} \cup \mathcal{X}_{j_1,\ldots,j_{\ell+1}} \cup \ldots, \cup \mathcal{X}_{j_1,\ldots,j_{m-1}}) \\
&= \int p(\mathcal{X}_{j_1,\ldots,j_m}, \mathcal{X}_{j_1,\ldots,j_{m-1}}, \ldots, \mathcal{X}_{j_1,\ldots,j_\ell}, \mathcal{A}_{j_1,\ldots,j_\ell}) d(\mathcal{A}_{j_1,\ldots,j_\ell} \cup \mathcal{X}_{j_1,\ldots,j_{\ell+1}} \cup \ldots, \cup \mathcal{X}_{j_1,\ldots,j_{m-1}}) \\
&= p(\mathcal{X}_{j_1,\ldots,j_m}, \mathcal{X}_{j_1,\ldots,j_\ell})
\end{aligned}$$

This means that $\hat{p}(\mathcal{X}_{j_1,\ldots,j_m}, \mathcal{X}_{j_1,\ldots,j_\ell}) = p(\mathcal{X}_{j_1,\ldots,j_m}, \mathcal{X}_{j_1,\ldots,j_\ell})$, or that the marginal distribution of $\mathcal{X}_{j_1,\ldots,j_m}$ and $\mathcal{X}_{j_1,\ldots,j_\ell}$ in (3) is the same as in the true distribution p. Because p is Gaussian, it follows that $\hat{p}(x_i, x_j) = p(x_i, x_j)$. This ends the proof.

□

Proof of Proposition 3. We use $l^{\text{inc}}_{i,j}$, $l_{i,j}$, $\sigma_{i,j}$, $\hat{\sigma}_{i,j}$ to denote the (i,j)-th elements of $\mathbf{L}^{\text{inc}} = \text{ichol}(\boldsymbol{\Sigma}, \mathbf{S})$, $\mathbf{L} = \text{chol}(\hat{\boldsymbol{\Sigma}})$, $\boldsymbol{\Sigma}$, $\hat{\boldsymbol{\Sigma}}$, respectively. It can be seen easily in Algorithm 1 that $\text{chol}(\boldsymbol{\Sigma}) = \text{ichol}(\boldsymbol{\Sigma}, \mathbf{S^1})$, where $\mathbf{S^1}_{i,j} = 1$ for $i \geq j$ and 0 otherwise.

We prove that $l^{\text{inc}}_{i,j} = l_{i,j}$ by induction over the elements of the Cholesky factor, following the order in which they are computed. First, we observe that $l^{\text{inc}}_{1,1} = l_{1,1}$. Next, consider the computation of the (i,j)-th entry, assuming that we have $l^{\text{inc}}_{k,q} = l_{k,q}$ for all previously computed entries. According to Algorithm 1, we have

$$l_{i,j} = \tfrac{1}{l_{j,j}}\big(\hat{\sigma}_{i,j} - \textstyle\sum_{k=1}^{j-1} l_{i,k} l_{j,k}\big), \qquad l^{inc}_{i,j} = \tfrac{s_{i,j}}{l_{j,j}}\big(\sigma_{i,j} - \textstyle\sum_{k=1}^{j-1} l_{i,k} l_{j,k}\big).$$

Now, if $s_{i,j} = 1 \iff x_j \in \mathcal{C}_i$, then Claim 2 tells us that $\sigma_{i,j} = \hat{\sigma}_{i,j}$, and hence $l_{i,j} = l^{\text{inc}}_{i,j}$. If $s_{i,j} = 0 \iff x_j \notin \mathcal{C}_i$, then $l^{\text{inc}}_{i,j} = 0$, and also $l_{i,j} = 0$ by Proposition 1.1(a). This completes the proof via induction. □

Claim 3. *Let* $\mathbf{x}$ *have density* $\hat{p}(\mathbf{x})$ *as in* (3)*, and let each conditioning set* $\mathcal{C}_i$ *have size at most* N*. Then* $\mathbf{\Lambda}$*, the precision matrix of* $\mathbf{x}$*, has* $\mathcal{O}(nN)$ *nonzero elements. Moreover, the columns of* $\mathbf{U} = \text{rchol}(\mathbf{\Lambda})$ *and the rows of* $\mathbf{L} = \text{chol}(\mathbf{\Lambda}^{-1})$ *each have at most* N *nonzero elements.*

Proof. Because the precision matrix is symmetric, it is enough to show that there are only $\mathcal{O}(nN)$ nonzero elements $\mathbf{\Lambda}_{i,j}$ in the upper triangle (i.e., with $i < j$). Let $x_i \notin \mathcal{C}_j$. This means that $x_i \not\to x_j$ and because by (3) all edges go from a lower index to a higher index, x_i and x_j are not connected. Moreover because $\mathcal{A}_{j_1,\dots,j_m} \supset \mathcal{A}_{j_1,\dots,j_{m-1}}$ there is also no x_k with $k > \max(i,j)$ such that $x_i \to x_k$ and $x_j \to x_k$. Thus, using Proposition 3.2 in Katzfuss and Guinness (2019) we conclude that $\mathbf{\Lambda}_{i,j} = 0$ for $x_i \notin \mathcal{C}_j$. This means that each row i has at most $|\mathcal{C}_i| \leq N$ nonzero elements, and the entire lower triangle has $\mathcal{O}(nN)$ nonzero values.

Proposition 1 implies that the i-th column of $\mathbf{U}$ has at most as many nonzero entries as there elements in the conditioning set $\mathcal{C}_i$, which we assumed to be of size at most N. Similarly, the i-th row of $\mathbf{L}$ has at most as many nonzero entries as the number of elements in the conditioning set $\mathcal{C}_i$. □

Claim 4. *Let* $\mathbf{A}$ *be an* $n \times n$ *lower triangular matrix with at most* $N < n$ *nonzero elements in each row at known locations. Letting* a_{ij} *and* $\tilde{a}_{ij}$ *be the* (i,j)*-th element of* $\mathbf{A}$ *and* $\mathbf{A}^{-1}$*, respectively, assume that* $\tilde{a}_{i,j} = 0$ *if* $a_{i,j} = 0$*. Then, the cost of calculating* $\mathbf{A}^{-1}$ *from* $\mathbf{A}$ *is* $\mathcal{O}(nN^2)$*.*

Proof. Notice that calculating $\tilde{\mathbf{a}}_k$, the kth column of $\mathbf{A}^{-1}$, is equivalent to solving a linear system of the form

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} \tilde{a}_{1k} \\ \tilde{a}_{2k} \\ \tilde{a}_{3k} \\ \vdots \\ \tilde{a}_{nk} \end{bmatrix} = e_k,$$

where $e_{ik} = 1$ if $k = i$ and 0 otherwise. Using forward substitution, the i-th element of $\tilde{\mathbf{a}}_k$ can be calculated as $\tilde{a}_{ik} = \frac{1}{a_{kk}} \left(e_{ik} - \sum_{j=1}^{i-1} a_{ij} \tilde{a}_{jk} \right)$. This requires $\mathcal{O}(N)$ time, because our assumptions imply that there are at most N nonzero terms under the summation. Moreover, we also assumed that $\tilde{\mathbf{a}}_k$ has at most N nonzero elements at known locations, and so we only need to calculate those. Thus computing $\tilde{\mathbf{a}}_k$ has $\mathcal{O}(N^2)$ time complexity. As there are n columns to calculate, this ends the proof. □

Proof of Proposition 4. Starting with the ichol() procedure in Line 1, obtaining each nonzero element $l_{i,j}$ requires calculating the outer product of previously computed segments of rows i and j. Because Claim 3 implies that each row of $\mathbf{L}$ has at most N nonzero elements, obtaining $l_{i,j}$ is $\mathcal{O}(N)$. Claim 3 also shows that for each i, there are at most N nonzero elements $s_{i,j} = 1$, which implies that each row of the incomplete Cholesky factor can be calculated in $\mathcal{O}(N^2)$. Finally, because the matrix to be decomposed has n rows, the overall cost of the algorithm is $\mathcal{O}(nN^2)$. In Line 2, because $\mathbf{L}^{-1} = \mathbf{U}^\top$, Proposition 1 tells us exactly which elements of $\mathbf{L}^{-1}$ need to be calculated (i.e., are non-zero), and that there are only N of them (Claim 3). Using Claim 4, this means that computing $\mathbf{L}^{-1}$ can be accomplished in $\mathcal{O}(nN^2)$ time. Analogous reasoning and Proposition 2 allow us to conclude that computing $\widetilde{\mathbf{L}}$ in Line 5 has the same complexity. The cost of Line 3 is dominated by taking the outer product of $\mathbf{U}$, because $\mathbf{H}$ and $\mathbf{R}$ are assumed to have only one non-zero element in each row. However, $\mathbf{U}\mathbf{U}^\top$ is by definition equal to the precision matrix of $\mathbf{x}$ under (3). Therefore, by Claim 3 there are at most $\mathcal{O}(nN)$ elements to calculate and each requires multiplication of two rows with at most N nonzero elements. This means that this step can be accomplished in $\mathcal{O}(nN^2)$ time. The most expensive operation in Line 4 is taking the Cholesky factor. However, its cost proportional to the square of the number of nonzero elements in each column (e.g., Toledo, 2007, Thm. 2.2), which by Claim 3 we know to be N. As there are n columns, this step requires $\mathcal{O}(nN^2)$ time. Finally, the most expensive operation in Line 6 is the multiplication of a vector by matrix $\widetilde{\mathbf{L}}$. By Proposition 2, $\widetilde{\mathbf{L}}$ has the same number of nonzero elements per row as $\mathbf{L}$, which is at most N by Claim 3. Thus, multiplication of $\widetilde{\mathbf{L}}$ and any dense vector can be performed in $\mathcal{O}(nN)$ time. To conclude, each line of Algorithm 2 can be computed in at most $\mathcal{O}(nN^2)$ time, and so the total time complexity of the algorithm is also $\mathcal{O}(nN^2)$.

Regarding memory complexity, notice that by Claims 3 and 1, matrices $\mathbf{L}$, $\mathbf{U}$, $\widetilde{\mathbf{L}}$, $\widetilde{\mathbf{U}}$, and $\mathbf{\Lambda}$ have $\mathcal{O}(nN)$ nonzero elements, and (1) implies that matrices $\mathbf{H}$ and $\mathbf{R}$ have at most n entries. Further, the incomplete

Cholesky decomposition in Line 1 requires only those elements of $\boldsymbol{\Sigma}$ that correspond to the nonzero elements of $\mathbf{S}$. Because $\mathbf{S}$ has at most N non-zero elements in each row by construction, each of the matrices that are decomposed can be stored using $\mathcal{O}(nN)$ memory, and so the memory requirement for Algorithm 2 is $\mathcal{O}(nN)$. □

References

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(2):252–265.

Anderson, J. L. (2001). An ensemble adjustment Kalman filter for data assimilation. *Monthly Weather Review*, 129(12):2884–2903.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(4):825–848.

Bonat, W. H. and Ribeiro Jr, P. J. (2016). Practical likelihood analysis for spatial generalized linear mixed models. *Environmetrics*, 27(2):83–89.

Burgers, G., Jan van Leeuwen, P., and Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724.

Cressie, N. (1993). *Statistics for Spatial Data, revised edition*. John Wiley & Sons, New York, NY.

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

DelSole, T. and Yang, X. (2010). State and parameter estimation in stochastic dynamical models. *Physica D: Nonlinear Phenomena*, 239(18):1781–1788.

Douc, R., Cappé, O., and Moulines, E. (2005). Comparison of resampling schemes for particle filtering. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.

Doucet, A. and Johansen, A. M. (2011). A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovskii, B., editors, *The Oxford Handbook of Nonlinear Filtering*, number December, pages 656–704. Oxford University Press.

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(C5):10143–10162.

Evensen, G. (2007). *Data Assimilation: The Ensemble Kalman Filter*. Springer.

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(8):2873–2884.

Geoga, C. J., Anitescu, M., and Stein, M. L. (2018). Scalable Gaussian Process Computations Using Hierarchical Matrices. *arXiv:1808.03215*.

Gneiting, T. and Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151.

Golub, G. H. and Van Loan, C. F. (2012). *Matrix Computations*. JHU Press, 4th edition.

Grewal, M. S. and Andrews, A. P. (1993). *Kalman Filtering: Theory and Applications*. Prentice Hall.

Guinness, J. (2018). Permutation methods for sharpening Gaussian process approximations. *Technometrics*, 60(4):415–429.

Hackbusch, W. (2015). *Hierarchical Matrices: Algorithms and Analysis*, volume 49. Springer.

Heaton, M. J., Datta, A., Finley, A. O., Furrer, R., Guinness, J., Guhaniyogi, R., Gerber, F., Gramacy, R. B., Hammerling, D., Katzfuss, M., Lindgren, F., Nychka, D. W., Sun, F., and Zammit-Mangion, A. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological, and Environmental Statistics*, 24(3):398–425.

Johannesson, G., Cressie, N., and Huang, H.-C. (2003). Dynamic multi-resolution spatial models. In Higuchi, T., Iba, Y., and Ishiguro, M., editors, *Proceedings of AIC2003: Science of Modeling*, volume 14, pages 167–174, Tokyo. Institute of Statistical Mathematics.

Julier, S. J. and Uhlmann, J. K. (1997). New extension of the Kalman filter to nonlinear systems. In *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, pages 182–193.

Jurek, M. and Katzfuss, M. (2018). Multi-resolution filters for massive spatio-temporal data. *arXiv:1810.04200*.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45.

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

Katzfuss, M. and Cressie, N. (2011). Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. *Journal of Time Series Analysis*, 32(4):430–446.

Katzfuss, M. and Gong, W. (2019). A class of multi-resolution approximations for large spatial datasets. *Statistica Sinica*, accepted.

Katzfuss, M. and Guinness, J. (2019). A general framework for Vecchia approximations of Gaussian processes. *Statistical Science*, accepted.

Katzfuss, M., Guinness, J., Gong, W., and Zilber, D. (2020a). Vecchia approximations of Gaussian-process predictions. *Journal of Agricultural, Biological, and Environmental Statistics*, accepted.

Katzfuss, M., Guinness, J., and Lawrence, E. (2020b). Scaled Vecchia approximation for fast computer-model emulation. *arXiv:2005.00386*.

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2016). Understanding the ensemble Kalman filter. *The American Statistician*, 70(4):350–357.

Katzfuss, M., Stroud, J. R., and Wikle, C. K. (2020c). Ensemble Kalman methods for high-dimensional hierarchical dynamic space-time models. *Journal of the American Statistical Association*, 115(530):866–885.

Kaufman, C. G., Bingham, D., Habib, S., Heitmann, K., and Frieman, J. A. (2011). Efficient emulators of computer experiments using compactly supported correlation functions, with an application to cosmology. *The Annals of Applied Statistics*, 5(4):2470–2492.

Kennedy, M. C. and O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B*, 63(3):425–464.

Lauritzen, S. (1996). *Graphical Models*. Oxford Statistical Science Series. Clarendon Press.

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(5):3734–3749.

Liu, H., Ong, Y.-S., Shen, X., and Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE Transactions on Neural Networks and Learning Systems*.

Lorenz, E. N. (1996). Predictability – A problem partly solved. *Seminar on Predictability, Vol. 1, ECMWF*.

Lorenz, E. N. (2005). Designing chaotic models. *Journal of the Atmospheric Sciences*, 62(5):1574–1587.

Nychka, D. W. and Anderson, J. L. (2010). Data assimilation. In Gelfand, A. E., Diggle, P. J., Fuentes, M., and Guttorp, P., editors, *Handbook of Spatial Statistics*, chapter 27, pages 477–494. CRC Press.

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(3-4):323–340.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rue, H. and Held, L. (2010). Discrete spatial variation. In *Handbook of Spatial Statistics*, chapter 12, pages 171–200. CRC Press.

Sacks, J., Welch, W., Mitchell, T., and Wynn, H. (1989). Design and analysis of computer experiments. *Statistical Science*, 4(4):409–435.

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(1):015009.

Schäfer, F., Katzfuss, M., and Owhadi, H. (2020). Sparse Cholesky factorization by Kullback-Leibler mini-

mization. *arXiv:2004.14455*.

Schäfer, F., Sullivan, T. J., and Owhadi, H. (2017). Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity. *arXiv:1706.02205*.

Sigrist, F., Künsch, H. R., and Stahel, W. A. (2015). Stochastic partial differential equation based modelling of large space–time data sets. *Journal of the Royal Statistical Society, Series B*.

Stein, M. L. (2002). The screening effect in kriging. *Annals of Statistics*, 30(1):298–323.

Stein, M. L. (2011). When does the screening effect hold? *Annals of Statistics*, 39(6):2795–2819.

Stein, M. L., Chi, Z., and Welty, L. (2004). Approximating likelihoods for large spatial data sets. *Journal of the Royal Statistical Society: Series B*, 66(2):275–296.

Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86.

Toledo, S. (2007). Lecture Notes on Combinatorial Preconditioners, Chapter 3. http://www.tau.ac.il/˜stoledo/Support/chapter-direct.pdf.

Vecchia, A. (1988). Estimation and model identification for continuous spatial processes. *Journal of the Royal Statistical Society, Series B*, 50(2):297–312.

Verlaan, M. and Heemink, A. W. (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*, pages 247–252. World Meteorological Organization.

Wikle, C. K. and Cressie, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika*, 86(4):815–829.

Zilber, D. and Katzfuss, M. (2019). Vecchia-Laplace approximations of generalized Gaussian processes for big non-Gaussian spatial data. *arXiv:1906.07828*.