Statistical Science
2021, Vol. 36, No. 1, 124–141
https://doi.org/10.1214/19-STS755

A General Framework for Vecchia Approximations of Gaussian Processes

Matthias Katzfuss and Joseph Guinness

Abstract. Gaussian processes (GPs) are commonly used as models for functions, time series, and spatial fields, but they are computationally infeasible for large datasets. Focusing on the typical setting of modeling data as a GP plus an additive noise term, we propose a generalization of the Vecchia (*J. Roy. Statist. Soc. Ser. B* **50** (1988) 297–312) approach as a framework for GP approximations. We show that our general Vecchia approach contains many popular existing GP approximations as special cases, allowing for comparisons among the different methods within a unified framework. Representing the models by directed acyclic graphs, we determine the sparsity of the matrices necessary for inference, which leads to new insights regarding the computational properties. Based on these results, we propose a novel sparse general Vecchia approximation, which ensures computational feasibility for large spatial datasets but can lead to considerable improvements in approximation accuracy over Vecchia's original approach. We provide several theoretical results and conduct numerical comparisons. We conclude with guidelines for the use of Vecchia approximations in spatial statistics.

Key words and phrases: Computational complexity, covariance approximation, directed acyclic graphs, large datasets, sparsity, spatial statistics.

1. INTRODUCTION

Gaussian processes (GPs) have become popular choices as models or prior distributions for functions, time series, and spatial fields (see, e.g., Banerjee, Carlin and Gelfand, 2015, Rasmussen and Williams, 2006, Cressie and Wikle, 2011). The defining feature of a GP is that the joint distribution of a finite number of observations is multivariate normal. However, since computing with multivariate normal distributions incurs quadratic memory and cubic time complexity in the number of observations, GP inference is infeasible when the data size is in the tens of thousands or higher, limiting the direct use of GPs for many large datasets available today.

To achieve computational feasibility, numerous approaches have been proposed in the statistics and machine-learning literatures. These include approaches leading to sparse covariance matrices (Furrer, Genton and Nychka, 2006, Kaufman, Schervish and Nychka, 2008, Du, Zhang and Mandrekar, 2009), sparse inverse covariance (i.e., precision) matrices (Rue and Held, 2005, Lindgren, Rue and Lindström, 2011, Nychka et al., 2015), and low-rank matrices (see, e.g., Higdon, 1998, Wikle and Cressie, 1999, Quiñonero-Candela and Rasmussen, 2005, Banerjee et al., 2008, Cressie and Johannesson, 2008, Katzfuss and Cressie, 2011). Several other approaches are described in Section 3. Heaton et al. (2019) review and compare many of these methods, plus several algorithmic approaches (Gramacy and Apley, 2015, Gerber et al., 2018, Guhaniyogi and Banerjee, 2018).

In this article, we extend and study Vecchia's approach (Vecchia, 1988), one of the earliest proposed GP approximations, which leads to a sparse Cholesky factor of the precision matrix. Based on some ordering of the GP observations, Vecchia's approximation replaces the high-dimensional joint distribution with a product of univariate conditional distributions, in which each conditional distribution conditions on only a small subset of previous observations in the ordering. This approximation incurs low computational and memory burden, it has been shown to be highly accurate in terms of Kullback–Leibler divergence from the true model (see, e.g., Guinness, 2018), and it is amenable to parallel computing because each term can be computed separately.

Matthias Katzfuss is Associate Professor, Department of Statistics, Texas A&M University, College Station, Texas 77843, USA (e-mail: katzfuss@gmail.com). Joseph Guinness is Associate Professor, Department of Statistics and Data Science, Cornell University, Ithaca, New York 14853, USA (e-mail: guinness@cornell.edu).

We consider the typical setting of spatial data modeled as a GP plus an additive noise or nugget component. Datta et al. (2016a) proposed to apply Vecchia's approximation to the latent GP instead of the noisy observations, but Finley et al. (2019) noted that this approach "require[d] an excessively long run time." Here, we propose a generalized version of the Vecchia approximation, which allows conditioning on both latent and observed variables. We show that our general Vecchia approach contains several popular GP approximations as special cases, allowing for comparisons among the different approaches within a unified framework. We give a formula for efficient computation of the likelihood in the presence of noise. Further, we describe how approximations within the general Vecchia framework can be represented by directed acyclic graph (DAG) models, and we use the connection to DAGs to prove results about the sparsity of the matrices appearing in the inference algorithms. The results lead to new insights regarding computational properties, including shedding light on the computational challenges with latent Vecchia noted in Finley et al. (2019). Based on these results, we propose a particular instance of the general Vecchia framework, which we call sparse general Vecchia (SGV), that provides guaranteed levels of sparsity in its matrix representation but can lead to considerable improvements in approximation accuracy over Vecchia's original approach. In addition to the theoretical results, we provide numerical studies exploring different options within the general Vecchia framework and comparing our novel SGV to existing approximations.

This article is organized as follows. In Section 2, we review Vecchia's approximation, introduce our general Vecchia framework, and detail connections to DAGs. In Section 3, we describe several existing GP approximations as special cases of the framework. In Section 4, we consider inference within the framework, including introducing the necessary matrices and studying their sparsity, and deriving the computational complexity. In Section 5, we describe our new SGV approximation and contrast it with two existing approaches. Section 6 contains additional insights on ordering and conditioning. Numerical results and comparisons can be found in Section 7. In Section 8, we conclude and provide guidelines for the use of Vecchia approximations. Appendices A–F contain further details and proofs. The methods and algorithms proposed here are implemented in the R package `GPvecchia`.

2. A GENERAL VECCHIA APPROACH

2.1 Noisy Observations of a Gaussian Process

Let $\{y(\mathbf{s}) : \mathbf{s} \in \mathcal{D}\}$, or $y(\cdot)$, be a process of interest on a continuous (i.e., nongridded) domain $\mathcal{D} \subset \mathbb{R}^d$, $d \in \mathbb{N}^+$. We assume that $y(\cdot) \sim \mathrm{GP}(0, K)$ is a zero-mean Gaussian process (GP) with covariance function $K : \mathcal{D} \times \mathcal{D} \to \mathbb{R}$. We place no restrictions on K, other than assuming that it is a positive-definite function that is known up to a vector of parameters, $\boldsymbol{\theta}$. Usually, K will be a continuous covariance function without a nugget component, which will be added in the next paragraph. In most applications, $y(\cdot)$ will not have zero mean, but estimating and subtracting the mean is typically not a computational problem, so we ignore the mean here for simplicity. Further, let $\mathcal{S}$ be a vector of vectors of locations, meaning that $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$, where $\mathcal{S}_i$ is a vector of r_i locations in $\mathcal{D}$. (Our vector and indexing notation is explained in detail in Appendix A.) Then define $\mathbf{y}_i = y(\mathcal{S}_i)$ to be the Gaussian process vector at locations $\mathcal{S}_i$, and form the vector $\mathbf{y} := (\mathbf{y}_1, \ldots, \mathbf{y}_\ell)$.

We observe $\mathbf{z}_i = \mathbf{y}_i + \boldsymbol{\varepsilon}_i$, where the noise or nugget terms $\boldsymbol{\varepsilon}_i$ are independent $\mathcal{N}_{r_i}(\mathbf{0}, \tau^2\mathbf{I})$. The noisy-observation assumption is ubiquitous in spatial statistics, GP regression, and functional data, and has been proposed for the modeling of computer experiments (Gramacy and Lee, 2012). In this work, we assume that we observe the subset $\mathbf{z}_o$ of $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$, where $o \subset (1, \ldots, \ell)$. Parameters $\boldsymbol{\theta}$ and τ^2 are assumed to be known for now; parameter inference will be discussed in Section 4.2.

2.2 Review of Vecchia's Approximation

Define $h_o(i) := o \cap (1, \ldots, i-1)$ to be the observed "history" of i with $h_o(1) = \varnothing$, allowing us to write the joint density for the observed vector $\mathbf{z}_o$ as

$$f(\mathbf{z}_o) = \prod_{i \in o} f(\mathbf{z}_i | \mathbf{z}_{h_o(i)}). \tag{1}$$

Working with or evaluating the density in (1) directly incurs $O(n_z^2)$ memory and $\mathcal{O}(n_z^3)$ computational cost, and is thus infeasible for large n_z, where n_z is the number of individual observations in $\mathbf{z}_o$.

To avoid these computational difficulties, Vecchia's approximation (Vecchia, 1988) replaces $h_o(i)$ with a subvector $g(i)$, where $g(i)$ is often chosen to contain those indices corresponding to observations nearby in distance to the ith vector of observations. We refer to $g(i)$ as the ith conditioning index vector, and to $\mathbf{z}_{g(i)}$ as the conditioning vector for $\mathbf{z}_i$. This leads to the Vecchia approximation of the joint density in (1):

$$\widehat{f}(\mathbf{z}_o) = \prod_{i \in o} f(\mathbf{z}_i | \mathbf{z}_{g(i)}). \tag{2}$$

Vecchia (1988) considered only the case of $\mathbf{z}_i$ as singletons, whereas Stein, Chi and Welty (2004) are credited with the generalization to vector $\mathbf{z}_i$. Cressie and Davidson (1998) showed that (2) implies a Markov random field model with sparse precision matrix. Stein, Chi and Welty (2004) showed that maximizing (2) corresponds to solving a set of unbiased estimating equations, and they proposed a residual maximum likelihood (REML) method for estimating covariance parameters.

2.3 The General Vecchia Framework

The standard Vecchia approach in Section 2.2 applies to the vector of observations, $\mathbf{z}_o$. We propose a general Vecchia approach, which applies Vecchia's approximation to a vector $\mathbf{x} = \mathbf{y} \cup \mathbf{z}_o$ consisting of the data $\mathbf{z}_o$ and latent variables $\mathbf{y}$:

$$\widehat{f}(\mathbf{x}) = \prod_{i=1}^{b} f(\mathbf{x}_i | \mathbf{x}_{g(i)}), \tag{3}$$

where b is the number of subvectors in $\mathbf{x}$, and $g(i) \subset h(i) = (1, \ldots, i-1)$. Here, the elements of $\mathbf{y}$ and $\mathbf{z}_o$ are interweaved within $\mathbf{x}$. Specifically, using the notation from Appendix A, the ordering in $\mathbf{x}$ is defined as $\#(\mathbf{y}_i, \mathbf{x}) < \#(\mathbf{y}_j, \mathbf{x})$ when $i < j$, and $\#(\mathbf{z}_i, \mathbf{x}) = \#(\mathbf{y}_i, \mathbf{x}) + 1$. In words, the $\mathbf{y}_i$ vectors retain their relative ordering in $\mathbf{x}$, and $\mathbf{z}_i$ is inserted directly after $\mathbf{y}_i$ when $i \in o$. Then, the general Vecchia approximation in (3) can be written as

$$\widehat{f}(\mathbf{x}) = \left(\prod_{i=1}^{\ell} f(\mathbf{y}_i | \mathbf{y}_{q_y(i)}, \mathbf{z}_{q_z(i)}) \right) \left(\prod_{i \in o} f(\mathbf{z}_i | \mathbf{y}_i) \right). \tag{4}$$

For the conditioning vector of $\mathbf{y}_i$, $j \in q_y(i)$ means that $\mathbf{y}_i$ conditions on $\mathbf{y}_j$, while $j \in q_z(i)$ means that $\mathbf{y}_i$ conditions on $\mathbf{z}_j$. It can be more accurate but also more computationally expensive to condition on $\mathbf{y}_j$ rather than on $\mathbf{z}_j$; we will explore this tradeoff in Section 5. We always pick $\mathbf{y}_i$ as the conditioning vector for $\mathbf{z}_i$, because $\mathbf{z}_i$ was defined to be conditionally independent of all other vectors given $\mathbf{y}_i$. For the same reason, there is nothing to be gained by conditioning $\mathbf{y}_i$ on both $\mathbf{y}_j$ and $\mathbf{z}_j$, and so we always take $q_y(i) \cap q_z(i) = \varnothing$. We call $q(i) = (q_y(i), q_z(i))$ the conditioning index vector. Note that if $j \notin o$, assuming $j \in q_z(i)$ is equivalent to removing j from $q(i)$.

Usually, it is of interest to evaluate an approximation to $f(\mathbf{z}_o)$, which involves integrating the approximation for the joint density of $\mathbf{z}_o$ and $\mathbf{y}$ in (3) and (4) over the latent vector $\mathbf{y}$:

$$\widehat{f}(\mathbf{z}_o) = \int \widehat{f}(\mathbf{x})\, d\mathbf{y}. \tag{5}$$

If the conditioning vectors are equal to the respective history vectors (i.e., $g(i) = h(i)$ for all i), the exact distribution $f(\mathbf{z}_o)$ in (1) is recovered. In this sense, the general Vecchia approximation converges to the truth as the conditioning vectors grow larger. However, large conditioning vectors negate the computational advantages, and thus the case of small conditioning vectors is of interest here.

In summary, a general Vecchia approximation $\widehat{f}(\mathbf{x})$ of $f(\mathbf{x})$, and the implied approximation $\widehat{f}(\mathbf{z}_o)$ of $f(\mathbf{z}_o)$, are determined by the following choices:

C1: The n_y locations $\mathcal{S}$, usually a superset of the observed locations.

C2: The partitioning of $\mathcal{S}$ into $\ell \leq n_y$ vectors of locations.

C3: The ordering of the location vectors as $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$.

C4: For each i, the conditioning index vector $q(i) \subset (1, \ldots, i-1)$ for $\mathbf{y}_i$.

C5: For each i, the partitioning of $q(i)$ into $q_y(i)$ and $q_z(i)$; that is, for each $j \in q(i)$, whether $\mathbf{y}_i$ should condition on $\mathbf{y}_j$ or $\mathbf{z}_j$.

For C1, the default choice is often to set $\mathcal{S}$ equal to the observed locations. A major focus of this article is C5, which is discussed in Section 5. In Section 6, we provide some insights into C2–C4, and in Section 7 we explore C3–C5 numerically.

2.4 Connections to Directed Acyclic Graphs

There are strong connections between the Vecchia approach and directed acyclic graphs (DAGs; cf. Datta et al., 2016a). A brief review of DAGs is provided in Appendix B. The conditional-independence structure implied by the Vecchia approximation in (3) can be well represented by a DAG. Viewing $\mathbf{x}_1, \ldots, \mathbf{x}_b$ as the vertices in the DAG, we have $\mathbf{x}_j \to \mathbf{x}_i$ if and only if $j \in g(i)$, and so $\mathbf{x}_{g(i)}$ is the vector formed by the set of all parents of $\mathbf{x}_i$. Note that, because Vecchia approximations allow conditioning only on previous variables in the ordering, we always have $\mathbf{x}_i \not\to \mathbf{x}_j$ if $i > j$. DAG representations are illustrated in Figure 1. We will use this connection between Vecchia approaches and DAGs to study the sparsity of the matrices needed for inference in Section 4.4.

3. EXISTING METHODS AS SPECIAL CASES

Many existing GP approximations fall into the framework described above. Each of these special cases corresponds to particular choices of C1–C5. We give some examples here. Most of these examples are illustrated in Figure 1.

3.1 Standard Vecchia and Extensions

Vecchia's original approximation (Vecchia, 1988) specifies singleton vectors ($r_i = 1$), ordering locations by a spatial coordinate (henceforth referred to as coord ordering), and conditioning only on observations $\mathbf{z}_i$ (as opposed to latent $\mathbf{y}_i$); that is, $q_z(i) = q(i)$, $q_y(i) = \varnothing$, and $o = (1, \ldots, \ell)$. Using (5), this results in the approximation

$$\begin{aligned}
\widehat{f}(\mathbf{z}) &= \int \prod_{i=1}^{\ell} f(\mathbf{z}_i | \mathbf{y}_i) f(\mathbf{y}_i | \mathbf{z}_{q(i)})\, d\mathbf{y} \\
&= \prod_{i=1}^{\ell} \int f(\mathbf{z}_i | \mathbf{y}_i, \mathbf{z}_{q(i)}) f(\mathbf{y}_i | \mathbf{z}_{q(i)})\, d\mathbf{y}_i \\
&= \prod_{i=1}^{\ell} f(\mathbf{z}_i | \mathbf{z}_{q(i)}),
\end{aligned}$$

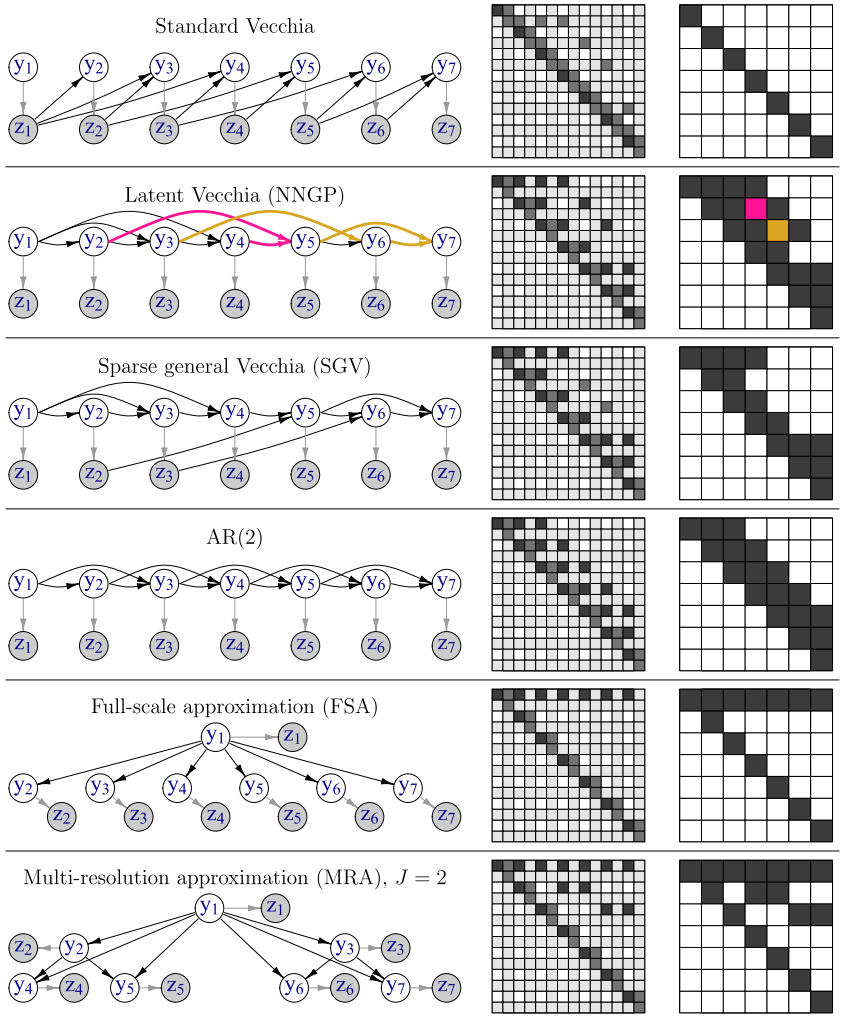


FIG. 1. *Toy examples of special cases of our general Vecchia approach (see Section 3) with* $\ell = 7$ *and* $o = (1, \ldots, 7)$*, including sparse general Vecchia (Section 5). First column: DAGs (see Section 2.4). Second column: sparsity of* $\mathbf{U}$ *(elements corresponding to* $\mathbf{z}_i$ *in gray). Third column: sparsity of* $\mathbf{V}$ *(Section 4.4). Computational complexity depends on the number of off-diagonal nonzeros in each column of* $\mathbf{U}$ *and* $\mathbf{V}$ *(Section 4.5). For all methods except latent Vecchia, these numbers are at most* $m = 2$*. For latent Vecchia, two nonzero-producing paths and the resulting nonzeros are highlighted (see Section 5).*

where $f(\mathbf{z}_i|\mathbf{y}_i) = f(\mathbf{z}_i|\mathbf{y}_i, \mathbf{z}_{q(i)})$ because $\mathbf{z}_i$ is conditionally independent of $\mathbf{z}_{q(i)}$ given $\mathbf{y}_i$. Stein, Chi and Welty (2004) recommended including in $q(i)$ the indices of some close and some far-away observations, and grouping observations (i.e., $r_i > 1$) for computational advantages. Guinness (2018) considered an adaptive grouping scheme, and discovered that ordering schemes other than coord ordering can improve approximation accuracy. Vecchia (1988) and Stein, Chi and Welty (2004) focus on likelihood approximation, but Guinness (2018) also considers spatial prediction via conditional simulation. Further extensions were proposed in Sun and Stein (2016) and Huang and Sun (2018). Some asymptotics are provided in Zhang (2012).

3.2 Nearest-Neighbor GP (NNGP)

The NNGP (Datta et al., 2016a, 2016b, 2016c) considers explicit data models (such as the additive Gaussian noise assumed here), and conditions only on latent variables: $q_y(i) = q(i)$ and $q_z(i) = \varnothing$. A GP is defined by setting $r_i = 1$, $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_{\ell_y+\ell_z})$, $o = (\ell_y + 1, \ldots, \ell_y + \ell_z)$, and enforcing the constraint that $q(i) \subset (1, \ldots, \ell_y)$ for all i. This means that variables at the observed locations can condition only on variables in the knot set $(\mathcal{S}_1, \ldots, \mathcal{S}_{\ell_y})$.

3.3 Independent Blocks

The simplest special case is given by empty conditioning index vectors $q(i) = \varnothing$ for every i:

$$\widehat{f}(\mathbf{z}) = \int \prod_{i=1}^{\ell} f(\mathbf{z}_i|\mathbf{y}_i) f(\mathbf{y}_i)\, d\mathbf{y}$$

$$= \prod_{i=1}^{\ell} \int f(\mathbf{z}_i|\mathbf{y}_i) f(\mathbf{y}_i)\, d\mathbf{y} = \prod_{i=1}^{\ell} f(\mathbf{z}_i),$$

which treats the ℓ subvectors $\mathbf{z}_1, \ldots, \mathbf{z}_\ell$ independently and assumes $o = (1, \ldots, \ell)$. When each subvector or block corresponds to a contiguous subregion in space, Stein (2014) showed that this approximation can be quite competitive as a surrogate for the likelihood, and also computationally inexpensive since each term in the product incurs small computational cost and can be computed in parallel. One difficulty with this approach is characterization of joint uncertainties in predictions, due to the independence assumption embedded in the approximation.

3.4 Latent Autoregressive Process of Order m (AR(m))

Latent (vector-)AR processes of order m, also called state-space models, are common in time-series settings. They condition only on the latest m sets of latent variables for some ordering: $q(i) = q_y(i) = (i - m, \ldots, i - 1)$. Inference in this type of model is typically carried out using the Kalman filter and smoother (Kalman, 1960, Rauch, Tung and Striebel, 1965).

3.5 Modified Predictive Process (MPP)

The MPP (Finley et al., 2009) is obtained by defining $\mathcal{S}_1$ as a vector of "knot" locations, typically $1 \notin o$, and for all $i > 1$, set $r_i = 1$ and $q(i) = q_y(i) = 1$. This means that all variables condition on the same vector $\mathbf{y}_1$ and are assumed to be conditionally independent.

3.6 Full-Scale Approximation (FSA)

As in the MPP, the FSA-block (Snelson and Ghahramani, 2007, Sang, Jun and Huang, 2011) is obtained by designating a common conditioning vector $\mathbf{y}_1$, and setting $q(i) = q_y(i) = 1$ for $i > 1$. However, the FSA allows $r_i > 1$ and groups all remaining variables by spatial region as in the independent-blocks case in Section 3.3.

When $q(i) = q_y(i) = 1$ for all $i > 1$ in general Vecchia, we have $\widehat{f}(\mathbf{y}) = f(\mathbf{y}_1) \prod_{i=2}^{\ell} f(\mathbf{y}_i|\mathbf{y}_1)$, and so $\widehat{f}(\mathbf{y}_1) = f(\mathbf{y}_1)$, $\widehat{f}(\mathbf{y}_i) = \int f(\mathbf{y}_1) f(\mathbf{y}_i|\mathbf{y}_1)\, d\mathbf{y}_1 = f(\mathbf{y}_i)$ for $i > 1$ (i.e., the marginal distributions are exact), and $\widehat{f}(\mathbf{y}_i, \mathbf{y}_j) = \int f(\mathbf{y}_1) f(\mathbf{y}_i|\mathbf{y}_1) f(\mathbf{y}_j|\mathbf{y}_1)\, d\mathbf{y}_1$. Hence, as for the FSA, we have $\widehat{\mathrm{var}}(\mathbf{y}_i) = \mathrm{var}(\mathbf{y}_i)$, and for $i \neq j > 1$,

$$
\begin{aligned}
&\widehat{\mathrm{cov}}(\mathbf{y}_i, \mathbf{y}_j) \\
&\quad = \int\int\int \mathbf{y}_i \mathbf{y}_j' f(\mathbf{y}_1) f(\mathbf{y}_i|\mathbf{y}_1) f(\mathbf{y}_j|\mathbf{y}_1)\, d\mathbf{y}_i\, d\mathbf{y}_j\, d\mathbf{y}_1 \\
&\quad = \int \left(\int \mathbf{y}_i f(\mathbf{y}_i|\mathbf{y}_1)\, d\mathbf{y}_i \right) \\
&\qquad \times \left(\int \mathbf{y}_j' f(\mathbf{y}_j|\mathbf{y}_1)\, d\mathbf{y}_j \right) f(\mathbf{y}_1)\, d\mathbf{y}_1 \\
&\quad = \mathrm{cov}\big(\mathrm{E}(\mathbf{y}_i|\mathbf{y}_1), \mathrm{E}(\mathbf{y}_j|\mathbf{y}_1)\big),
\end{aligned}
$$

where $\mathrm{E}(\mathbf{y}_i|\mathbf{y}_1)$ is the predictive process with knots $\mathcal{S}_1$ evaluated at $\mathcal{S}_i$. The recently proposed smoothed FSA (Zhang, Sang and Huang, 2019), which is billed as a generalization of the Vecchia approach, can also be viewed as a special case of general Vecchia, for which the conditioning vectors include some nearby blocks in addition to the knot vector $\mathbf{y}_1$.

3.7 Multi-Resolution Approximation (MRA)

The MRA (Katzfuss, 2017) is an iterative extension of the FSA-block, in which the domain $\mathcal{D}$ is iteratively partitioned into J subregions, and we select r_i variables in each of the resulting subregions, such that $\mathcal{S}_i \subset \mathcal{D}_i$. For example, if $J = 4$, let $\mathcal{D}_1 = \mathcal{D}$, and define $\{\mathcal{D}_2, \ldots, \mathcal{D}_5\}$ to be a partition of $\mathcal{D}_1$, $\{\mathcal{D}_6, \ldots, \mathcal{D}_9\}$ to be a partition of $\mathcal{D}_2$, $\{\mathcal{D}_{10}, \ldots, \mathcal{D}_{13}\}$ to be a partition of $\mathcal{D}_3$, and so forth. Set $q(i) = \{j : \mathcal{D}_i \subset \mathcal{D}_j\}$, and $q_y(i) = q(i)$, so that the conditioning vector consists of latent variables associated with locations above it in the hierarchy.

The FSA and MPP are special cases of the MRA. All three methods allow latent variables at unobserved locations, such that $\mathcal{S}$ is different from the set of observed locations, which can be handled in our framework by $o \subset (1, \ldots, \ell)$.

3.8 Related Approach: Composite Likelihood (CL)

CL is a popular approach for fast GP inference. Varin, Reid and Firth (2011) categorize CL methods as either marginal or conditional. A common marginal CL approach is pairwise blocks, which approximates the likelihood as $\widehat{f}(\mathbf{z}) = \prod f(\mathbf{z}_i, \mathbf{z}_j)$, where the product is often over all pairs (i, j) of neighboring blocks (e.g., Eidsvik et al., 2014). Conditional CL is an approximation of the form (2), except that more general conditioning index vectors $g(i) \subset (1, \ldots, i-1, i+1, \ldots, \ell)$ are considered. In contrast to Vecchia approaches, CL-based inference is not generally guaranteed to become exact as the number of considered pairs or conditioning variables increases, and $\widehat{f}(\mathbf{z})$ is not generally guaranteed to be a valid joint density, which can make CL-based Bayesian inference difficult (see, e.g., Shaby, 2014). While the Vecchia approaches reviewed in Section 3.1 are special cases of conditional CL, our general Vecchia framework in (3) is not a CL approach, in that is defined on $\mathbf{x}$, not on $\mathbf{z}$ alone, and so it generally cannot be written in the form (2). Simulation studies comparing parameter estimation using Vecchia and CL approaches can be found in Appendix D.

4. INFERENCE AND COMPUTATIONS

In this section, we describe matrix representations of general Vecchia approximations, which enable fast inference. Further, we examine the sparsity of the involved matrices and derive the computational complexity.

4.1 Matrix Representations of General Vecchia

For any two subvectors $\mathbf{x}_i$ and $\mathbf{x}_j$ of $\mathbf{x}$, we write $C(\mathbf{x}_i, \mathbf{x}_j) = E(\mathbf{x}_i \mathbf{x}_j')$, the cross-covariance between $\mathbf{x}_i$ and $\mathbf{x}_j$. This gives $C(\mathbf{y}_i, \mathbf{y}_j) = C(\mathbf{z}_i, \mathbf{y}_j) = K(\mathcal{S}_i, \mathcal{S}_j)$, $C(\mathbf{z}_i, \mathbf{z}_j) = K(\mathcal{S}_i, \mathcal{S}_j)$ for $i \neq j$, and $C(\mathbf{z}_i, \mathbf{z}_i) = K(\mathcal{S}_i, \mathcal{S}_i) + \tau^2 \mathbf{I}_{r_i}$.

We can write the general Vecchia approximation in (3) as

$$\widehat{f}(\mathbf{x}) = \prod_{i=1}^{b} f(\mathbf{x}_i | \mathbf{x}_{g(i)}) = \prod_{i=1}^{b} \mathcal{N}(\mathbf{x}_i | \mathbf{B}_i \mathbf{x}_{g(i)}, \mathbf{D}_i), \tag{6}$$

where $\mathbf{B}_i = C(\mathbf{x}_i, \mathbf{x}_{g(i)}) C(\mathbf{x}_{g(i)}, \mathbf{x}_{g(i)})^{-1}$ and $\mathbf{D}_i = C(\mathbf{x}_i, \mathbf{x}_i) - \mathbf{B}_i C(\mathbf{x}_{g(i)}, \mathbf{x}_i)$. We view $\mathbf{B}_i$ as a block matrix, and so $(\mathbf{B}_i)_{\#(j,g(i))}$ is the block of $\mathbf{B}_i$ corresponding to $\mathbf{x}_j$ when $j \in g(i)$.

For any symmetric, positive-definite matrix $\mathbf{A}$, let $\text{chol}(\mathbf{A})$ be the lower-triangular Cholesky factor of $\mathbf{A}$, and let $\mathbf{P}$ be a permutation matrix so that $\mathbf{PA}$ reorders the rows of matrix $\mathbf{A}$ in reverse order. Then, we call $\text{rchol}(\mathbf{A}) := \mathbf{P}(\text{chol}(\mathbf{PAP}))\mathbf{P}$ the reverse Cholesky factor of $\mathbf{A}$ (i.e., the row-column reversed Cholesky factor of the row-column reversed $\mathbf{A}$). The following proposition is a standard result for the multivariate normal distribution.

PROPOSITION 1. *For the density in (6), we have $\widehat{f}(\mathbf{x}) = \mathcal{N}_n(\mathbf{x}|\mathbf{0}, \widehat{\mathbf{C}})$, where $\widehat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}'$, $\mathbf{U}$ is a sparse upper triangular $b \times b$ block matrix with (j,i)th block*

$$\mathbf{U}_{ji} = \begin{cases} \mathbf{D}_i^{-1/2}, & i = j, \\ -(\mathbf{B}_i)'_{\#(j,g(i))} \mathbf{D}_i^{-1/2}, & j \in g(i), \\ \mathbf{0}, & \text{otherwise}, \end{cases} \tag{7}$$

and $\mathbf{D}_i^{-1} = \mathbf{D}_i^{-1/2} (\mathbf{D}_i^{-1/2})'$. Further, $\mathbf{U} = \text{rchol}(\widehat{\mathbf{C}}^{-1})$ is the reverse Cholesky factor of $\widehat{\mathbf{C}}^{-1}$.

All proofs can be found in Appendix F.

4.2 Likelihood

By integrating $\widehat{f}(\mathbf{x})$ with respect to the latent $\mathbf{y}$, the general Vecchia approximation implies a distribution for the observed vector $\mathbf{z}_o$ as in (5). For large n_y, numerical integration with respect to the n_y-dimensional vector $\mathbf{y}$ is challenging (see Finley et al., 2019). Hence, we consider the analytically integrated density instead.

PROPOSITION 2. *The general Vecchia likelihood can be computed as*

$$\begin{aligned} -2 \log \widehat{f}(\mathbf{z}_o) = &\sum_{i=1}^{b} \log |\mathbf{D}_i| + 2 \sum_{i=1}^{\ell} \log |\mathbf{V}_{ii}| \\ &+ \tilde{\mathbf{z}}' \tilde{\mathbf{z}} - (\mathbf{V}^{-1} \mathbf{U}_Y \tilde{\mathbf{z}})' (\mathbf{V}^{-1} \mathbf{U}_Y \tilde{\mathbf{z}}) \\ &+ n_z \log(2\pi), \end{aligned} \tag{8}$$

where $\mathbf{V} := \text{rchol}(\mathbf{W})$, $\mathbf{W} := \mathbf{U}_Y \mathbf{U}_Y'$, $\tilde{\mathbf{z}} := \mathbf{U}_Z' \mathbf{z}_o$, and $\mathbf{U}_Y := \mathbf{U}_{\#(\mathbf{y},\mathbf{x})\bullet}$ and $\mathbf{U}_Z := \mathbf{U}_{\#(\mathbf{z}_o,\mathbf{x})\bullet}$ consist of the rows of $\mathbf{U}$ corresponding to $\mathbf{y}$ and $\mathbf{z}_o$, respectively.

Note that, analogously to $\mathbf{U} = \text{rchol}(\widehat{\mathbf{C}}^{-1})$ in Proposition 1, we compute $\mathbf{V} = \text{rchol}(\mathbf{W})$ as the reverse Cholesky factor of $\mathbf{W}$. This allows us to derive the sparsity structure of $\mathbf{V}$ in Proposition 3 below, and ensures low computational complexity for certain configurations of general Vecchia.

Thus, for these configurations, the likelihood $\widehat{f}(\mathbf{z}_o)$ (integrated over $\mathbf{y}$) can be evaluated quickly for any given value of the parameters $\boldsymbol{\theta}$ and τ^2, which enables likelihood-based parameter inference even for very large datasets. Our framework is agnostic with respect to the inferential paradigm, allowing both frequentist and Bayesian inference. Frequentist inference can be carried out by finding the parameter values that maximize $\widehat{f}(\mathbf{z}_o)$. Appendix D details a simulation in which we compared SGV to the exact likelihood and two composite likelihood methods on estimating a spatial range parameter. We found that SGV gave very similar parameter estimates to the exact maximum likelihood estimates. Appendix E demonstrates how the SGV likelihood can be used to conduct Bayesian inference. This example considered both numerically integrated posteriors and a Metropolis–Hastings algorithm, based on which we obtained the posterior predictive distribution of $y(\cdot)$ at unobserved locations.

No matter the inferential paradigm, our models can be viewed as approximations of GP models, or as valid probability models in their own right (see (6)). All our inference is exact from the latter perspective. The error due to the Vecchia approximation itself disappears for large $m = n - 1$, and it is examined numerically for smaller m in Section 7.

4.3 Prediction

For prediction, we can compute the posterior distribution of the error-free process vector given by $\mathbf{y}|\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{W}^{-1})$, where $\boldsymbol{\mu} := -\mathbf{W}^{-1} \mathbf{U}_y \tilde{\mathbf{z}}$. However, this requires consideration of complex issues, such as how to guarantee fast computation of relevant summaries of this distribution, and what ordering and conditioning strategies work well in the context of prediction at observed and unobserved locations. Thus, we refer to Katzfuss et al. (2020) for details on how to extend the general Vecchia framework to GP prediction.

4.4 Sparsity Structures

In the following proposition, we use connections between Vecchia approaches and DAGs (see Section 2.4 and Appendix B) to verify the sparsity structure of $\mathbf{U}$ in (7) and to determine the sparsity of $\mathbf{W}$ and $\mathbf{V} = \text{rchol}(\mathbf{W})$, which must be computed for inference.

PROPOSITION 3.

1. $\mathbf{U}_{ji} = \mathbf{0}$ *if* $i \neq j$ *and* $\mathbf{x}_j \not\to \mathbf{x}_i$.

2. $\mathbf{W}_{ji} = \mathbf{0}$ *if* $i \neq j$, $\mathbf{y}_j \not\to \mathbf{y}_i$, $\mathbf{y}_i \not\to \mathbf{y}_j$, *and there is no* $k > \max(i, j)$ *such that both* $\mathbf{y}_i \to \mathbf{y}_k$ *and* $\mathbf{y}_j \to \mathbf{y}_k$.

3. $\mathbf{V}_{ji} = \mathbf{0}$ *if* $j > i$. *For* $j < i$, $\mathbf{V}_{ji} = \mathbf{0}$ *if there is no path between* $\mathbf{y}_i$ *and* $\mathbf{y}_j$ *on the subgraph* $\{\mathbf{y}_i, \mathbf{y}_j\} \cup \{\mathbf{y}_k : k > i, \mathbf{y}_k$ *has at least one observed descendant*$\}$.

Thus, $\mathbf{U}$ and $\mathbf{V}$ are upper triangular, and the sparsity of the upper triangle depends on the Vecchia specification. For $j < i$, the (j, i) block of $\mathbf{W}$ is not only nonzero if $j \in q_y(i)$, but also if $\mathbf{y}_i$ and $\mathbf{y}_j$ appear in a conditioning vector together (i.e., $i, j \in q_y(k)$ for some k). Note that this sparsity structure corresponds to the adjacent vertices in the so-called moral graph (e.g., Lauritzen, 1996, Section 2.1.1). The matrix $\mathbf{V}$ is typically at least as dense as $\mathbf{W}$ (assuming that all or most $\mathbf{y}_k$ have observed descendants), in that it has the same nonzeros as $\mathbf{W}$ plus additional ones induced by more complicated paths. Figure 1 shows examples of DAGs with the corresponding sparsity structures of $\mathbf{U}$ and $\mathbf{V}$.

4.5 Computational Complexity

Recall that $\mathbf{x}$ consists of $\mathbf{y} = (\mathbf{y}_1, \ldots, \mathbf{y}_\ell)$ and $\mathbf{z}_o$, where $\mathbf{z} = (\mathbf{z}_1, \ldots, \mathbf{z}_\ell)$, and $\mathbf{y}_i$ and $\mathbf{z}_i$ are of length r_i. Let n be the total number of individual variables in $\mathbf{x}$. To simplify the sparsity and computational complexity calculations, assume that $o = (1, \ldots, \ell)$, all r_i are of the same order r (and so $n \approx br = 2\ell r$), and that all conditioning vectors consist of at most m subsets (of size r): $|g(i)| \leq m$.

Then, it is easy to see from (7) that $\mathbf{U}$ has $\mathcal{O}(b \cdot (mr^2)) = \mathcal{O}(nmr)$ nonzero elements and can be computed in $\mathcal{O}(b \cdot (m^3 r^3)) = \mathcal{O}(nm^3 r^2)$ time. Note that this time complexity is lower in r than in m, whose product mr makes up the total length of the conditioning vectors.

To evaluate the likelihood in (8), we also need to compute $\mathbf{W} = \mathbf{U}_Y \mathbf{U}_Y'$ and find its reverse Cholesky factor $\mathbf{V} = \text{rchol}(\mathbf{W})$. Each conditioning vector is of size $|g(i)| \leq m$ and so contains at most $m(m-1)/2$ pairs of elements. Therefore, from Proposition 3, $\mathbf{W}$ has at most $\mathcal{O}(\ell m^2)$ nonzero blocks. Since each block is of size $r \times r$ and $\ell = \mathcal{O}(n/r)$, $\mathbf{W}$ has at most $\mathcal{O}(nrm^2)$ nonzero elements.

The time complexity for obtaining a lower-triangular Cholesky factor is on the order of the sum of the squares of the number of nonzero elements per column in the factor (e.g., Toledo, 2007, Thm. 2.2). It can be easily verified that the same holds for our reverse Cholesky factor $\mathbf{V}$. Thus, for any particular Vecchia approximation, the time complexity for inference can in principle be determined based on the corresponding DAG using Proposition 3. We give examples in the next section.

5. SPARSE GENERAL VECCHIA (SGV) APPROXIMATION

We now study choice C5 from Section 2.3 for fixed choices C1–C4; that is, we assume that the grouping, ordering, and the conditioning index vectors $q(1), \ldots, q(\ell)$ are fixed. We consider three methods that differ in their choice of latent versus observed conditioning (C5): the existing methods standard Vecchia (Section 3.1) and latent Vecchia (used in the NNGP in Section 3.2), and a novel sparse general Vecchia (SGV) approach:

Standard Vecchia ($\widehat{f_s}$): $q_z(i) = q(i)$, condition only on observed vectors $\mathbf{z}_j$.

Latent Vecchia ($\widehat{f_l}$): $q_y(i) = q(i)$, condition only on latent vectors $\mathbf{y}_j$.

SGV ($\widehat{f_g}$): For each i, partition $q(i)$ into $q_y(i)$ and $q_z(i)$ such that j and k with $j < k$ can only both be in $q_y(i)$ if $j \in q_y(k)$.

In the terminology of Lauritzen (1996), Section 2.1.1, SGV ensures that the corresponding DAG forms a perfect graph. Different versions of SGV are possible for the same conditioning index vectors (and, in fact, standard Vecchia is one special case of SGV). Throughout this article, we consider the following strategy that attempts to maximize latent conditioning in the SGV: We obtain the latent-conditioning index vector $q_y(i)$ for each $i = 2, \ldots, \ell$ by first finding the index $k_i \in q(i)$ whose latent-conditioning index vector has the most overlap with $q(i)$: $k_i = \arg\max_{j \in q(i)} |q_y(j) \cap q(i)|$. In case of a tie, we choose the k_i for which the spatial distance between $\mathcal{S}_i$ and $\mathcal{S}_{k_i}$ is shortest. Then, we set $q_y(i) = (k_i) \cup (q_y(k_i) \cap q(i))$, with the remaining indices in $q(i)$ corresponding to observed conditioning: $q_z(i) = q(i) \setminus q_y(i)$.

The three approaches are illustrated in a toy example with $\ell = 7$ shown in Figure 1. For all three methods, we have the same $q(1), \ldots, q(\ell)$: $q(2) = (1)$, $q(3) = (1, 2)$, $q(4) = (1, 3)$, $q(5) = (2, 4)$, ... Like latent Vecchia, SGV uses $q_y(2) = (1)$, $q_y(3) = (1, 2)$, $q_y(4) = (1, 3)$, as, for example, $1 \in q_y(3)$, and so $q_y(4)$ can contain both 1 and 3. However, $2 \notin q_y(4)$, and so SGV does not allow both $2 \in q_y(5)$ and $4 \in q_y(5)$, and sets $q_y(5) = (4)$ and $q_z(5) = (2)$.

We now establish an ordering on the accuracy of the approximations to $f(\mathbf{x})$.

PROPOSITION 4. *The following ordering of Kullback–Leibler (KL) divergences holds*:

$$\text{KL}\big(f(\mathbf{x}) \| \widehat{f_l}(\mathbf{x})\big) \leq \text{KL}\big(f(\mathbf{x}) \| \widehat{f_g}(\mathbf{x})\big) \leq \text{KL}\big(f(\mathbf{x}) \| \widehat{f_s}(\mathbf{x})\big).$$

Thus, the approximation accuracy for the joint distribution of $\mathbf{x}$ is better for latent Vecchia than for SGV, which is better than that for standard Vecchia. Note, however, that this does not guarantee that the KL divergence for the implied distribution of the observations $\mathbf{z}_o$ follows the same ordering. For example, Proposition 4 says that $\text{E}^f(\log \widehat{f_g}(\mathbf{x})) \geq \text{E}^f(\log \widehat{f_s}(\mathbf{x}))$, but that does not guarantee that $\text{E}^f(\log \int \widehat{f_g}(\mathbf{x})\, d\mathbf{y}) \geq \text{E}^f(\log \int \widehat{f_s}(\mathbf{x})\, d\mathbf{y})$. Examples of this can be found in Figure 3(d).

Another important factor is the computational complexity of the different approaches. Standard Vecchia only

conditions on observed quantities, and so for any i, j we have $\mathbf{y}_j \not\to \mathbf{y}_i$, resulting in a diagonal $\mathbf{W}$ and $\mathbf{V}$ according to Proposition 3, and hence an overall time complexity of $\mathcal{O}(nm^3r^2)$ for standard Vecchia.

Finley et al. (2019) observed numerically that matrices in the NNGP (which uses the latent Vecchia approach) were less sparse than in standard Vecchia. We can examine this issue further using Proposition 3. In the toy example in Figure 1, latent Vecchia uses $q_y(5) = (2, 4)$, which creates the path $(\mathbf{y}_2, \mathbf{y}_5, \mathbf{y}_4)$ that leads to $\mathbf{V}_{2,4} \neq \mathbf{0}$. Setting $q_y(6) = (3, 5)$ creates the path $(\mathbf{y}_3, \mathbf{y}_6, \mathbf{y}_7, \mathbf{y}_5)$, leading to $\mathbf{V}_{3,5} \neq \mathbf{0}$. This results in $3 > m = 2$ nonzero off-diagonal elements in columns 4 and 5. We provide more insight into the increased computational cost for latent Vecchia in the following proposition.

PROPOSITION 5. *Consider the latent Vecchia approach with* $r_i = 1$, $o = (1, \ldots, \ell)$, $m \leq n_z^{1/d}$, *and coordinate-wise ordering for locations on an equidistant grid in a* d*-dimensional hypercube with nearest-neighbor conditioning. Then,* $\mathbf{V} = \text{rchol}(\mathbf{W})$ *has* $\mathcal{O}(n^{1-1/d}m^{1/d})$ *nonzero elements per column, requiring* $\mathcal{O}(n^{2-1/d}m^{1/d})$ *memory. The resulting time complexity for computing* $\mathbf{V} = \text{rchol}(\mathbf{W})$ *is* $\mathcal{O}(n^{3-2/d}m^{2/d})$.

Thus, the time complexity for obtaining $\mathbf{V}$ is $\mathcal{O}(nm^2)$ in $d = 1$ dimensions, $\mathcal{O}(n^2m)$ for $d = 2$, and approaching the cubic complexity in n of the original GP as d increases. For irregular observation locations, we expect roughly similar scaling if the locations can be considered to have been drawn from independent uniform distributions over the domain. Also note that using reordering algorithms for the Cholesky decomposition (as opposed to simple reverse ordering) could lead to different complexities, although our numerical results indicate that this might actually increase the computational complexity (see Figure 5(b)).

In contrast to latent Vecchia, SGV results in guaranteed sparsity. In the toy example, SGV sets $q_y(5) = (4)$ and $q_z(5) = (2)$ because $2 \notin q_y(4)$, and $q_y(6) = (5)$ and $q_z(6) = (3)$ because $3 \notin q_y(5)$, resulting in $\mathbf{V}_{2,4} = \mathbf{V}_{3,5} = \mathbf{0}$ (in contrast to latent Vecchia). More generally, SGV preserves the linear scaling of standard Vecchia, in any spatial dimension and for gridded or irregularly spaced locations.

PROPOSITION 6. *For SGV,* $\mathbf{V}$ *has at most* mr *off-diagonal elements per column, and so the time complexity for computing* $\mathbf{V} = \text{rchol}(\mathbf{W})$ *is only* $\mathcal{O}(nm^2r^2)$. *Thus, SGV has the same overall computational complexity as standard Vecchia.*

In summary, SGV provides improvements in approximation accuracy over standard Vecchia (Proposition 4) while retaining linear computational complexity in n (Proposition 6). Latent Vecchia results in improved approximation accuracy but can raise the computational complexity severely (Proposition 5), which can be infeasible for large n. Numerical illustrations of these results can be found in Section 7.

6. ORDERING AND CONDITIONING

We now provide some insight into choices C2–C4 of Section 2.3. For simplicity, we henceforth assume $r_i = 1$ unless stated otherwise.

6.1 Ordering (C3)

In one spatial dimension, a "left-to-right" ordering of the locations in $\mathcal{S}$ is natural. However, in two or more spatial dimensions, it is not obvious how the locations should be ordered. For Vecchia approaches, the default and most popular ordering is along one of the spatial coordinates (coord ordering). Datta et al. (2016a) only observed a negligible effect of the ordering on the quality of the Vecchia approximation, but Guinness (2018) showed that this is not always the case. He proposed different ordering schemes, including an approximate maximum-minimum-distance (maxmin) ordering, which sequentially picks each location in the ordering by aiming to maximize the distance to the nearest of the previous locations. Guinness (2018) showed that maxmin ordering can lead to substantial improvements over coord ordering in settings without any nugget or noise. We will examine the nonzero nugget case in Section 7. Note that the MRA (Section 3.7) implies an ordering scheme similar to maxmin, starting with a coarse grid over space and subsequently getting denser and denser.

6.2 Choosing m

For a given ordering, as part of C4 we must choose m, the size of the conditioning vectors.

For one-dimensional spatial domains, some guidance can be obtained for approximating a GP with a Matérn covariance on a one-dimensional domain. If the smoothness is $\nu = 0.5$, we have a Markov process of order 1, and so we can get an exact approximation for latent conditioning with $m = 1$ by ordering from left to right. Stein (2011) conjectures that for smoothness ν, approximate screening holds for any $m > \nu$. This conjecture is explored numerically in Section 7, specifically in Figure 2(a). Note that coord ordering in 1-D with m-nearest-neighbor conditioning amounts to an AR(m) model, and the corresponding latent or SGV inference is equivalent to a Kalman filter and smoother (cf. Eubank and Wang, 2002). For very smooth processes (i.e., very large ν), the m necessary for (approximate) screening won't be affordable any more, and alternative ordering and conditioning strategies might be advantageous (see Section 6.3 below).

For two or more dimensions, the necessary m will depend not only on the smoothness of the covariance function, but also on the chosen ordering, the observation locations (regular or irregular), and other factors. We suggest

starting with a relatively small m and gradually increasing it using warm-starts based on previously obtained parameter estimates, until the estimates have converged to a desired tolerance, or until the available computational resources have been exhausted.

6.3 Conditioning

For a given ordering and m, the most common strategy is to simply condition on the m nearest neighbors or locations (NN conditioning), although more elaborate conditioning schemes have been proposed (see, e.g., Stein, Chi and Welty, 2004, Gramacy and Apley, 2015). It can also be advantageous in some situations to place a coarse grid over space at the beginning of the ordering and to always condition on this grid. Properties of this same-conditioning-set (SCS) approach are described in Appendix C.

7. NUMERICAL STUDY

We examined numerically the propositions and claims made in previous sections. We explored C3–C5 from Section 2.3, with an emphasis on C5 by comparing the three approaches from Section 5. Throughout this section, we set $r_i = 1$ and $o = (1, \dots, \ell)$, so that each latent variable had a corresponding observed variable. The observation locations were equidistant grids on the unit interval or unit square, and the true GP was assumed to have Matérn covariance with variance σ^2, smoothness ν, and effective range λ (i.e., the distance at which the correlation drops to 0.05). We added noise with variance τ^2, set $\sigma^2 + \tau^2 = 1$, and so the signal proportion was $\sigma^2/(\sigma^2 + \tau^2) = \sigma^2$. For example, signal proportions of $1/2$ and $2/3$ correspond to signal-to-noise ratios (SNRs) of 1 and 2, respectively. We considered coordinate-wise (coord) ordering and the approximate maximum-minimum-distance (maxmin) ordering of Guinness (2018). We used nearest-neighbor (NN) conditioning for a given ordering, unless stated otherwise. Comparisons among methods are made using the Kullback–Leibler (KL) divergence between the approximate distribution $\widehat{f}(\mathbf{z})$ and the true distribution $f(\mathbf{z})$.

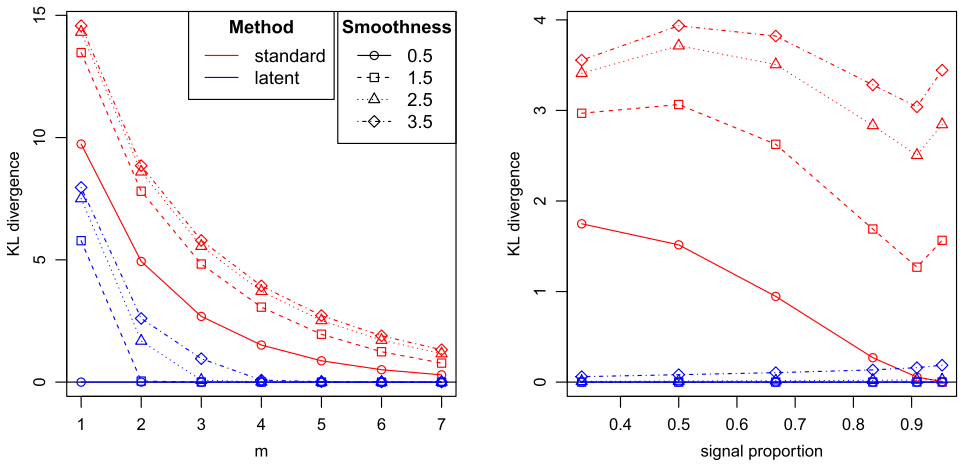
First, we assumed a one-dimensional spatial domain, $\mathcal{D} = [0, 1]$, and only considered the natural coord ordering "from left to right." The different methods from Section 5 then essentially correspond to latent or nonlatent AR(m) processes. From the results shown in Figure 2 with $n_z = 100$ and $\lambda = 0.9$, we can see that latent Vecchia and the equivalent SGV performed much better than standard Vecchia. Figure 2(a) also confirms numerically the conjecture from Section 6.2 that (approximate) screening holds for latent Vecchia if $m > \nu$.

The remaining results are for a two-dimensional domain, $\mathcal{D} = [0, 1]^2$. Exploring Proposition 4, Figure 3 shows KL divergences for different values of σ^2, ν, and m, all for $n_z = 6400$ and $\lambda = 0.9$. As we can see, the KL divergences for the three methods roughly followed the ordering from Section 5, with latent Vecchia performing better than SGV, which performed better than standard Vecchia. (For SNR $= \infty$, the methods are equivalent.) The screening effect is less clear in two dimensions. Also note that maxmin ordering often resulted in tremendous improvements over coord ordering, except for standard Vecchia, where the two orderings produced similar results.

We also considered very smooth covariances, which are less common in geostatistics but very popular in machine learning. We explored conditioning on the same first m variables in the maxmin ordering, which are spread throughout the domain. Figure 4 shows that this can result in strong improvements over NN ordering for SGV.

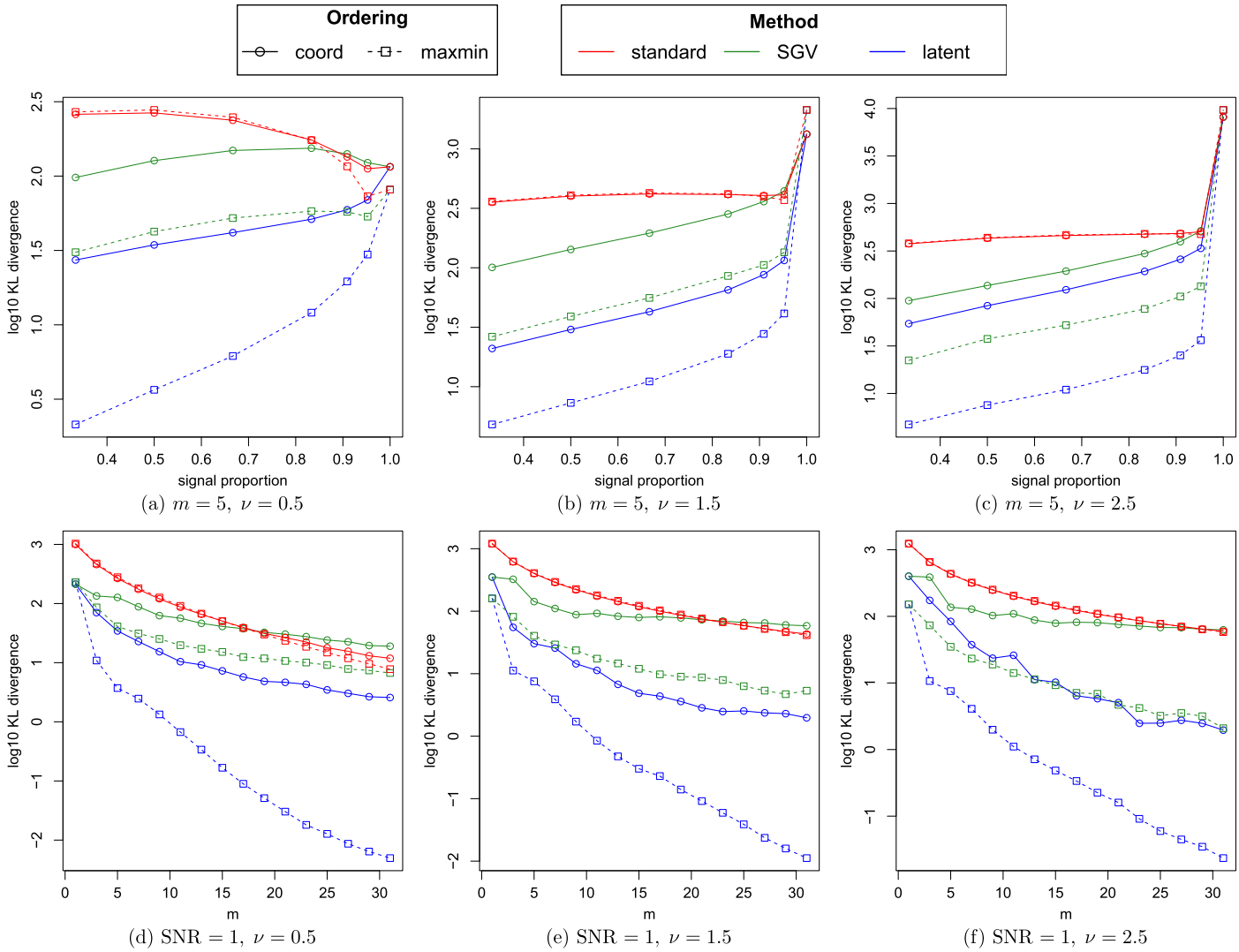
The computational feasibility of the methods is explored in Figure 5, which examines the sparsity of the matrix $\mathbf{V}$. We can see that SGV keeps the number of nonzero elements per column in $\mathbf{V}$ at or below m, as would be expected from Proposition 6, resulting in linear scaling as a function of n. For latent Vecchia, $\mathbf{V} = \text{rchol}(\mathbf{W})$ is considerably denser, and the computational complexity for obtaining $\mathbf{V}$ scales roughly as $\mathcal{O}(n^2)$, as expected from Proposition 5. MMD ordering of $\mathbf{W}$ did not improve the complexity for coord. Figure 5(c) shows actual computation times for obtaining $\mathbf{V}$ from $\mathbf{W}$ using the `chol` function in the R package `spam` (Furrer and Sain, 2010) on a 4-core machine (Intel Core i7-3770) with 3.4 GHz and 16 GB RAM. Despite the `chol` function being more highly optimized for the default MMD ordering than the user-supplied reverse ordering, latent Vecchia with MMD ordering is roughly two orders of magnitude slower than SGV with reverse ordering for n_z around 100,000. A further timing study in Katzfuss et al. (2020) shows that, for standard Vecchia and SGV, the time for computing $\mathbf{V}$ is negligible relative to that for $\mathbf{U}$; hence, for a given n and m, standard Vecchia and SGV require almost the same computation time. In contrast, latent Vecchia can be orders of magnitude slower when n is large.

Figure 6 shows a comparison for large n of four methods that all scale linearly, namely SGV, standard Vecchia, MRA (Section 3.7), and independent blocks (Section 3.3), using maxmin ordering where applicable. We set $\mathcal{D} = [0, 1]^2$, $\sigma = \tau = 1$, and $\lambda = 0.9$. In Figure 6(c), we explored the accuracy of the methods under infill asymptotics, by simulating data on a fine 280×280 grid, and then, starting with a coarse subset or subgrid of size 100×100, considering larger and larger subsets of the data. It is infeasible to compute the exact KL divergence for large n_z, and so we approximated it by subtracting each method's loglikelihood from the loglikelihood for SGV with large $m = 40$, all averaged over 10 simulated datasets. While the time complexity for independent blocks and MRA (see Appendix C) is only $\mathcal{O}(m^{2/3})$ of



(a) SNR $= 1$ (b) $m = 4$

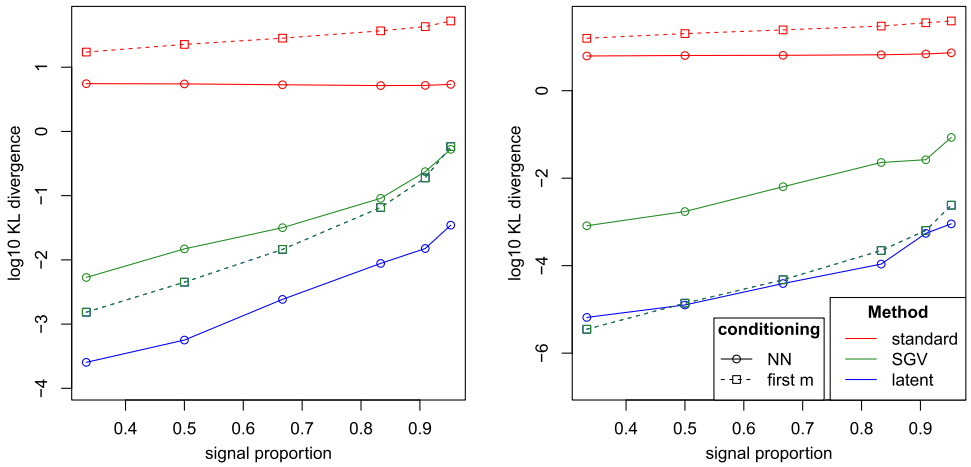
FIG. 2. *KL divergences for Vecchia approximations of a GP with Matérn covariance on the unit interval with coord ordering and NN conditioning. SGV is equivalent to latent Vecchia in this setting.*



(a) $m = 5,\ \nu = 0.5$ (b) $m = 5,\ \nu = 1.5$ (c) $m = 5,\ \nu = 2.5$

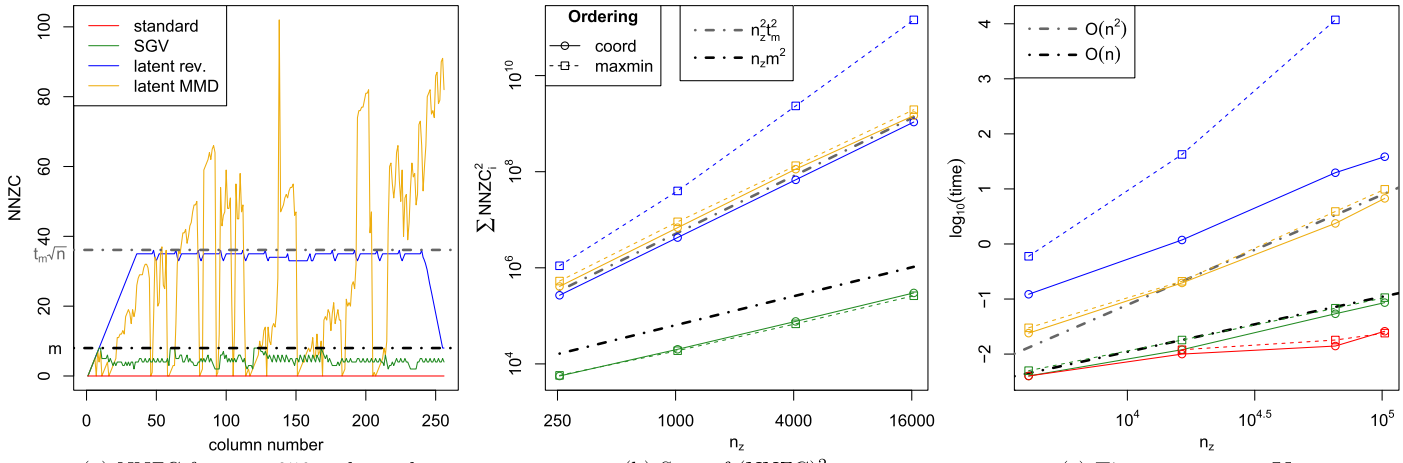
(d) SNR $= 1,\ \nu = 0.5$ (e) SNR $= 1,\ \nu = 1.5$ (f) SNR $= 1,\ \nu = 2.5$

FIG. 3. *KL divergences (on a log scale) for a Matérn covariance with smoothness ν on the unit square. Panels* (a)–(c): *fixed $m = 5$, varying signal proportion, with symbols corresponding to (from left to right) SNRs of* 0.5, 1, 2, 5, 10, 20, ∞, *respectively. Panels* (d)–(f): *fixed SNR $= 1$, varying m.*



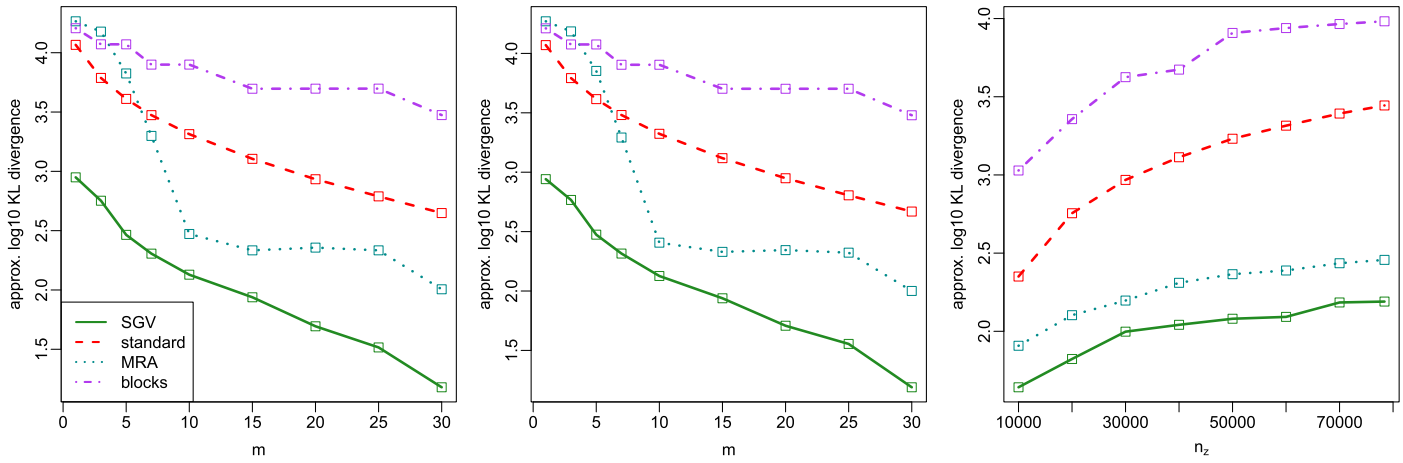
(a) $\nu = 3$ (b) $\nu = 10$

FIG. 4. *KL divergences (on a log scale) for smooth covariances, comparing nearest-neighbor (NN) conditioning versus always choosing the first m variables*; $n_z = 400$, $m = 16$, *maxmin ordering*, $\lambda \approx 2$. *For first-m conditioning, SGV and latent are equivalent.*



(a) NNZC for $n_z = 256$ and coord (b) Sum of $(\text{NNZC})^2$ (c) Time to compute $\mathbf{V}$

FIG. 5. *Sparsity, complexity, and actual computation times of obtaining* $\mathbf{V}$ *with* $m = 8$, *and so* $t_m = (2m/\pi)^{1/2} \approx 2.25$ (*see proof of Proposition 5*). *NNZC: number of nonzero off-diagonal elements per column in* $\mathbf{V}$; *MMD and rev.: multiple minimum degree and reverse ordering, respectively, for Cholesky algorithm.*



(a) $\nu = 0.5$, $n_z = 62{,}500$ (b) $\nu = 1.5$, $n_z = 62{,}500$ (c) $\nu = 1.5$, $m = 10$

FIG. 6. *Comparison of SGV and standard Vecchia to MRA and independent blocks.*

that for the Vecchia approaches, SGV outperformed all other approaches even when adjusting for differences in complexity.

8. CONCLUSIONS AND GUIDELINES

We have presented a general class of sparse GP approximations based on applying Vecchia's approximation to a vector consisting of latent GP realizations and their corresponding noisy observations. Several of the most commonly used GP approximations proposed in the literature are special cases of our class. We provided a formula for fast computation of the likelihood, and we studied the sparsity and computational complexity using connections between Vecchia approaches and directed acyclic graphs. We proposed a novel sparse general Vecchia (SGV), which can dramatically improve upon the approximation accuracy of standard Vecchia while maintaining its linear computational complexity. In contrast, we showed that latent Vecchia (which is used in the nearest-neighbor GP) can scale quadratically in the data size in two-dimensional space.

We now give some guidelines for using the general Vecchia approach in practice. In general, we recommend using our SGV approximation in the presence of nugget or noise, and standard Vecchia if the noise term is zero or almost zero. In one spatial dimension, left-to-right ordering and nearest-neighbor conditioning is most natural, and SGV is equivalent to latent. In addition, the size m of the conditioning vector can be chosen according to the smoothness (i.e., differentiability at the origin) of the covariance function. In two-dimensional space, we recommend maxmin ordering. While it is difficult to determine a suitable m a priori, a useful approach is to carry out inference for small m, and then gradually increase m until the inference converges or the computational resources are exhausted. Nearest-neighbor conditioning is suitable for low smoothness, while conditioning on the first m latent variables is preferable for higher smoothness when there is a large nugget. This first-m conditioning and its extensions (such as the MRA) has benefits beyond approximation accuracy, such as reduced computational complexity, exact marginal distributions for all variables, and sparse Cholesky factor of the posterior covariance matrix. While our methods are, in principle, applicable in more than two dimensions, a thorough investigation of their properties in this context is warranted and will be carried out in future work.

The methods and algorithms proposed here are implemented in the R package `GPvecchia`. Katzfuss et al. (2020) extend the general Vecchia framework to GP prediction at observed and unobserved locations. They also provide further details on computational issues and timing, and an application to a large satellite dataset.

APPENDIX A: VECTOR NOTATION

We define vectors to be objects that contain an ordered list of elements of the same type, equipped with union and intersection operations. We generally use nonbold lowercase letters for vectors of integers (e.g., o, p, q). We use bold lowercase letters (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{z}$) for vectors of real numbers or vectors of vectors. Using vectors of vectors makes some of the early definitions slightly cumbersome but greatly simplifies the main unifying results of the paper. Bold uppercase letters usually refer to matrices (e.g., $\mathbf{C}, \mathbf{K}$), and script letters (e.g., $\mathcal{S}$) for vectors of locations or vectors of vectors of locations.

For example, define $\mathbf{y} = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4, \mathbf{y}_5)$ as a vector of vectors. Subvectoring is accomplished with index vectors and uses subscript notation, for example if $o = (4, 1, 2)$ is a vector of indices, then $\mathbf{y}_o = (\mathbf{y}_4, \mathbf{y}_1, \mathbf{y}_2)$, respecting the ordering of the index vector. Unions of vectors are vectors and are defined when the two vectors have the same type and when the ordering of the union is defined. For example, if $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$, then $\mathbf{y}_o \cup \mathbf{z} = (\mathbf{y}_4, \mathbf{y}_1, \mathbf{z}_1, \mathbf{y}_2, \mathbf{z}_2)$ is a complete definition of the union of $\mathbf{y}_o$ and $\mathbf{z}$. Likewise, the intersection $\mathbf{y} \cap \mathbf{z}$ consists of the common elements of the two vectors and is fully defined when the ordering of the intersection is defined.

When the situation demands more abstractness, the ordering of the elements of the union or intersection can be defined via an index function # that inputs an element and a vector and returns the index occupied by the element in the vector. Continuing the example above, $\#(\mathbf{y}_4, \mathbf{y}) = 4$, whereas $\#(\mathbf{y}_4, \mathbf{y}_o \cup \mathbf{z}) = 1$. The index function is vectorized, meaning that $\#(\mathbf{z}, \mathbf{y} \cup \mathbf{z}) = (3, 5)$ returns the vector of indices occupied by $\mathbf{z}$ in $\mathbf{y} \cup \mathbf{z}$. This allows the index function to act as an inverse of the union operator, in the sense that $(\mathbf{y} \cup \mathbf{z})_{\#(\mathbf{z}, \mathbf{y} \cup \mathbf{z})} = \mathbf{z}$.

Vectors whose elements are real numbers are considered as the usual column vectors to which vector addition and multiplication rules apply. Matrices are simply two-dimensional vectors that use double subscripting, and all matrices are viewed as block matrices, with the blocks defined based on context. Functions are vectorized with respect to vectors of locations. For example, if $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\ell)$, $\mathbf{A} = K(\mathcal{S}, \mathcal{S})$ is an $\ell \times \ell$ block matrix with block $\mathbf{A}_{ij} = K(\mathcal{S}_i, \mathcal{S}_j)$. We use $\bullet$ to represent the vector of all indices, and so $\mathbf{A}_{i\bullet} = K(\mathcal{S}_i, \mathcal{S})$.

APPENDIX B: REVIEW OF DIRECTED ACYCLIC GRAPHS (DAGS)

Here we provide a brief review of DAGs (see, e.g., Rütimann and Bühlmann, 2009, Section 2). A directed graph consists of vertices, say $\{\mathbf{x}_1, \ldots, \mathbf{x}_b\}$, and directed edges (i.e., arrows). Two vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are called adjacent if there is an edge between them. If the edge is directed from $\mathbf{x}_j$ to $\mathbf{x}_i$, $\mathbf{x}_j$ is called a parent of $\mathbf{x}_i$, and we

write $\mathbf{x}_j \to \mathbf{x}_i$. If there is no directed edge from $\mathbf{x}_j$ to $\mathbf{x}_i$, we write $\mathbf{x}_j \not\to \mathbf{x}_i$. A path Q is a sequence of adjacent vertices, and a directed path follows the direction of the arrows. A vertex $\mathbf{x}_j$ on a path Q is said to be a collider on Q if it has converging arrows on Q (i.e., if the two edges in Q connected to $\mathbf{x}_j$ both point toward it). If there is a directed path from $\mathbf{x}_j$ to $\mathbf{x}_i$, then $\mathbf{x}_i$ is called a descendant of $\mathbf{x}_j$. A directed graph is called a DAG if it does not contain directed paths for which the first and last vertices coincide.

For any three disjoint subsets $\mathcal{A}, \mathcal{B}, \mathcal{C}$ of $\{\mathbf{x}_1, \ldots, \mathbf{x}_b\}$, $\mathcal{A}$ and $\mathcal{B}$ are called d-separated by $\mathcal{C}$ if, for every (undirected) path Q from a vertex in $\mathcal{A}$ to a vertex in $\mathcal{B}$, there is at least one vertex $\mathbf{x}_k \in Q$ that blocks the path in one of the following ways:

B1: $\mathbf{x}_k$ is not a collider on Q and $\mathbf{x}_k$ is in $\mathcal{C}$, or

B2: $\mathbf{x}_k$ is a collider on Q and neither $\mathbf{x}_k$ nor any of its descendants are in $\mathcal{C}$.

If $\mathbf{x}$ follows a multivariate normal distribution (as we assume here), then $\mathcal{A}$ and $\mathcal{B}$ are conditionally independent given $\mathcal{C}$ if and only if they are d-separated by $\mathcal{C}$.

APPENDIX C: SAME CONDITIONING SETS (SCS)

In the SCS approach, every $\mathbf{y}_i$ has the same conditioning vector $\mathbf{y}_1$ of size r_1; that is, $q(i) = (1)$ for all $i > 1$. This is the strategy employed by the MPP and FSA in Sections 3.5–3.6 with $r_i = 1$ and $r_i = r$, respectively, for $i > 1$. For example, one could choose the first r_1 variables in the maxmin ordering, which result in a coarse grid over $\mathcal{D}$.

SCS has several advantages. First, latent Vecchia automatically adheres to the SGV rules for SCS; or, in other words, the sparsity for the latent approach can be guaranteed. Second, as discussed in Section 6.2, if smoothness and range are large enough, no screening effect will hold. SCS is an extension of the predictive process, which tends to work well in such "smooth" situations, because it is equivalent to a Nyström approximation of the leading terms of the Karhunen–Loéve expansion of $y(\cdot)$ (Sang and Huang, 2012). Third, a lower computational complexity can be achieved, because $C(\mathbf{x}_{g(i)}, \mathbf{x}_{g(i)}) = C(\mathbf{y}_1, \mathbf{y}_1)$ in (6) is the same matrix for all $i = 2, \ldots, l$ and its Cholesky decomposition only needs to be computed once. Assuming $r_1 = rm$, the cost of the Cholesky decomposition is $\mathcal{O}((rm)^3)$, and each $\mathbf{B}_i$ and $\mathbf{D}_i$ can be computed in $\mathcal{O}(r_i^3 m^2)$ time, resulting in an overall time complexity for SCS of $\mathcal{O}(nm^2r^2)$ (i.e., reduced by factor m relative to the general case) if $r_i = r$ for $i > 1$. Fourth, the marginal distributions of the $\mathbf{x}_i$ (and hence also the variances) are exact (see Section 3.6). Fifth, $\mathbf{V}^{-1}$ has the same sparsity structure as $\mathbf{V}$, which allows fast calculation of the joint posterior predictive distribution for a large number of prediction locations, and extension to Kalman-filter-type inference for massive spatio-temporal data (Jurek and Katzfuss, 2018). All of these advantages also hold for the MRA, which can be viewed as an iterative SCS approach at multiple resolutions (Katzfuss, 2017, Katzfuss and Gong, 2020, Jurek and Katzfuss, 2018). However, SCS and MRA may require $r_1 = \mathcal{O}(\sqrt{n_z})$ for accurate approximations in two-dimensional space, which results in a time complexity of $\mathcal{O}(n_z^{3/2})$ (Minden et al., 2017).

APPENDIX D: COMPARISON TO COMPOSITE LIKELIHOOD

Using simulated data, we compared maximum likelihood estimation (MLE) using our SGV approach to two composite likelihood methods, full-conditional likelihood (FCL) and pairwise-block likelihood (PBL). The data were simulated from a GP model as in Section 2.1 with exponential covariance, $C(\mathbf{s}_1, \mathbf{s}_2) = \sigma^2 \exp(-\|\mathbf{s}_1 - \mathbf{s}_2\|/\alpha)$, where the process and noise variances were known, $\sigma^2 = 2$ and $\tau^2 = 1$, respectively, and the task was to estimate the unknown range α.

Our first simulation study considered a FCL, defined here as $\widehat{f}(\mathbf{z}) = \prod_{i=1}^n f(z_i|\mathbf{z}_{-i})$. As the FCL is expensive to compute, we considered a relatively small grid of size 30×30 with spacing 1. We simulated 300 datasets with true range $\alpha = 10$, and for each dataset i we computed the MLE $\widehat{\alpha}_{ij}$ using each method $j = 1, \ldots, 5$, namely exact likelihood, FCL, and SGV with $m = 10$, 15, and 20. Table 1(a) contains a summary of the results. We included 95% confidence intervals for the MSEs, based on a normal approximation of the squared errors. We also computed confidence intervals for the difference in MSEs, $(\widehat{\alpha}_{ij} - \alpha)^2 - (\widehat{\alpha}_{iJ} - \alpha)^2$, where J corresponds to SGV with $m = 20$. FCL was not competitive with SGV.

Our second simulation study considered PBLs, $\prod_{i \sim j} f(\mathbf{z}_i, \mathbf{z}_j)$, where each $\mathbf{z}_i$ corresponds to a contiguous rectangle in the spatial domain, and $i \sim j$ means that blocks i and j are spatial neighbors. We used a larger grid of size 100×100 and a larger range parameter $\alpha = 30$. We again simulated 300 datasets and computed the MLE of α using several settings of the PBL and our SGV. The results are given in Table 1(b). Note that even SGV with $m = 20$ performed better than PBL with 100 blocks of size 100 each.

APPENDIX E: ILLUSTRATION OF BAYESIAN INFERENCE

This section demonstrates how one can carry out Bayesian inference using the general Vecchia approximation. We used SGV with $m = 30$ under the same settings as in Appendix D.

First, we simulated a dataset in the setting of the 30×30 grid. Based on the prior $\log \alpha \sim \mathcal{N}(\log(10), 0.6^2)$, we

TABLE 1

Comparison of sparse general Vecchia (SGV) to composite likelihood: Estimation of range parameter from simulated data, including 95% confidence intervals (CIs) for the difference in MSE relative to the method in the last row

(a) 30×30 grid

Method	MSE	95% CI	CI for diff.
Exact lik.	3.21	(2.60, 3.81)	(−0.03, 0.27)
FCL	4.23	(3.48, 4.97)	(0.61, 1.67)
SGV $m = 10$	3.22	(2.63, 3.80)	(−0.15, 0.41)
SGV $m = 15$	3.07	(2.50, 3.63)	(−0.20, 0.15)
SGV $m = 20$	3.09	(2.53, 3.64)	(0.00, 0.00)

(b) 100×100 grid

Method	MSE	95% CI	CI for diff.
PBL 100 bl.	5.71	(4.79, 6.62)	(0.65, 1.92)
PBL 144 bl.	6.26	(5.22, 7.30)	(1.05, 2.63)
PBL 225 bl.	7.05	(5.73, 8.37)	(1.64, 3.61)
SGV $m = 20$	4.81	(4.07, 5.54)	(0.03, 0.74)
SGV $m = 40$	4.42	(3.65, 5.19)	(0.00, 0.00)

evaluated the exact posterior $f(\alpha|\mathbf{z}_o)$ and the posterior $\widehat{f}(\alpha|\mathbf{z}_o)$ implied by the SGV likelihood on a fine grid (see Figure 7(a)). Based on this discrete approximation to the posterior, Figure 7(b) shows the posterior predictive distribution $f(y(\mathbf{s}_*)|\mathbf{z}_o)$ at the unobserved point $\mathbf{s}_* = (15.5, 15.5)$ in the center of the grid, along with the distribution $\widehat{f}(y(\mathbf{s}_0)|\mathbf{z}_o)$ approximated using a general Vecchia prediction method called RF-full in Katzfuss et al. (2020). The Vecchia posteriors were almost identical to the exact distributions.

We also simulated a dataset in the setting of the 100×100 grid, randomly selecting $n_z = 9000$ data points as observed, with the remaining GP realizations used as test data. Based on the prior $\log\alpha \sim \mathcal{N}(\log(30), 0.6^2)$ and the SGV likelihood, we ran a Metropolis–Hastings sampler for $\log\alpha$ with a normal proposal distribution with standard deviation 0.5 for 1200 iterations, discarding the first 200 samples and thinning the remaining by a factor of 10. We then computed posterior predictive distributions $\widehat{f}(\mathbf{y}_{-o}|\mathbf{z}_o)$ using RF-full for the 1000 held-out test locations. Figure 7(c) shows the resulting posterior 80% intervals along with the true simulated values of $\mathbf{y}_{-o}$ at the test locations. 79.8% of the intervals covered the true values, indicating that the posterior predictive distributions obtained using general Vecchia were well calibrated.

APPENDIX F: PROOFS

In this section, we provide proofs for the propositions stated throughout the article.

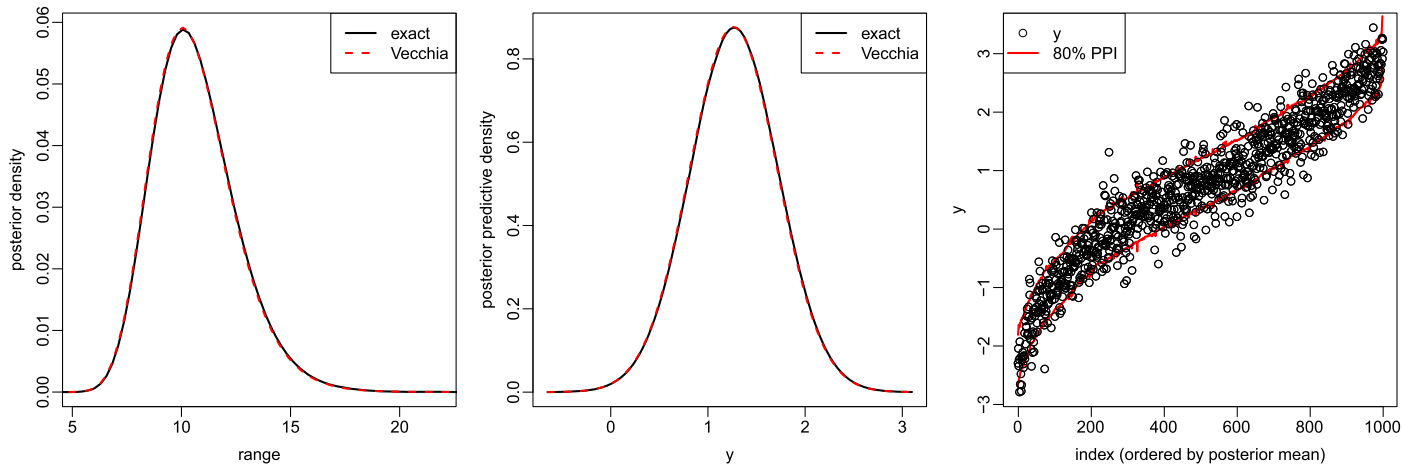
PROOF OF PROPOSITION 1. For the density in (6), we have $\widehat{f}(\mathbf{x}) \propto \exp(-w/2)$, where $w = \sum_{i=1}^{b} \mathbf{a}_i'\mathbf{a}_i$ and

$$\mathbf{a}_i = (\mathbf{D}_i^{-1/2})'(\mathbf{x}_i - \mathbf{B}_i\mathbf{x}_{g(i)})$$

$$= (\mathbf{D}_i^{-1/2})'\mathbf{x}_i + \sum_{j\in g(i)} (-(\mathbf{D}_i^{-1/2})'\mathbf{B}_i^j\mathbf{x}_j) = \sum_{j=1}^{b} \mathbf{U}_{ji}'\mathbf{x}_j,$$

with $\mathbf{U}$ defined as in (7). Hence, $w = \sum_{i=1}^{b}(\mathbf{U}_i'\mathbf{x})'(\mathbf{U}_i'\mathbf{x}) = \mathbf{x}'\mathbf{U}\mathbf{U}'\mathbf{x}$, where $\mathbf{U}_i$ is the ith block of columns in $\mathbf{U}$. Because $\mathbf{U}$ is a nonsingular matrix, we have $\widehat{f}(\mathbf{x}) = \mathcal{N}_n(\mathbf{x}|\mathbf{0}, \widehat{\mathbf{C}})$ with $\widehat{\mathbf{C}}^{-1} = \mathbf{U}\mathbf{U}'$, which proves the first part of the proposition. Note that a proof for a similar expression of the approximate joint density can be found in Datta et al. (2016a), App. A2.

Then, because $\mathbf{P}$ is a symmetric matrix, we have $\mathbf{P} = \mathbf{P}' = \mathbf{P}^{-1}$ (and $\mathbf{PMP}$ results in reverse row-column ordering of the square matrix $\mathbf{M}$). Thus, we can write $\mathbf{P}\widehat{\mathbf{C}}^{-1}\mathbf{P} =$



(a) $f(\alpha|\mathbf{z}_o)$ for 30×30 grid (b) $f(y(\mathbf{s}_*)|\mathbf{z}_o)$ for 30×30 grid (c) PPIs for 100×100 grid

FIG. 7. *Results for Bayesian inference based on general Vecchia. PPI: posterior predictive interval.*

$\mathbf{PUU}'\mathbf{P} = \mathbf{PUPPU}'\mathbf{P} = (\mathbf{PUP})(\mathbf{PUP})'$. The matrix $\mathbf{PUP}$ is lower triangular with positive values on the diagonal, and so it must be the Cholesky factor of $\mathbf{P}\widehat{\mathbf{C}}^{-1}\mathbf{P}'$ since the Cholesky factor is the unique such lower triangular matrix. Therefore, we have $\mathbf{U} = \mathbf{P}\,\mathrm{chol}(\mathbf{P}\widehat{\mathbf{C}}^{-1}\mathbf{P})\mathbf{P} = \mathrm{rchol}(\widehat{\mathbf{C}}^{-1})$. □

PROOF OF PROPOSITION 2. By rearranging the definition of a conditional density, we obtain $\widehat{f}(\mathbf{z}_o) = \widehat{f}(\mathbf{x})/\widehat{f}(\mathbf{y}|\mathbf{z}_o)$, which holds for any $\mathbf{y}$, and so we simply set $\mathbf{y} = \mathbf{0}$. Letting $\mathbf{x}_0$ be $\mathbf{x}$ with $\mathbf{y} = \mathbf{0}$, we have

$$\widehat{f}(\mathbf{z}_o) = \frac{\mathcal{N}(\mathbf{x}_0|\mathbf{0}, \widehat{\mathbf{C}})}{\mathcal{N}(\mathbf{0}|\boldsymbol{\mu}, \mathbf{W}^{-1})}, \tag{9}$$

where $\boldsymbol{\mu} := \mathrm{E}(\mathbf{y}|\mathbf{z}_o)$ and $\mathbf{W} := \mathrm{var}(\mathbf{y}|\mathbf{z}_o)^{-1}$. For the numerator in (9), using the factorization $\widehat{\mathbf{C}}^{-1} = \mathbf{UU}'$ supplied by the Vecchia approach, we have $\log|\widehat{\mathbf{C}}| = -2\log|\mathbf{U}| = \sum_{i=1}^{b}\log|\mathbf{D}_i|$ and $\mathbf{x}_0'\widehat{\mathbf{C}}^{-1}\mathbf{x}_0 = (\mathbf{U}'\mathbf{x}_0)' \times (\mathbf{U}'\mathbf{x}_0) = \tilde{\mathbf{z}}'\tilde{\mathbf{z}}$.

For the denominator in (9), according to Theorem 12.2 in Rue and Held (2010), we have $\mathbf{W} = \mathbf{U}_Y\mathbf{U}_Y'$ and $\boldsymbol{\mu} = -\mathbf{W}^{-1}\mathbf{U}_Y\mathbf{U}_Z'\mathbf{z}_o$. Because $\mathbf{W} = \mathbf{VV}'$ and $\mathbf{V}$ is upper triangular, we have $\log|\mathbf{W}^{-1}| = -2\log|\mathbf{V}| = -2\sum_i \log|\mathbf{V}_{ii}|$. The quadratic form can be obtained as $\boldsymbol{\mu}'\mathbf{W}\boldsymbol{\mu} = \tilde{\mathbf{z}}'\mathbf{U}_Y' \times \mathbf{W}^{-1}\mathbf{W}\mathbf{W}^{-1}\mathbf{U}_Y\tilde{\mathbf{z}} = (\mathbf{V}^{-1}\mathbf{U}_Y\tilde{\mathbf{z}})'(\mathbf{V}^{-1}\mathbf{U}_Y\tilde{\mathbf{z}})$. □

PROOF OF PROPOSITION 3. It can be easily verified that $\mathrm{rchol}(\mathbf{A}) = \mathbf{P}(\mathrm{chol}(\mathbf{PAP}))\mathbf{P}$ is an upper-triangular matrix for any symmetric, positive-definite $\mathbf{A}$, because $\mathrm{chol}(\cdot)$ was defined to return the lower-triangular Cholesky factor. Hence, both $\mathbf{U} = \mathrm{rchol}(\mathbf{C}^{-1})$ and $\mathbf{V} = \mathrm{rchol}(\mathbf{W})$ are upper triangular. Further, $\mathbf{W} = \mathbf{U}_Y\mathbf{U}_Y'$ is symmetric. Therefore, we only consider the case $j < i$ in the remainder of this proof.

It is well known for precision matrices in multivariate normal distributions (see, e.g., Rue and Held, 2010, Thm. 12.1) that $\mathbf{W}_{ji} = \mathbf{0}$ if $\mathbf{y}_i$ and $\mathbf{y}_j$ are conditionally independent given all other variables in the model (i.e., conditional on $\mathcal{C}_W = \{\mathbf{y}_{-ij}, \mathbf{z}_o\}$). A similar result for the sparsity of the Cholesky factor (see, e.g., Rue and Held, 2010, Thm. 12.5) can be rephrased for our reverse Cholesky decomposition ($\mathbf{U} = \mathrm{rchol}(\widehat{\mathbf{C}}^{-1})$) to say that $\mathbf{U}_{ji} = \mathbf{0}$ if $\mathbf{x}_j$ and $\mathbf{x}_i$ are conditionally independent given $\mathcal{C}_U = \{\mathbf{x}_{h(i)\setminus j}\}$. For $\mathbf{V}$, which is the Cholesky factor of the posterior precision matrix $\mathbf{W}$ (i.e., conditional on $\mathbf{z}_o$), we have $\mathbf{V}_{ji} = \mathbf{0}$ if $\mathbf{y}_i$ and $\mathbf{y}_j$ are conditionally independent given $\mathcal{C}_V = \{\mathbf{y}_{h(i)\setminus j}, \mathbf{z}_o\}$. Thus, $\mathbf{U}_{ji} = \mathbf{0}$ if and only if $\mathbf{x}_i$ and $\mathbf{x}_j$ are d-separated by $\mathcal{C}_U$ in the DAG, and $\mathbf{W}_{ji} = \mathbf{0}$ and $\mathbf{V}_{ji} = \mathbf{0}$ if and only if $\mathbf{y}_i$ and $\mathbf{y}_j$ are d-separated by $\mathcal{C}_W$ and $\mathcal{C}_V$, respectively.

Note that d-separation cannot hold if $\mathbf{x}_j \to \mathbf{x}_i$, and so we only consider (paths between) nonadjacent $\mathbf{x}_j$ and $\mathbf{x}_i$ in the remainder of the proof. Any path between such $\mathbf{x}_j$ and $\mathbf{x}_i$ must pass through at least one noncollider $\mathbf{x}_l$ with $l < i$, or through a collider $\mathbf{y}_k$ with $2k - 1 > i$, because arrows in the Vecchia approach can only go forward in the ordering and the only parent for each $\mathbf{z}_k$ is $\mathbf{y}_k$. As we have $\mathcal{C}_U = \{\mathbf{x}_l : l < i, l \neq j\}$, this means that any path between (nonadjacent) $\mathbf{x}_i$ and $\mathbf{x}_j$ is either blocked by a noncollider $\mathbf{x}_l \in \mathcal{C}_U$ (condition B1) or by a collider $\mathbf{x}_k \notin \mathcal{C}_U$ (B2), which proves part 1 of the proposition.

For $\mathbf{W}$, as all vertices other than $\mathbf{y}_i$ and $\mathbf{y}_j$ are in $\mathcal{C}_W$, we only need to consider condition B1. The only paths between $\mathbf{y}_j$ and $\mathbf{y}_i$ that do not contain a vertex $\mathbf{x}_k \in \mathcal{C}_W$ that is not a collider (and hence blocks the path), are paths of the form $(\mathbf{y}_j, \mathbf{y}_k, \mathbf{y}_i)$ with $\mathbf{y}_j \to \mathbf{y}_k$ and $\mathbf{y}_i \to \mathbf{y}_k$. This proves part 2.

For $\mathbf{V}$, any path between $\mathbf{y}_i$ and $\mathbf{y}_j$ that passes through $\mathcal{C}_V = \{\mathbf{y}_{h(i)\setminus j}, \mathbf{z}_o\}$ includes a noncollider in $\mathcal{C}_V$ and is thus blocked (B1). Thus, $\mathbf{V}_{ji}$ can only be nonzero if there is a path between $\mathbf{y}_i$ and $\mathbf{y}_j$ on the subgraph $\{\mathbf{y}_i, \mathbf{y}_j\} \cup \{\mathbf{y}_k : k > i, \mathbf{y}_k \text{ has at least one observed descendant}\}$. □

PROOF OF PROPOSITION 4. Note that for any $p(i) \subset h(i)$, we have $f(\mathbf{y}_i|\mathbf{y}_{p(i)}) = f(\mathbf{y}_i|\mathbf{y}_{p(i)}, \mathbf{z}_{p(i)})$, due to conditional independence between $\mathbf{z}_k$ and any other variable in the model given $\mathbf{y}_k$. Thus, for latent Vecchia we can change the conditioning vector of $\mathbf{y}_i$ in (4) from $\mathbf{y}_{q(i)}$ to $(\mathbf{y}_{q(i)}, \mathbf{z}_{q(i)})$ without changing the approximation. Likewise, for SGV we can change the conditioning vector of $\mathbf{y}_i$ from $(\mathbf{y}_{q_y(i)}, \mathbf{z}_{q_z(i)})$ to $(\mathbf{y}_{q_y(i)}, \mathbf{z}_{q(i)})$ without changing the approximation, since $q_y(i) \cup q_z(i) = q(i)$. Further, note that $\mathbf{z}_{q(i)}$ is a subset of $(\mathbf{y}_{q_y(i)}, \mathbf{z}_{q(i)})$, which is in turn a subset of $(\mathbf{y}_{q(i)}, \mathbf{z}_{q(i)})$. Thus, the proposition follows using Thm. 1 in Guinness (2018), which says that adding variables to the conditioning vector in Vecchia approximations cannot increase the KL divergence from the true model. □

PROOF OF PROPOSITION 5. Without loss of generality, we assume that the locations lie on a regular unit-distance grid on the d-dimensional hypercube with $n_z^{1/d}$ unique values in each dimension, and we assume a lexicographic ordering in which locations are ordered first by their first coordinate, for those with same first coordinate by their second coordinate, and so forth. Let $\mathbf{s}_i = (s_{i1}, \ldots, s_{id})$ be the location of $\mathbf{y}_i$ (and $\mathbf{z}_i$). Consider a pair of locations $\mathbf{s}_i$ and $\mathbf{s}_j$ with $\mathbf{s}_j = (s_{i1} - t, s_{j2}, \ldots, s_{jd})$, which under lexicographic ordering gives $j < i$ when $t > 0$. For $\mathbf{s}_a = (s_{i1} + 1, s_{j2}, \ldots, s_{jd})$, we have $a > i$. Further, ignoring edge cases, we have $\mathbf{y}_j \to \mathbf{y}_a$ when $1 \le t \le t_m$, where $t_m = \mathcal{O}(m^{1/d})$, since the conditioning vector of $\mathbf{y}_a$ corresponds to the m locations roughly in a semi-ball around $\mathbf{s}_a$ of radius t_m (e.g., $t_m = (2m/\pi)^{1/2}$ for $d = 2$). Also consider $\mathbf{s}_p = (s_{i1} + 1, s_{i2}, \ldots, s_{id})$, for which also $p > i$. We can find a path between $\mathbf{y}_a$ and $\mathbf{y}_p$ on the subgraph $\{\mathbf{y}_k : k > i\}$, since all variables on the hyperplane $(s_{i1} + 1, \cdot, \ldots, \cdot)$ are connected (if $m \ge d$) and have index greater than i. We also have $\mathbf{y}_i \to \mathbf{y}_p$. Therefore, there is a path from $\mathbf{y}_j$ to $\mathbf{y}_i$ on the subgraph $\{\mathbf{y}_j, \mathbf{y}_i\} \cup \{\mathbf{y}_k :$

$k > i\}$, which by Proposition 3 means that $\mathbf{V}_{ji}$ is nonzero. Since this is true for any $\mathbf{s}_j = \{s_{i1} - t, s_{j2}, \ldots, s_{jd}\}$ with $1 \le t \le t_m$, there are $\mathcal{O}(n^{1-1/d}m^{1/d})$ nonzero elements in each of the n_z columns of $\mathbf{V}$, giving a memory complexity of $\mathcal{O}(n^{2-1/d}m^{1/d})$. As the time complexity for obtaining the reordered Cholesky factor $\mathbf{V}$ is on the order of the sum of the squares of the number of nonzero elements per column in $\mathbf{V}$ (see, e.g., Toledo, 2007, Thm. 2.2), the time complexity of obtaining $\mathbf{V}$ from $\mathbf{W}$ is $\mathcal{O}(n^{3-2/d}m^{2/d})$. □

PROOF OF PROPOSITION 6. First, we show that, for SGV, $\mathbf{V}$ has at most mr off-diagonal nonzero elements per column. Using Proposition 3, that means that we need to show that, for any $j < i$, there is no path between $\mathbf{y}_i$ and $\mathbf{y}_j$ on the subgraph $\mathcal{G}_{ij}^{\ell}$ if $\mathbf{y}_j \not\to \mathbf{y}_i$, where $\mathcal{G}_{ij}^{k} := \{\mathbf{y}_i, \mathbf{y}_j\} \cup \{\mathbf{y}_t : \max(i, j) < t \le k\}$. (Note that this statement then also holds if we restrict the subgraph to vertices with observed descendants.) Define $D_i^k := \{t : \mathbf{y}_t$ is a descendant of $\mathbf{y}_i$ in $\mathcal{G}_{ij}^{k}\}$, and analogously for D_j^k. Thus, assuming that $\mathbf{y}_j \not\to \mathbf{y}_i$, we need to show that $D_i^{\ell} \cap D_j^{\ell} = \varnothing$, which we will do by induction. We have $\mathcal{G}_{ij}^{i+1} = \{\mathbf{y}_i, \mathbf{y}_j, \mathbf{y}_k\}$, where $q_y(k)$ can only contain either i or j by the rules of the SGV, because $j \notin q_y(i)$, and so $D_i^{i+1} \cap D_j^{i+1} = \varnothing$. Now, assume that $D_i^k \cap D_j^k = \varnothing$ for $k > i$. Then, for any $t_i \in D_i^k$ and $t_j \in D_j^k$, $\mathbf{y}_{t_i}$ and $\mathbf{y}_{t_j}$ cannot be adjacent. Hence, by the rules of the SGV, $q_y(k+1)$ can only contain either elements of D_i^k or of D_j^k, and so $D_i^{k+1} \cap D_j^{k+1} = \varnothing$. In summary, for the SGV and $j < i$, $\mathbf{V}_{ji} = \mathbf{0}$ unless $\mathbf{y}_j \to \mathbf{y}_i$, and so $\mathbf{V}$ has at most mr off-diagonal elements per column.

The time complexity for obtaining the reordered Cholesky factor $\mathbf{V}$ (and the selected inverse of $\mathbf{W}$) is on the order of the sum of the squares of the number of nonzero elements per column in $\mathbf{V}$ (e.g., Toledo, 2007, Thm. 2.2). Hence, the time complexity for computing $\mathbf{W}$, its decomposition, and its selected inverse is $\mathcal{O}(nm^2r^2)$. The time and memory complexity for computing $\mathbf{U}$ is $\mathcal{O}(nm^3r^2)$ and $\mathcal{O}(nmr)$ (i.e., at least as high as that for computing $\mathbf{V}$), and so SGV has the same computational complexity as standard Vecchia. □

ACKNOWLEDGMENTS

Katzfuss' research was supported in part by National Science Foundation (NSF) Grant DMS-1521676 and NSF CAREER Grant DMS-1654083. Guinness' research was supported in part by NSF Grant DMS-1613219 and NIH Grant R01ES027892. We also acknowledge support from the NSF Research Network for Statistical Methods for Atmospheric and Oceanic Sciences, No. 1107046. Wenlong Gong, Marcin Jurek, and Daniel Zilber contributed to the R package `GPvecchia`. We would like to thank the reviewers for helpful comments and suggestions.

REFERENCES

BANERJEE, S., CARLIN, B. P. and GELFAND, A. E. (2015). *Hierarchical Modeling and Analysis for Spatial Data*, 2nd ed. *Monographs on Statistics and Applied Probability* **135**. CRC Press, Boca Raton, FL. MR3362184

BANERJEE, S., GELFAND, A. E., FINLEY, A. O. and SANG, H. (2008). Gaussian predictive process models for large spatial data sets. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **70** 825–848. MR2523906 https://doi.org/10.1111/j.1467-9868.2008.00663.x

CRESSIE, N. and DAVIDSON, J. L. (1998). Image analysis with partially ordered Markov models. *Comput. Statist. Data Anal.* **29** 1–26. MR1665674 https://doi.org/10.1016/S0167-9473(98)00052-8

CRESSIE, N. and JOHANNESSON, G. (2008). Fixed rank kriging for very large spatial data sets. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **70** 209–226. MR2412639 https://doi.org/10.1111/j.1467-9868.2007.00633.x

CRESSIE, N. and WIKLE, C. K. (2011). *Statistics for Spatio-Temporal Data*. *Wiley Series in Probability and Statistics*. Wiley, Hoboken, NJ. MR2848400

DATTA, A., BANERJEE, S., FINLEY, A. O. and GELFAND, A. E. (2016a). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *J. Amer. Statist. Assoc.* **111** 800–812. MR3538706 https://doi.org/10.1080/01621459.2015.1044091

DATTA, A., BANERJEE, S., FINLEY, A. O. and GELFAND, A. E. (2016b). On nearest-neighbor Gaussian process models for massive spatial data. *Wiley Interdiscip. Rev.: Comput. Stat.* **8** 162–171. MR3544254 https://doi.org/10.1002/wics.1383

DATTA, A., BANERJEE, S., FINLEY, A. O., HAMM, N. A. S. and SCHAAP, M. (2016c). Nonseparable dynamic nearest neighbor Gaussian process models for large spatio-temporal data with an application to particulate matter analysis. *Ann. Appl. Stat.* **10** 1286–1316. MR3553225 https://doi.org/10.1214/16-AOAS931

DU, J., ZHANG, H. and MANDREKAR, V. S. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *Ann. Statist.* **37** 3330–3361. MR2549562 https://doi.org/10.1214/08-AOS676

EIDSVIK, J., SHABY, B. A., REICH, B. J., WHEELER, M. and NIEMI, J. (2014). Estimation and prediction in spatial models with block composite likelihoods. *J. Comput. Graph. Statist.* **23** 295–315. MR3215812 https://doi.org/10.1080/10618600.2012.760460

EUBANK, R. L. and WANG, S. (2002). The equivalence between the Cholesky decomposition and the Kalman filter. *Amer. Statist.* **56** 39–43. MR1939395 https://doi.org/10.1198/000313002753631349

FINLEY, A. O., SANG, H., BANERJEE, S. and GELFAND, A. E. (2009). Improving the performance of predictive process modeling for large datasets. *Comput. Statist. Data Anal.* **53** 2873–2884. MR2667597 https://doi.org/10.1016/j.csda.2008.09.008

FINLEY, A. O., DATTA, A., COOK, B. C., MORTON, D. C., ANDERSEN, H. E. and BANERJEE, S. (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *J. Comput. Graph. Statist.* **28** 401–414.

FURRER, R., GENTON, M. G. and NYCHKA, D. (2006). Covariance tapering for interpolation of large spatial datasets. *J. Comput. Graph. Statist.* **15** 502–523. MR2291261 https://doi.org/10.1198/106186006X132178

FURRER, R. and SAIN, S. R. (2010). spam: A sparse matrix R package with emphasis on MCMC methods for Gaussian Markov random fields. *J. Stat. Softw.* **36** 1–25.

GERBER, F., FURRER, R., SCHAEPMAN-STRUB, G., DE JONG, R. and SCHAEPMAN, M. E. (2018). Predicting missing values in spatio-temporal satellite data. *IEEE Trans. Geosci. Remote Sens.* **56** 2841–2853.

GRAMACY, R. B. and APLEY, D. W. (2015). Local Gaussian process approximation for large computer experiments. *J. Comput. Graph. Statist.* **24** 561–578. MR3357395 https://doi.org/10.1080/10618600.2014.914442

GRAMACY, R. B. and LEE, H. K. H. (2012). Cases for the nugget in modeling computer experiments. *Stat. Comput.* **22** 713–722. MR2909617 https://doi.org/10.1007/s11222-010-9224-x

GUHANIYOGI, R. and BANERJEE, S. (2018). Meta-kriging: Scalable Bayesian modeling and inference for massive spatial datasets. *Technometrics* **60** 430–444. MR3878099 https://doi.org/10.1080/00401706.2018.1437474

GUINNESS, J. (2018). Permutation and grouping methods for sharpening Gaussian process approximations. *Technometrics* **60** 415–429. MR3878098 https://doi.org/10.1080/00401706.2018.1437476

HEATON, M. J., DATTA, A., FINLEY, A. O., FURRER, R., GUINNESS, J., GUHANIYOGI, R., GERBER, F., GRAMACY, R. B., HAMMERLING, D. et al. (2019). A case study competition among methods for analyzing large spatial data. *J. Agric. Biol. Environ. Stat.* **24** 398–425.

HIGDON, D. (1998). A process-convolution approach to modelling temperatures in the North Atlantic Ocean. *Environ. Ecol. Stat.* **5** 173–190.

HUANG, H. and SUN, Y. (2018). Hierarchical low rank approximation of likelihoods for large spatial datasets. *J. Comput. Graph. Statist.* **27** 110–118. MR3788305 https://doi.org/10.1080/10618600.2017.1356324

JUREK, M. and KATZFUSS, M. (2018). Multi-resolution filters for massive spatio-temporal data. Available at arXiv:1810.04200.

KALMAN, R. E. (1960). A new approach to linear filtering and prediction problems. *Trans. ASME Ser. D. J. Basic Eng.* **82** 35–45. MR3931993

KATZFUSS, M. (2017). A multi-resolution approximation for massive spatial datasets. *J. Amer. Statist. Assoc.* **112** 201–214. MR3646566 https://doi.org/10.1080/01621459.2015.1123632

KATZFUSS, M. and CRESSIE, N. (2011). Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets. *J. Time Series Anal.* **32** 430–446. MR2841794 https://doi.org/10.1111/j.1467-9892.2011.00732.x

KATZFUSS, M. and GONG, W. (2020). A class of multi-resolution approximations for large spatial datasets. *Statist. Sinica.* **30** 2203–2226.

KATZFUSS, M., GUINNESS, J., GONG, W. and ZILBER, D. (2020). Vecchia approximations of Gaussian-process predictions. *J. Agric. Biol. Environ. Stat.*. **25** 383–414.

KAUFMAN, C. G., SCHERVISH, M. J. and NYCHKA, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *J. Amer. Statist. Assoc.* **103** 1545–1555. MR2504203 https://doi.org/10.1198/016214508000000959

LAURITZEN, S. L. (1996). *Graphical Models. Oxford Statistical Science Series* **17**. The Clarendon Press, Oxford University Press, New York. MR1419991

LINDGREN, F., RUE, H. and LINDSTRÖM, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **73** 423–498. MR2853727 https://doi.org/10.1111/j.1467-9868.2011.00777.x

MINDEN, V., DAMLE, A., HO, K. L. and YING, L. (2017). Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations. *Multiscale Model. Simul.* **15** 1584–1611. MR3720352 https://doi.org/10.1137/17M1116477

NYCHKA, D., BANDYOPADHYAY, S., HAMMERLING, D., LINDGREN, F. and SAIN, S. (2015). A multiresolution Gaussian process model for the analysis of large spatial datasets. *J. Comput. Graph. Statist.* **24** 579–599. MR3357396 https://doi.org/10.1080/10618600.2014.914946

QUIÑONERO-CANDELA, J. and RASMUSSEN, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **6** 1939–1959. MR2249877

RASMUSSEN, C. E. and WILLIAMS, C. K. I. (2006). *Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR2514435

RAUCH, H. E., TUNG, F. and STRIEBEL, C. T. (1965). Maximum likelihood estimates of linear dynamic systems. *AIAA J.* **3** 1445–1450. MR0181489 https://doi.org/10.2514/3.3166

RUE, H. and HELD, L. (2005). *Gaussian Markov Random Fields: Theory and Applications. Monographs on Statistics and Applied Probability* **104**. CRC Press/CRC, Boca Raton, FL. MR2130347 https://doi.org/10.1201/9780203492024

RUE, H. and HELD, L. (2010). Discrete spatial variation. In *Handbook of Spatial Statistics. Chapman & Hall/CRC Handb. Mod. Stat. Methods* 171–200. CRC Press, Boca Raton, FL. MR2730942 https://doi.org/10.1201/9781420072884-c12

RÜTIMANN, P. and BÜHLMANN, P. (2009). High dimensional sparse covariance estimation via directed acyclic graphs. *Electron. J. Stat.* **3** 1133–1160. MR2566184 https://doi.org/10.1214/09-EJS534

SANG, H. and HUANG, J. Z. (2012). A full scale approximation of covariance functions for large spatial data sets. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **74** 111–132. MR2885842 https://doi.org/10.1111/j.1467-9868.2011.01007.x

SANG, H., JUN, M. and HUANG, J. Z. (2011). Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors. *Ann. Appl. Stat.* **5** 2519–2548. MR2907125 https://doi.org/10.1214/11-AOAS478

SHABY, B. A. (2014). The open-faced sandwich adjustment for MCMC using estimating functions. *J. Comput. Graph. Statist.* **23** 853–876. MR3224659 https://doi.org/10.1080/10618600.2013.842174

SNELSON, E. and GHAHRAMANI, Z. (2007). Local and global sparse Gaussian process approximations. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics* (*AISTATS*) 524–531.

STEIN, M. L. (2011). 2010 Rietz Lecture: When does the screening effect hold? *Ann. Statist.* **39** 2795–2819. MR3012392 https://doi.org/10.1214/11-AOS909

STEIN, M. L. (2014). Limitations on low rank approximations for covariance matrices of spatial data. *Spat. Stat.* **8** 1–19. MR3326818 https://doi.org/10.1016/j.spasta.2013.06.003

STEIN, M. L., CHI, Z. and WELTY, L. J. (2004). Approximating likelihoods for large spatial data sets. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **66** 275–296. MR2062376 https://doi.org/10.1046/j.1369-7412.2003.05512.x

SUN, Y. and STEIN, M. L. (2016). Statistically and computationally efficient estimating equations for large spatial datasets. *J. Comput. Graph. Statist.* **25** 187–208. MR3474043 https://doi.org/10.1080/10618600.2014.975230

TOLEDO, S. (2007). Lecture notes on combinatorial preconditioners, Chapter 3. Available at http://www.tau.ac.il/~stoledo/Support/chapter-direct.pdf.

VARIN, C., REID, N. and FIRTH, D. (2011). An overview of composite likelihood methods. *Statist. Sinica* **21** 5–42. MR2796852

VECCHIA, A. V. (1988). Estimation and model identification for continuous spatial processes. *J. Roy. Statist. Soc. Ser. B* **50** 297–312. MR0964183

WIKLE, C. K. and CRESSIE, N. (1999). A dimension-reduced approach to space-time Kalman filtering. *Biometrika* **86** 815–829. MR1741979 https://doi.org/10.1093/biomet/86.4.815

ZHANG, H. (2012). Asymptotics and computation for spatial statistics. In *Advances and Challenges in Space-Time Modelling of Natural Events* 239–252. Springer, Berlin.

ZHANG, B., SANG, H. and HUANG, J. Z. (2019). Smoothed full-scale approximation of Gaussian process models for computation of large spatial data sets. *Statist. Sinica* **29** 1711–1737. MR3970332