# Efficient Variational Inference for Log-Gaussian Cox Processes via Voronoi Tessellation

**Yanshuo Liu**[*], **Jiaheng Qu**[*]
School of Data Science
Fudan University
Shanghai, China
yanshuoliu22@gmail.com, jhqu24@m.fudan.edu.cn

**Sha Cao, Chi Zhang**
Department of Biomedical Engineering
Oregon Health
Science University
Portland, OR, US
{caosh, zhangchi}@ohsu.edu

**Nan Zhang**[†]
School of Statistics and Data Science
Shanghai University of Finance and Economics
Shanghai, China
zhangnanpku@gmail.com

## Abstract

Log-Gaussian Cox Processes (LGCP) are essential for modeling spatial point patterns but suffer from high computational costs and intractable likelihoods. We propose VoGCAM, an efficient Variational Voronoi Gaussian Coordinate Ascent Maximization framework for fitting LGCPs. Our approach first approximates the intractable integral in the LGCP likelihood using a flexible Voronoi tessellation, which incorporates observed points as integration nodes. By applying variational Gaussian approximation, we derive an Evidence Lower Bound (ELBO) that admits an explicit and closed-form expression. To optimize this objective, we develop a novel coordinate ascent algorithm that updates parameter blocks via Newton and fixed-point methods. We further enhance scalability by adopting a Nearest Neighbor Gaussian Process (NNGP) prior and utilizing the Woodbury formula to reduce matrix inversion costs. Theoretically, we prove the existence and uniqueness of the optimal solution and establish the convergence of our algorithm. Numerical experiments on synthetic and real-world data demonstrate that VoGCAM achieves a competitive balance between computational efficiency and inferential performance compared with existing methods.

## 1 Introduction

Consider a spatial point pattern $\mathbf{Y}$ consisting of a set of observations at locations $\mathcal{D} = \{\mathbf{s}_i\}_{i=1}^N$ in a bounded region $\Omega \subset \mathbb{R}^2$. Log-Gaussian Cox processes (LGCP, Møller et al. (1998)) is a popular point process for modeling the point pattern. Our goal is to infer the LGCP intensity surface $\lambda(\mathbf{s})$ at any location $\mathbf{s}$ given the observed point pattern $\mathbf{Y}$, where $\lambda(\mathbf{s})$ denotes the expected number of points per unit area at $\mathbf{s}$. In LGCP model, the log-intensity surface consists of a linear predictor and a latent Gaussian random field,

$$\log \lambda(\mathbf{s}) = \mathbf{X}(\mathbf{s})\boldsymbol{\beta} + Z(\mathbf{s}),\ Z(\mathbf{s}) \sim \mathcal{GP}\left(\mathbf{0}, K_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{s}')\right), \tag{1}$$

where $\mathbf{X}(\mathbf{s})$ is a spatial covariate of dimension $m$, $\boldsymbol{\beta} \in \mathbb{R}^m$ is the spatial regression coefficient, and $K_{\boldsymbol{\theta}}$ is the covariance kernel for the Gaussian random field $Z(\mathbf{s})$. Conditional on a realization

[*]These authors contributed equally to this work.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

of the intensity surface $\lambda(\mathbf{s})$, the point pattern $\mathbf{Y}$ follow an inhomogeneous Poisson process. This doubly stochastic, hierarchical construction provides the LGCP with greater flexibility than standard inhomogeneous Poisson models (Illian et al., 2008; Dovers et al., 2023).

The likelihood of LGCP with intensity values $\boldsymbol{\lambda}_{\mathcal{D}} = [\lambda(\mathbf{s}_1), \ldots, \lambda(\mathbf{s}_N)]$ at observed sites $\mathcal{D}$ is

$$p\left(\mathbf{Y} \mid \boldsymbol{\lambda}_{\mathcal{D}}\right) = \exp\left\{-\int_{\Omega} \lambda(\mathbf{s})\, \mathrm{d}\mathbf{s}\right\} \prod_{i=1}^{N} \lambda(\mathbf{s}_i). \tag{2}$$

The specific derivation details of the likelihood function can be found in Daley & Vere-Jones (2007). Due to the hierarchical stochastic structure and the existence of the integral term in the likelihood, the problem of fitting LGCP is quite challenging (Murray et al., 2006; Simpson et al., 2016).

In this paper, we formulate the problem of fitting LGCP within the Bayesian framework. Based on the observed point pattern, we aim to model the posterior $\lambda(\mathbf{s}) \mid \mathbf{Y}$ of the current log-intensity at any $\mathbf{s}$. Our proposed method is termed **VoGCAM** (short for **V**ariational V**o**ronoi **G**aussian **C**oordinate **A**scent **M**aximization).

We take a variational approach to modeling the target posterior. First, we approximate the integral term in the LGCP likelihood using a Voronoi tessellation approach. This method not only replaces the original intractable likelihood with an explicit surrogate likelihood, but also incorporates the points of interest as integration nodes. Next, we use variational Gaussian approximation (Challis & Barber, 2013; Wainwright et al., 2008) to obtain the optimal Gaussian approximation to the target posterior by maximizing the evidence lower bound (ELBO). The original problem of posterior distribution inference is thus transformed into an optimization problem with an explicit objective function. Moreover, we prove that the ELBO is strongly concave, and establish the existence and uniqueness of its maximizer.

To solve this optimization problem, we design a novel coordinate ascent maximization algorithm. At each iteration, we update the variational Gaussian mean and the spatial regression coefficient by the Newton method, and update the variational Gaussian covariance matrix by the fixed-point method. To achieve computational efficiency, we approximate the Gaussian random field prior with a nearest-neighbor Gaussian process (Datta et al., 2016) which induces sparse precision matrix. Additionally, we effectively apply the Woodbury formula to reduce the cost of inverting large matrices. Finally, we establish the convergence of the proposed algorithm with fixed or varying marginal variance hyperparameter and verify the results in numerical experiments.

**Related work** Many methods have been developed for fitting LGCP. Initially, approaches based on Markov Chain Monte Carlo (MCMC) posterior sampling were explored (Møller et al., 1998; Taylor et al., 2013; Shirota & Gelfand, 2017; Teng et al., 2017). However, while such MCMC-based methods offer theoretical guarantees of asymptotic accuracy, in practice, they are burdened by substantial computational cost and challenging convergence assessment (Taylor & Diggle, 2014).

Various fast approximate algorithms have been developed with computational efficiency. Teng et al. (2017) proposed a variational Bayes mean-field method to approximate the target posterior. Dovers et al. (2023) combined variational inference with low-rank matrix representations of fixed-rank kriging (Cressie & Johannesson, 2008; Zammit-Mangion & Cressie, 2021), yielding substantial speed-ups but limited capacity for modeling local structures (Stein, 2008). Beside the above variational methods, Integrated Nested Laplace Approximation (INLA) introduced by Rue et al. (2009) has become another leading approach to fitting LGCP (Lindgren & Rue, 2015; Simpson et al., 2016; Rue et al., 2017; Bachl et al., 2019; Fuglstad et al., 2019; Flagg & Hoegh, 2023). In particular, Lindgren et al. (2011) enhanced INLA's efficiency by representing the latent Gaussian field as a Gaussian Markov random field. While INLA excels in both accuracy and speed, it focuses on marginal inference and does not directly yield joint posterior distributions (Taylor & Diggle, 2014).

In the machine learning literature, many works have focused on efficiently fitting Gaussian processes. Sparse Gaussian processes and their variants are well studied, such as inducing points (Quinonero-Candela & Rasmussen, 2005; Banerjee et al., 2008) and nearest neighbor GP (NNGP, Datta et al. (2016)). Variational frameworks for these sparse models were subsequently developed by Titsias (2009); Hensman et al. (2013); van der Wilk et al. (2020); Wu et al. (2022). Leveraging these advancements, Lloyd et al. (2015) and Shirota & Banerjee (2019) addressed Cox processes where the intensity surface is connected with Gaussian random field via squared and probit link functions,

respectively. However, none of the existing works apply to the popular LGCP model which uses the exponential link function.

The rest of the paper is organized as follows. Section 2 introduces the variational method of fitting LGCP, details the Voronoi tessellation based likelihood approximation, constructs the closed-form ELBO, and examines its theoretical properties. Section 3 presents our VoGCAM algorithm for solving the optimization problem and establishes its convergence. Section 4 presents numerical experiments to evaluate the performance of the proposed algorithm and compare it with existing approaches. All the technical proofs and experimental settings are collected in the Appendix.

## 2 Variational Inference for Fitting LGCP

### 2.1 Likelihood Approximation: a Voronoi Tessellation Approach

First, we approximate the integral term in LGCP likelihood by numerical integration

$$\int_{\Omega} \lambda(\mathbf{s})\, \mathrm{d}\mathbf{s} \approx \sum_{i=1}^{n} w_i \exp\left\{\mathbf{X}(\widetilde{\mathbf{s}}_i)\boldsymbol{\beta} + Z(\widetilde{\mathbf{s}}_i)\right\}, \tag{3}$$

where $\mathcal{I} = \{\widetilde{\mathbf{s}}_i\}_{i=1}^n$ represents $n$ pre-selected deterministic integration points in $\Omega$, and $\{w_i\}_{i=1}^n$ are the corresponding weights, satisfying $\sum_{i=1}^n w_i = |\Omega|$. Plugging (3) into the likelihood (2) leads to

$$\begin{aligned} p(\mathbf{Y} \mid \boldsymbol{\lambda}_{\mathcal{D}}, \boldsymbol{\lambda}_{\mathcal{I}}) &\approx \exp\left\{-\sum_{i=1}^{n} w_i \exp\left(\mathbf{X}(\widetilde{\mathbf{s}}_i)\boldsymbol{\beta} + Z(\widetilde{\mathbf{s}}_i)\right) + \sum_{i=1}^{N} \left(\mathbf{X}(\mathbf{s}_i)\boldsymbol{\beta} + Z(\mathbf{s}_i)\right)\right\} \\ &= \exp\left\{-\mathbf{w}^{\top} \exp\left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\mathbf{Z}\right) + \mathbf{1}_N^{\top}\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\mathbf{Z}\right)\right\}, \end{aligned} \tag{4}$$

where the matrices $\widetilde{\mathbf{X}}_{n\times m}$ and $\mathbf{X}_{N\times m}$ contain the covariate values at the integration points and the observed points, respectively, with their elements defined as $\widetilde{\mathbf{X}}_{ij} = \mathbf{X}_j(\widetilde{\mathbf{s}}_i)$ and $\mathbf{X}_{ij} = \mathbf{X}_j(\mathbf{s}_i)$. The $(n+N)$-dimensional vector $\mathbf{Z}$ is the realization of $Z(\mathbf{s})$ on $\mathcal{I} \cup \mathcal{D}$, denoted as $\mathbf{Z} = [Z(\widetilde{\mathbf{s}}_1), \ldots, Z(\widetilde{\mathbf{s}}_n), Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_N)]^{\top}$. The selection matrices $\widetilde{\mathbf{A}}_{n\times(n+N)}$ and $\mathbf{A}_{N\times(n+N)}$ are defined as $\widetilde{\mathbf{A}} = [\mathbf{I}_n, \mathbf{0}]$ and $\mathbf{A} = [\mathbf{0}, \mathbf{I}_N]$, respectively; $\mathbf{1}_N$ is a vector of ones of length $N$.

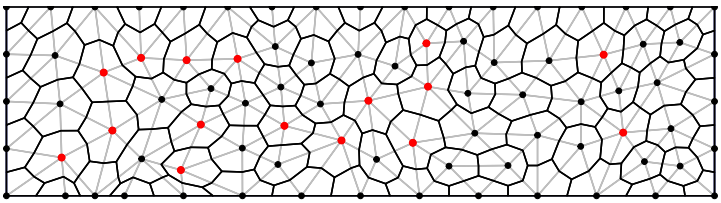


Figure 1: Voronoi tessellation on the observation region. The red solid points denote the target locations $\mathcal{S}$ and the complete set of solid points constitutes the integration locations $\mathcal{I}$ for numerical quadrature. Each integration weight is given by the area of the Voronoi cell around its corresponding integration point, and the gray lines mark the edges of the associated Delaunay triangulation.

In this study, we adopt a Voronoi-tessellation approach to determine the integration points and their corresponding weights (Simpson et al., 2016). First, we perform Delaunay triangulation on the observation area using the positions in the target location set $\mathcal{S}$ as the initial triangle vertices. Next, based on the constructed Delaunay triangulation, we can obtain the Voronoi tessellation that is geometrically dual to it. Finally, we can obtain the integration point set $\mathcal{I}$ and the corresponding weight set $\{w_i\}_{i=1}^n$ as described in Figure 1.

Compared with conventional methods, such as dividing the entire observation domain into a series of uniform grid cells (Teng et al., 2017; Dovers et al., 2023), the Voronoi tessellation offers several advantages. First, it allows for flexible control over the density of the partitioned domain by adjusting the minimum edge length of the Delaunay triangulation. Second, it ensures that information from $\mathcal{S}\setminus\mathcal{D}$ is incorporated into the likelihood function. Third, this approximation method enjoys theoretical

guarantee. Simpson et al. (2016) demonstrated that this LGCP likelihood approximation converges to the true likelihood, and further showed that the resulting posterior distribution converges to the true posterior under the Hellinger distance metric. Therefore, substituting the approximated likelihood for the original is reasonable for our subsequent analysis.

### 2.2 The Closed-form ELBO and Theoretical Properties

We begin by treating $\boldsymbol{\beta}$ as a fixed but unknown parameter and then focus on inferring the posterior the latent field realization $\mathbf{Z}_{\mathcal{S}} = [Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_p)]^\top$ at the target sites $\mathcal{S}$. Rather than considering $\mathbf{Z}_{\mathcal{S}}$ only, we infer the full latent vector $\mathbf{Z}$ which combines the values at the integration points $\mathcal{I}$, the observed points $\mathcal{D}$, and the target locations $\mathcal{S}$. The posterior $p(\mathbf{Z} \mid \mathbf{Y})$ offers comprehensive knowledge over the domain besides $\mathbf{Z}_{\mathcal{S}}$. Under the Gaussian field prior $p(\mathbf{Z}) = N(\mathbf{Z} \mid \mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}})$, all the variability in the log-intensity comes from $Z(\mathbf{s})$, and the approximated likelihood (4) can be denoted as $p(\mathbf{Y} \mid \mathbf{Z})$.

Next, we approximate the true posterior $p(\mathbf{Z} \mid \mathbf{Y})$ by a distribution $q^\star$ chosen from a tractable variational family $\mathcal{Q}$ to minimize the Kullback–Leibler divergence $\mathrm{KL}\big(q(\mathbf{Z}) \| p(\mathbf{Z} \mid \mathbf{Y})\big)$. The standard variational inference shows that minimizing the KL divergence is equivalent to maximizing ELBO

$$q^\star = \arg\max_{q \in \mathcal{Q}} \int q(\mathbf{Z}) \log \frac{p(\mathbf{Z}) p(\mathbf{Y} \mid \mathbf{Z})}{q(\mathbf{Z})} \, \mathrm{d}\mathbf{Z}. \tag{5}$$

We choose $\mathcal{Q}$ to be Gaussian such that $q(\mathbf{Z}) = N(\mathbf{Z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with the variational parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Substituting the Gaussian prior $p(\mathbf{Z}) = N(\mathbf{Z} \mid \mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}})$ and the approximated LGCP likelihood $p(\mathbf{Y} \mid \mathbf{Z})$ into the ELBO (5), we obtain an objective function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with an explicit form in terms of the unknown parameters $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$,

$$\begin{aligned} E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = &- \mathbf{w}^\top \exp\left\{ \widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\mathrm{diag}\left(\widetilde{\mathbf{A}}\boldsymbol{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\} + \mathbf{1}_N^\top (\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}) \\ &- \frac{1}{2}\boldsymbol{\mu}^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right) + \frac{1}{2}\log|\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}| + \frac{n+N}{2}, \end{aligned} \tag{6}$$

where $\mathrm{diag}(\cdot)$ is an operator that extracts the diagonal elements of a matrix to form a vector. In the following, we use the shorthand notation

$$\boldsymbol{\psi}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \exp\left\{ \widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\mathrm{diag}\left(\widetilde{\mathbf{A}}\boldsymbol{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\},$$

and sometimes just $\boldsymbol{\psi}$ without ambiguity. The derivation of $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is provided in Appendix A.

We provide an interpretation from the frequentist perspective for $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. The two terms on the right-hand side of the first line in (6) is a goodness-of-fit term, resembling the approximated LGCP log-likelihood (4) only differing by $\frac{1}{2}\mathrm{diag}(\widetilde{\mathbf{A}}\boldsymbol{\Sigma}\widetilde{\mathbf{A}}^\top)$. In the second line, the leading term penalizes deviation of $\boldsymbol{\mu}$ from $\mathbf{0}$ via a $\mathbf{K}_{\boldsymbol{\theta}}^{-1}$-weighted squared norm, while the rest three terms penalizes $\boldsymbol{\Sigma}$ for deviating from $\mathbf{K}_{\boldsymbol{\theta}}$ via a KL divergence measure; see Lemma A.1 for detail. Consequently, maximizing $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is equivalent to maximizing a penalized likelihood, thereby avoiding potential overfitting.

The following two results establish the key theoretical properties. Denote by $\mathcal{S}_p^+$ the collection of positive definite matrices of dimension $p$.

**Theorem 2.1.** *For any prior covariance matrix $\mathbf{K}_{\boldsymbol{\theta}} \in \mathcal{S}_{n+N}^+$ , the objective function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is strictly jointly concave with respect to $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$.*

The proof of Theorem 2.1 can be directly obtained from the expression of $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in (6), combined with the fact that both $-\exp\{\cdot\}$ and $\log|\cdot|$ are operations that preserve strong concavity.

Based on the strict concavity established by Theorem 2.1, we prove that the maximizer of the objective function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ uniquely exists. Its proof is in Appendix B.

**Theorem 2.2.** *The optimization problem*

$$\max_{\boldsymbol{\beta} \in \mathrm{R}^m, \boldsymbol{\mu} \in \mathrm{R}^{n+N}, \boldsymbol{\Sigma} \in \mathcal{S}_{n+N}^+} E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

*admits a unique solution, denoted by $(\boldsymbol{\beta}^\star, \boldsymbol{\mu}^\star, \boldsymbol{\Sigma}^\star)$.*

## 3 Efficient Algorithm for Optimizing the ELBO

### 3.1 Approximate prior by NNGP

In each iteration of our algorithm, evaluating the ELBO $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ requires operations on the $(n+N)$-dimensional prior covariance $\mathbf{K}_{\boldsymbol{\theta}}$, which incur $\mathcal{O}((n+N)^3)$ time and $\mathcal{O}((n+N)^2)$ storage cost. To alleviate this burden, we approximate the Gaussian process $Z(\mathbf{s})$ by a NNGP, in which each full conditional distributions is replaced by one conditioned on a small set of nearest neighbors. The resulting finite-dimensional prior has a sparse precision matrix, yielding the approximate prior

$$p(\mathbf{Z}) \approx \widetilde{p}(\mathbf{Z}) = N\left(\mathbf{Z} \mid \mathbf{0}, \sigma^2 \boldsymbol{\Gamma}^{-1}\right), \tag{7}$$

where $\sigma^2$ is the marginal variance and $\boldsymbol{\Gamma}$ is a sparse precision matrix parameterized by hyperparameters including range and smoothness. Further details on constructing $\boldsymbol{\Gamma}$ are provided in Appendix C.1.

### 3.2 Coordinate Ascent Maximization Algorithm

Substituting the NNGP approximate covariance matrix $\sigma^2\boldsymbol{\Gamma}^{-1}$ into the ELBO expression (6) yields the current objective function, which is given by

$$\begin{aligned} E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = &- \mathbf{w}^\top \boldsymbol{\psi} + \mathbf{1}_N^\top (\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}) - \frac{\sigma^{-2}}{2} \boldsymbol{\mu}^\top \boldsymbol{\Gamma} \boldsymbol{\mu} \\ &- \frac{\sigma^{-2}}{2} \mathrm{tr}(\boldsymbol{\Gamma}\boldsymbol{\Sigma}) + \frac{1}{2} \log |\sigma^{-2}\boldsymbol{\Gamma}\boldsymbol{\Sigma}| + \frac{n+N}{2}. \end{aligned} \tag{8}$$

We first consider the case where the hyperparameters $\boldsymbol{\theta}$ defining $\sigma^2$ and $\boldsymbol{\Gamma}$ are fixed. Then, the optimization problem is given by

$$\max_{\boldsymbol{\beta} \in \mathrm{R}^m, \boldsymbol{\mu} \in \mathrm{R}^{n+N}, \boldsymbol{\Sigma} \in \mathcal{S}_{n+N}^+} E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}). \tag{9}$$

Since the parameters $\boldsymbol{\beta}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are coupled in the objective function, we employ block coordinate ascent method: at each iteration, we update $\boldsymbol{\beta}$ (with $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ fixed) and $\boldsymbol{\mu}$ (with $\boldsymbol{\beta}, \boldsymbol{\Sigma}$ fixed) via Newton method, then update $\boldsymbol{\Sigma}$ (with $\boldsymbol{\beta}, \boldsymbol{\mu}$ fixed) via a fixed-point method.

**Newton method for updating $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$.** The gradients of the objective function $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ are given by

$$\nabla_{\boldsymbol{\beta}} E = \mathbf{X}^\top \mathbf{1}_N - \widetilde{\mathbf{X}}^\top (\mathbf{w} \circ \boldsymbol{\psi}), \qquad \nabla_{\boldsymbol{\mu}} E = \mathbf{A}^\top \mathbf{1}_N - \widetilde{\mathbf{A}}^\top (\mathbf{w} \circ \boldsymbol{\psi}) - \sigma^{-2}\boldsymbol{\Gamma}\boldsymbol{\mu}. \tag{10}$$

The Hessian matrices of $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ are given by

$$\nabla^2_{\boldsymbol{\beta}} E = -\widetilde{\mathbf{X}}^\top \mathrm{Diag}[\mathbf{w} \circ \boldsymbol{\psi}] \widetilde{\mathbf{X}}, \qquad \nabla^2_{\boldsymbol{\mu}} E = -\widetilde{\mathbf{A}}^\top \mathrm{Diag}[\mathbf{w} \circ \boldsymbol{\psi}] \widetilde{\mathbf{A}} - \sigma^{-2}\boldsymbol{\Gamma}. \tag{11}$$

Here, $\mathbf{x} \circ \mathbf{y}$ denotes the Hadamard product of vectors $\mathbf{x}$ and $\mathbf{y}$. The operator $\mathrm{Diag}(\cdot)$ acts on a vector, constructing a diagonal matrix with the elements of that vector on its main diagonal.

Given initial values $\boldsymbol{\beta}_0$ and $\boldsymbol{\mu}_0$, we update $\boldsymbol{\beta}$ and $\boldsymbol{\mu}$ using Newton method. The updating rules at the $k$-th iteration are given by

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - (\nabla^2_{\boldsymbol{\beta}} E_k)^{-1} \nabla_{\boldsymbol{\beta}} E_k, \qquad \boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k - (\nabla^2_{\boldsymbol{\mu}} E_k)^{-1} \nabla_{\boldsymbol{\mu}} E_k. \tag{12}$$

The major computational cost of Newton method lies in solving the linear systems involving the Hessians. Since the negative Hessians are positive definite, these systems can be solved efficiently using the conjugate gradient method in practice.

**Fixed-point method for updating $\boldsymbol{\Sigma}$.** For the matrix variable $\boldsymbol{\Sigma}$, using Newton method for updates would involve solving a large-scale linear system. Instead, we design a fixed-point iteration method. Since the objective function $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is strongly concave with respect to $\boldsymbol{\Sigma}$, finding the optimal $\boldsymbol{\Sigma}^\star$ is equivalent to solving the equation $\nabla_{\boldsymbol{\Sigma}} E = \mathbf{0}$. The equation can be simplified as

$$-\widetilde{\mathbf{A}}^\top \mathrm{Diag}(\mathbf{w} \circ \boldsymbol{\psi}) \widetilde{\mathbf{A}} - \sigma^{-2}\boldsymbol{\Gamma} + \boldsymbol{\Sigma}^{-1} = \mathbf{0}. \tag{13}$$

See Lemma C.1 in Appendix C.2 for its derivation.

Based on (13), we construct a fixed-point method to update $\boldsymbol{\Sigma}$. Specifically, given the initial value $\boldsymbol{\Sigma}_0 = \sigma^2\boldsymbol{\Gamma}^{-1}$, the updating rule at the $k$-th iteration is given by

$$\boldsymbol{\Sigma}_{k+1} = \left(\mathbf{D}_k + \sigma^{-2}\boldsymbol{\Gamma}\right)^{-1}, \tag{14}$$

where $\mathbf{D}_k = \widetilde{\mathbf{A}}^\top \mathrm{Diag}\left(\mathbf{w} \circ \boldsymbol{\psi}_k\right) \widetilde{\mathbf{A}}$. Both the even- and odd-indexed subsequences of $\{\boldsymbol{\Sigma}_k\}_{k\geq 0}$ generated by (14) are guaranteed to converge. Its proof is in Theorem C.2 in Appendix C.3.

**Matrix inversion with the Woodbury formula.** In each iteration, the dominant cost arises from inverting the $(n+N)$-dimensional matrix $\left(\mathbf{D}_k + \sigma^{-2}\boldsymbol{\Gamma}\right)^{-1}$ in (14), which scales as $\mathcal{O}((n+N)^3)$. To reduce the complexity, we employ the Woodbury formula after decomposing the precision matrix into a diagonal plus low-rank form,

$$\boldsymbol{\Gamma} \approx \boldsymbol{\Gamma}_{\text{diag}} + \mathbf{L}\mathbf{L}^\top, \tag{15}$$

where $\boldsymbol{\Gamma}_{\text{diag}}$ is chosen so that $\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{\text{diag}}$ remains positive definite, and $\mathbf{L}_{(N+n)\times r}$ (with $r \ll n+N$) derives from its Cholesky factor. It is necessary to retain the diagonal component. By its definition (14), $\mathbf{D}_k$ is an $(n+N)$-dimensional diagonal matrix whose last $N$ diagonal elements are zero. The diagonal $\boldsymbol{\Gamma}_{\text{diag}}$ ensures that $\widetilde{\mathbf{D}}_k$ in (16) is invertible such that the Woodbury formula is applicable.

With the approximation of $\boldsymbol{\Gamma}$ in (15), the fixed-point iteration is

$$\boldsymbol{\Sigma}_{k+1} \approx \left(\mathbf{D}_k + \sigma^{-2}\boldsymbol{\Gamma}_{\text{diag}} + \sigma^{-2}\mathbf{L}\mathbf{L}^\top\right)^{-1} \triangleq \left(\widetilde{\mathbf{D}}_k + \sigma^{-2}\mathbf{L}\mathbf{L}^\top\right)^{-1}, \tag{16}$$

where $\widetilde{\mathbf{D}}_k = \mathbf{D}_k + \sigma^{-2}\boldsymbol{\Gamma}_{\text{diag}}$ is an invertible diagonal matrix. It follows from the Woodbury formula that

$$\boldsymbol{\Sigma}_{k+1} = \widetilde{\mathbf{D}}_k^{-1} - \sigma^{-2}\widetilde{\mathbf{D}}_k^{-1}\mathbf{L}\left(\mathbf{I}_r + \sigma^{-2}\mathbf{L}^\top\widetilde{\mathbf{D}}_k^{-1}\mathbf{L}\right)^{-1}\mathbf{L}^\top\widetilde{\mathbf{D}}_k^{-1}. \tag{17}$$

The complexity for each update becomes $\mathcal{O}\left((n+N)^2 r + (n+N)r^2 + r^3\right)$. It is significantly more efficient than the cost of the direct matrix inversion with complexity $\mathcal{O}\left((n+N)^3\right)$.

**Algorithm and its convergence guarantee.** When all hyperparameters $\boldsymbol{\theta}$ are fixed, the process of solving the optimization problem (9) is detailed in Algorithm 1.

**Algorithm 1** VoGCAM with fixed hyperparameters

**Input:** Covariate matrices $\mathbf{X}, \widetilde{\mathbf{X}}$; Selection matrices $\mathbf{A}, \widetilde{\mathbf{A}}$; Dual mesh weights $\mathbf{w}$;
Fixed hyperparameters $\boldsymbol{\theta}$; NNGP approximate precision matrix $\boldsymbol{\Gamma}$.
**Approximate** $\boldsymbol{\Gamma}$ with the diagonal plus low-rank structure given in (15).
**Initialize** $\widehat{\boldsymbol{\beta}}_0 \leftarrow \boldsymbol{\beta}_0$; $\widehat{\boldsymbol{\mu}}_0 \leftarrow \boldsymbol{\mu}_0$; $\widehat{\boldsymbol{\Sigma}}_0 \leftarrow \boldsymbol{\Sigma}_0$.
**while** $E_{\boldsymbol{\theta}}\left(\widehat{\boldsymbol{\beta}}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k\right)$ *not converged* and $k \leq k_{max}$ **do**
  **Update** $\widehat{\boldsymbol{\beta}}_k$ and $\widehat{\boldsymbol{\mu}}_k$ by Newton method (12);
  **Update** $\widehat{\boldsymbol{\Sigma}}_k$ by fixed-point method (17);
**Return** $\widehat{\boldsymbol{\beta}}^\star, \widehat{\boldsymbol{\mu}}^\star, \widehat{\boldsymbol{\Sigma}}^\star$.

The convergence analysis of Algorithm 1 is given by the following theorem. The proof is in Appendix C.

**Theorem 3.1.** *The sequences $\{\boldsymbol{\beta}_k\}_{k\geq 0}$ and $\{\boldsymbol{\mu}_k\}_{k\geq 0}$ generated by Algorithm 1 converge. Moreover, the even- and odd-indexed subsequences of the covariance iterates $\{\boldsymbol{\Sigma}_k\}_{k\geq 0}$ also converge.*

### 3.3 Hyperparameter Configuration

Prior to running the algorithm, we first need to fix all hyperparameters defining the precision matrix $\boldsymbol{\Gamma}$ (e.g. the Matérn range and smoothness) using strategies such as K-fold cross-validation (Finley et al., 2019) or empirical estimation methods based on the spatial properties (Lindgren et al., 2011). The practical procedure for determining the aforementioned hyperparameters is detailed in Appendix E. Then, we turn to estimation of the marginal variance $\sigma^2$.

**Algorithm 2** VoGCAM

**Input:** Covariate matrices $\mathbf{X}, \widetilde{\mathbf{X}}$; Selection matrices $\mathbf{A}, \widetilde{\mathbf{A}}$; Dual mesh weights $\mathbf{w}$; Fixed hyperparameters $\boldsymbol{\theta}$ except for $\sigma$; NNGP approximate precision matrix $\boldsymbol{\Gamma}$.
**Approximate** $\boldsymbol{\Gamma}$ with the diagonal plus low-rank structure given in (15).
**Initialize** $\widehat{\boldsymbol{\beta}}_0 \leftarrow \boldsymbol{\beta}_0$; $\widehat{\boldsymbol{\mu}}_0 \leftarrow \boldsymbol{\mu}_0$; $\widehat{\boldsymbol{\Sigma}}_0 \leftarrow \boldsymbol{\Sigma}_0$.
**while** $E\left(\widehat{\boldsymbol{\beta}}_k, \widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k; \widehat{\sigma}_k\right)$ *not converged* and $k \leq k_{max}$ **do**
  **Update** $\widehat{\boldsymbol{\beta}}_k$, $\widehat{\boldsymbol{\mu}}_k$ and $\widehat{\boldsymbol{\Sigma}}_k$ by Algorithm 1;
  **Update** $\widehat{\sigma}_k$ by fixed-point method (20);
**Return** $\widehat{\boldsymbol{\beta}}^\star, \widehat{\boldsymbol{\mu}}^\star, \widehat{\boldsymbol{\Sigma}}^\star, \widehat{\sigma}^\star$.

**The new ELBO that contains $\sigma$** To incorporate hierarchical modeling of the marginal variance $\sigma^2$, we augment the ELBO in (5) by including the prior $p(\sigma^2 \mid \boldsymbol{\alpha}_\sigma)$, which yields

$$\begin{aligned} E\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \sigma\right) &= \int q(\mathbf{Z}) \log \frac{p\left(\mathbf{Z}, \mathbf{Y} \mid \sigma^2\right) p\left(\sigma^2 \mid \boldsymbol{\alpha}_\sigma\right)}{q(\mathbf{Z})} \, \mathrm{d}\mathbf{Z} \\ &= E_{\boldsymbol{\theta}}\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) + \log p\left(\sigma^2 \mid \boldsymbol{\alpha}_\sigma\right), \end{aligned} \tag{18}$$

where $\boldsymbol{\alpha}_\sigma$ collects the hyperparameters of the prior on $\sigma^2$. In particular, we place an inverse-Gamma prior, $p\left(\sigma^2 \mid \boldsymbol{\alpha}_\sigma\right) = \mathrm{IG}\left(\sigma^2 \mid a, b\right)$, with shape $a$ and scale $b$. Next, combining $E_{\boldsymbol{\theta}}\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ in (8) and the inverse-Gamma prior for $\sigma^2$, we write (18) as

$$\begin{aligned} E\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \sigma\right) = &-\mathbf{w}^\top \boldsymbol{\psi} + \mathbf{1}_N^\top \left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}\right) - \frac{\sigma^{-2}}{2} \boldsymbol{\mu}^\top \boldsymbol{\Gamma} \boldsymbol{\mu} \\ &- \frac{\sigma^{-2}}{2} \mathrm{tr}\left(\boldsymbol{\Gamma}\boldsymbol{\Sigma}\right) + \frac{1}{2} \log \left|\sigma^{-2} \boldsymbol{\Gamma}\boldsymbol{\Sigma}\right| + \frac{n+N}{2} \\ &+ (a+1) \log \sigma^{-2} - b\sigma^{-2} + a \log b - \log \Gamma\left(a\right). \end{aligned} \tag{19}$$

**Fixed-point method for updating $\sigma$.** To maximize $E\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}; \sigma\right)$, we partition the parameters into two blocks, $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\sigma$, and then alternately update the two blocks. Specifically, in the $k$-th iteration, we first fix $\sigma$ at its current value, $\sigma_k$. Next, as described in Section 3.2, we can find the optimal values that maximize $E_{\boldsymbol{\theta}_k}\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$, denoted as $\left(\boldsymbol{\beta}^\star_{\sigma_k}, \boldsymbol{\mu}^\star_{\sigma_k}, \boldsymbol{\Sigma}^\star_{\sigma_k}\right)$. Then, fixing $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ at these optimal values, we update $\sigma$ by solving $\partial E\left(\boldsymbol{\beta}^\star_{\sigma_k}, \boldsymbol{\mu}^\star_{\sigma_k}, \boldsymbol{\Sigma}^\star_{\sigma_k}; \sigma\right) / \partial(\sigma^2) = 0$. This yields a fixed-point iteration update

$$\sigma_{k+1} = \sqrt{\frac{\boldsymbol{\mu}^\top_{\sigma_k} \boldsymbol{\Gamma} \boldsymbol{\mu}_{\sigma_k} + \mathrm{tr}\left(\boldsymbol{\Gamma}\boldsymbol{\Sigma}_{\sigma_k}\right) + 2b}{n + N + 2a + 2}}. \tag{20}$$

We have obtained the complete algorithm summarized in Algorithm 2. The following theorem guarantees the convergence of the update for $\sigma$. Its proof is provided in Appendix D.

**Theorem 3.2.** *For any initial value $\sigma_0 > 0$, the sequence $\{\sigma_k\}_{k \geq 0}$ generated by* (20) *converges monotonically to some $\sigma_\star > 0$.*

### 3.4 Computational Complexity

We summarize the computational complexity of the major steps in VoGCAM in Table 1. By exploiting the sparsity of the NNGP precision matrix $\boldsymbol{\Gamma}$, each conjugate gradient iteration for updating $\boldsymbol{\mu}$ costs $\mathcal{O}((n+N)M^2)$, with total cost depending on the number of iterations. For the $\boldsymbol{\Sigma}$ update, we report only the leading-order term, since $r \ll n+N$. The $\sigma^2$ update requires evaluating a quadratic form and a trace involving the sparse precision matrix $\boldsymbol{\Gamma}$, which costs $\mathcal{O}((n+N)M^2)$ given $\boldsymbol{\Sigma}$.

## 4 Experiments

Our proposed method VoGCAM is evaluated through both simulation studies and real data applications. We compare VoGCAM against several baselines: INLA (Simpson et al., 2016; Fuglstad

Table 1: Computational complexity of the major steps in VoGCAM.

| Step | Time complexity |
|---|---|
| Voronoi tessellation | $\mathcal{O}(n \log n)$ (Aurenhammer, 1991) |
| NNGP construction | $\mathcal{O}((n+N)M^3)$ (Datta et al., 2016) |
| $\boldsymbol{\beta}$ update | $\mathcal{O}(nm^2 + m^3)$ |
| $\boldsymbol{\mu}$ update | $\mathcal{O}((n+N)M^2)$ |
| $\boldsymbol{\Sigma}$ update | $\mathcal{O}((n+N)^2 r)$ |
| $\sigma^2$ update | $\mathcal{O}((n+N)M^2)$ |

et al., 2019) (via `INLA`), VIFRK (Dovers et al., 2023) (via `scampr`), and MINCON (Baddeley et al., 2015) with its quasi-likelihood variant denoted as QUASI Guan et al. (2015) (via `kppm`). Note that MINCON and QUASI do not model random effects. We additionally compare VoGCAM with VBPP (Lloyd et al., 2015; Aglietti et al., 2019) on the neuronal dataset, where no fixed effects are included, as the current VBPP implementation does not support them. All numerical simulations and comparative experiments of different methods were completed on a Windows 11 laptop equipped with an Intel Core i7-10750H processor (2.60 GHz, 12 threads) and 8 GB of memory. Moreover, all the code of VoGCAM was implemented in `R` language.

**Simulation.** We generate point patterns on the bounded region $\Omega = [0,1]^2$ based on two LGCP intensity functions, differing in the relative contributions of the fixed and random effects. See Figure 3 in Appendix E for visualization. Case I exhibits a dominating fixed effect, which is helpful to test methods' ability to recover the fixed structure. Meanwhile, Case II has a dominating random effect, providing a setting to evaluate the capability to capture stochastic patterns. Competing methods are evaluated by the training log-likelihood, predictive log-likelihood, $\ell_2$ error (distance between the fitted and true intensity functions), $\ell_{2,\log}$ error (distance between the fitted and true latent GPs), and running time. Additional details on the experimental setup, evaluation metrics, sensitivity analysis, and convergence verification of the VoGCAM algorithm are provided in Appendix E.

Table 2 reveals clear patterns of different methods in Cases I and II. VoGCAM achieves the highest training log-likelihood, indicating the closest fit to the observed data. Meanwhile, its predictive log-likelihood shows only modest improvement, which may imply slight overfitting. In contrast, INLA produces the lowest $\ell_2$ error for the intensity function, demonstrating superior recovery of high-intensity regions, while VoGCAM achieves comparable $\ell_{2,\log}$ errors, delivering accurate reconstruction across both high- and low-intensity areas. In terms of running time, VoGCAM requires a moderate amount of computation, balancing efficiency with predictive performance. Overall, the results remain consistent across the two cases, confirming that the relative ranking of the methods is robust to differences in the dominance of fixed or random effects. We further investigate the effect of the integration point selection strategy by comparing Voronoi tessellation with uniform grid cell sampling, and assess the stability of the results through 10 repeated runs under Case II. The corresponding ablation study and results across multiple runs are provided in Appendix E.

Table 2: Comparison of methods under Cases I and II. Metrics include training log-likelihood ($l_{train}$), predictive log-likelihood ($l_{pred}$), $\ell_2$ error of intensity function ($\ell_2$), $\ell_2$ error of log-intensity function ($\ell_{2,\log}$), and running time (Time in seconds). Bold numbers indicate the best value for each metric.

| Method | Case I | | | | | Case II | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $l_{train}$ | $l_{pred}$ | $\ell_2$ | $\ell_{2,\log}$ | Time | $l_{train}$ | $l_{pred}$ | $\ell_2$ | $\ell_{2,\log}$ | Time |
| VoGCAM | **16852** | **14167** | 933 | 0.36 | 1.505 | **17565** | **14916** | 1687 | **0.65** | 0.849 |
| INLA | 15233 | 14154 | **911** | **0.35** | 3.251 | 15742 | 14898 | **1679** | **0.65** | 1.948 |
| VIFRK | 15225 | 14148 | 969 | 0.36 | **0.130** | 15816 | 14845 | 1895 | 0.68 | **0.080** |
| MINCON | 14933 | 13974 | 1451 | 0.59 | 0.890 | 14616 | 13879 | 3278 | 1.44 | 1.050 |
| QUASI | 14932 | 13970 | 1457 | 0.60 | 6.780 | 14611 | 13879 | 3281 | 1.46 | 3.120 |

Figure 2 compares the true LGCP intensity with the posterior mean estimates from the baseline methods. Unlike the numerical patterns reported in Table 2, the visualizations display distinct behaviors for the two cases. In the left panel for Case I, all methods accurately capture the overall

structure of the fixed effect; VoGCAM produces a coarser reconstruction, whereas the other methods generate similar but smoother intensity surfaces. In the right panel for Case II, MINCON and QUASI fail to fit the log-intensity functions. In contrast, VoGCAM, INLA, and VIFRK recover the true intensity trends clearly, and their reconstructions are visually similar. Results on posterior variance are provided in Appendix E demonstrating competitive performance of VoGCAM in uncertainty quantification.

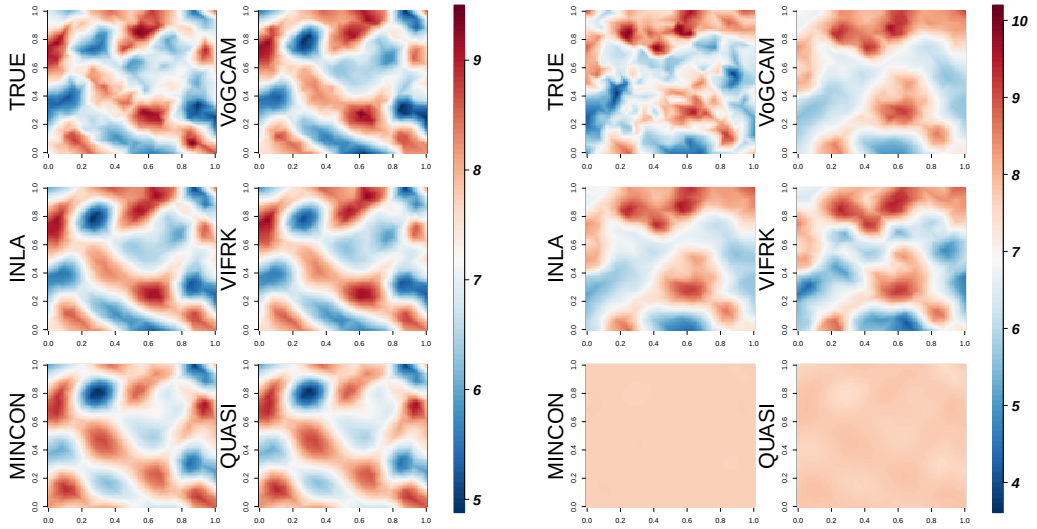



Figure 2: True and posterior mean intensity surfaces under Case I (left) and Case II (right) estimated by the five methods from the simulated point pattern datasets.

**Computational Cost Breakdown.** We profile the computational cost of VoGCAM under Case II in the simulation study. Each component is profiled separately over 10 independent runs, and we report the mean and standard error (SE) of its runtime. For iterative components, we report the total runtime over all iterations of each run, rather than the average runtime per iteration. As shown in Table 3, the $\boldsymbol{\mu}$ update accounts for the largest runtime among the iterative components, as each Newton step requires solving a large sparse linear system involving the NNGP precision matrix. Although the $\boldsymbol{\Sigma}$ update involves matrix inversions, we limit the number of inner fixed-point iterations, keeping its total runtime relatively low.

Table 3: Mean runtime and standard error (SE) of the major computational components of VoGCAM under Case II, computed over 10 independent runs. For iterative components, runtime refers to the total time spent on each component across all iterations within a run, rather than the average time per iteration.

| Stage | Mean (sec) | SE (sec) | Type |
|---|---|---|---|
| Voronoi tessellation | 0.4105 | 0.0384 | Preprocessing |
| NNGP construction | 0.1975 | 0.0062 | Preprocessing |
| $\boldsymbol{\beta}$ update | 0.0772 | 0.0013 | Iterative update (total) |
| $\boldsymbol{\mu}$ update | 0.5980 | 0.0127 | Iterative update (total) |
| $\boldsymbol{\Sigma}$ update | 0.1214 | 0.0070 | Iterative update (total) |

**Real Data.** We further evaluate the proposed method on two real-world datasets to assess its performance in practical scenarios. The first dataset consists of neuronal spike locations recorded from the medial entorhinal cortex of freely foraging rats (Sargolini et al., 2006), which has been used as a benchmark in spatial point process modeling (Aglietti et al., 2019). Following Aglietti et al. (2019), we use 583 training observations and 29,710 testing observations, without including fixed effects in this analysis. The second dataset comprises spatially resolved transcriptomics data

from the breast cancer S1R1 sample released by 10x Genomics through its Xenium platform. We focus on a region containing approximately 5,000 cells and model the spatial distribution of the epithelial marker KRT15, using the local abundances of DCIS_1 cells and Myoepi_KRT15$^+$ cells as covariates. The spatial coordinates are rescaled by dividing by 1,000, and the covariates are standardized. We randomly split the spatial locations into training and testing sets using a 70/30 split. For the mesh-based representation, covariate values at mesh vertices are assigned from the nearest training location.

Table 4: Comparison of the training log-likelihood ($l_{train}$), normalized predictive log-likelihood ($l^N_{pred}$), and running time (Time in seconds) for different methods on two real-world datasets. Bold numbers indicate the best value for each metric. VBPP is included only for the neuronal dataset because the current implementation does not support fixed effects.

| Method | Neuronal data | | | Transcriptomics data | | |
|---|---|---|---|---|---|---|
| | $l_{train}$ | $l^N_{pred}$ | Time | $l_{train}$ | $l^N_{pred}$ | Time |
| VoGCAM | $\mathbf{-1697.19}$ | $\mathbf{-9.04}$ | 0.38 | $\mathbf{17517.94}$ | $-0.95$ | 2.45 |
| INLA | $-2119.08$ | $-9.08$ | 2.10 | 17117.56 | $-1.63$ | 4.60 |
| VIFRK | $-2094.39$ | $-9.10$ | $\mathbf{0.05}$ | 17130.57 | $-1.87$ | $\mathbf{0.10}$ |
| VBPP | $-2066.35$ | $-9.05$ | 1.36 | – | – | – |
| MINCON | $-2233.88$ | $-9.20$ | 0.22 | 15993.54 | $-0.02$ | 1.64 |
| QUASI | $-2234.75$ | $-9.20$ | 1.78 | 15985.15 | $\mathbf{-0.01}$ | 6.59 |

We evaluated our proposed VoGCAM and competing methods on the datasets. Table 4 presents the training log-likelihood, normalized predictive log-likelihood, and computational time. The results obtained from the neuronal dataset exhibit a pattern similar to our simulation study. Specifically, VoGCAM achieves higher training and test log-likelihoods, whereas VIFRK is the fastest among the compared methods. For the transcriptomics dataset, our proposed method consistently delivers the best training log-likelihood result, while the normalized predictive log-likelihoods attained by MINCON and QUASI surpass those of the other methods, suggesting that the fixed effect may adequately capture the stochastic variation. A detailed analysis of this phenomenon is provided in Appendix E. We also compare Voronoi tessellation with uniform grid cell sampling on the two real-world datasets; the results are reported in Appendix E.

## 5 Discussion

In this paper, we propose a variational inference approach to fitting the LGCP models. By using the variational Gaussian approximation, we transform the intractable likelihood-based inference problem into an optimization problem. A novel and efficient coordinate ascent maximization algorithm is devised with theoretical guarantee. Numerical experiments also validate the effectiveness of our algorithm.

Our method can be readily extended to more general Cox processes to model a wider range of point pattern types. In the standard LGCP framework, all spatial heterogeneity is governed by a single intensity function, yielding a single-structure point process. However, many real-world phenomena require multi-structure point processes that combine several spatio-temporal random components. For instance, when examining the spatial distributions of multiple tree species in a forest, each species may follow its own point process. To accommodate such complexities, the intensity function must be generalized to capture common multi-structure mechanisms, including hybridization, sparsification, and superposition. Performing variational inference in this multi-structure setting poses significant challenges, as the resulting objective functions generally lack closed-form expressions. Addressing these challenges will therefore require the development of novel modeling frameworks and inference techniques. Another interesting direction for future work is to incorporate the estimation of currently fixed kernel hyperparameters, such as the range and smoothness parameters, into the variational inference procedure.

## Acknowledgments and Disclosure of Funding

This work was supported by the LinGang Laboratory under Grant No. LGL-1987.

## References

Aglietti, V., Bonilla, E. V., Damoulas, T., and Cripps, S. Structured variational inference in continuous cox process models. *Advances in Neural Information Processing Systems*, 32, 2019.

Arridge, S. R., Ito, K., Jin, B., and Zhang, C. Variational Gaussian approximation for Poisson data. *Inverse Problems*, 34(2):025005, 2018.

Aurenhammer, F. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM computing surveys (CSUR)*, 23(3):345–405, 1991.

Bachl, F. E., Lindgren, F., Borchers, D. L., and Illian, J. B. `inlabru`: an R package for Bayesian spatial modelling from ecological survey data. *Methods in Ecology and Evolution*, 10(6):760–766, 2019.

Baddeley, A. and Turner, R. `spatstat`: an R package for analyzing spatial point patterns. *Journal of Statistical Software*, 12(6):1–42, 2005.

Baddeley, A., Rubak, E., and Turner, R. *Spatial point patterns: methodology and applications with R*. CRC press, 2015.

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

Bertsekas, D. P. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Challis, E. and Barber, D. Gaussian Kullback-Leibler approximate inference. *The Journal of Machine Learning Research*, 14(1):2239–2286, 2013.

Cressie, N. and Johannesson, G. Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):209–226, 2008.

Daley, D. J. and Vere-Jones, D. *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Springer Science & Business Media, 2007.

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

Dovers, E., Brooks, W., Popovic, G. C., and Warton, D. I. Fast, approximate maximum likelihood estimation of log-Gaussian Cox processes. *Journal of Computational and Graphical Statistics*, 32 (4):1660–1670, 2023.

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., and Banerjee, S. Efficient algorithms for bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics*, 28(2):401–414, 2019.

Flagg, K. and Hoegh, A. The integrated nested Laplace approximation applied to spatial log-Gaussian Cox process models. *Journal of Applied Statistics*, 50(5):1128–1151, 2023.

Fuglstad, G.-A., Simpson, D., Lindgren, F., and Rue, H. Constructing priors that penalize the complexity of Gaussian random fields. *Journal of the American Statistical Association*, 114(525): 445–452, 2019.

Guan, Y., Jalilian, A., and Waagepetersen, R. Quasi-likelihood for spatial point processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 77(3):677–697, 2015.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pp. 282. Citeseer, 2013.

Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley & Sons, 2008.

Kelley, C. T. *Iterative methods for linear and nonlinear equations*. SIAM, 1995.

Lindgren, F. and Rue, H. Bayesian spatial modelling with R-INLA. *Journal of Statistical Software*, 63(19):1–25, 2015.

Lindgren, F., Rue, H., and Lindström, J. 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(4):423–498, 2011.

Lloyd, C., Gunter, T., Osborne, M., and Roberts, S. Variational inference for Gaussian process modulated Poisson processes. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 1814–1822, 2015.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998.

Murray, I., Ghahramani, Z., and MacKay, D. J. C. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence*, pp. 359–366, 2006.

Quinonero-Candela, J. and Rasmussen, C. E. A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

Rue, H., Martino, S., and Chopin, N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

Rue, H., Riebler, A., Sørbye, S. H., Illian, J. B., Simpson, D. P., and Lindgren, F. K. Bayesian computing with INLA: a review. *Annual Review of Statistics and Its Application*, 4:395–421, 2017.

Sargolini, F., Fyhn, M., Hafting, T., McNaughton, B. L., Witter, M. P., Moser, M.-B., and Moser, E. I. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312 (5774):758–762, 2006.

Shirota, S. and Banerjee, S. Scalable inference for space-time Gaussian Cox processes. *Journal of Time Series Analysis*, 40(3):269–287, 2019.

Shirota, S. and Gelfand, A. E. Space and circular time log Gaussian Cox processes with application to crime event data. *The Annals of Applied Statistics*, pp. 481–503, 2017.

Simpson, D., Illian, J. B., Lindgren, F., Sørbye, S. H., and Rue, H. Going off grid: Computationally efficient inference for log-Gaussian Cox processes. *Biometrika*, 103(1):49–70, 2016.

Stein, M. L. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Science & Business Media, 1999.

Stein, M. L. A modeling approach for large spatial datasets. *Journal of the Korean Statistical Society*, 37(10):3–10, 2008.

Taylor, B. M. and Diggle, P. J. INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes. *Journal of Statistical Computation and Simulation*, 84 (10):2266–2284, 2014.

Taylor, B. M., Davies, T. M., Rowlingson, B. S., and Diggle, P. J. `lgcp`: an R package for inference with spatial and spatio-temporal log-Gaussian Cox processes. *Journal of Statistical Software*, 52: 1–40, 2013.

Teng, M., Nathoo, F., and Johnson, T. D. Bayesian computation for log-Gaussian Cox processes: A comparative analysis of methods. *Journal of Statistical Computation and Simulation*, 87(11): 2227–2252, 2017.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pp. 567–574. PMLR, 2009.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. A framework for interdomain and multioutput Gaussian processes. *Stat*, 1050:2, 2020.

Wainwright, M. J., Jordan, M. I., et al. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1–305, 2008.

Wu, L., Pleiss, G., and Cunningham, J. P. Variational nearest neighbor Gaussian process. In *International Conference on Machine Learning*, pp. 24114–24130. PMLR, 2022.

Zammit-Mangion, A. and Cressie, N. FRK: an R package for spatial and spatio-temporal prediction with large datasets. *Journal of Statistical Software*, 98(4):1–48, 2021.

Zhang, L., Datta, A., and Banerjee, S. Practical Bayesian modeling and inference for massive spatial data sets on modest computing environments. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 12(3):197–209, 2019.

# A Derivation of the explicit expression of ELBO

First, we derive the explicit expression of $E\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ in 6. Based on the definition of ELBO in 5, we have:

$$\begin{aligned} E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \int q(\mathbf{Z}) \log \frac{p\left(\mathbf{Y} \mid \mathbf{Z}\right) p(\mathbf{Z})}{q(\mathbf{Z})} \, \mathrm{d}\mathbf{Z} \\ &= \int q(\mathbf{Z}) \log p\left(\mathbf{Y} \mid \mathbf{Z}\right) \, \mathrm{d}\mathbf{Z} + \int q(\mathbf{Z}) \log p\left(\mathbf{Z}\right) \, \mathrm{d}\mathbf{Z} - \int q(\mathbf{Z}) \log q\left(\mathbf{Z}\right) \, \mathrm{d}\mathbf{Z} \\ &= \mathrm{E}_{q(\mathbf{Z})}\left[\log p\left(\mathbf{Y} \mid \mathbf{Z}\right)\right] + \mathrm{E}_{q(\mathbf{Z})}\left[\log p\left(\mathbf{Z}\right)\right] - \mathrm{E}_{q(\mathbf{Z})}\left[\log q\left(\mathbf{Z}\right)\right]. \end{aligned} \tag{21}$$

Next, we will calculate the explicit expressions of the three parts in (21) separately. For the derivation of the first part, we use the conclusion of the moment–generating function in the calculation. The specific details are as follows:

$$\begin{aligned} \mathrm{E}_{q(\mathbf{Z})}\left[\log p\left(\mathbf{y} \mid \mathbf{Z}\right)\right] =& \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-\mathbf{w}^{\top} \exp\left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\mathbf{Z}\right) + \mathbf{1}_n^{\top}\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\mathbf{Z}\right)\right] \\ =& -\mathbf{w}^{\top} \exp\left(\widetilde{\mathbf{X}}\boldsymbol{\beta}\right) \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\sum_{i=1}^{n} \exp\left(\widetilde{\mathbf{A}}_i\mathbf{Z}\right)\right] + \mathbf{1}_N^{\top}\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}\right) \\ =& -\mathbf{w}^{\top} \exp\left(\widetilde{\mathbf{X}}\boldsymbol{\beta}\right) \cdot \sum_{i=1}^{n} \exp\left(\widetilde{\mathbf{A}}_i^{\top}\boldsymbol{\mu} + \frac{1}{2}\widetilde{\mathbf{A}}_i\boldsymbol{\Sigma}\widetilde{\mathbf{A}}_i^{\top}\right) + \mathbf{1}_N^{\top}\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}\right) \\ =& -\mathbf{w}^{\top} \exp\left\{\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\mathrm{diag}\left(\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^{\top}\right)\right\} + \mathbf{1}_N^{\top}\left(\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}\right), \end{aligned} \tag{22}$$

where the vector $\widetilde{\mathbf{A}}_i$ represents the $i$-th row of the matrix $\widetilde{\mathbf{A}}$.

For the derivation of the second part, we perform the Cholesky decomposition on the matrix $\boldsymbol{\Sigma}$ to get $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^{\top}$. Thus, $\mathbf{Z}$ can be expressed as $\mathbf{Z} = \boldsymbol{\mu} + \mathbf{L}\mathbf{x}$, where $\mathbf{x} \sim N\left(\mathbf{0}, \mathbf{I}_{n+N}\right)$. Therefore, we can

obtain:

$$\begin{aligned}
\mathrm{E}_{q(\mathbf{Z})}\left[\log p\left(\mathbf{Z}\right)\right] &= \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{Z} \mid \mathbf{0}, \mathbf{K}_{\boldsymbol{\theta}}\right)\right] \\
&= \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left(-\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{1}{2}\mathbf{Z}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{Z}\right) \\
&= -\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{1}{2}\mathrm{E}_{N(\mathbf{x}|\mathbf{0},\mathbf{I}_{n+N})}\left[\left(\boldsymbol{\mu}+\mathbf{L}\mathbf{x}\right)^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\left(\boldsymbol{\mu}+\mathbf{L}\mathbf{x}\right)\right] \\
&= -\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{1}{2}\boldsymbol{\mu}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu} - \frac{1}{2}\mathrm{E}_{N(\mathbf{x}|\mathbf{0},\mathbf{I}_{n+N})}\left(\mathbf{x}^{\top}\mathbf{L}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{L}\mathbf{x}\right) \\
&= -\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{1}{2}\boldsymbol{\mu}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu} - \frac{1}{2}\mathrm{tr}\left(\mathbf{L}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\mathbf{L}\right) \\
&= -\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{1}{2}\boldsymbol{\mu}^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right),
\end{aligned} \tag{23}$$

$$\begin{aligned}
\mathrm{E}_{q(\mathbf{Z})}\left[\log q\left(\mathbf{Z}\right)\right] &= \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{Z} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)\right] \\
&= \mathrm{E}_{N(\mathbf{Z}|\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[-\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\left(\mathbf{Z}-\boldsymbol{\mu}\right)^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\left(\mathbf{Z}-\boldsymbol{\mu}\right)\right] \\
&= -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\mathrm{E}_{N(\mathbf{x}|\mathbf{0},\mathbf{I}_{n+N})}\left[\left(\boldsymbol{\mu}+\mathbf{L}\mathbf{x}-\boldsymbol{\mu}\right)^{\top}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}+\mathbf{L}\mathbf{x}-\boldsymbol{\mu}\right)\right] \\
&= -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\mathrm{E}_{N(\mathbf{x}|\mathbf{0},\mathbf{I}_{n+N})}\left(\mathbf{x}^{\top}\mathbf{L}^{\top}\boldsymbol{\Sigma}^{-1}\mathbf{L}\mathbf{x}\right) = -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\mathrm{tr}\left(\mathbf{L}^{\top}\boldsymbol{\Sigma}^{-1}\mathbf{L}\right) \\
&= -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{1}{2}\mathrm{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\right) = -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| - \frac{n+N}{2}.
\end{aligned} \tag{24}$$

Substituting (22), (23) and (24) into (21), we can obtain the explicit expression of ELBO as shown in the equation (6).

Next, in the following Lemma, we prove that the third line in the explicit expression of $E\left(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ in (6) is equivalent to the KL divergence between two Gaussian distributions with the same mean but different covariance matrices.

**Lemma A.1.** *The expression for the KL divergence between two Gaussian distributions with the same mean $f(\mathbf{x}) = N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}\right)$ and $f_0(\mathbf{x}) = N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \mathbf{K}_{\boldsymbol{\theta}}\right)$ is*

$$D\left(\boldsymbol{\Sigma}, \mathbf{K}_{\boldsymbol{\theta}}\right) = \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right) - \frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| + \frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| - \frac{n+N}{2}. \tag{25}$$

*Proof.* According to the definition of KL divergence, we can obtain

$$\begin{aligned}
D\left(\boldsymbol{\Sigma}, \mathbf{K}_{\boldsymbol{\theta}}\right) &= \int f\left(\mathbf{x}\right)\log\frac{f\left(\mathbf{x}\right)}{f_0\left(\mathbf{x}\right)}\,\mathrm{d}\mathbf{x} \\
&= \int f\left(\mathbf{x}\right)\log f\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x} - \int f\left(\mathbf{x}\right)\log f_0\left(\mathbf{x}\right)\,\mathrm{d}\mathbf{x}. \\
&= \mathrm{E}_{f(\mathbf{x})}\left[\log f\left(\mathbf{x}\right)\right] - \mathrm{E}_{f(\mathbf{x})}\left[\log f_0\left(\mathbf{x}\right)\right] \\
&= \mathrm{E}_{N(\mathbf{x}|\boldsymbol{\mu}_m,\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}\right)\right] - \mathrm{E}_{N(\mathbf{x}|\boldsymbol{\mu}_m,\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \mathbf{K}_{\boldsymbol{\theta}}\right)\right]
\end{aligned} \tag{26}$$

Based on the derivation of (24), we can get

$$\begin{aligned}
\mathrm{E}_{N(\mathbf{x}|\boldsymbol{\mu}_m,\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \boldsymbol{\Sigma}\right)\right] &= -\frac{1}{2}\log\left|\boldsymbol{\Sigma}\right| \\
&\quad - \frac{1}{2}\left(\boldsymbol{\mu}_m-\boldsymbol{\mu}_m\right)^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\left(\boldsymbol{\mu}_m-\boldsymbol{\mu}_m\right) - \frac{n}{2},
\end{aligned} \tag{27}$$

$$\begin{aligned}
\mathrm{E}_{N(\mathbf{x}|\boldsymbol{\mu}_m,\boldsymbol{\Sigma})}\left[\log N\left(\mathbf{x} \mid \boldsymbol{\mu}_m, \mathbf{K}_{\boldsymbol{\theta}}\right)\right] &= -\frac{1}{2}\log\left|\mathbf{K}_{\boldsymbol{\theta}}\right| \\
&\quad - \frac{1}{2}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_m\right)^{\top}\mathbf{K}_{\boldsymbol{\theta}}^{-1}\left(\boldsymbol{\mu}-\boldsymbol{\mu}_m\right) - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right).
\end{aligned} \tag{28}$$

Substituting the expressions (27) and (28) into (26) completes the proof. □

## B The existence and uniqueness of the optimal solution of ELBO

Here, we prove Theorem 2.2. To ensure the existence of the maximum value, we need to prove that $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is coercive on the convex set $\mathrm{R}^m \times \mathrm{R}^n \times \mathcal{S}_{n+N}^+$, that is, to prove that when any one of the parameters in $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ approaches the boundary of its domain or infinity, the value of the function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \to -\infty$. Next, we discuss the three cases corresponding to the three parameters respectively.

For $\boldsymbol{\mu}$, the expression of the term related to $\boldsymbol{\mu}$ in the function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$E_1(\boldsymbol{\mu}) = -\mathbf{w}^\top \exp\left(\widetilde{\mathbf{A}}\boldsymbol{\mu}\right) + \mathbf{1}_N^\top \mathbf{A}\boldsymbol{\mu} - \frac{1}{2}\boldsymbol{\mu}^\top \mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\mu}.$$

When $\|\boldsymbol{\mu}\|_2 \to \infty$, for the quadratic term, since the covariance matrix $\mathbf{K}_{\boldsymbol{\theta}}$ is positive definite, the quadratic term approaches $-\infty$. Because the decay rate of the quadratic term is faster than that of the linear term, the sum of the quadratic term and the linear term must approach $-\infty$. For the exponential term $-\mathbf{w}^\top \exp\left(\widetilde{\mathbf{A}}\boldsymbol{\mu}\right)$, if one component in $\widetilde{\mathbf{A}}\boldsymbol{\mu}$ approaches $-\infty$, the exponential term approaches 0; if one component in $\widetilde{\mathbf{A}}\boldsymbol{\mu}$ approaches $+\infty$, the exponential term approaches $-\infty$. Therefore, in summary, when $\|\boldsymbol{\mu}\|_2 \to \infty$, the value of the function $E_1(\boldsymbol{\mu})$ approaches $-\infty$.

For $\boldsymbol{\Sigma}$, the expression of the term related to $\boldsymbol{\Sigma}$ in the function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$E_2(\boldsymbol{\Sigma}) = -\mathbf{w}^\top \exp\left\{\frac{1}{2}\mathrm{diag}\left(\widetilde{\mathbf{A}}\boldsymbol{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right) + \frac{1}{2}\log|\boldsymbol{\Sigma}|.$$

When $\boldsymbol{\Sigma}$ approaches the boundary of $\mathcal{S}_{n+N}^+$, that is, $\boldsymbol{\Sigma}$ becomes singular, which is equivalent to at least one eigenvalue of $\boldsymbol{\Sigma}$ satisfying $\lambda_i(\boldsymbol{\Sigma}) \to 0^+$. In this case, since $\log|\boldsymbol{\Sigma}| = \sum \log \lambda_i(\boldsymbol{\Sigma}) \to -\infty$, and both the exponential term and the linear term are negative, the value of the function $E_2(\boldsymbol{\Sigma})$ approaches $-\infty$; When $\boldsymbol{\Sigma}$ approaches "infinity", for example, when the maximum eigenvalue satisfies $\lambda_{\max}(\boldsymbol{\Sigma}) \to \infty$, the linear term $-\frac{1}{2}\mathrm{tr}\left(\mathbf{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{\Sigma}\right)$ approaches $-\infty$. Because the decay rate of the linear term is faster than the growth rate of the logarithmic term, and the exponential term is always negative, the value of the function $E_2(\boldsymbol{\Sigma})$ approaches $-\infty$.

For $\boldsymbol{\beta}$, the expression of the term related to $\boldsymbol{\beta}$ in the function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$E_3(\boldsymbol{\beta}) = -\mathbf{w}^\top \exp\left(\widetilde{\mathbf{X}}\boldsymbol{\beta}\right) + \mathbf{1}_N^\top (\mathbf{X}\boldsymbol{\beta}).$$

When $\|\boldsymbol{\beta}\|_2 \to \infty$, for the exponential term, since $\widetilde{\mathbf{X}}$ is the value of the covariate at $n$ integration points, in practice, a positive value of one component of $\widetilde{\mathbf{X}}\boldsymbol{\beta}$ can definitely be found. At this time, the exponential term approaches $-\infty$. Because the decay rate of the exponential term is faster than that of the linear term, when $\|\boldsymbol{\beta}\|_2 \to \infty$, the value of the function $E_3(\boldsymbol{\beta})$ approaches $-\infty$.

In conclusion, we have proved that the function $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is a coercive function on the convex set, which is equivalent to proving that $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ has a maximum value on the convex set. According to the fact that $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is strongly concave obtained in Theorem 2.1, the maximum point is unique at this time.

## C Derivation Details and Theoretical Property of the Coordinate Ascent Maximization Algorithm

### C.1 Derivation of $\boldsymbol{\Gamma}$ based on NNGP

For the convenience, we abbreviate the $\mathbf{Z} = [Z(\widetilde{\mathbf{s}}_1), \ldots, Z(\widetilde{\mathbf{s}}_n), Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_N)]^\top$ as $\mathbf{Z} = [Z_1, \ldots Z_{n+N}]^\top$. Next, we can express the joint distribution of $\mathbf{Z}$ as

$$p(\mathbf{Z}) = p(Z_1) \prod_{i=2}^{n+N} p(Z_i \mid Z_{1:i-1}).$$

In NNGP, we let the realization of $Z(\mathbf{s})$ at any $\mathbf{s}_i \in \mathcal{I} \cup \mathcal{D}$ be conditional on, at most, the realizations at a pre-specified number of nearest neighbor locations to $\mathbf{s}_i$ in $\mathcal{I} \cup \mathcal{D}$, and this set is denoted as

$\text{Pa}[\mathbf{s}_i]$. Then, the approximate joint distribution can be expressed as

$$\widetilde{p}(\mathbf{Z}) = p(Z_1) \prod_{i=2}^{n+N} p\left(Z_i \mid Z_{\text{Pa}[\mathbf{s}_i]}\right).$$

Datta et al. (2016) proved that $\widetilde{p}(\mathbf{Z})$ is the joint density of a multivariate Gaussian distribution with a sparse precision matrix, which satisfies

$$\sigma^{-2}\mathbf{\Gamma} = (\mathbf{I} - \mathbf{A})^\top \mathbf{D}^{-1} (\mathbf{I} - \mathbf{A}), \quad \widetilde{p}(\mathbf{Z}) = N\left(\mathbf{Z} \mid \mathbf{0}, \sigma^2 \mathbf{\Gamma}\right).$$

where $\mathbf{I}$ is the $(n+N)$-dimensional identity matrix. The matrix $\mathbf{A}$ describes the neighbor information of all $n+N$ locations. The $i$-th row ($i > 1$) of $\mathbf{A}$ has non-zero entries at the positions indexed by $\text{Pa}[\mathbf{s}_i]$, and these non-zero entries are calculated as follows

$$\mathbf{A}\left(i, \text{Pa}[\mathbf{s}_i]\right) = \mathbf{K}_{\boldsymbol{\theta}}\left(\mathbf{s}_i, \text{Pa}[\mathbf{s}_i]\right) \left(\mathbf{K}_{\boldsymbol{\theta}}\left(\text{Pa}[\mathbf{s}_i], \text{Pa}[\mathbf{s}_i]\right)\right)^{-1}.$$

The matrix $\mathbf{D}$ is a diagonal matrix, and its $i$-th diagonal element describes the variance information at $\mathbf{s}_i$, which satisfies

$$\mathbf{D}(i,i) = \mathbf{K}_{\boldsymbol{\theta}}(\mathbf{s}_i, \mathbf{s}_i) - \mathbf{K}_{\boldsymbol{\theta}}\left(\mathbf{s}_i, \text{Pa}[\mathbf{s}_i]\right) \left(\mathbf{K}_{\boldsymbol{\theta}}\left(\text{Pa}[\mathbf{s}_i], \text{Pa}[\mathbf{s}_i]\right)\right)^{-1} \mathbf{K}_{\boldsymbol{\theta}}\left(\text{Pa}[\mathbf{s}_i], \mathbf{s}_i\right).$$

Finley et al. (2019) discussed methods for efficiently constructing $\mathbf{A}$ and $\mathbf{D}$. On this basis, Zhang et al. (2019) implemented the above construction in practice based on `rStan` in `R`.

### C.2 The Derivation Details of the Expression of $\nabla_{\mathbf{\Sigma}} E$

**Lemma C.1.** *The gradient of the objective function $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \mathbf{\Sigma})$ with respect to $\mathbf{\Sigma}$ is*

$$\nabla_{\mathbf{\Sigma}} E = \frac{1}{2}\left[-\widetilde{\mathbf{A}}^\top \text{Diag}\left[\mathbf{w} \circ \exp\left\{\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\text{diag}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\}\right]\widetilde{\mathbf{A}} - \sigma^{-2}\mathbf{\Gamma} + \mathbf{\Sigma}^{-1}\right].$$

*Proof.* Define the matrix function $g(\mathbf{\Sigma}) = \widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\text{diag}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)$, where the $i$-th component of $g(\mathbf{\Sigma})$ satisfies

$$g_i(\mathbf{\Sigma}) = \left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu}\right)_i + \frac{1}{2}\left[\text{diag}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right]_i = \left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu}\right)_i + \frac{1}{2}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii}$$

Given that the expansion of the elements of the quadratic form matrix can be expressed as $\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii} = \sum_{j,k}\widetilde{\mathbf{A}}_{ij}\mathbf{\Sigma}_{jk}\widetilde{\mathbf{A}}_{ik}$, thus the partial derivative of $g_i(\mathbf{\Sigma})$ with respect to the matrix elements can be obtained as

$$\frac{\partial\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii}}{\partial\mathbf{\Sigma}_{jk}} = \frac{\partial\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii}}{\partial\mathbf{\Sigma}_{jk}} = \frac{1}{2}\widetilde{\mathbf{A}}_{ij}\widetilde{\mathbf{A}}_{ik}$$

Writing it in matrix form is equivalent to

$$\frac{\partial g_i(\mathbf{\Sigma})}{\partial\mathbf{\Sigma}} = \frac{1}{2}\widetilde{\mathbf{A}}_i^\top\widetilde{\mathbf{A}}_i$$

where $\widetilde{\mathbf{A}}_i$ represents the $i$-th row vector of the matrix $\widetilde{\mathbf{A}}$.

Based on the chain rule, the gradient of the first term with respect to the matrix $\mathbf{\Sigma}$ can be derived as follows

$$\begin{aligned}
\frac{\partial}{\partial\mathbf{\Sigma}}\left(-\mathbf{w}^\top\exp\{g(\mathbf{\Sigma})\}\right) &= -\sum_{i=1}^{N} w_i \frac{\partial}{\partial\mathbf{\Sigma}}\exp\left\{g_i(\mathbf{\Sigma})\right\} = -\sum_{i=1}^{N} w_i \exp\left\{g_i(\mathbf{\Sigma})\right\}\frac{\partial g_i(\mathbf{\Sigma})}{\partial\mathbf{\Sigma}} \\
&= -\sum_{i=1}^{N} w_i \exp\left\{\left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu}\right)_i + \frac{1}{2}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii}\right\}\cdot\frac{1}{2}\widetilde{\mathbf{A}}_i^\top\widetilde{\mathbf{A}}_i \\
&= -\frac{1}{2}\sum_{i=1}^{N} w_i \exp\left\{\left(\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu}\right)_i + \frac{1}{2}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)_{ii}\right\}\widetilde{\mathbf{A}}_i^\top\widetilde{\mathbf{A}}_i \\
&\triangleq -\frac{1}{2}\widetilde{\mathbf{A}}^\top\text{Diag}\left[\mathbf{w} \circ \exp\left\{\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\text{diag}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\}\right]\widetilde{\mathbf{A}}.
\end{aligned}$$

For the gradients of the second and third terms with respect to the matrix $\Sigma$, according to the following properties

$$\frac{\partial}{\partial \mathbf{\Sigma}} \mathrm{tr}(\mathbf{\Gamma}\mathbf{\Sigma}) = \mathbf{\Gamma}^\top, \quad \frac{\partial}{\partial \mathbf{\Sigma}} \log|\mathbf{\Sigma}| = (\mathbf{\Sigma}^{-1})^\top,$$

and since both $\mathbf{\Gamma}$ and $\mathbf{\Sigma}$ are symmetric, thus $\mathbf{\Gamma}^\top = \mathbf{\Gamma}$, $(\mathbf{\Sigma}^{-1})^\top = \mathbf{\Sigma}^{-1}$.

By organizing the gradient results of the three parts with respect to the matrix $\Sigma$, the result of Lemma C.1 can be obtained. □

### C.3 The Convergence Analysis of Fixed-point Method for Updating $\mathbf{\Sigma}$

In this paragraph, we first prove in Theorem C.2 that the sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$ obtained based on the fixed-point method (14) is bounded and therefore admits a convergent subsequence.

**Theorem C.2.** *The sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$ obtained by iteration based on the fixed-point method has a convergent subsequence.*

*Proof.* According to the iterative relationship of the fixed-point method (14), for any $k = 1, 2, \ldots$, we have $\mathbf{\Sigma}_k \preceq \sigma^2\mathbf{\Gamma}^{-1}$, which is equivalent to $\|\mathbf{\Sigma}_k\|_2 \leq \|\sigma^2\mathbf{\Gamma}^{-1}\|_2$. Therefore, the sequence of matrix variables $\{\mathbf{\Sigma}_k\}$ is bounded in the spectral norm. Since every bounded sequence has a convergent subsequence, there exists a subsequence $\left\{\mathbf{\Sigma}_{(n)}\right\}_{n\geq 0}$ such that it converges to the limit $\mathbf{\Sigma}_\star$ in the sense of the 2-norm. □

Furthermore, we give the specific form of the above convergent subsequence in Lemma C.3 and Theorem C.4.

For any positive definite matrices $\mathbf{C}, \mathbf{D} \in \mathcal{S}_{n+N}^+$, assume that $\mathbf{C} \preceq \mathbf{D}$, if the mapping $T$ satisfies $T(\mathbf{C}) \succeq T(\mathbf{D})$, then the mapping $T$ is said to be anti-monotonic.

**Lemma C.3.** *The mapping $F$ corresponding to the fixed-point method* (14) *is given by*

$$F(\mathbf{\Sigma}) = \left[\widetilde{\mathbf{A}}^\top \mathrm{Diag}\left[\mathbf{w} \circ \exp\left\{\widetilde{\mathbf{X}}\boldsymbol{\beta} + \widetilde{\mathbf{A}}\boldsymbol{\mu} + \frac{1}{2}\mathrm{diag}\left(\widetilde{\mathbf{A}}\mathbf{\Sigma}\widetilde{\mathbf{A}}^\top\right)\right\}\right]\widetilde{\mathbf{A}} + \sigma^{-2}\mathbf{\Gamma}\right]^{-1}. \tag{29}$$

*Then, the mapping $F$ is anti-monotonic.*

*Proof.* The following proof refers to (Arridge et al., 2018, Appendix A.). Given two positive definite matrices $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$, assume that $\mathbf{\Sigma}_1 \preceq \mathbf{\Sigma}_2$. Then, each component value of the vector $\mathrm{diag}(\widetilde{\mathbf{A}}\mathbf{\Sigma}_1\widetilde{\mathbf{A}}^\top)$ is less than the value of the corresponding position of the vector $\mathrm{diag}(\widetilde{\mathbf{A}}\mathbf{\Sigma}_2\widetilde{\mathbf{A}}^\top)$, denoted as $\mathrm{diag}(\widetilde{\mathbf{A}}\mathbf{\Sigma}_1\widetilde{\mathbf{A}}^\top) \leq \mathrm{diag}(\widetilde{\mathbf{A}}\mathbf{\Sigma}_2\widetilde{\mathbf{A}}^\top)$. At this time, we can obtain

$$T(\mathbf{\Sigma}_1) - T(\mathbf{\Sigma}_2) = T(\mathbf{\Sigma}_1)\left\{(T(\mathbf{\Sigma}_2))^{-1} - (T(\mathbf{\Sigma}_1))^{-1}\right\}T(\mathbf{\Sigma}_2) \succeq \mathbf{0}.$$

Therefore, the lemma is proved. □

Next, we construct two convergent subsequences of the maximization sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$ obtained by the fixed-point method (14).

**Theorem C.4.** *Given the initial iteration value $\mathbf{\Sigma}_0 = \sigma^2\mathbf{\Gamma}^{-1}$, the maximization sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$ has two convergent subsequences $\{\mathbf{\Sigma}_{2k}\}_{k\geq 0}$ and $\{\mathbf{\Sigma}_{2k+1}\}_{k\geq 0}$.*

*Proof.* The following proof refers to (Arridge et al., 2018, Appendix A.). According to the expression of the fixed - point method (14), for any $k \geq 0$, we have

$$\mathbf{0} \preceq \mathbf{\Sigma}_k \preceq \mathbf{\Sigma}_0,$$

so the maximization sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$ is always bounded. Based on the iterative relation of the fixed - point method (14) and the conclusion of the anti - monotonicity of $F$ obtained from Lemma C.3, if $\mathbf{\Sigma}_i \succeq \mathbf{\Sigma}_j$, then $\mathbf{\Sigma}_{i+1} \preceq \mathbf{\Sigma}_{j+1}$; similarly, if $\mathbf{\Sigma}_i \preceq \mathbf{\Sigma}_j$, then $\mathbf{\Sigma}_{i+1} \succeq \mathbf{\Sigma}_{j+1}$. Therefore, in the sense of the 2 - norm, the sequence $\{\mathbf{\Sigma}_{2k}\}_{k\geq 0}$ is a decreasing sequence, and the sequence $\{\mathbf{\Sigma}_{2k+1}\}_{k\geq 0}$ is an increasing sequence. Since a monotonic and bounded sequence must converge, $\{\mathbf{\Sigma}_{2k}\}_{k\geq 0}$ and $\{\mathbf{\Sigma}_{2k+1}\}_{k\geq 0}$ are two convergent subsequences of the maximization sequence $\{\mathbf{\Sigma}_k\}_{k\geq 0}$, and both are the limits of the fixed-point mapping $F^2$. □

The above result establishes convergence of the even- and odd-indexed subsequences, but does not guarantee that they converge to the same limit. If the two limits coincide, the entire sequence converges; otherwise, the iteration may approach a period-two orbit. Although convergence of the entire sequence to a single fixed point is not established theoretically, the numerical results suggest stable behavior in practice. For the initializations considered in our numerical experiments, the iterates converge to a single fixed point without observable period-two oscillations.

### C.4 The Convergence Analysis of Algorithm 1

For the maximizing sequence $\{\boldsymbol{\beta}_k\}_{k\geq 0}$ and $\{\boldsymbol{\mu}_k\}_{k\geq 0}$ obtained by the Newton method iteration. From the expression of the Hessian matrices (11), it is easy to see that $-\nabla^2_{\boldsymbol{\beta}} E$ and $-\nabla^2_{\boldsymbol{\mu}} E$ are strictly positive definite. Consequently, the Newton method for these updates is guaranteed to converge to the unique optimum of each subproblem (Kelley, 1995). For the maximizing sequence $\{\boldsymbol{\beta}_k\}_{k\geq 0}$ obtained by the fixed - point method, Theorem C.2 guarantees the convergence of the sequence.

Finally, according to the convergence criterion of the coordinate ascent method given in Proposition 2.7.1 of Bertsekas (1997), if for each coordinate direction, there exists a unique point that maximizes the objective function in the current coordinate direction, then the coordinate sequence obtained by the coordinate ascent method is convergent. In the current context, based on the joint strong concavity of $E(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ with respect to $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ obtained from Theorem 2.1, and the convergence of the coordinate sequences obtained by the Newton method and the fixed - point method in their respective directions, the sequences $\{\boldsymbol{\beta}_k\}_{k\geq 0}, \{\boldsymbol{\mu}_k\}_{k\geq 0}, \{\boldsymbol{\Sigma}_k\}_{k\geq 0}$ obtained by Algorithm 1 are convergent.

## D Analysis of the Convergence of the Hyperparameter Monotonic Convergence Algorithm

To analyze the convergence of Algorithm 2, that is, to prove Theorem 3.2, we first need to introduce the following lemmas.

For convenience, when $\sigma$ is fixed, we denote the optimal parameter values obtained based on Algorithm 1 as $\boldsymbol{\beta}_\sigma$, $\boldsymbol{\mu}_\sigma$ and $\boldsymbol{\Sigma}_\sigma$. During the iterative process, since only the hyperparameter $\sigma$ in $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ changes, we will denote $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ as $E_\sigma(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ in the following.

Next, we further decompose the function $E_{\boldsymbol{\theta}}(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ as follows:

$$E_\sigma(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = g(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \sigma^{-2} h(\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \frac{n+N}{2} \log \sigma^{-2},$$

where

$$\begin{aligned} g(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= -\mathbf{w}^\top \boldsymbol{\psi} + \mathbf{1}_N^\top (\mathbf{X}\boldsymbol{\beta} + \mathbf{A}\boldsymbol{\mu}) + \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \log |\boldsymbol{\Gamma}^{-1}|, \\ h(\boldsymbol{\mu}, \boldsymbol{\Sigma}) &= -\frac{1}{2} \boldsymbol{\mu}^\top \boldsymbol{\Gamma} \boldsymbol{\mu} - \frac{1}{2} \mathrm{tr}(\boldsymbol{\Gamma}\boldsymbol{\Sigma}). \end{aligned}$$

Next, we introduce the following lemma.

**Lemma D.1.** *The function $h(\boldsymbol{\mu}_\sigma, \boldsymbol{\Sigma}_\sigma)$ is monotonically decreasing with respect to $\sigma$.*

*Proof.* The proof refers to Lemma 5.3 of Arridge et al. (2018). Given any two positive numbers $\sigma_1$ and $\sigma_2$, according to Theorem 2.1 and Theorem 2.2, we can find the optimal values of the parameters $(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ for the currently fixed $\sigma$. When $\sigma$ is fixed to $\sigma_1$ and $\sigma_2$ respectively, we denote the optimal values of the parameters as $(\boldsymbol{\beta}_{\sigma_1}, \boldsymbol{\mu}_{\sigma_1}, \boldsymbol{\Sigma}_{\sigma_1})$ and $(\boldsymbol{\beta}_{\sigma_2}, \boldsymbol{\mu}_{\sigma_2}, \boldsymbol{\Sigma}_{\sigma_2})$ respectively. Thus, we can obtain

$$\begin{aligned} E_{\sigma_1}(\boldsymbol{\beta}_{\sigma_1}, \boldsymbol{\mu}_{\sigma_1}, \boldsymbol{\Sigma}_{\sigma_1}) &\geq E_{\sigma_1}(\boldsymbol{\beta}_{\sigma_2}, \boldsymbol{\mu}_{\sigma_2}, \boldsymbol{\Sigma}_{\sigma_2}), \\ E_{\sigma_2}(\boldsymbol{\beta}_{\sigma_2}, \boldsymbol{\mu}_{\sigma_2}, \boldsymbol{\Sigma}_{\sigma_2}) &\geq E_{\sigma_2}(\boldsymbol{\beta}_{\sigma_1}, \boldsymbol{\mu}_{\sigma_1}, \boldsymbol{\Sigma}_{\sigma_1}). \end{aligned}$$

Adding the left - hand sides and right - hand sides of the above two inequalities respectively, and then continuing to simplify, we can get

$$\left(\sigma_1^{-2} - \sigma_2^{-2}\right) \left(h(\boldsymbol{\mu}_{\sigma_1}, \boldsymbol{\Sigma}_{\sigma_1}) - h(\boldsymbol{\mu}_{\sigma_2}, \boldsymbol{\Sigma}_{\sigma_2})\right) \geq 0.$$

Therefore, it can be proved that $h\left(\boldsymbol{\mu}_\sigma, \boldsymbol{\Sigma}_\sigma\right)$ is monotonically increasing with respect to $\sigma^{-2}$. It should be noted that since $\sigma > 0$, this conclusion is equivalent to being monotonically decreasing with respect to $\sigma$. □

Based on the conclusion of Lemma D.1, we will prove Theorem 3.2, which refers to Theorem 5.1 of Arridge et al. (2018).

Based on the expression for updating $\sigma$ given in (20), for any $k \geq 1$, we can obtain

$$\sigma_{k+1}^2 - \sigma_k^2 = \frac{-h\left(\boldsymbol{\mu}_{\sigma_k}, \boldsymbol{\Sigma}_{\sigma_k}\right) + h\left(\boldsymbol{\mu}_{\sigma_{k-1}}, \boldsymbol{\Sigma}_{\sigma_{k-1}}\right)}{n + N + 2a + 2}.$$

At this point, the magnitude relationship between $\sigma_{k+1}^2$ and $\sigma_k^2$ depends only on the magnitude relationship between the function values $h\left(\boldsymbol{\mu}_{\sigma_k}, \boldsymbol{\Sigma}_{\sigma_k}\right)$ and $h\left(\boldsymbol{\mu}_{\sigma_{k-1}}, \boldsymbol{\Sigma}_{\sigma_{k-1}}\right)$. Based on Lemma D.1, if $\sigma_k^2 \geq \sigma_{k-1}^2$, then $h(\boldsymbol{\mu}_{\sigma_k}, \boldsymbol{\Sigma}_{\sigma_k}) \leq h(\boldsymbol{\mu}_{\sigma_{k-1}}, \boldsymbol{\Sigma}_{\sigma_{k-1}})$, which means $\sigma_{k+1}^2 \geq \sigma_k^2$. Similarly, if $\sigma_k^2 \leq \sigma_{k-1}^2$, then $\sigma_{k+1}^2 \leq \sigma_k^2$. Therefore, the magnitude relationship between $\sigma_{k+1}^2$ and $\sigma_k^2$ always remains the same as that between $\sigma_k^2$ and $\sigma_{k-1}^2$, indicating that the sequence $\{\sigma_k^2\}$ is monotonic. In addition, since $-h(\boldsymbol{\mu}, \boldsymbol{\Sigma}) > 0$, and according to the boundedness of the parameters given in Theorem 2.2, the sequence $\{\sigma_k^2\}_{k\geq 0}$ is bounded. Since a monotonic and bounded sequence must converge, there exists a $\sigma_\star^2$ such that $\lim_{k\to+\infty} \sigma_k^2 = \sigma_\star^2$. Moreover, since $b > 0$, the update equation implies that

$$\sigma_{k+1}^2 \geq \frac{2b}{n + N + 2a + 2} > 0.$$

Consequently, $\sigma_\star^2 > 0$. Since $\sigma_k > 0$, we conclude that $\lim_{k\to\infty} \sigma_k = \sigma_\star > 0$.

# E Numerical Experiments Details

In this section, we provide supplementary details on the simulation settings and algorithmic parameter settings, along with additional experimental results for Section 4.

**LGCP point pattern simulation settings** Based on the `rLGCP()` function in the `spatstat` package Baddeley & Turner (2005), we simulated a set of point patterns following LGCP within the square observation region $\Omega = [0, 1] \times [0, 1]$, and the expression of its intensity function is

$$\log \lambda(\mathbf{s}) = \beta_0 + X(\mathbf{s})\beta_1 + Z(\mathbf{s}),$$

where $X(\mathbf{s})$ is defined as a one - dimensional covariate with the expression

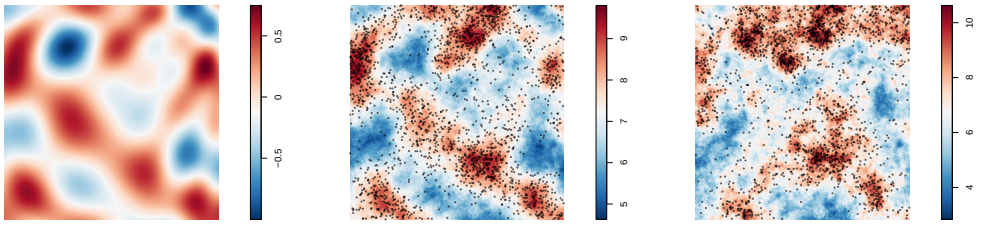
$$\begin{aligned} X(\mathbf{s}) = \frac{1}{2.8}\Bigg[ & 0.9\,\sin\left(2\pi(1.3s_1 + 1.7s_2)\right) + 0.7\,\cos\left(2\pi(2.1s_1 - 1.4s_2)\right) \\ & + 0.5\,\sin\left(2\pi(3.2s_1 s_2)\right) + 0.4\,\cos\left(2\pi(4.0(s_1 - 0.5)^2 + 3.5(s_2 - 0.5)^2)\right) \\ & + 0.3\,\sin\left(2\pi(2.8s_1^2 - 2.3s_2^2)\right)\Bigg]; \end{aligned}$$

the parameters $\beta_0$ and $\beta_1$ represent the intercept and the coefficient of the covariate respectively; For the last term $Z(\mathbf{s})$, we choose a Gaussian random field with mean $\mathbf{0}$ and a kernel function from the Matérn class. The expression of the Matérn kernel at $\mathbf{s}_1$ and $\mathbf{s}_2$ satisfies

$$K_{\boldsymbol{\theta}}(\mathbf{s}, \mathbf{s}') = \frac{\sigma^2}{2^{\nu-1}\Gamma\left(\nu\right)} \left(\kappa\|\mathbf{s} - \mathbf{s}'\|\right)^\nu K_\nu\left(\kappa\|\mathbf{s} - \mathbf{s}'\|\right), \tag{30}$$

where $\|\mathbf{s} - \mathbf{s}'\|$ is the Euclidean distance between $\mathbf{s}$ and $\mathbf{s}'$, $\sigma^2$ is the marginal variance, $\kappa > 0$ is the scale parameter, and $\nu > 0$ is the smoothness parameter, $K_\nu(\cdot)$ is the modified Bessel function of the second kind. In Case I we set the intercept and covariate coefficient to $(\beta_0, \beta_1) = (7.3, 2.5)$, and the hyperparameters of the Matérn random field to $\boldsymbol{\theta} = \left(\sigma^2, \kappa, \nu\right) = (0.5, 0.25, 0.5)$. In Case II we set the intercept and covariate coefficient to $(\beta_0, \beta_1) = (7.3, 0.5)$, and the hyperparameters of the Matérn random field to $\boldsymbol{\theta} = \left(\sigma^2, \kappa, \nu\right) = (0.5, 0.25, 2.5)$.

Figure 3 presents the LGCP intensity functions and simulated point patterns under Cases I and II.



(a) Fixed effect (b) Case I (c) Case II

Figure 3: Visualization of the fixed effect and simulated data. (a) Fixed effect. (b) Posterior log-intensity (Case I) with observed points. (c) Posterior log-intensity (Case II) with observed points. Case I closely matches the fixed effect, whereas Case II deviates more.

**Parameter initialization settings in the VoGCAM algorithm** In our method, we should first select $\mathbf{Z}(\mathbf{s})$ as the Matérn field defined in (30), which is widely used in spatial statistics to model random fields (Stein, 1999). Next, we discuss the selection of hyperparameters $\nu$ and $\kappa$.Empirically, the smoothness parameter $\nu$ is often difficult to estimate well due to multi-modality issues in spatial statistics (Finley et al., 2019). Therefore, $\nu$ is typically set to common values such as $0.5$ or $1.5$. In our setting, we initialize it as $\nu = 0.5$. For estimating $\kappa$, we adopt the empirical estimation method proposed by Lindgren et al. (2011) by set $\kappa = \sqrt{8\nu}/\rho$, where $\rho$ is the range parameter. In practice, $\rho$ can be set to the maximum Euclidean distance between any two points within the current observation domain. Thus, by determining the values of $\nu$ and $\rho$, we can determine $\kappa$. In our setting, we choose $\rho = \sqrt{2}$, which is corresponding to $\kappa = 1.4$.

**Evaluation metrics** Recall that $\mathcal{D} = \{\mathbf{s}_i\}_{i=1}^N$ is the set of observations in a bounded region $\Omega \subset \mathbb{R}^2$, and $\lambda_0(\mathbf{s})$ denotes the true intensity function. The test set is denoted as $\mathcal{D}_{test} = \{\mathbf{s}_i^{'}\}_{i=1}^{N_{test}}$. We approximate the integral term in the LGCP likelihood via numerical integration:

$$\int_\Omega \lambda(\mathbf{s})\, \mathrm{d}\mathbf{s} \approx \sum_{i=1}^n w_i \exp\{\lambda(\tilde{\mathbf{s}}_i)\},$$

where $\mathcal{I} = \{\tilde{\mathbf{s}}_i\}_{i=1}^n$ are $n$ pre-selected deterministic integration points in $\Omega$, and $\{w_i\}_{i=1}^n$ are the corresponding weights satisfying $\sum_{i=1}^n w_i = |\Omega|$. Denote $n_i$ as the number of points in the test set that fall within the cell centered at $\tilde{\mathbf{s}}_i$. For a fitted intensity function $\hat{\lambda}(\mathbf{s})$, we consider the following evaluation metrics:

- **Training log-likelihood** ($l_{train}$):

$$l_{train} = \sum_{i=1}^N \log \hat{\lambda}(\mathbf{s}_i) - \sum_{i=1}^n w_i \exp\{\hat{\lambda}(\tilde{\mathbf{s}}_i)\}.$$

  This metric measures the fit to the observed training data.

- **Predictive log-likelihood** ($l_{pred}$):

$$l_{pred} = \sum_{i=1}^{N_{test}} n_i \log \hat{\lambda}(\tilde{\mathbf{s}}_i) - \sum_{i=1}^n w_i \exp\{\hat{\lambda}(\tilde{\mathbf{s}}_i)\}.$$

  This metric evaluates predictive performance on a new Poisson realization with the same intensity function as the training data, and is used in the simulation study.

- **Normalized predictive log-likelihood** ($l_{pred}^N$):

$$l_{pred}^N = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} n_i \log \hat{\lambda}(\tilde{\mathbf{s}}_i) - \log \left[ \sum_{i=1}^n w_i \exp\{\hat{\lambda}(\tilde{\mathbf{s}}_i)\} \right].$$

  This metric discards the scaling of the estimated intensity and is used in real data analysis, where the numbers of training and test points may differ.

- $\ell_2$ **error** ($\ell_2$):

$$\ell_2 = \left\{ \sum_{i=1}^{n} \alpha_i (\hat{\lambda}(\tilde{\mathbf{s}}_i) - \lambda_0(\tilde{\mathbf{s}}_i))^2 \right\}^{1/2}.$$

  Since LGCP intensities tend to be large in high-intensity regions, a small $\ell_2$ error indicates accurate recovery of hotspots.

- **Log-$\ell_2$ error** ($\ell_{2,\log}$):

$$\ell_{2,\log} = \left\{ \sum_{i=1}^{n} \alpha_i (\log \hat{\lambda}(\tilde{\mathbf{s}}_i) - \log \lambda_0(\tilde{\mathbf{s}}_i))^2 \right\}^{1/2}.$$

  This metric emphasizes recovery across both high- and low-intensity regions.

- **Running time**: The computational time required by the core algorithm.

In the simulation study, we consider all evaluation metrics except the normalized predictive log-likelihood. In real data analysis, since the true intensity function is unavailable, we restrict the comparison to the training log-likelihood, normalized predictive log-likelihood, and running time.

**Comparison of Posterior Variance** We compute the posterior variance of VoGCAM, INLA, and VIFRK in Cases I and II (displayed in Figure 4). MINCON and QUASI are not included in this comparison, as they only infer the fixed effect and do not produce a posterior distribution. In Case I, all three methods exhibit higher posterior variance in low-intensity regions, with VoGCAM showing comparatively larger variances than the other two. In Case II, VoGCAM and INLA display similar posterior variances, while VIFRK produces lower variances.

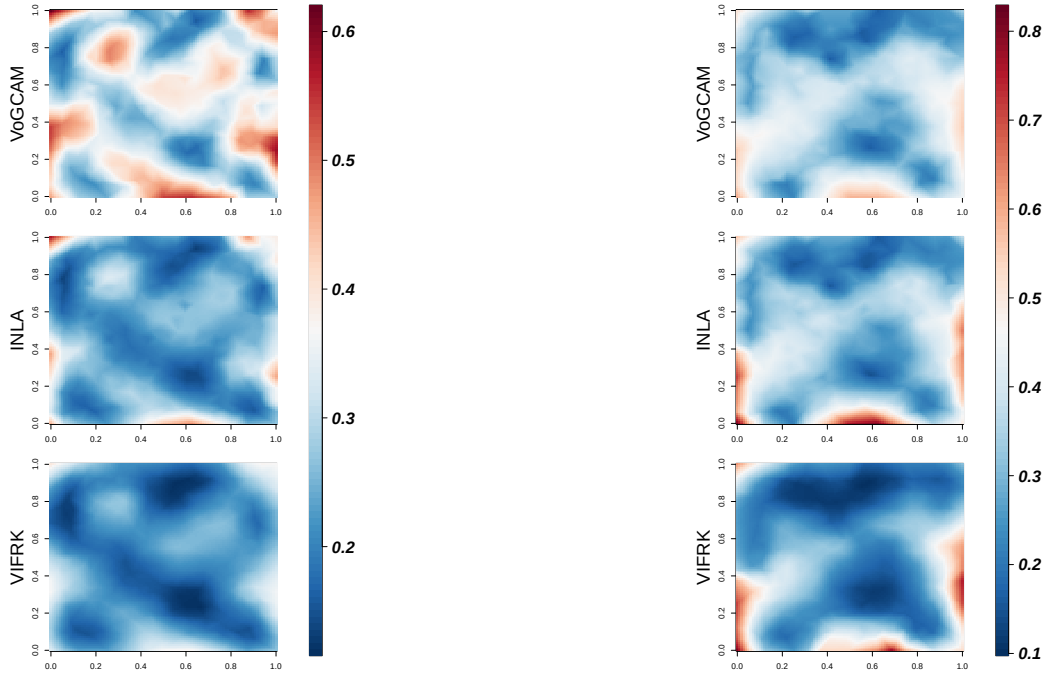



Figure 4: Posterior variance of Case I (left figure) and Case II (right figure) of the log-intensity fitted by the three methods based on the observation points.

**Verification of Algorithmic Convergence.** Under the data pattern as shown in Figure 3 (b), we verify the convergence of the VoGCAM algorithm 2 by selecting two different sets of initial parameter values. Figure 5 illustrates the evolution of the target parameters and the ELBO across iterations for two different initializations. At iteration $k$, we quantify parameter updates by the Euclidean

norms $\|\delta\boldsymbol{\beta}\|_2$ and $\|\delta\boldsymbol{\mu}\|_2$, and by the Frobenius norm $\|\delta\boldsymbol{\Sigma}\|_{\mathrm{F}}$. The corresponding ELBO value is $E\left(\boldsymbol{\beta}_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k; \sigma_k\right)$. In both experiments, all parameter sequences and the ELBO converge. Hence, once the ELBO values stabilize, we consider the algorithm to be converged.

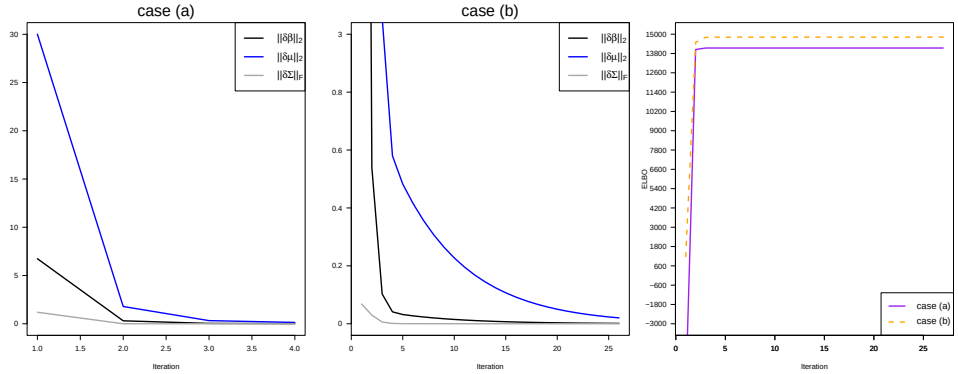



Figure 5: The relative error of the target parameters changing with the number of iterations under two different sets of initial values. The relative error $\|\delta\mathbf{x}\|$ at the $k$-th step is defined as $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|$. The rightmost figure shows the ELBO changing with the number of iterations under the initial values corresponding to the two initializations.

**Sensitivity Analysis** We evaluate the performance of VoGCAM under different values of $M$, the number of neighbors used in constructing the NNGP matrices. Table 5 presents results for $M = \{3, 5, 10, 20, 30, 50, 100\}$ under Cases I and II. In both scenarios, the accuracy of VoGCAM appears to converge as $M$ increases, while the computational time grows substantially. These results suggest that $M = 10$ is sufficient for achieving a good balance between accuracy and efficiency.

Table 5: Comparison of methods under Cases I and II with varying $M$. Metrics include training log-likelihood ($l_{train}$), predictive log-likelihood ($l_{pred}$), $\ell_2$ error of intensity function ($\ell_2$), $\ell_2$ error of log-intensity function ($\ell_{2,\log}$), and running time (Time in seconds).

| $M$ | Case I | | | | | Case II | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $l_{train}$ | $l_{pred}$ | $\ell_2$ | $\ell_{2,\log}$ | Time | $l_{train}$ | $l_{pred}$ | $\ell_2$ | $\ell_{2,\log}$ | Time |
| 3 | 17329 | 14165 | 1023 | 0.36 | 1.25 | 18448 | 14901 | 1457 | 0.63 | 0.51 |
| 5 | 17004 | 14174 | 947 | 0.35 | 1.51 | 17854 | 14918 | 1583 | 0.64 | 0.73 |
| 10 | 16852 | 14167 | 933 | 0.36 | 1.35 | 17565 | 14916 | 1687 | 0.65 | 0.84 |
| 20 | 16830 | 14168 | 936 | 0.36 | 1.70 | 17576 | 14919 | 1675 | 0.65 | 1.05 |
| 30 | 16840 | 14169 | 937 | 0.36 | 1.75 | 17615 | 14920 | 1666 | 0.64 | 1.05 |
| 50 | 16843 | 14170 | 939 | 0.35 | 2.44 | 17642 | 14920 | 1657 | 0.64 | 1.56 |
| 100 | 16840 | 14170 | 939 | 0.35 | 3.50 | 17644 | 14920 | 1655 | 0.64 | 4.05 |

**Ablation Study: Voronoi vs. Uniform Grid.** To assess the effect of the integration point selection strategy, we compare Voronoi tessellation with uniform grid cell sampling under otherwise identical model and algorithm settings. In the simulation study, we use a $20 \times 20$ uniform grid and a Voronoi tessellation with a comparable number of cells, and repeat the comparison over 10 runs. We report the mean and standard error (SE) of the paired differences, defined as the results obtained using Voronoi tessellation minus those obtained using uniform grid cell sampling. For the real-world datasets, we follow the same comparison protocol and evaluate the normalized predictive log-likelihood differences over 10 runs. The results are reported in Tables 6 and 7, respectively. Positive differences in log-likelihood and negative differences in error indicate better performance of Voronoi integration.

**Results Across Multiple Runs.** To assess the variability of the results in Case II, we repeat the experiments 10 times using different training and test datasets. For each repetition, we compute the paired differences between VoGCAM and each baseline method under the same experimental

conditions, and report their mean and standard error (SE) in Table 8. This paired comparison accounts for variation shared across methods on the same replicate and directly summarizes their performance differences. Positive differences in log-likelihood indicate an advantage for VoGCAM, whereas negative differences in error and running time indicate lower errors and shorter running times, respectively.

Table 6: Ablation study comparing Voronoi tessellation with uniform grid cell sampling in the simulation study. Entries report the mean and standard error (SE) of paired differences (Voronoi minus uniform grid) over 10 runs.

| Metric | Mean difference | SE |
|---|---|---|
| Training log-likelihood ($l_{\text{train}}$) | $-6906.86$ | $343.46$ |
| Predictive log-likelihood ($l_{\text{pred}}$) | $156.23$ | $43.97$ |
| $\ell_2$ error | $-184.22$ | $86.02$ |
| Log-$\ell_2$ error ($\ell_{2,\log}$) | $-0.35$ | $0.08$ |

Table 7: Ablation study comparing Voronoi tessellation with uniform grid cell sampling on real-world datasets. Entries report the mean and standard error (SE) of paired differences in normalized predictive log-likelihood (Voronoi minus uniform grid) over 10 runs.

| Dataset | Mean difference | SE |
|---|---|---|
| Neuronal | $0.0028$ | $0.0027$ |
| Transcriptomics | $0.3647$ | $0.0671$ |

Table 8: Paired performance differences between VoGCAM and baseline methods in Case II over 10 runs. Each entry reports the mean difference (SE), computed as VoGCAM minus the baseline method. Positive differences in log-likelihood and negative differences in error and running time favor VoGCAM. Running time is measured in seconds.

| Baseline | $l_{\text{train}}$ | $l_{\text{pred}}$ | $\ell_2$ | $\ell_{2,\log}$ | Time |
|---|---|---|---|---|---|
| INLA | $1868.822$ (35.1956) | $15.347$ (2.8344) | $42.019$ (13.5976) | $0.0079$ (0.0022) | $-1.1955$ (0.0579) |
| VIFRK | $1785.068$ (29.2655) | $79.017$ (5.4376) | $-155.4998$ (14.1296) | $-0.0357$ (0.0062) | $0.7441$ (0.0111) |
| MINCON | $2943.896$ (34.106) | $1074.426$ (7.2371) | $-1391.0596$ (27.5436) | $-0.7899$ (0.0039) | $0.1311$ (0.0779) |
| QUASI | $2952.060$ (34.0194) | $1081.281$ (6.8232) | $-1399.6409$ (27.2083) | $-0.8169$ (0.0049) | $-1.8679$ (0.0864) |

**Investigation on Transcriptomics Data** To investigate why MINCON and QUASI achieve better predictive performance on the transcriptomics dataset, we examine the effect of the random-effect variance $\sigma^2$. A smaller $\sigma^2$ shrinks the random-effect mean $\boldsymbol{\mu}$ toward zero, making the fitted model rely more heavily on fixed effects. Table 9 shows that a large $\sigma^2 = 10^2$ leads to a substantially higher training log-likelihood but poorer predictive performance, suggesting overfitting. As $\sigma^2$ decreases, the training log-likelihood approaches $16{,}000$, while the normalized predictive log-likelihood improves to approximately $-0.02$, approaching the performance of MINCON and QUASI. This suggests that the random effect may induce overfitting in this dataset, helping explain the strong predictive performance of the fixed-effect methods.

Table 9: Performance of VoGCAM under different prior variances $\sigma^2$. Metrics include training log-likelihood ($l_{train}$), normalized predictive log-likelihood ($l^N_{pred}$), and running time (Time in seconds).

| $\sigma^2$ | $l_{train}$ | $l^N_{pred}$ | Time |
|---|---|---|---|
| $10^2$ | 65763.50 | -3.961 | 10.983 |
| $10^0$ | 17517.94 | -0.946 | 2.597 |
| $10^{-2}$ | 16298.20 | -0.117 | 0.193 |
| $10^{-4}$ | 16007.65 | -0.025 | 0.099 |
| $10^{-6}$ | 16002.47 | -0.024 | 0.097 |

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly present the proposed VoGCAM method and its main contributions. These claims are consistent with the theoretical development and experimental results provided in the paper, and do not overstate the scope or performance of the method.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations of the current framework, including its restriction to single-structure LGCP models. Extending the method to more general multi-structure Cox processes introduces significant modeling and inference challenges, which are left for future work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provides theoretical analysis of the proposed method, including convergence properties and convexity of the ELBO. The assumptions are clearly stated, and complete derivations and proofs are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The objective function and iteration steps are clearly described in the main text, and experimental settings are provided in the appendix. While some implementation details (e.g., stopping criteria and specific NNGP implementations) are not exhaustively specified, they follow standard practices and are not expected to materially affect the main conclusions of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The code for the proposed method and simulation experiments are available in the supplementary materials. The neuronal dataset used in the experiments is publicly accessible and properly cited. However, the spatial transcriptomics dataset is privately held and cannot be released due to data access restrictions. Detailed descriptions of experimental settings are provided to facilitate reproducibility on similar datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The data generation process and train-test splits are clearly specified in the paper. All baseline models and evaluation metrics are introduced with detailed descriptions of their respective characteristics. The training procedures, optimization schemes, and implementation details of the proposed method are provided in the main text and appendix, including the hyperparameter settings reported in the appendix. These descriptions collectively ensure that the experimental results can be reproduced to the extent required to understand and validate the main conclusions of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The experiments are conducted under fixed data splits and deterministic inference procedures, so no repeated runs or statistical significance tests are performed. Since the proposed method does not involve stochastic training or random initialization, the reported results are expected to be stable under the same settings. Moreover, the quantitative results are consistent with the qualitative visualizations of the estimated intensity functions.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments were conducted on a standard personal computer using CPU-based implementations. The computational cost of all methods is moderate, and each individual experiment can be reproduced within a reasonable amount of time. No specialized GPU clusters or large-scale computing resources were required.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

Justification: The research involves synthetic experiments, publicly available datasets, and a privately held spatial transcriptomics dataset used under appropriate data access permissions. The study does not involve human subjects, identifiable personal data, or deployed systems. The work is purely methodological and does not raise identifiable ethical concerns. Therefore, the study is consistent with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper focuses on methodological development for inference in log-Gaussian Cox processes and does not target a specific application or deployment scenario. While the method may be applied in domains such as ecology or biomedical research, it is a general statistical methodology and does not directly introduce societal implications within the scope of this work. As such, no specific societal impacts are discussed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release models or datasets with potential for misuse. The proposed method is purely methodological and does not involve high-risk applications such as generative models or sensitive data dissemination. Therefore, no specific safeguards are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The publicly available neuronal dataset used in the paper is properly cited, and its source and access information are provided. The spatial transcriptomics dataset is privately held and used under appropriate data access permissions, and is not publicly released due to usage restrictions. All baseline methods and software tools are appropriately credited. Therefore, all existing assets are used in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, models, or code packages as reusable assets. Therefore, no additional documentation for released assets is required.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The study does not involve human subjects or crowdsourcing experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve human subjects; hence IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology and experiments do not involve the use of large language models. Therefore, this question is not applicable.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.