Received: 1 December 2020 | Revised: 7 June 2021 | Accepted: 18 August 2021

DOI: 10.1111/sjos.12555

ORIGINAL ARTICLE

Scandinavian Journal of Statistics

Large spatial data modeling and analysis: A Krylov subspace approach

Jialuo Liu[1] | **Tingjin Chu**[2] | **Jun Zhu**[3] | **Haonan Wang**[1]

[1]Department of Statistics, Colorado State University, Fort Collins, Colorado, USA

[2]School of Mathematics and Statistics, University of Melbourne, Melbourne, Victoria, Australia

[3]Department of Statistics, University of Wisconsin-Madison, Madison, Wisconsin, USA

Correspondence
Tingjin Chu, School of Mathematics and Statistics, University of Melbourne, Melbourne, VIC, Australia.
Email: tingjin.chu@unimelb.edu.au

Funding information
National Science Foundation, Grant/Award Number: DMS0-1737795, DMS-1923142 and CNS-1932413; U.S. Geological Survey

Abstract

Estimating the parameters of spatial models for large spatial datasets can be computationally challenging, as it involves repeated evaluation of sizable spatial covariance matrices. In this paper, we aim to develop Krylov subspace-based methods that are computationally efficient for large spatial data. Specifically, we approximate the inverse and the log-determinant of the spatial covariance matrix in the log-likelihood function via conjugate gradient and stochastic Lanczos on a Krylov subspace. These methods reduce the computational complexity from $O(N^3)$ to $O(N^2 \log N)$ and $O(N \log N)$ for dense and sparse matrices, respectively. Moreover, we quantify the difference between the approximated log-likelihood function and the original log-likelihood function and establish the consistency of parameter estimates. Simulation studies are conducted to examine the computational efficiency as well as the finite-sample properties. For illustration, our methodology is applied to analyze a large dataset comprising LiDAR estimates of forest canopy height in western Alaska.

KEYWORDS

computational efficiency, conjugate gradient, maximum likelihood estimation, spatial linear models, spatial statistics

1 | INTRODUCTION

Technological advances in geographical data acquisition have led to a rapidly increasing abundance of very large spatial datasets, for which geostatistical modeling and analysis could be applied in principle (see, e.g., Cressie, 1993; Stein, 1999). For spatial linear models, maximum likelihood estimation is often used, since the estimates are consistent and asymptotically normal (Mardia & Marshall, 1984). However, evaluation of the likelihood function involves the inverse and determinant of large spatial covariance matrices which usually require $O(N^3)$ flops and $O(N^2)$ memory, where N is the number of observations (i.e., sample size). For a large sample size N, evaluation of the likelihood function becomes time-consuming, if not infeasible, which necessitates the development of new statistical methods. The purpose of this paper is to develop computationally efficient methods based on Krylov subspace for analyzing large spatial data and establish the asymptotic properties of our proposed methods.

To address the computational challenges of large spatial data, various approaches have been proposed (Bradley et al., 2016; Heaton et al., 2019; Sun et al., 2012). One way to reduce the computational burden is to impose a low-rank structure. Low-rank models approximate Gaussian processes on a lower dimensional subspace by, for example, predictive process models (Banerjee et al., 2008; Finley et al., 2009), fixed-rank kriging (Cressie et al., 2010; Cressie & Johannesson, 2008), or multiresolution approximations (Katzfuss, 2017; Nychka et al., 2015). Mostly such methods have been shown to be linear time $O(N)$; however, it remains unclear whether the resulting parameter estimates are consistent.

Sparse methods, on the other hand, enforce sparsity on either the precision matrices or the covariance matrices. Gaussian Markov random fields are also used to approximate spatial Gaussian fields, which yields sparse precision matrices and thus is fast to compute (Rue et al., 2009; Rue & Held, 2005). An explicit link between Gaussian fields and Gaussian Markov random fields is established through the stochastic partial differential equations (SPDE). That is, many covariance functions including the Matérn class are the solution of SPDE and Gaussian Markov random-field approximation represents an approximate stochastic weak solution to the same differential equations (Lindgren et al., 2011). Recently, Bolin and Kirchner (2020) proposed a rational SPDE approach for Gaussian random fields with both stationary and nonstationary covariance functions. Moreover, the rational SPDE approach is able to estimate the smoothness parameter of the Matérn class and does not require the sparsity of the covariance matrices or the precision matrices (Bolin & Kirchner, 2020; Herrmann et al., 2020).

One popular way to achieve the sparsity of the covariance matrices is to approximate the likelihood function by a product of lower dimensional conditional distributions based only on the nearest neighbors (Stein et al., 2004; Vecchia, 1988). It is well-known that the performance of the Vecchina's approximation depends on the ordering of the observations. Taking a Bayesian approach, Datta et al. (2016) generalized the Vecchina's approach and proposed a scalable nearest neighbor Gaussian process model. Furthermore, Guinness (2018) proposed random-ordering and maximum minimum distance ordering, which can improve the approximation of the Vecchina's approximation. An alternative approach is covariance tapering that sets the covariance at two sufficiently distant locations to be zero (Chu et al., 2011; Du et al., 2009; Furrer et al., 2006; Kaufman et al., 2008). Thus, the resulting covariance matrix is sparse and faster to compute. However, the evaluation of the exact tapered likelihood relies on sparse Cholesky factorization, which generally needs to permute the rows and columns of the matrix, which can still be computationally cumbersome.

In general, for evaluating the log-likelihood function, the computational difficulties mainly arise from frequent operations on large covariance matrices, including the *inversion* and the *log determinant*. This computational challenge also exists in many other research fields, such as the Gaussian process. Utilizing the hierarchical matrix technique, the likelihood approximation achieves a log-linear computational cost and storage (Litvinenko et al., 2019; Minden et al., 2017). Moreover, hierarchical matrix approximation is paralleled in HLIBCov package (Litvinenko et al., 2020). Another popular method is to utilize the Krylov subspace and has received increasing attention in the Gaussian process (Anitescu et al., 2012; Cunningham et al., 2008; Dong et al., 2017; Gardner et al., 2018; Stein et al., 2013; Ubaru et al., 2017; Wang et al., 2019). However, it remains relatively unknown in the field of spatial statistics, and one major obstacle for its application in spatial statistics is that the theoretical properties of the parameter estimates are not well-established.

In this work, both the theoretical and computational aspects of the Krylov subspace methods are studied in the context of parameter estimation for large spatial datasets. In particular, we approximate matrix inversion through the conjugate gradient, and the log determinant through the Lanczos quadrature approximation for large spatial datasets. A key contribution of our work is that the approximation error can be quantified and can be shown to tend to zero asymptotically, which enables us to establish the theoretical property of the parameter estimates. On the computational aspects, the proposed methods work for both dense and sparse covariance matrices. In particular, the computational complexity of our method is $O(N \log N)$ for sparse matrices, and thus, it can further speed up the covariance tapering method, which has a computational complexity $O(N^{3/2})$ (Bolin & Wallin, 2016; Lipton et al., 1979). Promisingly, simulation studies in Section 3.5 show that this improved computational efficiency is achieved with little efficiency loss compared to the covariance tapering method, especially for the regression coefficient estimates. Thirdly, the trade-off between the convergence rate of parameter estimates and the computational complexity of the computational algorithms is also established, and a root-N convergence of the parameter estimates is established under the increasing domain asymptotic framework.

The remainder of the paper is organized as follows. Section 2 develops our Krylov subspace based methodology and gives theoretical justifications. We provide analysis of the computational complexity and conduct simulation studies to investigate the computational efficiency and statistical properties in Section 3. In Section 4, we illustrate the methodology by a dataset with over 5 million observations comprising light detection and ranging (LiDAR) estimates of forest canopy height in western Alaska. The theoretical development is given in Appendix.

2 | METHODOLOGY

Consider a spatial process $\{y(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{D} \subset \mathbb{R}^d\}$, where $\mathcal{D}$ is a spatial domain of interest in $\mathbb{R}^d$. For ease of exposition, we assume that the mean of the spatial process is zero and focus on the main computational challenges that arise in estimating the spatial covariance structure. Denote the spatial covariance between $y(\boldsymbol{s})$ and $y(\boldsymbol{s}')$ at two locations $\boldsymbol{s}, \boldsymbol{s}' \in \mathcal{D}$ by

$$\gamma\left(\boldsymbol{s}, \boldsymbol{s}'; \boldsymbol{\theta}\right) = \mathrm{Cov}\left\{y(\boldsymbol{s}), y(\boldsymbol{s}')\right\},$$

which is assumed to be known up to a q-dimensional vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^q$. The inference on $\boldsymbol{\theta}$ is based on N observations $\boldsymbol{y} = (y(\boldsymbol{s}_1), \ldots, y(\boldsymbol{s}_N))^\top$ collected at N spatial sampling locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N \in \mathcal{D}$. The log-likelihood function can thus be written as

$$\ell(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\log|\boldsymbol{\Gamma}(\boldsymbol{\theta})| - (1/2)\boldsymbol{y}^\top\boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1}\boldsymbol{y}, \tag{1}$$

where $\boldsymbol{\Gamma}(\boldsymbol{\theta}) = [\gamma(\boldsymbol{s}_j, \boldsymbol{s}_{j'}, \boldsymbol{\theta})]_{j,j'=1}^N$ is the spatial covariance matrix of $\boldsymbol{y}$. Henceforth, we suppress $\boldsymbol{\theta}$ in $\boldsymbol{\Gamma}(\boldsymbol{\theta})$ for notational convenience except for Section 2.3.

Maximizing the log-likelihood function (1) yields the maximum likelihood estimates (MLE) of $\boldsymbol{\theta}$, but can be challenging particularly when the sample size N is large. A computational bottleneck is repeated computation of the inverse and the log-determinant of the spatial covariance matrix $\boldsymbol{\Gamma}$ evaluated at different $\boldsymbol{\theta}$ values. Most commonly used approaches for solving linear systems involve Cholesky decompositions, which generally require $O(N^3)$ operations and $O(N^2)$ memory. Here, to alleviate the computational burden, we consider Krylov subspace-based methods, known for their effectiveness in solving large linear systems and finding the leading eigenpairs of large matrices, especially when the linear systems or the matrices are sparse.

2.1 | Matrix inversion via conjugate gradient

Since the log-likelihood function (1) does not require an explicit storage of $\boldsymbol{\Gamma}^{-1}$ but rather a vector $\boldsymbol{\Gamma}^{-1}\boldsymbol{y}$, iterative techniques based only on matrix-vector multiplications can be applied (Shewchuk 1994). A class of such iterative methods relies on the Krylov subspace. For an invertible matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and a nonzero vector $\boldsymbol{v} \in \mathbb{R}^N$, the kth Krylov subspace generated by the pair $(\boldsymbol{A}, \boldsymbol{v})$ is defined as

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{v}) = \text{span}\left\{\boldsymbol{v}, \boldsymbol{A}\boldsymbol{v}, \ \dots\ , \boldsymbol{A}^{k-1}\boldsymbol{v}\right\},$$

for $k \geq 1$ and $\mathcal{K}_0(\boldsymbol{A}, \boldsymbol{v}) = \{\boldsymbol{0}\}$. Consider a linear system $\boldsymbol{r}_0 = \boldsymbol{A}\boldsymbol{z}$, where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{z}, \boldsymbol{r}_0 \in \mathbb{R}^N$. Let $p(\cdot)$ be the minimal polynomial of $\boldsymbol{A}$ with degree L for $L \leq N$. Thus, $p(\boldsymbol{A}) = \sum_{l=0}^L \zeta_l \boldsymbol{A}^l = 0$, where ζ_l is the coefficients of the minimal polynomial, and we have $\boldsymbol{A}^{-1}\boldsymbol{r}_0 = -\zeta_0^{-1}\sum_{l=1}^L \zeta_l \boldsymbol{A}^{l-1}\boldsymbol{r}_0$. This suggests that the solution to the linear system, $\boldsymbol{z} = \boldsymbol{A}^{-1}\boldsymbol{r}_0$, lies in the Lth Krylov subspace $\mathcal{K}_L(\boldsymbol{A}, \boldsymbol{r}_0)$. The Krylov subspace-based methods seek to find an approximate solution in the growing Krylov subpace $\mathcal{K}_l(\boldsymbol{A}, \boldsymbol{r}_0), l = 1, 2, \ \dots\ , L$, and in L steps, an exact solution can be obtained.

One commonly used Krylov subspace-based method is the conjugate gradient (CG), built upon a set of mutually conjugate search directions $\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ \dots\ \}$ such that $\boldsymbol{d}_l^\top \boldsymbol{A}\boldsymbol{d}_{l'} = 0$ for $l \neq l'$. Among Krylov subspace-based methods, CG is optimal in the sense that it minimizes the energy function $f(\boldsymbol{z}) = (1/2)\boldsymbol{z}^\top\boldsymbol{A}\boldsymbol{z} - \boldsymbol{r}_0^\top\boldsymbol{z}$ at each iteration. The solution at the lth iteration can be formulated as

$$\boldsymbol{z}_l = \underset{\boldsymbol{z}\in\mathcal{K}_l(\boldsymbol{A},\boldsymbol{r}_0)}{\arg\min} f(\boldsymbol{z}), \quad l \geq 0.$$

For a positive definite matrix $\boldsymbol{A}$, the sequence of solutions $\{\boldsymbol{z}_l\}$ converges to the unique solution $\boldsymbol{z}^* = \boldsymbol{A}^{-1}\boldsymbol{r}_0$ in at most N iterations. Let ψ_l be the relative error measured at the lth iteration. It is well known that

$$\psi_l = \frac{f(\boldsymbol{z}_l) - f(\boldsymbol{z}^*)}{f(\boldsymbol{z}_0) - f(\boldsymbol{z}^*)} = \frac{\|\boldsymbol{z}_l - \boldsymbol{z}^*\|_{\boldsymbol{A}}^2}{\|\boldsymbol{z}^*\|_{\boldsymbol{A}}^2} \leq 2\left\{\frac{\kappa(\boldsymbol{A})^{1/2} - 1}{\kappa(\boldsymbol{A})^{1/2} + 1}\right\}^l, \tag{2}$$

where $\boldsymbol{z}_0 = \boldsymbol{0}$ is the starting vector, $\kappa(\boldsymbol{A}) = \lambda_{\max}/\lambda_{\min}$ is the condition number of $\boldsymbol{A}$, and $\|\boldsymbol{z}\|_{\boldsymbol{A}} \equiv \boldsymbol{z}^\top\boldsymbol{A}\boldsymbol{z}$. The CG algorithm is monotonically improving, since ψ_l decreases as l increases (Golub & Van Loan, 2012).

To maximize the log-likelihood function (1), let $\widehat{\boldsymbol{\theta}}_{\mathrm{MLE}} = \arg\ \max_{\theta}\{\ell(\boldsymbol{\theta})\}$ denote the MLE of $\boldsymbol{\theta}$ and let $\boldsymbol{\theta}_0$ be the true value of $\boldsymbol{\theta}$. With $\boldsymbol{z} = \boldsymbol{\Gamma}^{-1}\boldsymbol{y}$, we rewrite (1) as

$$\ell(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\log|\boldsymbol{\Gamma}| - (1/2)\boldsymbol{y}^{\top}\boldsymbol{z}.$$

To circumvent the computation of matrix inverse, we approximate $\boldsymbol{z}$ by the CG algorithm, which is efficient computationally and in data storage.

The theoretical property of the maximum likelihood estimator is considered under the increasing domain asymptotic framework (Mardia & Marshall, 1984). Following Lahiri (2003), we denote $\mathcal{U}_0 \subset \mathbb{R}^d$ as an open and connected subset of $(-1/2,\ 1/2\,]^d$, and the prototype of sampling region $\mathcal{D}_0$ is a Borel set satisfying $\mathcal{U}_0 \subset \mathcal{D}_0 \subset cl(\mathcal{U}_0)$, where $cl(\mathcal{U}_0)$ denotes the closure of $\mathcal{U}_0$. At stage n, the sampling region is denoted as $\mathcal{D}_n = \tau_n\mathcal{D}_0$, where $\{\tau_n\}$ is an increasing sequence of positive numbers. Let N_n be the sample size at the nth stage of the asymptotics, which is at the rate of τ_n^d. Thus, the density at stage n is $N_n\tau_n^{-d}|\mathcal{D}_0|^{-1}$ and bounded under the increasing domain asymptotic framework.

Let $\boldsymbol{z}_l$ be an approximation of $\boldsymbol{z}$ in the lth iteration of the CG algorithm and write the approximate log-likelihood function at the lth iteration as

$$\tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\log|\boldsymbol{\Gamma}| - (1/2)\boldsymbol{y}^{\top}\boldsymbol{z}_l. \tag{3}$$

For ease of notation, we suppress the subscript n except for the theorems and the technical details in the Appendix. Let $\ell'(\boldsymbol{\theta}) = \partial\ell(\boldsymbol{\theta})/\partial\boldsymbol{\theta}$ and $\ell''(\boldsymbol{\theta},\boldsymbol{\theta}) = \partial^2\ell(\boldsymbol{\theta})/\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\top}$ denote the first- and second-order derivatives of $\ell(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, respectively. For $\iota, \iota' = 1, \ldots, q$, the ιth element of $\ell'(\boldsymbol{\theta})$ is $-(1/2)\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota) - (1/2)\boldsymbol{y}^{\top}\boldsymbol{\Gamma}^{\iota}\boldsymbol{y}$, where $\boldsymbol{\Gamma}_\iota = \partial\boldsymbol{\Gamma}/\partial\theta_\iota$ and $\boldsymbol{\Gamma}^{\iota} = \partial\boldsymbol{\Gamma}^{-1}/\partial\theta_\iota = -\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}$. The (ι,ι')th entry of $\ell''(\boldsymbol{\theta},\boldsymbol{\theta})$ is $-(1/2)\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota\iota'} + \boldsymbol{\Gamma}^{\iota}\boldsymbol{\Gamma}_{\iota'}) - (1/2)\boldsymbol{y}^{\top}\boldsymbol{\Gamma}^{\iota\iota'}\boldsymbol{y}$, where $\boldsymbol{\Gamma}^{\iota\iota'} = \partial^2\boldsymbol{\Gamma}^{-1}/\partial\theta_\iota\partial\theta_{\iota'} = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'} + \boldsymbol{\Gamma}_{\iota'}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota - \boldsymbol{\Gamma}_{\iota\iota'})\boldsymbol{\Gamma}^{-1}$. Moreover, let $\boldsymbol{I}_n(\boldsymbol{\theta}) = \mathbb{E}(-\ell''(\boldsymbol{\theta},\boldsymbol{\theta}))$ denote the information matrix of $\boldsymbol{\theta}$, then the (ι,ι')th entry of the information matrix is ${}^{n}t_{\iota\iota'}/2$, where ${}^{n}t_{\iota\iota'} = \mathrm{tr}({}^{n}\boldsymbol{\Gamma}^{-1}{}^{n}\boldsymbol{\Gamma}_\iota{}^{n}\boldsymbol{\Gamma}^{-1}{}^{n}\boldsymbol{\Gamma}_{\iota'})$.

Let $\lambda_1 \le \ldots \le \lambda_N$, $|\lambda_1^{\iota}| \le \ldots \le |\lambda_N^{\iota}|$ and $|\lambda_1^{\iota\iota'}| \le \ldots \le |\lambda_N^{\iota\iota'}|$, $\iota,\iota' = 1, \ldots q$ denote the eigenvalues of $\boldsymbol{\Gamma}$, $\boldsymbol{\Gamma}^{\iota}$ and $\boldsymbol{\Gamma}^{\iota\iota'}$, respectively. Moreover, $X_n = O_p(a_n)$ if and only if X_n/a_n is bounded in probability. The following regularity conditions are assumed for Theorems 1–3.

(A1) Given $d \ge 0$, the covariance function $\gamma(\boldsymbol{s},\boldsymbol{s}';\boldsymbol{\theta})$ is twice continuously differentiable with respect to $\boldsymbol{\theta}$.

(A2) $limsup_{n\to\infty}\lambda_{N_n} = c < \infty$, $limsup_{n\to\infty}\lambda^{\iota}_{N_n} = c^{\iota} < \infty$, $limsup_{n\to\infty}\lambda^{\iota\iota'}_{N_n} = c^{\iota\iota'} < \infty$ for some positive constants c, c^{ι} and $c^{\iota\iota'}$ and all $\iota,\iota' = 1, \ldots, q$.

(A3) For all $k,k' = 1, \ldots, q$, ${}^{n}a_{\iota\iota'} = \lim_{n\to\infty}\{{}^{n}t_{\iota\iota'}/({}^{n}t_{\iota\iota}{}^{n}t_{\iota'\iota'})^{1/2}\}$ exists and ${}^{n}\boldsymbol{A} = ({}^{n}a_{\iota\iota'})_{\iota,\iota'=1}^{q}$ is a nonsingular matrix.

(A4) $\lim_{n\to\infty} N_n^{-1}\boldsymbol{I}_n(\boldsymbol{\theta}) \to \boldsymbol{J}(\boldsymbol{\theta})$, where $\boldsymbol{J}(\boldsymbol{\theta})$ is nonsingular.

(A5) $\sup_{\theta\in\Omega}\kappa(\boldsymbol{\theta}) \equiv \kappa_0 = O_p(1)$, where Ω is an open subset of $\mathbb{R}^q$ such that $\boldsymbol{\theta}_0 \in \Omega$.

Assumption (A1) requires the covariance functions to be twice continuously differentiable, which is satisfied by most popular covariance functions, such as the Matérn class and the Gaussian covariance function (Mardia & Marshall, 1984). Assumption (A2) is related to the decay of the covariance functions and their derivatives as well as the sampling design. A general condition is given in theorem 3 of Chu (in press) for the increasing domain asymptotics. Assumption (A3) ensures the information is nonsingular at the limit and therefore, the elements of $\boldsymbol{\theta}$ are not

asymptotically linear dependent (Mardia & Marshall, 1984). Assumption (A4) ensures the convergence of information matrix and a similar assumption is made in Chu et al. (2011). Assumption (A5) assumes that the condition number of the covariance matrix is uniformly bounded in $\boldsymbol{\theta}$ and a general condition is given in Lemma 7.

Theorem 1. *Under (A1)–(A5), there exists, with probability tending to one, a local maximizer* $\widehat{\boldsymbol{\theta}}^{(l)}$ *of* $\tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y})$*, such that*

$$\|\widehat{\boldsymbol{\theta}}^{(l)} - \boldsymbol{\theta}_0\| = O_p\left(\max\left\{\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^{l/2}, N_n^{-1/2}\right\}\right),$$

where $\kappa_0 \equiv \sup_{\theta\in\Omega}\kappa(\boldsymbol{\theta})$*,* $\kappa(\boldsymbol{\theta})$ *is the condition number of* $\boldsymbol{\Gamma}(\boldsymbol{\theta})$ *and* Ω *is an open subset of* $\mathbb{R}^q$ *such that* $\boldsymbol{\theta}_0 \in \Omega$*. In particular, if* $l > (\sqrt{\kappa_0}+1)/2\log N_n$*, we have* $\|\widehat{\boldsymbol{\theta}}^{(l)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$.

Theorem 1 establishes the existence of a local maximizer for $\tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y})$ given in (3). The consistency of $\widehat{\boldsymbol{\theta}}^{(l)}$ is determined by the sample size N, the condition number κ_0, and the number of iterations l in the CG algorithm. For the increasing domain asymptotics considered here, κ_0 is bounded as the sample size increases, as shown in lemma 7 of appendix A.4. Given the bounded condition number, the estimate $\widehat{\boldsymbol{\theta}}^{(l)}$ achieves root-N consistency in $O(\log N)$ iterations. The proof of Theorem 1 is given in A.0.1. In practice, l is selected by some predetermined stopping criterion for each iteration and hence may vary by $\boldsymbol{\theta}$.

2.2 | Log-determinant approximation via stochastic Lanczos

Next, we consider an approximation of the log-determinant of $\boldsymbol{\Gamma}$. For a positive-definite matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, its eigen decomposition can be written as $\boldsymbol{A} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is the $N \times N$ matrix whose ith column is the ith eigenvector of $\boldsymbol{A}$ and $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \ldots, \lambda_N\}$ is a diagonal matrix with the eigenvalues of $\boldsymbol{A}$ in ascending order along the diagonal. It is well known that $\log\det(\boldsymbol{A}) = \sum_{i=1}^N \log(\lambda_i)$ and the matrix log-determinant can be computed by the eigen decomposition, but it can be computationally costly for large matrices.

An alternative approach is to use a stochastic trace estimator (Hutchinson, 1990). The logarithm of $\boldsymbol{A}$ can be expressed as $\log(\boldsymbol{A}) = \boldsymbol{Q}\log(\boldsymbol{\Lambda})\boldsymbol{Q}^\top$, and the eigenvalues of $\log(\boldsymbol{A})$ are $\log(\lambda_1), \ldots, \log(\lambda_N)$. Thus, we have

$$\log\det(\boldsymbol{A}) = \text{tr}(\log(\boldsymbol{A})) = \sum_{i=1}^N \log(\lambda_i). \tag{4}$$

In addition, for any square matrix $\boldsymbol{B} \in \mathbb{R}^{N\times N}$, we have

$$\text{tr}(\boldsymbol{B}) = \mathbb{E}(\boldsymbol{u}^\top\boldsymbol{B}\boldsymbol{u}), \tag{5}$$

where $\boldsymbol{u} = (u_1, \ldots, u_N)^\top$ is a vector of independent samples from a random variable with mean 0 and variance 1. By (4) and (5), we have

$$\log\det(\boldsymbol{A}) = \mathbb{E}(\boldsymbol{u}^\top\log(\boldsymbol{A})\boldsymbol{u}). \tag{6}$$

Thus, a Monte Carlo estimator of $\mathbb{E}(\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u})$ can be used to approximate $\log\det(\boldsymbol{A})$. Among all zero mean unit variance random variables, the Rademacher random variable is shown to achieve the minimum variance of $\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u}$ (Hutchinson, 1990). This leads to the Hutchinson trace estimator

$$\log\det(\boldsymbol{A}) \approx N_v^{-1}\sum_{i=1}^{N_v} \boldsymbol{\chi}_i^\top \log(\boldsymbol{A})\boldsymbol{\chi}_i = N/N_v \sum_{i=1}^{N_v} \boldsymbol{u}_i^\top \log(\boldsymbol{A})\boldsymbol{u}_i, \tag{7}$$

where $\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_{N_v}$ are *i.i.d.* Rademacher random variables, $\boldsymbol{u}_i = \boldsymbol{\chi}_i/\|\boldsymbol{\chi}_i\|_2$, and N_v is the Monte Carlo sample size.

Evaluation of (7) still requires an eigen decomposition of $\log(\boldsymbol{A})$ and thus, we further approximate the quadratic form $\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u}$ numerically. The analytic function $\log(\cdot)$ can be approximated using the orthonormal polynomial techniques, namely, Taylor's expansions (Zhang & Leithead, 2007), Chebyshev expansions (Han et al., 2015) and their variants (Boutsidis et al., 2017). Here, we adopt a Gaussian quadrature rule, which outperforms the aforementioned methods (Ubaru et al., 2017). Let α_i denote the ith element of $\boldsymbol{Q}^\top\boldsymbol{u}$. We have $\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u} = \boldsymbol{u}^\top\boldsymbol{Q}\log(\boldsymbol{\Lambda})\boldsymbol{Q}^\top\boldsymbol{u} = \sum_{i=1}^{N}\log(\lambda_i)\alpha_i^2$, which can be written as a Riemann–Stieltjes integral with piecewise constant measure

$$\sum_{i=1}^{N}\log(\lambda_i)\alpha_i^2 = \int_{\lambda_1}^{\lambda_N}\log(\lambda)d\alpha(\lambda), \tag{8}$$

where $\alpha(t) = 0$ if $\lambda_i \le t < \lambda_1$, $\alpha(t) = \sum_{k=1}^{i}\alpha_k^2$ if $\lambda_i \le t < \lambda_{i+1}$ and $\alpha(t) = \sum_{k=1}^{N}\alpha_k^2$ if $t \ge \lambda_N$. We approximate (8) via the Gaussian quadrature rule, with a general form given by

$$\int_{\lambda_1}^{\lambda_N}\log(\lambda)d\alpha(\lambda) \approx \sum_{i=0}^{m-1}\omega_i\log(\phi_i), \tag{9}$$

where $\{(\omega_i, \phi_i), i = 0, 1, \dots, m-1\}$ are the weight-node pairs of the m-point Gaussian quadrature rule for $m << N$ and can be computed by the Lanczos algorithm (Golub & Welsch, 1969). In Algorithms 1 and 2, we perform an orthonormalization of the Krylov subspace and approximate the log-determinant by the Gaussian quadrature rule, respectively.

Algorithm 1. Lanczos algorithm for orthonormalization of the Krylov subspace $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{u})$

Initialization: $\boldsymbol{q}_1 = \boldsymbol{u}/\|\boldsymbol{u}\|_2$, $\boldsymbol{Q}_1 = [\boldsymbol{q}_1]$, $a_1 = (\boldsymbol{q}_1)^\top\boldsymbol{A}\boldsymbol{q}_1$, $\boldsymbol{\xi}_1 = (\boldsymbol{A} - a_1\boldsymbol{I})\boldsymbol{q}_1$.
for $k = 2, \dots, m$ **do**
  $b_k = \|\boldsymbol{\xi}_{k-1}\|_2$
  if $b_k = 0$ **then**
    return $(\boldsymbol{Q}; a_1, \dots, a_k; b_1, \dots, b_{k-1})$
  $\boldsymbol{q}_k = \boldsymbol{\xi}_{k-1}/b_k$; $\boldsymbol{Q}_k = [\boldsymbol{Q}_{k-1}, \boldsymbol{q}_k]$
  $a_k = (\boldsymbol{q}_k)^\top\boldsymbol{A}\boldsymbol{q}_k$
  $\boldsymbol{\xi}_k = (\boldsymbol{A} - a_k\boldsymbol{I})\boldsymbol{q}_k - b_k\boldsymbol{q}_{k-1}$
end
output: $(\boldsymbol{Q}_m; a_1, \dots, a_m; b_1, \dots, b_{m-1})$

Algorithm 1 is the standard Lanczos algorithm, which orthonormalizes the Krylov subspace $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{u})$ and yields an $n \times m$ matrix $\boldsymbol{Q}_m$ whose columns are orthonormal bases of the Krylov subspace $\mathcal{K}_m(\boldsymbol{A}, \boldsymbol{u})$ and an $m \times m$ tridiagonal matrix $\boldsymbol{T}_m$ with the diagonal elements being $(a_1, \ldots, a_m)$ and the sub-diagonal elements and super-diagonal elements being $(b_1, \ldots, b_{m-1})$. Let $p(\cdot)$ denote the polynomial of the smallest degree such that $p(\boldsymbol{A})\boldsymbol{u} = 0$ and let M denote the degree of $p(\cdot)$. Then $\boldsymbol{A}\boldsymbol{Q}_M = \boldsymbol{Q}_M\boldsymbol{T}_M$ and the eigenvalues of $\boldsymbol{T}_M$ are also eigenvalues of $\boldsymbol{A}$. Let $\{(\phi_k, \boldsymbol{\Phi}_k), k = 0, 1, \ldots, m-1\}$ be the eigenpairs of $\boldsymbol{T}_m$. We approximate $\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u}$ by

$$\boldsymbol{u}^\top \log(\boldsymbol{A})\boldsymbol{u} \approx \sum_{k=0}^{m-1} \omega_k \log(\phi_k) \quad \text{with} \quad \omega_k = (\boldsymbol{e}_1^\top \boldsymbol{\Phi}_k)^2, \tag{10}$$

where $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^\top \in \mathbb{R}^m$.

The log-determinant of $\boldsymbol{A}$ can then be approximated as

$$\log|\boldsymbol{A}| \approx N/N_v \sum_{i=1}^{N_v} \left\{ \sum_{k=0}^{m-1} \omega_k^{(i)} \log(\phi_k^{(i)}) \right\}, \tag{11}$$

where $\{(\phi_k^{(i)}, \omega_k^{(i)}), k = 0, \ldots, m-1\}$ are the eigenvalues and the square of the first element of the eigenvectors corresponding to the ith starting vector (Ubaru et al., 2017).

Algorithm 2. Log-determinant approximation by the Gaussian quadrature rule

Input : A positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, degree m and number of starting vectors N_v.

Output: $\Xi_{m,N_v} = N/N_v \sum_{i=1}^{N_v} \left\{ \sum_{k=0}^{m-1} \omega_k^{(i)} \log(\phi_k^{(i)}) \right\}$.

1 **for** $i = 1, \ldots, N_v$ **do**

2 Draw a Rademacher random vector χ_i as the ith starting vector

3 Calculate $\boldsymbol{T}_m^{(i)}$ through Algorithm 1 with $\boldsymbol{A} = \boldsymbol{\Gamma}$ and $\boldsymbol{v} = \chi_i$

4 Calculate eigenpairs $(\phi_k^{(i)}, \boldsymbol{\Phi}_k^{(i)})$ of $\boldsymbol{T}_m^{(i)}$ and compute $\omega_k^{(i)} = (\boldsymbol{e}_1^\top \boldsymbol{\Phi}_k^{(i)})^2$ for $k = 0, 1, \ldots, m-1$.

Combining with the CG algorithm, we further approximate the log-likelihood function with

$$\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}; \boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\Xi_{m,N_v} - (1/2)\boldsymbol{y}^\top \boldsymbol{z}_l, \tag{12}$$

where Ξ_{m,N_v} is the approximation of $\log\det(\boldsymbol{\Gamma})$ by Algorithm 2, and $\boldsymbol{z}_l$ is an approximate solution to $\boldsymbol{\Gamma}^{-1}\boldsymbol{y}$ at the lth iteration of the CG algorithm. Maximizing $\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}; \boldsymbol{y})$ yields an approximate estimate of $\boldsymbol{\theta}$. We establish the existence and consistency of such estimator in the following Theorem 2.

Theorem 2. *Under (A1)–(A5), there exists, with probability tending to one, a local maximizer* $\widehat{\boldsymbol{\theta}}^{(l,m,N_v)}$ *of* $\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}; \boldsymbol{y})$*, such that*

$$\|\widehat{\boldsymbol{\theta}}^{(l,m,N_v)} - \boldsymbol{\theta}_0\| = O_p\left(\max\left\{ \left(\frac{\sqrt{\kappa_0} - 1}{\sqrt{\kappa_0} + 1} \right)^{l/2}, \left(\frac{\sqrt{\kappa_0} - 1}{\sqrt{\kappa_0} + 1} \right)^{m}, N_n^{-1/2} \right\} \right).$$

If $l > \frac{\sqrt{\kappa_0}+1}{2}\log N_n$ *and* $m > \frac{\sqrt{\kappa_0}}{4}\log(N_n C_1)$, *where* $C_1 = \lambda_{\max}\sqrt{\kappa_0}\log(\lambda_{\max}+\lambda_{min})$, *we have* $\|\widehat{\boldsymbol{\theta}}^{(l,m,N_v)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$.

The proof of Theorem 2 is given in A.0.2. By Theorem 2, the consistency of $\widehat{\boldsymbol{\theta}}^{(l,m,N_v)}$ is determined by the sample size N, the condition number κ_0, the number of iterations in the CG algorithm l, and the order of the Gaussian quadrature rule m. Given a well-conditioned covariance matrix, the estimate $\widehat{\boldsymbol{\theta}}^{(l)}$ achieves root-N consistency in $O(\log N)$ iterations. In addition, the convergence result does not depend on the Monte Carlo sample size N_v, which will be illustrated by simulation in Section 3.3. Also will be seen in simulation is that N_v has negligible impact on the accuracy of the log-likelihood evaluation.

2.3 | Generalization to spatial linear regression

We now generalize the previous two subsections to the setting of spatial linear regression. Consider a Gaussian process $\{y(\boldsymbol{s}) : \boldsymbol{s} \in \mathcal{D} \subset \mathbb{R}^d\}$ such that

$$y(\boldsymbol{s}) = \boldsymbol{x}(\boldsymbol{s})^\top \boldsymbol{\beta} + \varepsilon(\boldsymbol{s}), \tag{13}$$

where $\boldsymbol{x}(\boldsymbol{s}) = (x_1(\boldsymbol{s}), \ldots, x_p(\boldsymbol{s}))^\top$ is a $p \times 1$ vector of covariates and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is a $p \times 1$ vector of regression coefficients. The error process $\varepsilon(\boldsymbol{s})$ is assumed to have zero mean, and the spatial covariance between $\varepsilon(\boldsymbol{s})$ and $\varepsilon(\boldsymbol{s}')$ is $\gamma(\boldsymbol{s}, \boldsymbol{s}'; \boldsymbol{\theta}) = \mathrm{Cov}\{\varepsilon(\boldsymbol{s}), \varepsilon(\boldsymbol{s}')\}$. For $j = 1, \ldots, p$, we write $\boldsymbol{x}_j = (x_j(\boldsymbol{s}_1), \ldots, x_j(\boldsymbol{s}_N))^\top$, where $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N \in \mathcal{D}$ are the spatial sampling locations. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p] \in \mathbb{R}^{N\times p}$ denote the design matrix.

The log-likelihood function can be written as

$$\ell(\boldsymbol{\beta}, \boldsymbol{\theta}) = -(N/2)\log(2\pi) - (1/2)\log|\boldsymbol{\Gamma}(\boldsymbol{\theta})| - (1/2)(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}). \tag{14}$$

For a given $\boldsymbol{\theta}$, maximizing $\ell(\boldsymbol{\beta}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}$ yields the profile likelihood estimate (PLE) of $\boldsymbol{\beta}$,

$$\widehat{\boldsymbol{\beta}}_{\mathrm{PLE}}(\boldsymbol{\theta}) = (\boldsymbol{X}^\top \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1}\boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{\Gamma}(\boldsymbol{\theta})^{-1}\boldsymbol{y}. \tag{15}$$

The term $\boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta})\boldsymbol{X}$ in (15) requires solving linear systems with the same coefficient matrix but different right-hand sides, which can also be carried out iteratively using the CG algorithm. However, the CG algorithm is still computationally demanding, especially when the number of covariates p is large. We consider a computationally more efficient approach with two steps. In the first step, we obtain a consistent estimate of $\boldsymbol{\beta}$. As shown in Appendix A.3, $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$ is a consistent estimate. In the second step, we estimate $\boldsymbol{\theta}$ based on the vector of residuals using the algorithms in Sections 2.1 and 2.2. For the spatial linear regression model, the following regularity condition for the design matrix $\boldsymbol{X}$ is needed.

(A6) The fixed design matrix $\boldsymbol{X}$ satisfies $\lim_{n\to\infty} N_n^{-1}(\boldsymbol{X}^\top\boldsymbol{X})^{-1} \to \boldsymbol{C}$, where $\boldsymbol{C}$ is some positive definite matrix.

In Theorem 3, we show that the resulting estimate of $\boldsymbol{\theta}$ is consistent and achieve the same convergence rate as that in Theorem 2.

Theorem 3. *Under (A1)–(A6), for any given* $\widetilde{\boldsymbol{\beta}}$ *satisfying* $\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(N_n^{-1/2})$*, there exists, with probability tending to one, a local maximizer* $\widehat{\boldsymbol{\theta}}^{(l,m,N_v)}$ *of* $\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}; \boldsymbol{y} - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})$*, such that the results in Theorem 2 hold.*

In practice, we propose to use the ordinary least squares (OLS) estimate $\widehat{\boldsymbol{\beta}}_{\text{OLS}}$, which is root-$N$ consistent; see Appendix A.3. Thus, our two-step approach above will result in a consistent estimate of $\boldsymbol{\theta}$. The proof of Theorem 3 is given in Appendix A.3.

3 | COMPUTATIONAL ASPECTS

In Section 3.1, we evaluate the computational complexity of our methodology and in Section 3.2, we provide a fast Krylov covariance tapering method. The trade-off between accuracy and computational complexity, using different choices of m and N_v, is explored in Section 3.3. We also demonstrate the finite-sample properties of our parameter estimation for compactly supported covariances in Section 3.4. In Section 3.5, we compare our methods with competitors including covariance tapering and the nearest neighbor Gaussian process models (Datta et al., 2016).

3.1 | Computational complexity

The computational complexity of evaluating the approximate log-likelihood function (12) is dominated by the algorithms in Sections 2.1 and 2.2. The CG algorithm, involves only scalar product of vectors, inner products of vectors, and matrix-vector multiplications of dimension N in each iteration of the algorithm. Thus, the CG algorithm can be performed in $O(lN^2)$ flops, where l is the number of iterations. Theorem 1 shows that $O(\frac{\sqrt{\kappa_0}+1}{2}\log N)$ number of iterations ensures consistent estimation of the parameters. Algorithm 2 for the log-determinant approximation involves N_v Monte Carlo samples, and each sample involves a Krylov subspace orthogonalization of dimension m via Algorithm 1 and the eigen decomposition of an $m \times m$ matrix. Similar to the CG algorithm, each iteration of the Lanczos algorithm requires only basic linear algebra subroutines and hence, can be performed in $O(mN^2)$ steps. The eigen decomposition can be achieved with a complexity lower than $O(m^3)$, which is dominated by the Lanczos algorithm. Thus, the log-determinant approximation algorithm requires $O(mN_vN^2)$ flops. By Theorem 2, consistency of the parameters is guaranteed by letting m increase with N at a rate of $O(\log N)$. Thus, the computational complexity of the log-determinant approximation is $O(N_vN^2 \log N)$.

For dense spatial covariance matrices, our method with the computational complexity $O(N^2 \log N)$ provides a substantial improvement over the traditional Cholesky decomposition with $O(N^3)$. For sparse covariance matrices, the covariance tapering has a computational complexity of $O(N^{3/2})$ (Bolin & Wallin, 2016; Lipton et al., 1979). For the proposed method, we can achieve quasi-linear complexity by exploiting sparsity. Indeed, sparse matrix-vector multiplications involves only $O(\|\boldsymbol{\Gamma}\|_0)$ operations and thus, our algorithm can be implemented in $O((mN_v + l)\|\boldsymbol{\Gamma}\|_0)$ flops, where $\|\boldsymbol{\Gamma}\|_0$ is the number of nonzero entries of $\boldsymbol{\Gamma}$. The implementation of Algorithms 1–3 is carried out through `ViennaCL`.

In practice, the number of iteration of CG algorithm is closely related to the conditional number, and can be improved by various preconditioning methods (Chen, 2013; Chow & Saad, 2014;

Cutajar et al., 2016; Gardner et al., 2018; Stein et al., 2012). The computational efficiency of the proposed method can be further improved by the block CG method, which takes advantage of the same covariance matrix in the algorithm (Chen, 2011). We will leave these for future research.

3.2 | Krylov covariance tapering

We develop an additional Krylov subspace-based method, which achieves similar accuracy to Cholesky decomposition-based methods but is faster to compute. As noted in Sections 3.1, our Krylov subspace-based methodology can achieve quasi-linear complexity by exploiting sparsity. We now introduce sparsity into computation by covariance tapering (Furrer et al., 2006; Kaufman et al., 2008).

The tapered covariance function, constructed by multiplying the covariance function with a compactly supported covariance function, is a valid covariance function but with compact support (Furrer et al., 2006). The covariance-tapered log-likelihood function is given by

$$\ell(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\log|\boldsymbol{\Gamma}_{\text{tap}}| - (1/2)\boldsymbol{y}^{\top}\boldsymbol{\Gamma}_{\text{tap}}^{-1}\boldsymbol{y}, \tag{16}$$

where $\boldsymbol{\Gamma}_{\text{tap}}$ is the tapered covariance matrix. The covariance-tapered log-likelihood function can then be approximated by

$$\tilde{\ell}_{\text{tap}}^{(l,m,N_v)}(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\Xi_{\text{tap},m,N_v} - (1/2)\boldsymbol{y}^{\top}\boldsymbol{z}_{\text{tap}}^{(l)}, \tag{17}$$

where Ξ_{tap,m,N_v} is the stochastic Lanczos estimator of $\log\det(\boldsymbol{\Gamma}_{\text{tap}})$, and $\boldsymbol{z}_{\text{tap}}^{(l)}$ is the approximation of $\boldsymbol{\Gamma}_{\text{tap}}^{-1}\boldsymbol{y}$ at the lth iteration. Maximizing (17) yields an approximate MLE of the covariance parameters. For spatial linear regression, a procedure similar to Section 2.3 can be applied.

3.3 | Computational efficiency

To evaluate the computational efficiency of our Krylov subspace base methodology, we simulate datasets from a Gaussian process as follows. First, the spatial locations are sampled from a two-dimensional spatial domain $[0, \sqrt{N}/2]^2$, where N is the sample size, to guarantee a fixed sampling density of 4. Then each simulated dataset is generated from a zero-mean Gaussian process with an exponential covariance function

$$\gamma_{\exp}(\boldsymbol{s},\boldsymbol{s}';\boldsymbol{\theta}) = \sigma^2(1-c)\exp(-\|\boldsymbol{s}-\boldsymbol{s}'\|/r), \tag{18}$$

where $r = 2$ is the range parameter, $c = 0.2$ is the nugget proportion such that $c\sigma^2$ is the nugget effect, $\sigma^2 = 9$ is the variance, and $\boldsymbol{\theta} = (r, c, \sigma^2)^{\top}$. We simulate datasets with sample sizes ranging roughly from 5000 to 250,000. We also construct the tapered covariance using the Wendland tapering kernel

$$\gamma_{\delta}(\boldsymbol{s},\boldsymbol{s}') = \left(1 - \|\boldsymbol{s}-\boldsymbol{s}'\|/\delta\right)_{+}^{4}\left(1 + 4\|\boldsymbol{s}-\boldsymbol{s}'\|/\delta\right), \tag{19}$$

where δ is a tapering threshold parameter that controls the sparsity of the covariance matrix (Wendland, 1995).

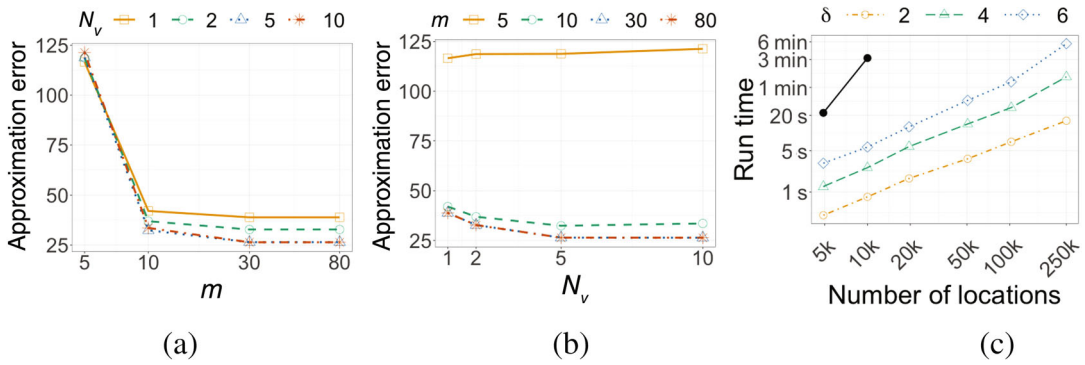

FIGURE 1 (a) Approximation error (i.e., the absolute difference between the approximated and the true negative log-likelihood) by the order of the Gaussian quadrature rule m, under different Monte Carlo sample sizes N_v. (b) Approximation error by the Monte Carlo sample size N_v, under different m. (c) Log run time by log number of spatial sampling locations for a single simulation at different levels of sparsity

The accuracy of our Krylov subspace-based methodology depends on the order of the Gaussian quadrature rule m and the Monte Carlo sample size N_v. The effect of m is shown in Figure 1a with the sample size N fixed at 4900 and the tapering threshold parameter δ at 6. The approximation error (i.e., the absolute difference between the approximated and the true negative log-likelihood) is plotted against m with $N_v = 1, 2, 5$, and 10. For small m, the accuracy of likelihood evaluation is relatively low. Increasing m to 10 effectively reduces the approximation error. We can further see that, for $m \geq 30$, reduction in approximation error is very moderate. This finding is very interesting since, as demonstrated in Theorem 2, m is of the order $\mathcal{O}(\log N)$ to achieve the estimation consistency. From our own experience, $m = 30$ seems to be sufficient. The effect of N_v is shown in Figure 1b with the approximation error plotted against N_v for m fixed at 5, 10, 30, and 80. There is negligible influence of N_v on the quality of the likelihood approximation, even when $N_v = 1$. Based on these findings, we let $m = 30$ and $N_v = 1$ for the remainder of the numerical examples.

Figure 1c shows the run time (averaged over 10 replicates) for a single iteration of likelihood evaluation with sample size N. The solid line represents the time needed for evaluating the exact likelihood, and the run time is reported for N under 10,000. For Krylov covariance tapering, $\delta = 2, 4$, and 6 are considered for the tapering threshold parameter. As δ increases, the sparsity of covariance matrix decreases and the run time increases. In addition, the run time of the Krylov covariance tapering method increases at a slower rate than the exact likelihood as the sample size increases. For $\delta = 2, 4$, and 6, the run time increases approximately linearly, which supports the computational complexity analysis in Section 3.1.

3.4 | Finite-sample properties

As demonstrated in Sections 3.1–3.3, our Krylov subspace-based methodology can achieve quasi-linear complexity by introducing sparsity into computation using correlation functions with compact support. It is not uncommon in practical applications that the spatial correlations among observations are negligible beyond a certain distance, in which cases compactly supported covariances are applicable (Gneiting, 2002). For the spatial linear model (13), we generate two covariates from a Gaussian distribution with unit variance and cross-covariate correlation of 0.5.

The covariates are then standardized to have mean 0 and variance 1. The regression coefficients are set to $\boldsymbol{\beta} = (2, 1)^\top$. The compactly supported covariance function considered is the product of the exponential covariance function (18) and the Wendland kernel function (19)

$$\gamma_{\text{csc}}(\boldsymbol{s}, \boldsymbol{s}') = \gamma_{\text{exp}}(\boldsymbol{s}, \boldsymbol{s}') \cdot \gamma_\delta(\boldsymbol{s}, \boldsymbol{s}'). \tag{20}$$

Therefore, the underlying covariance matrix of simulated datasets is sparse.

Here we benchmark our method against the exact method, whose parameter estimates are obtained through directly maximizing (14) using the `spam` package in `R` (Furrer & Gerber, 2008). In addition, we evaluate the effectiveness of our methodology in Section 2.3, referred to as Krylov-OLS, and compare it with an alternative method by letting the consistent estimate of $\boldsymbol{\beta}$ in the first step be the approximated PLE $\widetilde{\boldsymbol{\beta}}_{\text{PLE}}(\boldsymbol{\theta})$ using the CG algorithm, referred to as Krylov-GLS.

We generate 100 simulated datasets of sample size $N = 4900$. The tapering threshold parameter δ is set to be 6, 10, 12, corresponding to different levels of sparsity. Figure 2 shows boxplots of both the run time and the parameter estimates. At the sparsity level $\delta = 6$, the run time is reduced by a factor of approximately 20 and 33 for Krylov-GLS and Krylov-OLS, respectively. The simulation results show that our Krylov approximation can greatly improve computational efficiency for sparse matrices. For regression coefficients, both the point estimates and variability of the regression coefficients based on Krylov-GLS are nearly identical to the MLE, for each level of sparsity considered. As expected, the Krylov-OLS method has slightly larger bias and variance compared with the other two methods, indicating a loss of statistical efficiency when spatial correlation is unaccounted for. For covariance parameters, the performance of Krylov-OLS is nearly as good as Krylov-GLS, showing that the estimation of the mean trend has little association with covariance estimation. Furthermore, simulation study shows that parameter estimates using Krylov subspace-based methods is unbiased, supporting the conclusions in Theorem 3. However, both Krylov subspace-based methods have larger variances than the exact method, showing reduced efficiency of our method due to the approximation error.

3.5 | Further comparisons

We now compare Krylov-GLS and Krylov-OLS in Section 3.4 with the standard covariance tapering (Tapering) and the nearest-neighbor Gaussian process (NNGP) in the literature, while the exact MLE serves again as the benchmark. These approaches are implemented in `R`, using the `spam` package for covariance tapering, `spNNGP` for NNGP, and our `spKrylov` for the Krylov subspace-based methods.

Consider two simulated examples of sample sizes 5390 and 11,000 both from a Gaussian process with a linear mean function as described in Section 3.4 and with a Matérn covariance function defined as

$$\gamma_{\text{mat}}(\boldsymbol{s}, \boldsymbol{s}'; \boldsymbol{\theta}) = \sigma^2(1-c)\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}\frac{\|\boldsymbol{s}-\boldsymbol{s}'\|}{r}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{\|\boldsymbol{s}-\boldsymbol{s}'\|}{r}\right). \tag{21}$$

Here, $\sigma^2 = 9$ is the variance, $c = 0.2$ is the nugget proportion such that $c\sigma^2$ is the nugget effect, $r = 2$ is the scale parameter, and ν is the smooth parameter. We set $\nu = 0.5$ for the first simulated example and $\nu = 2.5$ for the second one. In each simulated example, 90% of the observations are used for model fitting, and the rest are used for evaluating the predictive performance. For

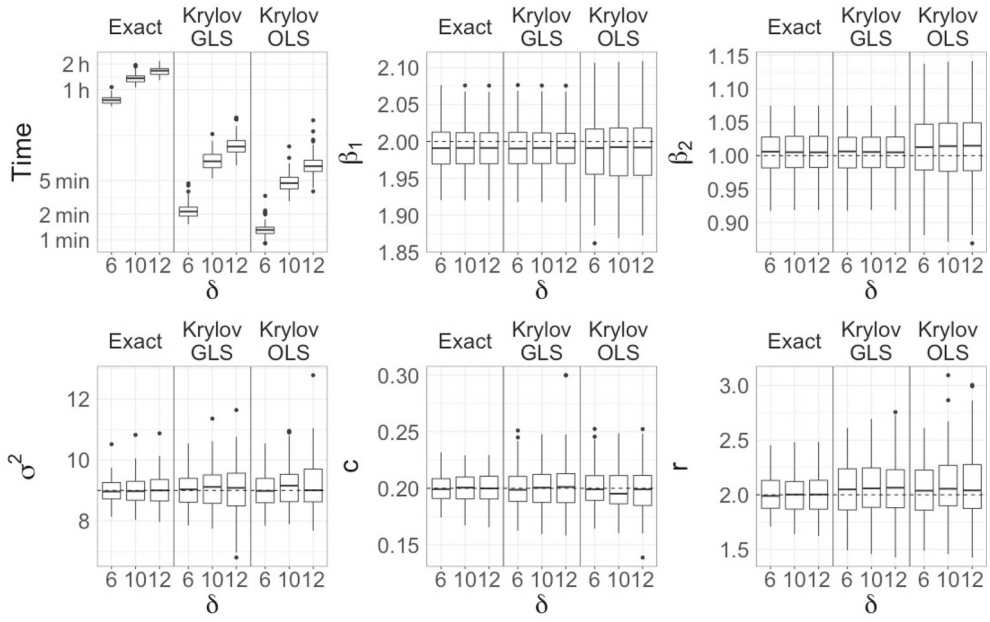

FIGURE 2 Boxplots for run time, estimates of regression coefficients (β_1, β_2), and covariance parameters (σ^2, c, r), under $\delta = 6, 10, 12$ by maximum likelihood estimation (exact) and two Krylov subspace-based methods Krylov-GLS and Krylov-ordinary least squares

observations in the hold-out set, denoted as $y_{i,\text{new}}$ and predicted to be $\widehat{y}_{i,\text{new}}$, $i = 1, \ldots, N_{\text{new}}$, we use the mean squared prediction error (MSPE) to evaluate the prediction accuracy

$$\text{MSPE} = N_{\text{new}}^{-1} \sum_{i=1}^{N_{\text{new}}} (y_{i,\text{new}} - \widehat{y}_{i,\text{new}})^2.$$

For NNGP, the numbers of nearest neighbors are fixed at 10, 20, and 30. For covariance tapering, we taper the Matérn covariance function by

$$\gamma_{\text{tap}}(\boldsymbol{s}, \boldsymbol{s}') = \gamma_{\text{mat}}(\boldsymbol{s}, \boldsymbol{s}') \cdot \gamma_{\delta}(\boldsymbol{s}, \boldsymbol{s}'). \tag{22}$$

Three different levels of sparsity are considered with the tapering threshold parameter δ set to be 6, 10, or 12. The first simulated example is moderately large to accommodate both the exact and the tapered likelihood calculations, whereas the second simulated example has a larger sample size and highlights that the Krylov methods can outperform NNGP. Both simulations are repeated 100 times and the results are summarized in Figures 3 and 4 for the first and second simulated examples, respectively.

Our Krylov covariance tapering involves two stages of approximation. At the first stage, covariance tapering is applied, which assumes a misspecified covariance with distant location pairs forced to be independent. At the second stage, numerical approximations to the tapered likelihood function are obtained within the Krylov subspace. To illustrate the approximation capabilities of the first stage, we compare covariance tapering to the exact MLE. In terms of prediction accuracy and regression coefficient estimates, covariance tapering has nearly indistinguishable

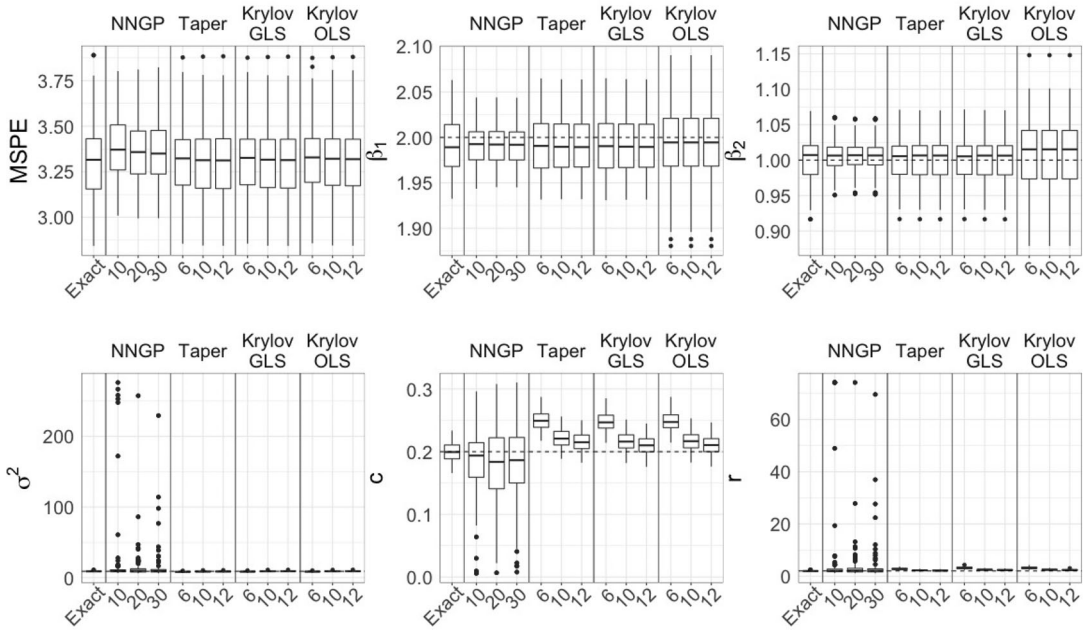

FIGURE 3 Simulated example 1: Boxplots for mean squared prediction error (MSPE), regression coefficients (β_1, β_2) and covariance parameters (σ^2, c, r) for $\delta \in \{6, 10, 12\}$ under maximum likelihood method (Exact), Krylov covariance tapering methods (Krylov-GLS and Krylov-OLS), NNGP method, covariance tapering method (Taper). For nearest-neighbor Gaussian process, the numbers of the nearest neighbors are 10, 20, and 30. For Krylov-GLS and Krylov-OLS, $\delta = 6, 10, 12$

performance from the exact MLE. However, the tapered covariance parameter estimates have larger biases than the exact MLE especially when δ is small. As expected, these biases decrease as we increase the value of δ thereby decrease the sparsity. For the second stage, Krylov-OLS and Krylov-GLS exhibit almost identical estimation and prediction performance to covariance tapering, for all three sparsity levels ($\delta = 6, 10, 12$), with the exception that Krylov-OLS has larger variances of regression coefficient estimates.

For regression coefficient estimates, NNGP has similar accuracy but superior efficiency compared with all other approaches including the exact MLE. However, NNGP is outperformed by tapering-based methods for both covariance estimation and prediction. As shown in Table 1, among all the methods considered, the point estimates of $\boldsymbol{\beta}$ are accurate and the empirical confidence intervals contain the true value. Krylov-GLS has slightly wider confidence intervals than others, which might be due to the approximation error in finding $\boldsymbol{\Gamma}^{-1}\boldsymbol{X}$. Both covariance tapering and Krylov subspace-based methods give similar accuracy and precision in the estimation of covariance parameters. In contrast, for NNGP, the covariance parameter estimates are quite unstable and inaccurate, as shown in Figure 3. Additionally, the MSPE of NNGP ranges from 3.36 to 3.38, which is slightly higher than that of our methodology, ranging from 3.31 to 3.32.

To further illustrate the efficiency and accuracy of the NNGP method and Krylov subspace based methods, we fit a model for the second simulated example using NNGP with $10, 20, 30$ and 50 nearest neighbors and Krylov subspace-based methods with tapering threshold parameter $\delta = 4, 6$ and 10. We record the run time and the MSPE for both methods in Figure 4. As expected, increasing the number of nearest neighbors in NNGP decreases MSPE at the loss of computational

TABLE 1 Percentiles of the estimates of regression coefficients and covariance parameters under maximum likelihood method (Exact), Krylov covariance tapering methods (Krylov-GLS and Krylov-OLS), nearest-neighbor Gaussian process (NNGP) model, and covariance tapering method (Taper). For NNGP, the numbers of nearest neighbors are 10, 20, and 30. For Krylov-GLS and Krylov-OLS, $\delta = 6, 10, 12$

						$\delta = 6$			$\delta = 10$			$\delta = 12$		
Method	**%**	**Exact**	**NNGP (10)**	**NNGP (20)**	**NNGP (30)**	**Taper**	**Krylov GLS**	**Krylov OLS**	**Taper**	**Krylov GLS**	**Krylov OLS**	**Taper**	**Krylov GLS**	**Krylov OLS**
$\beta_1 = 2$	50	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99	1.99
	2.5	1.95	1.95	1.95	1.95	1.95	1.90	1.95	1.95	1.90	1.95	1.95	1.90	1.95
	97.5	2.06	2.04	2.03	2.03	2.06	2.08	2.06	2.06	2.08	2.06	2.06	2.08	2.06
$\beta_2 = 1$	50	1.01	1.01	1.01	1.01	1.01	1.02	1.01	1.01	1.02	1.01	1.01	1.02	1.01
	2.5	0.95	0.96	0.96	0.96	0.94	0.90	0.95	0.95	0.90	0.95	0.95	0.90	0.95
	97.5	1.06	1.05	1.05	1.05	1.06	1.10	1.06	1.06	1.10	1.06	1.06	1.10	1.06
$\sigma^2 = 9$	50	8.91	9.12	9.75	9.44	8.22	8.53	8.52	8.61	9.01	9.01	8.70	9.08	9.08
	2.5	7.97	6.90	6.67	6.62	7.46	7.70	7.70	7.76	8.09	8.08	7.82	8.13	8.15
	97.5	10.18	255.76	44.58	87.92	9.06	9.44	9.44	9.61	10.13	10.13	9.75	10.26	10.26
$c = 0.2$	50	0.20	0.19	0.18	0.19	0.25	0.25	0.25	0.22	0.22	0.22	0.22	0.21	0.21
	2.5	0.17	0.01	0.04	0.02	0.22	0.22	0.22	0.19	0.19	0.19	0.19	0.18	0.18
	97.5	0.23	0.26	0.27	0.27	0.28	0.28	0.28	0.25	0.24	0.24	0.24	0.24	0.24
$r = 2$	50	1.95	2.06	2.15	2.09	2.73	3.11	3.11	2.21	2.46	2.46	2.13	2.34	2.34
	2.5	1.66	1.27	1.23	1.26	2.20	2.48	2.48	1.84	2.03	2.03	1.79	1.96	1.96
	97.5	2.43	74.16	12.28	25.16	3.59	4.15	4.16	2.76	3.08	3.08	2.64	2.91	2.91
MSPE	—	3.31	3.38	3.36	3.36	3.32	3.32	3.32	3.31	3.32	3.31	3.31	3.32	3.31

Abbreviations: MSPE, mean squared prediction error; OLS, ordinary least squares.

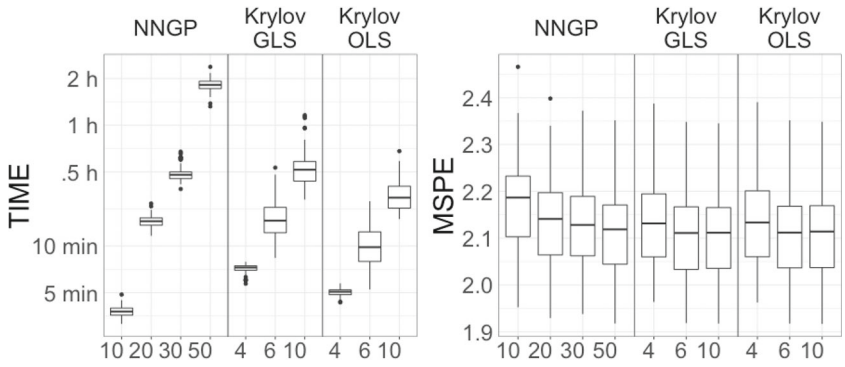

FIGURE 4 Simulated example 2: Boxplots for run time of parameter estimation (in seconds) and mean squared prediction error (MSPE) under Krylov covariance tapering methods (Krylov-GLS and Krylov-OLS) and NNGP model. For nearest-neighbor Gaussian process, the numbers of nearest neighbors are 10, 20, 30, and 50. For Krylov-GLS and Krylov-OLS, $\delta = 4, 6, 10$

efficiency. Similarly for Krylov subspace-based methods, a significant drop in MSPE is obtained by increasing δ from 4 to 6, followed by a plateau, showing that $\delta = 6$ provides an approximation sufficiently close to the exact method. On the contrary, the NNGP method with 50 nearest neighbors has slightly inferior performance and the computation becomes extremely time consuming. Additionally, Krylov-OLS achieves notable computational savings over Krylov-GLS with almost identical predictive performance, for all levels of sparsity considered.

4 | LIDAR DATA EXAMPLE

In this section, we employ our Krylov subspace-based method methodology to analyze a dataset with $N = 5{,}025{,}000$ LiDAR estimates of forest canopy height in western Alaska during a 2014 Tanana Inventory Unit campaign (Cook et al., 2013; Finley et al., 2019). The two covariates of interest are tree cover at a spatial resolution of 30 m and occurrence of forest fire (Hansen et al., 2013). The tree cover is measured in percentage for peak growing season in 2010, and the fire occurrence is encoded as 1 if the fire ever occurred within the past 20 years and 0 otherwise. See Figure 5.

To characterize the relationship between the forest canopy height and the two covariates (tree cover and forest fire), we fit a spatial linear regression model with an exponential covariance function (18) parameterized by σ^2, c, and r. As discussed in Section 3.2, we consider a tapered approximation of the likelihood function using the Wendland covariance function (19) with a tapering threshold $\delta = 0.6$. Further, we encode the dataset to 200 blocks, with 25,000 observations within each block, using a mean-distance-ordered blocking structure motivated by an efficient mean-distance-ordered search algorithm for image vector quantization (Ra & Kim, 1993). The idea is based on the inequality between arithmetic mean and quadratic mean $(a+b)^2 \leq 2(a^2+b^2)$. For two locations $\boldsymbol{s} = (s_1, s_2)^\top$ and $\boldsymbol{s}' = (s_1', s_2')^\top$, we define $w = (s_1 + s_2)/\sqrt{2}$ and $w' = (s_1' + s_2')/\sqrt{2}$. Then,

$$\|\boldsymbol{s} - \boldsymbol{s}'\| = \sqrt{(s_1 - s_1')^2 + (s_2 - s_2')^2} \geq (|s_1 + s_2 - s_1' - s_2'|)/\sqrt{2} = |w - w'|.$$

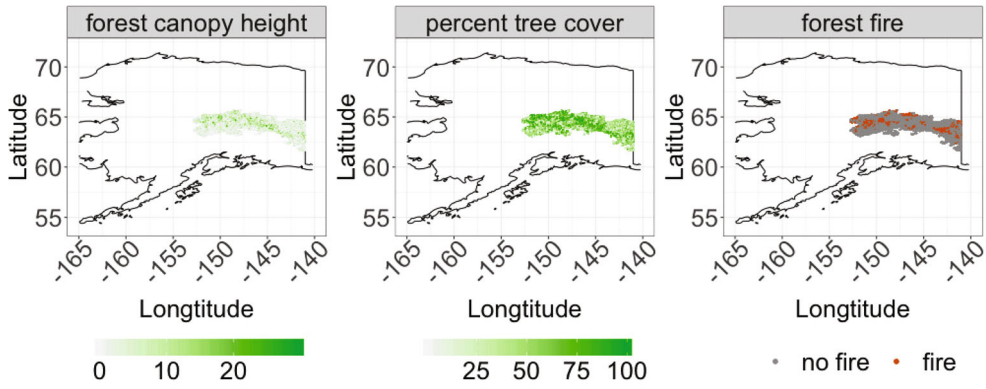

FIGURE 5 Maps for forest canopy height, tree cover and forest fire in west Alaska

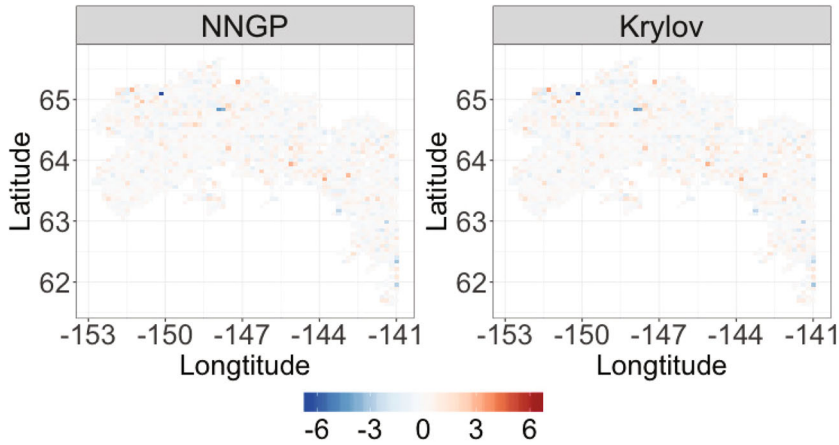

FIGURE 6 Residuals maps by the Krylov method (left) and the nearest-neighbor Gaussian process method (right)

That is, the Euclidean distance between two locations is bounded from below by a multiple of the absolute difference of the mean of the coordinate vectors. The mean-distance-ordered blocking scheme is outlined as follows: (1) let $\boldsymbol{w} = (w_1, \ldots, w_N)^\top$ be the code vector of $(\boldsymbol{s}'_1, \ldots, \boldsymbol{s}'_N)$, where $\boldsymbol{s}_i = (s_{i1}, s_{i2})^\top$ and $w_i = (s_{i1} + s_{i2})/\sqrt{2}$; (2) order the dataset according to the $\boldsymbol{w}$ ordering; (3) collapse adjacent observations to subsets of an approximately equal size. The purpose of the blocking is to divide the whole dataset into several blocks, which reduces the computational burden.

We use both the Krylov and the NNGP methods for parameter estimation and prediction. The regression coefficients include an intercept and two slopes, denoted as β_0, $\beta_{\text{tree cover}}$, and β_{fire}. Table 2 shows that both the Krylov method and the NNGP method yield comparable parameter estimates and prediction, which is also reflected in Figure 6. The residual maps for the hold-out dataset using the Krylov and the NNGP method are very similar. Indeed, the root mean squared prediction error (RMSPE) of the 25,000 hold-out observations for the Krylov method is only slightly smaller than the RMSPE for the NNGP method.

TABLE 2 Point estimates of regression coefficients and covariance parameters and root mean squared prediction error (RMSPE) for the Krylov and nearest-neighbor Gaussian process (NNGP) methods

Method	β_0	$\beta_{\text{tree cover}}$	β_{fire}	σ^2	c	r	RMSPE
Krylov	2.398	0.022	0.747	18.012	0.067	0.201	1.707
NNGP	2.429	0.022	0.54	19.455	0.053	0.183	1.709

5 | CONCLUSION

In this paper, we have studied both the theoretical and computational aspects of the Krylov subspace methods in the context of parameter estimation for large spatial datasets. The approximation error is shown to tend to zero under the increasing domain asymptotic framework and the consistency of the parameter estimators is established. The computational complexity of the proposed method is $O(N \log N)$ for sparse covariance matrices and $O(N^2 \log N)$ for dense covariance matrices.

Besides Krylov subspace methods, a Chebyshev polynomial approximation can also be used for approximating the likelihood function for large spatial dataset (Han et al., 2015). Moreover, a direct link can be established between Gaussian Markov random field approximation and the Chebyshev polynomial approximation. In terms of computational efficiency, the current form of our code is implemented in R, deploying the proposed method in Matlab, Python, or C++ could be considered in the future (Gardner et al., 2018; Wang et al., 2019). In terms of memory, an $\mathcal{O}(mN + lN)$ space is required to guarantee the consistency of parameters, where $m = \mathcal{O}(\log N)$ and $l = \mathcal{O}(\log N)$. The practical choice of m is an open question. Ideally, if we can obtain the level of approximation error for some m at a sequence of preselected parameter values of $\boldsymbol{\theta}$, we can choose m in a data-driven manner. However, this approach may not be computationally feasible since the calculation of the original log-likelihood function is time-consuming for large sample sizes.

ACKNOWLEDGEMENTS

The authors thank the Editor, the Associate Editor and the referees for their helpful comments. The research of Haonan Wang was partially supported by National Science Foundation grants DMS0-1737795, DMS-1923142 and CNS-1932413. This work is in part supported by the U.S. Geological Survey under a Grant/Cooperative Agreement. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the opinions or policies of the U.S. Geological Survey. Mention of trade names or commercial products does not constitute their endorsement by the U.S. Geological Survey.

ORCID

Tingjin Chu https://orcid.org/0000-0001-9849-8369

REFERENCES

Anitescu, M., Chen, J., & Wang, L. (2012). A matrix-free approach for solving the parametric Gaussian process maximum likelihood problem. *SIAM Journal on Scientific Computing*, *34*, A240–A262.

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

Bolin, D., & Kirchner, K. (2020). The rational SPDE approach for Gaussian random fields with general smoothness. *Journal of Computational and Graphical Statistics*, *29*, 274–285.

Bolin, D., & Wallin, J. (2016). Spatially adaptive covariance tapering. *Spatial Statistics*, *18*, 163–178.

Boutsidis, C., Drineas, P., Kambadur, P., Kontopoulou, E.-M., & Zouzias, A. (2017). A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and Its Applications*, *533*, 95–117.

Bradley, J. R., Cressie, N., & Shi, T. (2016). A comparison of spatial predictors when datasets could be very large. *Statistics Surveys*, *10*, 100–131.

Chen, J. (2011). A deflated version of the block conjugate gradient algorithm with an application to Gaussian process maximum likelihood estimation. Preprint ANL/MCS-P1927-0811 (p. 415). Argonne National Laboratory.

Chen, J. (2013). On the use of discrete Laplace operator for preconditioning kernel matrices. *SIAM Journal on Scientific Computing*, *35*, A577–A602.

Chow, E., & Saad, Y. (2014). Preconditioned Krylov subspace methods for sampling multivariate Gaussian distributions. *SIAM Journal on Scientific Computing*, *36*, A588–A608.

Chu, T. (in press). Mixed domain asymptotics for geostatistical processes. *Statistica Sinica*.

Chu, T., Zhu, J., & Wang, H. (2011). Penalized maximum likelihood estimation and variable selection in geostatistics. *The Annals of Statistics*, *39*, 2607–2625.

Cook, B., Nelson, R., Middleton, E., Morton, D., McCorkel, J., Masek, J., Ranson, K., Ly, V., & Montesano, P. (2013). NASA Goddard's LiDAR, hyperspectral and thermal (G-LiHT) airborne imager. *Remote Sensing*, *5*, 4045–4066.

Cressie, N. (1993). *Statistics for spatial data*. Wiley, revised.

Cressie, N., & Johannesson, G. (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society, Series B*, *70*, 209–226.

Cressie, N., Shi, T., & Kang, E. (2010). Fixed rank filtering for spatio-temporal data. *Journal of Computational and Graphical Statistics*, *19*, 724–745.

Cunningham, J. P., Shenoy, K. V., & Sahani, M. (2008). Fast Gaussian process methods for point process intensity estimation. *Proceedings of the 25th International Conference on Machine Learning* (pp. 192–199).

Cutajar, K., Osborne, M., Cunningham, J., & Filippone, M. (2016). Preconditioning kernel matrices. *Proceedings of the International Conference on Machine Learning* (pp. 2529–2538).

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

Dong, K., Eriksson, D., Nickisch, H., Bindel, D., & Wilson, A. G. (2017). *Scalable log determinants for Gaussian process kernel learning*. In *Advances in neural information processing systems*. (pp. 6327–6337). Curran Associates Inc.

Du, J., Zhang, H., & Mandrekar, V. S. (2009). Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *Annals of Statistics*, *37*, 3330–3361.

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., & Banerjee, S. (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, *28*, 1–14.

Finley, A. O., Sang, H., Banerjee, S., & Gelfand, A. E. (2009). Improving the performance of predictive process modeling for large datasets. *Computational Statistics & Data Analysis*, *53*, 2873–2884.

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

Furrer, R., & Gerber, F. (2008). spam: Sparse matrix. *R package version 0.14-1*.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., & Wilson, A. G. (2018). *Gpytorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration*. In *Advances in neural information processing systems* (pp. 7576–7586). Curran Associates Inc.

Gneiting, T. (2002). Compactly supported correlation functions. *Journal of Multivariate Analysis*, *83*, 493–508.

Golub, G. H., & Van Loan, C. F. (2012). *Matrix computations* (Vol. *3*). Johns Hopkins University Press.

Golub, G. H., & Welsch, J. H. (1969). Calculation of Gauss quadrature rules. *Mathematics of Computation*, *23*, 221–230.

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

Han, I., Malioutov, D., & Shin, J. (2015). Large-scale log-determinant computation through stochastic chebyshev expansions. *Proceedings of the International Conference on Machine Learning* (pp. 908–917).

Hansen, M. C., Potapov, P. V., Moore, R., Hancher, M., Turubanova, S., Tyukavina, A., Thau, D., Stehman, S., Goetz, S., Loveland, T. R., & Kommareddy, A. (2013). High-resolution global maps of 21st-century forest cover change. *Science*, *342*, 850–853.

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

Herrmann, L., Kirchner, K., & Schwab, C. (2020). Multilevel approximation of Gaussian random fields: Fast simulation. *Mathematical Models and Methods in Applied Sciences*, *30*, 181–223.

Hutchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, *19*, 433–450.

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

Kaufman, C. G., Schervish, M. J., & Nychka, D. W. (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, *103*, 1545–1555.

Lahiri, S. (2003). Central limit theorems for weighted sums of a spatial process under a class of stochastic and fixed designs. *Sankhya, Series A*, *65*, 356–388.

Lindgren, F., Rue, H., & Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: The stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*, 423–498.

Lipton, R. J., Rose, D. J., & Tarjan, R. E. (1979). Generalized nested dissection. *SIAM Journal on Numerical Analysis*, *16*, 346–358.

Litvinenko, A., Kriemann, R., Genton, M. G., Sun, Y., & Keyes, D. E. (2020). HLIBCov: Parallel hierarchical matrix approximation of large covariance matrices and likelihoods with applications in parameter identification. *MethodsX*, *7*, 100600.

Litvinenko, A., Sun, Y., Genton, M. G., & Keyes, D. E. (2019). Likelihood approximation with hierarchical matrices for large spatial datasets. *Computational Statistics & Data Analysis*, *137*, 115–132.

Mardia, K. V., & Marshall, R. J. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, *71*, 135–146.

Minden, V., Damle, A., Ho, K. L., & Ying, L. (2017). Fast spatial Gaussian process maximum likelihood estimation via skeletonization factorizations. *Multiscale Modeling & Simulation*, *15*, 1584–1611.

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

Ra, S.-W., & Kim, J.-K. (1993). A fast mean-distance-ordered partial codebook search algorithm for image vector quantization. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, *40*, 576–579.

Rue, H., & Held, L. (2005). *Gaussian Markov random fields: Theory and applications*. Chapman & Hall/CRC Press.

Rue, H., Martino, S., & Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *71*, 319–392.

Shewchuk, J. R. (1994). An introduction to the conjugate gradient method without the agonizing pain.

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

Stein, M. L. (1999). *Interpolation of spatial data: Some theory for kriging*. Springer.

Stein, M. L., Chen, J., & Anitescu, M. (2012). Difference filter preconditioning for large covariance matrices. *SIAM Journal on Matrix Analysis and Applications*, *33*, 52–72.

Stein, M. L., Chen, J., & Anitescu, M. (2013). Stochastic approximation of score functions for Gaussian processes. *The Annals of Applied Statistics*, *7*, 1162–1191.

Sun, Y., Li, B., & Genton, M. (2012). *Geostatistics for large datasets*. In *Advances and challenges in space-time modelling of natural events* (pp. 55–77). Springer.

Ubaru, S., Chen, J., & Saad, Y. (2017). Fast estimation of $tr(f(A))$ via stochastic lanczos quadrature. *SIAM Journal on Matrix Analysis and Applications*, *38*, 1075–1099.

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

Wang, H., & Zhu, J. (2009). Variable selection in spatial regression via penalized least squares. *Canadian Journal of Statistics*, *37*, 607–624.

Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K. Q., & Wilson, A. G. (2019). *Exact Gaussian processes on a million data points*. In *Advances in neural information processing systems* (pp. 14648–14659). Curran Associates Inc.

Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, *4*, 389–396.

Zhang, Y., & Leithead, W. E. (2007). Approximate implementation of the logarithm of the matrix determinant in Gaussian process regression. *Journal of Statistical Computation and Simulation*, *77*, 329–348.

How to cite this article: Liu, J., Chu, T., Zhu, J., & Wang, H. (2022). Large spatial data modeling and analysis: A Krylov subspace approach. *Scandinavian Journal of Statistics*, *49*(3), 1115–1143. https://doi.org/10.1111/sjos.12555

APPENDIX. TECHNICAL DETAILS

A.1 Proof of Theorem 1

Proof. Note that

$$\begin{aligned}\ell(\boldsymbol{\theta};\boldsymbol{y}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y}) &= -(1/2)\boldsymbol{y}^\top(\boldsymbol{z}-\boldsymbol{z}_l) = -(1/2)\boldsymbol{z}^\top\boldsymbol{\Gamma}(\boldsymbol{z}-\boldsymbol{z}_l)\\ &= -(1/2)(\boldsymbol{z}-\boldsymbol{z}_l)^\top\boldsymbol{\Gamma}(\boldsymbol{z}-\boldsymbol{z}_l) - (1/2)\boldsymbol{z}_l^\top\boldsymbol{\Gamma}(\boldsymbol{z}-\boldsymbol{z}_l)\\ &= -(1/2)(\boldsymbol{z}-\boldsymbol{z}_l)^\top\boldsymbol{\Gamma}(\boldsymbol{z}-\boldsymbol{z}_l) \equiv -(1/2)\|\boldsymbol{z}-\boldsymbol{z}_l\|_{\boldsymbol{\Gamma}}^2. \end{aligned} \tag{A1}$$

The second term in (A1) vanishes since $\boldsymbol{z}_l \in \mathcal{K}_l(\boldsymbol{\Gamma},\boldsymbol{y})$ and $\boldsymbol{\Gamma}(\boldsymbol{z}-\boldsymbol{z}_l) = \boldsymbol{y} - \boldsymbol{\Gamma}\boldsymbol{z}_l = \boldsymbol{r}_l$ is orthogonal to $\mathcal{K}_l(\boldsymbol{\Gamma},\boldsymbol{y})$. Thus, we have

$$|\ell(\boldsymbol{\theta};\boldsymbol{y}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y})| \le \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^l \|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 \le \left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l \|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 \equiv g^{(l)}(\boldsymbol{\theta}),$$

where $\|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 = \boldsymbol{y}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{y}$.

For ease of notation, we omit $\boldsymbol{y}$ in the log-likelihood functions. Let $\delta_n = N_n^\alpha$, where $\alpha \in [-1/2, 0]$. First, we show that, for a given constant $\epsilon > 0$, there is a constant C such that, for a sufficiently large n,

$$P\left\{\sup_{\|\boldsymbol{u}\|=C} \tilde{\ell}^{(l)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) < \tilde{\ell}^{(l)}(\boldsymbol{\theta}_0)\right\} \ge 1-\epsilon,$$

where $\boldsymbol{u} \in \mathbb{R}^q$. We have

$$\begin{aligned}\tilde{\ell}^{(l)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta}_0) &= \{\ell(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \ell(\boldsymbol{\theta}_0)\} + \left\{\ell(\boldsymbol{\theta_0}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta_0})\right\}\\ &\quad - \left\{\ell(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u})\right\}\\ &\le \{\ell(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \ell(\boldsymbol{\theta}_0)\} + g^{(l)}(\boldsymbol{\theta}_0) + g^{(l)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}).\end{aligned}$$

By Taylor's expansion, we obtain

$$\ell(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \ell(\boldsymbol{\theta}_0) = \delta_n\ell'(\boldsymbol{\theta}_0)^\top\boldsymbol{u} - (1/2)N_n\delta_n^2\boldsymbol{u}^\top\boldsymbol{J}(\boldsymbol{\theta}_0)\boldsymbol{u}\{1 + o_p(1)\}. \tag{A2}$$

Under (A1)–(A5), using lemma 1 in Chu et al. (2011), we have $\ell'(\boldsymbol{\theta}_0) = O_p(N_n^{1/2})$. Thus, the first term of (A2) is $O_p(N_n^{1/2}\delta_n \boldsymbol{u})$ and the second term of (A2) is $O_p(N_n\delta_n^2 \boldsymbol{u}^\top \boldsymbol{u})$. For $\alpha = -1/2$ and a sufficiently large C, the second term dominates the first term in (A2) for all n. If $\alpha > -1/2$, the second term dominates the first term in (A2) for sufficiently large n.

Since $\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1} < 1$, there exists an l such that $\delta_n \geq \left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^{l/2}$. As $\|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 \leq \|\boldsymbol{\Gamma}\|_2 \boldsymbol{y}^\top \boldsymbol{y} = O_p(N_n)$ by (A2), we have

$$g^{(l)}(\boldsymbol{\theta}) = \left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l \|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 = O_p\left(\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l N_n\right) = O_p(N_n\delta_n^2),$$

which is dominated by the second term of (A2), for a sufficiently large C.

In the special case when $l > \frac{\sqrt{\kappa_0}+1}{2}\log N_n$, we have $\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l = O(N_n^{-1})$. Thus,

$$g^{(l)}(\boldsymbol{\theta}) = \left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l \|\boldsymbol{z}\|_{\boldsymbol{\Gamma}}^2 = O_p(1).$$

For $\delta_n = N_n^{-1/2}$, we have $\|\widehat{\boldsymbol{\theta}}^{(l)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$. ∎

A.2 Proof of Theorem 2

In the following Lemmas 1–6, we provide some technical results for establishing the convergence results of the algorithms, which will be used in the proofs of Theorems 2 and 3.

Lemma 1. *Under Assumption (A1), we have*

$$\operatorname{tr}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\right) = \operatorname{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota) = \operatorname{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota),$$

and

$$\operatorname{tr}\left(\frac{\partial^2 \log \boldsymbol{\Gamma}}{\partial \theta_\iota \partial \theta_{\iota'}}\right) = \operatorname{tr}(-\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota'} + \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota\iota'}) = \operatorname{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'} - \boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota\iota'}),$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix whose diagonal elements are the eigenvalues of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Lambda}_\iota = \partial\boldsymbol{\Lambda}/\partial\theta_\iota$ and $\boldsymbol{\Lambda}_{\iota\iota'} = \partial^2\boldsymbol{\Lambda}/\partial\theta_\iota\partial\theta_{\iota'}$.

Proof. Consider the eigen decomposition $\boldsymbol{\Gamma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$, we have

$$\begin{aligned}
\operatorname{tr}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\right) &= \operatorname{tr}\left(\frac{\partial \boldsymbol{Q}}{\partial \theta_\iota}\log\boldsymbol{\Lambda}\boldsymbol{Q}^\top + \boldsymbol{Q}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_\iota}\boldsymbol{Q}^\top + \boldsymbol{Q}\log\boldsymbol{\Lambda}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_\iota}\right)\\
&= \operatorname{tr}\left(\left(\boldsymbol{Q}^\top\frac{\partial\boldsymbol{Q}}{\partial\theta_\iota} + \frac{\partial\boldsymbol{Q}^\top}{\partial\theta_\iota}\boldsymbol{Q}\right)\log\boldsymbol{\Lambda} + \boldsymbol{Q}\frac{\partial\log\boldsymbol{\Lambda}}{\partial\theta_\iota}\boldsymbol{Q}^\top\right)\\
&= \operatorname{tr}\left(\boldsymbol{Q}\frac{\partial\log\boldsymbol{\Lambda}}{\partial\theta_\iota}\boldsymbol{Q}^\top\right) = \operatorname{tr}\left(\frac{\partial\log\boldsymbol{\Lambda}}{\partial\theta_\iota}\right) = \operatorname{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota),
\end{aligned}$$

$$\begin{aligned}\operatorname{tr}\left(\frac{\partial^2 \log \boldsymbol{\Gamma}}{\partial\theta_\iota \partial\theta_{\iota'}}\right) &= \operatorname{tr}\left(\frac{\partial}{\partial\theta_{\iota'}}\left(\frac{\partial \boldsymbol{Q}}{\partial\theta_\iota}\log\boldsymbol{\Lambda}\boldsymbol{Q}^\top + \boldsymbol{Q}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_\iota}\boldsymbol{Q}^\top + \boldsymbol{Q}\log\boldsymbol{\Lambda}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_\iota}\right)\right)\\ &= \operatorname{tr}\left(\frac{\partial^2 \boldsymbol{Q}}{\partial\theta_\iota\partial\theta_{\iota'}}\log\boldsymbol{\Lambda}\boldsymbol{Q}^\top + \frac{\partial \boldsymbol{Q}}{\partial\theta_\iota}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_{\iota'}}\boldsymbol{Q}^\top + \frac{\partial \boldsymbol{Q}}{\partial\theta_\iota}\log\boldsymbol{\Lambda}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_{\iota'}}\right)\\ &\quad + \operatorname{tr}\left(\frac{\partial \boldsymbol{Q}}{\partial\theta_{\iota'}}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_\iota}\boldsymbol{Q}^\top + \boldsymbol{Q}\frac{\partial^2 \log\boldsymbol{\Lambda}}{\partial\theta_\iota\partial\theta_{\iota'}}\boldsymbol{Q}^\top + \boldsymbol{Q}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_\iota}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_{\iota'}}\right)\\ &\quad + \operatorname{tr}\left(\frac{\partial \boldsymbol{Q}}{\partial\theta_{\iota'}}\log\boldsymbol{\Lambda}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_\iota} + \boldsymbol{Q}\frac{\partial \log\boldsymbol{\Lambda}}{\partial\theta_{\iota'}}\frac{\partial \boldsymbol{Q}^\top}{\partial\theta_\iota} + \boldsymbol{Q}\log\boldsymbol{\Lambda}\frac{\partial^2 \boldsymbol{Q}^\top}{\partial\theta_\iota\partial\theta_{\iota'}}\right) +\\ &= \operatorname{tr}\left(-\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota'} + \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota\iota'}\right).\end{aligned}$$

Since $\log\det\boldsymbol{\Gamma} = \log\det\boldsymbol{\Lambda}$, by taking first- and second-order derivatives with respect to θ on both sides, we have $\operatorname{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota) = \operatorname{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota)$ and $\operatorname{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'} - \boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota\iota'}) = \operatorname{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota'} - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota\iota'})$. ∎

For ease of notation, we define an intermediate approximation to the log-likelihood function

$$\tilde{\ell}^{(N_v)}(\boldsymbol{\theta}) = -(N/2)\log(2\pi) - \frac{1}{2N_v}\sum_{i=1}^{N_v}\boldsymbol{\chi}_i^\top \log(\boldsymbol{\Gamma})\boldsymbol{\chi}_i - (1/2)\boldsymbol{y}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{y}, \tag{A3}$$

where $\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_{N_v}$ are *i.i.d.* Rademacher random variables.

Lemma 2. *Under Assumption (A1), we have*

$$\widetilde{\boldsymbol{I}}_n^{(N_v)}(\boldsymbol{\theta}) = \boldsymbol{I}_n(\boldsymbol{\theta}), \quad \text{for} \quad N_v = 1, 2, \ldots, \tag{A4}$$

where $\widetilde{\boldsymbol{I}}_n^{(N_v)}(\boldsymbol{\theta}) = \mathbb{E}\{-(\tilde{\ell}^{(N_v)})''(\boldsymbol{\theta}, \boldsymbol{\theta})\}$.

Proof. We have

$$\begin{aligned}\mathbb{E}\left(-\frac{\partial^2\tilde{\ell}^{(N_v)}(\boldsymbol{\theta})}{\partial\theta_\iota\partial\theta_{\iota'}}\right) &= \frac{1}{2N_v}\sum_{i=1}^{N_v}\operatorname{tr}\left(\frac{\partial^2 \log\boldsymbol{\Gamma}}{\partial\theta_\iota\partial\theta_{\iota'}}\right) + \frac{1}{2}\operatorname{tr}(\boldsymbol{\Gamma}^{\iota\iota'}\boldsymbol{\Gamma})\\ &= \frac{1}{2}\operatorname{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_\iota\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota'} - \boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{\iota\iota'}) - \frac{1}{2}\operatorname{tr}(\boldsymbol{\Gamma}^{\iota\iota'}\boldsymbol{\Gamma}) = \frac{1}{2}\operatorname{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'})\end{aligned}$$

where $\boldsymbol{\Gamma}^{\iota\iota'} = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'} + \boldsymbol{\Gamma}_{\iota'}\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota - \boldsymbol{\Gamma}_{\iota\iota'})\boldsymbol{\Gamma}^{-1}$. ∎

Lemma 3. *Under Assumption (A1), we have*

$$\operatorname{Var}(\tilde{\ell}^{(N_v)'}(\boldsymbol{\theta})) \preccurlyeq \left(1 + \frac{1}{N_v}\right)\boldsymbol{I}_n(\boldsymbol{\theta}). \tag{A5}$$

Here, we write $\boldsymbol{A} \preccurlyeq \boldsymbol{B}$ if $\boldsymbol{B} - \boldsymbol{A}$ is positive semi-definite.

Proof. First, we have $\operatorname{Var}(\tilde{\ell}^{(N_v)'}(\boldsymbol{\theta})) = \boldsymbol{I}_n(\boldsymbol{\theta}) + \frac{1}{4N_v}\boldsymbol{D}(\boldsymbol{\theta})$, where the (ι, ι')th element of $\boldsymbol{D}(\boldsymbol{\theta})$

$$\begin{aligned}\boldsymbol{D}_{\iota\iota'}(\boldsymbol{\theta}) &= \operatorname{Cov}\left(\boldsymbol{\chi}_1^\top\frac{\partial \log\boldsymbol{\Gamma}}{\partial\theta_\iota}\boldsymbol{\chi}_1, \boldsymbol{\chi}_1^\top\frac{\partial \log\boldsymbol{\Gamma}}{\partial\theta_{\iota'}}\boldsymbol{\chi}_1\right)\\ &= \sum_{i=1}^{N_n}\sum_{j=1}^{N_n}\sum_{l=1}^{N_n}\sum_{m=1}^{N_n}\operatorname{Cov}\left(\chi_{1,i}\chi_{1,j}\left(\frac{\partial \log\boldsymbol{\Gamma}}{\partial\theta_\iota}\right)_{i,j}, \chi_{1,l}\chi_{1,m}\left(\frac{\partial \log\boldsymbol{\Gamma}}{\partial\theta_{\iota'}}\right)_{l,m}\right)\end{aligned}$$

$$
\begin{aligned}
&= \sum_{i\neq j}\mathrm{Cov}\left(\chi_{1,i}\chi_{1,j}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,j}, \chi_{1,i}\chi_{1,j}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)_{i,j}\right)\\
&\quad + \sum_{i\neq j}\mathrm{Cov}\left(\chi_{1,i}\chi_{1,j}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,j}, \chi_{1,i}\chi_{1,j}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)_{j,i}\right)\\
&= \mathrm{tr}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right) + \mathrm{tr}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}^{\top}\right) - 2\sum_{i=1}^{N_n}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,i}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)_{i,i},
\end{aligned}
$$

where $\chi_{1,i}$ is the ith element of χ_i and $\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,j}$ is the (i,j)th element of $\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)$. Further, we note that

$$
\begin{aligned}
&\mathrm{tr}\left(\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)\right)\\
&= \mathrm{tr}\Bigg\{\left(\boldsymbol{Q}_t \log \boldsymbol{\Lambda}\boldsymbol{Q}^{\top} + \boldsymbol{Q}\frac{\partial \log \boldsymbol{\Lambda}}{\partial \theta_t}\boldsymbol{Q}^{\top} + \boldsymbol{Q}\log \boldsymbol{\Lambda}\boldsymbol{Q}_t^{\top}\right)\cdot\\
&\quad \left(\boldsymbol{Q}_{t'} \log \boldsymbol{\Lambda}\boldsymbol{Q}^{\top} + \boldsymbol{Q}\frac{\partial \log \boldsymbol{\Lambda}}{\partial \theta_{t'}}\boldsymbol{Q}^{\top} + \boldsymbol{Q}\log \boldsymbol{\Lambda}\boldsymbol{Q}_{t'}^{\top}\right)\Bigg\}\\
&= \mathrm{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_t\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{t'} + \boldsymbol{Q}_t \log \boldsymbol{\Lambda}\boldsymbol{Q}^{\top}\boldsymbol{Q}_{t'} \log \boldsymbol{\Lambda}\boldsymbol{Q}^{\top} + \log \boldsymbol{\Lambda} \log \boldsymbol{\Lambda}\boldsymbol{Q}_{t'}^{\top}\boldsymbol{Q}_t\\
&\quad + \log \boldsymbol{\Lambda} \log \boldsymbol{\Lambda}\boldsymbol{Q}_t^{\top}\boldsymbol{Q}_{t'} + \boldsymbol{Q}\log \boldsymbol{\Lambda}\boldsymbol{Q}_t^{\top}\boldsymbol{Q}\log \boldsymbol{\Lambda}\boldsymbol{Q}_{t'}^{\top})\\
&= \mathrm{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_t\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{t'}) - \sum_{m=1}^{N_n}\sum_{l=1}^{N_n}(\log(\lambda_m) + \log(\lambda_l))^2(\boldsymbol{Q}_t^{\top}\boldsymbol{Q})_{(m,l)}(\boldsymbol{Q}_{t'}^{\top}\boldsymbol{Q})_{(m,l)}.
\end{aligned}
$$

Since

$$
\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_t\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{t'}) = \mathrm{tr}(\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_t\boldsymbol{\Lambda}^{-1}\boldsymbol{\Lambda}_{t'}) + \sum_{m=1}^{N_n}\sum_{l=1}^{N_n}\left(\sqrt{\frac{\lambda_m}{\lambda_l}} - \sqrt{\frac{\lambda_l}{\lambda_m}}\right)^2(\boldsymbol{Q}_t^{\top}\boldsymbol{Q})_{(m,l)}(\boldsymbol{Q}_{t'}^{\top}\boldsymbol{Q})_{(m,l)},
$$

we have

$$
\mathrm{tr}\left(\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)\right) = \mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_t\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{t'}) - \sum_{m=1}^{N_n}\sum_{l=1}^{N_n}a_{m,l}(\boldsymbol{Q}_t^{\top}\boldsymbol{Q})_{(m,l)}(\boldsymbol{Q}_{t'}^{\top}\boldsymbol{Q})_{(m,l)}
$$

where $a_{m,l} = \left(\sqrt{\frac{\lambda_m}{\lambda_l}} - \sqrt{\frac{\lambda_l}{\lambda_m}}\right)^2 + (\log(\lambda_m) + \log(\lambda_l))^2 \geq 0$ and $a_{m,l} = a_{l,m}$.

For any real numbers $u_1, \ldots, u_q$,

$$
\begin{aligned}
&\sum_{k=1}^{q}\sum_{k'=1}^{q}\sum_{m=1}^{N_n}\sum_{l=1}^{N_n}a_{m,l}u_ku_{k'}(\boldsymbol{Q}_k^{\top}\boldsymbol{Q})_{(m,l)}(\boldsymbol{Q}_{k'}^{\top}\boldsymbol{Q})_{(m,l)} = \sum_{m=1}^{N_n}\sum_{l=1}^{N_n}a_{m,l}\left\{\sum_{k=1}^{q}u_k(\boldsymbol{Q}_k^{\top}\boldsymbol{Q})_{(m,l)}\right\}^2 \geq 0\\
&\sum_{k=1}^{q}\sum_{k'=1}^{q}u_ku_{k'}\sum_{i=1}^{N_n}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,i}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_{t'}}\right)_{i,i} = \sum_{i=1}^{N_n}\left\{\sum_{k=1}^{q}u_k\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_t}\right)_{i,i}\right\}^2 \geq 0
\end{aligned}
$$

■

Lemma 4. *Under Assumption (A1) and (A4), we have*

$$
N_n^{-1/2}\tilde{\ell}^{(N_v)'}(\boldsymbol{\theta}) = O_p(1). \tag{A6}
$$

Proof. By Lemma 1, we have

$$\mathbb{E}\left(\frac{\partial \tilde{\ell}^{(N_v)}(\boldsymbol{\theta})}{\partial \theta_\iota}\right) = -\frac{1}{2}\mathrm{tr}\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\right) + \frac{1}{2}\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota) = 0.$$

Further, by (A4), $N_n^{-1}\boldsymbol{I}_n(\boldsymbol{\theta}) \to \boldsymbol{J}(\theta)$. By Lemma 3,

$$\mathrm{Var}(\tilde{\ell}^{(N_v)'}(\boldsymbol{\theta})) \preccurlyeq \left(1 + \frac{1}{N_v}\right)\boldsymbol{I}_n(\boldsymbol{\theta}),$$

so $\mathrm{Var}(\tilde{\ell}^{(N_v)'}(\boldsymbol{\theta})) = O(N_n)$, ∎

Let $\widetilde{\boldsymbol{\theta}}^{(N_v)} = \arg\ \max_\theta\{\tilde{\ell}^{(N_v)}(\boldsymbol{\theta})\}$ denote the maximizer of the approximate likelihood function (A3). The following lemma establishes the consistency of the estimator $\widetilde{\boldsymbol{\theta}}^{(N_v)}$.

Lemma 5. *Under Assumptions (A1)–(A5), there exists, with probability tending to one, a local maximizer* $\widehat{\boldsymbol{\theta}}^{(N_v)}$ *of* $\tilde{\ell}^{(N_v)}(\boldsymbol{\theta};\boldsymbol{y})$*, such that* $\|\widehat{\boldsymbol{\theta}}^{(N_v)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$.

Proof. Let $\delta_n = N_n^\alpha$, where $\alpha \in [-1/2, 0]$. To establish $\|\widetilde{\boldsymbol{\theta}}^{(N_v)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$, it suffices to show that, for a given constant $\epsilon > 0$, there is a constant C such that, for a sufficiently large n, we have

$$P\left\{\sup_{\|\boldsymbol{u}\|=C} \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) < \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0)\right\} \geq 1 - \epsilon,$$

where $\boldsymbol{u} \in \mathbb{R}^q$.

Write $h(\boldsymbol{\theta}) = -\frac{1}{2}\log|\boldsymbol{\Gamma}| + \frac{1}{2N_v}\sum_{i=1}^{N_v}\boldsymbol{\chi}_i^\top \log(\boldsymbol{\Gamma})\boldsymbol{\chi}_i$. Then

$$\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_\iota} = -\frac{1}{2}\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota) + \frac{1}{2N_v}\sum_{i=1}^{N_v}\boldsymbol{\chi}_i^\top \frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\boldsymbol{\chi}_i.$$

By Lemma 1, we have $\mathbb{E}\left(\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_\iota}\right) = 0$. In addition,

$$\begin{aligned}\mathrm{Var}\left(\frac{\partial h(\boldsymbol{\theta})}{\partial \theta_\iota}\right) &\leq \frac{1}{2N_v}\mathrm{tr}\left(\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\right)\left(\frac{\partial \log \boldsymbol{\Gamma}}{\partial \theta_\iota}\right)\right)\\ &= \frac{1}{2N_v}\left(\mathrm{tr}(\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_\iota\boldsymbol{\Gamma}^{-1}\boldsymbol{\Gamma}_{\iota'}) - \sum_{m=1}^{N_n}\sum_{l=1}^{N_n} a_{m,l}(\boldsymbol{Q}_k^\top\boldsymbol{Q})_{(m,l)}(\boldsymbol{Q}_k^\top\boldsymbol{Q})_{(m,l)}\right) = O(N_n N_v^{-1})\end{aligned}$$

where $a_{m,l} \geq 0$ is defined in Lemma 3. Hence $h'(\boldsymbol{\theta}) = O_p(N_n^{1/2}N_v^{-1/2})$.

By Taylor's expansion, we obtain

$$\begin{aligned}\tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0) &= \ell(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - \ell(\boldsymbol{\theta}_0) + h(\boldsymbol{\theta}_0) - h(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u})\\ &= \delta_n\ell'(\boldsymbol{\theta}_0)^\top\boldsymbol{u} - \frac{1}{2}N_n\delta_n^2\boldsymbol{u}^\top\boldsymbol{J}(\boldsymbol{\theta}_0)\boldsymbol{u}\{1 + o_p(1)\} - \delta_n h'(\boldsymbol{\theta}^*)^\top\boldsymbol{u}, \qquad \text{(A7)}\end{aligned}$$

where $\boldsymbol{I}_n(\boldsymbol{\theta}) = \mathbb{E}\{-\ell''(\boldsymbol{\theta},\boldsymbol{\theta})\}$, $N_n^{-1}\boldsymbol{I}_n(\boldsymbol{\theta}) \to \boldsymbol{J}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}^*$ is between $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}$. Since $\ell'(\boldsymbol{\theta}_0) = O_p(N_n^{1/2})$ and $h'(\boldsymbol{\theta}^*) = O_p(N_n^{1/2}N_v^{-1/2})$, if we further assume $\delta_n = N_n^{-1/2}$, the first and third terms of

(A7) are of order $O_p(\boldsymbol{u})$. The second term of (A7) is at the rate of $O_p(\boldsymbol{u}^\top \boldsymbol{u})$. Thus, for a sufficiently large C, the second term dominates other terms in (A7). ∎

Using Algorithm 2, we have the following approximation of the log-likelihood function

$$\tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta};\boldsymbol{y}) = -(N/2)\log(2\pi) - (1/2)\Xi_{m,N_v} - (1/2)\boldsymbol{y}^\top \boldsymbol{\Gamma}^{-1}\boldsymbol{y}, \tag{A8}$$

where Ξ_{m,N_v} is the approximate log-determinant of the covariance matrix.

Lemma 6. *Under Assumptions (A1)-(A5), there exists, with probability tending to one, a local maximizer* $\widehat{\boldsymbol{\theta}}^{(m,N_v)}$ *of* $\tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta};\boldsymbol{y})$, *such that* $\|\widehat{\boldsymbol{\theta}}^{(m,N_v)} - \boldsymbol{\theta}_0\| = O_p\left(\max\left\{\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^m, N_n^{-1/2}\right\}\right)$. *In particular, if* $m > \frac{\sqrt{\kappa_0}}{4}\log(N_n C_1)$, *where* $C_1 = \lambda_{\max}\sqrt{\kappa_0}\log(\lambda_{\max}+\lambda_{min})$, *we have* $\|\widehat{\boldsymbol{\theta}}^{(m,N_v)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$.

Proof. Continue to define δ_n as in Lemma 5. By lemma 4.4 in Ubaru et al. (2017), we have $|\frac{1}{N_v}\sum_{i=1}^{N_v}\boldsymbol{\chi}_i^\top \log(\boldsymbol{\Gamma})\boldsymbol{\chi}_i - \Xi_{m,N_v}| \le \frac{N_n C}{\rho^{2m}}$, where $\rho = \frac{\sqrt{\kappa_0}+1}{\sqrt{\kappa_0}-1}$, $C = \frac{(\lambda_{\max}-\lambda_{\min})(\sqrt{\kappa_0}-1)^2\log(\lambda_{\max}+\lambda_{\min})}{2\sqrt{\kappa_0}}$. By Assumption (A5), there exists an α such that $\delta_n > \frac{\sqrt{C}}{\rho^m}$. Thus,

$$\begin{aligned}
&\tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0)\\
&= \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0) + (\tilde{\ell}^{(N_v)}(\boldsymbol{\theta_0}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta_0}))\\
&\quad - (\tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}))\\
&\le \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0) + \left|\tilde{\ell}^{(N_v)}(\boldsymbol{\theta_0}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta_0})\right|\\
&\quad + \left|\tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u})\right|\\
&\le \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(N_v)}(\boldsymbol{\theta}_0) + O_p(N_n\delta_n^2)\\
&\le \delta_n\ell'(\boldsymbol{\theta}_0)^\top\boldsymbol{u} - \frac{1}{2}N_n\delta_n^2\boldsymbol{u}^\top\boldsymbol{J}(\boldsymbol{\theta}_0)\boldsymbol{u}\{1+o_p(1)\} - O_p(N_n^{1/2}N_v^{-1/2}\delta_n\boldsymbol{u}) + O_p(N_n\delta_n^2).
\end{aligned} \tag{A9}$$

For a sufficiently large C, the second term in (A9) dominates the other terms. In particular, if $m > \frac{\sqrt{\kappa_0}}{4}\log(N_n C_1)$, $|\frac{1}{N_v}\sum_{i=1}^{N_v}\boldsymbol{\chi}_i^\top f(A)\boldsymbol{\chi}_i - \Xi_{m,N_v}| \le \frac{N_n\lambda_{\max}\sqrt{\kappa_0}\log(\lambda_{\max}+\lambda_{\min})}{2\rho^{2m}} \le 1$. Let $\delta_n = N_n^{-1/2}$, we have $\|\widehat{\boldsymbol{\theta}}^{(m,N_v)} - \boldsymbol{\theta}_0\| = O_p(N_n^{-1/2})$. ∎

Now, we prove Theorem 2.

Proof of Theorem 2.

$$\begin{aligned}
&\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0)\\
&= \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0) + (\tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta_0}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta_0}))\\
&\quad - (\tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}))\\
&\le \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u}) - \tilde{\ell}^{(m,N_v)}(\boldsymbol{\theta}_0) + g^{(l)}(\boldsymbol{\theta}_0) + g^{(l)}(\boldsymbol{\theta}_0+\delta_n\boldsymbol{u})\\
&\le \delta_n\ell'(\boldsymbol{\theta}_0)^\top\boldsymbol{u} - \frac{1}{2}N_n\delta_n^2\boldsymbol{u}^\top\boldsymbol{J}(\boldsymbol{\theta}_0)\boldsymbol{u}\{1+o_p(1)\} - O_p(N_n^{1/2}N_v^{-1/2}\delta_n\boldsymbol{u}) + O_p(N_n\delta_n^2).
\end{aligned}$$

For a sufficiently large C, the second term in (A9) dominates the other terms. Hence we complete the proof. ∎

A.3 Proof of Theorem 3

Remark. We first show that under (A.2), (A.5), and (A.6), the least squares estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$ satisfies $\|\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}} - \boldsymbol{\beta}_0\| = O_p(N_n^{-1/2})$. In fact, $\mathbb{E}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = \boldsymbol{\beta}$ and

$$\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}(\boldsymbol{X}^\top\boldsymbol{\Gamma}^{-1}\boldsymbol{X})(\boldsymbol{X}^\top\boldsymbol{X})^{-1} \le \|\boldsymbol{\Gamma}^{-1}\|_2(\boldsymbol{X}^\top\boldsymbol{X})^{-1}.$$

By Assumption (A.2), (A.5), and (A.6), we have $\mathrm{Var}(\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}}) = O_p(N_n^{-1})$ and $\|\widehat{\boldsymbol{\beta}}_{\mathrm{OLS}} - \boldsymbol{\beta}_0\| = O_p(N_n^{-1/2})$ (Wang & Zhu, 2009). Next, we prove Theorem 3.

Proof of Theorem 3. For any given $\widetilde{\boldsymbol{\beta}}$ satisfying $\|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}_0\| = O_p(N_n^{-1/2})$, we minimize the criterion $\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}})$, where $\widetilde{\boldsymbol{y}} = \boldsymbol{y} - \boldsymbol{X}\widetilde{\boldsymbol{\beta}}$. Let $R_1(\boldsymbol{\theta}) = \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta};\boldsymbol{y}_0)$, where $\boldsymbol{y}_0 = \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0$, then

$$\begin{aligned}\tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u};\widetilde{\boldsymbol{y}}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0;\widetilde{\boldsymbol{y}}) &= \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u};\boldsymbol{y}_0) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta}_0;\boldsymbol{y}_0)\\ &\quad + R_1(\boldsymbol{\theta}_0 + \delta_n\boldsymbol{u}) - R_1(\boldsymbol{\theta}_0).\end{aligned}$$

Notice that

$$\begin{aligned}R_1(\boldsymbol{\theta}) &= \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) - \tilde{\ell}^{(l,m,N_v)}(\boldsymbol{\theta};\boldsymbol{y}_0) = \tilde{\ell}^{(l)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) - \tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y}_0)\\ &= (\tilde{\ell}^{(l)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) - \ell(\boldsymbol{\theta};\widetilde{\boldsymbol{y}})) + (\ell(\boldsymbol{\theta};\boldsymbol{y}_0) - \tilde{\ell}^{(l)}(\boldsymbol{\theta};\boldsymbol{y}_0)) + (\ell(\boldsymbol{\theta};\boldsymbol{y}_0) - \ell(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}))\\ &= (I_1) + (I_2) + (I_3).\end{aligned}$$

For (I_1), let $\widetilde{\boldsymbol{z}} = \boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta})\widetilde{\boldsymbol{y}}$ and $\widetilde{\boldsymbol{z}}_l$ be the solution from CG algorithm at the lth iteration.

$$\tilde{\ell}^{(l)}(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) - \ell(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) = -\frac{1}{2}\|\widetilde{\boldsymbol{z}} - \widetilde{\boldsymbol{z}}_l\|_{\boldsymbol{\Gamma}}^2 \le \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^l \|\widetilde{\boldsymbol{z}}\|_{\boldsymbol{\Gamma}}^2 \le \left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l \|\widetilde{\boldsymbol{z}}\|_{\boldsymbol{\Gamma}}^2.$$

Note that,

$$\begin{aligned}\|\widetilde{\boldsymbol{z}}\|_{\boldsymbol{\Gamma}}^2 &= (\boldsymbol{y} - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})^\top\boldsymbol{\Gamma}^{-1}(\boldsymbol{y} - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})\\ &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)\boldsymbol{\Gamma}^{-1}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0) + 2(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)^\top\boldsymbol{\Gamma}^{-1}(\boldsymbol{X}\boldsymbol{\beta}_0 + \boldsymbol{X}\widetilde{\boldsymbol{\beta}})\\ &\quad - (\boldsymbol{X}\boldsymbol{\beta}_0 - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})^\top\boldsymbol{\Gamma}^{-1}(\boldsymbol{X}\boldsymbol{\beta}_0 - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})\\ &= O_p(N_n) + O_p(1) + O(1) = O_p(N_n).\end{aligned}$$

From the proof of Theorem 1, both (I_1) and (I_2) are $O_p\left(\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l N_n\right)$. For (I_3), we have

$$\begin{aligned}\ell(\boldsymbol{\theta};\boldsymbol{y}_0) - \ell(\boldsymbol{\theta};\widetilde{\boldsymbol{y}}) &= (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0)^\top\boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta})(\boldsymbol{X}\boldsymbol{\beta}_0 - \boldsymbol{X}\widetilde{\boldsymbol{\beta}}) + \frac{1}{2}(\boldsymbol{X}\boldsymbol{\beta}_0 - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})^\top\boldsymbol{\Gamma}^{-1}(\boldsymbol{\theta})(\boldsymbol{X}\boldsymbol{\beta}_0 - \boldsymbol{X}\widetilde{\boldsymbol{\beta}})\\ &= O_p(1).\end{aligned}$$

Thus, $R_1(\boldsymbol{\theta})$ is of order $O_p\left(\left(\frac{\sqrt{\kappa_0}-1}{\sqrt{\kappa_0}+1}\right)^l N_n\right)$. By (32), we have the desired result. ∎

A.4 Proof on the bounded conditional number

1. There exists $\alpha > d$, such that $|\gamma(u;\boldsymbol{\theta})| = \mathcal{O}(u^{-\alpha})$, as $d \to \infty$.
2. There exist constants $u_{\min} > 0$, such that for all sampling location s_i, it satisfies $\min_{j:j\neq i} \|\boldsymbol{s}_i - \boldsymbol{s}_j\| \geq u_{\min}$, for sufficient large n.

Lemma 7. *Under (C1)and (C2), the condition number κ_0 is bounded.*

The condition (C1) is imposed on the covariance function. The Matérn covariance function satisfies (C1), since it decreases exponentially as the distance u increases. The condition (C2) is used to avoid the scenarios that too many sampling locations are added to the same small region as sample size N_n increases.

Proof. Let $u_{ii'} = \|\boldsymbol{s}_i - \boldsymbol{s}_{i'}\|_2$, and $B_m = \{i' : mc_1 < u_{ii'} \leq (m+1)c_1\}$, where c_1 is independent of n. By Assumption (C2), there exists a constant ρ, such that the number of elements in B_m is bounded by $\rho m^{d-1} c_1^d$. Under Assumption (C1) and (C2), we have

$$\begin{aligned}\|\boldsymbol{\Sigma}\|_\infty &= \max_{1\leq i\leq N_n} \sum_{i'=1}^{N_n} \left|\gamma(u_{ii'},\boldsymbol{\theta})\right| = \max_{1\leq i\leq N_n} \sum_{m=0}^{\infty} \sum_{i'\in B_m} \left|\gamma(u_{ii'},\boldsymbol{\theta})\right| \\ &\leq \max_{1\leq i\leq N_n} \sum_{m=0}^{\infty} \left\{ \rho m^{d-1} c_1^d \max_{mc_1<u\leq(m+1)c_1} |\gamma(u,\boldsymbol{\theta})| \right\} = \mathcal{O}(1).\end{aligned}$$

That is, the condition number is bounded. ∎