Title

Scalable Gaussian process models for changepoint detection and spatio-temporal predictions with large correlated data

Author

Li, Hanmo

Publication Date

2024

Peer reviewed|Thesis/dissertation

University of California
Santa Barbara

Scalable Gaussian process models for changepoint detection and spatio-temporal predictions with large correlated data

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Statistics and Applied Probability

by

Hanmo Li

Committee in charge:

Professor Mengyang Gu, Chair
Professor Yuedong Wang
Professor Wendy Meiring
Professor Somayeh Dodge

March 2024

The Dissertation of Hanmo Li is approved.

Professor Yuedong Wang

Professor Wendy Meiring

Professor Somayeh Dodge

Professor Mengyang Gu, Committee Chair

March 2024

Scalable Gaussian process models for changepoint detection and spatio-temporal predictions with large correlated data

by

Hanmo Li

To my beloved parents, for their endless love and support,
shaping my journey with their guidance and faith.
And to my girlfriend, for her enduring support and kindness, a
beacon of light in my life's journey.

Acknowledgements

My journey to this point has been both challenging and enriching, and many have contributed to my success. Foremost, I owe a deep sense of gratitude to Prof. Mengyang Gu, my advisor. His guidance has been more than academic; he has taught me how to navigate life's challenges with grace and determination. Prof. Gu's clear, constructive feedback and regular one-on-one meetings have been instrumental in solving research quandaries. His support extended beyond academia, helping me in my career development, and encouraging my application to the NSF-MSGI summer internship as well as the internship at Capital One. His belief in me has been a constant source of motivation.

I am equally thankful to Prof. Yuedong Wang for his invaluable support and advice on the COVID-19 project. His mentorship has been a lesson in professionalism and perseverance, and I have greatly benefited from his insights into research methodologies and coding styles.

A special mention to my managers and co-workers at Capital One – Rongwen Wu, Heng Sun, Chandra Banerjee, and Tangxin Jin. Their guidance made my summer internship a period of significant learning and personal growth. The work environment they fostered was both enjoyable and conducive to productivity.

I am much obliged to the National Institute of Health and National Science Foundation. My research at UCSB is supported by NIH under award number R01DK130067, NSF under award number 2053423, and the UCSB Office of Research COVID-19 seed grant program. I am also grateful to Fresenius Medical Care North America for providing de-identified data of health records for dialysis patients.

Lastly, I extend my heartfelt thanks to my friends Huiyu Jiang and Haozhe Yang. Their companionship during the isolating times of the COVID-19 quarantine was more than just support; it was a beacon of hope and camaraderie. We shared struggles and triumphs, and their presence made those challenging times bearable.

This acknowledgment would be incomplete without mentioning the entire community that has supported me in various ways. Each one of you has left an indelible mark on my journey, and for that, I am eternally grateful.

Curriculum Vitæ

Hanmo Li

Education

2024	Ph.D. in Statistics and Applied Probability (Expected), University of California, Santa Barbara.
2018	M.S. in Data Science, University of Wisconsin, Madison.
2017	B.S. in Statistics, Shandong University.

Experience

2023	Data Scientist Intern, Capital One Bank.
2022	NSF-MSGI Summer Intern, Larwrence Livermore National Laboratory.
2020	Graduate Student Researcher, Department of Statistics and Applied. Probability, University of California, Santa Barbara.
2018-2024	Graduate Teaching Assistant, Department of Statistics and Applied Probability, University of California, Santa Barbara.

Publications

[1] "A High Computationally Efficient Parallel Partial Gaussian Process for Large-scale Power System Probabilistic Transient Stability Assessment." Ye, Ketian, Junbo Zhao, Hanmo Li, Mengyang Gu. IEEE Transactions on Power Systems (2023).

[2] "Predicting SARS-CoV-2 infection among hemodialysis patients using multimodal data." Duan, Juntao, Hanmo Li, Xiaoran Ma, Hanjie Zhang, Rachel Lasky, Caitlin K Monaghan, Sheetal Chaudhuri, Len A Usvyat, Mengyang Gu, Wensheng Guo, Peter Kotanko, Yuedong Wang. Frontiers in Nephrology, 1179342, vol. 3 (2023).

[3] "Gaussian orthogonal latent factor processes for large incomplete matrices of correlated data." Gu, Mengyang, Hanmo Li. Bayesian Analysis 17, no. 4 (2022).

[4] "Robust estimation of SARS-CoV-2 epidemic in US counties." Li, Hanmo, Mengyang Gu. Scientific Reports 11, no. 1 (2021).

Preprints

[1] "Sequential Kalman filter of fast online changepoint detection for correlated data." Li, Hanmo, Yuedong Wang, Mengyang Gu. arXiv preprint arXiv:2310.18611 (2023).

Presentations

University of California, Santa Barbara February 2023

Graduate Student Research Lightning talks

University of California, Santa Barbara February 2022

Graduate Student Research Lightning talks

University of California, Santa Barbara March 2021

Graduate Student Research Lightning talks

Software

"SKFCPD: Fast Online Changepoint Detection for Temporally Correlated Data" Li, Hanmo, Yuedong Wang, Mengyang Gu. R package version 0.2.4 (2023).

Abstract

Scalable Gaussian process models for changepoint detection and spatio-temporal predictions with large correlated data

by

Hanmo Li

Uncertainty generally exists in various research stages, including experimentation, model formulation, input specification, parameter estimation, and predictions. Therefore, quantifying the uncertainty through statistical inference is essential for different disciplines, including physics, chemistry, biology, geography, ecology, epidemiology, and power systems management. The Gaussian process (GP) model is a suitable choice for predicting nonlinear relationships in different applications, due to the availability of uncertainty assessment and statistical efficiency. However, its application to large-scale datasets is limited by computational challenges, primarily because computing the inverse covariance matrix and the determinant of the covariance matrix in the likelihood function requires $O(n^3)$ operations, where n is the number of observations. To address this issue, we develop fast GP models by utilizing the connection between GPs with Matérn kernels of 1D input and the dynamic linear model, which enables the use of Kalman filter and Rauch-Tung-Striebel smoother for theoretically reducing the computational complexity. The connection enables us to develop fast algorithms for changepoint detection, predicting spatio-temporal data or functional data with multi-dimensional inputs. We focus on two main objectives for applications. First, motivated by assessing the COVID-19 pandemic since 2020, we introduce two new models for patient-level and regional-level detection. The first model aims to detect changepoints in patients' biomarker data efficiently and identify the COVID-19 infection date for each patient in the dialysis facilities.

The second model aims to detect the transmission dynamics for the COVID-19 pandemic in more than 3,000 US counties and update the analysis on a weekly basis. The second objective of the thesis is to develop efficient computation of the GP model for spatio-temporal data and functional data with high-dimensional inputs. We develop fast algorithms for latent factor processes with an orthogonal factor loading matrix, particularly for scalable computations on large, incomplete lattice datasets. We further study the GP models for predicting computer simulations of power systems with high-dimensional inputs and outputs and outline a few future research goals.

Chapter 1 introduces the background of the GP model and the connection to the dynamic linear model or linear state space model. We also review the Kalman filter and Rauch-Tung-Striebel smoother for efficiently computing dynamic linear models. Chapters 2 and 3 focus on applying the GP model in COVID-19 research. Chapter 2 introduces the sequential Kalman filter online changepoint detection (SKFCPD) algorithm for detecting changes in temporally correlated data modeled by GP, which has linear computational complexity at each time step without any approximation. One challenge is to include a large number of the covariates in the model, while a large proportion of the covariates are missing. We propose a two-step approach that integrates the classification methods with the SKFCPD algorithm for fast and accurate detection of COVID-19 infection dates based on patients' biomarker data. Chapter 3 introduces a new mathematical model that integrates the information of regional cases and death counts. By utilizing the GP model for quantifying the uncertainties in predictions, our model can provide real-time, robust estimation with uncertainty quantification of COVID-19 transmission dynamics for over 3,000 U.S. counties. Chapter 4 proposes the Gaussian orthogonal latent factor (GOLF) model for efficient computations on large correlated spatial, spatiotemporal, and functional data. Chapter 5 concludes the previous chapters, explores the application of the GP model on large-scale power systems, and discusses future research directions.

Contents

Chapter 1

Introduction

Measurements from the modern world, such as time series, spatio-temporal and functional data, often contain correlation. The probabilistic framework is a natural way of modeling the correlation and assessing uncertainty for forward prediction and inverse estimation. However, there are two grand challenges for statistical learning of correlated data. First, as sample sizes increase, developing a method with high statistical learning efficiency and low computational cost becomes challenging. The Gaussian Process (GP) model is known for its statistical efficiency in learning nonlinear relationships in various applications, including modeling spatially and spatio-temporally correlated data [1, 2, 3], emulating expensive computer simulations [4, 5, 6], and modeling discrepancy function for inverse estimation [7, 8, 9, 10]. Yet the computational bottleneck limits the GP model's application to large-scale data, as calculating the inverse and determinant of the covariance matrix in the likelihood function takes $O(n^3)$ computational operations, where n is the number of observations. This limitation of one-dimensional (1D) inputs can be mitigated by connecting the GP with half-integer Matén covariance [11] to the dynamic linear model (DLM) [12, 13] or linear state space model [14, 15]. The Kalman filter [16] and Rauch-Tung-Striebel smoother [17], or the forward filtering and backward smoothing (FFBS) algorithm, can then be implemented to reduce the computational complexity for problems with 1D inputs to $O(n)$ operations without approximation. This thesis develops new methods that extend these fast algorithms for online changepoint detection of correlated measurements, such as time sequences, and for large incomplete lattice measurements with multiple dimensions for spatially and spatio-temporally correlated data.

The second grand challenge comes from the large dimensionality of the input space in some applications. For spatial or spatio-temporal data, the inputs usually have two and three dimensions, respectively. However, in building fast statistical emulators for expensive computer simulations [4, 18, 19], the dimension of the input space can be large. For instance, the inputs for emulating atomic forces and potential energy from molecular simulation are often assumed to be the pairwise distance of atomic positions [20, 21, 22], which can have thousands of dimensions even for a single molecule with tens of atoms. In other scenarios, the input can be a function, such as the initial conditions and external potential for developing a statistical surrogate model [23, 24, 25]. This thesis studies another application in characterizing power system dynamics. The increasing prevalence of new energy sources such as wind and solar power introduces more uncertainties across various power system parameters. One such parameter is the rotor angles in generators. When the rotor angles surpass the physical upper limits, it escalates the overall risk within the power system. The traditional approach to model this risk is through differential and algebraic equations [26], where the inputs, including loads and photovoltaic devices (PVs), have more than 2000 dimensions for predicting a large number of rotor angles for the Texas 2000-bus system [27, 28]. However, this traditional approach is computationally expensive for large-scale power systems with such a high-dimensional input space. Fortunately, for all these applications, the input coordinates are often not independent as correlation exists across nearby input coordinates. This property substantially reduces the input space and enables statistical models for precise and efficient predictions.

This thesis is motivated by two sets of research goals. The first research goal focuses on two applications in the context of the SARS-CoV-2 (COVID-19) outbreak. Since 2020, this pandemic has led to millions of deaths in the U.S. alone. The two applications we study, including (1) detecting the COVID-19 infection timing at the patient level and (2) modeling COVID-19 local transmission dynamics at the county level are both vital issues that could facilitate medical institutions and local governments to mitigate the pandemic. In the first application, we focus on dialysis patients, as these patients typically are older and in compromised health status, resulting in higher rates of COVID-19 infection and mortality [29]. Additionally, these patients visit the dialysis clinics approximately three times per week, where various biomarker measurements are routinely collected, thus providing ample data for our research. In this study, we utilize a large dataset of daily biomarker information for over 150,000 dialysis patients from January 2020 to March 2022 collected by Fresenius Medical Care. This dataset includes a wide range of biomarkers such as body temperature, blood pressure, weight, respiration rate, pulse rate, oxygen level, interdialytic weight gain, average blood flow rate, and average dialysis flow rate [30, 31]. Previous studies on COVID-19 infection detection algorithms [32, 33, 30] in patient-level longitudinal data rely on statistical learning algorithms such as random forecast [34, 35] and XGBoost [36] of patients, and utilize a predetermined threshold of infection probabilities to identify infections. The threshold of methods is held the same for all patients and temporal correlation in the longitudinal data is not considered in the modeling, which can be restrictive for detecting COVID-19, as noises from the data can lead to abrupt changes in the prediction probabilities of infections and patients with mild or moderate symptoms may have an increasing trend of probabilities that may not exceed the threshold. To address this challenge, we develop new online changepoint detection algorithms that incorporate temporal patterns in the prediction probabilities. Online changepoint detection algorithms aim to identify the time when the distribution properties, such as mean, variance, or correlation, of a data sequence change, while the data is being observed or collected sequentially. Integrating temporal correlations within probability sequences has been considered in online changepoint detection studies but a scalable computational algorithm remains to be a difficult task.

Several studies have attempted to integrate temporal correlation into online changepoint detection algorithms, utilizing methods such as piecewise polynomial regression models [37], which, however, overlook within-segment temporal correlations. Other strategies include detecting mean shifts in series with autocorrelated noise [38] and employing GP models with rank 1 updates [39], whereas the computational cost is still high. To address this issue, we propose a GP-based online changepoint detection algorithm in Chapter 2. Our proposed method develops a new approach, called the sequential Kalman filter, to achieve efficient and accurate changepoint detections in temporally correlated data with linear computational complexity with respect to the sample size at each time step. The second application is to evaluate the transmission dynamics of the COVID-19 pandemic across over 3,000 U.S. counties every week. Researchers build epidemiology compartmental models [40, 41, 42, 43, 44], stochastic agent-based models [45, 46], branching processes [47], and network analysis [48] to monitor the COVID-19 transmission dynamics. However, most of them focus on states or counties in a short time period, ignoring smaller ones with fewer populations and COVID-19 infection cases. To address this issue, we develop a method that integrates the discretized SIRDC model with the GP model in Chapter 3, enabling robust estimation and uncertainty quantification of COVID-19 transmission dynamics across over 3,000 U.S. counties, leading to precise forecast for counties with both large and small populations.

The second research goal of this thesis is to develop fast GP models for spatial data, spatio-temporal data, and functional data with high-dimensional inputs. Various studies aim to approximate GP models for massive data, including Vecchia approximation [49, 50], inducing point approach [51], stochastic partial differential equation approach [52, 53], hierarchical nearest neighbor methods [54], multi-resolution process [55], local Gaussian process approach [56], periodic embedding [57, 58] and covariance tapering [59]. These approaches have limitations, such as the constraints of the number of neighbors or induced points, and the dependence between the neighbors and prediction inputs. To address this issue, we develop fast algorithms for latent factor processes with an orthogonal factor loading matrix in Chapter 4, particularly for scalable computations on large, incomplete lattice datasets, which are usual for spatial datasets, such as satellite radar interferograms [60, 61]. In this new approach, we found the posterior distributions of the factor processes are independent if the basis is orthogonal and the prior factor processes are independent. This key property enables the use of the Kalman filter to accelerate the computation. Furthermore, for problems with high-dimensional input, reliable predictions in the entire input space can be challenging. Fortunately, inputs are often correlated in practice, which leads to small intrinsic input dimensions. This characteristic offers possibilities for emulating the behavior of systems with high dimensional input by statistical emulators. In this thesis, we explore the application of the parallel partial Gaussian process model [62] for predicting computer simulations of power systems with high-dimensional inputs and outputs in Section 5.1.

In the following sections of this chapter, we will provide a brief introduction to GP models and its connection with dynamic linear models. Additionally, we will review the Kalman filter for forecast computation without approximation.

1.1 Gaussian Process Model

We denote $y(\mathbf{x}) \in \mathbb{R}$ as a real-valued outcome with a p-dimensional real-valued input $\mathbf{x} \in \mathbb{R}^p$. The Gaussian stochastic process (GaSP) or the Gaussian process (GP) model for noisy outcome $y(\cdot)$ is denoted by $y(\cdot) \sim \mathcal{GP}(\mu(\cdot), \sigma^2 c(\cdot,\cdot) + \sigma_0^2)$, with a mean function $\mu()$, a variance parameter σ^2, a correlation function $c(\cdot,\cdot)$ and a noise variance parameter σ_0^2. For a set of inputs $\{\mathbf{x}_1, ..., \mathbf{x}_n\}$, the marginal distribution of the corresponding outcome vector follows a multivariate normal distribution [4]

$$\left(y\left(\mathbf{x}_1\right), \ldots, y\left(\mathbf{x}_n\right)\right)^T \mid \boldsymbol{\beta}, \sigma^2, \sigma_0^2, \mathbf{R} \sim \mathcal{MN}\left(\left(\mu\left(\mathbf{x}_1\right), \ldots, \mu\left(\mathbf{x}_n\right)\right)^T, \sigma^2 \mathbf{R} + \sigma_0^2 \mathbf{I}_n\right), \quad (1.1)$$

where σ^2 is the unknown variance parameter, $\mathbf{R}$ is the correlation matrix with the (i, j)-th element represented by the correlation function $c(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{I}_n$ is an identity matrix with size n. It is common to model the mean $\mu(\mathbf{x})$ as linear combinations of the basis functions,

$$\mu(\mathbf{x}) = \mathbf{h}(\mathbf{x})\boldsymbol{\beta} = \sum_{t=1}^{q} h_t(\mathbf{x})\beta_t, \quad (1.2)$$

where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), ..., h_q(\mathbf{x}))$ is a vector of basis functions and $\boldsymbol{\beta} = (\beta_1, ..., \beta_q)^T$ is a vector of unknown regression parameters for the basis function $\mathbf{h}(\mathbf{x})$.

There are two common ways correlation function $c(\cdot,\cdot)$, namely the *isotropic* correlation and *product* correlation. An isotropic correlation function means that $c(\cdot,\cdot)$ is a function that depends on $d_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$, for any $\mathbf{x}_i = (x_{i1}, ..., x_{ip})^T$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jp})^T$, where $||\cdot||$ is the Euclidean distance.

This isotropic assumption can be restrictive for describing the correlation in some applications. For instance, some variables can have different units and the correlation lengthscales can be different for each dimension of the inputs. The product correlation function is often assumed for emulating computer model experiments [5],

$$c(\mathbf{x}_i, \mathbf{x}_j) = \prod_{l=1}^{p} c_l(x_{il}, x_{jl}), \tag{1.3}$$

where $c_l(\cdot, \cdot)$ is the one-dimensional correlation function for the l-th coordinate of the input vector. As we define the product correlation, the correlation matrix can be decomposed as follows,

$$\mathbf{R} = \mathbf{R}_1 \circ \mathbf{R}_2 \circ \ldots \circ \mathbf{R}_p, \tag{1.4}$$

where $\mathbf{R}_l$ is the correlation matrix on the l-th coordinate of the input vector with the (i, j)-th element determined by $c_l(x_{il}, x_{jl})$ and $\circ$ denotes the Hadamard product.

Next, we discuss the commonly used correlation functions. The most frequently used correlation function is the square exponential (SE) correlation with the roughness parameter $\alpha = 2$, which is also called Gaussian correlation, defined as

$$c_l(x_{il}, x_{jl}) = \exp\left(-\frac{d_{ijl}^2}{\gamma_l}\right), \tag{1.5}$$

where γ_l is the range parameter on the l-th input dimension and $d_{ijl} = |x_{il} - x_{jl}|$. This correlation function is infinitely mean square differentiable [63], meaning that the GP with the SE correlation is very smooth. However, infinitely differentiable processes can be overly smooth for some computer experiments and real datasets [64]. The SE correlation is a special case of the power exponential correlation with the following definition,

$$c_l(x_{il}, x_{jl}) = \exp\left(-\frac{d_{ijl}^\alpha}{\gamma_l}\right), \tag{1.6}$$

where the parameter $\alpha \in (0, 2]$. Notably, when $\alpha < 2$, the power exponential correlation function is not once differentiable, resulting in a process that might be too rough for certain experiments and datasets.

Another common choice is to use Matérn correlation with the roughness parameter $\alpha \in (0, +\infty)$, which is defined as

$$c_l(x_{il}, x_{jl}) = \frac{1}{2^{\alpha-1}\Gamma(\alpha)} \left(\frac{d_{ijl}}{\gamma_l}\right)^{\alpha} \mathcal{K}_{\alpha}\left(\frac{d_{ijl}}{\gamma_l}\right), \tag{1.7}$$

where $\Gamma(\cdot)$ is the gamma function and $\mathcal{K}_{\alpha}(\cdot)$ is the modified Bessel function of the second kind. The Matérn correlation owns many appealing properties. Firstly, the Matérn correlation is $(\lceil \alpha \rceil - 1)$-times mean square differentiable [63], which means we could directly control the smoothness of the process by modifying the roughness parameter α. When α goes to positive infinity, the Matérn correlation converges to the SE correlation. Moreover, the Matérn correlation has the closed-form expression when the roughness parameter α is half-integer, i.e. $\alpha = \frac{2m-1}{2}$, where $m \in \mathbb{N}^{+}$. A common choice with the Matérn correlation function is by setting the roughness parameter $\alpha = \frac{1}{2}$, which is also known as the exponential kernel,

$$c_l(x_{il}, x_{jl}) = \exp\left(-\frac{d_{ijl}}{\gamma_l}\right). \tag{1.8}$$

The range parameter γ_l determines the correlation between two inputs x_{il} and x_{il}. Specifically, a larger value of γ_l results in a stronger correlation, and conversely, a smaller γ_l leads to a weaker correlation. In the application, the roughness parameter α is usually prespecified and the range parameter γ_l is estimated from the data.

Another common choice with the Matérn correlation function is by setting the roughness parameter $\alpha = \frac{5}{2}$, which has the following expression,

$$c_l(x_{il}, x_{jl}) = \left(1 + \frac{\sqrt{5}d_{ijl}}{\gamma_l} + \frac{5d_{ijl}^2}{3\gamma_l^2}\right) \exp\left(\frac{\sqrt{5}d_{ijl}}{\gamma_l}\right). \tag{1.9}$$

Next, we discuss parameter estimation in the GP model. For a GP model with a Matérn correlation function and the roughness parameter $\alpha = \frac{5}{2}$, based on Equations (1.1) and (1.9), there are four sets of parameters to estimate, including the mean parameters $\boldsymbol{\beta}$, signal variance parameter σ^2, noise variance parameter σ_0^2 and the range parameters $\boldsymbol{\gamma} = \{\gamma_1, \ldots, \gamma_p\}$. Bayesian inference is a natural way that considers the uncertainty in parameter estimations. A common choice is the objective Bayesian inference for parameter estimations, specifically utilizing the standard reference prior for these parameters [65, 66] as follows,

$$\pi^R(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}. \tag{1.10}$$

The likelihood function of a GP model on the outcome sequence $\mathbf{y} = (y(\mathbf{x}_1), ..., y(\mathbf{x}_n))^T \in \mathbb{R}^n$ has the following expression,

$$p\left(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}, \eta\right) = \left(2\pi\sigma^2\right)^{-\frac{n}{2}} |\mathbf{K}|^{-\frac{1}{2}} \exp\left(-\frac{(\mathbf{y} - \mathbf{H}\boldsymbol{\beta})^T \mathbf{K}^{-1} (\mathbf{y} - \mathbf{H}\boldsymbol{\beta})}{2\sigma^2}\right) \tag{1.11}$$

where $\mathbf{K} = (\mathbf{R} + \eta \mathbf{I}_n)$ with the nugget parameter $\eta = \frac{\sigma_0^2}{\sigma^2}$ defined as the ratio of noise variance to signal variance, $\mathbf{H} = (\mathbf{h}(\mathbf{x}_1), ..., \mathbf{h}(\mathbf{x}_n))^T$ is a n-by-q basis function matrix. The marginal distribution of the range parameters $\boldsymbol{\gamma}$ and nugget parameter η, $p(\mathbf{y} \mid \boldsymbol{\gamma}, \eta) \propto \int p\left(\mathbf{y} \mid \boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}, \eta\right) \pi^R(\boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2$, is as follows,

$$p(\mathbf{y} \mid \boldsymbol{\gamma}, \eta) \propto |\mathbf{K}|^{-\frac{1}{2}} \left|\mathbf{H}^T \mathbf{K}^{-1} \mathbf{H}\right|^{-\frac{1}{2}} \left(S^2\right)^{-\frac{(n-q)}{2}} \tag{1.12}$$

where $S^2 = (\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}})^T \mathbf{K}^{-1} (\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}})$ and $\hat{\boldsymbol{\beta}} = \left(\mathbf{H}^T \mathbf{K}^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{K}^{-1} \mathbf{y}$. Consequently, parameters $\boldsymbol{\gamma}$ and η can be estimated by maximizing the marginal likelihood,

$$(\hat{\boldsymbol{\gamma}}, \hat{\eta}) = \underset{\gamma, \eta}{\operatorname{argmax}} \left\{ p(\mathbf{y} \mid \boldsymbol{\gamma}, \eta) \right\}, \tag{1.13}$$

where the numerical optimization is commonly conducted by the Quasi-Newton optimization method [67, 68]. Furthermore, the robust estimation for the range and nugget parameters through maximizing marginal posterior with objective prior can be achieved by utilizing the robust GaSP emulator [69] and the jointly robust prior [70].

After obtaining the estimate of the range parameter $\boldsymbol{\gamma}$ and the noise variance to variance ratio η, the predictive distribution of $y(\mathbf{x}^*)$, where $\mathbf{x}^*$ is a new input, given $\mathbf{y}$ and estimated parameters $\hat{\boldsymbol{\gamma}}$ and $\hat{\eta}$, can be obtained by integrating out the mean parameter $\boldsymbol{\beta}$ and the variance parameter σ^2.

$$p(y\left(\mathbf{x}^*\right) \mid \mathbf{y}, \hat{\boldsymbol{\gamma}}, \hat{\eta}) = \int p(y\left(\mathbf{x}^*\right) \mid \mathbf{y}, \boldsymbol{\beta}, \sigma^2, \hat{\boldsymbol{\gamma}}, \hat{\eta}) \pi^R(\boldsymbol{\beta}, \sigma^2) d\boldsymbol{\beta} d\sigma^2 \tag{1.14}$$

Given the standard reference prior $\pi^R(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}$, the predictive distribution $p(y\left(\mathbf{x}^*\right) \mid \mathbf{y}, \hat{\boldsymbol{\gamma}}, \hat{\eta})$ follows a Student t-distribution with $n - q$ degrees of freedom.

$$y\left(\mathbf{x}^*\right) \mid \mathbf{y}, \hat{\gamma}, \hat{\eta} \sim \mathcal{T}\left(\hat{y}\left(\mathbf{x}^*\right), \hat{\sigma}^2 c^{**}, n - q\right), \tag{1.15}$$

where

$$\hat{y}\left(\mathbf{x}^*\right) = \mathbf{h}\left(\mathbf{x}^*\right) \hat{\boldsymbol{\beta}} + \mathbf{r}^T\left(\mathbf{x}^*\right) \mathbf{K}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \tag{1.16}$$

$$\hat{\sigma}^2 = \frac{1}{(n-q)}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}})^T \mathbf{K}^{-1}(\mathbf{y} - \mathbf{H}\hat{\boldsymbol{\beta}}), \tag{1.17}$$

and

$$\begin{aligned} c^{**} =& c\left(\mathbf{x}^{*}, \mathbf{x}^{*}\right) + \hat{\eta} - \mathbf{r}^{T}\left(\mathbf{x}^{*}\right)\mathbf{K}^{-1}\mathbf{r}\left(\mathbf{x}^{*}\right) + \Big(\mathbf{h}\left(\mathbf{x}^{*}\right) - \mathbf{r}\left(\mathbf{x}^{*}\right)^{T}\mathbf{K}^{-1}\mathbf{H}\Big) \\ & \times \left(\mathbf{H}^{T}\mathbf{K}^{-1}\mathbf{H}\right)^{-1}\Big(\mathbf{h}\left(\mathbf{x}^{*}\right) - \mathbf{r}\left(\mathbf{x}^{*}\right)^{T}\mathbf{K}^{-1}\mathbf{H}\Big)^{T} \end{aligned} \tag{1.18}$$

with $\hat{\boldsymbol{\beta}} = \left(\mathbf{H}^{T}\mathbf{K}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{T}\mathbf{K}^{-1}\mathbf{y}$ being the generalized least squares estimator for the mean parameter $\boldsymbol{\beta}$ and $\mathbf{r}(\mathbf{x}^{*}) = (c(\mathbf{x}^{*}, \mathbf{x}_1), ..., c(\mathbf{x}^{*}, \mathbf{x}_n))^{T}$ is obtained by plugging in the estimated range parameter $\hat{\boldsymbol{\gamma}}$. The predictive mean can be written as a weighted average of the data or bases (see Corollary 1 in [71]), which is often used for predictions. An advantage of the GP model lies in the uncertainty quantification of the predictions from the predictive distribution in Equation (1.15).

However, calculating the predictive mean and variance in Equations (1.16) and (1.17) or the likelihood function can be computationally expensive for a large number of observations. The main roadblock is the inversion of the correlation matrix $\mathbf{K}$ of size n, which has a computational complexity of $O(n^3)$. To tackle this issue, we first introduce the connection between the GP model having Matérn correlation with half-integer roughness parameters and the Dynamic Linear Model. The Kalman filter approach can be used to accelerate the computation of the likelihood function for problems with 1D inputs. This approach reduces the complexity to linear time complexity $\mathcal{O}(n)$ with respect to the sample size without any approximation. This connection will be used to develop new algorithms for changepoint detection and GPs with multi-dimensional inputs.

1.2 Connections between the Gaussian Process Model and the Dynamic Linear Model

In this section, we build the connection between the GP model of one-dimensional inputs: $\mathbf{t} = \{t_1, \ldots, t_n\}$ with $t_i \leq t_j$ for $i < j$ and the dynamic linear model (DLM). A DLM has the following expression, for $k = 1, ..., n$,

$$\begin{aligned} y(t_k) &= \mu(t_k) + \mathbf{F}_k \boldsymbol{\theta}_k + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}\left(0, \sigma_0^2\right), \\ \boldsymbol{\theta}_k &= \mathbf{G}_k \boldsymbol{\theta}_{k-1} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{MN}\left(0, \mathbf{W}_k\right), \end{aligned} \tag{1.19}$$

where $\mu(t_k)$ is the mean parameter defined in Equation (1.2), $\boldsymbol{\theta}_k$ is a m-dimensional latent state process with the initial state $\boldsymbol{\theta}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{B}_0)$, $\mathbf{F}_k$ is a $1 \times m$ vector, $\mathbf{B}_0$, and $\mathbf{G}_k$ and $\mathbf{W}_k$ are $m \times m$ matrices.

A GP model with a Matérn correlation function, as expressed in Equation (1.7) and a half-integer roughness parameter α, can equivalently be represented through a DLM [72, 11]. For instance, the Matérn correlation function with a roughness parameter $\alpha = 0.5$, also called the Exponential correlation function, follows

$$c(t_i, t_j) = \exp\left(-\frac{d_{ij}}{\gamma}\right), \tag{1.20}$$

where $d_{ij} = t_j - t_i$ for $i < j$ and γ is the range parameter. The GP model defined in Equation (1.1) with exponential correlation function in Equation (1.20) is equivalent to a DLM in Equation (1.19) with $F_k = 1$, $G_k = \rho_k$, $W_k = \sigma^2\left(1 - \rho_k^2\right)$, $\rho_k = \exp\left(-\frac{d_{ij}}{\gamma}\right)$, and $B_0 = \sigma^2$. The Matérn with roughness parameter being 2.5, as another example, follows

$$c(t_i, t_j) = \left(1 + \frac{\sqrt{5} d_{ij}}{\gamma} + \frac{5 d_{ij}^2}{3\gamma^2}\right) \exp\left(-\frac{\sqrt{5} d_{ij}}{\gamma}\right). \tag{1.21}$$

As shown in [11, 73], the DLM in Equation (1.19) is equivalent to the GP model defined in Equations (1.1) and (1.21), where $\lambda = \frac{\sqrt{5}}{\gamma}$ and $d_k = t_k - t_{k-1}$,

$$\mathbf{F}_k = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},$$

$$\mathbf{G}_k = e^{-\frac{\lambda d_k}{2}} \begin{bmatrix} \lambda^2 d_k^2 + 2\lambda + 2 & 2\lambda d_k^2 + 2d_k & d_k^2 \\ -\lambda^3 d_k^2 & -2(\lambda^2 d_k^2 - \lambda d_k - 1) & 2 - \lambda d_k^2 \\ \lambda^4 d_k^2 - 2\lambda^3 d_k & 2(\lambda^3 d_k^2 - 3\lambda^2 d_k) & \lambda^2 d_k^2 - 4\lambda d_k + 2 \end{bmatrix},$$

$$\mathbf{W}_k = \frac{4\sigma^2\lambda^5}{3} \begin{bmatrix} W_{1,1}^k & W_{1,2}^k & W_{1,3}^k \\ W_{2,1}^k & W_{2,2}^k & W_{2,3}^k \\ W_{3,1}^k & W_{3,2}^k & W_{3,3}^k \end{bmatrix},$$

with

$$W_{1,1}^k = \frac{e^{-2\lambda d_k}(3 + 6\lambda d_k + 6\lambda^2 d_k^2 + 4\lambda^3 d_k^3 + 2\lambda^4 d_k^4) - 3}{-4\lambda^5},$$

$$W_{1,2}^k = W_{2,1}^k = \frac{e^{-2\lambda d_k}}{2},$$

$$W_{1,3}^k = W_{3,1}^k = \frac{e^{-2\lambda d_k}(1 + 2\lambda d_k + 2\lambda^2 d_k^2 + 4\lambda^3 d_k^3 - 2\lambda^4 d_k^4) - 1}{4\lambda^3},$$

$$W_{2,2}^k = \frac{e^{-2\lambda d_k}(1 + 2\lambda d_k + 2\lambda^2 d_k^2 - 4\lambda^3 d_k^3 + 2\lambda^4 d_k^4) - 1}{-4\lambda^3},$$

$$W_{2,3}^k = W_{3,2}^k = \frac{e^{-2\lambda d_k} d_k^2 (4 - 4\lambda d_k + \lambda^2 d_k^2)}{2},$$

$$W_{3,3}^k = \frac{e^{-2\lambda d_k}(-3 + 10\lambda d_k - 22\lambda^2 d_k^2 + 12\lambda^3 d_k^3 - 2\lambda^4 d_k^4) + 3}{4\lambda}, \text{ and}$$

$$\mathbf{B}_0 = \begin{bmatrix} \sigma^2 & 0 & -\sigma^2\lambda^2/3 \\ 0 & \sigma^2\lambda^2/3 & 0 \\ -\sigma^2\lambda^2/3 & 0 & \sigma^2\lambda^4 \end{bmatrix}.$$

Note that GP with the Matérn correlation function having $\alpha \geq 3/2$ is a differentiable, continuous time process, which is not often used in modeling time series, e.g. financial time series are often assumed to be continuous but not differentiable. Thus, this connection extends the class of the DLM to be used for modeling time series in practice.

After connecting the GP model with the DLM, we could speed up the calculation of the likelihood function of the GP model by leveraging the Kalman filter approach. The computation complexity can be reduced from cubic to linear for univariate outcomes. In the next section, we introduce the Kalman filter approach and its implementation in the GP model.

1.3 The Kalman Filter

In this section, we briefly discuss the Kalman filter [16, 74], approach for all DLMs, which contains the GP with half-integer Matérn covariance and 1D input. As shown in [12, 13], the Kalman filter can be applied to compute the likelihood and the RTS smoother can be used to make the predictions for the DLM with linear computational complexity. The Kalman filter will be extended to construct the new sequential Kalman filter for changepoint detection in Section 2. Specifically, the implementation of the Kalman filter approach on the DLM in Equation (1.19) can be divided into three steps.

Lemma 1 *(Kalman Filter on DLM) Denote* $\mathbf{y}_{1:k} = (y(t_1), ..., y(t_k))^T$*. For* $k = 1, \dots, n$*, given* $\boldsymbol{\theta}_{k-1} \mid \mathbf{y}_{1:(k-1)} \sim \mathcal{MN}\left(\mathbf{m}_{k-1}, \mathbf{M}_{k-1}\right)$*, we iteratively compute the distribution of* $\boldsymbol{\theta}_k$ *given* $\mathbf{y}_{1:k}$ *by the following three steps, where the DLM is defined in Equation (1.19),*

1. *we compute the one-step-ahead predictive distribution of* $\boldsymbol{\theta}_k$ *given* $\mathbf{y}_{1:(k-1)}$

$$\boldsymbol{\theta}_k \mid \mathbf{y}_{1:(k-1)} \sim \mathcal{MN}\left(\mathbf{b}_k, \mathbf{B}_k\right), \tag{1.22}$$

with $\mathbf{b}_k = \mathbf{G}_k \mathbf{m}_{k-1}$ *and* $\mathbf{D}_k = \mathbf{G}_k \mathbf{M}_{k-1} \mathbf{G}_k^T + \mathbf{W}_k$*.*

2. *Next, we compute the one-step-ahead predictive distribution of $y(t_k)$ given $\mathbf{y}_{1:(k-1)}$ below,*

$$y(t_k) \mid \mathbf{y}_{1:(k-1)} \sim \mathcal{N}\left(f_k, Q_k\right), \tag{1.23}$$

with $f_k = \mathbf{F}_k \mathbf{b}_k$, and $Q_k = \mathbf{F}_k \mathbf{B}_k \mathbf{F}_k^T + \sigma_0^2$.

3. *In the last step, we compute the filtering distribution of $\boldsymbol{\theta}_k$ given $\mathbf{y}_{1:k}$, i.e.,*

$$\boldsymbol{\theta}_k \mid \mathbf{y}_{1:k} \sim \mathcal{MN}\left(\mathbf{m}_k, \mathbf{M}_k\right), \tag{1.24}$$

with $\mathbf{m}_k = \mathbf{b}_k + \mathbf{B}_k \mathbf{F}_k^T Q_k^{-1}\left(y(t_k) - f_k\right)$ and $\mathbf{M}_k = \mathbf{B}_k -\ \mathbf{B}_k \mathbf{F}_k^T Q_k^{-1} \mathbf{F}_k \mathbf{B}_k$.

Next, we briefly discuss how to implement the Kalman filter in Lemma 1 to achieve efficient computation of the likelihood for the GP model. As discussed in Section 1.1, the main roadblock for computing the likelihood is to inverse the correlation matrix $\mathbf{K}$, which has $O(n^3)$ computational complexity. Denote $\mathbf{K} = \mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix in the Cholesky decomposition of $\mathbf{K}$. To efficiently compute the likelihood function in Equation (1.11), we need to evaluate two terms, $|\mathbf{K}|$ and $\mathbf{L}^{-1}\mathbf{y}$ with low computational complexity. Fortunately, with the Kalman filter, the one-step-ahead predictive distribution $y(t_k) \mid \mathbf{y}_{1:(k-1)} \sim \mathcal{N}(f_k, Q_k)$ can be derived iteratively for $k = 1, ..., n$, where each iteration only takes $O(1)$ operation. As the likelihood function of $\mathbf{y}_{1:n}$ is a product of predictive probabilities, i.e. $p(\mathbf{y}_{1:n}) = p\left(y(t_1)\right) \prod_{k=2}^{n} p\left(y(t_k) \mid \mathbf{y}_{1:(k-1)}\right)$, we could write the likelihood function in this following form [75],

$$p\left(\mathbf{y}_{1:n} \mid \sigma^2, \sigma_0^2, \gamma\right) = \prod_{k=1}^{N} \left(2\pi Q_k\right)^{-\frac{1}{2}} \exp\left\{-\sum_{k=1}^{N} \frac{\left(y(t_k) - f_k\right)^2}{2Q_k}\right\} \tag{1.25}$$

where the computation can be done in linear time $O(n)$ rather than the cubic time complexity. Further, we have the following expressions for the computationally expensive terms in the likelihood function,

$$|\mathbf{K}| = \prod_{k=1}^{n} Q_k, \quad \text{and} \quad \mathbf{L}^{-1}\mathbf{y} = \left(\frac{y(t_1) - f_1}{\sqrt{Q_1}}, \ldots, \frac{y(t_n) - f_n}{\sqrt{Q_n}} \right)^T. \tag{1.26}$$

According to Lemma 1, the Kalman filter parameters $f_1, \ldots, f_n$ and $Q_1, \ldots, Q_n$ can be sequentially computed with an overall computational complexity of $O(n)$. This allows us to compute the computationally expensive terms $|\mathbf{K}|$ and $\mathbf{L}^{-1}\mathbf{y}$ in $O(n)$, enabling the whole likelihood function of the GP model to be computed in a linear time.

Furthermore, we can obtain the predictive distribution of the parameter $\boldsymbol{\theta}_t$ in the DLM as introduced in Equation (1.19) with linear computational complexity using the Rauch–Tung–Striebel (RTS) smoother [17]. The RTS smoother's application to DLM [12, 13] is defined as follows.

Lemma 2 *(RTS smoother on DLM) Consider observations* $\mathbf{y}_{1:n} = (y(t_1), \ldots, y(t_n))$*. If the predictive distribution* $\boldsymbol{\theta}_{k+1} \mid \mathbf{y}_{1:n} \sim \mathcal{MN}(\mathbf{s}_{k+1}, \mathbf{S}_{k+1})$*, then*

$$\boldsymbol{\theta}_k \mid \mathbf{y}_{1:n} \sim \mathcal{MN}(\mathbf{s}_k, \mathbf{S}_k), \quad \textit{for } k = 1, \ldots, n, \tag{1.27}$$

where

$$\begin{aligned} \mathbf{s}_k &= \mathbf{m}_k + \mathbf{M}_k \mathbf{G}_{k+1}^T \mathbf{B}_{k+1}^{-1} (\mathbf{s}_{k+1} - \mathbf{b}_{k+1}) \\ \mathbf{S}_k &= \mathbf{M}_k - \mathbf{M}_k \mathbf{G}_{k+1}^T \mathbf{B}_{k+1}^{-1} \left(\mathbf{B}_{k+1} - \mathbf{S}_{k+1} \right) \mathbf{B}_{k+1}^{-1} \mathbf{G}_{k+1} \mathbf{M}_k \end{aligned} \tag{1.28}$$

By applying the Kalman filter forward, as introduced in Lemma 1, to observations $\mathbf{y}_{1:n}$, we can compute the distribution $\boldsymbol{\theta}_n \mid \mathbf{y}_{1:n}$ with $O(n)$ complexity. Subsequently, by applying the RTS smoother backward from $k = n$ to 1, as described in Lemma 2, we obtain the predictive distributions $\boldsymbol{\theta}_k \mid \mathbf{y}_{1:n}$ also in $O(n)$.

1.4 Challenges and Research Questions

The development of the statistical models in this thesis tackles three main problems: scalable computation when the number of observations is large, robust estimation for a small number of noisy observations, efficient predictions and uncertainty quantification when the data has high-dimensional inputs and outputs.

The first challenge is illustrated by the dialysis patients' biomarker data, collected by Fresenius Kidney Care North America. This dataset includes daily treatment and laboratory features of over 150,000 dialysis patients from January 2020 to March 2022. Based on this dataset, we aim to develop an algorithm to detect COVID-19 infection dates for each dialysis patient. Several studies have tried to detect COVID-19 infection at the patient level using longitudinal data [32, 33, 30]. These models often calculate prediction probabilities for COVID-19 infection and use a predetermined threshold to determine infections, a method that overlooks the temporal patterns within the data. For example, patients with mild or moderate symptoms of COVID-19 may show a gradual increase in prediction probabilities, but the prediction probabilities might still fall below the pre-specified threshold. Ignoring such a trend can lead to low sensitivity in detection. To address this issue, online changepoint detection algorithms could be applied to capture the temporal patterns in the prediction probabilities. However, the challenge remains in modeling temporal correlations in the probability sequences efficiently (see Figures A.1 and A.2 of autocorrelation in Appendix A). Various studies have attempted to integrate temporal correlation into changepoint detection algorithms. One approach [37] employs a piecewise polynomial regression model to account for temporal correlations across different segments, whereas it assumes independence within each temporal segment. Another approach [38] focuses on detecting shifts in the mean within time series affected by autocorrelated noise. Moreover, the GP model has been used for modeling temporal covariance between observations at each time point for changepoint detection [39] whereas the computational cost is huge. However, none of the methods has enough efficiency and flexibility to model the temporal correlations and capture various types of changepoints.

Chapter 2 discusses a method that can efficiently detect various types of changepoints in temporally correlated data. It uses the GP model to capture the temporal correlation and leverages the sequential Kalman filter to achieve linear computation complexity at each time step. Our simulated experiments demonstrate that this method outperforms other online changepoint detection approaches in accurately detecting shifts in mean, variance, or correlation within temporally correlated data. Moreover, we propose a new way to integrate classification and changepoint detection approaches that improve the detection delay and accuracy for detecting COVID-19 infection compared to other alternatives.

The second issue addressed in this thesis is to obtain robust computation for heterogeneous and noisy data. This issue is illustrated in the context of COVID-19 as well, where we aim to estimate the heterogeneous progression of SARS-CoV-2 in over 3,000 U.S. counties. The main challenge is to provide robust estimation and uncertainty quantification of the COVID-19 transmission parameters for small counties, as the variability for the COVID-19 infection cases and deaths is large due to the small number of observations for these counties, particularly at the beginning of the epidemic. A wide range of models has been applied to analyze the transmission dynamics of COVID-19. The epidemiology compartmental models such as SIR, SEIR, SIRD, and their extensions [40, 41, 42, 43, 44], stochastic agent based models [45, 46], branching processes [47], and network analysis [48] have advanced our understanding of transmission rates and incubation period of SARS-CoV-2, which are connected to the traffic flow and mobility during the COVID-19 outbreaks at different regions [76, 77]. The disease progression characteristics, such as the transmission rate, are often estimated based on the daily death toll [40, 43, 45, 46]. However, the focus has predominantly been on larger states and counties, leaving smaller ones less studied.

Chapter 3 discussed a robust methodology that combines the discretized SIRDC model with the GP model to accurately estimate and quantify the uncertainties of COVID-19 transmission dynamics across U.S. counties, including smaller ones. Furthermore, we propose a metric called the daily probability of contracting (PoC) SARS-CoV-2 for a susceptible individual to quantify the risk of SARS-CoV-2 transmission in a community. This work yields a dynamic map at the county level, aiding local officials in formulating effective policies and informing the public about the daily risks of contracting SARS-CoV-2.

The third issue discussed in this thesis is the computation challenge of the GP model when the data has massive high-dimensional correlated output. Numerous studies have been made to approximate a GP model for such datasets in recent studies, including, for example, stochastic partial differential equation approach [52, 53], hierarchical nearest neighbor methods [54], multi-resolution process [55], local Gaussian process approach [56], periodic embedding [57, 58] and covariance tapering [59], which have obtained wide attention in recent years. However, efficient computation without direct approximation to the likelihood function remains less explored. In Chapter 4, we introduce a new orthogonal factor process to model spatial and spatio-temporal data in incomplete lattice without directly approximating the likelihood function. Additional approximation can be applied to use a small number of factor processes to further reduce the computation.

1.5 Outline

Chapter 2 focuses on efficiently and accurately detecting COVID-19 infection dates by patients' biomarker data. Traditional methods typically rely on data-driven models with prespecified threshold values to identify the infection date, which leads to miss detections for patients with mild symptoms. We can address this issue by integrating the data-driven model with changepoint detection algorithms, as these algorithms could track temporal trends in the data. One of the widely used methods is the Bayesian Online changepoint detection (BOCPD) algorithm [78, 79], which is empirically shown to be the most efficient method across various real-world datasets [80]. However, the data are assumed to be independently distributed in BOCPD, which can be restrictive to longitudinal biomarker data of patients. To overcome this issue, we introduce the Sequential Kalman Filter for Online Changepoint Detection (SKFCPD) algorithm in this chapter. This approach utilizes the GP model to account for temporal correlations and incorporates the sequential Kalman filter approach to achieve linear computational complexity at each time step. We evaluate the SKFCPD method's speed and accuracy in detecting infection dates using COVID-19 patients' biomarker data compared to other data-driven and changepoint detection methods.

In Chapter 3, we introduce an epidemic model that efficiently estimates the COVID-19 daily transmission dynamics in real-time across more than 3,000 U.S. counties. This model categorizes the population in each county into five groups: Susceptible, Infectious, Resolving, Deceased, and Recovered (SIRDC). It offers robust daily COVID-19 infection and death toll estimates in each county. Furthermore, the model offers a 21-day forecast of the death toll and integrates the GP model to generate posterior confidence intervals for this forecast. Additionally, a metric named the Probability of Contracting (PoC) COVID-19 is introduced. This metric calculates the daily average probability that a healthy individual in any given county will contract COVID-19, providing crucial insight into county-level COVID-19 transmission dynamics.

Chapter 4 introduces a GP model for large correlated spatial, spatiotemporal, and functional lattice data. Although there is a vast literature on the efficient implementation of the GP model on spatial data, most studies rely on the sparse assumption over the correlation matrix, which could be unrealistic for highly correlated spatial data. To address this, the chapter proposes a GP model with independent latent factor processes and an orthogonal factor loading matrix. This design simplifies the decomposition of the likelihood function, allowing for the application of the Kalman filter to each component, thereby reducing computational complexity. Another advantage of this model is that it can handle irregular missing values in the data, which is essential for fields like weather forecasting or precipitation prediction, where satellite data frequently contains gaps. To demonstrate the accuracy and efficacy of our model, we analyze real satellite data of sea surface temperatures, comparing our GP model's predictive accuracy with existing models.

Chapter 5 concludes the works presented in the previous chapters and outlines future research directions in three fields: 1) online changepoint detection methods, 2) epidemiology compartmental model, and 3) GP models for data with massive output. In the area of online changepoint detection, the SKFCPD method introduced in Chapter 2 could be further extended to handle multidimensional datasets and more complex change dynamics. To enhance the estimation accuracy of the epidemiology compartmental model, the SIRDC model introduced in Chapter 3 could be extended to include more flexible pandemic parameters and integrate additional spatial information. To accelerate the computation of GP models with multi-dimensional inputs, the model introduced in Chapter 4 has the potential to be applied to non-lattice data with irregular missingness. Another important research direction involves modeling data with both massive high-dimensional output and inputs. In Chapter 5, we investigate the Partial Gaussian Process (PPGP) model and its application to large-scale power systems and outline limitations and future directions of the topics in Chapter 2-4.

Chapter 2

Sequential Kalman filter for fast online changepoint detection in longitudinal health records

In this chapter, we introduce the sequential Kalman filter, a computationally scalable approach for online changepoint detection with temporally correlated data. The temporal correlation was not considered in the Bayesian online changepoint detection approach due to the large computational cost. Motivated by detecting COVID-19 infections for dialysis patients from millions of daily health records with a large number of covariates, we develop a scalable approach to detect multiple changepoints from correlated data by sequentially stitching Kalman filters of subsequences to compute the joint distribution of the observations, which has linear computational complexity with respect to the number of observations between the last detected changepoint and the current observation at each time point, without approximating the likelihood function. Compared to other online changepoint detection methods, simulated experiments show that our approach is more precise in detecting single or multiple changes in mean, variance, or correlation for temporally correlated data. Furthermore, we propose a new way to integrate classification and changepoint detection approaches that improve the detection delay and accuracy for detecting COVID-19 infection compared to other alternatives.

2.1 Introduction

It is crucial to identify shifts in distribution proprieties from time series or longitudinal data, such as changes in mean, variance, and correlation, a process generally referred to as changepoint detection. Changepoint detection has become a widely utilized technique across various fields [81], including DNA copy number variants [82], financial data [83], power systems [84], meteorology [85] and cellular processes [86], as a changepoint signals a deviation from the baseline data-generating process.

In this work, we develop a computationally scalable and accurate approach to detect changepoints in time-dependent outcomes. Our aim is to detect whether a patient receiving dialysis treatment contracts severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) or COVID-19. The dialysis patient data are collected by Fresenius Kidney Care North America, which operates over 2,400 dialysis clinics in most US states and provides treatment for approximately one-third of the US dialysis patients. The dataset includes daily treatment and laboratory records for over 150,000 dialysis patients from January 2020 to March 2022.

We highlight that the longitudinal detection scenario considered herein is challenging as only around 0.4% of the observations are in the COVID-19 infection periods formally defined in Section 2.4, whereas other studies [30, 87] consider the “cross-sectional” observations, which matches one PCR test record to a few negative records, inducing a data set where $15\% - 20\%$ of the records are COVID-19 positive, $30 - 50$ times higher than our setting which is closer to the real-world setting during the pandemic. The low positive rates in the longitudinal setting make the detection of COVID-19 infections more challenging. Our goal is to develop a new online changepoint detection method for identifying changes from longitudinal health records with a large number of measurements from lab tests, which are common in healthcare practice [88]. Various challenges exist for detecting COVID-19 infection, including large irregular missingness of time-dependent laboratory covariates of patients and temporal correlations in the probability sequences (see Figures A.1 and A.2 of autocorrelation in Appendix A). To address these challenges, we propose an accurate and scalable changepoint detection algorithm that can integrate the results from state-of-the-art classification methods, such as XGBoost, and substantially improve the performance of classification methods.

Our main interest lies in detecting changepoints as new data arrives sequentially, a key aspect distinguishing online from offline changepoint detection scenarios [89, 90, 91, 83, 92, 93, 94]. One popular framework of online change detection is the Bayesian online changepoint detection [78, 79], which was shown to have high accuracy compared to other alternatives [80]. However, one limitation of the Bayesian online changepoint detection is the assumption of mutual independence among observations, while correlations are common for temporal data. A few subsequent studies focus on detecting changepoints in the data with temporal correlations. For example, the study in [39] utilizes a Gaussian process to model temporal correlation within the subsequences separated by changepoints. Although this approach reduces the computational complexity of detecting the change at each time step from $\mathcal{O}(n^4)$ to $\mathcal{O}(n^3)$ computational operations by using rank 1 updates [95] for n observations, the computational complexity is still prohibitively large. The study in [37] models the temporal correlations across segments using piecewise polynomial regression, assuming that correlations are Markov and observations within the same segments are independent. The study in [38] models time series with autocorrelated noise and detects mean changes through dynamic programming recursion that maximizes the penalized likelihood. These approaches do not provide a flexible class of models of the temporal correlation between observations at each time point.

Our main contributions are twofold. First, we propose an efficient online changepoint detection algorithm, applicable for all dynamical linear models commonly used for modeling time sequences [12, 96]. The new algorithm is capable of sequentially detecting multiple changepoints with computational complexity $\mathcal{O}(n')$ at each time, where n' is the number of observations between the last detected changepoint and the current observation, making it significantly more efficient than the Gaussian process changepoint detection algorithm [39]. We achieve this computational order by sequentially stitching Kalman filters of subsequences for computing likelihood and predictive distributions. This approach is generally applicable to all dynamic linear models with equally or unequally spaced time points. Second, when data contain a massive number of observations and high-dimensional covariates with a large proportion of missingness, it is challenging to apply any existing changepoint detection method or state space model directly. Our real application of detecting COVID-19 infection for dialysis patients is one such example, where a large number of lab covariates are missing as patients do not take all lab measurements in each of their visits. To address this challenge, we propose an integrated approach. We first use supervised learning, such as XGBoost, to compute the posterior probability[1] of a time point being a changepoint for all patients. Conventional analysis often proceeds by choosing a threshold for the posterior probability to make detection decisions, which overlooks the changes in the longitudinal data that a changepoint detection algorithm could capture. We apply our changepoint detection algorithm to each patient to detect changes in classification probabilities. We found that the performance was dramatically improved compared to a supervised learning approach alone. The ap-

[1] The posterior probabilities from the XGBoost model are calibrated by a sigmoid transformation to ensure they correspond well with the COVID-19 positive rate in the real-world dataset.

proach is general, as it's adaptable to any statistical machine learning method providing classification probabilities. Additionally, we provide SKFCPD, an R package for efficient implementation of our algorithm, to be released on CRAN along with the publication of this work.

This paper is structured as follows. Section 2.2.1 provides an overview of Bayesian online changepoint detection. In Sections 2.2.2-2.2.3, we introduce the sequential Kalman filter approach, an efficient online changepoint detection algorithm for temporally correlated data, and illustrate the computational advantage over direct computation in Section 2.2.4. In Section 2.3, we demonstrate the advantage of our proposed approach using simulated data with shifts in mean, variance, and correlation. Section 2.4 introduces the new approach that integrates classification methods with the new changepoint detection approach for COVID-19 infection detection. Proofs of lemmas, theorems, and additional numerical results are provided in the Appendix A.

2.2 Fast Online Changepoint Detection for Correlated Data

2.2.1 Background: Bayesian Online Changepoint Detection

Let us consider the time series $\mathbf{y}_{1:n} = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ for time points $\{t_1, \ldots, t_n\}$, such that $t_j < t_i$ for any $1 \leq j < i \leq n$. We assume time segments separated by any two changepoints are independent of each other, whereas data within each segment can be temporally correlated. Each segment can have distinct distributions characterized by different mean, variance, or correlation parameters.

We define C_n as the most recent changepoint at or before the current time point t_n. For instance, if $C_n = t_4$, it indicates that t_4 is the only changepoint in the time period

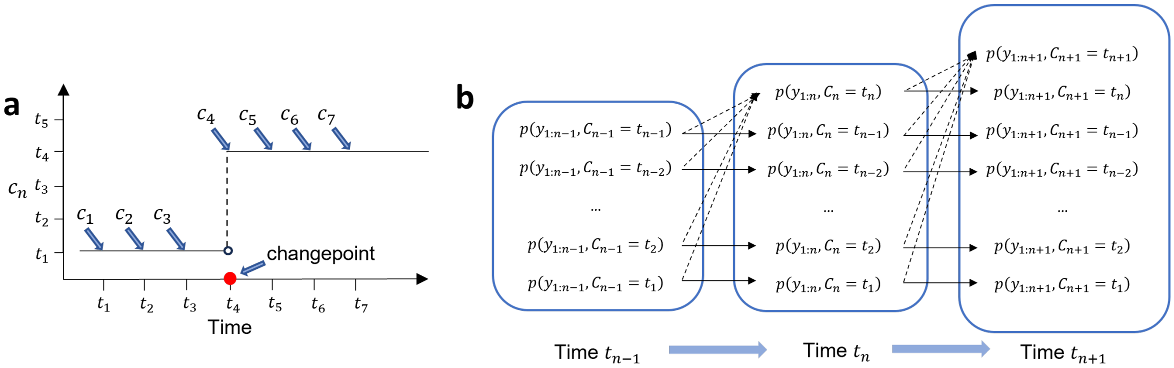

a
t_5
t_4
C_n t_3
t_2
t_1
C_1 C_2 C_3
C_4 C_5 C_6 C_7
changepoint
t_1 t_2 t_3 t_4 t_5 t_6 t_7
Time
b
$p(y_{1:n-1}, C_{n-1} = t_{n-1})$
$p(y_{1:n-1}, C_{n-1} = t_{n-2})$
...
$p(y_{1:n-1}, C_{n-1} = t_2)$
$p(y_{1:n-1}, C_{n-1} = t_1)$
$p(y_{1:n}, C_n = t_n)$
$p(y_{1:n}, C_n = t_{n-1})$
$p(y_{1:n}, C_n = t_{n-2})$
...
$p(y_{1:n}, C_n = t_2)$
$p(y_{1:n}, C_n = t_1)$
$p(y_{1:n+1}, C_{n+1} = t_{n+1})$
$p(y_{1:n+1}, C_{n+1} = t_n)$
$p(y_{1:n+1}, C_{n+1} = t_{n-1})$
$p(y_{1:n+1}, C_{n+1} = t_{n-2})$
...
$p(y_{1:n+1}, C_{n+1} = t_2)$
$p(y_{1:n+1}, C_{n+1} = t_1)$
Time t_{n-1}
Time t_n
Time t_{n+1}

Figure 2.1: Panel a: value of the state C_n that shows the most recent changepoint before or at the time t_n for $n = 1, \ldots, 7$. The time point t_4, marked by a red dot, is the only changepoint before t_7. Panel b: the recursive process of computing the joint distribution $p(\mathbf{y}_{1:n}, C_n)$ from time t_{n-1} to t_{n+1} based on the Equation (2.1). The black arrow means the latter probability can be sequentially computed from the former one.

$[t_4, t_n]$. As shown in Figure 2.1a, where time t_4 is the only changepoint before time t_7, C_n shifts from t_1 to t_4 at time t_4. We define run length, r_n, as the length of the time interval from the most recent changepoint to the current time point, calculated as $r_n = n - C_n + 1$.

The objective of online changepoint detection is to sequentially estimate the changepoint C_n upon receiving a new observation at the current time t_n. A popular online changepoint detection framework is the BOCPD method [78, 79], which has a few assumptions.

Assumption 1 *The segments partitioned by changepoints are mutually independent.*

Assumption 2 *The state on the current time point C_n, conditioning on the state of the previous time point C_{n-1}, is independent of the observations of $\mathbf{y}_{1:(n-1)}$.*

Based on the second assumption, given the previous state $C_{n-1} = t_j$ for $1 \leq j \leq n-1$, C_n can either be t_n if t_n is a changepoint, or remain as t_j if t_n is not a changepoint. Thus, C_n is restricted to either t_j or t_n. Following the BOCPD framework, we define the prior distribution of the conditional distribution of the most recent changepoint as

$p(C_n = t_i \mid C_{n-1} = t_j)$, where it takes the value of $1 - H(t_i)$ if $i = j$, $H(t_i)$ if $i = n$ and is zero in all other cases. $H(\cdot)$ is the hazard function, measuring the probability that a changepoint occurs at any time point.

In BOCPD, the hazard function is often defined as $H(t_i) = \frac{1}{\lambda_i}$, where $\frac{1}{\lambda_i}$ represents the prior probability of time t_i being a changepoint, typically held fixed in practice. For applications such as detecting COVID-19 infection, a time-dependent hazard function can be used to integrate local infection information.

We allow the observations to be mutually dependent within each segment of the changepoints, which relaxes the additional assumption of independence between each observation within one segment in [78]. This modification offers a more realistic modeling for time series data, where observations are often correlated. Furthermore, Assumption 2 means information from the previous observations $\mathbf{y}_{1:(n-1)}$ is contained in C_{n-1}, the latent state indicating whether the previous time point is a changepoint. Based on Assumptions 1 and 2 from BOCPD, we compute the joint distribution of the state $C_n = t_i$ and the observations $\mathbf{y}_{1:n}$ by integrating out the previous state $C_{n-1} = t_j$,

$$\begin{aligned}
&p\left(\mathbf{y}_{1:n}, C_n = t_i\right) \\
=&\underbrace{p\left(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i\right)}_{\text{predictive distribution}} \sum_{j=1}^{n-1} \underbrace{p\left(C_n = t_i \mid C_{n-1} = t_j\right)}_{\text{hazard}} p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j\right) \\
=&\begin{cases} p\left(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i\right)\left(1 - H(t_i)\right) p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_i\right), & i < n, \\ p\left(y_n \mid C_n = t_n\right) H(t_n) \sum_{j=1}^{n-1} p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j\right), & i = n, \end{cases}
\end{aligned} \tag{2.1}$$

where the derivation is given in Section A.2 of the appendix.

After obtaining the joint probability $p(\mathbf{y}_{1:n}, C_n = t_i)$ for $i = 1, \ldots, n$, one can estimate the state $\hat{C}_n$ by calculating the maximum *a posteriori* (MAP) estimate of the joint distribution [79],

$$\hat{C}_n = \operatorname*{argmax}_{t_1 \leq t_i \leq t_n} p(\mathbf{y}_{1:n}, C_n = t_i). \tag{2.2}$$

The probability of $p(\mathbf{y}_{1:n}, C_n = t_i)$ needs to be computed for all possible time points t_i, for $i = 1, ..., n$, to obtain the MAP of the changepoint upon receiving new data at time t_n. Figure 2.1b shows the recursive computational process for the joint probability $p(\mathbf{y}_{1:n}, C_n = t_i)$ in Equation (2.1). At time t_n, we can recursively compute the probability $p(\mathbf{y}_{1:n}, C_n = t_i)$ from the previous step $p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_i)$, where $i < n$, as indicated by the solid black arrows. Furthermore, the probability $p(\mathbf{y}_{1:n}, C_n = t_n)$ can be computed through probabilities of changepoints occurring at previous time points $\{p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j)\}_{j=1}^{n-1}$ and the marginal distribution of the current time point being a changepoint $p(y_n \mid C_n = t_n)$, as shown by the black dashed arrows in Figure 2.1b.

By considering different combinations of changepoints in the joint distribution $p(\mathbf{y}_{1:n}, C_n)$, the recursive formula in Equation (2.1) enables the algorithm to sequentially detect multiple changepoints. For problems with multiple changepoints, we may exclude the observations prior to the most recently detected changepoint to further reduce computational complexity. Specifically, instead of summing over all time indices j from 1 to $n-1$ to compute the joint distribution in Equation (2.1), we can truncate the summation to the range from $t_{\hat{j}}$ to $n-1$, where $t_{\hat{j}} = \hat{C}_{n-1}$ denotes the most recently detected changepoint at or before time t_{n-1}. Note that the estimated most recent changepoint at $(n-1)$th time point $t_{\hat{j}} = \hat{C}_{n-1}$ and $t_j = C_{n-1}$ defined in Equation (2.1) can be different. This approach is more efficient, as two subsequences separated by a changepoint are mutually independent and their distributions can contain distinct parameters. Both simulated and real data studies validate that this approach effectively reduces the computational cost for detecting multiple changepoints without compromising accuracy.

The computation of the joint probability $p(\mathbf{y}_{1:n}, C_n = t_i)$ in Equation (2.1) demands an efficiently evaluation of prediction probabilities $\{p(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i)\}_{i=1}^{n-1}$. To scalably compute the joint distribution, [78] assumed the observations are independent and identically distributed (i.i.d.) random variables with the exponential family of distributions. However, the i.i.d. assumption of observations may not hold for many real-world datasets. To address temporal correlations in time sequences, [79] proposed a method that utilizes the particle filter [97] to approximate the predictive distributions, which may compromise accuracy due to the approximation of the likelihood and the choice of the inducing inputs. To overcome these challenges, we propose a new approach for online changepoint detection applicable for dynamic linear models to model temporally correlated data, which efficiently computes the predictive distributions without approximating the likelihood function.

2.2.2 Dynamic Linear Models for Online Changepoint Detection

Gaussian processes (GPs) have been used to model temporally correlated measurements for online changepoint detection [39]. By Assumption 1, the GP model can have different parameters across segments. The marginal distribution of the $(m+1)$-th segment follows a multivariate normal distribution $\left(\left(y(t_{\tau_m}), \ldots, y(t_{\tau_{m+1}-1})\right)^T \mid \mu_m, \sigma_m^2, \gamma_m, \sigma_{0,m}^2\right) \sim \mathcal{MN}(\mu_m, \sigma_m^2 \mathbf{R}_{\tau_{m+1}-\tau_m} + \sigma_{0,m}^2 \mathbf{I}_{\tau_{m+1}-\tau_m})$, where $\mathbf{R}_{\tau_{m+1}-\tau_m}$ is a $(\tau_{m+1} - \tau_m) \times (\tau_{m+1} - \tau_m)$ correlation matrix having parameter γ_m, with τ_m being the time index of the mth changepoint, for $m = 1, \ldots, M-1$, and when $m = 0$, we let $\tau_0 = 1$. For simplicity, we focus on the time range from t_i to t_n, where t_i is larger than the previously detected changepoint $\hat{C}_{n-1}$. The total number of observations within this segment is denoted by $n' = n - i + 1$. The subscript m in the parameters $(\mu_m, \sigma_m^2, \gamma_m, \sigma_{0,m}^2)$ will be dropped to simplify the notations.

Directly calculating the likelihood and predictive distributions by a GP, however, can be computationally expensive, as it requires computing inversion of the correlation matrix $\mathbf{R}_{n'}$, which takes $\mathcal{O}(n'^3)$ operations. The predictive distribution of the online changepoint detection needs to be calculated numerous times, which further exacerbates the computational challenge. Here we model the temporally dependent observations by dynamic linear models (DLMs) [12, 14, 98], a large class of models for scalable computation.

For simplicity, we denote $y_{i+k-1} = y(t_{i+k-1})$, the real-valued observation at time t_{i+k-1}, which does not need to be equally spaced, for $k = 1, \ldots, n'$. We consider a DLM below,

$$\begin{aligned} y_{i+k-1} &= \mu + \mathbf{F}_k \boldsymbol{\theta}_k + \epsilon_k, \quad \epsilon_k \sim \mathcal{N}\left(0, \sigma_0^2\right), \\ \boldsymbol{\theta}_k &= \mathbf{G}_k \boldsymbol{\theta}_{k-1} + \mathbf{w}_k, \quad \mathbf{w}_k \sim \mathcal{MN}\left(0, \mathbf{W}_k\right), \end{aligned} \tag{2.3}$$

where μ is the mean parameter, $\boldsymbol{\theta}_k$ is a q-dimensional latent state process with the initial state $\boldsymbol{\theta}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{B}_0)$, $\mathbf{F}_k$ is a $1 \times q$ vector, $\mathbf{B}_0$, and $\mathbf{G}_k$ and $\mathbf{W}_k$ are $q \times q$ matrices.

As an example, a GP with a Matérn covariance function that contains half-integer roughness parameters [64, 99] can be written as a DLM [11, 72]. For instance, the Matérn covariance function with a roughness parameter being 0.5 follows

$$\sigma^2 c(t, t') = \sigma^2 \exp\left(-\frac{|d|}{\gamma}\right), \tag{2.4}$$

where $d = t - t'$ and γ is the range parameter, for any t and t'. The GP with covariance in (2.4) is equivalent to a DLM in Equation (2.3) with $F_k = 1$, $G_k = \rho_k$, $W_k = \sigma^2\left(1 - \rho_k^2\right)$, $\rho_k = \exp\left(-\frac{|t_{i+k-1} - t_{i+k-2}|}{\gamma}\right)$, and $B_0 = \sigma^2$.

The Matérn with roughness parameter being 2.5, as another example, follows

$$\sigma^2 c(t, t') = \sigma^2 \left(1 + \frac{\sqrt{5}|d|}{\gamma} + \frac{5d^2}{3\gamma^2}\right) \exp\left(-\frac{\sqrt{5}|d|}{\gamma}\right). \tag{2.5}$$

The equivalent representation of a DLM is discussed in Section 1.2 of Chapter 1. We will use GPs with the two covariance functions in Equations (2.4) and (2.5) for illustrative purposes, but our approach is generally applicable to all DLMs, which includes a much larger class of processes.

From Equation (2.1), to evaluate the joint distribution $p\left(\mathbf{y}_{1:n}, C_n = t_i\right)$ for the last segment $(t_i, \ldots, t_n)$ where $C_n = t_i$ is the most recent changepoint prior to t_n, we need to efficiently compute the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}\right)$. For the computational reason, we define the noise variance to signal variance ratio $\eta = \frac{\sigma_0^2}{\sigma^2}$, and the covariance matrix of observations $\mathbf{y}_{i:n}$ is given by

$$\sigma^2 \mathbf{K}_{n'} = \sigma^2(\mathbf{R}_{n'} + \eta \mathbf{I}_{n'}). \tag{2.6}$$

After this transformation, the parameter set is given by $\mathbf{\Theta} = \{\mu, \sigma^2, \gamma, \eta\}$. In the following, we present the direct computation of the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}\right)$.

Assuming the objective prior for the mean and variance parameter $p(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$, the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$, after integrating out (μ, σ^2), follows,

$$\begin{aligned} p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right) &= \frac{p\left(\mathbf{y}_{i:n} \mid \gamma, \eta\right)}{p\left(\mathbf{y}_{i:(n-1)} \mid \gamma, \eta\right)} \\ &\propto \begin{cases} \frac{\Gamma(\frac{n'-1}{2})}{\Gamma(\frac{n'-2}{2})} \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \exp\left(-S_{n'}^2\right), & i < n-1 \\ \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \left(\mathbf{y}_{(n-1):n}^T \mathbf{M}_{n'} \mathbf{y}_{(n-1):n}\right)^{-1/2}, & i = n-1 \end{cases} \end{aligned} \tag{2.7}$$

where $S_{n'}^2 = \left(\frac{n'-1}{2}\right) \log\left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) - \left(\frac{n'-2}{2}\right) \log\left(\mathbf{y}_{i:(n-1)}^T \mathbf{M}_{n'-1} \mathbf{y}_{i:(n-1)}\right)$ and $\mathbf{M}_{n'} = \mathbf{K}_{n'}^{-1} - \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1}$. Note that when $i = n-1$, there is not enough information to simultaneously integrate out μ and σ^2 for $p(y_n \mid y_{n-1})$. We develop a new procedure for this problem. When $i = n-1$, we first integrate out μ in the joint distribution using the prior probability $\pi(\mu) \propto 1$. Then, we integrate out σ^2 in the predictive distribution $p(y_n \mid y_{n-1})$ using the prior probability $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$. Through simulation and real data analysis, we found that this procedure at $i = n - 1$ provides a stable evaluation of the predictive distribution and avoids mistakenly detected changepoints. The derivation of Equation (2.7) is given in Section A.3 of the appendix.

Directly applying Equation (2.7) to compute the predictive distribution requires $\mathcal{O}\left(n'^3\right)$ computational operations, due to matrix inversion and determinant calculation. This makes the computation impractical as the predictive distribution must be computed for all previous time points. In the following section, we develop the sequential Kalman filter to improve the computational efficiency of the Equation (2.7) without any approximation.

2.2.3 Sequential Kalman Filter for Fast Changepoint Detection

In this section, we introduce a fast algorithm, called sequential Kalman filter (SKF) to reduce the complexity of computing $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$ from $\mathcal{O}(n'^3)$ to $\mathcal{O}(1)$ with $n' = n - i + 1$, for each $i = 1, \ldots, n-1$. First, we discuss the Cholesky decomposition to draw the connection between the predictive distribution and the Kalman filter (KF). Denote the Cholesky decomposition of the correlation matrix as $\mathbf{K}_{n'} = \mathbf{L}_{n'}\mathbf{L}_{n'}^T$, where $\mathbf{L}_{n'}$ is an $n' \times n'$ lower triangular matrix. Consequently, the inverse correlation matrix can be decomposed as $\mathbf{K}_{n'}^{-1} = \mathbf{U}_{n'}^T\mathbf{U}_{n'}$, where $\mathbf{U}_{n'} = \mathbf{L}_{n'}^{-1}$. As computing Cholesky decomposition takes $\mathcal{O}(n'^3)$ operations, we extend the KF in Lemma 3 and Theorem 1 to compute two n'-vectors $\mathbf{u}_{n'} = \mathbf{U}_{n'}\mathbf{1}_{n'}$ and $\mathbf{v}_{i,n'} = \mathbf{U}_{n'}\mathbf{y}_{i:n}$, where $\mathbf{u}_{n'} = (u_1, \ldots, u_{n'})^T$ and $\mathbf{v}_{i,n'} = (v_{i,1}, \ldots, v_{i,n'})^T$ are both n'-vectors. The proofs for Lemma 3 and Theorem 1 are given in Sections A.4 and A.5 of the appendix, respectively.

Lemma 3 *For $k = 1, \ldots, n'$, the kth element of $\mathbf{u}_{n'} = \mathbf{U}_{n'}\mathbf{1}_{n'}$ and $\mathbf{v}_{i,n'} = \mathbf{U}_{n'}\mathbf{y}_{i:n}$ can be* *sequentially computed as follows*

$$u_k = \frac{1 - f^u_k}{\sqrt{Q^u_k}}, \tag{2.8}$$

$$v_{i,k} = \frac{y_{i+k-1} - f^v_{i,k}}{\sqrt{Q^v_{i,k}}}, \tag{2.9}$$

where for $k \geq 2$, we have $f^u_k = \mathbb{E}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{1}_{k-1}, \gamma, \eta] = g^u_k(f^u_{k-1}, Q^u_{k-1})$, $Q^u_k = \mathbb{V}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{1}_{k-1}, \gamma, \eta] = h^u_k(Q^u_{k-1})$, $f^v_{i,k} = \mathbb{E}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{y}_{i:(i+k-2)}, \gamma, \eta] = g^v_{i,k}\left(f^v_{i,k-1}, Q^v_{i,k-1}\right)$, and $Q^v_{i,k} = \mathbb{V}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{y}_{i:(i+k-2)}, \gamma, \eta] = h^v_{i,k}(Q^v_{i,k-1})$, with $\mathbf{Y}_{1:n'}$ denotes a random output vector in a DLM with covariance $\sigma^2\mathbf{K}_{n'}$. The functions $g^u_k(\cdot)$, $h^u_k(\cdot)$, $g^v_{i,k}(\cdot)$, and $h^v_{i,k}(\cdot)$ are given in Equations (A.12)-(A.16) of the appendix.

In Lemma 3, the KF is iteratively applied for computing the parameters f^u_k, Q^u_k, $f^v_{i,k}$, and $Q^v_{i,k}$ from the parameters at the previous time point f^u_{k-1}, Q^u_{k-1}, $f^v_{i,k-1}$, and $Q^v_{i,k-1}$. Once we obtain these parameters, u_k and $v_{i,k}$ can be computed with $\mathcal{O}(1)$ operations for each $k = 1, \ldots, n'$ by using Equations (2.8) and (2.9), respectively. The derivation of Lemma 3 is provided in Section A.4 of the appendix.

Theorem 1 *After obtaining each term of $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$ from Equations (2.8) and (2.9), the predictive distribution in Equation (2.7) can be computed below*

$$p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta) \propto \begin{cases} \frac{\Gamma(\frac{n'-1}{2})}{\Gamma(\frac{n'-2}{2})} \left(Q^u_{n'}\right)^{-\frac{1}{2}} \left(\frac{\mathbf{u}^T_{n'}\mathbf{u}_{n'}}{\mathbf{u}^T_{n'-1}\mathbf{u}_{n'-1}}\right)^{-1/2} \exp\left(-S^2_{n'}\right), & i < n-1 \\ \left(Q^u_{n'}\right)^{-\frac{1}{2}} \left(\frac{\mathbf{u}^T_{n'}\mathbf{u}_{n'}}{\mathbf{u}^T_{n'-1}\mathbf{u}_{n'-1}}\right)^{-1/2} \left(\mathbf{y}^T_{i:n}\mathbf{M}_{n'}\mathbf{y}_{i:n}\right)^{-1/2}, & i = n-1 \end{cases} \tag{2.10}$$

where $S^2_{n'} = \left(\frac{n'-1}{2}\right) \log\left(\mathbf{y}^T_{i:n}\mathbf{M}_{n'}\mathbf{y}_{i:n}\right) - \left(\frac{n'-2}{2}\right) \log\left(\mathbf{y}^T_{i:(n-1)}\mathbf{M}_{n'-1}\mathbf{y}_{i:(n-1)}\right)$ and $\mathbf{y}^T_{i:n}\mathbf{M}_{n'}\mathbf{y}_{i:n} = \mathbf{v}^T_{i,n'}\mathbf{v}_{i,n'} - (\mathbf{u}^T_{n'}\mathbf{u}_{n'})^{-1}(\mathbf{v}^T_{i,n'}\mathbf{u}_{n'})^2$.

When a new observation y_n is available at time t_n, we apply Lemma 3 to update the variables $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$ with $\mathcal{O}(1)$ operations, and compute the predictive distribution

$p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta)$ based on Equation (2.10), which is significantly faster than directly computing the inversion of the correlation matrix in Equation (2.7).

As the estimation of range and nugget parameter (γ, η) typically does not have closed-form expressions, we maximize the likelihood function over a set of training samples from an initial or control period containing no changepoint:

$$(\hat{\gamma}, \hat{\eta}) = \underset{(\gamma,\eta)}{\operatorname{argmax}}\, p(\mathbf{y}_{\mathcal{S}_{tr}} \mid \gamma, \eta), \tag{2.11}$$

where $\mathcal{S}_{tr}$ is an index set of n_{tr} time indices in the training time period. We employ the KF to compute the likelihood function, which only requires $\mathcal{O}(n_{tr})$ operations [75]. We plug the estimated range and nugget parameters into the SKF algorithm for online changepoint detection, effectively capturing the temporal correlations in the data. It is important to note that the mean and variance parameters in the SKF algorithm are integrated out based on all available observations, which enables the algorithm to incorporate the latest information for online changepoint detection. To avoid large computational costs, the range parameters and nugget parameters were estimated using training sequences, similar to the GPCPD approach. In Section 2.3, we empirically show that SKF can accurately detect mean, variance, and correlation changes.

We summarize our approach in Algorithm 1 for detecting the most recent changepoint. First, we apply Theorem 1 multiple times to obtain the sequence of predictive distributions, i.e., $p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta)$ for $i = 1, \dots, n-1$. Next, given the predictive distributions, we compute the joint distribution $p(\mathbf{y}_{1:n}, C_n = t_i)$ using Equations (2.12) and (2.13). Finally, we estimate $\hat{C}_n$, the most recent changepoint before or at time t_n, by the MAP of the joint distribution, i.e., $\hat{C}_n = \operatorname{argmax}_{t_1 \leq t_i \leq t_n} p(\mathbf{y}_{1:n}, C_n = t_i)$.

We call the online changepoint detection approach in Algorithm 1 the *sequential Kalman filter* (SKF) because in Step 2, we sequentially compute the predictive distribu-

Algorithm 1 Sequential Kalman Filter algorithm for fast changepoint detection

Input: New observation y_n, previously estimated changepoint time index $\hat{j}$, previous parameters f^u_{n-1},Q^u_{n-1}, $f^v_{i,n-1}$ and $Q^v_{i,n-1}$ for $\hat{j} \leq i \leq n-1$ defined in by Lemma 3, the joint distribution $p(\mathbf{y}_{1:(n-1)}, C_{n-1} \mid \hat{\gamma}, \hat{\eta})$, estimated nugget and range parameters $(\hat{\gamma}, \hat{\eta})$

Output: The estimated most recent changepoint $\hat{C}_n$, current parameters f^u_n,Q^u_n, $f^v_{i,n}$ and $Q^v_{i,n}$ for $\hat{j} \leq i \leq n-1$ and the joint distribution $p(\mathbf{y}_{1:n}, C_n \mid \hat{\gamma}, \hat{\eta})$

1. **Update parameters through Kalman filter**

   We iteratively compute parameters $\left(f^u_n, Q^u_n, f^v_{i,n}, Q^v_{i,n}\right)$ from $\left(f^u_{n-1}, Q^u_{n-1}, f^v_{i,n-1}, Q^v_{i,n-1}\right)$ for $\hat{j} \leq i \leq n-1$ by Lemma 3.

2. **Compute predictive distributions**

   We sequentially compute the predictive distribution $p(y_n \mid \mathbf{y}_{i:(n-1)}, \hat{\gamma}, \hat{\eta})$ by parameters f^u_n,Q^u_n, $f^v_{i,n}$ and $Q^v_{i,n}$ based on Equation (2.10), for $\hat{j} \leq i \leq n-1$.

3. **Update joint distributions**

   When t_n is not a changepoint, we have $C_n < t_n$. For $\hat{j} \leq i \leq n-1$,

   $$p(\mathbf{y}_{1:n}, C_n = t_i \mid \hat{\gamma}, \hat{\eta}) = p(y_n \mid \mathbf{y}_{i:(n-1)}, \hat{\gamma}, \hat{\eta})\,(1 - H(t_i))\,p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_i \mid \hat{\gamma}, \hat{\eta}). \tag{2.12}$$

   When t_n is a changepoint, we have $C_n = t_n$. Then

   $$p(\mathbf{y}_{1:n}, C_n = t_n \mid \hat{\gamma}, \hat{\eta}) = p(y_n \mid \hat{\gamma}, \hat{\eta}) H(t_n) \sum_{j=\hat{j}}^{n-1} p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j \mid \hat{\gamma}, \hat{\eta}). \tag{2.13}$$

4. **Determine the most recent changepoint by**

   $$\hat{C}_n = \underset{t_{\hat{j}} \leq t_i \leq t_n}{\operatorname{argmax}}\, p(\mathbf{y}_{1:n}, C_n = t_i \mid \hat{\gamma}, \hat{\eta}). \tag{2.14}$$

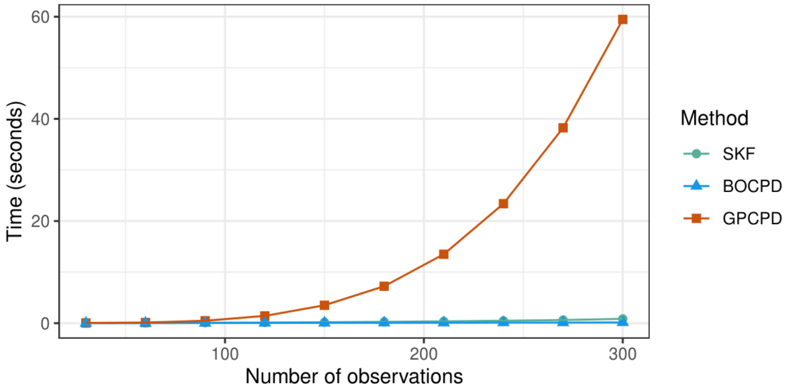


Figure 2.2: The comparison of the computational cost between the SKF, BOCPD, and GPCPD methods.

tion $p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta)$ for $i = 1, \ldots, n-1$ using the KF, and employ them for computing the joint distributions in Equations (2.12) and (2.13) in Step 3. This sequential approach iterates over different starting values of time index i by stitching different KFs together.

2.2.4 Computational Complexity

Let n denote the total number of observations. When there is no changepoint detected before, the SKF algorithm requires $\mathcal{O}(n)$ operations at time t_n by computing $n-1$ predictive distributions $p(y_n \mid \mathbf{y}_{i:(n-1)})$ for $i = 1, \ldots, n-1$, each taking $O(1)$ operation according to Theorem 1. When there is at least one changepoint detected before, by applying the truncation approach described in Section 2.2.1, we only need to compute predictive distributions $p(y_n \mid \mathbf{y}_{i:(n-1)})$ for $i = \hat{C}_{n-1}, \ldots, n-1$, which reduce the computational complexity to $\mathcal{O}(n')$, where $n' = n - \hat{C}_{n-1}$, and $\hat{C}_{n-1}$ denotes the most recently detected changepoint before the time t_{n-1}. When there is no changepoint detected before, we have $n' = n$.

In Figure 2.2, we compare the computational time for three distinct methods as the number of observations increases on a Windows 10 PC with two 3.00GHz i7-9700 CPUs. The computational cost of SKF is substantially smaller than GPCPD, as direct computation requires inversion of the correlation matrix. The fast computation enables us to deploy the scalable SKF algorithm for real-world scenarios with a large number of observations. On the other hand, the cost of SKF is similar to that of BOCPD, whereas the temporal correlation is modeled in SKF but not in BOCPD. As temporal correlation widely exists in real-world data sets, modeling the correlation can improve the accuracy of the changepoint detection. More detailed comparisons with other methods are provided in Section A.6 of the appendix.

2.3 Simulation Studies

This section compares different approaches for estimating single and multiple changepoints from temporally correlated data. We consider three types of changes: mean, variance, and correlation. For initial states before the changes, the data is sampled from a Gaussian process with mean $\mu = 0$, variance $\sigma^2 = 1$, and nugget parameter $\eta = \frac{\sigma_0^2}{\sigma^2} = 0.1$. We employ covariance functions from Equations (2.4) and (2.5) in simulations, setting range parameters at $\gamma = 12$ and $\gamma = 4$, respectively. The parameters μ, σ, and γ can vary under different change scenarios specified later. We compare the SKF approach with the BOCPD approach and CUSUM algorithm [100] summarized in Section A.7 of the appendix. For the SKF algorithm, the range and nugget parameters in the covariance matrix are estimated by maximizing the marginal likelihood function in Equation (2.11) for computing the predictive distributions. In Section A.8 of the appendix, we also compare different approaches for the scenarios when a covariance function is misspecified, and the conclusion is in line with the results herein.

2.3.1 Single Changepoint

We first compare the performance of different approaches for time sequences with a single changepoint. In this scenario, each method can report at most one changepoint during the whole detection period and thus only the first detected changepoint will be recorded. We apply two commonly used metrics for online changepoint detection algorithms [101, 102] to evaluate the performance of each method: Average Detection Delay (ADD) and Average Run Length (ARL). The ADD, defined as $\mathbb{E}_\tau\left[(\Gamma - \tau)^+\right]$, where the metric Γ represents the earliest time we detect a changepoint around the latent changepoint τ. ADD measures the average time lag between a changepoint occurrence and the time of its first detection, which may be compared with the power of a statistical test in hypothesis testing. A small value of the ADD indicates that the method is more powerful in detecting a latent changepoint. The ARL, defined as $\mathbb{E}_\infty\left[\Gamma\right]$, measures the average time of the first detected changepoint when there is no changepoint in the data, which can be interpreted as the type-I error in hypothesis testing. We evaluate SKF, BOCPD, and CUSUM using 100 random sampled time series, each with $n = 100$ observations. The observations are equally spaced in time, where the first $n_0 = 50$ observations serve as the training samples. To ensure a fair comparison, we let the ARL be approximately 50 across all methods, through specifying the hazard parameter value in BOCPD or SKF, and the threshold value for the CUSUM method, based on the time period with no changepoint.

Figure 2.3 shows that SKF consistently outperforms BOCPD and CUSUM in all tested scenarios, achieving the lowest ADD. In particular, when the change contains a small mean shift, a variance or correlation shift, both CUSUM and BOCPD have a large delay in detecting, whereas the SKF method has a relatively low detection delay for all scenarios. The SKF performs better than other approaches as it captures the

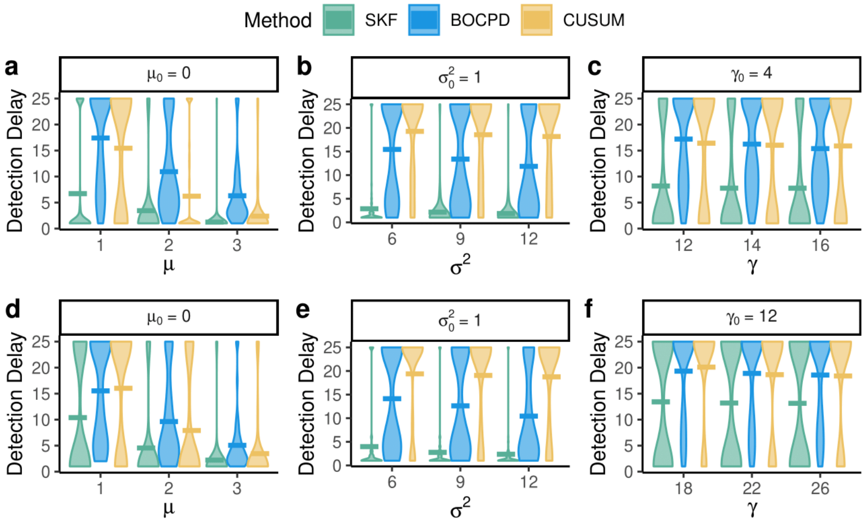


Figure 2.3: Violin plots comparing average detection delay for SKF, BOCPD, and CUSUM methods for 100 simulations. The upper and lower panels show the detection delay of each method when the data are simulated with the Matérn correlation with the roughness parameter being 2.5 and exponential correlation, respectively. A method with a low average detection delay is better. μ_0, σ_0^2, and γ_0 represent pre-change parameter values, while μ, σ^2 and γ on the x-axis stand for post-change parameter values.

temporal correlation from the observations. Furthermore, the computational complexity of the SKF is similar to BOCPD, which is crucial for real-world applications with a large number of samples.

Additionally, in Section A.8 of the appendix, we examine the SKF with a misspecified covariance function. Figure A.3 demonstrates that even with the misspecified covariance, the SKF performs comparably well to scenarios with the correct covariance, and still outperforms the BOCPD. This is because a method having a misspecified covariance with an estimated correlation length scale parameter is typically better than assuming the independence between observations to approximate the temporal covariance in the underlying data-generating process. This result reveals the robustness of the SKF for changepoint detection, even when certain configurations deviate from the true data-generating process.

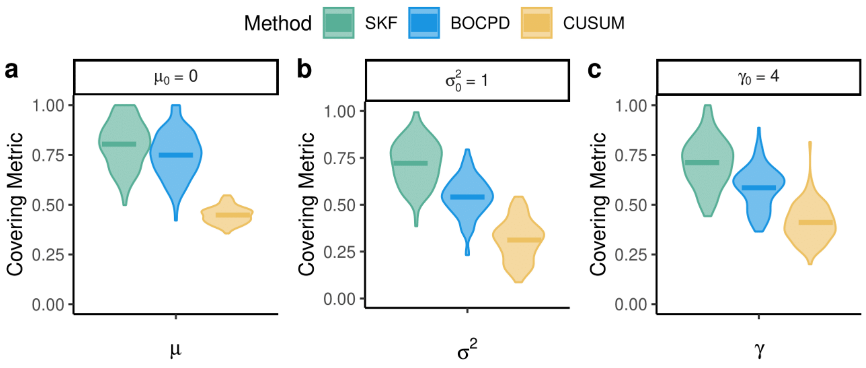


Figure 2.4: Violin plots of the covering metric (larger values are better) of the SKF, BOCPD, and CUSUM methods for simulated data with multiple changepoints.

2.3.2 Multiple changepoint detection

In this section, we assess the performance of different algorithms to detect multiple changepoints. We sample 100 time sequences, each containing $n = 150$ observations from a GP having the Matérn covariance in Equation (2.5) for demonstration purposes. Each time series have four changepoints at positions $\boldsymbol{\tau} = \{33, 66, 98, 130\}$. We investigate three scenarios where the changes occur in mean, variance, and covariance range parameters.

In multiple changepoints scenarios, metrics like ADD and ARL are not suitable as the number of detected and true changepoints may not be the same. Thus, we use the covering metric [80, 103] that measures how well the detected changepoints align with the true changepoints, defined in Section A.9 of the appendix. A method with a larger value of the covering metric is better.

Panels a-c in Figure 2.4 show the average covering metric for SKF, BOCPD, and CUSUM methods for the scenarios with the mean, variance, and correlation changes, respectively. Both BOCPD and SKF approaches outperform the CUSUM method in terms of the covering metric across all scenarios. This is because the CUSUM method relies on a prespecified threshold of the test statistics written as a cumulative summation of information to determine whether the current time point is a changepoint, which

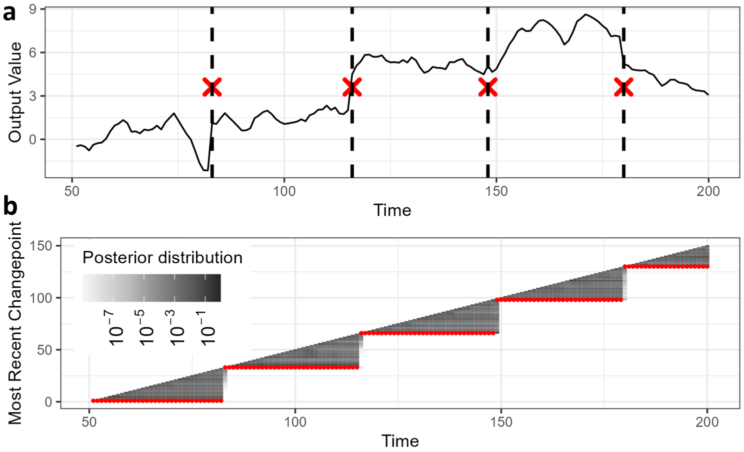


Figure 2.5: The black curves in panel a give the temporally correlated outcomes with 4 mean changes. The black dashed lines indicate the true changepoint locations and the red crosses give the estimated changepoints by SKF. Panel b shows the posterior distribution of the most recent changepoints at each time point, with MAP estimates graphed as red dots.

does not look back to find a changepoint in prior time points as BOCPD and SKF. In contrast, the predictive distributions in BOCPD and SKF contain information from a time period of previous subsequences, which enables the methods to detect a changepoint when information accumulates. Furthermore, the SKF method outperforms both BOCPD and CUSUM methods in terms of the covering metric, as the temporal correlations from the data are modeled in SKF, making SKF more accurate to approximate the data-generating mechanism.

Panel a in Figure 2.5 graphs the detected changepoints by SKF and the true changepoint for a simulated case with the mean shift. Panel b gives the classification probability of the most recent changepoint at each time point. The estimated most recent changepoints, marked by the red solid points, are the ones with the maximum posterior probability over all possible values. A new changepoint is identified if the most recently detected changepoint differs from the previously detected changepoint.

2.4 SARS-CoV-2 Detection among Dialysis Patients

2.4.1 Data Description

This study analyzes daily treatment data from over 150,000 dialysis patients collected by Fresenius Kidney Care between January 2020 and March 2022. Each patient visits the clinics about three times per week, producing a large data set with millions of observations. For each clinic visit, the data includes features such as sitting blood pressure, weight, temperature, respiration rate, pulse rate, oxygen level, interdialytic weight gain, average blood flow rate, and average dialysis flow rate. The dataset contains 15 million samples, with each patient owning around 94 samples on average, where only 0.4% of the observations are labeled as COVID-19 positive. We give an example of the mechanism of detection in Panel a Figure 2.6, where the COVID-19 positive window of a patient contains a three-day incubation period [104, 105] and a seven-day infection period post symptom onset [106]. A clinic visit is labeled as COVID-19 positive if it is within two days prior to, or seven days following day 0, which is the day for a positive COVID-19 PCR test. We conducted the sensitivity analysis with different choices of COVID-19 positive period in Section A.11 of the appendix and the results remain similar.

2.4.2 Experimental Setup and Results

We focus on detecting COVID-19 infection for a large number of dialysis patients in this section. Data-driven models have been extensively used for detecting COVID-19 infection dates in patient-level longitudinal data [30, 32, 33]. Most of these models generate prediction probabilities of COVID-19 infections, whereas a threshold is typically used to identify infections, and the temporal pattern among the longitudinal data was not modeled in these approaches. For example, some COVID-19 positive patients have

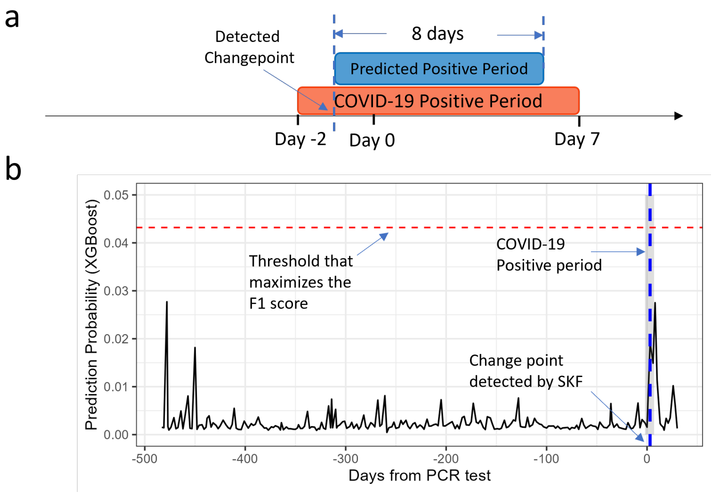


Figure 2.6: In panel a, the orange area is the COVID-19 positive period spanning from day -2 to day 7, where a patient has a positive COVID-19 PCR test at day 0. If a patient is detected to be COVID-19 positive, the detected changepoint and the subsequent seven days are marked as the predicted infection period, shown as the blue area. The black curve in panel b shows the probability sequence of being COVID-19 positive estimated by XGBoost, and a value larger than the threshold value shown as red dashed line is classified as COVID-19 positive. The blue dashed line marks the changepoint detected by SKF, and the grey area represents the COVID-19 positive period from day -2 to day 7.

only mild to moderate symptoms, which may result in an increasing trend of the prediction probabilities, but the prediction probabilities might still fall below the pre-specified threshold. Ignoring such a trend can lead to low sensitivity in detection. We will apply changepoint detection methods to probability sequences of COVID-19 infections to improve the detection performance.

We employed an integrated procedure to detect the changepoint from the COVID-19 infection summarized in Algorithm 4 of the appendix. We first apply a data-driven classification model to patients' clinical data, here chosen as the XGBoost method [36], which was previously found to be accurate in detecting COVID-19 among dialysis patients [30, 87] compared to a few other classification methods. Second, we apply SKF to detect the change in the daily prediction probabilities of COVID-19 infection from the XGBoost approach. Furthermore, we developed an additional screening step to detect the onset of an increasing subsequence in infection probabilities through a hypothesis test (Step 5 in Algorithm 4), as typically the increase of the probability sequences of infection should be detected. Once the detected changepoint passes this screening step, we mark the seven-day period after the detected changepoint as COVID-19 positive [106]. The integrated approach is generally applicable to detect changes from longitudinal data. Details of the integrated approach can be found in Section A.10 of the appendix. In both BOCPD and SKF methods, the hazard function was defined in proportion to the county-level daily probability of contracting COVID-19 [107], enhancing detection accuracy by the estimated daily transmission probability based on local infection and death counts.

A typical COVID-19 positive patient will have two changes during the infection period, characterized by an increasing trend at the beginning and a decreasing trend at the end of the infection probability. We aim to detect the first change, and such a detection scheme is useful for the onset of other diseases based on longitudinal data. This means the covering metric in Section 2.3.2 is not sensible. To better evaluate the effectiveness of detecting the infection, we use precision, recall, F1-score, and detection delay as our performance metrics, which are defined below. The threshold value for the classification probabilities from the XGBoost method is determined by maximizing the F1-score across all patients in the test data, which is defined as the harmonic mean of precision and recall: F1-score $= 2 \times \frac{\text{precision}\times\text{recall}}{(\text{precision}+\text{recall})}$, where precision $= \frac{\text{TP}}{\text{TP}+\text{FP}}$ is the ratio of the true positives out of all the positive predictions, and recall $= \frac{\text{TP}}{\text{TP}+\text{FN}}$ is ratios of true positive out of all the positively labeled samples, which quantifies the power of the algorithm. The true positives (TP), false positives (FP), and false negatives (FN) are defined as TP $= \sum_{j=1}^{n^*} I_{\{\hat{x}_j=1,x_j=1\}}$, FP $= \sum_{j=1}^{n^*} I_{\{\hat{x}_j=1,x_i=0\}}$, and FN $= \sum_{j=1}^{n^*} I_{\{\hat{x}_j=0,x_j=1\}}$, where $n^* = n - n_0$ denotes the number of samples in the test data. $I_{\{\cdot\}}$ is the indicator function, x_j are the actual COVID-19 labels and $\hat{x}_j$ is the predictive COVID-19 labels. For statistical learning models like XGBoost, the predicted label $\hat{x}_j$ is assigned a value of 1 if t_j is within a seven-day window following the date when the predicted probability exceeds the threshold value, and is set to 0 otherwise. For changepoint detection algorithms such as SKF and BOCPD, $\hat{x}_j$ is assigned the value of 1 if t_j falls within a seven-day period following a detected changepoint. The average detection delay is calculated as the average number of days from the start date of the COVID-19 positive period to the time a changepoint within the positive period is first detected. A lower average detection delay reflects a quicker response to the onset of changes. Furthermore, any detection made after 2 weeks of day 0 is considered as not useful in the online detection, which is not counted as a true positive, as they are too late to help. This threshold can be adjusted for detecting other diseases.

Table 2.1 compares the performance of different approaches for detecting COVID-19 infection. Here the baseline positive data only constitutes around 0.4% of the total observations. This setting differs from a few other COVID-19 detection schemes where each COVID-19 positive record is matched with a few negative samples [30, 87]. The practical

Table 2.1: Out-of-sample comparisons of the classification methods and online changepoint detection methods, including CUSUM, BOCPD, and SKF on COVID-19 detection with the baseline positive rate of 0.4%.

	Precision	Recall	F1-score	Detection Delay
Logistic regression	0.055	0.141	0.079	1.56
Random forests	0.083	0.123	0.099	2.15
XGBoost	0.082	0.179	0.113	2.22
CUSUM	0.028	0.023	0.025	3.73
BOCPD	0.174	0.168	0.171	5.3
SKF	0.218	0.179	0.197	4.13
SKF with screening	0.232	0.190	0.209	2.82

scheme is closer to the longitudinal detection scheme employed herein. We first found that changepoint detection algorithms, including BOCPD and SKF, perform better than the classification methods, such as logistic regression, random forecast, and XGBoost. This is because the change detection utilizes longitudinal information for identifying the change for each patient, while the classification methods rely on a unified threshold of probability sequences of being infected for all patients. Among the changepoint detection methods, the CUSUM algorithm is not as good as BOCPD and SKF. The good performance of SKF and BOCPD is largely due to their ability to recursively inspect whether each of the previous time points is a changepoint, in contrast to the CUSUM method which can only determine whether the current time point is a changepoint. Second, SKF outperforms BOCPD in both F1-score and detection delay. This advantage is largely attributed to SKF's ability to model temporal correlations, which helps reduce false detection. Notably, SKF can detect the infection about one day faster than BOCPD on average, which means the SKF requires less information to identify a COVID-19 infection. Furthermore, incorporating the screening method, as detailed in Steps 6 and 7 of Algorithm 4 in the appendix, further enhances the F-1 score and reduces detection delay in SKF. This improvement aligns with our expectations, since the screening method chooses changepoints related to an increasing subsequence, thereby improving

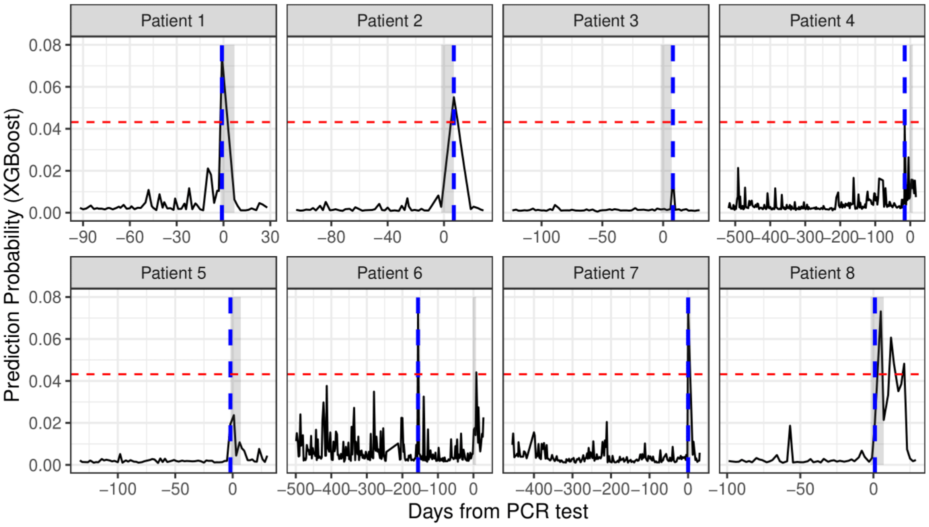


Figure 2.7: Results of SKF on the probability sequences of 8 COVID-19 patients. The red dashed line is the threshold value that maximizes the F1-score. The blue dashed line shows the changepoint detected by SKF. The grey area is the period that a patient is labeled as COVID-19-positive. Day 0 is the positive PCR test day.

the precision of COVID-19 detection.

Panel b in Figure 2.6 gives an example comparing the SKF detection with the XGBoost method. The probability sequences of COVID-19 infection from XGBoost, shown as the black curve, consistently remain beneath the threshold value, indicated by the red dashed line, suggesting that this patient is predicted as COVID-19 negative by XGBoost for the entire period. The increasing trend of probability sequence during the infection period, however, allows the SKF to successfully detect the changepoint from COVID-19 infection, indicated by the blue dashed line. Figure 2.7 gives further comparison between the SKF and XGBoost for a group of 8 randomly selected COVID-19 positive patients. Here we only show the plots of positive patients and there are around 74% of the patients who do not have a positive PCR test during the whole period. The SKF method successfully identifies the COVID-19 positive period for 6 of these patients and misses the COVID-19 infection for 2 patients. In comparison, the XGBoost method correctly identifies the COVID-19 positive period for only 4 patients and misses the infection for

4 patients. SKF may be preferred for this problem over a conventional classification approach as it can identify the probability subsequences with an increasing trend. Furthermore, the empirical autocorrelation of probability sequences of a few randomly selected patients is plotted in Section A.1 of the appendix. The autocorrelation of the probability sequences from the longitudinal data is modeled in SKF, which improves the detection accuracy. More numerical comparisons of online changepoint detection approaches for a few other real-world examples are provided in Section A.12 of the appendix, which confirms competitive performance by the SKF approach.

Chapter 3

Robust estimation of SARS-CoV-2 epidemic in US counties

In Chapter 2, we focus on patient-level detection of COVID-19 infection timing using patients' daily biomarker data. The proposed detection algorithm incorporates spatial information to describe the severity of COVID-19 transmission in specific regions as the COVID-19 outbreak is asynchronous in US counties. However, the direct integration of county-level COVID-19 confirmed case data into the detection algorithm could introduce bias, as these figures often underestimate actual case numbers and can be unstable in counties with small population sizes. Therefore, in Chapter 3, we propose a robust and efficient approach to monitor the heterogeneous progression of SARS-CoV-2 in all US counties having no less than 2 COVID-19 associated deaths and estimate the daily probability of contracting (PoC) SARS-CoV-2 for each county. The PoC SARS-CoV-2, representing the average daily probability that a susceptible individual in a specific county will contract SARS-CoV-2, quantifies the community-level transmission risk and is integrated into the detection algorithm in Chapter 2. Furthermore, based on the approach proposed in this chapter, we found that shortening by 5% of the infectious period of SARS-CoV-2 can reduce around 39% (or 78K, 95% CI: [66K ,89K]) of the COVID-19 associated deaths in the US as of 20 September 2020. Our findings also indicate that reducing infection and deaths by a shortened infectious period is more pronounced for areas with an effective reproduction number close to 1, suggesting that testing should be used along with other mitigation measures, such as social distancing and facial mask-wearing, to reduce the transmission rate. Our deliverable includes a dynamic county-level map for local officials to determine optimal policy responses and for the public to better understand the risk of contracting SARS-CoV-2 on each day.

3.1 Introduction

The outbreak of the new coronavirus 2019 (COVID-19) has caused nearly 200,000 deaths in the US, and among those, there are 2,277 counties with no less than 2 associated deaths as of 20 September 2020 [108]. The ongoing COVID-19 pandemic has led to unprecedented non-pharmaceutical interventions (NPIs), including travel restrictions, lockdowns, social distancing, facial masks wearing, and quarantine to reduce the spread of SARS-CoV-2 in the US. The COVID-19 outbreak is prolonged and asynchronous across regions. Thus it is critical to estimate the dynamics of COVID-19 epidemic to determine appropriate protective measures before the availability of effective vaccines.

A non-negligible proportion of SARS-CoV-2 infectious individuals is asymptomatic or have mild symptoms [109]. We term the individuals the *active infectious individuals* who can transmit the disease to others but may not be diagnosed yet. Identifying the number of active infectious individuals is crucial to monitor the transmission in a community. Another important time-dependent quantity is the expected number of secondary cases resulted from each active infectious individual, or *effective reproduction number*. In this article, we estimate these two time-dependent quantities for all US counties with no less than 2 COVID-19 associated deaths as of 20 September 2020; the population of some counties that falls within this category is even less than ten thousand. Furthermore, based on these two time-dependent quantities, a more interpretable measure, called the daily *probability of contracting* (PoC) SARS-CoV-2 for an individual at the county-level was used to quantify the risk. This static risk factor with fixed transmission rates was studied before [110]. Here we studied the dynamic transmission rate parameter, which is estimated by the number of deaths, test positive rates and the number of confirmed cases in a community. The risk factor can be extended to measure the risk of an event with different sizes [111]. The fine-grain estimation of disease progression characteristics allows the public to understand the risk of contracting COVID-19 on a daily basis.

Predictive mathematical models are useful for analyzing an epidemic to guide policy responses [112]. The epidemiology compartmental models such as SIR, SEIR, SIRD, and their extensions [40, 41, 42, 43, 44], stochastic agent based models [45, 46], branching processes [47], and network analysis [48] have advanced our understanding of transmission rates and incubation period of SARS-CoV-2, which are connected to the traffic flow and mobility during the COVID-19 outbreaks at different regions [76, 77]. The disease progression characteristics, such as the transmission rate, are often estimated based on the daily death toll [40, 43, 45, 46]. However, it is challenging to estimate the progression of the epidemic in US counties with small population, because the number of daily observed confirmed cases and COVID-19-related deaths is small. Meanwhile, using observed laboratory-confirmed COVID-19 cases (henceforth, observed confirmed cases) might significantly underestimate the population that have been infected with the SARS-CoV-2. It was found in [113] that around 9.3% of the US individuals (or roughly 30 million) may have contracted the COVID by July 2020 based on serology tests, whereas less than 4.8 million COVID-19 positive cases have been confirmed in the US before August 2020 [108]. Thus, it is important to estimate the number of individuals who contracted COVID-19 but had not tested positive. The focus herein is on integrating COVID-19-related death toll and test data to obtain a robust estimation of the disease progression characteristics of COVID-19 at county and community levels.

One critical quantity to evaluate an infectious disease outbreak is the time-dependent transmission rate, based on which one can compute the basic reproduction number and the effective reproduction number of the disease. Various approaches were proposed to estimate this parameter. The transmission rate was modeled as a decreasing function of the time in [40], a function of NPIs in [45] and a geometric Brownian motion in [114]. Unlike the outbreak in China or other countries in north-east Asia, transmission rates of the COVID-19 progression in the US does not monotonically decrease due to the prolonged duration of the outbreak, and it is challenging to determine a suitable parametric form of this parameter in terms of time. In [43], the transmission rate parameter was related to the initial values of infectious cases, resolving cases, and up to two derivatives of the daily death toll. This method provides a flexible way to estimate the time-dependent transmission rate from the death toll and its derivatives, yet unstable for counties with moderate or small population sizes, as numerical estimation of the daily death toll and its derivatives is often unstable.

In this work, we propose a robust approach of integrating test data and death toll to estimate COVID-19 transmission characteristics by a Susceptible, Infectious, Resolving (but not infectious), Deceased, and reCovered (SIRDC) model initially studied in [43]. We illustrate that the transition between different stages of disease progression in the SIRDC model in part a of Figure 3.1. First, a part of the population is infected by active infectious individuals each day, depending on the transmission rate parameter (β_t). After γ^{-1} days, an active infectious individual is expected to be no longer infectious, denoted by the resolving compartment, meaning that this individual will not transmit COVID-19 to others as a result of hospitalization or self-quarantine. We term the average length

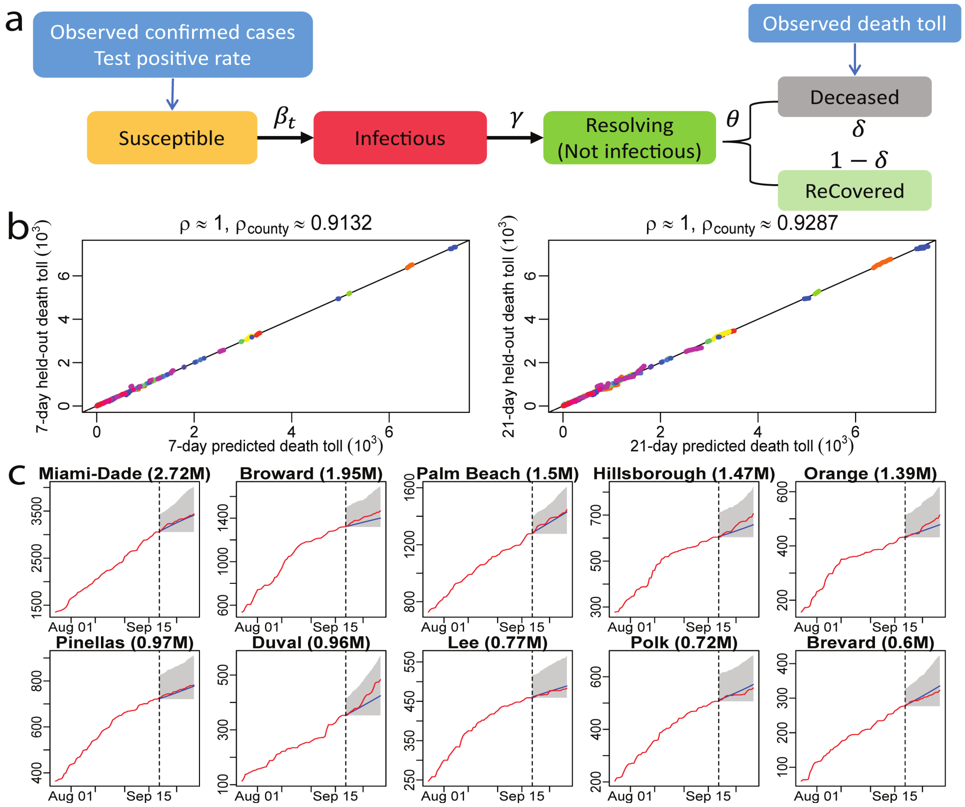
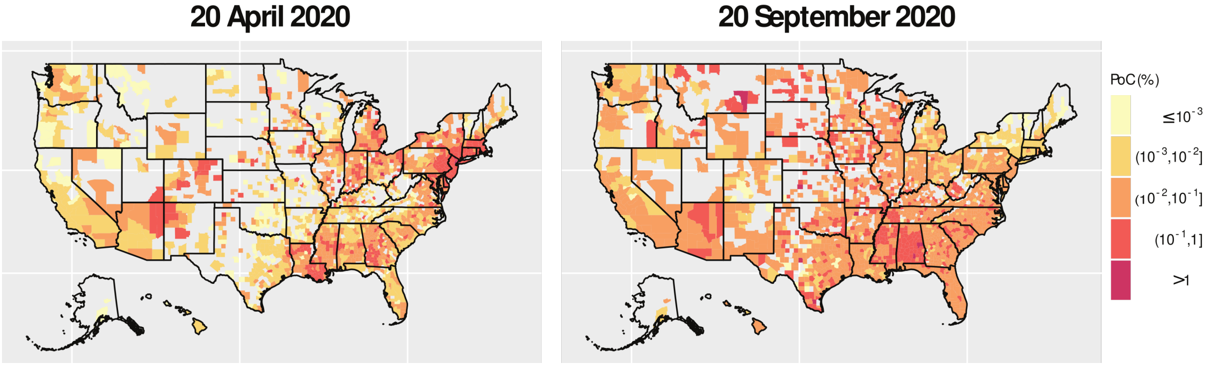


Figure 3.1: a, The SIRDC model and the data used for analysis. b, 7-day death toll forecast and 21-day death toll forecast against the held-out truth in 2,277 US counties with no less than 2 deaths as of 20 September 2020. Each dot is a cumulative death toll for one county at one held-out day. Counties from the same state are graphed using the same color. The Pearson correlation coefficient (ρ) of the nation and the weighted average of Pearson correlation coefficient for counties (ρ_{county}) are recorded. c, 21-day death toll forecasts in 10 counties with largest population in Florida, where the red line represents the observed death toll and blue line means the forecast. The forecast starts from 21 September 2020, marked by the vertical black dash line. The grey shadow area is the 95% confidence interval of the forecast. Numbers in the parentheses right after the county name are population in million.

Table 3.1: Policy summary

Background	The transmission of SARS-CoV-2 is heterogeneous and asynchronous in US counties. It is thus important to assess the risk before lifting or replacing any mitigation measure in the community. We have developed a novel approach to integrate test data and death toll to estimate the probability of contracting COVID-19, as well as the time-dependent transmission rate and number of active infectious individuals at the county level in the US.
Main findings and limitations	National level order of protective measures reduces the transmission rate and active number of infectious individuals for most US counties in April, whereas the risk of contracting SARS-CoV-2 rebounded between late June and early July, as the protective measures were relaxed. We found that when the infectious period of SARS-CoV-2 is shortened by 5% and 10%, the number of deaths can be reduced from 199K to 120K (95% CI: [109K, 132K]) and 80K (95% CI: [72K, 89K]) as of 20 September 2020, respectively, when other protective measures were kept the same. The reduction of the infectious period can be achieved by extra testing in addition to ongoing protective measures. Our model relies on the existing knowledge of the COVID-19 and model assumptions. Other information, such as demographic profiles, mobility, and serology test data, can be used to calibrate the model parameters and assumptions at the community level.
Policy implications	Our model indicates that extra testings, along with the current NPIs, can significantly reduce the number of deaths associated with COVID-19. The estimated probability of contracting COVID-19 can be used as an interpretable risk factor to guide community policy responses.


20 April 2020

20 September 2020

Figure 3.2: a, The estimated probability of contracting SARS-CoV-2 at 1,856 counties on 2020-04-20, and b, at 2,277 counties on 20 September 2020. The probability of contracting SARS-CoV-2 is truncated at 10^{-6}, whereas only 78 counties on 20 April and 45 counties on 20 September are below this level, respectively.

of an active infectious individual the *infectious period*. A resolving case is expected to be resolved (either recovered or deceased) after θ^{-1} days. The proportion of deaths from the number of resolved cases is controlled by the fatality rate parameter δ.

Our approach has three innovations. First, we solve the compartmental models using a midpoint rule with a step size of 1 day, as the confirmed cases and death toll are updated daily in most US counties, and this is discussed in the method section. Second, we combine test positive rates, confirmed cases and death toll to estimate the daily transmission rate parameter. Our estimate of transmission rates and reproduction numbers is robust and accurate to reproduce the number of the death toll and other compartments for counties with medium to small population sizes (Figure 3.3 and 3.4). The simulated studies in Figure 3.5 also suggest that our approach is more robust than the solution in [43], as our solution does not require estimating derivatives of the daily death toll. Only two parameters, the initial values of the number of active infectious individuals and the number of resolving cases, need to be estimated numerically for each county. Then we can solve the time-dependent transmission rates and all other compartments subsequently. Since only two parameters are estimated for each county, our estimation rarely depends on the initial values we choose for the optimization. Finally, we use a Gaussian process

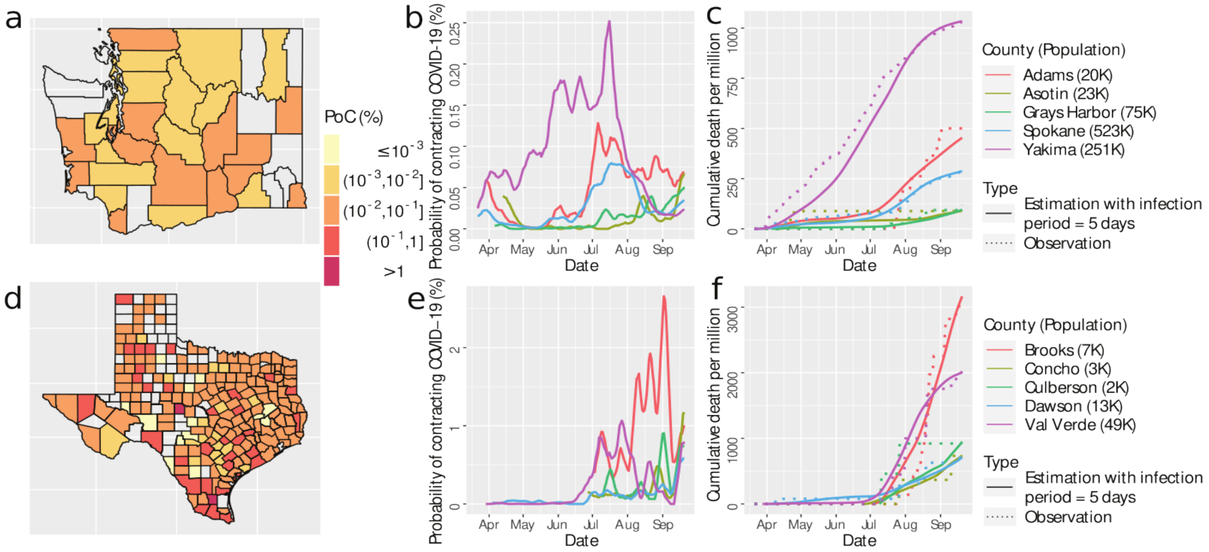


Figure 3.3: a, The estimated probability of contracting SARS-CoV-2 in Washington state on 20 September 2020. b, the probability of contracting SARS-CoV-2 from 5 counties in Washington state with the largest PoC SARS-CoV-2 values on 20 September 2020 . c, the observed (dots) and fitted (solid line) cumulative death toll in the 5 counties in figure b from the same time period. d-f, The results in Texas that have the same interpretation as a-c. Part e and f have different scales than part b and c, respectively.

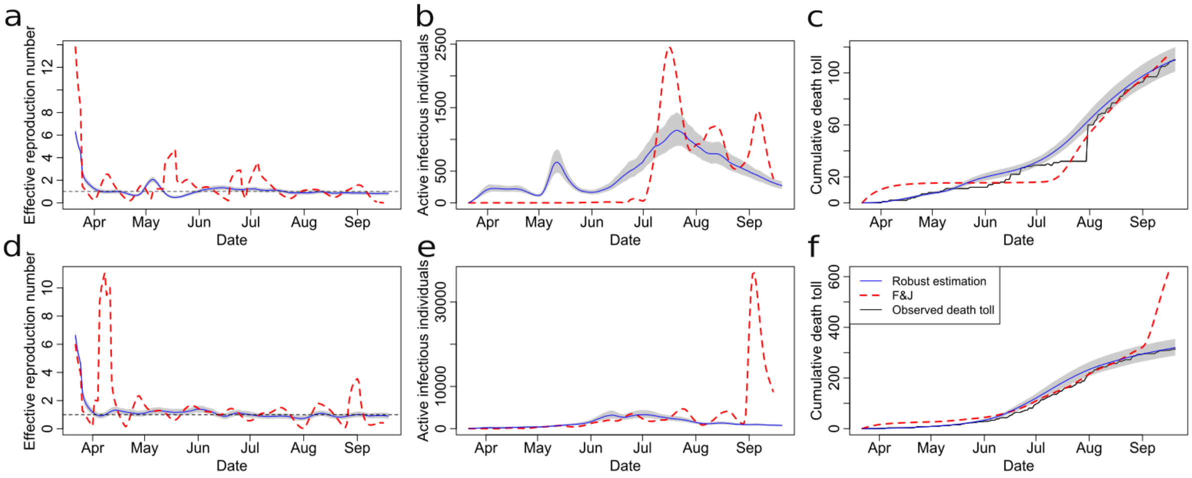


Figure 3.4: a-c, Comparisons between the estimation COVID-19 progression characteristics for Santa Barbara, CA as of 20 September 2020 by our algorithm 1 (blue solid curves) and the method $F\&J$ [43] (red dash curves) . The shaded area represents 95% confidence intervals. The black solid curve in part c is the observed cumulative death toll in Santa Barbara. d-f, Results for Imperial, CA as of 20 September 2020, which have the same interpretation as a-c. The transmission rate estimated from the method $F\&J$ is truncated to be within [0,10].

to model the residual between the observed death toll and that from the SIRDC model, leading to more accurate predictions and proper uncertainty quantification. A summary of the main findings, limitations, and policy implications are given in Table 3.1.

3.2 Data and Methods

In this section, we introduce the data and methods for this study. The main symbols used in this section and their definitions are provided in Table 3.2.

Data. For day t at the jth county in the ith state of the U.S., we utilize three datasets [108, 115] in this study: (1) $c^o_{i,j}(t)$, representing the county-level daily cumulative observed confirmed COVID-19 cases; (2) $d^o_{i,j}(t)$, denoting the county-level daily cumulative COVID-19 death toll; and (3) $p_i(t)$, which indicates the state-level daily COVID-19 test positive rate, where $i = 1, 2, \dots, k$, $j = 1, 2, \dots, n_i$ with n_i being the

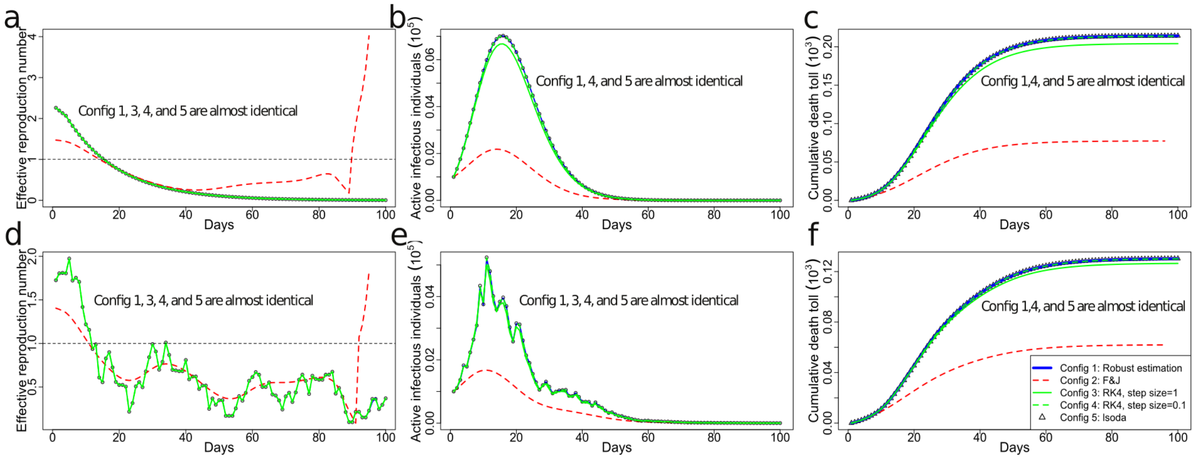


Figure 3.5: a-c, Simulated comparison with noise-free observations. The black circles are the solution of the ODEs of the SIRDC model via the default numerical solver lsoda in the function `ode` in `deSolve R` package. The green solid and dash curves are the numerical solutions from Runge–Kutta method with the 4th order integration and step size being 1 and 0.1, respectively. The Blue solid curves are the robust estimation from algorithm 1 and red dash curves are the estimation in [43]. In the simulation with noise-free observations, we let time duration be $T = 100$ days, the population size $N = 10^7$, the initial values of 5 compartments chosen as $(S(1), I(1), R(1), D(1), C(1)) = (N - 2000, 1000, 1000, 0, 0)$ and the transmission rate $\beta(t) = \exp\left(-0.7(\frac{9}{T-1}(t-1)+1)\right)$, for $1 \leq t \leq T$. d-f, results of the simulation with noisy observations, which have the same interpretation as a-c. In this simulation, we set the transmission rate $\beta(t) = \exp\left(-0.7(\frac{9}{T-1}(t-1)+1)\right) + \epsilon$, for $1 \leq t \leq T$ and $\epsilon \sim N(0, 0.04)$, and the other parameters are held the same as in the noise-free simulation. The transmission rates estimated from the method $F\&J$ are truncated to be within [0,10]. The solution from our robust estimation approach, the lsoda and the Runge–Kutta method with the 4th order and step size being 0.1 overlap for both scenarios.

Table 3.2: Main symbols and definitions in the Methods Section.

Symbol	**Definition**
$S(t)$	number of susceptible cases on day t
$I(t)$	number of infectious cases which can transmit COVID-19 on day t
$R(t)$	number of resolved cases which get infected but cannot transmit COVID-19 on day t
$D(t)$	number of deceased cases on day t
$M(t)$	number of recovered cases on day t
N	number of population in a given area
$\beta(t)$	transmission rate on day t
γ^{-1}	average number of days an individual can transmit COVID-19
θ^{-1}	average number of days for a case to get resolved
δ	proportion of deceased cases, a.k.a. fatality rate
$R_0(t)$	basic reproduction number on day t
$R_{eff}(t)$	effective reproduction number on day t
$P(t)$	average probability of contracting (PoC) SARS-CoV-2 on day t
$p(t)$	state-level test positive rate on day t
$d^o(t)$	cumulative number of observed death toll on day t
$c^o(t)$	cumulative number of observed confirmed cases on day t
$\Delta c^o(t)$	daily number of observed confirmed cases on day t
$c^u(t)$	cumulative number of unobserved confirmed cases on day t
α	power parameter for estimating the number of susceptible cases
ω	weight parameter for estimating the number of susceptible cases
z	zero-mean Gaussian process

number of counties of the ith state considered in the analysis, and $t = 1, \ldots, T_{i,j}$.

SIRDC compartmental models. The SIRDC model for the jth county in the ith state in the US is described below:

$$\begin{aligned}
\dot{S}_{i,j}(t) &= \frac{-\beta_{i,j}(t)S_{i,j}(t)I_{i,j}(t)}{N_{i,j}}, \\
\dot{I}_{i,j}(t) &= \frac{\beta_{i,j}(t)S_{i,j}(t)I_{i,j}(t)}{N_{i,j}} - \gamma I_{i,j}(t), \\
\dot{R}_{i,j}(t) &= \gamma I_{i,j}(t) - \theta R_{i,j}(t), \\
\dot{D}_{i,j}(t) &= \delta\theta R_{i,j}(t), \\
\dot{M}_{i,j}(t) &= (1-\delta)\theta R_{i,j}(t),
\end{aligned} \tag{3.1}$$

where $S_{i,j}(t)$, $I_{i,j}(t)$, $R_{i,j}(t)$, $D_{i,j}(t)$ and $M_{i,j}(t)$ denote the number of individuals at these 5 compartmental groups on day t, respectively, and $N_{i,j}$ denotes the number of individuals in county j from state i. The time-dependent transmission rate parameter is denoted by

$\beta_{i,j}(t)$ and the inverse of average number of days an infectious individual can transmit the COVID-19 is denoted by γ. The inverse of the average number of dates for a case to get resolved (i.e. deceased or recovered) is denoted by θ and the proportion of deceased cases (i.e. death rate) is denoted by δ. The parameters (γ, θ, δ) were invariant over time and held fixed in this study. Following [116], we assume the infectious period to be 5 days on average, and a case is expected to resolve after 10 days. The average death rate is assumed to be 0.66% [117]. Additional verification of these assumptions and sensitivity analysis of these parameters are provided in the appendix.

To determine the characteristics of the SARS-CoV-2 epidemic at US counties, we define the time-dependent *effective reproduction number*, i.e. the average number of secondary cases per primary cases as $\mathcal{R}_{eff}^{i,j}(t) = \mathcal{R}_0^{i,j}(t)S_{i,j}(t)/N_{i,j}$, where the $\mathcal{R}_0^{i,j}(t) = \beta_{i,j}(t)/\gamma$ denotes the *basic reproduction number* on day t. When $\mathcal{R}_{eff}^{i,j}(t) < 1$, it means that the number of the active infectious individuals will decrease (and vice versa, if $\mathcal{R}_{eff}^{i,j}(t) > 1$). The effective reproduction number was often used to quantify whether or not the disease is under control [118]. However, the effective reproduction number does not directly quantify risk of contracting SARS-COV-2 for a susceptible individual, as the number of active infectious individuals in a region was not taken into consideration. We compute the average probability of contracting (PoC) SARS-CoV-2, denoted as $P_{i,j}(t) = \mathcal{R}_{eff}^{i,j}(t)I_{i,j}(t)\gamma/(S_{i,j}(t)) = \beta_{i,j}(t)I_{i,j}(t)/N_{i,j}$, which quantifies the risk of a susceptible individual in county j from state i to catch SARS-CoV-2 on day t. Here the risk is on an average sense among all susceptible individuals in a region.

The most critical parameter of the SIRDC model is the transmission rate parameter, $\beta_{i,j}(t)$, as a function of time, based on which we obtain the reproduction number on day t. To estimate the time-dependent transmission rates for communities with small population sizes, we derive a more robust estimation of the transmission rate of each county based on the death toll and testing data, discussed below.

Closed-form expressions of the time-dependent transmission rates. Since the observations such as death toll and confirmed cases are generally updated daily, we solve the ordinary differential equations (ODEs) in the SIRDC model (Equation (3.1)) approximately by the midpoint rule of the integral with a step size of 1 day. For day $t \in \mathbb{N}^+$, the approximation is described below:

$$\frac{S_{i,j}(t+1)}{S_{i,j}(t)} \doteq \exp\left\{-\frac{\beta_{i,j}(t+0.5)}{2N_{i,j}}\left(I_{i,j}(t)+I_{i,j}(t+1)\right)\right\}, \tag{3.2}$$

$$\frac{I_{i,j}(t+1)}{I_{i,j}(t)} \doteq \exp\left\{\frac{\beta_{i,j}(t+0.5)}{2N_{i,j}}(S_{i,j}(t)+S_{i,j}(t+1))-\gamma\right\}, \tag{3.3}$$

$$R_{i,j}(t+1)-R_{i,j}(t) \doteq \gamma\frac{I_{i,j}(t)+I_{i,j}(t+1)}{2}-\theta\frac{R_{i,j}(t)+R_{i,j}(t+1)}{2}, \tag{3.4}$$

$$D_{i,j}(t+1)-D_{i,j}(t) \doteq \delta\theta\frac{R_{i,j}(t)+R_{i,j}(t+1)}{2}, \tag{3.5}$$

$$M_{i,j}(t+1)-M_{i,j}(t) \doteq (1-\delta)\theta\frac{R_{i,j}(t)+R_{i,j}(t+1)}{2}. \tag{3.6}$$

Further by assuming the transmission rate parameter $\beta_{i,j}(t)$ is day-to-day invariant (i.e. a step function with step size 1), based on Equations (3.2) and (3.3), we obtain $\beta_{i,j}(t+0.5)$ from $t = 1$ to $T_{i,j}-1$, iteratively, based on the sequence of susceptible individuals $\{S_{i,j}(t)\}_{t=1}^{T_{i,j}}$ and the initial number of active infectious individuals $I_{i,j}(1)$ described in algorithm 2.

Algorithm 2 Iterative approach for estimating transmission rate $\beta_{i,j}(t+0.5)$.

```
Require: ${S_{i,j}(t)}_{t=1}^{T_{i,j}}$, $I_{i,j}(1)$
Ensure: ${\beta_{i,j}(t+0.5)}_{t=1}^{T_{i,j}-1}$, ${I_{i,j}(t)}_{t=1}^{T_{i,j}}$ $S_1 = S_{i,j}(1)$  $S_2 = S_{i,j}(2)$  $I_1 = I_{i,j}(1)$
    for $t = 1$ to $(T_{i,j} - 1)$ do
        $\beta_{i,j}(t+0.5) = \left\{\beta : \frac{S_2}{S_1} - \exp\left\{-\frac{\beta I_1}{2N_{i,j}}(1+\exp\{\frac{\beta}{2N_{i,j}}(S_1+S_2)-\gamma\})\right\} = 0\right\}$
        $I_{i,j}(t+1) = I_1 \exp\left\{\frac{\beta_{i,j}(t+0.5)}{2N_{i,j}}(S_1+S_2)-\gamma\right\}$
        $S_1 = S_2$
        $S_2 = S_{i,j}(t+1)$
        $I_1 = I_{i,j}(t+1)$
    end for
```

After we get the number of active infective individuals ($I_{i,j}(t)$) on each day, sequences of the resolving, deceased and recovered compartments can be solved subsequently following the same manner using Equations (3.4)-(3.6), after specifying their initial values. Expressing the time-dependent transmission rate by the number of susceptive and infective cases is the key to integrating death toll and testing data for estimation.

In Extended Data Figure 1 and 2, we demonstrate that in order to solve the ODEs in the SIRDC model, our approach is more accurate and robust than the method $F\&J$ in [43] under both simulated and real scenarios. Other more accurate methods (such as the Runge-Kutta method) can also solve the ODEs of SIRDC model, but the time-dependent transmission rates can not easily be expressed as a function of the death toll and the number of active infectious individuals as the way they are in our solution.

Estimation of the number of susceptible individuals. Note that we have $S_{i,j}(t) + c^o_{i,j}(t) + c^u_{i,j}(t) = N_{i,j}$ for any t, where $c^o_{i,j}(t)$ and $c^u_{i,j}(t)$ are the number of cumulative observed confirmed cases and unobserved confirmed cases, respectively. Estimating the number of susceptible individuals is equivalent to estimating the number of unobserved confirmed cases $c^u_{i,j}(t)$, because the number of observed confirmed cases $c^o_{i,j}(t)$ and the population $N_{i,j}$ are known. Here we combine them with the positive test rates to estimate $c^u_{i,j}(t)$, as large positive test rates typically indicate a large number of unobserved confirmed cases. We assume that the total number of confirmed cases is equal to the observed confirmed cases, adjusted by the state-level test positive rate $p_i(t)$, a power parameter α_i and a weight parameter ω_i, leading to the following formula of the susceptible population:

$$S_{i,j}(t) = N_{i,j} - c^o_{i,j}(t) - c^u_{i,j}(t) = N_{i,j} - \frac{1}{\omega_{i,j}} \left\{ \mathbb{1}_{\{t\geq 2\}} \sum_{s=2}^{t} (p_i(s))^{\alpha_i} \Delta c^o_{i,j}(s) + (p_i(1))^{\alpha_i} c^o_{i,j}(1) \right\}, \tag{3.7}$$

where $\Delta c^o_{i,j}(t)$ is the observed daily confirmed cases on day t, for $t = 1, 2, ..., T_{i,j}$, $i = 1, 2, ..., k$ and $j = 1, 2, ..., n_i$. Since the positive test rates are only available at the state level, the power parameter $\alpha_i \in [0, 2]$ is estimated by the state-level observations. According to Equation (3.7), the time-invariant weight $\omega_{i,j}$ can be expressed below:

$$\omega_{i,j} = \frac{(p_i(1))^{\alpha_i} c^o_{i,j}(1)}{I_{i,j}(1) + R_{i,j}(1) + D_{i,j}(1) + M_{i,j}(1)}, \tag{3.8}$$

where $I_{i,j}(1)$, $R_{i,j}(1)$, $D_{i,j}(1)$ and $M_{i,j}(1)$ are the number of active infectious, resolving, deceased and recovered cases on day 1, respectively.

Estimation of initial values of infectious and resolving cases. We define day 1 of a county as the more recent date between 21 March 2020 and the date that the county has 5 observed confirmed cases for the first time. Since all counties were at an early stage of the epidemic on the starting day, we let the initial value of the death toll $D_{i,j}(1)$ be the observed death toll on the day 1, and the initial value of the recovered cases be 0. This assumption is not likely going to strongly influence our analysis, as the number of recovered cases is only a negligible proportion of the susceptible individual on the starting day if not zero. The only parameters to estimate are the number of infectious individuals $I_{i,j}(1)$ and the number of resolving cases $R_{i,j}(1)$ on the day 1 for county j from state i, after the power parameter α_i is estimated using the state-level observations to minimize the same loss function below:

$$\begin{aligned}(\hat{I}_{i,j}(1), \hat{R}_{i,j}(1)) = \operatorname{argmin} \sum_{t=1}^{T_{i,j}} \left(\frac{d^o_{i,j}(t) - D_{i,j}(t \mid I_{i,j}(1), R_{i,j}(1))}{T_{i,j} - t + 1} \right)^2, \ s.t. \\ 0 \leq I_{i,j}(1) + R_{i,j}(1) \leq U_{i,j},\ I_{i,j}(1) \geq 0, \text{ and } R_{i,j}(1) \geq 0,\end{aligned} \tag{3.9}$$

where the upper bound $U_{i,j}$ is chosen to guarantee the estimated number of the susceptible cases $S_{i,j}(t)$ to be larger than 0:

$$U_{i,j} = N_{i,j} \frac{(p_i(1))^{\alpha_i} c^o_{i,j}(1)}{\mathbb{1}_{\{T_{i,j} \geq 2\}} \sum_{s=2}^{T_{i,j}} (p_i(s))^{\alpha_i} \Delta c^o_{i,j}(s) + (p_i(1))^{\alpha_i} c^o_{i,j}(1)} - (D_{i,j}(1) + C_{i,j}(1)),$$

for $t = 1, 2, \ldots, T_{i,j}$.

After the initial values of infectious and resolving cases are estimated, we obtain the estimation of the susceptible cases from Equation (3.7), and the infectious cases and transmission rates on each date for each county from Algorithm 1. The resolving cases, deaths, and recovered cases can be derived subsequently from Equations (3.4)-(3.6), respectively. The estimated basic and effective reproduction rates can be derived by the fitted time-dependent transmission rate, and the estimated probability of contracting SARS-CoV-2 for an individual can be computed based on transmission rate and number of infectious individuals for each county on each day.

Forecast and uncertainty assessment. Our method can also be used as a tool for forecasting compartments (e.g., death toll), reproduction numbers, and the probability of contracting SARS-CoV-2 at each county for a short period. We extrapolate the transmission rate based on Gaussian processes implemented in `RobustGaSP` R package

[119] with robust parameter estimation [69, 70]. Based on the extrapolated transmission rates, the compartments can be solved iteratively based on Equations (3.2)-(3.6).

We also found that the forecast will generally be improved by modeling residuals between observed deaths and modeled deaths by a zero-mean Gaussian process (GP). One advantage of a GP model is the internal assessment of the uncertainty of the forecast from the predictive distribution, which is of crucial importance. The aggregated model that combines the SIRDC model and the GP model for county j from state i in the US is described as follows.

$$d^o_{i,j}(t) = D_{i,j}(t) + z_{i,j}(t) + \epsilon_{i,j,t}, \tag{3.10}$$

where $d^o_{i,j}(t)$ and $D_{i,j}(t)$ denote the observed death toll and estimated death toll via the SIRDC model, respectively; The noise follows independently as a Gaussian distribution $\epsilon_{i,j,t} \sim N(0, \sigma^2_{i,j,0})$ with variance parameter $\sigma^2_{i,j,0}$. The latent temporal process $z_{i,j}(t)$ is modeled by a zero-mean GP, meaning that for time points $\{1, 2, \ldots, T_{i,j}\}$, $\mathbf{z}_{i,j} = (z_{i,j}(1), \ldots, z_{i,j}(T_{i,j}))^T$ follows a multivariate normal distribution:

$$\mathbf{z}_{i,j} \sim \mathcal{MN}(\mathbf{0}, \sigma^2_{i,j}\mathbf{R}_{i,j}),$$

where $\sigma^2_{i,j}$ is the variance parameter and the (l, m) entry of $\mathbf{R}_{i,j}$ is parameterized by a correlation function $c_{i,j}(l, m)$ for $1 \leq l, m \leq T_{i,j}$. We use the power exponential correlation function:

$$c_{i,j}(l, m) = \exp\left\{-\left(\frac{\mid l - m \mid}{\gamma_{i,j}}\right)^a\right\},$$

where a is the roughness parameter fixed to be 1.9 as in other studies [18, 62], to avoid possible singularity in inversion of the covariance matrix using the Gaussian correlation ($a = 2$), and $\gamma_{i,j}$ is a range parameter for each county estimated from the data. We define the nugget parameter $\eta_{i,j} = \sigma^2_{i,j,0}/\sigma^2_{i,j}$. The range parameter $\gamma_{i,j}$, and the nugget parameter $\eta_{i,j}$ in Equation (3.10) are estimated based on the marginal posterior mode estimation using the `rgasp` function in the package `RobustGaSP` available on CRAN [69].

Denote $\mathbf{d}^0_{i,j} = (d^o_{i,j}(1), ..., d^o_{i,j}(T_{i,j}))^T$and $\mathbf{D}_{i,j} = (D_{i,j}(1), ..., D_{i,j}(T_{i,j}))^T$. After marginalizing out the variance parameter by the reference prior $\pi(\sigma^2_{i,j}) \propto 1/\sigma^2_{i,j}$, for any t^*, the predictive distribution of $z_{i,j}(t^*)$, conditional on the observations, range parameter $b_{i,j}$ and nugget parameter $\eta_{i,j}$, follows a non-central Student's t-distribution with degrees of freedom $T_{i,j}$ [69]

$$z_{i,j}\left(t^*\right) \mid \mathbf{d}^o_{i,j}, \mathbf{D}_{i,j}, \gamma_{i,j}, \eta_{i,j} \sim \mathcal{T}\left(\hat{z}_{i,j}\left(t^*\right), \hat{\sigma}^2_{i,j}\tilde{c}^*_{i,j}, T_{i,j}\right), \tag{3.11}$$

where

$$\begin{aligned}\hat{z}_{i,j}\left(t^*\right) =& D_{i,j}(t^*) + \mathbf{r}^T_{i,j}\left(t^*\right)\mathbf{K}^{-1}_{i,j}(\mathbf{d}^o_{i,j} - \mathbf{D}_{i,j}),\\ \hat{\sigma}^2_{i,j} =& \frac{(\mathbf{d}^o_{i,j} - \mathbf{D}_{i,j})^T\mathbf{K}^{-1}_{i,j}(\mathbf{d}^o_{i,j} - \mathbf{D}_{i,j})}{T_{i,j}},\\ \tilde{c}^*_{i,j} =& c_{i,j}\left(t^*, t^*\right) + \eta_{i,j} - \mathbf{r}^T_{i,j}\left(t^*\right)\mathbf{K}^{-1}_{i,j}\mathbf{r}_{i,j}\left(t^*\right),\end{aligned}$$

with $\mathbf{K}_{i,j} = \mathbf{R}_{i,j} + \eta_{i,j}\mathbf{I}_{T_{i,j}}$, the (l,m)th term of $\mathbf{R}_{i,j}$ being $c_{i,j}(l,m)$ for $1 \leq l, m \leq T_{i,j}$, and $\mathbf{r}_{i,j}(t^*) = (c_{i,j}(t^*, 1), \ldots, c_{i,j}(t^*, T_{i,j}))^T$, by plugging in the estimated range parameter $\gamma_{i,j}$ and nugget $\eta_{i,j}$. The predictive mean $\hat{z}_{i,j}(t^*)$ for forecasting the death toll of the jth county in the ith state at a future day t^* and the predictive interval can be computed based on the Student's t distribution. An overview of the forecast algorithm and the numerical comparison of different approaches in forecast is given in the appendix.

3.3 Results

We first verify our model performance by forecasting at the county level. The 7-day and 21-day death projections for $2,277$ US counties using data by 20 September 2020, for instance, are close to the held-out test death toll in these counties, shown

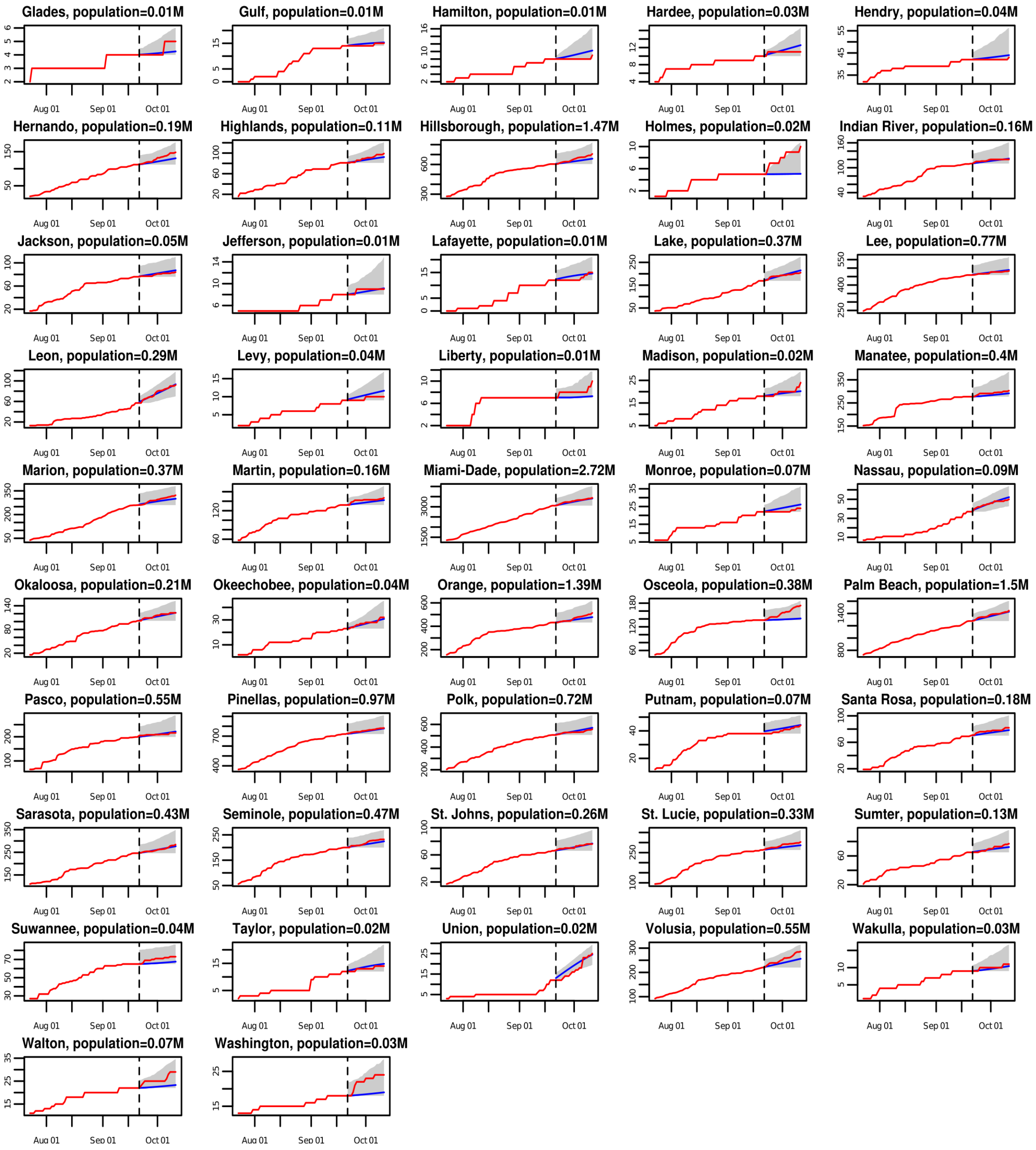


Figure 3.6: The 21-day forecast in 47 Florida counties with death toll no less than 2 as of 20 September 2020. The training period is from 21 March 2020 to 20 September 2020, whereas the forecast starts from 21 September 2020. The red curves are the cumulative observed death toll from 21 September 2020 to 11 October 2020 and the blue line indicates the forecast for the same period. The shaded area represents the 95% predictive intervals of the forecast for each analyzed county in Florida.

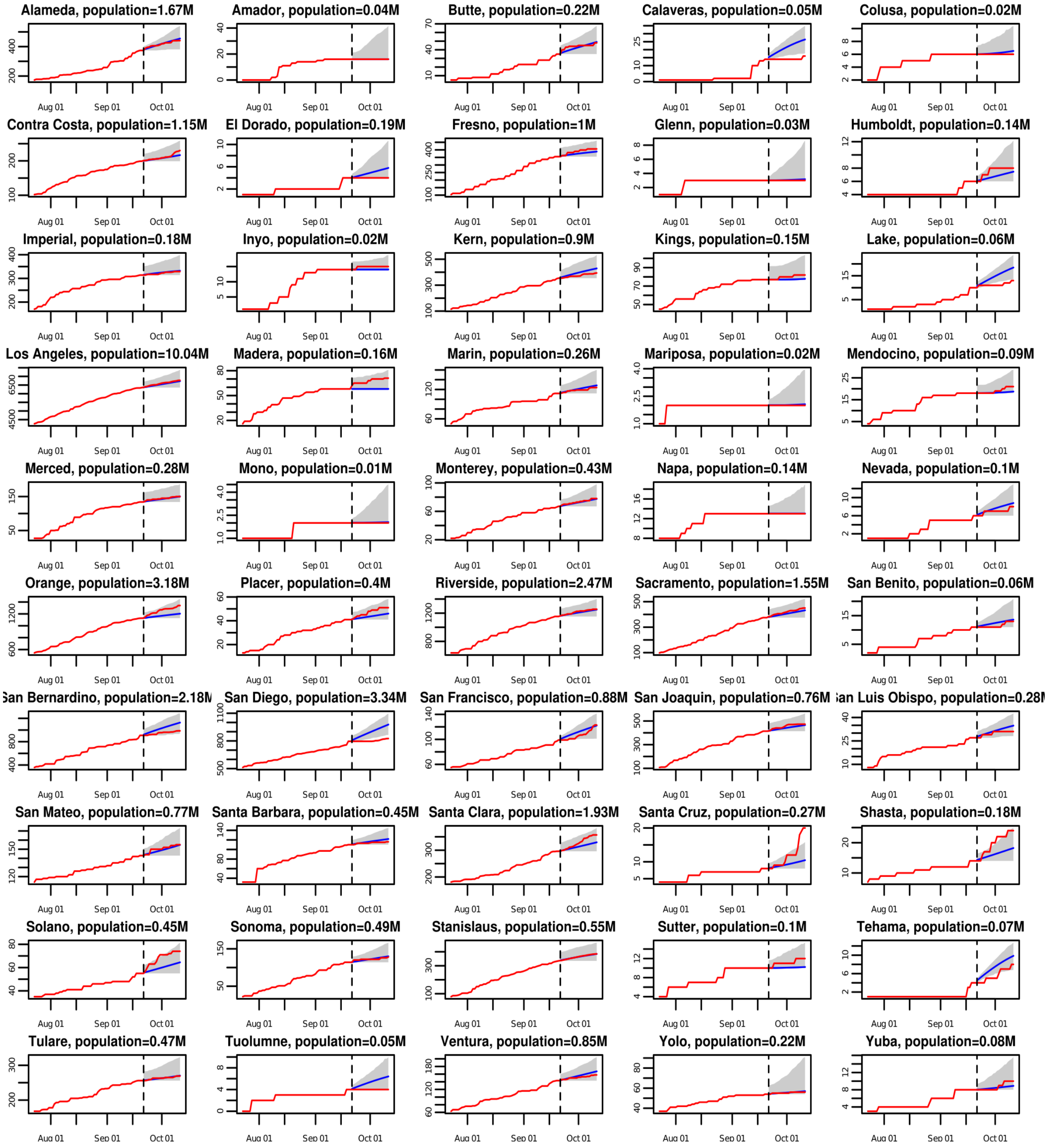


Figure 3.7: The 21-day forecast in 50 California counties with death toll no less than 2 as of 20 September 2020. The training period is from 21 March 2020 to 20 September 2020, whereas the forecast starts from 21 September 2020. The red curves are the cumulative observed death toll from 21 September 2020 to 11 October 2020 and the blue line indicates the forecast for the same period. The shaded area represents the 95% predictive intervals of the forecast for each analyzed county in California.

in part b and part c of Figure 3.1. The Pearson correlation coefficient (ρ) is larger than 0.999 in 7-day and 21-day forecasts. We also calculate the weighted average of the Pearson correlation coefficient for counties (ρ_{county}), which treats each county as a different population and population size is used to compute the weighted average of the Pearson correlation coefficient for counties. The 21-day forecast of each considered county in Florida and California using observations by 20 September 2020 is provided in Figures 3.6 and 3.7, respectively. The death toll forecast based on our model is accurate for most US counties, and around 95% of the held-out test data is covered by a nominal 95% predictive interval (Table S1 in appendix), indicating that the uncertainty assessment is accurate. To further test the predictive performance of our model, we use data by 1 December, 2020 to make 21-day and 90-day predictions of deaths in the 10 largest counties in Florida and California. The forecast results are shown in Figures 3.8 and 3.9, respectively. While this is a challenging scenario, as confirmed cases and deaths increase dramatically across the US during the winter, we found that our 21-day predictions are reasonably accurate for all 20 counties. Thus, our models can be used reliably for the short-term projection of COVID-19 related deaths at the county level during different periods of the epidemic. Furthermore, a 90-day accurate forecast of US counties before the winter may be an almost impossible task, and indeed we underestimate death counts for a few counties due to a rapid increase in death counts during the winter. On the other hand, our model that fuses test data and death toll correctly projects the rapid increase in death counts for most counties during the winter, even if death counts do not increase dramatically during the training period.

Based on the robust estimation of transmission rates, we derived the county-level estimation of daily PoC SARS-CoV-2. We classify the daily PoC SARS-CoV-2 in a community into five levels listed in Table 3.3. On 20 September 2020, out of 2,277 US counties, only 60 counties were at the controllable level and 311 counties were at the

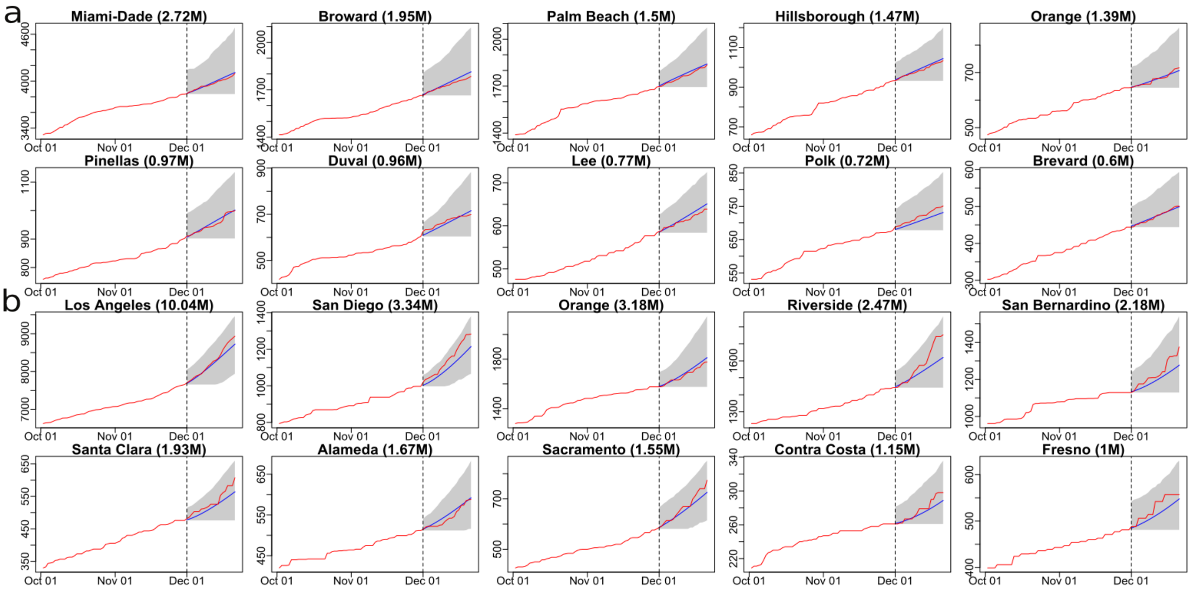


Figure 3.8: a, the 21-day death toll forecast in 10 counties with the largest population in Florida. The training period is from 21 March 2020 to 30 November 2020, whereas the forecast starts from 1 December 2020. The red curves are the cumulative observed death toll and the blue line indicates the forecast from 1 December 2020 to 21 December 2020. The shaded area represents the 95% predictive intervals of the forecast for each analyzed county in Florida. The numbers in the parentheses are the populations in million for each county. b. the 21-day forecast in 10 counties with the largest population in California. The interpretations are the same as a.

Table 3.3: Interpretation of the daily PoC SARS-CoV-2 in a community.

Daily PoC SARS-CoV-2	< 0.001%	0.001% to 0.01%	0.01% to 0.1%	0.1% to 1%	> 1%
Risk	controllable	moderate	alarming	strongly alarming	hazardous

moderate level, whereas 1906 counties were at the either alarming, strongly alarming, or hazardous level. The daily PoC SARS-CoV-2 measures the average probability to contract SARS-CoV-2 for a susceptible individual in a community, and the risk varies from individuals to individuals. Nonetheless, the PoC SARS-CoV-2 is an interpretable measure for public understanding of the average risk of contracting SARS-CoV-2 in a community on a given day.

We graph the estimated PoC SARS-CoV-2 of an individual at US counties on 20 April 2020 and 20 September 2020 in Figure 3.2. On 20 April 2020, the PoC SARS-CoV-2 is

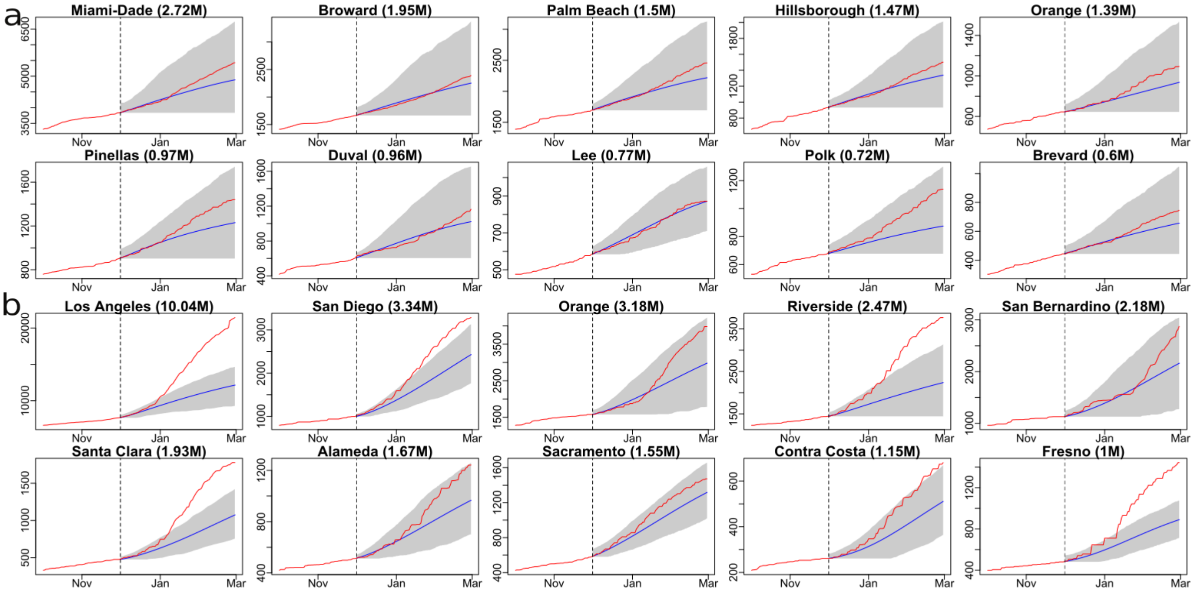


Figure 3.9: a, the 90-day forecast in 10 counties with the largest population in Florida. The training period is from 21 March 2020 to 30 November 2020, whereas the forecast starts from 1 December 2020. The red curves are the cumulative observed death toll and the blue line indicates the forecast from 1 December 2020 to 21 December 2020. The shaded area represents the 95% predictive intervals of the forecast for each analyzed county in Florida. The numbers in the parentheses are the populations in million for each county. b. the 90-day forecast in 10 counties with the largest population in California. The interpretations are the same as a.

large in northeastern regions and some southern states such as Arizona, New Mexico, and New Orleans. On 20 September 2020, the PoC SARS-CoV-2 is large in many inland states, for instance, Montana, North Dakota, Mississippi, and Alabama. Although the PoC SARS-CoV-2 on 20 September in northeastern regions is substantially lower than that on 20 April, the PoC SARS-CoV-2 for an individual is large in most other states on 20 September, suggesting that the relaxation of protective measures can lead to more population contracting COVID-19, and consequently more deaths at a rate no slower than that in late April.

Officials can use the daily PoC SARS-CoV-2 to determine whether the mitigation policies can be lifted or replaced by other measures for different regions. The probability of contracting COVID-19 in many counties in Texas on 20 September 2020, for example, is larger than those in Washington (part (a) and (d) in Figure 3.3), indicating that Texas should undertake more protective measures to reduce the risk. The nationwide lockdown order and social distancing in spring effectively reduced the PoC SARS-CoV-2 in 4 out of 5 counties in Washington, while the PoC SARS-CoV-2 of all counties increases in late June and early July, as some of the nonpharmaceutical interventions (NPIs) were lifted (part b in Figure 3.3). Part (c) shows that the model fits the death toll. With only two parameters estimated numerically for each county, the fit is reasonably good for these counties at a wide range of dates. In comparison, though the outbreak of 5 counties in Texas started in early summer, the PoC SARS-CoV-2 in these Texas counties is much higher than that in Washington counties on 20 September (part (e) in Figure 3.3). Our model also fits the death toll of the counties in Texas relatively well (part (f) in Figure 3.3). The county-level estimation and forecast are updated regularly on the COVID-19 US dashboard: https://covid19-study.pstat.ucsb.edu/.

The effectiveness of protective measures were studied to reduce the transmission rate [41, 42, 45, 46, 48, 116], whereas the efficacy of these measures depends on the reactions

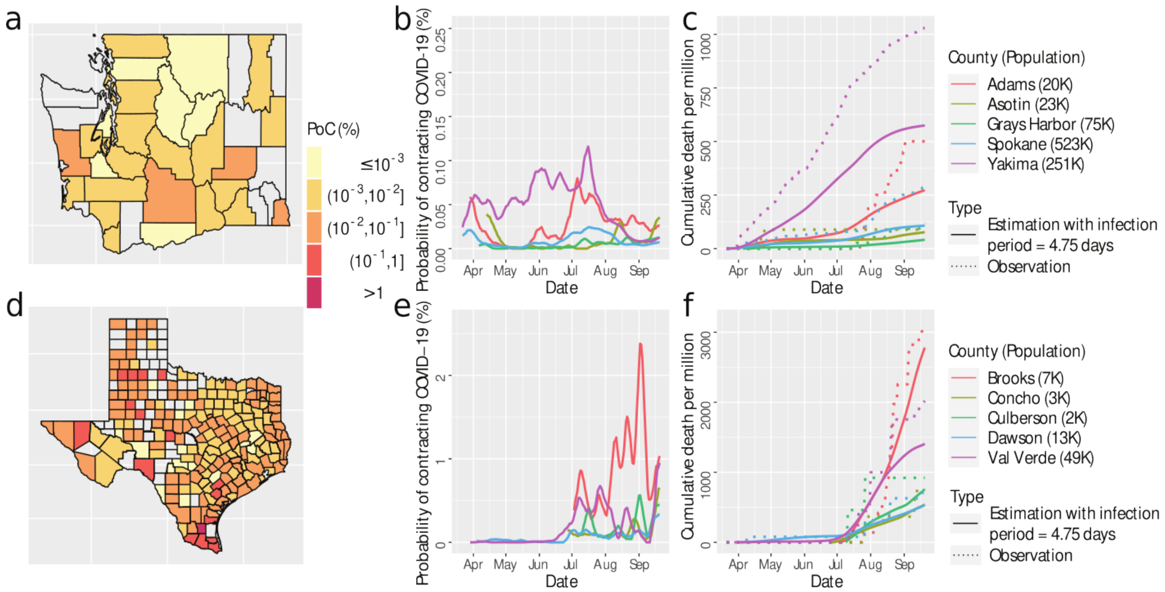


Figure 3.10: a-f, the simulated results of COVID-19 progression in Washington (the first row) and in Texas (the second row) that have the same interpretation as a-f in Figure 3 with the infection period changed from 5 days, to 4.75 days, whereas other parameters are held the same.

from the public, which is likely to vary from region to region. Another simultaneous effort to mitigate the spread of the COVID-19 outbreak is through testing and contact tracing, which reduces the infectious period, and consequently, the number of active infectious individuals. For Washington and Texas, we simulate the model output with infectious period reduced by 5% (or equivalently 4.75 days in total), while the transmission rate (β_t in SIRDC model) is held the same. We found that the PoC SARS-CoV-2 is reduced by 5 times for 12 counties out of 28 considered counties in Washington and 6 counties out of 209 considered counties in Texas, as shown in the Figure 3.10. Furthermore, when we reduce the infectious period by 10% (or equivalently 4.5 days in total), while the transmission rate (β_t in SIRDC model) is held the same, the PoC SARS-CoV-2 is reduced by 5 times for 26 out of 28 counties in Washington and 146 out of 209 counties in Texas, shown in Figure 3.11.

We graph the estimated effective reproduction number, the number of active infectious

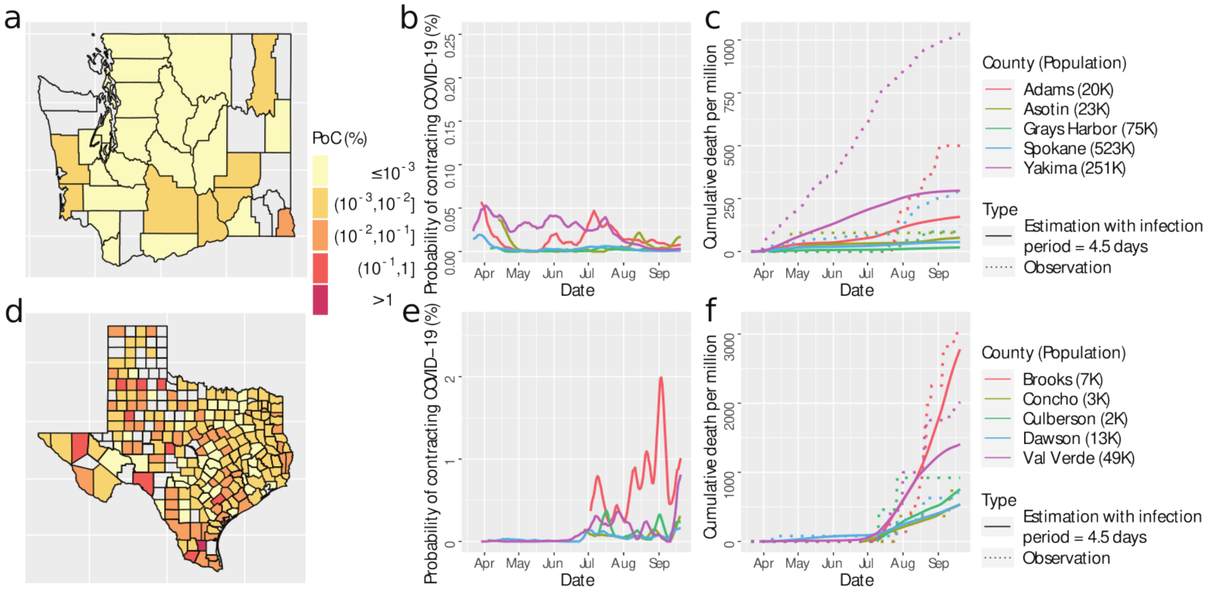


Figure 3.11: a-f, the simulated results of COVID-19 progression characteristics in Washington (the first row) and in Texas (the second row) that have the same interpretation as a-f in Figure 3 with the infection period changed from 5 days to 4.5 days, whereas other parameters are held the same.

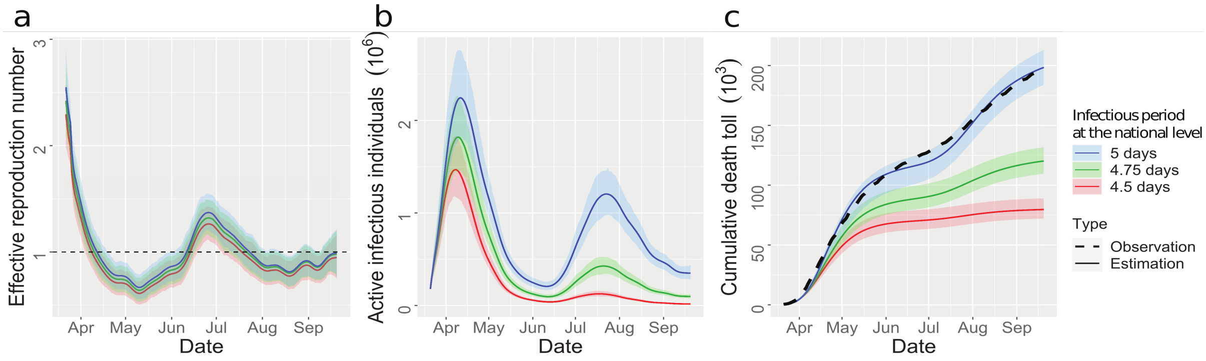


Figure 3.12: a, b, The estimate reproduction number and overall number of active infective individuals in the US, including 50 states and Washington D.C., from 21 March 2020 to 20 September 2020 with infectious period assumed to be 5 days (blue), 4.75 days (green) and 4.5 days (red). c, The estimate overall death toll in the US. The time period and interpretation of c are aligned with a and b, except that the black dots in c stand for the observed death toll in the US.

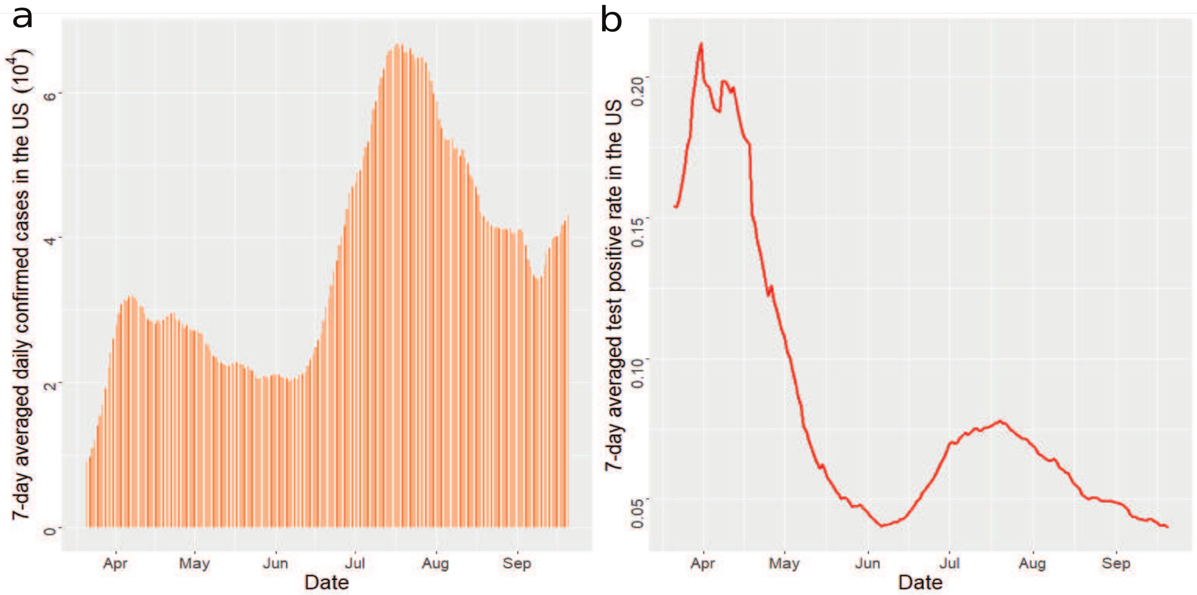


Figure 3.13: a, the 7-day averaged daily confirmed cases in the US from 21 March 2020 to 20 September 2020. b, the 7-day averaged test positive rate in the US from 21 March 2020 to 20 September 2020.

individuals, and the cumulative death toll in the US, along with the simulated values when the average infectious period is reduced from 5 days to 4.75 days and 4.5 days in Figure 3.12. First, we found that mitigation measures in March effectively reduced the effective reproduction number to below 1, whereas the value rebounded in summer after some of these measures were relaxed in different regions. Consequently, the US has experienced two waves of the outbreak in terms of the number of active infectious individuals (part b in Figure 3.12). The high test positive rate at the beginning of the epidemic (part b in Figure 3.13) indicates that a substantial number of active infectious individuals were not diagnosed in April due to the lack of diagnostic tests. According to our estimates, the peak of the first wave in April is larger than that of the second wave in July in terms of the number of active infectious individuals, whereas the peak of the daily observed confirmed cases in April is smaller than that of the second wave in July (part a in Figure 3.13).

Second, the simulated results suggest that shortening infectious period of SARS-CoV-2 by 5% and 10% can reduce the total deaths from 199K to 120K (95% CI: [109K, 132K]) and 80K (95% CI: [72K, 89K]), respectively, as of 20 September 2020, when other protective measures were held as the same (part c in Figure 3.12). Note that since we held the transmission rate parameter (β_t) to be the same (a scenario where the public adheres to the protective measure same as the reality), the effective reproduction number barely changes (part a in Figure 3.12). However, the slightly shortened infectious periods of SARS-CoV-2 can reduce the death toll substantially (part c in Figure 3.12), as the number of active infectious individuals decreases (part b in Figure 3.12).

We found that a shortened infectious period substantially reduces the number of active infectious individuals and fatalities in the second wave. However, the changes are smaller in the first wave, since the effective reproduction number in the second wave is smaller than that in the first wave (Figure 3.12). The county level estimation also validates this point (Figures 3.10 and 3.11). This finding indicates that the efforts to shorten the infectious period of SARS-CoV-2 should not replace the other protective measures, such as social distancing and facial mask-wearing to reduce the transmission rate.

Diagnostic tests can be used to shorten the length of the infectious period of an active infectious individual. Drastically reducing the infectious period may not be possible without contact tracing, which is challenging when there is a large number of active infective cases. Reducing the infectious period by around 5%, in comparison, may be achieved by periodically diagnostic tests every 20 days for each susceptible individual. More frequent testing or contact tracing may be needed to achieve this goal, as the infection is most likely to happen between days 2 and 6 after exposure due to the high viral load of SARS-CoV-2 [120]. Another efficient way is to test susceptible individuals with a high risk of contracting or spreading SARS-CoV-2, such as individuals with more daily contacts or have contacts with vulnerable populations, e.g., workers from senior living facilities. Our estimation of the PoC SARS-CoV-2 can be used as a response to develop regression models using covariates including demographic information and mobility to elicit personalized risk of contracting SARS-CoV-2 for susceptible individuals.

Finally, efforts on reducing the length of the infectious period should not replace other protective measures for reducing transmission rates of SARS-CoV-2, as the number of active infectious individuals and death toll can be effectively reduced only if the effective reproduction number is not substantially larger than 1.

Chapter 4

Gaussian orthogonal latent factor processes for large incomplete matrices of correlated data

In this chapter, we propose the Gaussian orthogonal latent factor (GOLF) processes, designed for the accurate modeling and prediction of extensive correlated datasets. The approach involves decomposing the likelihood function of a Gaussian random field into densities at orthogonal components, utilizing the continuous-time Kalman filter for efficient likelihood function computation without approximations. The method ensures the independence of the posterior distribution of factor processes, arising from the orthogonal nature of the factor loading matrix and the prior independence of the latent factor processes. For large-scale datasets, a flexible mean modeling approach is proposed, along with a solution for identifiability issues in parameter sampling. This method's effectiveness is validated through both simulated and real data applications.

4.1 Introduction

Large spatial, spatio-temporal, and functional data are commonly used in various studies, including geological hazard quantification, engineering, and medical imaging, to facilitate scientific discoveries. Many data sets are observed on incomplete matrices with missing values due to the limitation of the technique or computational cost.

Gaussian processes (GPs) are widely used for modeling correlated data [2, 121]. Computing the likelihood function from a GP model, however, generally takes $O(N_o^3)$ operations in finding the inverse and determinant of the covariance matrix, where N_o is the number of observations. The computational bottleneck prevents modeling a large correlated data set by GPs directly. Tremendous efforts have been made to approximate a GP model in recent studies, including, for example, stochastic partial differential equation approach [52, 53], hierarchical nearest neighbor methods [54], multi-resolution process [55], local Gaussian process approach [56], periodic embedding [57, 58] and covariance tapering [59], which have obtained wide attention in recent years. Compared to a large number of studies on approximating GPs, less progress has been made in efficiently computing the likelihood function without approximation.

In this work, we propose a flexible and computationally feasible approach to model large incomplete matrix observations of correlated data, called Gaussian orthogonal latent factor (GOLF) processes. Bayesian inference was derived to assess the uncertainty in parameter estimation and predictions. GPs with product covariance functions on lattice observations or semiparametric latent factor models [4, 7, 122] can be represented as full-rank GOLF processes, which permit much smaller computational costs than directly computing the likelihood function and making predictions. Further reducing the computational cost can be achieved by low-rank GOLF processes, where the computational cost is similar to the order of principal component analysis. See Section 4.3.3 for a detailed discussion of the computational complexity of the GOLF model in different scenarios.

We highlight a few contributions of this work. We first show that for GPs with product covariance functions or semiparametric latent factor models, if the latent factor loading matrix is orthogonal, prior independence of latent factor processes implies posterior independence of factor processes. The new finding allows one to decompose the likelihood function of lattice data into a product of densities of projected output, which greatly reduces the computational complexity. Separate continuous-time Kalman filters can be applied to compute the posterior distributions of factor processes at lower dimensional inputs in parallel, which has linear computational operations with respect to the number of observations. Second, as a large number of observations provide rich information, we introduce a flexible way to model the mean function and derive the marginal posterior distribution of the linear coefficients, to solve identifiability issues in posterior sampling. Furthermore, compared with the maximum marginal likelihood estimation of factor loadings derived in [123], our approach is applicable to model observations on incomplete lattice. Finally, we developed Bayesian inference for uncertainty assessment, which is critically important for inverse problems in applications [7, 6].

The purpose of this work is twofold. First, we aim to develop a pipeline of computationally efficient methods of modeling correlated data with multi-dimensional input without approximating the likelihood function. The properties of GOLF processes derived in this work are useful for developing an efficient approximation algorithm for scenarios with multi-dimensional input variables. Besides, the nonseparable covariance and coordinate-specific mean coefficients proposed in this work provide flexible choices for models of local information. Second, we primarily focus on applications based on a stack of images in this work, which includes inverse problems by satellite radar interferograms [60], and estimating dynamic information from microscopic videos [124]. Our approach allows for efficient Bayesian inference in a large sample scenario.

The rest of the chapter is organized as follows. In Section 4.2.1, we introduce the GOLF model with an emphasis on the orthogonal decomposition of the likelihood function and posterior independence of latent factor processes. The flexible mean function, spatial latent factor loading matrix, and kernel functions are discussed in Section 4.2.2-4.2.4, respectively. We introduce the Markov Chain Monte Carlo (MCMC) algorithm and discuss the computational complexity in Section 4.3.1. In Section 4.3.2, we introduce the continuous-time Kalman filter in computing the likelihood function with linear computational complexity. Section 4.4 compares our approach with other alternatives, and numerical results for comparing these approaches are presented in Section 4.5-4.6. Proofs of lemmas and theorems are given in the Appendix.

4.2 Gaussian Orthogonal Latent Factor Processes

4.2.1 Orthogonal Decomposition and Posterior Independence

Let $\mathbf{y}_s(\mathbf{x}) = (y_{s_1}(\mathbf{x}), ..., y_{s_{n_1}}(\mathbf{x}))^T$ be an $n_1 \times 1$ vector of observations at coordinates $\mathbf{s} =: (\mathbf{s}_1, ..., \mathbf{s}_{n_1})^T$ with $\mathbf{s}_i \in \mathbb{R}^{p_1}$ for $i = 1, ..., n_1$ and input $\mathbf{x} \in \mathbb{R}^{p_2}$. For spatially correlated data, for instance, s and x denote the latitude and longitude, respectively, and in spatio-temporal models, the spatial coordinates and time points can be defined as $\mathbf{s}$ and x, respectively.

Consider the latent factor model:

$$\mathbf{y}_s(\mathbf{x}) = \mathbf{m}_s(\mathbf{x}) + \mathbf{A}_s\mathbf{z}(\mathbf{x}) + \boldsymbol{\epsilon}, \tag{4.1}$$

where $\mathbf{A}_s = [\mathbf{a}_1, ..., \mathbf{a}_d]$ is a $n_1 \times d$ factor loading matrix and $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), ..., z_d(\mathbf{x}))^T$ is a d-dimensional factor processes with $d \leq n_1$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_0^2\mathbf{I}_{n_1})$ being independent Gaussian noises. The mean function $\mathbf{m}_s(\mathbf{x}) = (m_{s_1}(\mathbf{x}), ..., m_{s_{n_1}}(\mathbf{x}))^T$ is typically modeled via a linear trend of regressors, which will be discussed in Section 4.2.2.

As data are typically positively correlated at two nearby inputs, we assume $z_l(\cdot)$ independently follows a zero-mean Gaussian process (GP), meaning that for any $\{\mathbf{x}_1, ..., \mathbf{x}_{n_2}\}$, $\mathbf{Z}_l^T = (Z_l(\mathbf{x}_1), ..., Z_l(\mathbf{x}_{n_2}))^T$ is a multivariate normal distribution:

$$(\mathbf{Z}_l^T \mid \mathbf{\Sigma}_l) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_l) \tag{4.2}$$

where the (i, j)th entry of the covariance matrix is $\sigma_l^2 c_l(\mathbf{x}_i, \mathbf{x}_j)$ with kernel function $c_l(\cdot, \cdot)$ and variance parameter σ_l^2, for $l = 1, ..., d$. Here we assume independence between the factor processes *a priori*. A detailed comparison between our approach and other related approaches is discussed in Section 4.4.

Note that only the d-dimensional linear subspace of factor loadings $\mathbf{A}_s$ can be identified if not further specification of factor loading matrix $\mathbf{A}_s$ is made, as the model in Equation (4.1) is unchanged if the pair $(\mathbf{A}_s, \mathbf{z}(\mathbf{x}))$ is replaced by $(\mathbf{A}_s\mathbf{G}, \mathbf{G}^{-1}\mathbf{z}(\mathbf{x}))$ for any invertible matrix $\mathbf{G}$. Besides, the computation could be challenging when the number of factors or input parameters is large. Thus, we assume that the column of $\mathbf{A}_s$ is orthonormal.

Assumption 3

$$\mathbf{A}_s^T \mathbf{A}_s = \mathbf{I}_d. \tag{4.3}$$

Assumption 3 may be replaced by $\mathbf{A}_s^T \mathbf{A}_s = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix. Since we estimate variance parameters $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_d^2)^T$ of latent factor processes by data, diagonal terms of $\mathbf{\Lambda}$ are redundant. Thus we proceed with the Assumption 3.

Let us first assume we have an $n_1 \times n_2$ matrix of observations $\mathbf{Y} = [\mathbf{y}_s(\mathbf{x}_1), ..., \mathbf{y}_s(\mathbf{x}_{n_2})]$ at inputs $\{\mathbf{x}_1, ..., \mathbf{x}_{n_2}\}$, and then we extend our method to incomplete matrix observations in the Section 4.3. Denote $\mathbf{B}$ the regression parameters in the $n_1 \times n_2$ mean matrix $\mathbf{M} = (\mathbf{m}_s(\mathbf{x}_1), ..., \mathbf{m}_s(\mathbf{x}_{n_2}))$. Denote $\boldsymbol{\Theta} = (\mathbf{A}_s, \mathbf{B}, \boldsymbol{\sigma}^2, \boldsymbol{\gamma})$, which contains the factor loadings, mean parameters, variance parameters and range parameters in the kernel functions. Further let $\mathbf{A}_F = [\mathbf{A}_s, \mathbf{A}_c] = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{n_1}]$, where $\mathbf{A}_c$ is an $n_1 \times (n_1 - d)$ matrix of the orthogonal complement of $\mathbf{A}_s$. Assumption 3 allows us to decompose the marginal likelihood (after integrating out the random factor $\mathbf{Z}$) into a product of multivariate normal densities of the outcomes at the projected coordinates:

$$p(\mathbf{Y} \mid \boldsymbol{\Theta}) = \prod_{l=1}^{d} \mathcal{PN}(\tilde{\mathbf{y}}_l; \mathbf{0}, \tilde{\boldsymbol{\Sigma}}_l) \prod_{l=d+1}^{n_1} \mathcal{PN}(\tilde{\mathbf{y}}_l; \mathbf{0}, \sigma_0^2 \mathbf{I}_{n_1}), \tag{4.4}$$

where $\tilde{\mathbf{y}}_l = (\mathbf{Y} - \mathbf{M})^T \mathbf{a}_l$ for $l = 1, ..., d$, and $\tilde{\mathbf{y}}_l = (\mathbf{Y} - \mathbf{M})^T \mathbf{a}_l$ with $\mathbf{a}_l$ being the $(l-d)$th column of $\mathbf{A}_c$ for $l = d+1, ..., n_1$, $\tilde{\boldsymbol{\Sigma}}_l = \boldsymbol{\Sigma}_l + \sigma_0^2 \mathbf{I}_{n_2}$ and $\mathcal{PN}(\cdot\,; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the density of the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. In practice, note that we can avoid computing $\mathbf{A}_c$ by using the identity $\mathbf{A}_s \mathbf{A}_s^T + \mathbf{A}_c \mathbf{A}_c^T = \mathbf{I}_{n_1}$. The derivation of Equation (4.4) derivation is given in Section C.1.2 of the appendix.

The orthogonal factor loading matrix in Assumption 3 and prior independence of factor processes lead to the posterior independence of the factor processes, introduced in the following corollary.

Corollary 1 *For model in Equation (4.1) with Assumption 3:*

1. *The covariance of the posterior marginal distributions of any two factor processes is zero:* $\mathrm{Cov}[\mathbf{Z}_l^T, \mathbf{Z}_m^T \mid \mathbf{Y}, \boldsymbol{\Theta}] = \mathbf{0}_{n_2 \times n_2}$*, where* $l = 1, ..., d$*,* $m = 1, ..., d$ *and* $l \neq m$*.*
2. *For* $l = 1, ..., d$*, the posterior distribution* $(\mathbf{Z}_l^T \mid \mathbf{Y}, \boldsymbol{\Theta})$ *follows a multivariate normal distribution*

$$\mathbf{Z}_l^T \mid \mathbf{Y}, \boldsymbol{\Theta} \sim \mathcal{N}\left(\boldsymbol{\mu}_{Z_l}, \boldsymbol{\Sigma}_{Z_l}\right), \tag{4.5}$$

where $\boldsymbol{\mu}_{Z_l} = \boldsymbol{\Sigma}_l \tilde{\boldsymbol{\Sigma}}_l^{-1} \tilde{\mathbf{y}}_l$ *and* $\boldsymbol{\Sigma}_{Z_l} = \boldsymbol{\Sigma}_l - \boldsymbol{\Sigma}_l \tilde{\boldsymbol{\Sigma}}_l^{-1} \boldsymbol{\Sigma}_l$ *with* $\tilde{\boldsymbol{\Sigma}}_l = \boldsymbol{\Sigma}_l + \sigma_0^2 \mathbf{I}_{n_2}$*.*

We call the latent factor processes in Equation (4.1) with Assumption 3 *Gaussian orthogonal latent factor* (GOLF) processes, because of orthogonal decomposition of the likelihood function and posterior independence between two factor processes. The main idea is to decompose the likelihood of GP models with multi-dimensional inputs by a product of densities with low dimension input and to utilize the continuous-time Kalman filter for fast computation. As we will see in Section 4.3, these two properties dramatically ease the computational burden.

4.2.2 Flexible Mean Function and Marginalization

The mean function $m_s(\cdot)$ plays an important role in modeling and predicting correlated data. Computer models (such as the numerical solution of partial differential equations), for example, can be included as a part of the mean in an inverse problem [7]. Here for simplicity, we use only a linear basis function of $\mathbf{s}$ and $\mathbf{x}$, whereas additional terms may be included in the mean if available.

In a GP model, the regression coefficients are often assumed to be the same across one basis function. For instance, the mean function may be modeled as $\mathbf{m}_s(\mathbf{x}) = \mathbf{h}_1(\mathbf{s})\mathbf{b}_{1,0}$, or $\mathbf{m}_s(\mathbf{x}) = \mathbf{h}_2(\mathbf{x})\mathbf{b}_{2,0}$, where $\mathbf{h}_1(\mathbf{s})$ and $\mathbf{h}_2(\mathbf{x})$ are a set of $1 \times q_1$ and $1 \times q_2$ mean basis functions with $\mathbf{b}_{1,0}$ and $\mathbf{b}_{2,0}$ being $q_1 \times 1$ and $q_2 \times 1$ regression coefficients, respectively. The regression coefficients $\mathbf{b}_{1,0}$, for example, are shared across each $\mathbf{x}$.

The shared regression coefficients may be a restrictive assumption when data sets are large. Consider, for instance, the temperature data set used in [125], where the temperature values are shown in Figure 4.5. In Figure 4.1, we graph the fitted linear regression coefficients using latitudes or longitudes as regressors. The estimated regression coefficients are not the same across latitude or longitude. A natural extension of modeling the mean function, therefore, is to allow the mean parameters at each row or column

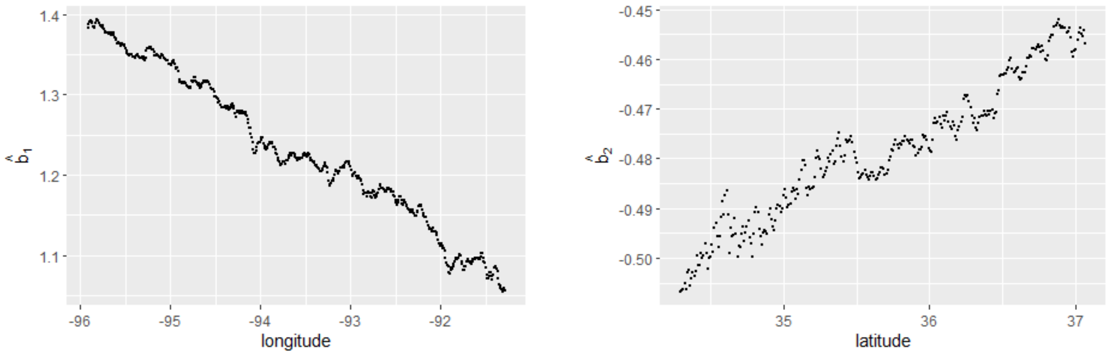


Figure 4.1: Estimated linear coefficients for temperature observations in [125]. In the left panel, the dots are the estimated coefficients in a linear regression of observations at each longitude separately using latitudes as regressors. The estimated linear coefficients for the observations at each latitude are graphed in the right panel, where longitudes are used as regressors.

Individual mean	$\mathbf{m}_{s_i}(\mathbf{x}_j)$	$\mathbf{M}$	coefficients $\mathbf{B}$
Linear trend of $\mathbf{s}$	$\mathbf{h}_1(\mathbf{s}_i)\mathbf{b}_{1,j}$	$\mathbf{H}_1\mathbf{B}_1$	$\mathbf{B}_1$
Linear trend of $\mathbf{x}$	$\mathbf{h}_2(\mathbf{x}_j)\mathbf{b}_{2,i}$	$(\mathbf{H}_2\mathbf{B}_2)^T$	$\mathbf{B}_2$
Mixed linear trend	$\mathbf{h}_1(\mathbf{s}_i)\mathbf{b}_{1,j} + \mathbf{h}_2(\mathbf{x}_j)\mathbf{b}_{2,i}$	$\mathbf{H}_1\mathbf{B}_1 + (\mathbf{H}_2\mathbf{B}_2)^T$	$[\mathbf{B}_1, \mathbf{B}_2]$

Table 4.1: Summary of the mean function studied in this work. In the third column, $\mathbf{H}_1 = (\mathbf{h}_1^T(\mathbf{s}_1), ..., \mathbf{h}_1^T(\mathbf{s}_{n_1}))^T$ and $\mathbf{H}_2 = (\mathbf{h}_2^T(\mathbf{x}_1), ..., \mathbf{h}_2^T(\mathbf{x}_{n_2}))^T$ are $n_1 \times q_1$ and $n_2 \times q_2$ mean basis matrices, respectively. Regression coefficients are denoted as $\mathbf{B}_1 = (\mathbf{b}_{1,1}, ..., \mathbf{b}_{1,n_2})$ and $\mathbf{B}_2 = (\mathbf{b}_{2,1}, ..., \mathbf{b}_{2,n_1})$ for the basis function $\mathbf{h}_1(\cdot)$ and $\mathbf{h}_2(\cdot)$, respectively.

of the observations to be different, e.g. $\mathbf{m}_{s_i}(\mathbf{x}_j) = \mathbf{h}_1(\mathbf{s}_i)\mathbf{b}_{1,j}$, or $\mathbf{m}_{s_i}(\mathbf{x}_j) = \mathbf{h}_2(\mathbf{x}_j)\mathbf{b}_{2,i}$, for $i = 1, ..., n_1$ and $j = 1, ..., n_2$. Some choices of the individual mean functions are summarized in Table 4.1.

The mean function may be specified based on model interpretation or exploratory data analysis. Models with different regression coefficients across different types of coordinates are more suitable to model a large number of observations, as they are more flexible to capture the trend.

To implement full Bayesian inference of the parameters, one may sample from the posterior distribution of regression parameters $p(\mathbf{B} \mid \mathbf{\Theta}_{-B}, \mathbf{Y}, \mathbf{Z})$. However, we found a severe identifiability problem between the mean $\mathbf{M}$ and $\mathbf{AZ}$, when the regression coefficients $\mathbf{B}$ are sampled from the full posterior distribution. This is because the likelihood function of the mean parameters is flat when data are very correlated. Consequently, the absolute values of the entries of these two matrices can be both big, making the MCMC algorithm very unstable. To alleviate the identifiability problem, we first integrate out factors and sample regression parameters from the marginal posterior distribution $p(\mathbf{B} \mid \mathbf{\Theta}_{-B}, \mathbf{Y})$. The marginal posterior distributions of the regression parameters are given in the following Theorem 2 and Theorem 3.

Theorem 2 *1. (Row regression coefficients). Assume* $\mathbf{M} = \mathbf{H}_1\mathbf{B}_1$ *and the objective prior* $\pi(\mathbf{B}_1) \propto 1$ *for* $\mathbf{B}_1$*. After marginalizing out the factor* $\mathbf{Z}$*, the posterior samples of* $\mathbf{B}_1$ *from* $p(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1})$ *can be obtained by*

$$\mathbf{B}_1 = \hat{\mathbf{B}}_1 + (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_s\tilde{\mathbf{B}}_{1,0,s}^T + \sigma_0(\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T(\mathbf{I}_{n_1} - \mathbf{A}_s\mathbf{A}_s^T)\mathbf{Z}_{0,1} \quad (4.6)$$

where $\hat{\mathbf{B}}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{Y}$*,* $\tilde{\mathbf{B}}_{1,0,s}$ *is an* $n_2 \times d$ *matrix with the lth column independently sampled from* $\mathcal{N}(\mathbf{0}, \tilde{\mathbf{\Sigma}}_l)$ *for* $l = 1, ..., d$*, and* $\mathbf{Z}_{0,1}$ *is an* $n_1 \times n_2$ *matrix with each entry independently sampled from the standard normal distribution.*

2. (Column regression coefficients). Assume $\mathbf{M} = (\mathbf{H}_2\mathbf{B}_2)^T$ *and the objective prior* $\pi(\mathbf{B}_2) \propto 1$ *for the regression parameters* $\mathbf{B}_2$*. After marginalizing out the factor* $\mathbf{Z}$*, the posterior samples of* $\mathbf{B}_2$ *from* $p(\mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_2})$ *can be obtained by*

$$\mathbf{B}_2 = \hat{\mathbf{B}}_2 + \tilde{\mathbf{B}}_{2,0,s}\mathbf{A}_s^T + \sigma_0\mathbf{L}_{H_2}\mathbf{Z}_{0,2}(\mathbf{I}_{n_1} - \mathbf{A}_s\mathbf{A}_s^T), \quad (4.7)$$

where $\hat{\mathbf{B}}_2 = \sum_{l=1}^{d}(\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{Y}^T\mathbf{a}_l\mathbf{a}_l^T + (\mathbf{H}_2^T\mathbf{H}_2)^{-1}\mathbf{H}_2^T\mathbf{Y}^T(\mathbf{I}_{n_1} - \mathbf{A}_s\mathbf{A}_s^T)$ *and* $\tilde{\mathbf{B}}_{2,0,s}$ *is a* $q_2 \times d$ *matrix with the lth column independently sampled from* $\mathcal{N}(\mathbf{0}, (\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1})$ *for* $l = 1, ..., d$*.* $\mathbf{L}_{H_2}$ *is a* $q_2 \times q_2$ *matrix such that* $\mathbf{L}_{H_2}\mathbf{L}_{H_2}^T = (\mathbf{H}_2^T\mathbf{H}_2)^{-1}$ *and* $\mathbf{Z}_{0,2}$ *is a* $q_2 \times n_1$ *matrix with each entry independently sampled from* *the standard normal distribution.*

When both the row regression coefficients and column regression coefficients are in the model, we found that $\mathbf{M}_1 = \mathbf{H}_1\mathbf{B}_1$ and $\mathbf{M}_2 = (\mathbf{H}_2\mathbf{B}_2)^T$ are not identifiable, if we sample $\mathbf{B}_1$ and $\mathbf{B}_2$ from the full conditional distribution. To avoid this problem, we first marginalizing out $\mathbf{B}_2$ and $\mathbf{Z}$ to sample $\mathbf{B}_1$ and then we condition $\mathbf{B}_1$ to sample $\mathbf{B}_2$.

Theorem 3 *Assume* $\mathbf{M} = \mathbf{H}_1\mathbf{B}_1 + (\mathbf{H}_2\mathbf{B}_2)^T$ *and let the objective prior* $\pi(\mathbf{B}_1, \mathbf{B}_2) \propto 1$ *for the regression parameters* $\mathbf{B}_1$ *and* $\mathbf{B}_2$*.*

1. *After marginalizing out* $\mathbf{Z}$ *and* $\mathbf{B}_2$*, the marginal posterior sample of* $\mathbf{B}_1$ *from* $p(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2})$ *can be obtained by*

$$\mathbf{B}_1 = \hat{\mathbf{B}}_1 + (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_s\tilde{\mathbf{B}}_{1,Q}^T + \sigma_0(\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T(\mathbf{I}_{n_1} - \mathbf{A}_s\mathbf{A}_s^T)\mathbf{Z}_{0,1}\mathbf{P}_0, \quad (4.8)$$

   where $\hat{\mathbf{B}}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{Y}$*,* $\tilde{\mathbf{B}}_{1,Q}$ *is an* $n_2 \times d$ *matrix with the lth column independently sampled from* $\mathcal{N}(\mathbf{0}, \mathbf{Q}_{1,l})$*, with* $\mathbf{Q}_{1,l} = \mathbf{P}_l\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{P}_l$ *where* $\mathbf{P}_l = \mathbf{I}_{n_2} - \mathbf{H}_2(\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}$ *for* $l = 1, ..., d$*.* $\mathbf{Z}_{0,1}$ *is an* $n_1 \times n_2$ *matrix with each entry independently sampled from standard normal distribution and* $\mathbf{P}_0 = (\mathbf{I}_{n_2} - \mathbf{H}_2(\mathbf{H}_2^T\mathbf{H}_2)^{-1}\mathbf{H}_2^T)$*.*

2. *Posterior samples of* $\mathbf{B}_2$ *from* $p(\mathbf{B}_2 \mid \mathbf{Y}_{B_1}, \mathbf{\Theta}_{-B_2})$ *can be obtained through equation (4.7) by replacing* $\mathbf{Y}$ *by* $\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1$*.*

In Theorem 2 and Theorem 3, the marginal posterior distribution of the regression coefficients depends on the $n_1 \times d$ factor loading matrix, but not the complement of the factor loading matrix ($\mathbf{A}_c$). Since we do not need to compute $\mathbf{A}_c$, the most computationally intensive terms are those containing the covariance matrix $\mathbf{\Sigma}_l$ and its inverse. Fortunately, each term can be computed with linear complexity with respect to n_2 instead of n_2^3 when the Matérn covariance is used, discussed in Section 4.3.2.

4.2.3 Spatial Latent Factor Loading Matrix

This section discusses a model of the latent factor loading matrix $\mathbf{A}_s$ that satisfies the orthogonal constraint in Equation (4.3). As output values are marginally correlated at two inputs $\mathbf{s}_a$ and $\mathbf{s}_b$, a natural choice is to let $\mathbf{A}_s$ be the eigenvectors corresponding to the largest d eigenvalues in the eigendecomposition of the correlation matrix $\mathbf{R}_s$, where the (i,j)th entry is specified by a kernel function $c_s(\mathbf{s}_i, \mathbf{s}_j)$, for $1 \leq i, j \leq n_1$. We give a few examples of models that can be written as special cases of the GOLF model when the $\mathbf{A}_s$ is specified as eigenvectors of $\mathbf{R}_s$. For simplicity, we assume the mean is zero. The first and second classes of models are the GP models with separable covariance functions of input with two dimensions and three dimensions, respectively.

Example 1 (Spatial model with separable covariance) *Consider a spatial model of* $\mathbf{Y}$ *at a regular* $n_1 \times n_2$ *lattice, where the* (i,j)*th input is* (s_i, x_j) *with* s_i *and* x_j *denoting the* i*th latitude coordinate and* j*th longitude coordinate, respectively. Assume the covariance of the spatial process is separable, meaning that* $\mathbf{Y} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{R}_s \otimes \mathbf{R}_x + \sigma_0^2 \mathbf{I}_{n_1 n_2})$*, where the* (l_1, m_1) *term of* $\mathbf{R}_s$ *is parameterized by the kernel function* $c_s(s_{l_1}, s_{m_1})$ *and the* (l_2, m_2) *term of* $\mathbf{R}_x$ *is* $c_x(x_{l_2}, x_{m_2})$ *for* $1 \leq l_1, m_1 \leq n_1$ *and* $1 \leq l_2, m_2 \leq n_2$*. Let* $\mathbf{R}_s = \mathbf{U}_s \mathbf{\Lambda}_s \mathbf{U}_s^T$*, where* $\mathbf{U}_s$ *is a matrix of eigenvectors and* $\mathbf{\Lambda}_s$ *is a diagonal matrix of eigenvalues of* $\mathbf{R}_s$ *with the* l*th diagonal term* λ_l*. The density of this spatial model is equivalent to the model in Equation (4.1) with* $\mathbf{A}_s = \mathbf{U}_s$*,* $\mathbf{\Sigma}_l = \sigma^2 \lambda_l \mathbf{R}_x$ *and* $d = n_1$*.*

Example 2 (Spatio-temporal model with separable covariance) *Consider a spatio-temporal model of* $\mathbf{Y}$ *at* $n_{1,1} \times n_{1,2} \times n_2$ *lattice, where the* (i,j,k)*th input is* $(s_{1,i}, s_{2,j}, x_k)$*, with* $s_{1,i}$ *and* $s_{2,j}$ *denoting the* i*th latitude coordinate and* j*th longitude coordinate, respectively, and* x_k *denoting the* k*th time point. Let* $n_1 = n_{1,1} \times n_{1,2}$*. Assume the covariance of the spatio-temporal process is separable, meaning that* $\mathbf{Y} \sim \mathcal{N}(0, \sigma^2 \mathbf{R}_{s_1} \otimes \mathbf{R}_{s_2} \otimes \mathbf{R}_x + \sigma_0^2 \mathbf{I}_{n_1 \times n_2})$ *with the* (l_i, m_i)*th term of* $\mathbf{R}_{s_i}$ *parameterized by the kernel function* $c_s(s_{l_i}, s_{m_i})$

with $1 \leq l_i, m_i \leq n_{1,i}$ *for* $i = 1, 2$*, and the* (l_3, m_3)*th term of* $\mathbf{R}_x$ *being* $c_x(x_{l_3}, s_{m_3})$ *with* $1 \leq l_3, m_3 \leq n_2$*. Let* $\mathbf{R}_{s_i} = \mathbf{U}_i \mathbf{\Lambda}_i \mathbf{U}_i^T$ *where* $\mathbf{U}_i$ *is a matrix of eigenvectors and* $\mathbf{\Lambda}_i$ *is a diagonal matrix of eigenvalues* λ_{l_i} *for* $1 \leq l_i \leq n_{1,i}$ *and* $i = 1, 2$*. The density of this spatio-temporal model is equivalent to the model in Equation (4.1) with* $\mathbf{A}_s = \mathbf{U}_1 \otimes \mathbf{U}_2$*,* $\mathbf{\Sigma}_l = \sigma^2 \lambda_{l_1} \lambda_{l_2} \mathbf{R}_x$ *with* $1 \leq l_i, m_i \leq n_{1,i}$ *for* $i = 1, 2$*,* $l = l_1 + (l_2 - 1)n_{1,2}$ *and* $d = n_1$*.*

The separable covariance is widely used in emulating and calibrating computationally expensive computer models with scalar output [4] and vector output [126, 127], whereas the isotropic covariance, i.e., the covariance as a function of Euclidean distance of inputs, is used more often in modeling spatially correlated data [1]. Some anisotropic kernels, such as the geometrically anisotropic kernel, were studied in [128] for modeling spatially correlated observations. Note that the covariance of GOLF processes in Equation (4.1) is not separable in general, as the variance and kernel parameters of each factor process $z_l(\cdot)$ can be different. Different kernel parameters make the model more flexible, as the factor processes corresponding to large eigenvalues are often found to be smoother than the ones corresponding to small eigenvalues. Separable covariance may be restrictive in this regard as factor processes are assumed to have the same kernel and parameters.

Computing the likelihood of GP with separable covariance on a complete $n \times n$ lattice data generally takes $O(N^{3/2})$ operations through eigen-decomposition of sub covariance matrices. This work generalizes this approach to nonseparable covariance for both complete and incomplete lattice observations. One can further reduce the computational complexity by selecting d eigenvectors corresponding to the d largest eigenvectors from the eigendecomposition of the correlation matrix $\mathbf{R}_s$. The proportion of summation of the d largest eigenvalues over the summation of total eigenvalues shall be chosen as large as possible to allow the model to explain the most variability of the signal [8]. We found that using more factors than the truth typically will not incur a large reduction of predictive accuracy, whereas using a much smaller number of factors than the truth will cause a large predictive error (Example 4 in simulated studies). Thus one should be cautious about using a very small number of factors.

4.2.4 Kernel Functions

We first discuss the kernel function for the factor process $Z_l(\cdot)$, $l = 1, ..., d$. We assume a product kernel between the inputs [4], i.e. for any input $\mathbf{x}_a = (x_{a1}, ..., x_{ap_2})$ and $\mathbf{x}_b = (x_{b1}, ..., x_{bp_2})$, $c_l(\mathbf{x}_a, \mathbf{x}_b) = \prod_{i=1}^{p_2} c_{l,i}(|x_{ai} - x_{bi}|)$, where $c_{l,i}(\cdot)$ is a kernel of the lth coordinate of the input for $l = 1, ..., d$ and $i = 1, ..., p_2$.

We focus on Matérn covariance [99] as kernel function $c_{l,i}(\cdot)$ in this work to model. Each kernel contains positive roughness parameter $\alpha_{l,i}$ and a nonnegative range parameter $\gamma_{l,i}$ for $l = 1, ..., d$ and $i = 1, ..., p_2$. The roughness parameter of the Matérn kernel controls the smoothness of the process. When $\alpha_{l,i} = \frac{1}{2}$, the Matérn kernel becomes the exponential kernel: $c_{l,i}(|x_{ai} - x_{bi}|) = \exp(-|x_{ai} - x_{bi}|/\gamma_{l,i})$, and when $\alpha_{l,i} \to \infty$, the Matérn kernel becomes the Gaussian kernel: $c_{l,i}(|x_{ai} - x_{bi}|) = \exp(-|x_{ai} - x_{bi}|^2/(2\gamma_{l,i}^2))$. The half-integer Matérn kernel (i.e. $(2\alpha_{l,i} + 1)/2 \in \mathbb{N}$) has a closed form expression. When $\alpha_{l,i} = 5/2$, for example, the Matérn kernel is

$$c_{l,i}(|x_{ai} - x_{bi}|) = \left(1 + \frac{\sqrt{5}|x_{ai} - x_{bi}|}{\gamma_{l,i}} + \frac{5|x_{ai} - x_{bi}|^2}{3\gamma_{l,i}^2}\right) \exp\left(-\frac{\sqrt{5}|x_{ai} - x_{bi}|}{\gamma_{l,i}}\right), \quad (4.9)$$

for $l = 1, ..., d$ and $i = 1, ..., p_2$.

In constructing GOLF processes, we decompose the density of the GP model with multi-dimensional input into a product of the orthogonal components with lower-dimensional input. This is because the likelihood and the predictive distribution of a GP model with a half-integer Matérn covariance can be computed through linear operations with respect to the sample size by the continuous-time Kalman filter [129] when $p_2 = 1$. The computational advantage will be discussed in Section 4.3.2.

For the factor loading matrix, we let $\mathbf{A}_s$ be the first d eigenvectors of $\mathbf{R}_s$. The kernel functions for $\mathbf{R}_s$ can be chosen similarly as the kernel for the latent factor processes. Without the loss of generality, we assume $\mathbf{R}_s$ is parameterized by a product kernel with the range parameters $\boldsymbol{\gamma}_0$, and the Matérn kernel being used for each coordinate of $\mathbf{s}$.

4.3 Posterior Sampling for GOLF Processes

4.3.1 A Markov Chain Monte Carlo Approach

In many applications, the observations contain missing values. Denote $\mathbf{Y}_v^o$ and $\mathbf{Y}_v^u$ the vectors of observed data and missing data in matrix $\mathbf{Y}$ with size N_o and N_u, respectively. Directly computing the likelihood includes calculating the inverse and determinant of an $N_o \times N_o$ covariance matrix, which has computational operations $O(N_o^3)$ in general, making it infeasible for large number of observations. Here we discuss a computationally feasible way for the GOLF model when observations are from incomplete matrices.

We start with a set of initial values at the locations with missing observations. Denote $\mathbf{Y}_v^{(t)} = \mathrm{vec}(\mathbf{Y}^{(t)}) = [(\mathbf{Y}_v^o)^T, (\mathbf{Y}_v^{u,(t)})^T]^T$ an N-vector, where $\mathbf{Y}_v^o$ and $\mathbf{Y}_v^{u,(t)}$ are vectors of observations and samples at the missing locations in the tth iteration, $t = 1, ..., T$. First, we use a Metropolis algorithm to sample $\boldsymbol{\Theta}^{(t+1)}$ from the marginal posterior distribution $p(\boldsymbol{\Theta} \mid \mathbf{Y}^{(t)})$, where the marginal density is given in Equation (4.4). In the second step, we sample $\mathbf{Z}_l^{(t+1)}$ from $p(\mathbf{Z}_l^{(t+1)} \mid \mathbf{Y}^{(t)}, \boldsymbol{\Theta}^{(t+1)})$ by Equation (4.5) for $l = 1, ..., d$, and then we generate $\mathbf{Y}^{(t+1)} = \mathbf{A}^{(t+1)}\mathbf{Z}^{(t+1)} + \mathbf{E}^{(t+1)}$, where $\mathbf{E}^{(t+1)}$ is an $n_1 \times n_2$ matrix of white noise with variance $\sigma_0^{(t+1)}$ and $\mathbf{A}^{(t+1)}$ is a $n_1 \times d$ matrix of the d eigenvectors corresponding to the d largest eigenvalues from the eigendecomposition of the correlation matrix $\mathbf{R}_s$ in the $(t+1)th$ iteration. We can obtain $\mathbf{Y}_v^{u,(t+1)}$ by the last N_u terms in $\mathbf{Y}_v^{(t+1)}$, for $t = 1, ..., T$.

Note that the observed data $\mathbf{Y}_v^o$ is never changed.

For computational reasons, we define the nugget parameter in each kernel (i.e. the inverse of the signal variance to the noise variance ratio parameter) $\eta_l = \sigma_0^2/\sigma_l^2$ for $l = 1, 2, ..., d$, and the inverse range parameter $\beta_{l,i} = 1/\gamma_{l,i}$, where $i = 1, ..., p_1$ when $l = 0$, and $i = 1, ..., p_2$ when $l \geq 1$. The transformed parameters $\tilde{\mathbf{\Theta}}$ contain the mean parameters $\mathbf{B}$, inverse range parameters $\boldsymbol{\beta} = (\boldsymbol{\beta}_0, ..., \boldsymbol{\beta}_d)$, nugget parameters $\boldsymbol{\eta} = (\eta_1, ..., \eta_d)$ of the factor processes and the variance of the noise σ_0^2.

For mean and noise variance parameters, we use an objective prior $\pi^R(\mathbf{B}, \sigma_0^2) \propto 1/\sigma_0^2$. We assume the jointly robust (JR) prior for the kernel parameters: $\pi^{JR}(\boldsymbol{\beta}_l, \eta_l) \propto (\sum_{i=1}^{p_2}(k_{l,2}\beta_{l,i} + \eta_l))^{c_{l,1}} \exp(-k_{l,3}\sum_{i=1}^{p_2}(k_{l,1}\beta_{l,i} + \eta_l))$ with default parameters $k_{l,1} = 1/2 - p_2$, $k_{l,2} = 1/2$, and $k_{l,3}$ being the average distance between the lth coordinate of two inputs for $l = 1, ..., d$ [70]. Note here $k_{l,1} = 1/2 - p_2$ is the default parameter for the MCMC algorithm, whereas this prior parameter is different if one maximizes the marginal posterior distribution. The jointly robust prior is equivalent to the inverse gamma prior when the input dimension is one without a nugget parameter. The inverse gamma prior is assumed for each coordinate of $\boldsymbol{\beta}_0$ with shape and rate parameter being $-1/2$ and 1, respectively. The JR prior can alleviate the potential numerical problem when the estimated range and nugget parameters are close to the boundary of the parameter space, as the density of the JR prior is close to zero at these scenarios. As the sample size is large, the bias inserted from the prior is small.

The MCMC algorithm of the GOLF model is given in Algorithm 3. In step (1) to step (4) of Algorithm 3, we marginalize out the factor processes to compute the posterior distribution of the parameters. This is critically important as we found severe identifiability problems between the mean matrix $\mathbf{M}$ and $\mathbf{AZ}$ if the parameters are sampled from the full conditional distributions. Moreover, after marginalizing out the factor processes, the covariance matrix of the distribution $\mathcal{PN}(\tilde{\mathbf{y}}_l; \mathbf{0}, \tilde{\mathbf{\Sigma}}_l)$ in Equation (4.4) contains a nugget

Algorithm 3 MCMC algorithm when the kernel parameters are different

```
(1) For $l = 1, ..., d$, sample $(\boldsymbol{\beta}_l^{(t+1)}, \eta_l^{(t+1)})$ from $p(\boldsymbol{\beta}_l, \eta_l \mid \tilde{\mathbf{y}}_l^{(t)})$.
(2) Sample $\boldsymbol{\beta}_0^{(t)}$ from $p(\boldsymbol{\beta}_0^{(t)} \mid \mathbf{Y}^{(t)}, \boldsymbol{\beta}_{1:d}^{(t+1)}, \boldsymbol{\eta}_{1:d}^{(t+1)}, \mathbf{B}^{(t)})$.
(3) Sample $\sigma_0^{(t+1)}$ from $p(\sigma_0^{(t+1)} \mid \mathbf{Y}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)}, \mathbf{B}^{(t)})$.
(4) Sample $\mathbf{B}^{(t+1)}$ from $p(\mathbf{B}^{(t+1)} \mid \mathbf{Y}^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$. Update the mean matrix $\mathbf{M}^{(t+1)}$ and the projected observations $\tilde{\mathbf{y}}_l^{(t)} = (\mathbf{Y} - \mathbf{M}^{(t+1)})^T \mathbf{a}_l$.
(5) For $l = 1, ..., d$, sample $\mathbf{Z}_l^{(t+1)}$ from $p(\mathbf{Z}_l^{(t+1)} \mid \tilde{\mathbf{y}}_l^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$ by Corollary 1 and sample $\mathbf{Y}^{(t+1)}$ by the model in Equation (4.1). Update $\mathbf{Y}_v^{u,(t+1)}$ by the last $N_u$ terms in $\mathbf{Y}_v^{(t+1)}$ and let $\tilde{\boldsymbol{y}}_l^{(t+1)} = (\mathbf{Y}^{(t+1)} - \mathbf{M}^{(t+1)})^T \mathbf{a}_l$.
(6) Update the posterior $p(\boldsymbol{\beta}_l^{(t+1)}, \eta_l^{(t+1)} \mid \tilde{\mathbf{y}}_l^{(t+1)})$ and go back to (1) when $t < T$.
```

term, which makes the computation stable.

The Algorithm 3 can be easily modified for different scenarios. When the factor processes have the same covariance matrix, we can combine step (1) and step (2) to sample the shared kernel and nugget parameter. Step (4) may be skipped if one has zero-mean or modified if one has the shared regression coefficients in the model.

Denote $\boldsymbol{\Sigma}_l = \mathbf{L}_l \mathbf{L}_l^T$ where $\mathbf{L}_l$ is a lower triangular matrix in the Cholesky decomposition of $\boldsymbol{\Sigma}_l$. We need to efficiently compute the terms $|\tilde{\boldsymbol{\Sigma}}_l|$, $\mathbf{L}_l^{-1}\mathbf{v}_l$, $\mathbf{L}_l\mathbf{v}_l$ for any real-valued vector $\mathbf{v}_l := (v_{l,1}, ..., v_{l,n_2})^T$ and sample $(\mathbf{Z}_l^{(t+1)})^T$ from $p((\mathbf{Z}_l^{(t+1)})^T \mid \tilde{\mathbf{y}}_l^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$ for $l = 1, ..., d$. Direct computation of the Cholesky decomposition of $\boldsymbol{\Sigma}_l$ requires $O(n_2^3)$ computational operations for each $l = 1, ..., d$. Luckily, for Matérn covariance with a half-integer roughness parameter and one-dimensional input, computing any of these terms only takes $O(n_2)$ operations without approximation.

4.3.2 Continuous-time Kalman Filter

We briefly review the continuous-time Kalman filter algorithm and the connection between the Gaussian Markov random field and GP with Matérn covariance. The spectral density of the Matérn covariance with the half-integer roughness parameter was shown to be the same as a continuous-time autoregressive process defined as a stochastic differential equation (SDE) [130]. Suppose the observations are $\tilde{\mathbf{y}}_l = (\tilde{y}_{1,1}, ..., \tilde{y}_{l,n_2})^T$. For $j = 1, ..., n_2$ and $l = 1, ..., d$, starting from the initial state $\boldsymbol{\theta}_l(s_0) \sim \mathrm{MN}(\mathbf{0}, \mathbf{W}_l(s_0))$, the solution of the SDE follows [11]:

$$\begin{aligned} \tilde{y}_{l,j} &= \mathbf{F}\boldsymbol{\theta}_l(x_j) + \epsilon_{l,j}, \\ \boldsymbol{\theta}_l(x_j) &= \mathbf{G}_l(x_{j-1})\boldsymbol{\theta}_l(x_{j-1}) + \mathbf{w}_l(x_j), \end{aligned} \tag{4.10}$$

where $\mathbf{w}_l(x_j) \sim \mathcal{N}(\mathbf{0}, \mathbf{W}_l(s_j))$, $\epsilon_{l,j}$ is an independent white noise for $l = 1, ..., d$ and $j = 1, ..., n_2$. For the Matérn kernel with a half-integer roughness parameter, the terms $\mathbf{G}_l(x_j)$, $\mathbf{W}_l(x_j)$, and $\mathbf{F}$ can be expressed explicitly as a function of $|x_j - x_{j-1}|$ and the range parameter of the kernel. Thus, the forward filtering and backward smoothing algorithm (FFBS) can be applied to compute the likelihood and to make predictions with linear computational operations of the number of observations (see e.g. Chapter 4 in [12] and Chapter 2 in [13] for the FFBS algorithm). The likelihood function and predictive distribution of a GP model having the Matérn kernel with roughness parameters being 1/2 and 5/2 through the FFBS algorithm are implemented in FastGaSP package available at CRAN. The computational complexity of the FFBS algorithm is only $O(n_2)$, with n_2 being the number of observations.

We briefly discuss how to apply the FFBS algorithm to compute terms $\mathbf{L}_l^{-1}\tilde{\mathbf{y}}_l$ and $|\tilde{\boldsymbol{\Sigma}}_l|$ needed in Algorithm 3, for $l = 1, ..., d$. In the FFBS algorithm, the one-step-ahead predictive distribution $(\tilde{y}_{l,j} \mid \tilde{y}_{l,1:j-1}) \sim \mathcal{N}(f_l(x_j), Q_l(x_j))$ can be derived iteratively for $j = 1, ..., n_2$ and for each $l = 1, ..., d$. Closed form expressions of $f_l(x_j)$ and $Q_l(x_j)$ for the Matérn covariance in Equation (4.9) are given in [73]. For $l = 1, ..., d$, we have following expressions for the computational expensive terms in the likelihood function:

$$|\tilde{\boldsymbol{\Sigma}}_l| = \prod_{j=1}^{n_2} Q_l(x_j), \quad \text{and} \quad \mathbf{L}_l^{-1}\tilde{\mathbf{y}}_l = \left(\frac{\tilde{y}_{l,1} - f_{l,1}}{\sqrt{Q_l(x_1)}}, ..., \frac{\tilde{y}_{l,1} - f_{l,n_2}}{\sqrt{Q_l(x_{n_2})}} \right)^T.$$

We use the backward sampling algorithm [13] to sample $\boldsymbol{\theta}_{l,n_2}$ from $p(\boldsymbol{\theta}_{l,n_2} \mid \tilde{\mathbf{y}}_l^{(t)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$ and $\boldsymbol{\theta}_{l,j}$ from $p(\boldsymbol{\theta}_{l,j} \mid \tilde{\mathbf{y}}_l^{(t)}, \theta_{l,j+1}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$ sequentially, for $j = n_2 - 1, ..., 1$. Posterior samples $\mathbf{Z}_l^T = (\mathbf{z}_l(x_1), ..., \mathbf{z}_l(x_{n_2}))^T$ can be obtained by the first entry of the posterior sample $\boldsymbol{\theta}_{l,j}$ from the backward sampling algorithm, for $j = 1, ..., n_2$. Furthermore, for any $n_2 \times 1$ real vector $\mathbf{v}_l$, we have $\mathbf{L}_l \mathbf{v}_l = (f_{l,1} + \sqrt{Q_l(x_1)} v_{l,1}, ..., f_{l,n_2} + \sqrt{Q_l(x_{n_2})} v_{l,n_2})^T$ for $l = 1, ..., d$ and $j = 1, ..., n_2$.

4.3.3 Computational Complexity

Denote $p = p_1 \times p_2$ the total dimension of the inputs $(\mathbf{s}, \mathbf{x})$ and suppose the observational matrix is $n_1 \times n_2$ with irregular missing values, where $n_1 \leq n_2$ and $N = n_1 n_2$. We discuss the computational complexity for three scenarios with $p = 2$ (e.g. spatially correlated data), $p = 3$ (e.g. spatio-temporal data), and $p > 3$ (e.g. functional data).

When $p = 2$, the computational complexity of the GOLF model with the half-integer Matérn kernel is $O(Nd)$. First, we compute the first d eigenvectors of $\boldsymbol{\Sigma}_s$ to obtain $\mathbf{A}_s$, which has $O(n_1^2 d)$ operations (see e.g. Chapter 4.5.5 in [131]). Second, computing the marginal likelihood and sampling the factor processes by the FFBS algorithm only cost $O(n_2 d)$ operations. The largest computational order is from the matrix multiplication $\tilde{\mathbf{Y}}^T = (\mathbf{Y} - \mathbf{M})^T \mathbf{A}_s$, which is at the order of $O(Nd)$.

For $p = 3$, we let $\mathbf{A}_s = \mathbf{A}_{s_1} \otimes \mathbf{A}_{s_2}$, where $\mathbf{A}_{s_1}$ and $\mathbf{A}_{s_2}$ are the first d_1 and d_2 eigenvectors of $n_{1,1} \times n_{1,1}$ matrix $\boldsymbol{\Sigma}_{s_1}$ and $n_{1,2} \times n_{1,2}$ matrix $\boldsymbol{\Sigma}_{s_2}$, respectively, with $n_{1,1} \times n_{1,2} = n_1$ and $\boldsymbol{\Sigma}_{s_1} \otimes \boldsymbol{\Sigma}_{s_2} = \boldsymbol{\Sigma}_s$. Without the loss of generality, assume $d_1 \leq d_2$ and $n_1 \leq n_2$. Let the total number of factor processes be $d = d_1 d_2$. The computational order of the GOLF model with a half-integer Matérn covariance function is $O(n_1 n_2 d_{max})$ where d_{max} is the maximum of d_1 and d_2 (noting this is smaller than $O(n_1 n_2 d)$). To see this, computing the eigendecomposition of $\boldsymbol{\Sigma}_{s_1}$ and $\boldsymbol{\Sigma}_{s_2}$ requires $O(d_1 n_{1,1}^2)$ and $O(d_2 n_{1,2}^2)$

operations, respectively. Second, using the FFBS algorithm to compute the marginal likelihood and to sample factor processes costs $O(dn_2)$ operations. At last, we do NOT directly compute $\mathbf{Y}^T\mathbf{A}_s$ as its computation operations are $O(Nd)$. Instead, we first write the observations as an $n_2 \times n_{1,2} \times n_{1,1}$ array $\mathbf{Y}^T_{ar}$, where the (i, j, k)th entry being the outcome at $(s_{1,i}, s_{2,j}, x_k)$. Then we do a 3-mode matrix product followed by a 2-mode matrix product $\tilde{\mathbf{Y}}^T_{ar} \times_3 \mathbf{A}_{s_1} \times_2 \mathbf{A}_{s_2}$ [132], which has the computation operations $O(n_2 n_1 d_1)$ and $O(n_2 n_{1,2} d)$, respectively. Finally we concatenate the second and third dimensions of $\tilde{\mathbf{Y}}^T_{ar}$ to obtain the $n_2 \times d$ matrix $\tilde{\mathbf{Y}}^T$.

For the case when $p > 3$, there might be two scenarios. In the first scenario, the data are observed in an $n_{1,1} \times n_{1,2} \times ... \times n_{1,k} \times n_2$ tensor with irregular missing values, where $n_{1,1} \times n_{1,2} \times ... \times n_{1,k} = n_1$. In this scenario, the computation will be Nd_{max}, where d_{max} is the maximum of $d_1, ..., d_k$ with similar deduction for the case with $p = 3$. In the second scenario, we have $p_2 > 1$. Examples include emulating a computationally expensive computer output with multivariate output [66, 126]. In this case, the Kalman filter algorithm may not be applied, so the additional computational order is $O(n_2^3)$, when the covariance of the factor process is the same. If the covariance is not the same, we need to additionally compute the inverse of covariance matrices of d multivariate normal distributions, which is at the order of $O(dn_2^3)$.

In sum, when data are from incomplete matrices or arrays, the computation in each step of MCMC algorithm is $O(Nd)$ for $p_2 = 2$ or $O(Nd_{max})$ for $p_2 > 2$, which is much better than $O(N_o^3)$ from directly inverting the covariance matrices. Besides, a few steps in the MCMC algorithm can be computed in parallel, such as FFBS algorithm to compute the product of d marginal densities of projected output and the matrix multiplication $\tilde{\mathbf{Y}}^T = (\mathbf{Y} - \mathbf{M})^T\mathbf{A}_s$, to further reduce the computational complexity.

4.4 Comparison and Connection with Other Related Models

GOLF processes are closely connected to a wide range of approaches on approximating GPs for modeling large correlated data. Model (4.1) is a linear model of coregionalization (LMC) [133], where the factor loading matrix is parameterized by input variables. Another widely used model for multivariate functional data is the semiparametric latent factor model (SLFM) [122], where the factor loading matrix can be estimated by the principal component analysis (PCA) [8]. However, the linear subspace estimated by PCA is equivalent to the maximum marginal likelihood estimator (MMLE) with independent factors ([134]), whereas the latent factors at different input variables are assumed to be correlated. The MMLE of factor loadings with correlated factors was derived in [123], called the generalized probabilistic principal component analysis (GPPCA). Our approach has two distinctions. First, our approach applies to observations with irregular missing values, whereas the observations are required to be matrices in GPPCA. Second, both inputs $\mathbf{s}$ and $\mathbf{x}$ are used for estimation, whereas only the input in latent processes is used in GPPCA and predictions can be more accurate.

To overcome the computational bottleneck of GPs, we project observations on orthogonal coordinates in a GOLF model, as the complexity of computing the likelihood of GPs with Matérn covariances with one dimension input is fast by the continuous-time Kalman Filter.

The computational complexity can be further reduced by only using factor processes with large eigenvalues. The reduced rank approach is used widely in modeling correlated data. For instance, the predictive process by a set of pre-specified knots was studied in [135], and the multiresolution local bisqaure functions were used in [136]. Limitations of the reduced-rank method are studied in [137]. Note that even for the full rank covariance, the computational order of GOLF is much less than $O(N_o^3)$. The primary goal is not to propose a reduced rank model herein, but to reduce the computational complexity of a GP model with a full-rank, flexible covariance function through orthogonal projections.

Many other approximation methods for GPs follow the framework of Vecchia's approximation [49, 50]. Vecchia's approximation is a broad framework that assumes the sparsity of the inverse of Cholesky decomposition of the covariance matrix of the latent processes, where the key is on selecting the order of the latent variables and imposing sensible conditional independence assumptions between variables. GOLF processes with Matérn kernel is closely related to Vecchia's approximation, in the sense that the model can be written as a vector autoregressive model with orthogonal factor loading matrix. Our way of computing likelihood and predictions based on the FFBS algorithm is exact, rather than an approximation to the likelihood function. We compare our approach with a few other methods that fall into the framework of Vecchia's approximation in Section 4.6.1.

4.5 Simulated Studies

We discuss two simulated examples in this section. We first study a simulated example with a small sample size to study the predictive performance and parameter inference between GOLF processes and the exact GP model by directly computing the inversion and determinant of the covariance matrix in the likelihood function. In the second simulated example, we generate observations from separable and nonseparable models to study the predictive performance of GOLF processes with a different number of factors, and with the same or different kernel parameters. For both examples, we implement $J = 100$ experiments in each scenario, and we generate $T = 5,000$ MCMC samples for each method with the first 20% of the samples used as the burn-in samples.

Denote $y^*_{i,j}$ the ith held-out data in the jth simulated experiment in each scenario, for $i = 1, ..., n^*$ and $j = 1, ..., J$. Let $\hat{y}^*_{ij}$ and $CI_{ij}(95\%)$ be the predictive mean and 95% predictive credible interval of the ith held-out data at the jth experiment, respectively. For both simulated examples, we record the root mean square error, the percentage of held-out observations percentage covered in the 95% predictive interval, and the average length of the 95% predictive interval of the jth experiment ($L_{CI_j}(95\%)$):

$$\text{RMSE}_j = \sqrt{\frac{\sum_{i=1}^{N^*}(\hat{y}^*_{ij} - y^*_{ij})^2}{N^*}}, \tag{4.11}$$

$$P_{CI_j}(95\%) = \frac{1}{N^*}\sum_{i=1}^{N^*} 1\{y^*_{ij} \in CI_{ij}(95\%)\}\,, \tag{4.12}$$

$$L_{CI_j}(95\%) = \frac{1}{N^*}\sum_{i=1}^{N^*} \text{length}\{CI_{ij}(95\%)\}\,, \tag{4.13}$$

for $j = 1, ..., J$. We compute average values of these three quantities over $J = 100$ simulations to evaluate each approach. A precise method should have a small average RMSE, $P_{CI}(95\%)$ close to the 95% nominal level, and short predictive interval lengths. Here we only consider the pairwise interval of responses at each coordinate as outputs are univariate on spatial or spatio-temporal domain. Simultaneous credible interval can be used for applications with multivariate responses [138].

Example 3 (GOLF processes and exact GP model) *Data are sampled from a zero-mean separable GP model with two-dimensional inputs at a* 25×25 *regular lattice in* $[0, 1]^2$*. Two missing patterns are considered, where the data are missing at random in the first case, and a disk in the centroid of the lattice is missing in the second case.*

We assume a small sample size in Example 3 because of the computational burden by the exact Gaussian process model. We use the unit-variance covariance matrix parameterized by the exponential kernel and the Matérn kernel with the roughness parameter

	Missing value		GOLF			Exact GP model			Difference		
Kernel	Percentage	Pattern	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$	ΔRMSE	ΔL	ΔU
Matérn	50%	random	0.106	0.954	0.425	0.106	0.952	0.423	0.002	0.006	0.006
	20%	random	0.103	0.952	0.410	0.103	0.952	0.411	0.001	0.007	0.007
	20%	disk	0.108	0.909	0.430	0.108	0.913	0.431	0.005	0.008	0.009
Exp	50%	random	0.129	0.955	0.518	0.128	0.953	0.513	0.005	0.009	0.008
	20%	random	0.120	0.947	0.472	0.120	0.948	0.471	0.003	0.009	0.009
	20%	disk	0.156	0.941	0.602	0.154	0.946	0.605	0.013	0.019	0.019

Table 4.2: Comparison between the exact GP model and GOLF processes. $J = 100$ simulated experiments are conducted for each scenario. ΔRMSE$= \frac{1}{J}\sum_{j=1}^{J}$ ΔRMSE$_j$ measures the average L_2 distance by the two methods, where ΔRMSE$_j = (\frac{1}{N^*}\sum_{i=1}^{N^*}(\hat{y}^*_{ij,GOLF} - \hat{y}^*_{ij,GP})^2)^{1/2}$ with $\hat{y}^*_{ij,GOLF}$ and $\hat{y}^*_{ij,GP}$ denote the predictive mean by GOLF processes and exact GP model, respectively. ΔL and ΔU measure the average absolute difference between the lower bound and upper bound of 95% predictive intervals of the GOLF processes and the exact GP model, respectively.

being 2.5 in Equation (4.9) to generate the data. The range parameters of Matérn kernel are chosen as $\gamma_0 = 1$ and $\gamma_1 = ... = \gamma_d = 1/3$. The range parameters of the exponential kernel are chosen to be $\gamma_0 = 4$ and $\gamma_1 = ... = \gamma_d = 1$. All the range parameters, the variance of the kernel, and noise are estimated by each method based on the MCMC algorithm.

We compare GOLF processes and the exact GP model where the inverse and determinant of the covariance matrix are directly computed. Both models use the same prior and proposal distribution in the MCMC algorithm to sample the kernel parameters. Table 4.2 gives the predictive performance of both methods for three scenarios, where 50% and 20% of the output are missing at random in the first two scenarios, and approximately 20% of the output is missing in a disk in the centroid of the lattice in the third scenario. Graphs of the observed data, full data, predictions, and trace plots of the posterior samples in one simulation are given in the appendix.

As shown in Table 4.2, both methods have accurate predictions and uncertainty assessment for all scenarios. Out-of-sample RMSE for predicting the held out observations is close to 0.1, the standard deviation of the noise. The 95% predictive confidence in-

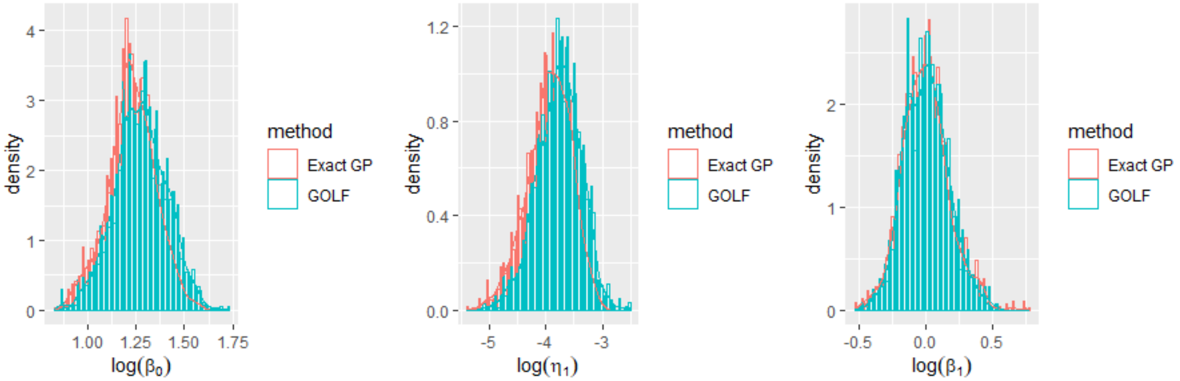


Figure 4.2: The histogram of posterior samples of the logarithm of the inverse range parameters and nugget parameters in one simulation of Example 3, where the data are generated using the Matérn kernel in (4.9) with 50% of the values missing at random.

tervals cover around 95% of the held-out observations, and the average length of the predictive confidence interval is small. Predictions of both methods are more precise for the cases when the data are missing at random than the ones when a disk of output is missing in the centroid of the lattice, as the estimated correlation between the held-out test output and nearby observations are relatively accurate.

For Example 3, note that GOLF processes and the exact GP model are the same with two different computational strategies. For GOLF processes, we sample the missing values to use the fast computational strategy, whereas the inverse and determinant of the covariance matrix are computed in the exact GP model directly. Therefore, the two different strategies have significantly different computational operations. The computational operations of GOLF processes is $O(Nd)$ with $N = n_1 \times n_2$ ($d = n_1$ in Example 3), whereas the computational operations of the exact GP model is $O(N_o^3)$, where N_o is the number of observations. Thus, GOLF processes are computationally feasible for a large data set. On the other hand, the difference in predictions and uncertainty assessment between the exact GP model and GOLF is small (last three columns in Table 4.2), since we do not make any approximation in computing GOLF processes.

Figure 4.2 shows the histogram of the 4000 after burn-in posterior samples from the

GOLF processes and exact GP model in one simulation of Example 3. The posterior samples of the two methods are close to each other. The difference becomes even smaller when we increase the number of MCMC samples.

Example 4 (GOLF processes: different factor numbers & kernel parameters) *The data are sampled from two scenarios with two-dimensional inputs being a* 100×100 *lattice in* $[0,1]^2$*. In the first scenario, the range parameters of the kernel of each factor process are the same, whereas these parameters are chosen to be different in the second scenario. In both scenarios, a disk of output in the centroid of the lattice is masked out for testing, corresponding to approximately* 20% *of the total number of data. We use* $d = 30$ *(low-rank) and* $d = 100$ *(full-rank) factors to generate the data. We test GOLF processes with a different number of factors, same or different range parameters.*

In Example 4, the factor processes are assumed to have the Matérn kernel in Equation (4.9) and unit variance. The kernel parameter is shared in the first scenario, where $\gamma_0 = 1/4$ and $\gamma_l = 1/2$, and in the second scenario $\gamma_0 = 1/3$ and $\gamma_l = 1/l$, for $l = 1, ..., d$. We estimate these parameters through the posterior samples from the MCMC algorithm.

Predictive performance of different approaches for data simulated by $d = 30$ latent processes are graphed in Figure 4.3. In the first row of the panels, since data are simulated by GOLF processes with different kernel parameters, nonseparable GOLF processes have smaller predictive RMSE and a shorter interval that covers almost 95% of the data. In the second row of the panels, GOLF processes with the same kernel parameter seem to be slightly better, as the true factor process has the same kernel parameter. The difference between the two methods in the second row is smaller, as the GOLF model with a separable kernel is a special case of the one with different kernel parameters.

From Figure 4.3, we found that when we use $d = 20$ factor processes or more, the predictive results seem to be similar, as the data are simulated using $d = 30$ factor

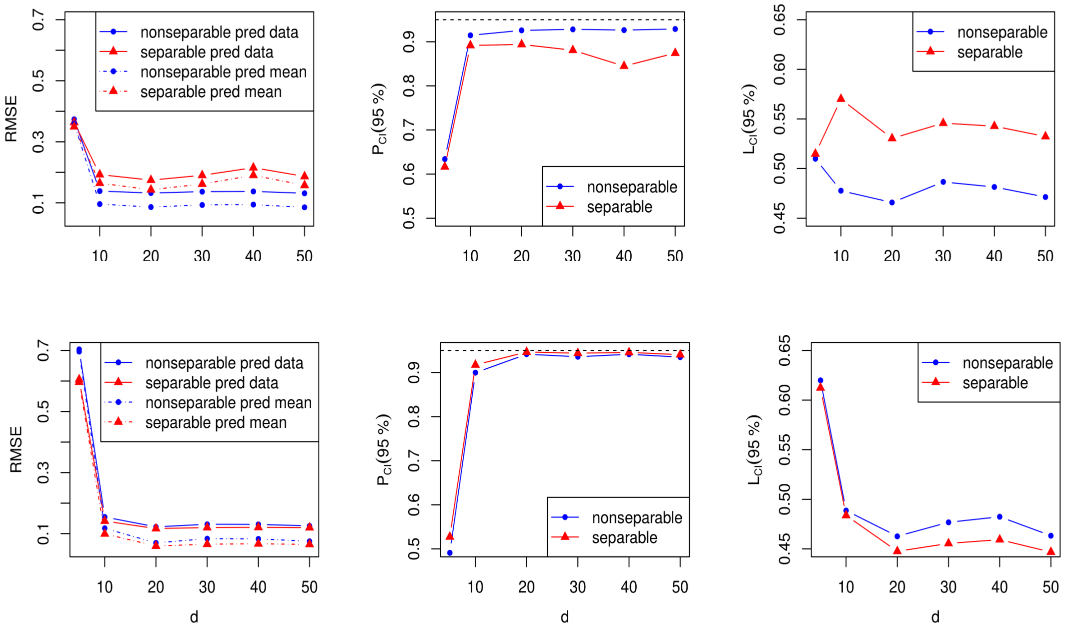


Figure 4.3: The predictive performance of GOLF process with $d = 5, 10, 20, 30, 40$ and 50 factors for Example 4. in the first row of panels, kernel parameters are different in simulating the data, whereas the parameters are the same in for simulation in the second row of panels. Blue curves and red curves denote the GOLF processes with the different kernel parameters and the same kernel parameter, respectively. In the left panels, the solid curves denote the RMSE for predicting the (noisy) observations, and the dashed curve denotes the RMSE for predicting the mean of the observations. The proportions of observations covered in the 95% predictive interval and the average length of the predictive interval are graphed in the middle and right panels, respectively.

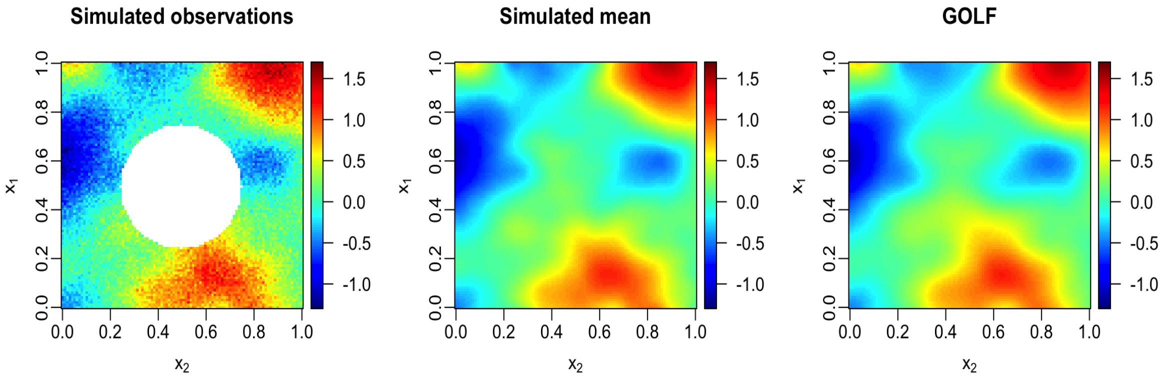


Figure 4.4: The left figure shows the observed data in one simulation of Example 4, where a disk of observations is missing. The middle figure contains the mean of the data and the right figure is the prediction from GOLF process.

processes. The way of selecting the number of factors is currently ad-hoc. One may select the number of factors to ensure a large proportion of the variance explained by the sum of the eigenvalues of the correlation matrix $\mathbf{R}_s$. This simulation suggests that using more factors may be better in prediction than using very few factors.

In Figure 4.4, we graph the simulated observations, simulated mean, and the prediction from the GOLF model with $d = 30$ in one simulation. Predictions look reasonably accurate. Results when the data are generated by a full rank kernel ($d = 100$) are provided in Figure C.3 of the appendix. Results are very similar to Figure 4.3.

4.6 Real Applications

4.6.1 Predicting Large Spatial Data on an Incomplete Lattice

We compare GOLF processes with different approaches on predicting the missing temperature values in [125]. This data set contains daytime land surface temperatures on August 4, 2016, at 300×500 spatial grids with the latitude and longitude ranging from 34.30 to 37.07, and from -95.91 to -91.28, respectively. The complete data set consists

Methods	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$	Run time (mins)
FRK	3.16	0.77	6.09	3.53
Gapfill	1.86	0.35	1.44	6.98
GOLF	1.46	0.92	4.95	48.6
LAGP	2.07	0.84	5.70	3.76
LatticeKrig	1.68	0.963	6.58	214.25
MRA	1.85	0.92	5.54	4.99
NNGP	1.64	0.95	5.84	1.14
Partition	1.80	0.82	4.56	827.37
SPDE	1.55	0.97	7.87	34.8

Table 4.3: Comparison for the dataset in [125]. The standard deviation of observations is 4.07. For each method, we compute RMSE, $P_{CI}(95\%)$ and $L_{CI}(95\%)$ defined in Equations (4.11)-(4.13). A satisfying method should have small RMSE and small $L_{CI}(95\%)$ and $P_{CI}(95\%)$ closed to be 95% nominal level. We compare the fixed rank kriging (FRK) ([136]), the Gapfill method ([139]), GOLF processes, the local approximate Gaussian processes (LAGP) ([56]), the lattice kriging (LatticeKrig) ([140]), the multiresolution approximation (MRA) ([55]), the nearest neighbor Gaussian processes (NNGP) ([54]), the spatial partitioning (Partition) ([141]), and stochastic partial differential equations (SPDE) ([52]).

of 148,309 observations with $1,791$ missing values due to cloud cover. The training data (plotted in the left panel in Figure 4.1) consists of 105,569 observations, whereas 42,740 observations were held out as the test data. Training observations and full observations are graphed in the upper panel in Figure 4.5.

We define GOLF processes on this dataset with s being latitude and x being longitude. Since areas with higher latitude typically have lower temperature on average, we assume a mean parameter for each latitude value, i.e. $\mathbf{M} = (\mathbf{H}_2\mathbf{B}_2)^T$, where $\mathbf{H}_2 = \mathbf{1}_{n_2}$ and $\mathbf{B}_2 = (b_{2,1}, ..., b_{2,n_1})^T$. We let $d = n_1/2$ and use exponential kernels with distinct variances and range parameters sampled from the marginal posterior distribution for GOLF processes. We compute $M = 6000$ posterior samples where the first 20% were used as the burn-in samples. Results of longer MCMC chains and different initial values of the parameters are given in the appendix.

In [125], 12 groups of researchers across the globe implemented their methods to predict missing temperature values for competition. Among this cohort of researchers are authors that conjured up some of the most popular methods for large spatially correlated data. Other than GOLF processes, we implement 8 of 12 approaches based on the code provided in [125]. We could not implement the other 4 approaches due to memory limitation of the computing facility or unavailability of the code. All computations are operated on a 3.60GHz 8 cores Intel i9 processor with 32 GB of RAM on a macOS Mojave operating system.

The predictive performance of different approaches is recorded in Table 4.3. Most of the results are consistent with what is shown in [125], whereas small differences remain for those requiring random starts or stochastic algorithms. E.g., 5 implementation of the SPDE method gives different RMSE ranging from 1.55 to 1.88. Besides, running time of some methods are slightly different. For SPDE and LatticeKrig, for instance, it takes 35 mins and 214 mins to run in our system, respectively, whereas it takes 138 mins and 78 mins to run in [125], respectively.

We acknowledge that held-out observations were not released in [125], adding difficulty for model specification. The good performance of the GOLF model may be explained by two reasons. First, different mean parameters are assumed at each latitude, which is more flexible to capture information from a large number of observations. Second, we assume different range and variance parameters of the factor processes, which are more flexible than the separable or isotropic kernel functions.

The 95% predictive interval of the GOLF model is the shortest, and it covers around 92% of the held out test data, as shown in Table 4.3. In appendix, we provide diagnostic plots of the fitted values from the GOLF model and predictive performance based on several configurations, including $40,000$ MCMC samples and different initial parameters. The predictive performance of the GOLF model at different configurations is similar. Besides, the computational time of GOLF per one MCMC iteration is around $0.49s$ for

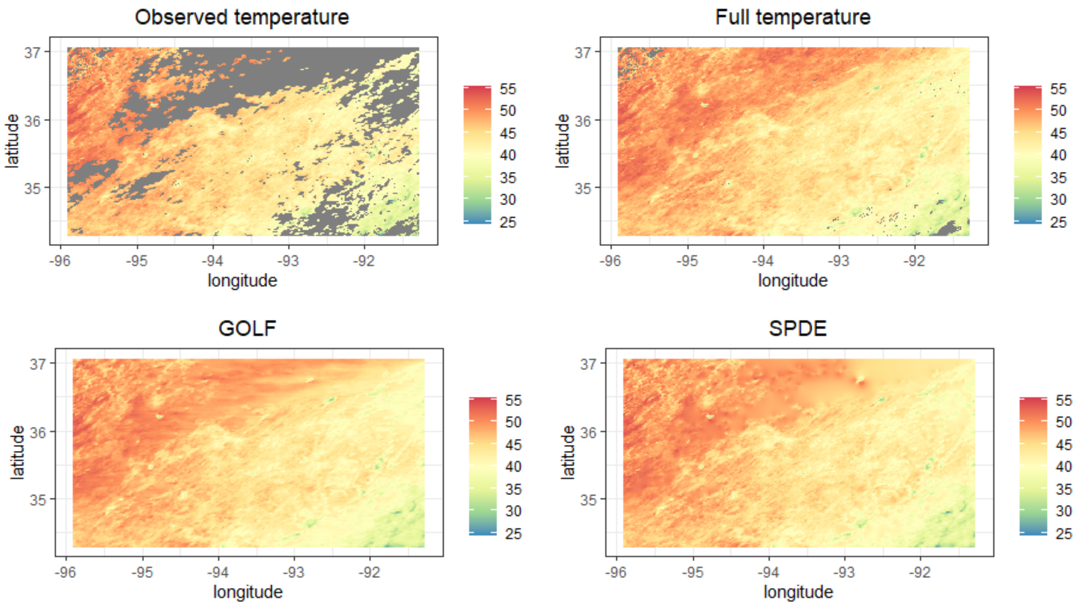


Figure 4.5: The top panels show the observed temperature and full temperature, respectively, where the gray area contains unobservable points. The bottom panel are the predictions from two methods, GOLF and SPDE, respectively.

this example, which is comparable to NNGP (0.53s) and faster than MRA (3.29s) for one iteration. The posterior sampling obtained here provided uncertainty quantification of model parameters, whereas most of the methods provided in Table 4.3 only provide a point estimator of the parameters. Future works are needed to reduce the number of iterations in GOLF to achieve a similar level of predictive accuracy.

The predictive mean of the GOLF processes and SPDE are graphed in the middle panel and right panel in Figure 4.5, respectively. Predictions from the GOLF processes are more accurate for predicting temperatures in areas with high latitude, possibly due to flexible mean parameters estimated from data. Both methods seem to be slightly oversmoothing. Yet predicting the missing values of this data set is challenging, as the observations are missing in spatial blocks. Both methods seem precise in prediction.

4.6.2 Analysis of Large Spatio-temporal Dataset

We consider the monthly gridded temperature anomalies from U.S. National Oceanic and Atmospheric Administration (NOAA) [1]. The data set contains the average air and marine temperate anomalies at 5 degrees longitude-latitude grids with respect to 1981-2010 base period. R code and examples to load NOAA gridded data can be found in [142]. We compare the predictive performance using the data from Jan 1999 to Dec 2018. For each month, we observe the temperature anomalies at $n_1 = 36 \times 28$ spatial grids with longitude ranging from 182.5 to 357.5 and with latitude ranging from -62.5 to 72.5, respectively. There are $11,122$ missing data, leaving the total number of observations to be $230,798$. We held out 50% randomly sampled temperature anomalies as the missing data, and the rest 50% is used as training data (i.e., $n = n^* = 115,399$). Predicting the missing values in this scenario is more difficult than the example in [123], where the data are missing in a set of locations over the same months.

We fit the GOLF processes with the covariance of each spatial coordinate modeled by the Matérn covariance, and the factors processes are defined on the temporal input with different kernel parameters. Due to computational limitation, we let the number of factors be $d = 0.75^2 n_1 = 567$ and assume the factor loadings to be a Kronecker product of the first three-quarters of the eigenvectors of the sub-covariance matrices for longitude and latitude. Although we have a large number of factors, the computational complexity is $O(Nd_{max})$ with $d_{max} = 0.75 \times 36 = 48$ rather than $O(Nd_1 d_2)$ by the mode multiplication of tensor (see Section 4.3.3 for the discussion). We assume the coefficients of the intercept and linear coefficients are different at each location, i.e. $\mathbf{M} = (\mathbf{H}_2 \mathbf{B}_2)^T$ where $\mathbf{H}_2 = [\mathbf{1}_{n_2}, \mathbf{x}]$, with $\mathbf{x}$ being 240 months and $\mathbf{B}_2$ being a matrix of $2 \times n_1$ coefficients. We use $M = 3000$ MCMC samples with the first 20% as the burn-in samples, as posterior

[1] `ftp://ftp.ncdc.noaa.gov/pub/data/noaaglobaltemp/operational`

Methods	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$	Run time (mins)
FRK	0.846	0.967	3.92	29.4
GOLF	0.325	0.942	1.08	43.9
LAGP	0.695	0.951	1.80	6.18
Spatial model 1	0.365	0.928	2.09	26.5
Spatial model 2	0.348	0.928	2.02	42.7

Table 4.4: Predictive performance of different approaches for the NOAA monthly gridded temperature dataset. The standard deviation of the outcomes in this dataset is 0.940. Results of the FRK, GOLF, and LAGP are given in the first to the third rows. For the results in the fourth and fifth rows, spatial models were fitted using the RobustGaSP package with one initial value and two initial values of the range and nugget parameters for finding their marginal posterior mode, respectively.

samples converge at a small number of iterations in this example.

In Table 4.4, we compare the GOLF processes with a few other spatial and spatio-temporal methods for the NOAA dataset. We fit two spatial models separately for each month using the RobustGaSP package available on CRAN. Also implemented are FRK and LAGP based on their packages [143, 144].

As shown in Table 4.4, GOLF processes have the smallest predictive RMSE and the shortest predictive interval that covers around 94% of the held-out output. Since the temporal input is not used, it is not surprising that the RMSE and the length of the predictive interval of the two spatial models are larger than the ones by GOLF processes. If we include the temporal inputs, the computation cost is too large for inverting the covariance matrix directly. FRK and LAGP also seem to have a larger predictive error, though both the spatial and temporal inputs are used in these methods.

Predictions from GOLF processes are more accurate due to three reasons. First, we can compute the model with a large number of factors efficiently, and no further approximation of the likelihood function is required. Second, mean and trend parameters at each location are different, making the model flexible to capture the dynamic trend of temperature values at different locations. Finally, Latent factor processes have different

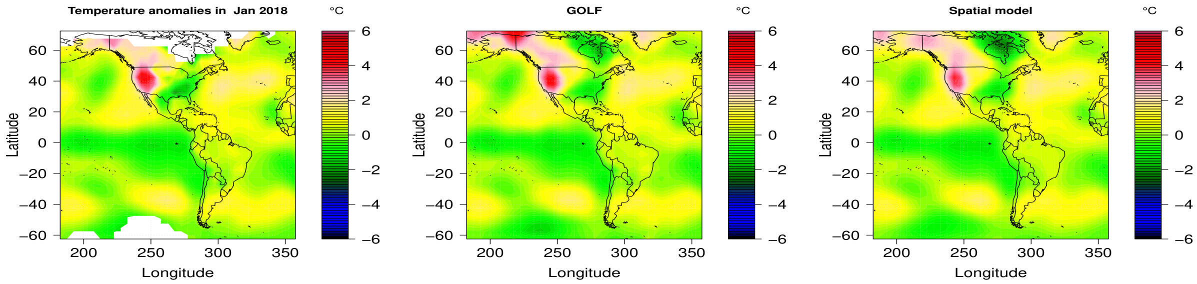


Figure 4.6: Full temperature anomalies in Jan 2018, predictions by the GOLF model and the spatial model by RobustGaSP package are shown in left, middle and right panels, respectively.

kernel parameters that fit diverse smoothness levels of projected observations.

In Figure 4.6, we graph the full temperature anomalies in Jan 2018, predictions from the GOLF and spatial GP model by RobustGaSP package. 50% of the observation in the left panel are held out for testing. Both models seem to be accurate. Since the temporal coordinate is used in prediction, the predictive error by GOLF processes is smaller.

Chapter 5

Conclusions and Future Research Directions

In Chapter 2, we introduce the Sequential Kalman Filter (SKF) for online changepoint detection for data with temporal correlations. The temporal correlation between each time point is modeled in SKF and the computational cost is dramatically reduced without approximating the likelihood function. Furthermore, we developed a new approach that integrates high-dimensional covariates and massive outcomes for detecting COVID-19 infection from a large longitudinal dataset of dialysis patients, overcoming the challenge of modeling massive longitudinal covariates with a large proportion of missingness. The new approach substantially improves detection accuracy compared to conventional classification and other online changepoint detection approaches.

Chapter 3 introduces a real-time epidemic model that tracks daily COVID-19 transmission across over 3,000 U.S. counties. It offers both short-term and long-term death toll predictions, while the uncertainties in the predictions are quantified by the GP model. The model also estimates the Probability of Contracting (PoC) COVID-19, reflecting the daily average infection risk for a healthy individual at the county level. Our analysis suggests that effective strategies aiming at shortening the infectious period significantly reduce infections and deaths, particularly in areas with an effective reproduction number close to 1. This underscores the need for integrating testing with other measures like social distancing and mask-wearing to decrease transmission rates. Additionally, we have created a dynamic county-level map, a valuable tool for local officials to formulate appropriate policies for the public to better assess daily COVID-19 risk.

In Chapter 4, we introduce the GOLF processes as a computationally feasible approach to model large incomplete lattice observations. For GPs with a product covariance function or LMC with orthogonal latent factor loadings, there are two intriguing properties. First, the likelihood can be decomposed into a product of multivariate normal densities. Second, the prior independence of factor processes leads to the posterior independence of factor processes. These two properties allow one to reduce the computational burden of GPs on incomplete lattice observations without approximating the likelihood function. Further computational reduction can be made by reducing the number of factors as well. Besides, we have introduced a flexible way to model the mean function and the closed-form marginal likelihood is derived to alleviate the identifiability issue. Moreover, we have developed an MCMC algorithm for Bayesian inference for large incomplete matrices of spatial and spatio-temporal data.

Another key direction not covered in previous chapters is to quantify the uncertainty for massive data with high dimensional input. As the dimensions of input variables increase, a substantial amount of data is needed to fill the large parameter space to obtain accurate predictions and reliable uncertainty quantification. In practice, however, input variables of concern can be highly correlated, which leads to small intrinsic dimensions for the input. This characteristic offers possibilities for employing surrogate models to approximate the behavior of such systems for problems of high dimensional input. Here we discuss a surrogate modeling approach based on the Parallel Partial Gaussian

Process (PPGP) model introduced in [62], for problems with a high-dimensional input. The PPGP assumes outputs at each coordinate to have different mean and variance parameters, whereas the correlation parameters are shared across each output coordinate. Despite its simplicity, PPGP has shown remarkable performance in various fields, including quantifying volcanic hazard [62, 60], emulating expensive molecular simulations [22, 145], and power system simulations [28], often outperforms more complex alternatives. In the following section, we will discuss the ideas behind the PPGP model and its application within the context of power systems.

5.1 Parallel Partial Gaussian Process model for computer model emulation with high-dimensional inputs

In this section, we discuss the application of the PPGP model on data with high-dimensional input, using large-scale power systems as an example. The integration of renewable energy sources and highly flexible demands has introduced significant uncertainties into power system operations [146], challenging traditional methods for transient stability analysis that do not accommodate these uncertainties, leading to potential overestimation of system risks [147]. We first introduce the simulation models and then discuss the PPGP emulator for predicting the simulation with high-dimensional inputs and outputs.

5.1.1 Power System Dynamic Modeling

Generally, power system dynamics are governed by the following differential and algebraic equations (DAEs) [26]:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{u}, \boldsymbol{\xi}), \tag{5.1a}$$

$$0 = g(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\xi}), \tag{5.1b}$$

where $\boldsymbol{x}$ denotes the state variables, such as rotor angles and speeds and $\boldsymbol{y}$ represents the algebraic variables, such as voltage magnitudes and phase angles; $\boldsymbol{\xi}$ collects all the uncertain resources, including flexible loads and intermittent renewable generations. Equation (5.1a) consists of differential equations that represent the controls and dynamics of synchronous generators and loads. Equation (5.1b) refers to algebraic equations regarding network and static components. By utilizing the structural-preserve approach, Equation (5.1) can be reformulated into the following differential equations:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{g}^{-1}(\boldsymbol{x}, \boldsymbol{\xi}), \boldsymbol{u}, \boldsymbol{\xi}), \tag{5.2}$$

where the algebraic variables $\boldsymbol{y}$ is represented by $\boldsymbol{g}^{-1}(\boldsymbol{x}, \boldsymbol{\xi})$. This can be further linearized following the Euler-based explicit scheme:

$$\boldsymbol{x}(t) = \boldsymbol{x}(t-1) + \Delta t \times \boldsymbol{f}(\boldsymbol{x}(t-1), \boldsymbol{g}^{-1}(\boldsymbol{x}(t-1), \boldsymbol{\xi}), \boldsymbol{u}, \boldsymbol{\xi}), \tag{5.3}$$

where Δt is the time step in the simulation process. For time domain simulation, the desired state variables are obtained step-by-step. To investigate the impact of uncertain resources on dynamic responses, the relationship between $\boldsymbol{\xi}$ and state variables are rewritten into the following compact form:

$$\boldsymbol{x} = \mathcal{M}(\boldsymbol{\xi}, t), \tag{5.4}$$

where in this paper $\boldsymbol{\xi}$ refers to uncertainties from loads and PVs and $\boldsymbol{x}$ contains rotor angle; $\mathcal{M}$ represents the simulator for dynamic responses. In the case that the uncertain input is unchanged during the dynamic simulation process ($\boldsymbol{\xi} \equiv \boldsymbol{\xi}_0$), the rotor angle at each time point can be viewed as a function of the initial condition, which means:

$$\mathrm{x}_t = \mathcal{M}_{t-1}(\mathrm{x}_{t-1}, \boldsymbol{\xi}_0) = \mathcal{M}_{t-1}(\mathcal{M}_{t-2} \cdots \mathcal{M}_0(\mathrm{x}_0, \boldsymbol{\xi}_0)), \tag{5.5}$$

Combining Equations (5.2)-(5.5), we build a surrogate model with the output being rotor angles over the whole simulation time.

5.1.2 PPGP Emulator for Large-scale Probabilistic TSA

We develop surrogate models to emulate the simulator in Equation (5.4) by learning the dynamic responses at many time points and rotor angles as in Equation (5.5) to accelerate the computation. For each input $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_p]$, the output rotor angle is a $T \times n_g$ matrix $\boldsymbol{x}^{mat}(\boldsymbol{\xi})$ with (t, g) entry being $\boldsymbol{x}_{t,g}(\boldsymbol{\xi})$ for time points $t = 1, ..., T$ and rotor $g = 1, ..., n_g$. We vectorize the rotor angle swing matrix to define a Tn_g-dimensional vector $\boldsymbol{x}(\boldsymbol{\xi}) = \mathrm{Vec}[\boldsymbol{x}^{mat}(\boldsymbol{\xi})]$ for any given input $\boldsymbol{\xi}$. In our numerical study, the dimensions of output and inputs on the Texas 2000-bus system [27] are $s = T \times n_g = 120 \times 282 = 33,840$ and $p = 2,352$, respectively, which are both large. To address the computational challenge of emulating functions with both large input and output dimensions, we extend the PPGP surrogate model [62] to emulate the simulator $\mathcal{M}$ that produces a high dimensional rotor angle swing vector for each input. Here we sample the input from distributions that mimic the real applications. Typical input distribution for power system simulations includes copula [148], which contains correlations between each input coordinate. This is intrinsically different from the traditional “space-filling” designs, such as the Latin hypercube design, as the aim is not to fill the entire input region but a subregion that represents the real scenarios. This choice is critical to approximate practical scenarios and reduce the high-dimensional input space that is not easily filled by the Latin hypercube design. Second, we use an isotropic kernel for faster and more robust optimization in high-dimensional input spaces, as opposed to the product kernel used in [62].

The system response for dynamic systems is time-dependent, leading to the evolution of statistics as a function of time. As a result, the PPGP emulator at each of the ith output $x_i(\boldsymbol{\xi})$, for $i=1,...,s$, assumes distinct mean parameter μ_i and variance parameter σ_i^2, which makes it flexible to capture variability in different rotors. On the other hand, the covariance parameters between two functions are assumed to be shared across each output in PPGP for computational purposes. Furthermore, we also include a small noise parameter to account for small numerical solution errors. Putting together, for the ith output rotor angle swing from Equation (5.4), the PPGP model follows

$$x_i(\boldsymbol{\xi}) = \mathcal{M}_i(\boldsymbol{\xi}) + \epsilon_i, \tag{5.6}$$

where ϵ_i is a Gaussian noise with variance $\sigma_i^2\eta$ and $\mathcal{M}_i(\boldsymbol{\xi})$ follows a stationary Gaussian process with mean μ_i and covariance $\sigma_i^2 c(\boldsymbol{\xi}, \boldsymbol{\xi}'; \gamma)$ with kernel range parameter γ and variance parameter σ_i^2. Integrating out the latent Gaussian process $\mathcal{M}_i(\boldsymbol{\xi})$, any marginal distribution of the ith output at any set of inputs $\{\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_n\}$ follows a multivariate normal (MN) distribution

$$(x_i(\boldsymbol{\xi}_1), ..., x_i(\boldsymbol{\xi}_n))^T \mid \mu_i, \sigma_i^2, \eta, \mathbf{R} \sim \mathcal{MN}(\mu_i \mathbf{1}_n, \sigma_i^2(\mathbf{R} + \eta \mathbf{I}_n)), \tag{5.7}$$

where $\sigma_i^2\mathbf{R}$ is a covariance matrix with the (j, j')th entry being $\sigma_i^2 R_{j,j'} = \sigma_i^2 c(\boldsymbol{\xi}_j, \boldsymbol{\xi}_{j'}; \gamma)$ and $\mathbf{I}_n$ is an $n \times n$ identity matrix. Here $\sigma_i^2 c(\boldsymbol{\xi}, \boldsymbol{\xi}'; \gamma)$ denotes a covariance function with the variance parameter σ_i^2 and kernel parameter γ. As the input dimension (p) is large, we use an isotropic kernel, meaning the covariance matrix is a function of the inputs' Euclidean distance $d = ||\boldsymbol{\xi} - \boldsymbol{\xi}'||$ for any inputs $\boldsymbol{\xi}$ and $\boldsymbol{\xi}'$ and Matérn covariance with roughness parameters $\alpha = 2.5$ in Equation (1.9).

The total number of parameters contain mean parameters $\boldsymbol{\mu} = (\mu_1, ..., \mu_s)^T$, variance parameters $\boldsymbol{\sigma}^2 = (\sigma_1^2, ..., \sigma_s^2)^T$, range and nugget parameter (γ, η). We assume an objective prior distribution of the parameters [62]:

$$\pi(\mu_1, \ldots, \mu_s, \sigma_1^2, \ldots, \sigma_s^2, \gamma, \eta) \propto \frac{\pi(\gamma, \eta)}{\prod_{i=1}^s \sigma_i^2}, \tag{5.8}$$

where $\pi(\gamma, \eta)$ denotes the prior for the range and nugget parameters, which will be discussed soon.

Assume we have n simulation runs on design inputs $\{\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_n\}$ which produces $n \times s$ output matrix $\boldsymbol{x}^{\mathcal{D}} = [\boldsymbol{x}_1^{\mathcal{D}}, ..., \boldsymbol{x}_s^{\mathcal{D}}]$, where $\boldsymbol{x}_i^{\mathcal{D}} = [x_i(\boldsymbol{\xi}_1), \ldots, x_i(\boldsymbol{\xi}_n)]^T$, for $i = 1, ..., s$. The key is that the large number of mean and variance parameters can be integrated out explicitly. Given a set of the range and nugget parameters, the predictive distribution of the ith output for any new test input $\boldsymbol{\xi}^*$ follows:

$$p(x_i(\boldsymbol{\xi}^*) \mid \boldsymbol{x}^{\mathcal{D}}, \gamma, \eta) = \int p(x_i(\boldsymbol{\xi}^*) \mid \boldsymbol{x}^{\mathcal{D}}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2, \gamma, \eta) d\boldsymbol{\mu} d\boldsymbol{\sigma}^2, \tag{5.9}$$

and the resulting predictive distribution of the left-hand side follows a t-distribution with

$n-1$ degrees of freedom:

$$x_i\left(\boldsymbol{\xi}^*\right) \mid \boldsymbol{x}^{\mathcal{D}}, \gamma, \eta \sim \mathcal{T}\left(\hat{x}_i\left(\boldsymbol{\xi}^*\right), \hat{\sigma}_i^2 c^{**}, n-1\right), \tag{5.10}$$

where the predictive mean and scale parameters follow

$$\hat{x}_i\left(\boldsymbol{\xi}^*\right) = \hat{\mu}_i + \mathbf{r}^T\left(\boldsymbol{\xi}^*\right)\mathbf{K}^{-1}\left(\boldsymbol{x}_i^{\mathcal{D}} - \hat{\mu}_i \mathbf{1}_n\right), \tag{5.11}$$

$$\hat{\sigma}_i^2 = \frac{1}{n-1}\left(\boldsymbol{x}_i^{\mathcal{D}} - \hat{\mu}_i \mathbf{1}_n\right)^T \mathbf{K}^{-1}\left(\boldsymbol{x}_i^{\mathcal{D}} - \hat{\mu}_i \mathbf{1}_n\right), \tag{5.12}$$

$$\begin{aligned} c^{**} &= 1 + \eta - \mathbf{r}^T\left(\boldsymbol{\xi}^*\right)\mathbf{K}^{-1}\mathbf{r}\left(\boldsymbol{\xi}^*\right) \\ &\quad + \left(\mathbf{1}_n^T \mathbf{K}^{-1} \mathbf{1}_n\right)^{-1}\left(1 - \mathbf{1}_n^T \mathbf{K}^{-1}\mathbf{r}\left(\boldsymbol{\xi}^*\right)\right)^2, \end{aligned} \tag{5.13}$$

with $\mathbf{K} = \mathbf{R} + \eta \mathbf{I}_n$ and $\hat{\mu}_i = \left(\mathbf{1}_n^T \mathbf{K}^{-1} \mathbf{1}_n\right)^{-1} \mathbf{1}_n^T \mathbf{K}^{-1} \boldsymbol{x}_i^{\mathcal{D}}$ being the generalized least squares estimator for μ_i, for $i = 1, ..., s$, $\mathbf{1}_n = (1, \ldots, 1)^T$ being an n-dimensional vector of ones and $\mathbf{r}\left(\boldsymbol{\xi}^*\right) = \left(c\left(d_1^*; \gamma\right), \ldots, c\left(d_n^*; \gamma\right)\right)^T$ with $d_j^* = ||\boldsymbol{\xi}^* - \boldsymbol{\xi}_j||$ for $j = 1, ..., n$.

The only two parameters that require numerical optimization are the range and nugget parameters (γ, η). Directly estimating these parameters by the maximum likelihood estimator could lead to unstable estimation, as the correlation matrix becomes close to an identity matrix and an all-one matrix [69]. A way to avoid this problem is to transform the range parameter to inverse range parameter $\beta = 1/\gamma$, and estimate (β, η) by the maximum marginal posterior mode estimation with a jointly robust prior [70]:

$$(\hat{\beta}, \hat{\eta}) = \underset{\beta, \eta}{\operatorname{argmax}} \log(p(\boldsymbol{x}^{\mathcal{D}} | \beta, \eta) \pi^{JR}(\beta, \eta)). \tag{5.14}$$

Here $p(\boldsymbol{x}^{\mathcal{D}} \mid \beta, \eta)$ is the marginal likelihood of (β, η) after integrating out the mean and variance parameters, which has a closed form expression [62]:

$$p(\boldsymbol{x}^{\mathcal{D}} \mid \beta, \eta) \propto |\mathbf{K}|^{-\frac{s}{2}} |\mathbf{1}_n^T \mathbf{K}^{-1} \mathbf{1}_n|^{-\frac{s}{2}} \prod_{i=1}^{s} (S_i^2)^{-\frac{n-1}{2}}, \tag{5.15}$$

where $S_i^2 = (\mathbf{x}_i^{\mathcal{D}})^T \mathbf{Q} \mathbf{x}_i^{\mathcal{D}}$ with $\mathbf{Q} = \mathbf{K}^{-1}\mathbf{P}$ and $\mathbf{P} = \mathbf{I}_n - \mathbf{1}_n \left(\mathbf{1}_n^T \mathbf{K}^{-1} \mathbf{1}_n\right)^{-1} \mathbf{1}_n^T \mathbf{K}^{-1}$, and $\pi^{JR}(\beta, \eta)$ is a jointly robust (JR) prior:

$$\pi^{JR}(\beta, \eta) \propto (C\beta + \eta)^a \exp\left(-bC\beta + \eta\right), \tag{5.16}$$

where $a > -2$, $b > 0$ and $C > 0$ are prior parameters. The default choice of a, b and C are discussed in [70]. The JR prior has a closed-form expression and approximates the reference prior [69], which helps avoid unstable estimation from the maximum profile likelihood and maximum marginal likelihood estimation. The low-storage quasi-Newton optimization method (L-BFGS) [67] is used for numerical estimation of the range and nugget parameters in Equation (5.14). After parameter estimation, we can compute the predictive distribution for predicting the responses of any new input using Equation (5.10).

5.1.3 Numerical Results

The proposed PPGP method for probabilistic TSA is verified on the Texas 2000-bus system [27]. The synchronous generators are modeled as the two-axis model according to [149]. A three-phase fault is applied at but 793 at 0.5s and is cleared after 5 cycles by opening the line 793-823. The inputs consist of uncertain loads and renewable generations. For verification purposes, the synthetic dataset is obtained where loads are assumed to follow Gaussian distribution and PVs are modeled by Beta distribution and wind generations follow Weibull distribution. The correlation between input variables

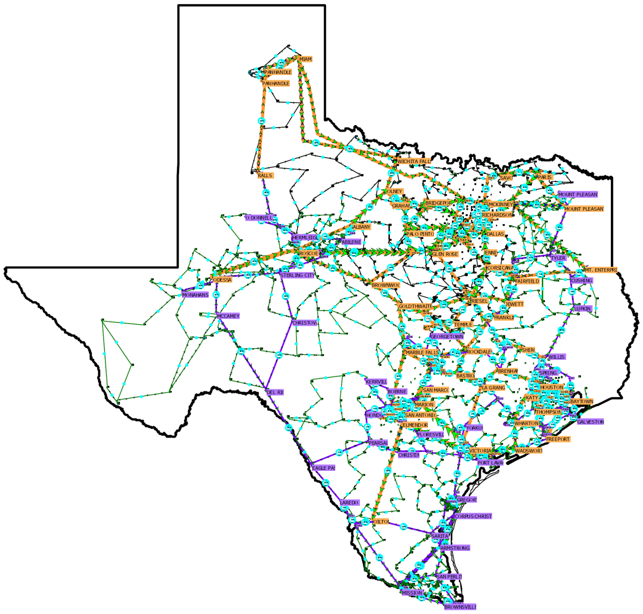


Figure 5.1: Diagram of the Texas 2000-bus system [27].

is characterized by copula [148]. Rotor angles of all generators are used as outputs for transient stability analysis.

We compare four methods: LHS, Many SGP (MSGP), SGP, and the proposed PPGP approaches. The benchmark result is obtained by MC simulation based on 10^5 samples. The mean absolute percentage error (MAPE) is utilized for measuring the overall model error:

$$e_{\mathcal{M}} = \frac{1}{T}\frac{1}{n_g}\sum_{T}\sum_{n_g}\left|\frac{x-\hat{x}}{x}\right| \times 100\%, \tag{5.17}$$

where T represents the number of simulation time points and n_g is the number of generators; x and $\hat{x}$ represent the true and estimated dynamic responses, respectively. Additionally, the MAPE of the mean and variance of rotor angle swings are also used as error indices for risk assessment, denoted as e_μ and e_{σ^2}. All simulations are conducted using MATLAB on Intel Core i5-12400 with a performance-core base frequency of 2.50 GHz. The CPU evaluation (testing) time is the average CPU time of 10 runs on 10,000 samples.

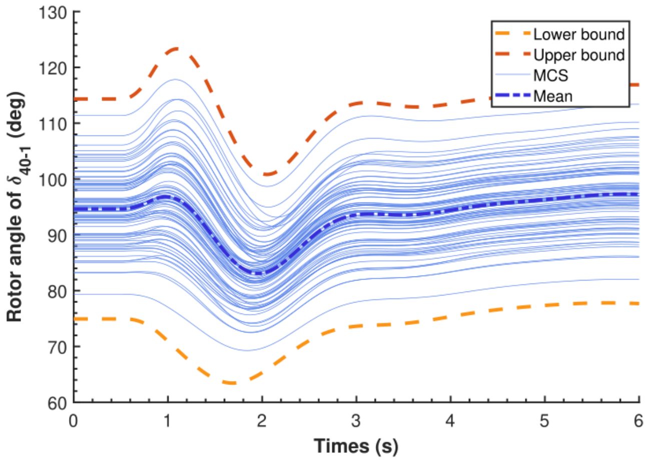


Figure 5.2: The upper and lower bounds of rotor angle δ_{40-1} in the 2000-bus system.

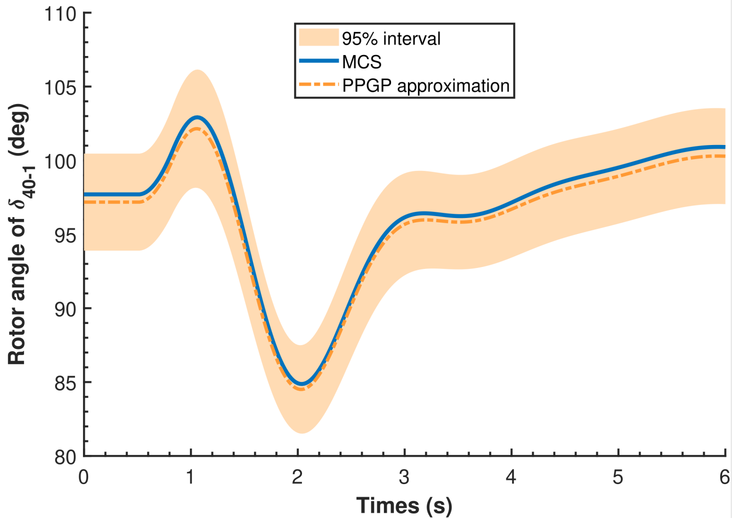


Figure 5.3: An example of the true dynamic response simulation and PPGP approximation with a 95% confidence interval.

Validation on the 2000-bus System

In this subsection, all the methods are tested on the Texas-2000 bus system that contains 1411 loads and 282 generators as shown in Equation (5.1). A total number of 471 PV farms and 470 wind generations are added into the system (as active power injections), yielding an input vector with 2352 variables. The number of simulation time points is 120 as the simulation lasts 6 seconds with time interval $\Delta t = 0.05$s. Along with 282 generators, the total number of outputs s reaches 33,840. For illustration purposes,

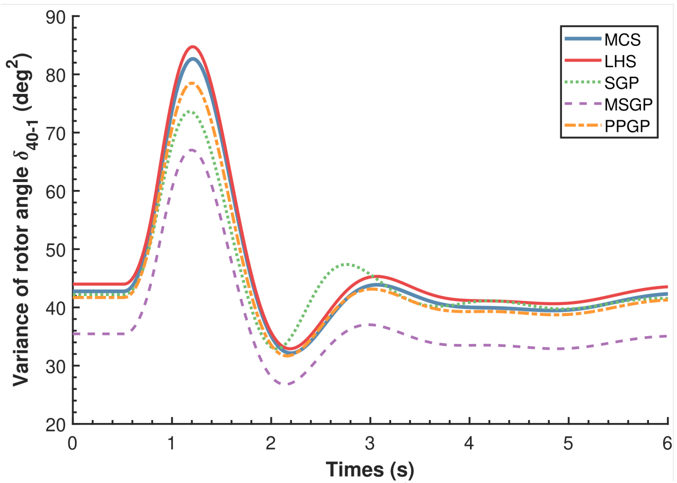


Figure 5.4: True and estimated variances of δ_{40-1} from different methods for the 2000-bus system.

Table 5.1: Performance Evaluation for Different Methods on the 2000-bus System.

Method	**MAPE**			**CPU Time** (s)	
	$e_{\mathcal{M}}(\%)$	$e_{\mu}(\times 10^{-2}\%)$	$e_{\sigma^2}(\%)$	Train	Test
MC	—	—	—	—	18589.82
LHS	—	3.73	2.89	—	5838.16
MSGP	4.88	3.98	17.09	11870.39	5297.86
SGP	1.32	5.41	3.81	6074.37	3157.92
PPGP	0.53	4.12	2.22	939.63	399.56

the angle of generator 40 with respect to reference generator 1 is chosen for demonstration. For LHS, 3,000 samples are used. MSGP and PPGP methods are trained based on 1,000 samples. For SGP, the training dataset is designed as 1,000 samples with 100 inducing points. Matérn kernel with roughness parameter $\alpha = 5/2$ is utilized for all GP-based methods.

Figure 5.2 shows how the rotor angle swing is affected by the uncertain resources with upper and lower bounds calculated using the 3σ rule and the variation of trajectories that arises from different initial conditions. Figure 5.3 illustrates an example of the response curve of one rotor over all time points with a held-out input. The out-of-sample prediction by PPGP, shown by the orange dashed curve, is accurate and the uncertainty of the prediction can be quantified by the predictive intervals shown as the shaded area. The predictive intervals cover most of the held-out observations. Furthermore, the estimated variance by the PP-GP is closer to the true variance than alternative methods, such as SGP and MSGP, demonstrated in Figure 5.4.

As shown in Table 5.1, the proposed PPGP has the smallest MAPE of the overall model error, $e_{\mathcal{M}}(\%)$, compared with MSGP and SGP. It also requires less computational time than the other two methods. In comparison, the $e_{\mathcal{M}}(\%)$ by MSGP is an order of magnitude larger than the one by PPGP. This is because the MSGP is not as stable as PPGP, since MSGP requires training s separate emulators and the parameters in each emulator need to be numerically optimized. The model performance might be improved with more samples introduced, but the cost of computation will also increase cubically fast to the number of observations, as MSGP has $\mathcal{O}(n^3 s)$ computational complexity. For the SGP method, the result seems closer to the true trajectory but it fails to approximate the overall shape of the true curve. SGP is able to utilize more samples due to the usage of inducing points, but the improvement of model performance by the increase of sample number will diminish rapidly as computational accuracy is hindered by the fixed number of inducing points. Increasing the number of inducing points may improve the performance but the computational complexity will also rapidly increase.

The model efficiency is represented by CPU time for model evaluation on 10,000 samples, i.e., CPU time of testing in Table 5.1. As a sampling method, the computational cost of LHS is linearly proportional to the number of samples involved. The time cost for MSGP evaluation is much larger than LHS, not to mention the time for training. By means of inducing points and variational inference, SGP manages to achieve faster evaluation time. Nevertheless, the efficiency improvement is not significant enough considering the loss of precision for forecasting dynamic responses. The PPGP has the smallest computational cost and overall predictive error. Even if the dimension of input

($d = 2,352$) and the dimension of the output (s=$33,840$) are both large and the overall predictive error ($e_{\mathcal{M}}(\%)$) by PPGP is less than 1%. This is because the mean and variance parameters in PPGP can be explicitly integrated out in calculating the predictive distribution in Equation (5.10), and the kernel parameters are assumed to be shared across different output coordinates. Increasing the number of observations can improve the predictive accuracy of PPGP, and the approximation methods, such as the ICML technique introduced before, can be used to further reduce the computational cost of PPGP.

5.2 Limitations and future directions for online changepoint detection method

In this section, we outline several limitations and future directions for the SKFCPD approach proposed in Chapter 2. This approach effectively models temporal correlations in the data, enabling the detection of changes in mean and variance for time series with a large number of observations. However, the SKFCPD method has its limitations that need further exploration.

- First, the SKFCPD is unable to model cross-dimension correlations for multidimensional data. While principal component analysis was used before the CUSUM algorithm for changepoint detection of multivariate time series [150], coherent statistical models that consider temporal correlation, such as vector autoregressive models [96] and latent factor processes [151], may be appealing to extend the scalable changepoint detection approach to multidimensional data.
- Second, while the SKFCPD method can detect various types of changes, it cannot distinguish them. Introducing a hypothesis testing step after detecting a change-

point, as shown in Algorithm 4, could help distinguish specific change types. Additionally, techniques like penalized likelihood methods [152] or Bayesian methods that offer posterior odds for change types, hold promise for enabling the change-point detection algorithm to identify the types of changes.

- Furthermore, SKFCPD is not specifically designed to detect the COVID-19 infection timing. Letting the SKFCPD focus on detecting the typical type of changes and potential timings in COVID-19 patients could further improve the detection performance. This could be achieved by incorporating the prior information into the algorithm, as seen in [153].

Beyond these limitations, further exploration could include (1) conducting distributed inference that maintains efficiency without requiring comprehensive patient information from various clinics or hospitals [154], where the modeling of temporal correlation from observations has often been neglected, and (2) studying shrinkage estimators to induce sparsity in the state space model, as seen in estimating Granger causality [155], to enhance efficient noise modeling and scalable estimation of dynamic changes.

5.3 Limitations and future directions for the epidemiology compartmental model

In this section, we list several limitations and future directions for the epidemiology compartmental model we developed in Chapter 3 that can robustly and efficiently estimate the COVID-19 transmission dynamics for over 3,000 U.S. counties.

- First, our findings are based on the available knowledge and model assumptions, as with all other studies. One critical parameter is the death rate, assumed to be

0.66% on average [117], whereas this parameter can vary across regions due to the demographic profile of the population and available medical resources. The studies of the prevalence of SARS-CoV-2 antibodies based on serology tests [113] can be used to determine the size of the population who have contracted SARS-CoV-2, and thus provides estimates on the death rate, as the death toll is observed.

- Second, we assume the infected population can develop immunity since recovery for a few months, which is commonly used in other models. The exact duration of immunity post-infection, however, remains unverified scientifically.
- Third, we assume that the number of susceptible individuals and, consequently, the number of individuals who have contracted SARS-CoV-2 can be written as a function of the number of observed confirmed cases and test positive rates, calibrated based on the death toll. More information such as the proportion of population adhere to the mitigation measures, mobility, and demographic profile can be used to improve the estimation of susceptible individuals in a region.

Beyond these limitations, our results can be used to mitigate the ongoing pandemic of SARS-COV-2 and other infectious disease outbreaks in the future. The estimated daily PoC SARS-CoV-2 at the county level, for example, is an interpretable measure to understand the risk of contracting COVID-19 on a daily basis and a surveillance marker to determine appropriate policy responses. Besides, Our method can be extended for vaccination [44]. Finally, further studies of this measure relative to different mobility, demographic information, and social-economic status can provide more precise guidance for local officials to protect vulnerable populations from contracting SARS-CoV-2.

5.4 Limitations and future directions for GP model on data with massive output

In this section, we present several limitations and future directions for the GOLF model proposed in Chapter 4. The GOLF model achieves fast computations by decomposing the likelihood function using orthogonal latent factor loadings and independent factor processes. However, improvements can be made to address some limitations of the GOLF model.

- First, the GOLF model assumes observations on a lattice, accommodating potential missing values. However, real-world datasets often display more complex and irregular arrangements, deviating from a perfect lattice structure. To extend the GOLF model's applicability to a wider range of data structures, exploring approximation methods for handling correlated data in non-lattice formats is essential. A promising direction is integrating Nearest Neighbor Gaussian Process (NNGP) approaches [54], known for their robustness in modeling spatially correlated data efficiently and scalably. Integrating NNGP with the GOLF model could significantly enhance its adaptability to complex real-world datasets.

- Second, determining the optimal number of factors in the GOLF model remains challenging, and reducing the number of factors can potentially improve computational efficiency. Techniques like Sparse PCA [156] could be valuable in developing systematic factor selection methods.

- Furthermore, direct marginalization of factor processes through an elementwise representation of GPs could further reduce computation times, especially when drawing numerous posterior samples.

Appendix A

Appendix of Chapter 2

A.1 Temporal Correlations in COVID-19 Patient Data

Figure A.1 shows autocorrelation function (ACF) and partial ACF [157] in the probability sequences of four COVID-19 patients, and the averages among all patients are shown in Figure A.2. We found that positive lag temporal correlations are common in longitudinal measurements of dialysis patients. The temporal correlation is modeled in SKF, enabling detection of the changepoint more quickly and precisely than other methods, such as BOCPD and CUSUM as shown in Table 2.1.

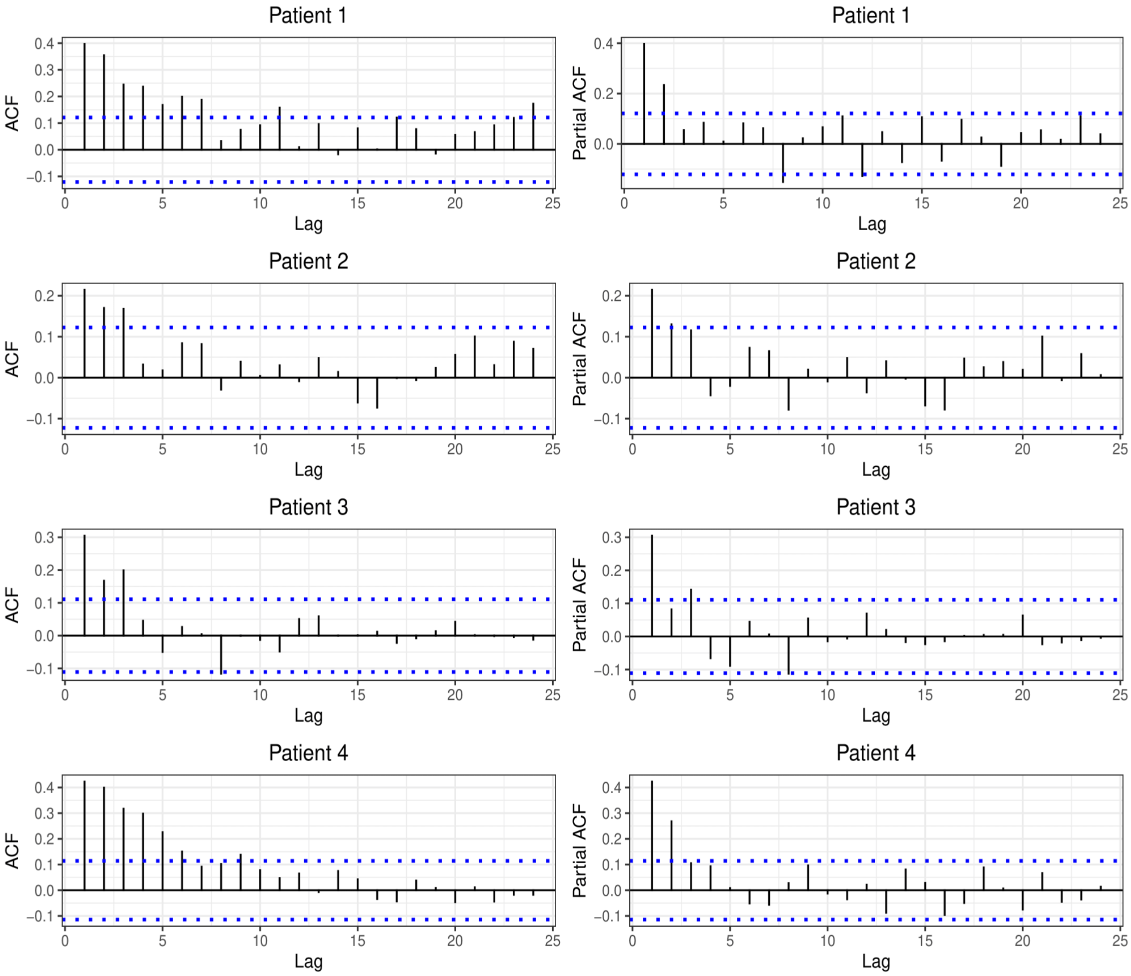


Figure A.1: ACF and partial ACF for the predictive probability sequences of four dialysis patients in Section 2.1.

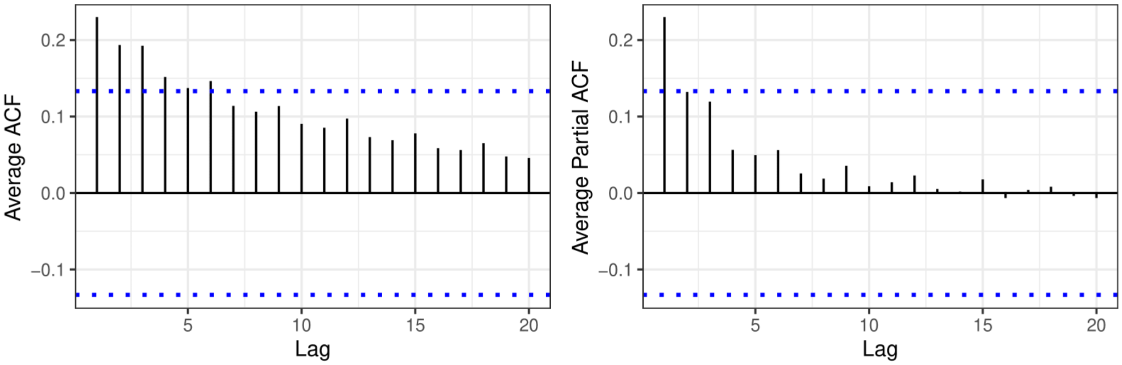


Figure A.2: Average ACF and Partial ACF for the predictive probability sequences of all dialysis patients.

A.2 Derivation of Equation (2.1)

We show the derivation of Equation (2.1) for BOCPD. At the nth time point, the joint distribution of measurements and most recent changepoint C_n can be derived below

$$
\begin{aligned}
& p\left(\mathbf{y}_{1:n}, C_n = t_i\right) \\
= & \sum_{j=1}^{n-1} p\left(\mathbf{y}_{1:n}, C_n = t_i, C_{n-1} = t_j\right) \\
= & \sum_{j=1}^{n-1} p(y_n, C_n = t_i \mid \mathbf{y}_{1:(n-1)}, C_{n-1} = t_j) p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j) \\
= & \sum_{j=1}^{n-1} p(y_n \mid \mathbf{y}_{1:(n-1)}, C_{n-1} = t_j, C_n = t_i) p(C_n = t_i \mid C_{n-1} = t_j, \mathbf{y}_{1:(n-1)}) p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j) \\
= & \underbrace{p\left(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i\right)}_{\text{predictive distribution}} \sum_{j=1}^{n-1} \underbrace{p\left(C_n = t_i \mid C_{n-1} = t_j\right)}_{\text{hazard}} p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j\right) \\
= & \begin{cases} p\left(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i\right) \left(1 - H(t_i)\right) p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_i\right), & i < n, \\ p\left(y_n \mid C_n = t_n\right) H(t_n) \sum_{j=1}^{n-1} p\left(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j\right), & i = n, \end{cases}
\end{aligned}
$$

where the first three equations directly follow from the conditional probability equation. The fourth equation is based on Assumptions 1 and 2. First, given the most recent changepoint $C_n = t_i$, the observations before and after time t_i are independent. i.e., $(\mathbf{y}_{1:i-1} \perp\!\!\!\perp \mathbf{y}_{i:n}) \mid C_n = t_i$. This leads to the expression $p(y_n \mid \mathbf{y}_{1:(n-1)}, C_{n-1} = t_j, C_n = t_i) = p(y_n \mid \mathbf{y}_{i:(n-1)}, C_n = t_i)$. Second, the time point of the most recent changepoint at the nth time C_n, conditioned on the most recent changepoint at the $(n-1)$th time C_{n-1}, is independent of the previous observations $\mathbf{y}_{1:(n-1)}$, resulting in the expression $p(C_n = t_i \mid C_{n-1} = t_j, \mathbf{y}_{1:(n-1)}) = p(C_n = t_i \mid C_{n-1} = t_j)$.

Two scenarios are considered in the last equation. First, when $i < n$, the most recent changepoint $C_n = t_i$ is prior to time t_n, indicating that time t_n is not a changepoint. By the definition of hazard function in Section 2.2.1, the summation over j from 1 to $n-1$ in the fourth equation is reduced to $j = i$ in the fifth equation when $i < n$. Second, when $i = n$, meaning that t_n is a changepoint, $C_{n-1} = t_j$ could take any values from t_1 to t_{n-1}. Consequently, the summation over $p(\mathbf{y}_{1:(n-1)}, C_{n-1} = t_j)$ in the fourth equation still holds in the fifth equation when $i = n$.

A.3 Derivation of Equation (2.7)

We show the derivation of Equation (2.7). Denote the observations from times t_i to t_n as $\mathbf{y}_{i:n} = (y_i, \ldots, y_n)^T$. $n' = n - i + 1$ represents the total number of observations from t_i to t_n. We assume that all observations in $\mathbf{y}_{i:n}$ are from the same time segment and follow the GP model with the same mean parameter μ and variance parameter σ^2. Denote the parameter set as $\mathbf{\Theta} = (\mu, \sigma^2, \gamma, \eta)$. Given an objective prior for μ and σ^2 such that $\pi(\mu, \sigma^2) \propto \frac{1}{\sigma^2}$, the likelihood function for $\mathbf{y}_{i:n}$, with μ and σ^2 integrated out, has the following form for any $i < n-1$.

$$\begin{aligned}
p(\mathbf{y}_{i:n} \mid \gamma, \eta) &= \int p\left(\mathbf{y}_{i:n} \mid \boldsymbol{\Theta}\right) \pi(\mu, \sigma^2) d\mu d\sigma^2 \\
&\propto (2\pi)^{-\frac{n'}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \\
&\quad \times \int \left(\sigma^2\right)^{-\frac{n'}{2}-1} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_{i:n} - \mu\mathbf{1}_{n'})^T \mathbf{K}_{n'}^{-1} (\mathbf{y}_{i:n} - \mu\mathbf{1}_{n'})\right) d\mu d\sigma^2 \\
&\propto (2\pi)^{-\frac{n'}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \int \left(\sigma^2\right)^{-\frac{n'}{2}-1} \exp\left(-\frac{1}{2\sigma^2}\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) \\
&\quad \times \exp\Bigg(-\frac{1}{2\sigma^2}\left(\mu - \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n}\right)^T \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \\
&\quad \times \left(\mu - \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n}\right) \Bigg) d\mu d\sigma^2 \\
&\propto (2\pi)^{-\frac{n'-1}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-\frac{1}{2}} \\
&\quad \times \int \left(\sigma^2\right)^{-\frac{n'-1}{2}-1} \exp\left(-\frac{1}{2\sigma^2}\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) d\sigma^2 \\
&\propto \left(\frac{\pi}{2}\right)^{-\frac{n'-1}{2}} \tau\left(\frac{n'-1}{2}\right) \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-\frac{1}{2}} \left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right)^{-\frac{n'-1}{2}}, \quad \text{(A.1)}
\end{aligned}$$

where

$$\mathbf{M}_{n'} = \mathbf{K}_{n'}^{-1} - \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1}.$$

Therefore, for any $i < n-1$, given the likelihood function for $\mathbf{y}_{i:n}$ in Equation (A.1), the predictive distribution of y_n given $\mathbf{y}_{i:(n-1)}$ can be derived as follows.

$$\begin{aligned}
p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta) &= \frac{p(\mathbf{y}_{i:n} \mid \gamma, \eta)}{p(\mathbf{y}_{i:(n-1)} \mid \gamma, \eta)} \\
&\propto \frac{\Gamma\left(\frac{n'-1}{2}\right)}{\Gamma\left(\frac{n'-2}{2}\right)} \left(\frac{\left|\mathbf{K}_{n'}\right|}{\left|\mathbf{K}_{n'-1}\right|}\right)^{-1/2} \left(\frac{\left|\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right|}{\left|\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}\right|}\right)^{-1/2} \exp\left(-S_{n'}^2\right) \qquad \text{(A.2)}
\end{aligned}$$

where $S_{n'}^2 = \left(\frac{n'-1}{2}\right) \log\left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) - \left(\frac{n'-2}{2}\right) \log\left(\mathbf{y}_{i:(n-1)}^T \mathbf{M}_{n'-1} \mathbf{y}_{i:(n-1)}\right)$.

However, when $i = n-1$, the predictive distribution of y_n given y_{n-1} can't be computed by Equation (A.2).This is due to the fact that n', representing the number of observations between time t_i and t_n, is equal to 2, which results in the divergent Gamma function $\Gamma(\frac{n'-2}{2}) = \Gamma(0)$. In this case, we need to separately integrate out the parameters μ and σ^2 in order to compute the predictive distribution $p(y_n \mid y_{n-1}, \gamma, \eta)$. First, we integrate out μ with prior distribution $\pi(\mu) \propto 1$ in the joint distribution follows

$$\begin{aligned}
p(\mathbf{y}_{i:n} \mid \sigma^2, \gamma, \eta) &= \int p\left(\mathbf{y}_{i:n} \mid \boldsymbol{\Theta}\right) \pi(\mu) d\mu \\
&\propto (2\pi)^{-\frac{n'}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \\
&\quad \times \int \left(\sigma^2\right)^{-\frac{n'}{2}} \exp\left(-\frac{1}{2\sigma^2}(\mathbf{y}_{i:n} - \mu\mathbf{1}_{n'})^T \mathbf{K}_{n'}^{-1} (\mathbf{y}_{i:n} - \mu\mathbf{1}_{n'})\right) d\mu \\
&\propto (2\pi)^{-\frac{n'}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \int \left(\sigma^2\right)^{-\frac{n'}{2}} \exp\left(-\frac{1}{2\sigma^2}\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) \\
&\quad \times \exp\Bigg(- \frac{1}{2\sigma^2}\left(\mu - \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n}\right)^T \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \\
&\quad \times \left(\mu - \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n}\right) \Bigg) d\mu \\
&\propto \left(2\pi\sigma^2\right)^{-\frac{n'-1}{2}} \left|\mathbf{K}_{n'}\right|^{-\frac{1}{2}} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right).
\end{aligned} \tag{A.3}$$

And the predictive distribution, when $i = n-1$, after integrating out μ, is as follows.

$$\begin{aligned}
p(y_n \mid \mathbf{y}_{i:(n-1)}, \sigma^2, \gamma, \eta) = p(y_n \mid y_{n-1}, \sigma^2, \gamma, \eta) = \frac{p(\mathbf{y}_{(n-1):n} \mid \sigma^2, \gamma, \eta)}{p(y_{n-1} \mid \sigma^2, \gamma, \eta)}. \\
\propto \left(\sigma^2\right)^{-\frac{1}{2}} \left(\frac{\left|\mathbf{K}_{n'}\right|}{\left|\mathbf{K}_{n'-1}\right|}\right)^{-1/2} \left(\frac{\left|\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right|}{\left|\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}\right|}\right)^{-1/2} \exp\left(-\frac{1}{2\sigma^2}\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right)
\end{aligned}$$

Then, we integrate out the parameter σ^2 with the prior distribution $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$.

$$
\begin{aligned}
p(y_n \mid y_{n-1}, \gamma, \eta) &= \int p(y_n \mid y_{n-1}, \sigma^2, \gamma, \eta)\pi(\sigma^2) d\sigma^2 \\
&\propto \int \left(\sigma^2\right)^{-3/2} \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \\
&\qquad \times \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{y}_{(n-1):n}^T \mathbf{M}_{n'} \mathbf{y}_{(n-1):n}\right)\right) d\sigma^2 \\
&\propto \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \left(\mathbf{y}_{(n-1):n}^T \mathbf{M}_{n'} \mathbf{y}_{(n-1):n}\right)^{-1/2}. \qquad \text{(A.4)}
\end{aligned}
$$

Combining Equations (A.2) and (A.4), we have for i from 1 to $n-1$,

$$
\begin{aligned}
p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right) &= \frac{p\left(\mathbf{y}_{i:n} \mid \gamma, \eta\right)}{p\left(\mathbf{y}_{i:(n-1)} \mid \gamma, \eta\right)} \\
&\propto \begin{cases} \frac{\Gamma(\frac{n'-1}{2})}{\Gamma(\frac{n'-2}{2})} \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \exp\left(-S_{n'}^2\right), & i < n-1 \\ \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}}{\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}}\right)^{-1/2} \left(\mathbf{y}_{(n-1):n}^T \mathbf{M}_{n'} \mathbf{y}_{(n-1):n}\right)^{-1/2}, & i = n-1 \end{cases}
\end{aligned}
$$

where $S_{n'}^2 = \left(\frac{n'-1}{2}\right) \log\left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) - \left(\frac{n'-2}{2}\right) \log\left(\mathbf{y}_{i:(n-1)}^T \mathbf{M}_{n'-1} \mathbf{y}_{i:(n-1)}\right)$ and $\mathbf{M}_{n'} = \mathbf{K}_{n'}^{-1} - \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1}$.

A.4 Proof of Lemma 3

Proof: Let the observations $\mathbf{Y}_{1:n'} = (Y_1, \dots, Y_{n'})^T$ follow a multivariate normal distribution with correlation matrix $\mathbf{K}_{n'}$, i.e.

$$
\mathbf{Y}_{1:n'} \sim \mathcal{MN}(0, \mathbf{K}_{n'}). \qquad \text{(A.5)}
$$

The likelihood function of $\mathbf{Y}_{1:n'}$ can be decomposed as follows.

$$\begin{aligned} l(\mathbf{Y}_{1:n'}; \mathbf{K}_{n'}) &= (2\pi)^{-\frac{n'}{2}} |\mathbf{K}_{n'}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\mathbf{Y}_{1:n'}^T \mathbf{K}_{n'}^{-1} \mathbf{Y}_{1:n'}\right) \\ &= (2\pi)^{-\frac{n'}{2}} |\mathbf{K}_{n'}|^{-\frac{1}{2}} \exp\left(\frac{1}{2}\mathbf{Y}_{1:n'}^T \mathbf{U}_{n'}^T \mathbf{U}_{n'} \mathbf{Y}_{1:n'}\right) \\ &= (2\pi)^{-\frac{n'}{2}} |\mathbf{K}_{n'}|^{-\frac{1}{2}} \exp\left(\frac{1}{2}\sum_{k=1}^{n'} (\mathbf{U}_{n'} \mathbf{Y}_{1:n'})_k^2\right), \end{aligned} \tag{A.6}$$

where $(\cdot)_k$ represents the k-th element in the vector. The second equation is a result of applying the Cholesky decomposition on the correlation matrix, where $\mathbf{K}_{n'} = \mathbf{L}_{n'}\mathbf{L}_{n'}^T$. Here, $\mathbf{L}_{n'}$ is a lower triangular matrix. Consequently, we have $\mathbf{K}_{n'}^{-1} = \mathbf{U}_{n'}^T \mathbf{U}_{n'}$, with $\mathbf{U}_{n'} = \mathbf{L}_{n'}^{-1}$.

Next, we show that the likelihood function $l(\mathbf{Y}_{1:n'}; \mathbf{K}_{n'})$ in Equation (A.6) is a function of the Kalman filter parameters in Lemma 1. Based on Equation (1.23), the predictive distribution of Y_k given $\mathbf{Y}_{1:(k-1)}$ for $k = 2, \ldots, n'$ takes the following form.

$$p(Y_k \mid \mathbf{Y}_{1:(k-1)}) \sim \mathcal{N}(f_k, Q_k), \tag{A.7}$$

where f_k, and Q_k are scalar Kalman filter parameters that can be computed in time complexity $O(1)$ iteratively. We will discuss the details of this computation later in this section. When $k = 1$, based on Equation (1.23), we have $p(Y_1) \sim \mathcal{N}(f_1, Q_1)$, where $f_1 = 0$ and $Q_1 = \mathbf{F}_1\mathbf{D}_1\mathbf{F}_1^T + \eta$. Therefore, the likelihood function of $\mathbf{Y}_{1:n'}$ from the Kalman filter follows.

$$\begin{aligned} l(\mathbf{Y}_{1:n'};\mathbf{K}_{n'}) &= p(Y_1)\prod_{k=2}^{n'} p(Y_k \mid \mathbf{Y}_{1:(k-1)}) \\ &= \prod_{k=1}^{n'}(2\pi)^{-\frac{1}{2}}Q_k^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\frac{(Y_k-f_k)^2}{Q_k}\right) \\ &= (2\pi)^{-\frac{n'}{2}}\prod_{k=1}^{n'}Q_k^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\sum_{k=1}^{n'}\frac{(Y_k-f_k)^2}{Q_k}\right), \end{aligned} \tag{A.8}$$

where the first equation is formulated using conditional probability, and the second equation is from Equation (A.7). Since Equations (A.6) and (A.8) are equivalent for any $n' > 0$, terms in Equation (A.6) can be replaced by the Kalman Filter parameters in Equation (A.8), i.e.

$$|\mathbf{K}_{n'}|^{-\frac{1}{2}} = \prod_{k=1}^{n'} Q_k^{-\frac{1}{2}} \text{ and} \tag{A.9}$$

$$(\mathbf{U}_{n'}\mathbf{Y}_{1:n'})_k = \frac{Y_k - f_k}{Q_k^{\frac{1}{2}}} \text{ for any } 1 \le k \le n' \tag{A.10}$$

We denote $\mathbf{u}_{n'} = \mathbf{U}_{n'}\mathbf{1}_{n'} = (u_1,\ldots,u_{n'})^T$ and $\mathbf{v}_{i,n'} = \mathbf{U}_{n'}\mathbf{y}_{i:n} = (v_{i,1},\ldots,v_{i,n'})^T$. By substituting $\mathbf{Y}_{1:n'}$ in Equation (A.10) with $\mathbf{1}_{n'}$ and $\mathbf{y}_{i:n}$, respectively, we have for $k = 1,\ldots,n'$,

$$\begin{aligned} u_k = (\mathbf{U}_{n'}\mathbf{1}_{n'})_k &= \frac{1-f_k^u}{\sqrt{Q_k^u}}, \\ v_{i,k} = (\mathbf{U}_{n'}\mathbf{y}_{i:n})_k &= \frac{y_{i+k-1}-f_{i,k}^v}{\sqrt{Q_{i,k}^v}}, \end{aligned} \tag{A.11}$$

where f_k^u and Q_k^u are the Kalman filter parameters when $\mathbf{Y}_{1:n'} = \mathbf{1}_{n'}$ in Equation (1.23), and $f_{i,k}^v$ and $Q_{i,k}^v$ are the Kalman filter parameters when $\mathbf{Y}_{1:n'} = \mathbf{y}_{i:n}$.

Next, we discuss in detail the sequential computation process of Kalman filter parameters f_k^u, Q_k^u, $f_{i,k}^v$ and $Q_{i,k}^v$ for $k = 1,\ldots,n'$. Based on Lemma 1, for the vector $\mathbf{u}_{n'}$, given the matrics $\mathbf{F}^u_{k-1}$, $\mathbf{G}^u_{k-1}$, $\mathbf{d}^u_{k-1}$ and $\mathbf{D}^u_{k-1}$ at the $(k-1)$-th step and the matrics $\mathbf{F}^u_k$, $\mathbf{G}^u_k$ and $\mathbf{W}^u_k$ at the k-th step, we have the updating equations below when $k \geq 2$

$$\begin{aligned} f^u_k &= \mathbb{E}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{1}_{k-1}, \gamma, \eta] \\ &= \mathbf{F}^u_k \mathbf{d}^u_k \\ &= \mathbf{F}^u_k \mathbf{G}^u_k \mathbf{m}^u_{k-1} \\ &= \mathbf{F}^u_k \mathbf{G}^u_k \mathbf{d}^u_{k-1} + \mathbf{F}^u_k \mathbf{G}^u_k \mathbf{D}^u_{k-1} (\mathbf{F}^u_{k-1})^T \frac{1 - f^u_{k-1}}{Q^u_{k-1}} \\ &= g^u_k(f^u_{k-1}, Q^u_{k-1}), \text{ and} \end{aligned} \tag{A.12}$$

$$\begin{aligned} Q^u_k &= \mathbb{V}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{1}_{k-1}, \gamma, \eta] \\ &= \mathbf{F}^u_k \mathbf{D}^u_k (\mathbf{F}^u_k)^T + \sigma^2_0 \\ &= \mathbf{F}^u_k \mathbf{G}^u_k \mathbf{B}^u_{k-1} (\mathbf{F}^u_k \mathbf{G}^u_k)^T + \mathbf{F}^u_k \mathbf{W}^u_k (\mathbf{F}^u_k)^T + \eta \\ &= \beta^u_{1,k} + \beta^u_{2,k} \frac{1}{Q^u_{k-1}} \\ &= h^u_k(Q^u_{k-1}), \end{aligned} \tag{A.13}$$

where $\beta^u_{1,k} = \mathbf{F}^u_k \mathbf{G}^u_k \mathbf{D}^u_{k-1} (\mathbf{F}^u_k \mathbf{G}^u_k)^T + \mathbf{F}^u_k \mathbf{W}^u_k (\mathbf{F}^u_k)^T + \eta$ and $\beta^u_{2,k} = -\left(\mathbf{F}^u_k \mathbf{G}^u_k \mathbf{D}^u_{k-1} (\mathbf{F}^u_{k-1})^T\right)^2$. For $k = 1$, the initialization of f^u_1 and Q^u_1 from Equation (1.23) follows

$$\begin{aligned} f^u_1 &= 0 \\ Q^u_1 &= \mathbf{F}^u_1 \mathbf{D}^u_1 (\mathbf{F}^u_1)^T + \eta. \end{aligned} \tag{A.14}$$

Similarly, for the vector $\mathbf{v}_{i,n'}$, given the matrics $\mathbf{F}^v_{i,k-1}$, $\mathbf{G}^v_{i,k-1}$, $\mathbf{d}^v_{i,k-1}$ and $\mathbf{D}^v_{i,k-1}$ at the $(k-1)$-th step and the matrics $\mathbf{F}^v_{i,k}$, $\mathbf{G}^v_{i,k}$ and $\mathbf{W}^v_{i,k}$ at the k-th step, we have the following updating equations for $k \geq 2$.

$$\begin{aligned}
f^v_{i,k} &= \mathbb{E}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{y}_{i:(i+k-2)}, \gamma, \eta] \\
&= \mathbf{F}^v_{i,k}\mathbf{d}^v_{i,k} \\
&= \mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{m}^v_{i,k-1} \\
&= \mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{d}^v_{i,k-1} + \mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{D}^v_{i,k-1}(\mathbf{F}^v_{i,k-1})^T\frac{y_{i+k-2} - f^v_{i,k-1}}{Q^v_{i,k-1}} \\
&= g^v_{i,k}(f^v_{i,k-1}, Q^v_{i,k-1}), \text{ and} \qquad \text{(A.15)}
\end{aligned}$$

$$\begin{aligned}
Q^v_{i,k} &= \mathbb{V}_{\mathbf{Y}_{1:k}}[Y_k \mid \mathbf{Y}_{1:(k-1)} = \mathbf{y}_{i:(i+k-2)}, \gamma, \eta] \\
&= \mathbf{F}^v_{i,k}\mathbf{D}^v_{i,k}(\mathbf{F}^v_{i,k})^T + \sigma_0^2 \\
&= \mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}(\mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{B}^v_{i,k-1})^T + \mathbf{F}^v_{i,k}\mathbf{W}^v_{i,k}(\mathbf{F}^v_{i,k})^T + \sigma_0^2 \\
&= \beta^v_{1,i,k} + \beta^v_{2,i,k}\frac{1}{Q^v_{i,k-1}} \\
&= h^v_{i,k}(Q^v_{i,k-1}), \qquad \text{(A.16)}
\end{aligned}$$

where $\beta^v_{1,i,k} = \mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{D}^v_{i,k-1}(\mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k})^T + \mathbf{F}^v_{i,k}\mathbf{W}^v_{i,k}(\mathbf{F}^v_{i,k})^T + \sigma_0^2$ and $\beta^v_{2,i,k} = -(\mathbf{F}^v_{i,k}\mathbf{G}^v_{i,k}\mathbf{D}^v_{i,k-1}(\mathbf{F}^v_{i,k-1})^T)^2$. For $k = 1$, the initialization of f^u_1 and Q^u_1 from Equation (1.23) follows

$$\begin{aligned}
f^v_{i,1} &= 0, \\
Q^v_{i,1} &= \mathbf{F}^v_{i,1}\mathbf{D}^v_{i,1}(\mathbf{F}^v_{i,1})^T + \eta.
\end{aligned} \qquad \text{(A.17)}$$

■

A.5 Proof of Theorem 1

Proof:

The fast computation of the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$ can be achieved by sequential Kalman filter. By applying the Cholesky decomposition on the correlation matrix $\mathbf{K}_{n'}$, we have $\mathbf{K}_{n'} = \mathbf{L}_{n'}\mathbf{L}_{n'}^T$, where $\mathbf{L}_{n'}$ is a lower triangular matrix, and $\mathbf{K}_{n'}^{-1} = \mathbf{U}_{n'}^T\mathbf{U}_{n'}$ with $\mathbf{U}_{n'} = \mathbf{L}_{n'}^{-1}$. Then, by denoting $\mathbf{u}_{n'} = \mathbf{U}_{n'}\mathbf{1}_{n'} = (u_1, \ldots, u_{n'})^T$ and $\mathbf{v}_{i,n'} = \mathbf{U}_{n'}\mathbf{y}_{i:n} = (v_{i,1}, \ldots, v_{i,n'})^T$, we have the following equations.

$$\begin{aligned}
\mathbf{1}_{n'}^T\mathbf{K}_{n'}^{-1}\mathbf{1}_{n'} &= \mathbf{1}_{n'}^T\mathbf{U}_{n'}^T\mathbf{U}_{n'}\mathbf{1}_{n'} = \mathbf{u}_{n'}^T\mathbf{u}_{n'}, \\
\mathbf{y}_{i:n}^T\mathbf{K}_{n'}^{-1}\mathbf{y}_{i:n} &= \mathbf{y}_{i:n}^T\mathbf{U}_{n'}^T\mathbf{U}_{n'}\mathbf{y}_{i:n} = \mathbf{v}_{i,n'}^T\mathbf{v}_{i,n'}, \\
\mathbf{y}_{i:n}^T\mathbf{K}_{n'}^{-1}\mathbf{1}_{n'} &= \mathbf{y}_{i:n}^T\mathbf{U}_{n'}^T\mathbf{U}_{n'}\mathbf{1}_{n'} = \mathbf{v}_{i,n'}^T\mathbf{u}_{n'}, \\
\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|} &= \frac{\Pi_{k=1}^{n'}Q_k^u}{\Pi_{k=1}^{n'-1}Q_k^u} = Q_{n'}^u,
\end{aligned} \tag{A.18}$$

where the last equation is derived based on Equation (A.9).

Given the Equation (A.18), the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$ in Equation (2.7) can be represented using the two vectors $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$. For $i < n-1$, we have

$$\begin{aligned}
&p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right) \\
&\propto \frac{\Gamma\left(\frac{n'-1}{2}\right)}{\Gamma\left(\frac{n'-2}{2}\right)} \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\left|\mathbf{1}_{n'}^T\mathbf{K}_{n'}^{-1}\mathbf{1}_{n'}\right|}{\left|\mathbf{1}_{n'-1}^T\mathbf{K}_{n'-1}^{-1}\mathbf{1}_{n'-1}\right|}\right)^{-1/2} \frac{\left(\mathbf{y}_{i:n}^T\mathbf{M}_{n'}\mathbf{y}_{i:n}\right)^{-\frac{n'-1}{2}}}{\left(\mathbf{y}_{i:(n-1)}^T\mathbf{M}_{n'-1}\mathbf{y}_{i:(n-1)}\right)^{\frac{-n'-2}{2}}} \\
&\propto \frac{\Gamma\left(\frac{n'-1}{2}\right)}{\Gamma\left(\frac{n'-2}{2}\right)} \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\left|\mathbf{1}_{n'}^T\mathbf{K}_{n'}^{-1}\mathbf{1}_{n'}\right|}{\left|\mathbf{1}_{n'-1}^T\mathbf{K}_{n'-1}^{-1}\mathbf{1}_{n'-1}\right|}\right)^{-1/2}
\end{aligned}$$

$$
\begin{aligned}
&\times \frac{\left(\mathbf{y}_{i:n}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n} - \mathbf{y}_{i:n}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'} \left(\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right)^{-1} \mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{y}_{i:n}\right)^{-\frac{n'-1}{2}}}{\left(\mathbf{y}_{i:(n-1)}^T \mathbf{K}_{n'-1}^{-1} \mathbf{y}_{i:(n-1)} - \mathbf{y}_{i:(n-1)}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1} \left(\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}\right)^{-1} \mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{y}_{i:(n-1)}\right)^{-\frac{n'-2}{2}}} \\
&\propto \frac{\Gamma\left(\frac{n'-1}{2}\right)}{\Gamma\left(\frac{n'-2}{2}\right)} (Q_{n'}^u)^{-1/2} \left(\frac{\mathbf{u}_{n'}^T \mathbf{u}_{n'}}{\mathbf{u}_{n'-1}^T \mathbf{u}_{n'-1}}\right)^{-1/2} \\
&\times \frac{\left(\mathbf{v}_{i,n'}^T \mathbf{v}_{i,n'} - \mathbf{v}_{i,n'}^T \mathbf{u}_{n'} \left(\mathbf{u}_{n'}^T \mathbf{u}_{n'}\right)^{-1} \mathbf{u}_{n'}^T \mathbf{v}_{i,n'}\right)^{-\frac{n'-1}{2}}}{\left(\mathbf{v}_{i,n'-1}^T \mathbf{v}_{i,n'-1} - \mathbf{v}_{i,n'-1}^T \mathbf{u}_{n'-1} \left(\mathbf{u}_{n'-1}^T \mathbf{u}_{n'-1}\right)^{-1} \mathbf{u}_{n'-1}^T \mathbf{v}_{i,n'-1}\right)^{-\frac{n'-2}{2}}} \\
&\propto \frac{\Gamma\left(\frac{n'-1}{2}\right)}{\Gamma\left(\frac{n'-2}{2}\right)} (Q_{n'}^u)^{-1/2} \left(\frac{\mathbf{u}_{n'}^T \mathbf{u}_{n'}}{\mathbf{u}_{n'-1}^T \mathbf{u}_{n'-1}}\right)^{-1/2} \exp\left(-S_{n'}^2\right),
\end{aligned}
$$

where $S_{n'}^2 = \left(\frac{n'-1}{2}\right) \log\left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right) - \left(\frac{n'-2}{2}\right) \log\left(\mathbf{y}_{i:(n-1)}^T \mathbf{M}_{n'-1} \mathbf{y}_{i:(n-1)}\right)$, and $\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n} = \mathbf{v}_{i,n'}^T \mathbf{v}_{i,n'} - \left(\mathbf{u}_{n'}^T \mathbf{u}_{n'}\right)^{-1} \left(\mathbf{v}_{i,n'}^T \mathbf{u}_{n'}\right)^2$. The third proportion is derived based on Equation (A.18). For $i = n-1$, similarly we have

$$
\begin{aligned}
&p(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta) \\
&\propto \left(\frac{|\mathbf{K}_{n'}|}{|\mathbf{K}_{n'-1}|}\right)^{-1/2} \left(\frac{\left|\mathbf{1}_{n'}^T \mathbf{K}_{n'}^{-1} \mathbf{1}_{n'}\right|}{\left|\mathbf{1}_{n'-1}^T \mathbf{K}_{n'-1}^{-1} \mathbf{1}_{n'-1}\right|}\right)^{-1/2} \left(\mathbf{y}_{(n-1):n}^T \mathbf{M}_{n'} \mathbf{y}_{(n-1):n}\right)^{-\frac{1}{2}} \\
&\propto (Q_{n'}^u)^{-\frac{1}{2}} \left(\frac{\mathbf{u}_{n'}^T \mathbf{u}_{n'}}{\mathbf{u}_{n'-1}^T \mathbf{u}_{n'-1}}\right)^{-1/2} \left(\mathbf{y}_{i:n}^T \mathbf{M}_{n'} \mathbf{y}_{i:n}\right)^{-1/2}.
\end{aligned}
$$

The main advantage of expressing the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$ in terms of vectors $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$ is the reduction of computational complexity. Based on Equation (A.11), vectors $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$ can be sequentially updated from $\mathbf{u}_{n'-1}$ and $\mathbf{v}_{i,n'-1}$ with only $\mathcal{O}(1)$ operations. Consequently, when written as functions of $\mathbf{u}_{n'}$ and $\mathbf{v}_{i,n'}$, the predictive distribution $p\left(y_n \mid \mathbf{y}_{i:(n-1)}, \gamma, \eta\right)$ can also be sequentially computed in $\mathcal{O}(1)$ operations.

■

Table A.1: Comparisons of time complexity between SKF and other online changepoint detection methods for detecting the most recent changepoint when there are n observations.

Method	Computational complexity	Temporal correlation	Allow detecting variance change
Cumulative sum chart [100]	$\mathcal{O}(1)$	No	No
Bayesian online changepoint detection [78, 79]	$\mathcal{O}(n)$	No	Yes
Bayesian analysis with dependence across regimes [37]	$\mathcal{O}(n)$	Dependence across segments	No
Detecting abrupt changes [38]	$\mathcal{O}(n)$	Dependence from autocorrelated noise	No
Gaussian process changepoint detection [39]	$\mathcal{O}(n^3)$	Dependence within segments	Yes
SKF	$\mathcal{O}(n')$	Dependence within segments	Yes

A.6 Comparison and Connection with Other Models

We compare various online changepoint detection methods in Table A.1. For instance, the cumulative sum (CUSUM) chart [100, 150] and BOCPD methods do not consider temporal correlation in the data. To address this issue, [37] introduces a piecewise polynomial regression model that considers temporal correlations between segments, yet this method doesn't model the temporal correlation within each segment, an aspect our approach effectively handles. Additionally, [38] specifically targets detecting mean shifts in time series with autocorrelated noise. However, our proposed approach can detect both mean and variance shifts, making it a more flexible solution for changepoint detection.

The GPCPD method [39] models correlation between observations at each time point, whereas it has a large computational cost. In comparison, the computational complexity of SKF scales linearly to the number of observations between the most recent changepoints and the previous changepoint in the SKF algorithm. Furthermore, the mean and variance are efficiently integrated out in SKF based on the most recent observations, which makes it particularly suitable for online changepoint detection.

Other techniques to reduce computational complexity include rank 1 update [95], which has higher computational complexity than SKF. Another approach to reduce the computational complexity is to factorizing the semi-separable covariance matrix in a backward recursive algorithm [158]. The SKF has two advantages over this approach. First, we integrate out mean and variance parameters in SKF, which is crucial for computing the predictive distribution based on the latest information for changepoint detection. Secondly, the SKF algorithm is applicable to all dynamical linear models that go beyond the Gaussian process with a Matérn kernel.

A.7 The CUSUM Algorithm

The cumulative sum (CUSUM) control chart is an online changepoint detection algorithm [100]. Denote $y_1, \ldots, y_n$ as observations from time t_1 to t_n. We define $y_j^* = \frac{y_j - \bar{y}_{1:j}}{\hat{\sigma}_{1:j}}$ as standardized observations for $j = 1, \ldots, n$, where the $\bar{y}_{1:j}$ and $\hat{\sigma}_{1:j}$ are the sample mean and standard deviation for all the observations until time t_j, respectively. A changepoint is detected at time t_n if either of the two CUSUM statistics conditions are met, i.e.,

$$S_n^+ > h \text{ or } S_n^- < -h, \tag{A.19}$$

where

$$\begin{aligned} S_n^+ &= \max(0, S_{n-1}^+ + y_n^* - K_n), \\ S_n^- &= -\max(0, S_{n-1}^- - y_n^* + K_n), \end{aligned} \tag{A.20}$$

are the upper and lower CUSUM statistics at time t_n, respectively. The parameters K_n and h control the sensitivity of the CUSUM method to detect the changepoints. Specifically, the parameter K_n is the deviation between the normalized observations y_n^*

and the standardized mean zero that we wish to detect. For example, $K_n = 0.5$ indicates that the mean shift we aim to detect for the normalized observations y_n^* is at least 0.5. Similarly, the parameter h controls the distance between the CUSUM statistics and the baseline zero we want to detect. In both simulations and real data analysis, we tune parameters K_n and h to control type-I errors during training. Once set, these parameters remain constant when identifying change points on testing samples.

A.8 Simulation Studies with Misspecified Configurations

To test the robustness of the SKF method, we perform simulated studies under misspecified conditions, using data from a GP model having a Matérn covariance function with roughness parameter being 2.5 while employing an Exponential correlation in the SKF algorithm for changepoint detection. For comparison, we also implemented the BOCPD and CUSUM methods, using the configurations from Section 2.3.1. This involves considering three types of changes, including changes in mean, variance, and correlation. For each configuration, 100 simulation experiments are conducted. In each of these, 100 observations are generated, with the initial 50 serving as training samples. The true changepoint is set at time 75, and we record the average detection delay for all methods.

Figure A.3 shows the average detection delay for the misspecified SKF method, the BOCPD method, and the CUSUM method. Notably, even when misspecified, the SKF method consistently outperforms the other methods with a lower average detection delay. The better performance of the SKF method, even with a misspecified correlation function, can likely be attributed to its ability to recognize and capture temporal correlations within the data. This ability enables the SKF method to avoid detecting false changepoints

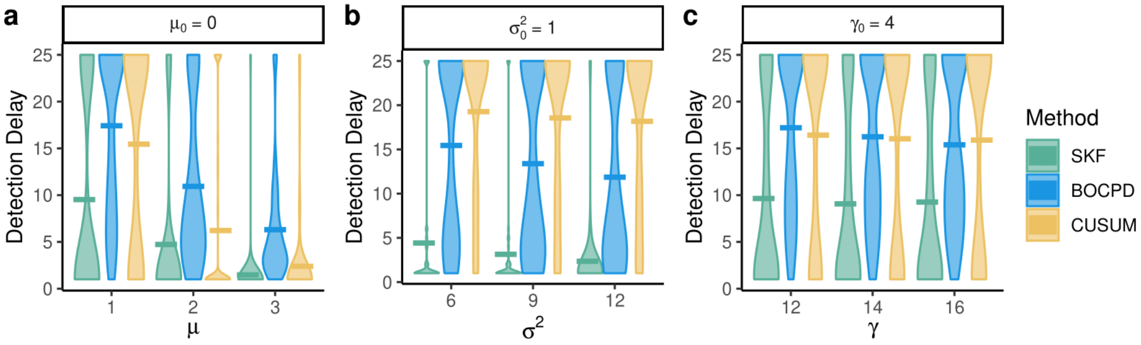


Figure A.3: Violin plots comparing average detection delay for SKF, BOCPD, and CUSUM methods for 100 simulations. The data are simulated with the Matérn correlation and the roughness parameter being 2.5, while we use **misspecified** Exponential correlation in the SKF method to detect changepoints. A method with a low average detection delay is better. μ_0, σ_0^2 and γ_0 represent pre-change parameter values, while μ, σ^2 and γ on the x-axis stand for post-change parameter values.

from large fluctuations caused by these temporal correlations, thereby reducing false detections. In contrast, methods such as BOCPD and CUSUM are not able to account for these temporal correlations, potentially leading to a higher rate of false positives. Given that we maintained a consistent type-I error rate across all methods during training, BOCPD and CUSUM required larger hazard parameters due to their increased sensitivity to false positives, resulting in higher average detection delays compared to SKF.

A.9 Definition of the Covering Metric

Define the ordered set of latent true changepoints as $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_{m_1}\}$, where τ_i takes values in time indices $1, \ldots, n$ and $\tau_i < \tau_j$ for $i < j$. This ordered set $\boldsymbol{\tau}$ induces a partition $\mathcal{G}$, separating the interval $[t_1, t_n]$ into $m_1 + 1$ disjoint sets $\boldsymbol{A} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{m_1+1}\}$. The ordered set of detected changepoints is denoted as $\hat{\boldsymbol{\tau}} = \{\hat{\tau}_1, \ldots, \hat{\tau}_{m_2}\}$, where m_2 may not be the same as m_1. The partition induces by $\hat{\boldsymbol{\tau}}$ is denoted as $\mathcal{G}'$, with the corresponding disjoint set represented as $\boldsymbol{A}' = \left\{\mathcal{A}'_1, \ldots, \mathcal{A}'_{m_2+1}\right\}$. The *covering metric* between two partitions is defined as:

$$C(\mathcal{G},\mathcal{G}') = \frac{1}{n}\sum_{i=1}^{m_1+1} |\mathcal{A}_i| \max_{1\le j\le m_2+1} J(\mathcal{A}_i,\mathcal{A}'_j), \tag{A.21}$$

where $|\mathcal{A}_i|$ represents the number of observations in $\mathcal{A}_i$, and $J(\mathcal{A}_i,\mathcal{A}'_j) = \frac{|\mathcal{A}_i\cap\mathcal{A}'_j|}{|\mathcal{A}_i\cup\mathcal{A}'_j|}$ is the Jaccard index.

A.10 An Integrated Algorithm to Detect the COVID-19 Infection

Algorithm 4 An integrated algorithm to detect COVID-19 infection

```
Require: Logit transformation of classification probabilities $\{y_t\}_{t=1}^{n}$ from a classification model, such as XGBoost.
Ensure: A set of detected changepoints $\hat{\boldsymbol{\tau}}$.
Estimate the range parameter and nugget parameter $(\hat{\gamma},\hat{\eta})$ by maximizing the marginal likelihood across all the patients using $n_0$ training time points, after integrating out the distinct mean and variance parameters for each patient.
Apply SKF in Algorithm 1 using $n_0$ training time points to control the type-I error to be 0.4%, the baseline COVID-19 positive proportion.
for $j$ in $(n_0+1):n$ of each patient do
    Run Algorithm 1 on the observations $\mathbf{y}_{(n_0+1):j}$ to obtain $\hat{C}_j$, the most recently detected changepoint before or at time $t_j$.
    When the detected changepoint $\hat{C}_j$ is within seven days before the current time $t_j$, i.e. $t_j - \hat{C}_j \le 7$, we test the following hypothesis to identify the increasing subsequence:
        $H_0 : \mu_1 = \mu_2$ vs $H_a : \mu_1 < \mu_2$,
    where $\mu_1$ and $\mu_2$ are the mean of the subsequences $\mathbf{y}_{(n_0+1):(\hat{C}_j-1)}$ and $\mathbf{y}_{\hat{C}_j:j}$. When the test statistics in Equation (A.25) of appendix fall into a rejection region, we add the value of $\hat{C}_j$ to the changepoint set $\hat{\boldsymbol{\tau}}$ and mark the 7-day period following the changepoint $\hat{C}_j$, i.e. $t \in [\hat{C}_j, \hat{C}_j + 7]$, as COVID-19 positive for this patient.
end for
```

An integrated procedure to detect the changepoint from the COVID-19 infection is summarized in Algorithm 4. We first apply a data-driven classification model to patients' clinical data, here chosen as the XGBoost method [36], which was previously found to be accurate in detecting COVID-19 among dialysis patients [30, 87] compared to a few other statistical learning algorithms. In principle, our changepoint detection method can be applied along with any statistical learning method that gives classification probabilities. We also implemented other statistical learning methods, such as logistic regression and random forests [35], for baseline comparisons. All methods were trained on longitudinal data from 20% of randomly selected patients, encompassing four million observations. For all approaches, we apply the logit transformation to the probability sequences for mapping the outcomes to the real line. We found that the estimated probabilities of infection are rarely 0 or 1. To prevent numerical errors in the logit transformation, one may replace 0 and 1 with the smallest and largest estimated probabilities within (0,1), respectively.

Second, we apply SKF to detect the change in the daily prediction probabilities of COVID-19 infection from a classification approach, chosen to be XGBoost herein. For demonstration purposes, we compared different approaches for patients with $n > 150$ samples, where the first $n_0 = 100$ samples, labeled as COVID-19 negative, are used as training data to estimate the parameters in changepoint detection approaches, and the remaining samples are used as testing data to evaluate the detection performance. We use an exponential covariance in Equation (2.4) with the shared range and nugget parameters across patients estimated by maximizing the likelihood of all patients in the training data set, which can improve estimation stability [62]. The mean and variance parameters for different patients and segments are allowed to be distinct, and these parameters are integrated out when computing marginal likelihood and predictive distributions. The distinct mean and variance parameters are flexible for modeling longitudinal observations from a large number of patients.

For both BOCPD and SKF, we utilized the county-level daily Probability of Contracting (PoC) COVID-19 [107] for specifying the hazard function, which is used to control the type I error, described in Step 2 in Algorithm 4. The time-dependent PoC quantifies the average daily COVID-19 transmission probability at the county level among susceptible individuals base on the daily infection and death counts. We found that specifying the hazard function proportional to PoC improves the detection accuracy with a constant hazard function.

Furthermore, we developed an additional screening step to detect the onset of an increasing subsequence in infection probabilities through a hypothesis test (Step 5 in Algorithm 4), as typically the increase of the probability sequences of infection should be detected. Once the detected changepoint passes this screening step, we mark the seven-day period after the detected changepoint as COVID-19 positive [106].

The integrated approach is generally applicable to detect changes from longitudinal data.

Next, we show the derivation of the test statistic in Step 5 of Algorithm 4. Denote the test dataset from time t_{n_0+1} to t_j as $\mathbf{y}_{n_0+1:j}$, where t_{n_0+1} is the start time of the testing samples and $j \geq n_0 + 1$. At the time t_j, if the estimated most recent changepoint $\hat{C}_j$ satisfies that $\hat{C}_j > n_0 + 1$ and $|t_j - \hat{C}_j| \leq 7$, we test the following hypothesis to identify if $\hat{C}_j$ is a valid changepoint, meaning that the subsequences around $\hat{C}_j$, i.e. $\mathbf{y}_{n_0+1:\hat{C}_j-1}$ and $\mathbf{y}_{\hat{C}_j:j}$, have an increasing trend. We assume that $\mathbf{y}_{n_0+1:\hat{C}_j-1} \sim \mathcal{MN}(\mu_1\mathbf{1}, \sigma^2\mathbf{K}_{n_1'})$ and $\mathbf{y}_{\hat{C}_j:j} \sim \mathcal{MN}(\mu_2\mathbf{1}, \sigma^2\mathbf{K}_{n_2'})$, where $n_1' = \hat{C}_j - n_0 - 1$, $n_2' = j - \hat{C}_j + 1$, and test the following hypothesis.

$$H_0 : \mu_1 = \mu_2 \text{ vs } H_a : \mu_1 < \mu_2.$$

The generalized least square estimators for the mean parameters are as follows.

$$\begin{aligned}\hat{\mu}_1 &= \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_1'}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \hat{\mathbf{K}}_{n_1'}^{-1} \mathbf{y}_{(n_0+1):(\hat{C}_j-1)},\\ \hat{\mu}_2 &= \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_2'}^{-1} \mathbf{1}\right)^{-1} \mathbf{1}^T \hat{\mathbf{K}}_{n_2'}^{-1} \mathbf{y}_{\hat{C}_j:j},\end{aligned}$$

where the correlation matrices $\hat{\mathbf{K}}_{n_1'}$ and $\hat{\mathbf{K}}_{n_2'}$ are determined using range and nugget parameters $\hat{\gamma}$ and $\hat{\eta}$ estimated from the training samples $\mathbf{y}_{1:n_0}$.

Under the null hypothesis, we have

$$\hat{\mu}_2 - \hat{\mu}_1 \sim \mathcal{MN}(\mu^*, (\sigma^*)^2), \tag{A.22}$$

where

$$\begin{aligned}\mu^* &= \mathbb{E}[\hat{\mu}_2] - \mathbb{E}[\hat{\mu}_1] = \mu_2 - \mu_1 = 0,\\ (\sigma^*)^2 &= \mathbb{V}[\hat{\mu}_1] + \mathbb{V}[\hat{\mu}_2]\\ &= \sigma^2 \left(\left(\mathbf{1}^T \hat{\mathbf{K}}_{n_1'}^{-1} \mathbf{1}\right)^{-1} + \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_2'}^{-1} \mathbf{1}\right)^{-1} \right). \end{aligned} \tag{A.23}$$

Therefore the test statistic has the following form when we assume the variance σ^2 and the correlation matrix $\mathbf{K}_1$ and $\mathbf{K}_2$ are known.

$$z = \frac{\hat{\mu}_2 - \hat{\mu}_1 - \mu^*}{\sqrt{(\sigma^*)^2}} = \frac{\hat{\mu}_2 - \hat{\mu}_1}{\sqrt{\sigma^2 \left(\left(\mathbf{1}^T \hat{\mathbf{K}}_{n_1'}^{-1} \mathbf{1}\right)^{-1} + \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_2'}^{-1} \mathbf{1}\right)^{-1} \right)}}, \tag{A.24}$$

Under the null hypothesis, $z \sim \mathcal{N}(0,1)$. In the real data analysis, we estimate the parameters, including the variance σ^2, the range parameter γ, and nugget parameter η in the correlation matrix $\hat{\mathbf{K}}_{n_1'}$ and $\hat{\mathbf{K}}_{n_2'}$, from the training data. Therefore, by plugging the parameters estimated from the training data into the test statistic z in Equation (A.24), we get the following test statistic.

$$z = \frac{\hat{\mu}_2 - \hat{\mu}_1}{\sqrt{\hat{\sigma}^2 \left(\left(\mathbf{1}^T \hat{\mathbf{K}}_{n_1'}^{-1} \mathbf{1} \right)^{-1} + \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_2'}^{-1} \mathbf{1} \right)^{-1} \right)}}, \tag{A.25}$$

where

$$\begin{aligned} \hat{\sigma}^2 &= \frac{1}{n-1} \left(\mathbf{y}_{1:n_0} - \mathbf{1}\hat{\mu}_0 \right)^T \hat{\mathbf{K}}_{n_0}^{-1} \left(\mathbf{y}_{1:n_0} - \mathbf{1}\hat{\mu}_0 \right), \\ \hat{\mu}_0 &= \left(\mathbf{1}^T \hat{\mathbf{K}}_{n_0}^{-1} \mathbf{1} \right)^{-1} \mathbf{1}^T \hat{\mathbf{K}}_{n_0}^{-1} \mathbf{y}_{1:n_0} \end{aligned}$$

The range and nugget parameters $(\hat{\gamma}, \hat{\eta})$ are plugged into correlation matrices to obtain $\hat{\mathbf{K}}_{n_1'}$ and $\hat{\mathbf{K}}_{n_2'}$. As the sample size in the training period is large, we approximate the threshold of the test statistic by normal approximation. We assume normal approximation for the threshold of the test statistic z when comparing the means of two long subsequences, $\mathbf{y}_{(n_0+1):(\hat{C}_j-1)}$ and $\mathbf{y}_{\hat{C}_j:j}$. The robustness of our approach is maintained by the hazard parameter in the SKF method, ensuring Type I error control irrespective of the threshold choice.

A.11 Sensitivity Analysis

We show the sensitivity analysis on the definition of the COVID-19 positive period. In the COVID-19 analysis described in Section 2.4, we defined the positive period from day -2 to day 7, with day 0 being the date the patient received a COVID-19 PCR test. This choice was driven by the average incubation period of COVID-19 infection, which is approximately 3 days [104, 105] before symptoms onset. However, this incubation period may vary among patients.

Table A.2: Comparisons of the statistical learning methods and online changepoint detection methods, including CUSUM, BOCPD, and SKF on COVID-19 patient predictions with the baseline positive rate of 0.4%. The COVID-19 positive period starts on day -4.

	Precision	Recall	F1-score	Detection Delay
Logistic Regression	0.055	0.133	0.077	1.538
Random Forest	0.087	0.125	0.087	2.086
XGBoost	0.082	0.179	0.113	1.799
CUSUM	0.032	0.020	0.025	3.142
BOCPD	0.200	0.154	0.174	4.856
SKF	0.268	0.145	0.188	3.886
SKF with screening	0.231	0.164	0.192	2.395

In this sensitivity analysis, we expanded the definition of the COVID-19 positive period to be from day -4 to day 7. The results of this analysis are shown in Table A.2. Notably, the comparative performance of the SKF and BOCPD methods remains consistent with our findings in Section 2.4. Specifically, the SKF and BOCPD methods continue to outperform all the statistical learning methods and the CUSUM method in terms of the F1-score. It could be attributed to their ability to utilize retrospective information through the recursive computation of the predictive distribution at each time step, as described in Equation (2.7). Furthermore, consistent with the conclusion we draw from Table 2.1, the SKF method displayed a lower detection delay and a higher F1-score than the BOCPD method, which can be attributed to SKF's ability to capture the temporal correlations within time segments. These findings suggest that the performance of the changepoint methods is not sensitive to the specific choice of the start date for the COVID-19-positive period.

Table A.3: Covering metric comparison for SKF, BOCPD and CUSUM. The highest values in each row are in bold type.

Dataset	SKF	BOCPD	CUSUM
brent_spot	**0.602**	0.383	0.535
businv	**0.814**	0.471	0.510
construction	**0.719**	0.398	0.506
iceland_tourism	0.624	**0.685**	0.617
jfk_passengers	0.843	**0.847**	0.687
lga_passengers	**0.591**	0.456	0.496
quality_control_4	**0.561**	0.552	0.442
seatbelts	0.585	**0.635**	0.573

A.12 More Numerical Comparisons on Real-world Datasets

In this section, we apply the SKF method and other classic online change point detection methods on a set of benchmark datasets in [80], which includes 37 time series from various domains with ground truth change points annotated by five human annotators. We select eight datasets with a relatively large number of observations and relatively strong temporal correlations to compare our SKF method with other changepoint detection methods. The training data is created from the initial 50 observations of each series, ensuring no changepoints are present. The distribution parameters for SKF and BOCPD are estimated by maximizing the likelihood function on the training data. The hazard parameters for all three changepoint methods are selected so that there is no changepoint detected on the training data. All estimated parameters are then applied to the test data to detect changepoints. As each time series may contain multiple changepoints, we utilize the covering metric defined in Section (2.3.2) to evaluate the performance of the CPD methods. Although all the real datasets are already collected at the time of testing, we apply the SKF as though data points are observed sequentially.

Table A.3 shows the covering metric of the SKF and other CPD methods on the eight real datasets. The SKF method has the best covering metric in five out of eight datasets, indicating its capability of detecting changepoints on temporally correlated data. We highlight the functionality of the SKF method by discussing the `businv` dataset. This dataset, sourced from the U.S. Census Bureau, contains the monthly total business inventories in U.S. dollars from 1992 to 2019. It tracks the combined value of goods held by manufacturers, wholesalers, and retailers each month, serving as an important indicator of the health of the supply chain. Five human annotators in [80] identified seven distinct actual change points within this data, which fall into two categories: (1) **dot-com crush**: changepoints in 2002 marks the end of the dot-com bubble; (2) **financial crisis**: the remaining change points in 2008 indicate the decline in total business inventories caused by the financial crisis.

We fitted a GP model on data from 1992 to 1996 to estimate the variance parameter σ^2, range parameter γ, and nugget parameter η. With these estimations, we employed the SKF method in the following years to detect the changepoints. The estimated changepoints, shown as red crossings in panel a of Figure A.4, match well with the actual changepoints related to the two major events, the dot-com crash, and the financial crisis. Further, panel b of this figure shows the posterior distribution of the most recent changepoints, with the red dashed line indicating the location of the changepoints at each time. Regarding the results in Table A.3, the SKF on the business inventory dataset outperforms the BOCPD and CUSUM methods for the covering metric, which can be attributed to the model of temporal correlation in the SKF method which was not considered in BOCPD and CUSUM models.

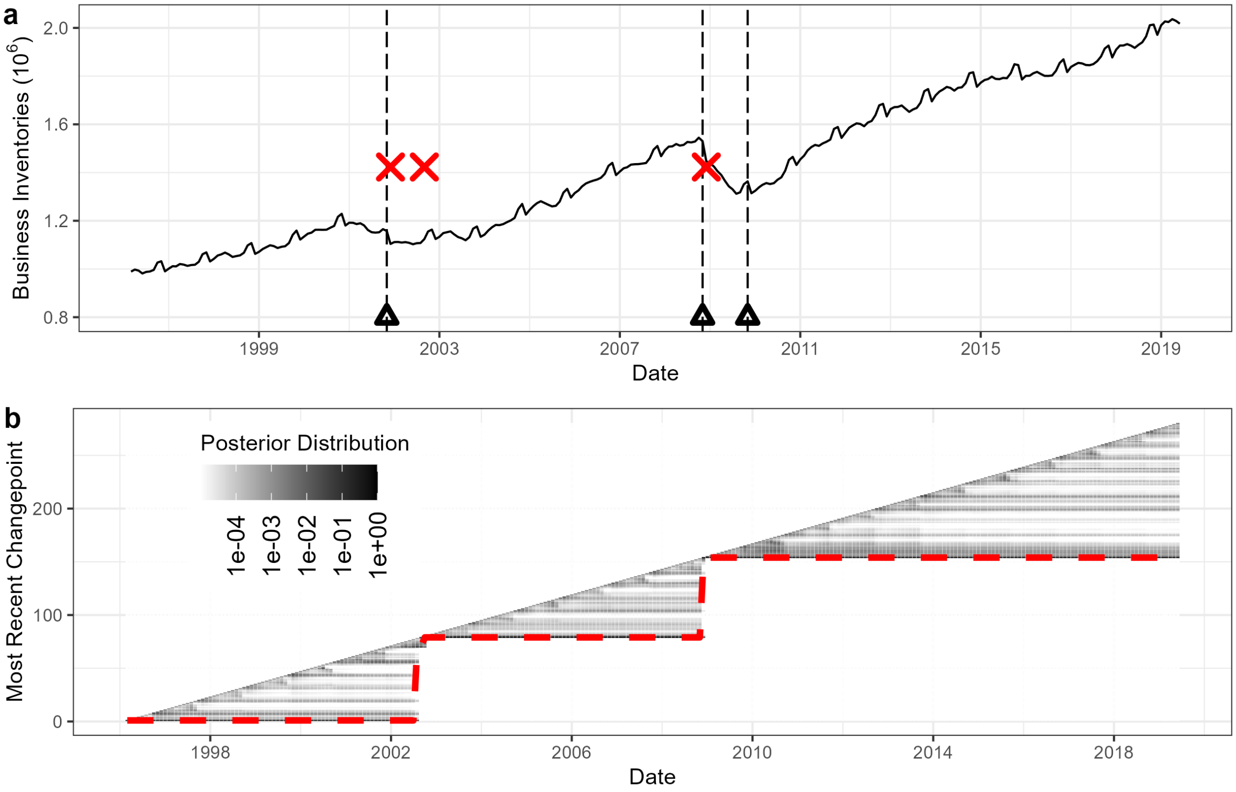


Figure A.4: The outputs of SKF on the US monthly total business inventory data from 1992 to 2019. Panel a shows the monthly total business inventory data. The actual changepoints are marked by the black dashed lines. The estimated changepoints by SKF are marked by the red crossings. Panel b displays the posterior probability matrix of the most recent changepoints. The red dashed curve gives the estimated most recent changepoints with the maximum *a posterior* (MAP) estimator at each time step.

Appendix B

Appendix of Chapter 3

This appendix contains two parts. In the first part, We discuss the details of model parameter specification and conduct a sensitivity analysis. The forecast algorithm and numerical comparison of different approaches are introduced in the second part.

B.1 Model parameter specification and sensitivity analysis

We discuss the choice of the model parameters and their sensitivity analysis. The following parameters of the SIRDC model were specified based on previous studies.

- The death rate or the infection fatality ratio (δ) that measures the proportion of death among all infected individuals. We assume $\delta = 0.66\%$ following [117].
- The inverse of the number of days an infectious individual can transmit the COVID-19 (γ). The average time of a COVID-19 patient to transmit disease is assumed to be 5 days in [116], indicating that $\gamma = 0.2$. Another evidence comes from the study of the incubation period. The latent period (exposed but not contagious)

for COVID-19 is found to be 3.69 days on average [109] and the mean incubation period (time from infection to onset of symptoms) is 5.2 days [159], meaning that the infectious period is around 1.5 days before the onset of symptom. The diagnostic test could take less than one day to up to a week. We thus assume 3.5 days to get the result of a diagnostic test on average. The total infectious period is around 5 days.

- The inverse of the number of dates for resolving case to get resolved (θ). According to the CDC report [160], for mild and moderate symptom, the replication-competent virus has not been recovered after 10 days following symptom onset, indicating the individuals remains infectious no longer than 10 days after symptom onset. The infectious period could be as long as 20 days for patients with more severe illness from COVID-19 infection. Since a majority of the COVID-19 infections are mild to moderate, we assume the infectious period to be 13.5 days, and after reducing 3.5 days from onset of the symptom to become resolving (after quarantine or hospitalization), it takes around 10 days for a resolving case to resolved on average.

We conduct a sensitivity analysis to examine the change of the estimation in 4 different configurations.

- (**Configuration 1**) $(\gamma, \theta, \delta) = (0.2, 0.1, 0.0066)$, the default parameter set.
- (**Configuration 2**) $(\gamma, \theta, \delta) = (0.14, 0.1, 0.0066)$. The average length of infectious period changes from 5 days to $\frac{1}{0.14} \approx 7$ days, whereas other parameters are held unchanged.
- (**Configuration 3**) $(\gamma, \theta, \delta) = (0.2, 0.067, 0.0066)$. The average length of resolving period changes from 10 days to $\frac{1}{0.067} = 15$ days, whereas other parameters are held

unchanged.

- (**Configuration 4**) $(\gamma, \theta, \delta) = (0.2, 0.1, 0.0075)$. The infection fatality ratio changes from 0.66% to 0.75%, whereas other parameters are held unchanged.

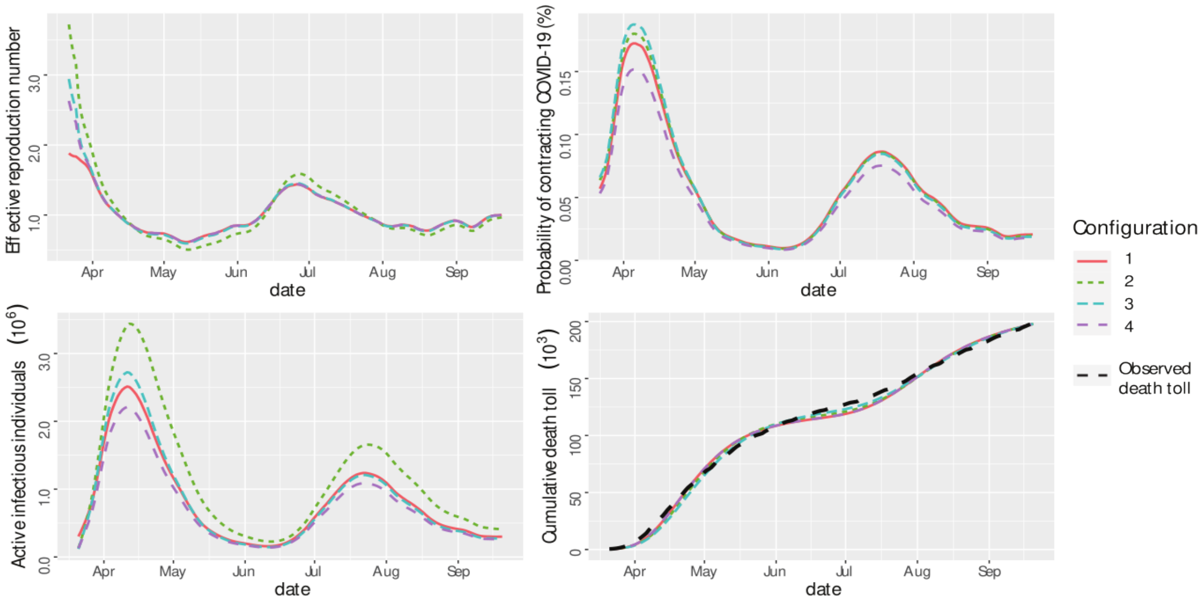


Figure B.1: sensitivity analysis for 4 configurations of the SIRDC model parameters. Part a-d shows the estimated effective reproduction number, PoC SARS-CoV-2, the number of active infectious individuals, and cumulative death toll, respectively.

After specifying the parameters (γ, θ, δ), the transmission rate $\beta(t)$ can be obtained from algorithm 1. Figure B.1 gives result of the sensitivity analysis. First, we found the estimated death toll for 4 scenarios is almost the same (part d in Figure B.1). Extending the infectious period from 5 to 7 days (Configuration 2) increases the number of active infectious individuals and effective reproduction number shown in part a and part c in Figure B.1, respectively. Consequently, the peak of average daily PoC SARS-CoV-2 slightly increases in the first wave, whereas the scale of increase is smaller than the change in the effective reproduction number and active infectious individuals. The average daily PoC SARS-CoV-2 has almost no change in other periods, indicating that the length of the average infectious period has almost no influence of our estimation on PoC SARS-CoV-2 for most of the days.

Second, when the average length of the resolving period changes from 10 to 15 days, the peak of PoC SARS-CoV-2, effective reproduction number, and the number of active infectious individuals increases in the first wave, whereas these quantities remain largely unchanged for the rest of the days (part a-c in Figure B.1). The result indicates that the average length of the resolving period also barely affects the estimated characteristics of COVID-19 progression for most of the days.

When the death rate increases from 0.66% to 0.75%, the effective reproduction number seems to have almost no change (part a in Figure B.1), whereas the PoC SARS-CoV-2 and the number of active infectious individuals (figure B.1 part b-c) both reduce. This is because when the death rates increase, the estimated number of individuals infected decreases, as the death toll is observed (and thus fixed). The death rate is a key parameter to calibrate, and studies of the prevalence of SARS-CoV-2 antibodies based on serology tests [113] can be used to estimate the death rate in each state.

In conclusion, parameter values of the average lengths of the infectious period and the resolving period barely change the COVID-19 progression characteristics for most of the days, including the fitted death toll. On the other hand, we found that the number of active infectious individuals and the daily PoC SARS-CoV-2 depend critically on the death rate parameter. Further studies of prevalence would be useful for estimating the death rate parameter in different regions.

```
Algorithm 5 Ensemble forecast and uncertainty assessment.
Require: $\mathbf{c}^o_{i,j}$, $\mathbf{D}_{i,j}$, $\mathbf{p}_i$, $\mathbf{c}^o_i$, and $\mathbf{D}_i$.
Ensure: Estimates of county-level epidemiological compartments $\hat{\boldsymbol{\beta}}_{i,j}$, $\hat{\mathbf{S}}_{i,j}$, $\hat{\mathbf{I}}_{i,j}$, $\hat{\mathbf{R}}_{i,j}$,
    $\hat{\mathbf{C}}_{i,j}$, forecast $\hat{\mathbf{D}}^*_{i,j}$, where $\hat{\mathbf{D}}^*_{i,j} := \left(\hat{D}_{i,j}(T_{i,j}+1), \ldots, \hat{D}_{i,j}(T_{i,j}+T^*)\right)^T$, and the uncertainty assessment of the compartments.
    Step 1 Conduct a three-parameter constrained optimization treating state-level power parameter $\alpha_i$ unknown to minimize the loss function in equation (9) using $\mathbf{p}_i$, $\mathbf{c}^o_i$ and $\mathbf{D}_i$.
    Step 2 For each county, set initial values $I_{i,j}(1) = R_{i,j}(1) = 1,000$, $C_{i,j}(1) = 0$ and $D_{i,j}(1)$ to be the observed death toll on day 1. Find the optimized values of $I_{i,j}(1)$ and $R_{i,j}(1)$ to minimize equation (9).
    Step 3 Simulate $S = 500$ samples of the observed confirmed cases sampled from the predictive distribution of a GP model of the observed confirmed cases. For each sample, obtain the other compartments and time-dependent transmission rate by equation (1)-(5) and algorithm 1 using the estimate of the initial values.
    Step 4 Extrapolate the time-dependent transmission rate parameters from a GP model for each sample and obtain $S = 500$ samples of the output death toll of the SIRDC at the forecast period.
    Step 5 Sample the residuals from the predictive distribution in Equation (11) in the main manuscript at the forecast period and obtain $S = 500$ samples of the ensemble forecast for the death toll. Compute the mean for forecast and 95% predictive interval to quantify uncertainty of forecast.
```

B.2 Algorithm of Forecast and Numerical Comparison

An overview of our algorithm for forecast and uncertainty assessment is given in algorithm 5, where inputs are the county-level observed cumulative number of confirmed cases $\mathbf{c}^o_{i,j} = (c^o_{i,j}(1), ..., c^o_{i,j}(T_{i,j}))^T$, the county-level observed cumulative death toll $\mathbf{D}_{i,j}$, the state-level test positive rate $\mathbf{p}_i = (p_i(1), ..., p_i(T_i))^T$, state-level confirmed cases $\mathbf{c}^o_i = (c^o_i(1), ..., c^o_i(T_i))^T$ and state-level death toll $\mathbf{D}_i = (D_i(1), ..., D_i(T_i))^T$.

To evaluate the performance of different approaches, we implement 7-day and 21-day forecasts on 2,277 US counties with a training period from 21 March 2020 to 20 September 2020, and with the forecast period starting from 21 September 2020. To compare the prediction performance of different methods, we computed the rooted mean square error (RMSE), the proportion of the observations covered in the 95% predictive interval ($P_{CI}(95\%)$) and length of the 95% confidence interval ($L_{CI}(95\%)$), defined as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^k \sum_{j=1}^{n_i} \sum_{s\in\mathbf{t}^*} (\hat{D}_{i,j}(s) - D_{i,j}(s))^2}{\sum_{i=1}^k n_i T^*}}$$

$$P_{CI}(95\%) = \frac{1}{\sum_{i=1}^k n_i T^*} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \sum_{s\in\mathbf{t}^*} \mathbb{1}_{\{D_{i,j}(s)\in CI_{i,j,s}(95\%)\}}$$

$$L_{CI}(95\%) = \frac{1}{\sum_{i=1}^k n_i T^*} \sum_{i=1}^{k} \sum_{j=1}^{n_i} \sum_{s\in\mathbf{t}^*} \text{length}\{CI_{i,j,s}(95\%)\}$$

where $\mathbf{t}^* := (T_{i,j}+1, \ldots, T_{i,j}+T^*)$, $T^* = 7$ and $T^* = 21$ for the 7-day forecast and 21-day forecast, respectively. A model with small RMSE, $P_{CI}(95\%)$ close to the nominal 95% and small $L_{CI}(95\%)$ is precise for forecast and uncertainty assessment.

A comparison between our approach and the other three approaches is recorded in

Prediction period	Method	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$
7 days	SIRDC+GP	**3.04**	95.06%	23.05
	SIRDC	4.12	/	/
	GP without linear trend	3.18	91.29%	**4.82**
	GP with linear trend	4.36	88.28%	5.51
21 days	SIRDC+GP	**6.81**	93.46%	28.37
	SIRDC	7.79	/	/
	GP without linear trend	7.20	92.14%	**11.74**
	GP with linear trend	11.93	76.94%	10.18

Table B.1: 7-day and 21-day forecast in 2,277 US counties with training period from 21 March 2020 to 20 September 2020 and with prediction period starting from 21 September 2020. Four methods are compared. Our proposed approach that combines the SIRDC model and a zero-mean GP to model the residuals is denoted as SIRDC+GP. Second, the death forecast by SIRDC model is denoted as SIRDC, which contains Steps 1 and 2 in the algorithm 5 and provides point projection of the death toll. Third, a GP with a constant mean function is denoted as GP without linear trend, which equivalently replaces the SIRDC model of a constant mean parameter estimated by the data for each county. The fourth model, denoted as GP with linear trend, is the same as the third method, except that the mean of GP contains a constant mean and a linear trend of time with two linear coefficient parameters estimated from the data. The best method under each criterion is highlighted.

Table B.1. Our approach (denoted in SIRDC+GP) has the lowest RMSE among 4 methods considered herein. Approximately 95% of the held-out death toll are covered by the 95% predictive interval by our approach, indicating our uncertainty assessment is accurate. Although other approaches produce a shorter length of the predictive interval, the number of held-out observations in the 95% predictive interval is smaller than ours. Therefore, combining the SIRDC model and GP for modeling the residuals may improve the predictive accuracy for forecasting COVID-19 associated death toll at US counties, compared to the one using the SIRDC model or the GP model alone.

Appendix C

Appendix of Chapter 4

This appendix contains three sections. The proof of Section 4.2 is given in Section C.1. The additional numerical results for the simulated studies and real applications are given in Section C.2 and Section C.3, respectively.

C.1 Proofs for Section 4.2

C.1.1 Auxiliary facts

1. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices,

$$(\mathbf{A} \otimes \mathbf{B})^T = (\mathbf{A}^T \otimes \mathbf{B}^T);$$

   further assuming $\mathbf{A}$ and $\mathbf{B}$ are invertible,

$$(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}.$$

2. Let $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ be the matrices such that the products $\mathbf{AC}$ and $\mathbf{BD}$ are matrices,
$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}).$$

3. For matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$,
$$(\mathbf{C}^T \otimes \mathbf{A})\text{vec}(\mathbf{B}) = \text{vec}(\mathbf{ABC});$$
further assuming $\mathbf{A}^T\mathbf{B}$ is a matrix,
$$\text{tr}(\mathbf{A}^T\mathbf{B}) = \text{vec}(\mathbf{A})^T\text{vec}(\mathbf{B}).$$

4. For any invertible $n \times n$ matrix $\mathbf{C}$,
$$|\mathbf{C} + \mathbf{AB}| = |\mathbf{C}||\mathbf{I}_n + \mathbf{BC}^{-1}\mathbf{A}|.$$

C.1.2 Proofs for Section 4.2.1

The following denotation are used in the proof: $\mathbf{Y}_{-M} = \mathbf{Y} - \mathbf{M}$, $\mathbf{Y}_{v,-M} = vec(\mathbf{Y} - \mathbf{M})$, $\mathbf{Z}_{vt} = vec(\mathbf{Z}^T)$ and $\mathbf{A}_v = [\mathbf{I}_n \otimes \mathbf{a}_1, ..., \mathbf{I}_n \otimes \mathbf{a}_d]$. Let $\mathbf{\Sigma}_v$ be an $nd \times nd$ matrix where the lth diagonal block is $\mathbf{\Sigma}_l$. Denote etr(.) = exp(tr(.)).

Proof of Equation (4.4): Denote $C_Y = (2\pi\sigma_0^2)^{-\frac{nk}{2}} \prod_{l=1}^{d} |\mathbf{\Sigma}_l/\sigma_0^2 + \mathbf{I}_k|^{-1/2}$. Directly marginalizing out $\mathbf{Z}$, one has

$$\begin{aligned}
&p(\mathbf{Y} \mid \mathbf{\Theta})\\
=&C_Y \exp\left(-\frac{\mathbf{Y}_{v,-M}^T \left(\mathbf{I}_{nk} - \sum_{l=1}^{d}(\sigma_0^2\mathbf{\Sigma}_l^{-1} + \mathbf{I}_n)^{-1} \otimes (\mathbf{a}_l\mathbf{a}_l^T)\right) \mathbf{Y}_{v,-M}}{2\sigma_0^2}\right)\\
=&C_Y \exp\left(-\frac{\mathbf{Y}_{v,-M}^T\mathbf{Y}_{v,-M} - \mathbf{Y}_{v,-M}^T \sum_{l=1}^{d} \mathrm{vec}(\mathbf{a}_l\mathbf{a}_l^T\mathbf{Y}_{v,-M}(\sigma_0^2\mathbf{\Sigma}_l^{-1} + \mathbf{I}_n)^{-1})}{2\sigma_0^2}\right)\\
=&C_Y \mathrm{etr}\left(-\frac{\mathbf{Y}_{-M}^T\mathbf{Y}_{-M} - \sum_{l=1}^{d} \tilde{\mathbf{y}}_l\tilde{\mathbf{y}}_l^T(\sigma_0^2\mathbf{\Sigma}_l^{-1} + \mathbf{I}_n)^{-1}}{2\sigma_0^2}\right)\\
=&C_Y \exp\left(-\frac{\sum_{l=1}^{d} \tilde{\mathbf{y}}_l^T(\mathbf{\Sigma}_l/\sigma_0^2 + \mathbf{I}_n)^{-1}\tilde{\mathbf{y}}_l + \sum_{l=d+1}^{n_1} \tilde{\mathbf{y}}_l^T\tilde{\mathbf{y}}_l}{2\sigma_0^2}\right),
\end{aligned}$$

where the first equation is based on Lemma 1 and the Woodbury matrix identity (to compute the normalizing constant C_Y); the second and third equations are from fact 3; the fourth equation is from Woodbury matrix identity. The Equation (4.4) follows immediately.

□

Proof of Corollary 1: The proof is implied by the proof of Theorem 4 in [123]. For completeness of this article, we include the proof below.

From Equation (4.1) and Equation (4.2), we have

$$\begin{aligned}
p(\mathbf{Z}_{vt} \mid \mathbf{Y}, \mathbf{\Theta}) &\propto \exp\left(\frac{(\mathbf{Y}_{v,-M} - \mathbf{A}_v\mathbf{Z}_{vt})^T(\mathbf{Y}_{v,-M} - \mathbf{A}_v\mathbf{Z}_{vt})}{2\sigma_0^2}\right) \exp\left(-\frac{1}{2}\mathbf{Z}_{vt}^T\mathbf{\Sigma}_v^{-1}\mathbf{Z}_{vt}\right)\\
&\propto \exp\left\{-\frac{1}{2}(\mathbf{Z}_{vt} - \boldsymbol{\mu}_{Z_{vt}})^T \left(\frac{\mathbf{A}_v^T\mathbf{A}_v}{\sigma_0^2} + \mathbf{\Sigma}_v^{-1}\right)(\mathbf{Z}_{vt} - \boldsymbol{\mu}_{Z_{vt}})\right\},
\end{aligned}$$

where $\boldsymbol{\mu}_{Z_{vt}} = (\mathbf{A}_v^T\mathbf{A}_v + \sigma_0^2\boldsymbol{\Sigma}_v^{-1})^{-1}\mathbf{A}_v^T\mathbf{Y}_{v,-M}$. Note $\mathbf{A}_v^T\mathbf{A}_v = \mathbf{I}_{nd}$, from which we have

$$\mathbf{Z}_{vt} \mid \mathbf{Y}, \boldsymbol{\Theta} \sim \mathrm{MN}\left(\boldsymbol{\mu}_{Z_{vt}}, \left(\frac{1}{\sigma_0^2}\mathbf{I}_{n_1 n_2} + \boldsymbol{\Sigma}_v^{-1}\right)^{-1}\right). \tag{C.1}$$

Based on vectorization, one has

$$\begin{aligned}
\boldsymbol{\mu}_{Z_{vt}} &= \begin{pmatrix} \left(\sigma_0^2\boldsymbol{\Sigma}_1^{-1} + \mathbf{I}_n\right)^{-1} \otimes \mathbf{a}_1^T \\ \vdots \\ \left(\sigma_0^2\boldsymbol{\Sigma}_d^{-1} + \mathbf{I}_n\right)^{-1} \otimes \mathbf{a}_d^T \end{pmatrix} \mathrm{vec}(\mathbf{Y}) = \begin{pmatrix} \mathrm{vec}\left(\mathbf{a}_1^T\mathbf{Y}_{-M}\left(\sigma_0^2\boldsymbol{\Sigma}_1^{-1} + \mathbf{I}_n\right)^{-1}\right) \\ \vdots \\ \mathrm{vec}\left(\mathbf{a}_d^T\mathbf{Y}_{-M}\left(\sigma_0^2\boldsymbol{\Sigma}_d^{-1} + \mathbf{I}_n\right)^{-1}\right) \end{pmatrix} \\
&= \mathrm{vec}\begin{pmatrix} \mathbf{a}_1^T\mathbf{Y}_{-M}\left(\sigma_0^2\boldsymbol{\Sigma}_1^{-1} + \mathbf{I}_n\right)^{-1} \\ \vdots \\ \mathbf{a}_d^T\mathbf{Y}_{-M}\left(\sigma_0^2\boldsymbol{\Sigma}_d^{-1} + \mathbf{I}_n\right)^{-1} \end{pmatrix}^T .
\end{aligned} \tag{C.2}$$

Note that the covariance matrix of $\boldsymbol{\mu}_{Z_{vt}}$ is a block diagonal matrix. The results follow by Equation (C.2) and the Woodbury matrix identity.

□

C.1.3 Proofs for Section 4.2.2

Note $\mathbf{A}_F = [\mathbf{A}_s, \mathbf{A}_c] = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{n_1}]$, where $\mathbf{A}_c$ is an $n_1 \times (n_1 - d)$ matrix of the orthogonal complement of $\mathbf{A}_s$. We need the following lemma to prove Theorem 2.

Lemma 4 *After marginalizing out the factors $\mathbf{Z}$, we have the marginal posterior distribution of the transformed regression coefficients,*

1. *(Marginal distribution of transformed row regression coefficients). Assume $\mathbf{M} = \mathbf{H}_1\mathbf{B}_1$ and the objective prior $\pi(\mathbf{B}_1) \propto 1$ for $\mathbf{B}_1$. Let $\tilde{\mathbf{B}}_1 = [\tilde{\boldsymbol{b}}_{1,1}, ..., \tilde{\boldsymbol{b}}_{1,n_1}] = \mathbf{B}_1^T\mathbf{H}_1^T\mathbf{A}_F$ be an $n_2 \times n_1$ matrix of transformed coefficients. Assume the marginal*

posterior distribution of $\tilde{\mathbf{B}}_1$ *follows*

$$p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1}) = \prod_{l=1}^{d} \mathcal{PN}(\tilde{\boldsymbol{b}}_{1,l}; \tilde{\mathbf{y}}_l, \tilde{\mathbf{\Sigma}}_l) \prod_{l=d+1}^{n_1} \mathcal{PN}(\tilde{\boldsymbol{b}}_{1,l}; \tilde{\mathbf{y}}_l, \sigma_0^2 \mathbf{I}_{n_2}), \tag{C.3}$$

where $\tilde{\mathbf{y}}_l$ *is defined in equation (4.4) and* $\tilde{\mathbf{\Sigma}}_l$ *is defined in corollary 1. Then we can sample* $(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1})$ *by* $(\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_\mathbf{1}^T$*, where* $\tilde{\mathbf{B}}_\mathbf{1}^T$ *are sampled from the* $p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1})$ *in equation (C.3).*

2. *(Marginal distribution of transformed column regression coefficients). Assume* $\mathbf{M} = (\mathbf{H}_2\mathbf{B}_2)^T$ *and the objective prior* $\pi(\mathbf{B}_2) \propto 1$ *for the regression parameters* $\mathbf{B}_2$*. Let* $\tilde{\mathbf{B}}_2 = [\tilde{\boldsymbol{b}}_{2,1}, ..., \tilde{\boldsymbol{b}}_{2,n_1}] = \mathbf{B}_2\mathbf{A}_F$ *be a* $q_2 \times n_1$ *matrix. The marginal posterior distribution of* $\tilde{\mathbf{B}}_2$ *follows*

$$p(\tilde{\mathbf{B}}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_2}) = \prod_{l=1}^{n_1} \mathcal{PN}(\tilde{\mathbf{b}}_{2,l}; \boldsymbol{\mu}_{\tilde{b}_{2,l}}, \mathbf{\Sigma}_{\tilde{b}_{2,l}}), \tag{C.4}$$

where $\boldsymbol{\mu}_{\tilde{b}_{2,l}} = (\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\tilde{\mathbf{y}}_l$ *and* $\mathbf{\Sigma}_{\tilde{b}_{2,l}} = (\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}$ *for* $l = 1, ..., d$*;* $\boldsymbol{\mu}_{\tilde{b}_{2,l}} = (\mathbf{H}_2^T\mathbf{H}_2)^{-1}\mathbf{H}_2^T\tilde{\mathbf{y}}_l$ *and* $\mathbf{\Sigma}_{\tilde{b}_{2,l}} = \sigma_0^2(\mathbf{H}_2^T\mathbf{H}_2)^{-1}$ *for* $l = d+1, ..., n_1$*.*

Proof of Lemma 4:

1. (Marginal distribution of transformed row regression coefficients).

   Denote $(\mathbf{B}_1^{aug}) = [\mathbf{B}_1^T, \tilde{\mathbf{B}}_{1,(q_1+1):n_1}]^T$, where $\tilde{\mathbf{B}}_{1,(q_1+1):n_1}$ are the last $n_1 - q_1$ columns of $\tilde{\mathbf{B}}_1$. Denote $p_{trans}(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1})$ and $p_{trans}(\mathbf{B}_1^{aug} \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1})$ the transformed marginal posterior distribution of $\mathbf{B}_1$ and $\mathbf{B}_1^{aug}$ derived by transforming $p(\tilde{\mathbf{B}}_1 \mid$

$\mathbf{Y}, \boldsymbol{\Theta}_{-B_1})$ in Equation (C.3). We have

$$\begin{aligned} p_{trans}(\mathbf{B}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1}) &\propto p_{trans}(\mathbf{B}_1^{aug} \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1}) = p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1}) \left| \frac{d\tilde{\mathbf{B}}_1}{d\mathbf{B}_1^{aug}} \right| \\ &\propto \exp\left\{ -\frac{1}{2} \sum_{l=1}^{d} (\tilde{\mathbf{b}}_{1,l} - \tilde{\mathbf{y}}_l)^T \tilde{\boldsymbol{\Sigma}}_l^{-1} (\tilde{\mathbf{b}}_{1,l} - \tilde{\mathbf{y}}_l) - \frac{1}{2\sigma_0^2} \sum_{l=d+1}^{n_1} (\tilde{\boldsymbol{b}}_{1,l} - \tilde{\mathbf{y}}_l)^T (\tilde{\boldsymbol{b}}_{1,l} - \tilde{\mathbf{y}}_l) \right\} \\ &\propto \exp\left\{ -\frac{1}{2} \sum_{l=1}^{d} \mathbf{a}_l^T (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1) \tilde{\boldsymbol{\Sigma}}_l^{-1} (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)^T \mathbf{a}_l \right. \\ &\qquad \left. - \frac{1}{2\sigma_0^2} \sum_{l=d+1}^{n_1} \mathbf{a}_l^T (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)(\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)^T \mathbf{a}_l \right\}, \end{aligned}$$

where the last line is the same as the posterior distribution of $\mathbf{B}_1$ based on the marginal likelihood in Equation (4.4) and the prior distribution $\pi(\mathbf{B}_1) \propto 1$. Thus if one sample $\tilde{\mathbf{B}}_{\mathbf{1}}$ from Equation (C.3), one can obtain the sample for $(\mathbf{B}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1})$ through $(\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_{\mathbf{1}}^T$.

2. (Marginal distribution of transformed column regression coefficients).

   Since $\pi(\mathbf{B}_2) \propto 1$ is a Jeffreys prior, and $\tilde{\mathbf{B}}_2$ is a linear transformation of $\mathbf{B}_2$ with the same dimension, we have $\pi(\tilde{\mathbf{B}}_2) \propto 1$.

   Based on the marginal likelihood in Equation (4.4) and the prior distribution, the posterior distribution of $\tilde{\mathbf{B}}_2$ follows:

$$\begin{aligned} & p(\tilde{\mathbf{B}}_2 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_2}) \\ &\propto \exp\left\{ -\frac{1}{2} \sum_{l=1}^{d} \mathbf{a}_l^T (\mathbf{Y} - \mathbf{B}_2^T\mathbf{H}_2^T) \tilde{\boldsymbol{\Sigma}}_l^{-1} (\mathbf{Y} - \mathbf{B}_2^T\mathbf{H}_2^T)^T \mathbf{a}_l \right. \\ &\qquad \left. - \frac{1}{2\sigma_0^2} \sum_{l=d+1}^{n_1} \mathbf{a}_l^T (\mathbf{Y} - \mathbf{B}_2^T\mathbf{H}_2^T)(\mathbf{Y} - \mathbf{B}_2^T\mathbf{H}_2^T)^T \mathbf{a}_l \right\} \end{aligned}$$

$$
\begin{aligned}
&\propto \exp\left\{-\frac{1}{2}\sum_{l=1}^{d}(\tilde{\mathbf{y}}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})^T\tilde{\boldsymbol{\Sigma}}_l^{-1}(\tilde{\mathbf{y}}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})\right.\\
&\qquad\qquad\left.-\frac{1}{2\sigma_0^2}\sum_{l=d+1}^{n_1}(\tilde{\mathbf{y}}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})^T(\tilde{\mathbf{y}}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})\right\}\\
&\propto \exp\left\{-\frac{1}{2}\sum_{l=1}^{d}\left(\tilde{\mathbf{b}}_{2,l} - \boldsymbol{\mu}_{\tilde{b}_{2,l}}\right)^T\mathbf{H}_2^T\tilde{\boldsymbol{\Sigma}}_l^{-1}\mathbf{H}_2\left(\tilde{\mathbf{b}}_{2,l} - \boldsymbol{\mu}_{\tilde{b}_{2,l}}\right)\right.\\
&\qquad\qquad\left.-\frac{1}{2\sigma_0^2}\sum_{l=d+1}^{n_1}\left(\tilde{\mathbf{b}}_{2,l} - \boldsymbol{\mu}_{\tilde{b}_{2,l}}\right)^T\mathbf{H}_2^T\mathbf{H}_2\left(\tilde{\mathbf{b}}_{2,l} - \boldsymbol{\mu}_{\tilde{b}_{2,l}}\right)\right\},
\end{aligned}
$$

from which Equation (C.4) follows.

□

We are ready to prove Theorem 2.

Proof of Theorem 2: After marginalizing out $\mathbf{Z}$, we have

1. (Row regression coefficients).

   From Lemma 4, the posterior mean of $(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1})$ is $\mathbf{Y}^T\mathbf{A}_F$, where $\mathbf{A}_F := [\mathbf{A}_s, \mathbf{A}_c]$. We denote the centered $\tilde{\mathbf{B}}_1$ by $\tilde{\mathbf{B}}_{1,0} = [\tilde{\mathbf{B}}_{1,0,s}, \tilde{\mathbf{B}}_{1,0,c}] = \tilde{\mathbf{B}}_1 - \mathbf{Y}^T\mathbf{A}_F$, where $\tilde{\mathbf{B}}_{1,0,s}$ is the first d columns of $\tilde{\mathbf{B}}_{1,0}$ and $\tilde{\mathbf{B}}_{1,0,c}$ is the last $(n_1 - d)$ columns of $\tilde{\mathbf{B}}_{1,0}$. Let $\tilde{\mathbf{b}}_{1,0,l}$ be the l-th column of $\tilde{\mathbf{B}}_{1,0}$. Then the posterior mean of $(\mathbf{B}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1})$ can be calculated below

$$
\begin{aligned}
\hat{\mathbf{B}}_1 = \mathbb{E}(\mathbf{B}_1 \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1}) &= \mathbb{E}\left((\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_1^T \mid \mathbf{Y}, \boldsymbol{\Theta}_{-\mathbf{B}_1}\right)\\
&= (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\mathbf{A}_F^T\mathbf{Y} = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{Y}
\end{aligned}
$$

   Note $\mathbf{B}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_1^T$, one has

$$
\mathbf{B}_1 - \hat{\mathbf{B}}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F(\tilde{\mathbf{B}}_{1,0})^T = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\left(\mathbf{A}_s\tilde{\mathbf{B}}_{1,0,s}^T + \mathbf{A}_c\tilde{\mathbf{B}}_{1,0,c}^T\right)
$$

where $\tilde{\mathbf{B}}_{1,0,s}$ is a $n_2 \times d$ matrix with the lth column independently sampled from $\mathcal{N}(\mathbf{0}, \tilde{\mathbf{\Sigma}}_l)$ for $l = 1, ..., d$. For the distribution of $\mathbf{A}_c \tilde{\mathbf{B}}_{1,0,c}^T$, using part 1 of Lemma 4, we have

$$\begin{aligned} p(\mathbf{A}_c \tilde{\mathbf{B}}_{1,0,c}^T \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1}) &\propto \exp\left\{ -\frac{1}{2\sigma_0^2} tr\left(\mathbf{A}_c^T \tilde{\mathbf{B}}_{1,0,c} \tilde{\mathbf{B}}_{1,0,c}^T \mathbf{A}_c \right) \right\} \\ &\propto \exp\left\{ -\frac{1}{2\sigma_0^2} tr\left((\mathbf{I} - \mathbf{A}_s \mathbf{A}_s^T) \tilde{\mathbf{B}}_{1,0,c} \tilde{\mathbf{B}}_{1,0,c}^T \right) \right\}. \end{aligned}$$

Thus we can sample $\mathbf{A}_c \tilde{\mathbf{B}}_{1,0,c}^T$ by $\sigma_0 (\mathbf{I} - \mathbf{A}_s \mathbf{A}_s^T) \mathbf{Z}_{0,1}$, where $\mathbf{Z}_{0,1}$ is an $n_1 \times n_2$ matrix with each entry independently sampled from standard normal distribution. The results soon follow.

2. (Column regression coefficients).

We first compute the posterior mean of $(\mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_2})$ below

$$\begin{aligned} \hat{\mathbf{B}}_2 &= \mathbb{E}(\mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_2}) = \mathbb{E}(\tilde{\mathbf{B}}_2 \mathbf{A}_F^T \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B_2}}) \\ &= \sum_{l=1}^{d} (\mathbf{H}_2^T \tilde{\mathbf{\Sigma}}_l^{-1} \mathbf{H}_2)^{-1} \mathbf{H}_2^T \tilde{\mathbf{\Sigma}}_l^{-1} \mathbf{Y}^T \mathbf{a}_l \mathbf{a}_l^T + (\mathbf{H}_2^T \mathbf{H}_2)^{-1} \mathbf{H}_2^T \mathbf{Y}^T (\mathbf{I}_{n_1} - \mathbf{A}_s \mathbf{A}_s^T) \end{aligned}$$

We denote the centered $\tilde{\mathbf{B}}_2$ by $\tilde{\mathbf{B}}_{2,0} = [\tilde{\mathbf{B}}_{2,0,s}, \tilde{\mathbf{B}}_{2,0,c}]$. We have

$$\mathbf{B}_2 - \hat{\mathbf{B}}_2 = \tilde{\mathbf{B}}_{2,0} \mathbf{A}_F^T = \tilde{\mathbf{B}}_{2,0,s} \mathbf{A}_s^T + \tilde{\mathbf{B}}_{2,0,c} \mathbf{A}_c^T$$

where $\tilde{\mathbf{B}}_{2,0,s}$ is a $q_2 \times d$ matrix with the lth column independently sampled from

$\mathcal{N}(\mathbf{0}, (\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1})$ for $l = 1, ..., d$. For the distribution of $\tilde{\mathbf{B}}_{2,0,c}\mathbf{A}_c^T$, we have

$$\begin{aligned} p(\tilde{\mathbf{B}}_{2,0,c}\mathbf{A}_c^T \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_2}) &\propto exp\left\{-\frac{1}{2\sigma_0^2} tr\left(\mathbf{A}_c\mathbf{A}_c^T\tilde{\mathbf{B}}_{2,0,c}\mathbf{H}_2^T\mathbf{H}_2\tilde{\mathbf{B}}_{2,0,c}^T\right)\right\} \\ &\propto exp\left\{-\frac{1}{2\sigma_0^2} tr\left((\mathbf{I} - \mathbf{A}_s\mathbf{A}_s^T)\tilde{\mathbf{B}}_{2,0,c}\mathbf{H}_2^T\mathbf{H}_2\tilde{\mathbf{B}}_{2,0,c}^T\right)\right\}. \end{aligned}$$

Thus we can sample $\tilde{\mathbf{B}}_{2,0,c}\mathbf{A}_c^T$ by $\sigma_0(\mathbf{I}_{n_1} - \mathbf{A}_s\mathbf{A}_s^T)\mathbf{Z}_{0,2}^T\mathbf{L}_{H_2}^T$, where $\mathbf{L}_{H_2}$ is a $q_2 \times q_2$ matrix such that $\mathbf{L}_{H_2}\mathbf{L}_{H_2}^T = (\mathbf{H}_2^T\mathbf{H}_2)^{-1}$ and $\mathbf{Z}_{0,2}$ is a $q_2 \times n_1$ matrix with each entry independently sampled from standard normal distribution.

□

Lemma 5 *Assume* $\mathbf{M} = \mathbf{H}_1\mathbf{B}_1 + (\mathbf{H}_2\mathbf{B}_2)^T$ *and let the objective prior* $\pi(\mathbf{B}_1, \mathbf{B}_2) \propto 1$ *for the regression parameters* $\mathbf{B}_1$ *and* $\mathbf{B}_2$*. Denote* $\tilde{\mathbf{B}}_1 = [\tilde{\mathbf{b}}_{1,1}, ..., \tilde{\mathbf{b}}_{1,n_1}] = \mathbf{B}_1^T\mathbf{H}_1^T\mathbf{A}_F$ *and* $\tilde{\mathbf{B}}_2 = [\tilde{\mathbf{b}}_{2,1}, ..., \tilde{\mathbf{b}}_{2,n_1}] = \mathbf{B}_2\mathbf{A}_F$*.*

1. *After marginalizing out* $\mathbf{Z}$ *and* $\mathbf{B}_1$*, assume the marginal posterior distribution of* $\tilde{\mathbf{B}}_1$ *follows*
$$p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}) = \prod_{l=1}^{n_1} \mathcal{PN}(\tilde{\boldsymbol{b}}_{1,l}; \tilde{\mathbf{y}}_l, \mathbf{Q}_{1,l}). \tag{C.5}$$
where $\mathbf{Q}_{1,l} = \mathbf{P}_l^T(\tilde{\mathbf{\Sigma}}_l)^{-1}\mathbf{P}_l$ *where* $\mathbf{P}_l = \mathbf{I} - \mathbf{H}_2(\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}$ *for* $l = 1, ..., d$ *and* $\mathbf{Q}_{1,l} = \sigma_0^2\mathbf{P}_0$ *with* $\mathbf{P}_0 = (\mathbf{I} - \mathbf{H}_2(\mathbf{H}_2^T\mathbf{H}_2)^{-1}\mathbf{H}_2^T)$ *for* $l = d+1, ..., n_1$*. The sample* $(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2})$ *can be obtained by* $(\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_1^T$*, where* $\tilde{\mathbf{B}}_1$ *sampled from the* $p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2})$ *in Equation (C.5).*

2. *After marginalizing out* $\mathbf{Z}$ *and conditional on* $\mathbf{B}_1$*, the marginal posterior distribution of* $\tilde{\mathbf{B}}_2$ *follows Equation in (C.4) by replacing* $\tilde{\mathbf{y}}_l$ *by* $\tilde{\mathbf{y}}_{l,B_1} = (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)^T\mathbf{a}_l$ *for* $l = 1, ..., d$*.*

Proof of Lemma 5: Denote $\mathbf{Y}_0 = \mathbf{Y} - \mathbf{H}_1\mathbf{B}_1 - \mathbf{B}_2^T\mathbf{H}_2^T$. Define $\mathbf{G} = [\mathbf{g}_1, \mathbf{g}_2, ..., \mathbf{g}_{n_1}] = (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)^T\mathbf{A}_F$. That is, $\mathbf{g}_l = (\mathbf{Y} - \mathbf{H}_1\mathbf{B}_1)^T\mathbf{a}_l$.

First we have the joint posterior distribution $(\mathbf{B}_1, \mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2})$

$$
\begin{aligned}
& p(\mathbf{B}_1, \mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}) \\
\propto & \exp\left\{-\frac{1}{2}\sum_{l=1}^{d}\mathbf{a}_l^T\mathbf{Y}_0^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{Y}_0\mathbf{a}_l - \frac{1}{2\sigma_0^2}\sum_{l=d+1}^{n_1}\mathbf{a}_l^T\mathbf{Y}_0^T\mathbf{Y}_0\mathbf{a}_l\right\} \\
\propto & \exp\left\{-\frac{1}{2}\sum_{l=1}^{d}(\mathbf{g}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})^T\tilde{\mathbf{\Sigma}}_l^{-1}(\mathbf{g}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l}) - \frac{1}{2\sigma_0^2}\sum_{l=d+1}^{n_1}(\mathbf{g}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})^T(\mathbf{g}_l - \mathbf{H}_2\tilde{\mathbf{b}}_{2,l})\right\},
\end{aligned}
$$

where $\tilde{\mathbf{b}}_{2,l}$ is a transformation of $\mathbf{B}_2$ defined in part 2 in Lemma 4.

After integrating out $\tilde{\mathbf{b}}_{2,l}$ from $p(\mathbf{B}_1, \mathbf{B}_2 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2})$ for $l = 1, 2..., n_1$, one has

$$
\begin{aligned}
& p(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}) \\
\propto & \exp\left\{-\frac{\sum_{l=1}^{d}(\mathbf{g}_l - \mathbf{H}_2\hat{\mathbf{b}}_{2,l})^T\tilde{\mathbf{\Sigma}}_l^{-1}(\mathbf{g}_l - \mathbf{H}_2\hat{\mathbf{b}}_{2,l})}{2} - \frac{\sum_{l=d+1}^{n_1}(\mathbf{g}_l - \mathbf{H}_2\hat{\mathbf{b}}_{2,l})^T(\mathbf{g}_l - \mathbf{H}_2\hat{\mathbf{b}}_{2,l})}{2\sigma_0^2}\right\} \\
\propto & \exp\left\{-\frac{\sum_{l=1}^{d}\mathbf{g}_l^T\mathbf{P}_l^T(\tilde{\mathbf{\Sigma}}_l)^{-1}\mathbf{P}_l\mathbf{g}_l}{2} - \frac{\sum_{l=d+1}^{n_1}\mathbf{g}_l^T\mathbf{P}_0\mathbf{g}_l}{2\sigma_0^2}\right\} \\
\propto & \exp\left\{-\frac{\sum_{l=1}^{n_1}\mathbf{g}_l^T\mathbf{Q}_{1,l}\mathbf{g}_l}{2}\right\}
\end{aligned}
$$

Where

$$
\hat{\mathbf{b}}_{2,l} = \begin{cases} (\mathbf{H}_2^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{H}_2)^{-1}\mathbf{H}_c^T\tilde{\mathbf{\Sigma}}_l^{-1}\mathbf{g}_l & l = 1, 2, ..., d \\ (\mathbf{H}_2^T\mathbf{H}_2)^{-1}\mathbf{H}_2^T\mathbf{g}_l & l = d+1, ..., n_1 \end{cases}
$$

Denote $\mathbf{B}_1^{aug} = [\mathbf{B}_1^T, \tilde{\mathbf{B}}_{1,(q_1+1):n_1}]^T$, where $\tilde{\mathbf{B}}_{1,(q_1+1):n_1}$ is the last $n_1 - q_1$ columns of $\tilde{\mathbf{B}}_1$. Denote the marginal posterior distribution $p_{trans}(\mathbf{B}_1^T \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}$ and $p_{trans}(\mathbf{B}_1^{aug} \mid$

$\mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}$ derived by the transformation of $p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2})$. One has

$$\begin{aligned} & p_{trans}(\mathbf{B}_1^T \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2}) \\ \propto & p(\mathbf{B}_1^{aug} \mid \mathbf{Y}, \Theta_{-\mathbf{B}_1,-\mathbf{B}_2}) \\ = & p(\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \Theta_{-\mathbf{B}_1,-\mathbf{B}_2}) \left| \frac{d\tilde{\mathbf{B}}_1}{d\mathbf{B}_1^{aug}} \right| \\ \propto & \exp\left\{ -\frac{\sum_{l=1}^{n_1} \mathbf{g}_l^T \mathbf{Q}_{1,l} \mathbf{g}_l}{2} \right\} \end{aligned}$$

Because $\mathbf{Q}_{1,l}$ is idempotent, i.e. $\mathbf{Q}_{1,l}\mathbf{Q}_{1,l} = \mathbf{Q}_{1,l}$, the Moore–Penrose inverse of $\mathbf{Q}_{1,l}$ is $\mathbf{Q}_{1,l}$ itself. Therefore for $l = 1, ..., d$, $\tilde{\mathbf{b}}_{1,l} \mid \mathbf{Y}, \mathbf{\Theta}_{-\mathbf{B}_1,-\mathbf{B}_2} \sim \mathcal{M}(\tilde{\mathbf{y}}_l,\, \mathbf{Q}_{1,l})$, from which the part 1 follows. Part 2 follows Lemma 4.

□

We are ready to prove Theorem 3.

Proof of Theorem 3: By Lemma 5, the posterior mean of $\tilde{\mathbf{B}}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2}$ is $\mathbf{Y}^T\mathbf{A}_F$, where $\mathbf{A}_F := [\mathbf{A}_s, \mathbf{A}_c]$. We denote the centered $\tilde{\mathbf{B}}_1$ by $\tilde{\mathbf{B}}_{1,0} = [\tilde{\mathbf{B}}_{1,Q}, \tilde{\mathbf{B}}_{1,0,c}] = \tilde{\mathbf{B}}_1 - \mathbf{Y}^T\mathbf{A}_F$, where $\tilde{\mathbf{B}}_{1,Q}$ is the first d columns of $\tilde{\mathbf{B}}_{1,0}$ and $\tilde{\mathbf{B}}_{1,0,c}$ is the next $(n_1 - d)$ columns of $\tilde{\mathbf{B}}_{1,0}$. Then the posterior mean of $\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2}$ can be calculated below

$$\begin{aligned} \hat{\mathbf{B}}_1 &= \mathbb{E}\left(\mathbf{B}_1 \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2}\right) = \mathbb{E}\left((\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_1^T \mid \mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2} \right) \\ &= (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\mathbf{A}_F^T\mathbf{Y} = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{Y} \end{aligned}$$

Note $\mathbf{B}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F\tilde{\mathbf{B}}_1^T$, one has

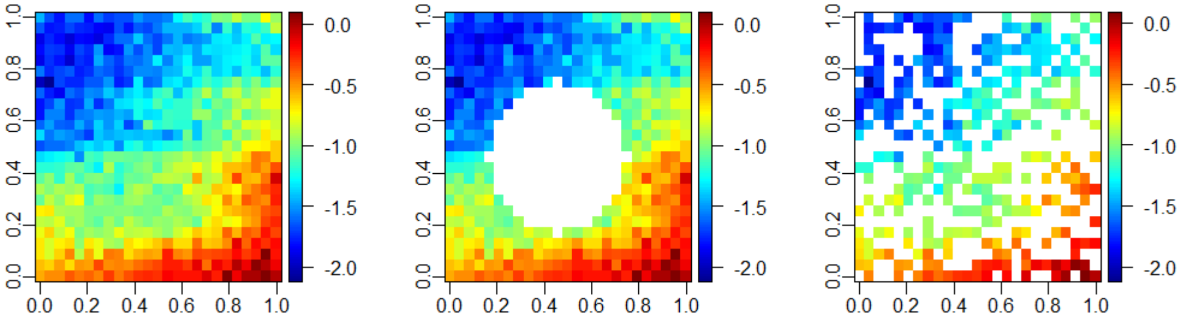


Figure C.1: The simulated data with full observations, disk missing pattern and missing-at-random pattern with 50% of the missing values are graphed in the left, middle and right panels, respectively.

$$\mathbf{B}_1 - \hat{\mathbf{B}}_1 = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T\mathbf{A}_F(\tilde{\mathbf{B}}_{1,0})^T = (\mathbf{H}_1^T\mathbf{H}_1)^{-1}\mathbf{H}_1^T(\mathbf{A}_s(\tilde{\mathbf{B}}_{1,Q})^T + \mathbf{A}_c(\tilde{\mathbf{B}}_{1,0,c})^T)$$

where by Lemma 5, $\tilde{\mathbf{B}}_{1,Q}$ is an $n_2 \times d$ matrix with the lth column independently sampled from $\mathcal{N}(\mathbf{0}, \mathbf{Q}_{1,l})$ for $l = 1, ..., d$. For the distribution of $\mathbf{A}_c\tilde{\mathbf{B}}_{1,0,c}^T$, using part 1 of Lemma 5, we have

$$\begin{aligned} & p(\mathbf{A}_c\tilde{\mathbf{B}}_{1,0,c}^T|\mathbf{Y}, \mathbf{\Theta}_{-B_1,-B_2}) \\ \propto & \exp\left\{\frac{1}{2\sigma_0^2}tr\left(\mathbf{A}_c\mathbf{A}_c^T\tilde{\mathbf{B}}_{1,0,c}\mathbf{P}_0(\tilde{\mathbf{B}}_{1,0,c})^T\right)\right\} \\ \propto & \exp\left\{-\frac{1}{2\sigma_0^2}tr\left((\mathbf{I} - \mathbf{A}_s\mathbf{A}_s^T)\tilde{\mathbf{B}}_{1,0,c}\mathbf{P}_0(\tilde{\mathbf{B}}_{1,0,c})^T\right)\right\}. \end{aligned}$$

Thus we can sample marginal posterior distribution of $\mathbf{A}_c\tilde{\mathbf{B}}_{1,0,c}^T$ by $\sigma_0(\mathbf{I} - \mathbf{A}_s\mathbf{A}_s^T)\mathbf{Z}_{0,1}\mathbf{P}_0$, where $\mathbf{Z}_{0,1}$ is an $n_1 \times n_2$ matrix with each entry independently sampled from standard normal distribution. The results soon follow.

□

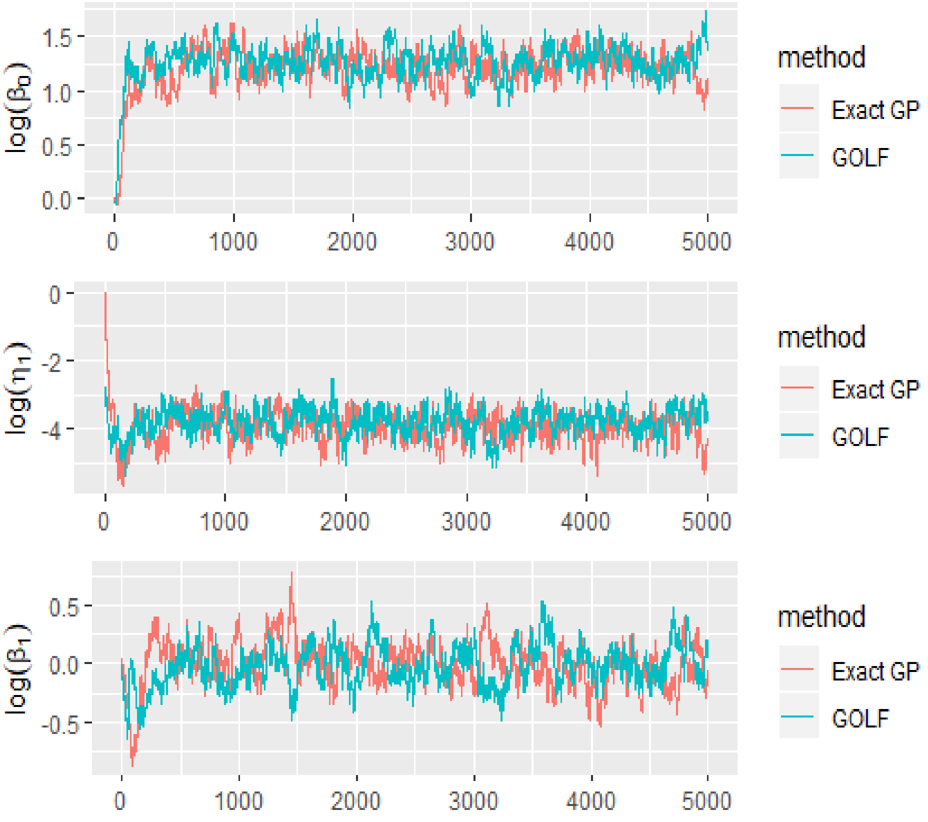


Figure C.2: Trace plots of the posterior samples of the parameters in the exact GP model and the GOLF processes for the simulated data in figure C.1.

C.2 Additional Results of Simulated Studies in Section 4.5

We provide additional results for the simulated studies in Example 3 in Figure C.1 and Figure C.2. We graph the simulated data set with full observations, disk missing pattern and missing-at-random pattern with 50% of the missing values in Figure C.1. The posterior samples of the logarithm of the inverse range parameter of factor loading matrix, the nugget parameter and the inverse range parameter of the factors are graphed from the upper to lower panels in Figure C.2, respectively. The posterior samples of parameters in the exact GP model and GOLF processes are similar to each other.

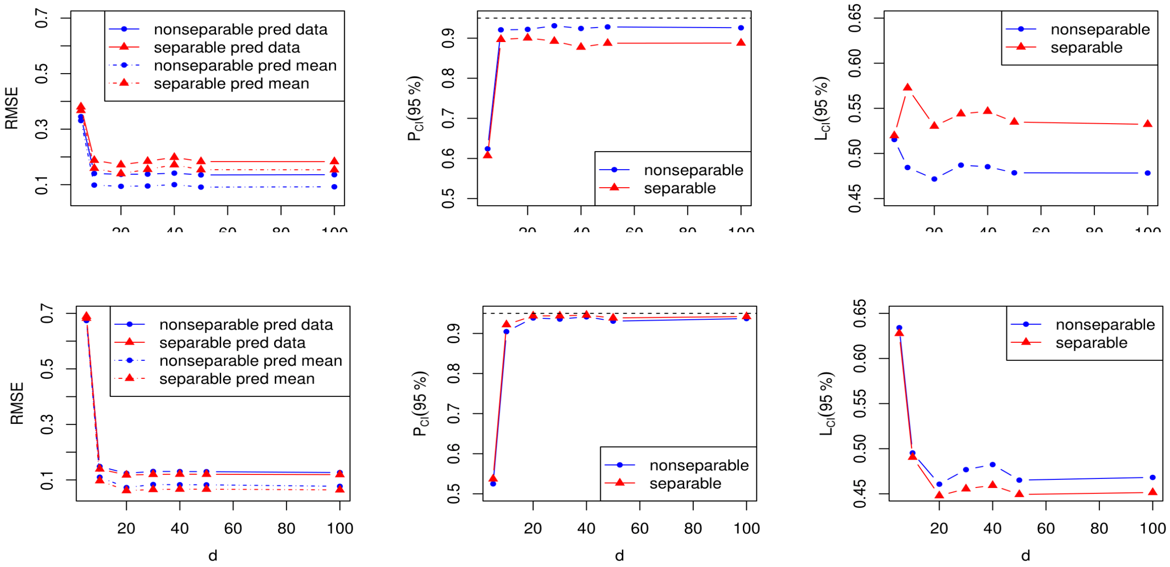


Figure C.3: The predictive performance of GOLF process with $d = 5, 10, 20, 30, 40, 50$ and 100 factors for Example 4, when the true number of factor is $d_{real} = 100$ in generating the data. The nonseparable kernel with distinct kernel parameters is assumed to generate the data in the first row of panels, and separable kernel with the same kernel parameter of each factor process is used for simulation in the second row of panels. The blue curves and red curves denote the performance by the GOLF processes with the different kernel parameters and the same kernel parameter, respectively. In the left panels, the solid curves denote the RMSE for predicting the (noisy) observations, and the dashed curve denote the RMSE for predicting the mean of the observations. The proportions of observations covered in the 95% predictive interval and the average length of the predictive interval are graphed in the middle and right panels, respectively.

C.3 Additional Results for Real Applications in Section 4.6.1

In this section, we include additional results for GOLF processes predicting the missing values of the temperature data set discussed in [125]. We show the details of 5 different configurations of GOLF processes, where the result reported in the main body of the article is the configuration 1. For all the configurations, the proportion of the burn-in samples is 20%. We use the normal distribution centered on the previous values as the proposal distribution of the logarithm of the inverse range parameters and logarithm of the nugget parameters. For the logarithm of the inverse range parameters of the factor loading matrix, the standard deviation of the proposal distribution is $40/n_1$. For the logarithm of the inverse range parameters and the nugget parameters of the factor processes, the standard deviation of the posterior distribution is set to be $40/n_2$.

	sample size	system	initial $Y^*_{v,i}$	initial $log(\beta_0)$	initial $\log(\beta_l)$
Conf. 1	6000	Mac	mean at each latitude	3	0
Conf. 2	6000	Win	mean at each latitude	3	0
Conf. 3	40000	Mac	mean at each latitude	3	0
Conf. 4	40000	Mac	overall mean + noise	3	0
Conf. 5	40000	Mac	mean at each latitude	Unif[-1,1]	Unif[-1,1]

Table C.1: Detailed settings of 5 different configurations of GOLF processes for the data set in [125]. The number of samples and the computing system are shown in the second column and third column, respectively. The choice of the initial values of the missing data is given in the fourth column, using either the mean of the observations at each latitude or overall mean of the observations with a small random Gaussian noise (with standard deviation being 0.1 times of the standard deviation of the observations). The initial values of the logarithm of the inverse range parameters are either chosen to be a fixed value or randomly sampled from the uniform distribution, shown in column 5-6.

The details of 5 configurations are given in Table C.1. The predictive RMSE, $P_{CI}(95\%)$ and $L_{CI}(95\%)$ of the 5 configurations are given in Table C.2. The predictive RMSE is similar for all 5 configurations. Increasing the posterior sample size seems to slightly

Methods	RMSE	$P_{CI}(95\%)$	$L_{CI}(95\%)$
Configuration 1	1.46	0.92	4.95
Configuration 2	1.50	0.91	4.92
Configuration 3	1.44	0.94	7.70
Configuration 4	1.48	0.94	7.75
Configuration 5	1.51	0.93	5.16

Table C.2: Predictive performance of 5 different implementations for the data set in [125].

increase the proportion of the samples contained in the 95% predictive interval.

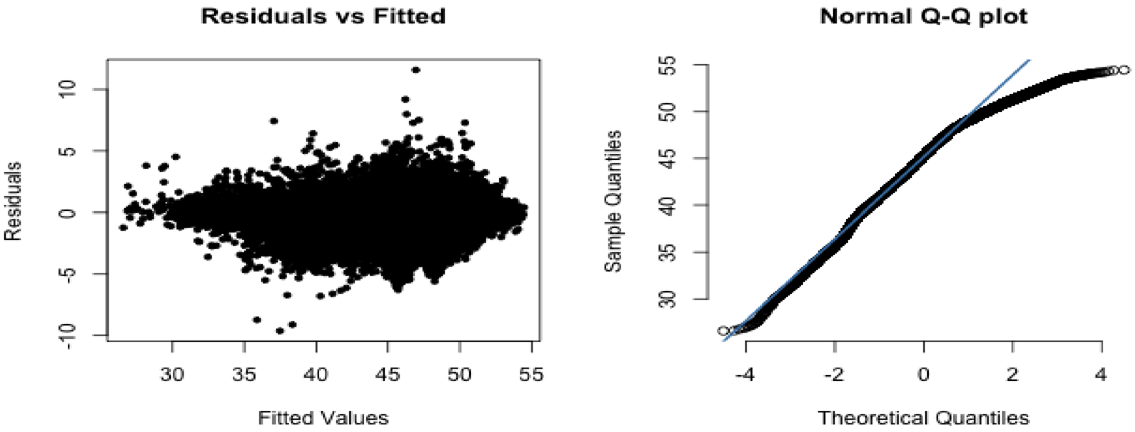


Figure C.4: Diagnostic plots of the GOLF processes for the data set in [125].

The fitted values from the GOLF processes in configuration 1 against the residuals and the normal Q-Q plot are graphed in left panel and the right panel in Figure C.4, respectively. The Q-Q plot indicates the fitted values are slightly left-skewed and slightly under-dispersed.

Bibliography

[1] A. E. Gelfand, P. Diggle, P. Guttorp, and M. Fuentes, *Handbook of spatial statistics.* CRC Press, 2010.

[2] S. Banerjee, B. P. Carlin, and A. E. Gelfand, *Hierarchical modeling and analysis for spatial data.* CRC Press, 2014.

[3] N. Cressie and C. K. Wikle, *Statistics for spatio-temporal data.* John Wiley & Sons, 2015.

[4] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, *et. al.*, *Design and analysis of computer experiments*, *Statistical science* **4** (1989), no. 4 409–423.

[5] T. J. Santner, B. J. Williams, and W. I. Notz, *The design and analysis of computer experiments.* Springer Science & Business Media, 2003.

[6] M. J. Bayarri, J. O. Berger, R. Paulo, J. Sacks, J. A. Cafeo, J. Cavendish, C.-H. Lin, and J. Tu, *A framework for validation of computer models*, *Technometrics* **49** (2007), no. 2 138–154.

[7] M. C. Kennedy and A. O'Hagan, *Bayesian calibration of computer models*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63** (2001), no. 3 425–464.

[8] D. Higdon, J. Gattiker, B. Williams, and M. Rightley, *Computer model calibration using high-dimensional output*, *Journal of the American Statistical Association* **103** (2008), no. 482 570–583.

[9] M. Gu and L. Wang, *Scaled Gaussian stochastic process for computer model calibration and prediction*, *SIAM/ASA Journal on Uncertainty Quantification* **6** (2018), no. 4 1555–1583.

[10] M. Gu, F. Xie, and L. Wang, *A theoretical framework of the scaled Gaussian stochastic process in prediction and calibration*, *SIAM/ASA Journal on Uncertainty Quantification* **10** (2022), no. 4 1435–1460.

[11] J. Hartikainen and S. Särkkä, *Kalman filtering and smoothing solutions to temporal Gaussian process regression models*, in *2010 IEEE international workshop on machine learning for signal processing*, pp. 379–384, IEEE, 2010.

[12] M. West and P. J. Harrison, *Bayesian Forecasting & Dynamic Models.* Springer Verlag, 2nd ed., 1997.

[13] G. Petris, S. Petrone, and P. Campagnoli, *Dynamic linear models*, in *Dynamic linear models with R*, pp. 31–84. Springer, 2009.

[14] J. Durbin and S. J. Koopman, *Time series analysis by state space methods*, vol. 38. OUP Oxford, 2012.

[15] S. Särkkä and L. Svensson, *Bayesian filtering and smoothing*, vol. 17. Cambridge university press, 2023.

[16] R. E. Kalman, *A new approach to linear filtering and prediction problems*, *Journal of basic Engineering* **82** (1960), no. 1 35–45.

[17] H. E. Rauch, F. Tung, and C. T. Striebel, *Maximum likelihood estimates of linear dynamic systems*, *AIAA journal* **3** (1965), no. 8 1445–1450.

[18] M. J. Bayarri, J. O. Berger, E. S. Calder, K. Dalbey, S. Lunagomez, A. K. Patra, E. B. Pitman, E. T. Spiller, and R. L. Wolpert, *Using statistical and computer models to quantify volcanic hazards*, *Technometrics* **51** (2009), no. 4 402–413, [https://www.jstor.org/stable/40586650].

[19] R. B. Gramacy, *Surrogates: Gaussian process modeling, design, and optimization for the applied sciences.* Chapman and Hall/CRC, 2020.

[20] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. Von Lilienfeld, *Fast and accurate modeling of molecular atomization energies with machine learning*, *Physical review letters* **108** (2012), no. 5 058301.

[21] S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, and K.-R. Müller, *Machine learning of accurate energy-conserving molecular force fields*, *Science advances* **3** (2017), no. 5 e1603015.

[22] H. Li, M. Zhou, J. Sebastian, J. Wu, and M. Gu, *Efficient force field and energy emulation through partition of permutationally equivalent atoms*, *The Journal of Chemical Physics* **156** (2022), no. 18.

[23] M. Bogojeski, L. Vogt-Maranto, M. E. Tuckerman, K.-R. Müller, and K. Burke, *Quantum chemical accuracy from density functional approximations via machine learning*, *Nature communications* **11** (2020), no. 1 5223.

[24] X. Fang, M. Gu, and J. Wu, *Reliable emulation of complex functionals by active learning with error control*, *The Journal of Chemical Physics* **157** (12, 2022) 214109.

[25] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Neural operator: Learning maps between function spaces with applications to pdes*, *Journal of Machine Learning Research* **24** (2023), no. 89 1–97.

[26] P. W. Sauer and M. A. Pai, *Power system dynamics and stability.* Wiley, Urbana, IL, USA, 1998.

[27] A. B. Birchfield *et. al.*, *Grid Structural Characteristics as Validation Criteria for Synthetic Networks*, *IEEE Transactions on Power Systems* **32** (2017), no. 4 3258–3265.

[28] K. Ye, J. Zhao, H. Li, and M. Gu, *A High Computationally Efficient Parallel Partial Gaussian Process for Large-Scale Power System Probabilistic Transient Stability Assessment*, *IEEE Transactions on Power Systems* (2023).

[29] J. Navarrete, *SARS-CoV-2 infection and death rates among maintenance dialysis patients during Delta and early omicron waves—United States, June 30, 2021–September 27, 2022*, *MMWR. Morbidity and Mortality Weekly Report* **72** (2023).

[30] C. K. Monaghan, J. W. Larkin, S. Chaudhuri, H. Han, Y. Jiao, K. M. Bermudez, E. D. Weinhandl, I. A. Dahne-Steuber, K. Belmonte, L. Neri, P. Kotanko, J. P. Kooman, J. L. Hymes, R. J. Kossmann, L. A. Usvyat, and F. W. Maddux, *Machine learning for prediction of patients on hemodialysis with an undetected SARS-CoV-2 infection*, *Kidney360* **2** (2021), no. 3 456.

[31] Y. Wang, *Early detection, containment, and management of COVID-19 in dialysis facilities using multi-modal data sources*, *NIH COVID Rapid Acceleration of Diagnostics (RADx) Data Hub* (2021) [https://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/study.cgi?study_id=phs002657.v1.p1].

[32] W. T. Li, J. Ma, N. Shende, G. Castaneda, J. Chakladar, J. C. Tsai, L. Apostol, C. O. Honda, J. Xu, L. M. Wong, T. Zhang, A. Lee, A. Gnanasekar, T. K. Honda, S. Z. Kuo, M. A. Yu, E. Y. Chang, M. R. Rajasekaran, and W. M. Ongkeko, *Using machine learning of clinical data to diagnose COVID-19: a systematic review and meta-analysis*, *BMC medical informatics and decision making* **20** (2020), no. 1 1–13.

[33] Y. Zoabi, S. Deri-Rozov, and N. Shomron, *Machine learning-based prediction of COVID-19 diagnosis based on symptoms*, *npj digital medicine* **4** (2021), no. 1 3.

[34] T. K. Ho, *Random decision forests*, in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[35] L. Breiman, *Random forests*, *Machine learning* **45** (2001) 5–32.

[36] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[37] P. Fearnhead and Z. Liu, *Efficient Bayesian analysis of multiple changepoint models with dependence across segments*, *Statistics and Computing* **21** (2011), no. 2 217–229.

[38] G. Romano, G. Rigaill, V. Runge, and P. Fearnhead, *Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise*, *Journal of the American Statistical Association* **117** (2022), no. 540 2147–2162.

[39] Y. Saatçi, R. D. Turner, and C. E. Rasmussen, *Gaussian process change point models*, in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 927–934, 2010.

[40] Q. Lin, S. Zhao, D. Gao, Y. Lou, S. Yang, S. S. Musa, M. H. Wang, Y. Cai, W. Wang, L. Yang, and H. Dai, *A conceptual model for the outbreak of coronavirus disease 2019 (COVID-19) in Wuhan, China with individual reaction and governmental action*, *International journal of infectious diseases* **93** (2020) 211–216, [https://doi.org/10.1016/j.ijid.2020.02.058].

[41] G. Giordano, F. Blanchini, R. Bruno, P. Colaneri, A. Di Filippo, A. Di Matteo, and M. Colaneri, *Modelling the COVID-19 epidemic and implementation of population-wide interventions in Italy*, *Nature Medicine* **26** (2020) 855–860, [https://doi.org/10.1038/s41591-020-0883-7].

[42] J. Dehning, J. Zierenberg, F. P. Spitzner, M. Wibral, J. P. Neto, M. Wilczek, and V. Priesemann, *Inferring change points in the spread of COVID-19 reveals the effectiveness of interventions*, *Science* **369** (2020), no. 6500 [https://science.sciencemag.org/content/369/6500/eabb9789].

[43] J. Fernández-Villaverde and C. I. Jones, *Estimating and Simulating a SIRD model of COVID-19 for Many Countries, States, and Cities*, Working Paper 27128, National Bureau of Economic Research, May, 2020.

[44] D. A. Swan, A. Goyal, C. Bracis, M. Moore, E. Krantz, E. R. Brown, F. Cardozo-Ojeda, D. B. Reeves, F. Gao, P. B. Gilbert, *et. al.*, *Vaccines that prevent SARS-CoV-2 transmission may prevent or dampen a spring wave of COVID-19 cases and deaths in 2021*, *medRxiv* (2020) [https://doi.org/10.1101/2020.12.13.20248120].

[45] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton, *et. al.*, *Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe*, *Nature* **584** (2020), no. 7820 257–261, [https://doi.org/10.1038/s41586-020-2405-7].

[46] N. Hoertel, M. Blachier, C. Blanco, M. Olfson, M. Massetti, M. S. Rico, F. Limosin, and H. Leleu, *A stochastic agent-based model of the SARS-CoV-2 epidemic in France*, *Nature Medicine* **26** (2020) 1417–1421, [https://doi.org/10.1038/s41591-020-1001-6].

[47] A. L. Bertozzi, E. Franco, G. Mohler, M. B. Short, and D. Sledge, *The challenges of modeling and forecasting the spread of COVID-19*, *Proceedings of the National Academy of Sciences* **117** (2020) 16732–16738, [https://www.pnas.org/content/117/29/16732].

[48] J. A. Firth, J. Hellewell, P. Klepac, S. Kissler, A. J. Kucharski, and L. G. Spurgin, *Using a real-world network to model localized COVID-19 control strategies*, *Nature medicine* **26** (2020) 1616–1622, [https://doi.org/10.1038/s41591-020-1036-8].

[49] A. V. Vecchia, *Estimation and model identification for continuous spatial processes*, *Journal of the Royal Statistical Society: Series B (Methodological)* **50** (1988), no. 2 297–312.

[50] M. Katzfuss and J. Guinness, *A General Framework for Vecchia Approximations of Gaussian Processes*, *Statistical Science* **36** (2021), no. 1 124 – 141.

[51] E. Snelson and Z. Ghahramani, *Sparse Gaussian processes using pseudo-inputs*, *Advances in Neural Information Processing Systems* **18** (2006) 1259–1266.

[52] F. Lindgren, H. Rue, and J. Lindström, *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** (2011), no. 4 423–498.

[53] H. Rue, S. Martino, and N. Chopin, *Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71** (2009), no. 2 319–392.

[54] A. Datta, S. Banerjee, A. O. Finley, and A. E. Gelfand, *Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets*, *Journal of the American Statistical Association* **111** (2016), no. 514 800–812.

[55] M. Katzfuss, *A multi-resolution approximation for massive spatial datasets*, *Journal of the American Statistical Association* **112** (2017), no. 517 201–214.

[56] R. B. Gramacy and D. W. Apley, *Local Gaussian process approximation for large computer experiments*, *Journal of Computational and Graphical Statistics* **24** (2015), no. 2 561–578.

[57] J. Guinness and M. Fuentes, *Circulant embedding of approximate covariances for inference from gaussian data on large lattices*, *Journal of computational and Graphical Statistics* **26** (2017), no. 1 88–97.

[58] J. R. Stroud, M. L. Stein, and S. Lysen, *Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice*, *Journal of computational and Graphical Statistics* **26** (2017), no. 1 108–120.

[59] C. G. Kaufman, M. J. Schervish, and D. W. Nychka, *Covariance tapering for likelihood-based estimation in large spatial data sets*, *Journal of the American Statistical Association* **103** (2008), no. 484 1545–1555.

[60] K. R. Anderson, I. A. Johanson, M. R. Patrick, M. Gu, P. Segall, M. P. Poland, E. K. Montgomery-Brown, and A. Miklius, *Magma reservoir failure and the onset of caldera collapse at Kīlauea Volcano in 2018*, *Science* **366** (2019), no. 6470 eaaz1822.

[61] M. Gu, K. Anderson, and E. McPhillips, *Calibration of imperfect geophysical models by multiple satellite interferograms with measurement bias*, *Technometrics* (2023) 1–12.

[62] M. Gu and J. O. Berger, *Parallel partial Gaussian process emulation for computer models with massive output*, *The Annals of Applied Statistics* **10** (2016), no. 3 1317–1347.

[63] C. E. Rasmussen, *Gaussian processes for machine learning.* MIT Press, 2006.

[64] M. L. Stein, *Interpolation of spatial data: some theory for kriging.* Springer Science & Business Media, 1999.

[65] J. O. Berger, V. De Oliveira, and B. Sansó, *Objective Bayesian analysis of spatially correlated data*, *Journal of the American Statistical Association* **96** (2001), no. 456 1361–1374.

[66] R. Paulo, *Default priors for Gaussian processes*, *Annals of statistics* **33** (2005), no. 2 556–582.

[67] J. Nocedal, *Updating quasi-newton matrices with limited storage*, *Mathematics of computation* **35** (1980), no. 151 773–782.

[68] D. C. Liu and J. Nocedal, *On the limited memory bfgs method for large scale optimization*, *Mathematical programming* **45** (1989), no. 1-3 503–528.

[69] M. Gu, X. Wang, and J. O. Berger, *Robust Gaussian stochastic process emulation*, *The Annals of statistics* **46** (2018) 3038–3066, [https://doi.org/10.1214/17-AOS1648].

[70] M. Gu, *Jointly robust prior for Gaussian stochastic process in emulation, calibration and variable selection*, *Bayesian Analysis* **14** (2019), no. 3 857 – 885.

[71] M. Gu, Y. Lin, V. C. Lee, and D. Y. Qiu, *Probabilistic forecast of nonlinear dynamical systems with uncertainty quantification*, *Physica D: Nonlinear Phenomena* **457** (2024) 133938.

[72] P. Whittle, *On stationary processes in the plane*, *Biometrika* **41** (1954), no. 3/4 434–449.

[73] M. Gu and Y. Xu, *Fast nonseparable gaussian stochastic process with application to methylation level interpolation*, *Journal of Computational and Graphical Statistics* **29** (2020), no. 2 250–260.

[74] R. E. Kalman and R. S. Bucy, *New results in linear filtering and prediction theory*, *Journal of basic engineering* **83** (1961), no. 1 95–108.

[75] M. Gu, X. Liu, X. Fang, and S. Tang, *Scalable Marginalization of Correlated Latent Variables with Applications to Learning Particle Interaction Kernels*, *The New England Journal of Statistics in Data Science* (2022) 1–15.

[76] J. S. Jia, X. Lu, Y. Yuan, G. Xu, J. Jia, and N. A. Christakis, *Population flow drives spatio-temporal distribution of COVID-19 in China*, *Nature* **582** (2020) 389–394, [https://doi.org/10.1038/s41586-020-2284-y].

[77] H. S. Badr, H. Du, M. Marshall, E. Dong, M. M. Squire, and L. M. Gardner, *Association between mobility patterns and COVID-19 transmission in the USA: a mathematical modelling study*, *The Lancet Infectious Diseases* **20** (2020) 1247–1254, [https://doi.org/10.1016/S1473-3099(20)30553-3].

[78] R. P. Adams and D. J. MacKay, *Bayesian online changepoint detection*, *arXiv preprint arXiv:0710.3742* (2007).

[79] P. Fearnhead and Z. Liu, *On-line inference for multiple changepoint problems*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **69** (2007), no. 4 589–605.

[80] G. J. van den Burg and C. K. Williams, *An evaluation of change point detection algorithms*, *arXiv preprint arXiv:2003.06222* (2020).

[81] M. Basseville and I. V. Nikiforov, *Detection of abrupt changes : theory and applications*, vol. 104. Prentice-Hall, Englewood Cliffs (New Jersey), 1993.

[82] N. R. Zhang, D. O. Siegmund, H. Ji, and J. Z. Li, *Detecting simultaneous changepoints in multiple sequences*, *Biometrika* **97** (2010), no. 3 631–645.

[83] P. Fryzlewicz, *Wild binary segmentation for multiple change-point detection*, *The Annals of Statistics* **42** (2014), no. 6 2243 – 2281.

[84] Y. C. Chen, T. Banerjee, A. D. Domínguez-García, and V. V. Veeravalli, *Quickest line outage detection and identification*, *IEEE Transactions on Power Systems* **31** (2016), no. 1 749–758.

[85] T. Harris, B. Li, and J. D. Tucker, *Scalable multiple changepoint detection for functional data sequences*, *Environmetrics* **33** (2022), no. 2 e2710.

[86] W. Zhang, M. Griffin, and D. S. Matteson, *Modeling a nonlinear biophysical trend followed by long-memory equilibrium with unknown change point*, *The Annals of Applied Statistics* **17** (2023), no. 1 860–880.

[87] J. Duan, H. Li, X. Ma, H. Zhang, R. Lasky, C. K. Monaghan, S. Chaudhuri, L. A. Usvyat, M. Gu, W. Guo, P. Kotanko, and Y. Wang, *Predicting SARS-CoV-2 infection among hemodialysis patients using multimodal data*, *Frontiers in Nephrology* **3** (2023) 1179342.

[88] Y. Ni, P. Müller, and Y. Ji, *Bayesian double feature allocation for phenotyping with electronic health records*, *Journal of the American Statistical Association* **115** (2020), no. 532 1620–1634.

[89] D. V. Hinkley, *Inference about the change-point in a sequence of random variables*, *Biometrika* **57** (1970), no. 1 1 – 17.

[90] P. Fearnhead, *Exact and efficient Bayesian inference for multiple changepoint problems*, *Statistics and computing* **16** (2006), no. 2 203–213.

[91] R. Killick, P. Fearnhead, and I. A. Eckley, *Optimal detection of changepoints with a linear computational cost*, *Journal of the American Statistical Association* **107** (2012), no. 500 1590–1598.

[92] D. S. Matteson and N. A. James, *A nonparametric approach for multiple change point analysis of multivariate data*, *Journal of the American Statistical Association* **109** (2014), no. 505 334–345.

[93] K. Haynes, P. Fearnhead, and I. A. Eckley, *A computationally efficient nonparametric approach for changepoint detection*, *Statistics and computing* **27** (2017) 1293–1305.

[94] L. Chu and H. Chen, *Asymptotic distribution-free change-point detection for multivariate and non-Euclidean data*, *The Annals of Statistics* **47** (2019), no. 1 382 – 414.

[95] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press, 2018.

[96] R. Prado and M. West, *Time series: modeling, computation, and inference*. Chapman and Hall/CRC, 2010.

[97] A. Doucet, S. Godsill, and C. Andrieu, *On sequential Monte Carlo sampling methods for Bayesian filtering*, *Statistics and computing* **10** (2000) 197–208.

[98] J. R. Stroud, P. Müller, and B. Sansó, *Dynamic models for spatiotemporal data*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63** (2001), no. 4 673–689.

[99] M. S. Handcock and M. L. Stein, *A Bayesian analysis of kriging*, *Technometrics* **35** (1993), no. 4 403–410.

[100] E. S. Page, *Continuous inspection schemes*, *Biometrika* **41** (1954), no. 1/2 100–115.

[101] M. Basseville, I. Nikiforov, and A. Tartakovsky, *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. CRC Press, 2014.

[102] H. Chen, *Sequential change-point detection based on nearest neighbors*, *The Annals of Statistics* **47** (2019), no. 3 1381–1407.

[103] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, *IEEE transactions on pattern analysis and machine intelligence* **33** (2010), no. 5 898–916.

[104] L. Jansen, B. Tegomoh, K. Lange, K. Showalter, J. Figliomeni, B. Abdalhamid, P. C. Iwen, J. Fauver, B. Buss, and M. Donahue, *Investigation of a SARS-CoV-2 B.1.1.529 (Omicron) Variant Cluster — Nebraska, November–December 2021*, *MMWR. Morbidity and Mortality Weekly Report* **70** (12, 2021) 1782–1784.

[105] J. S. Song, J. Lee, M. Kim, H. S. Jeong, M. S. Kim, S. G. Kim, H. N. Yoo, J. J. Lee, H. Y. Lee, S.-E. Lee, E. J. Kim, J. E. Rhee, I. H. Kim, and Y.-J. Park, *Serial Intervals and Household Transmission of SARS-CoV-2 Omicron Variant, South Korea, 2021*, *Emerging Infectious Diseases* **28** (2022) 756–759.

[106] S. Hakki, J. Zhou, J. Jonnerby, A. Singanayagam, J. L. Barnett, K. J. Madon, A. Koycheva, C. Kelly, H. Houston, S. Nevin, J. Fenn, R. Kundu, M. A. Crone, T. D. Pillay, S. Ahmad, N. Derqui-Fernandez, E. Conibear, P. S. Freemont, G. P. Taylor, and N. Ferguson, *Onset and window of SARS-CoV-2 infectiousness and temporal correlation with symptom onset: a prospective, longitudinal, community cohort study*, *The Lancet Respiratory Medicine* **10** (2022), no. 11 1061–1073.

[107] H. Li and M. Gu, *Robust estimation of SARS-CoV-2 epidemic in US counties*, *Scientific reports* **11** (2021), no. 1 11841.

[108] E. Dong, H. Du, and L. Gardner, *An interactive web-based dashboard to track COVID-19 in real time*, *The Lancet infectious diseases* **20** (2020) 533–534, [https://doi.org/10.1016/S1473-3099(20)30120-1].

[109] R. Li, S. Pei, B. Chen, Y. Song, T. Zhang, W. Yang, and J. Shaman, *Substantial undocumented infection facilitates the rapid dissemination of novel coronavirus (SARS-CoV-2)*, *Science* **368** (2020) 489–493, [https://science.sciencemag.org/content/368/6490/489].

[110] H. W. Hethcote, *The mathematics of infectious diseases*, *SIAM review* **42** (2000), no. 4 599–653, [https://doi.org/10.1137/S0036144500371907].

[111] A. Chande, S. Lee, M. Harris, Q. Nguyen, S. J. Beckett, T. Hilley, C. Andris, and J. S. Weitz, *Real-time, interactive website for US-county-level COVID-19 event risk assessment*, *Nature Human Behaviour* **4** (2020), no. 12 1313–1319, [https://doi.org/10.1038/s41562-020-01000-9].

[112] I. Holmdahl and C. Buckee, *Wrong but useful—what COVID-19 epidemiologic models can and cannot tell us*, *New England Journal of Medicine* **383** (2020), no. 4 303–305, [https://doi.org/10.1056/NEJMp2016822].

[113] S. Anand, M. Montez-Rath, J. Han, J. Bozeman, R. Kerschmann, P. Beyer, J. Parsonnet, and G. M. Chertow, *Prevalence of SARS-CoV-2 antibodies in a large nationwide sample of patients on dialysis in the USA: a cross-sectional study*, *The Lancet* **396** (2020) 1335–1344, [https://doi.org/10.1016/S0140-6736(20)32009-2].

[114] A. J. Kucharski, T. W. Russell, C. Diamond, Y. Liu, J. Edmunds, S. Funk, R. M. Eggo, F. Sun, M. Jit, J. D. Munday, *et. al.*, *Early dynamics of transmission and control of COVID-19: a mathematical modelling study*, *The lancet infectious diseases* **20** (2020) 553–558, [https://doi.org/10.1016/S1473-3099(20)30144-4].

[115] Covid Tracking Project Team, *Covid tracking project.*, https://covidtracking.com/.

[116] N. G. Davies, A. J. Kucharski, R. M. Eggo, A. Gimma, W. J. Edmunds, T. Jombart, K. O'Reilly, A. Endo, J. Hellewell, E. S. Nightingale, *et. al.*, *Effects of non-pharmaceutical interventions on COVID-19 cases, deaths, and demand for hospital services in the UK: a modelling study*, *The Lancet Public Health* **5** (2020) e375 – e385, [https://doi.org/10.1016/S2468-2667(20)30133-X].

[117] R. Verity, L. C. Okell, I. Dorigatti, P. Winskill, C. Whittaker, N. Imai, G. Cuomo-Dannenburg, H. Thompson, P. G. Walker, H. Fu, *et. al.*, *Estimates of the severity of coronavirus disease 2019: a model-based analysis*, *The Lancet infectious diseases* **20** (2020) 669–677, [https://doi.org/10.1016/S1473-3099(20)30243-7].

[118] H. Nishiura and G. Chowell, *The Effective Reproduction Number as a Prelude to Statistical Estimation of Time-Dependent Epidemic Trends*, in *Mathematical and Statistical Estimation Approaches in Epidemiology*, pp. 103–121. Springer Netherlands, 2009.

[119] M. Gu, J. Palomo, and J. O. Berger, *RobustGaSP: Robust Gaussian stochastic process emulation in R*, *R Journal* **11** (2019), no. 1.

[120] A. Goyal, D. B. Reeves, E. F. Cardozo-Ojeda, J. T. Schiffer, and B. T. Mayer, *Viral load and contact heterogeneity predict SARS-CoV-2 transmission and super-spreading events*, *Elife* **10** (2021) e63537, [https://doi.org/10.7554/eLife.63537].

[121] N. A. Cressie and N. A. Cassie, *Statistics for spatial data.* Wiley, New York, 1993.

[122] Y. W. Teh, M. Seeger, and M. I. Jordan, *Semiparametric latent factor models*, in *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics* (R. G. Cowell and Z. Ghahramani, eds.), vol. R5 of *Proceedings of Machine Learning Research*, pp. 333–340, PMLR, 06–08 Jan, 2005.

[123] M. Gu and W. Shen, *Generalized probabilistic principal component analysis of correlated data*, *The Journal of Machine Learning Research* **21** (2020), no. 1 428–468.

[124] R. Cerbino and V. Trappe, *Differential dynamic microscopy: probing wave vector dependent dynamics with a microscope*, *Physical review letters* **100** (2008), no. 18 188102.

[125] M. J. Heaton, A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, *et. al.*, *A case study competition among methods for analyzing large spatial data*, *Journal of Agricultural, Biological and Environmental Statistics* **24** (2019), no. 3 398–425.

[126] S. Conti and A. O'Hagan, *Bayesian emulation of complex multi-output and dynamic computer models*, *Journal of statistical planning and inference* **140** (2010), no. 3 640–651.

[127] R. Paulo, G. García-Donato, and J. Palomo, *Calibration of computer models with multivariate output*, *Computational Statistics and Data Analysis* **56** (2012), no. 12 3959–3974.

[128] D. L. Zimmerman, *Another look at anisotropy in geostatistics*, *Mathematical Geology* **25** (1993), no. 4 453–470.

[129] S. Särkkä and J. Hartikainen, *Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression*, in *International Conference on Artificial Intelligence and Statistics*, pp. 993–1001, 2012.

[130] P. Whittle, *Stochastic process in several dimensions*, *Bulletin of the International Statistical Institute* **40** (1963), no. 2 974–994.

[131] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. v. d. Vorst, *Templates for the solution of algebraic eigenvalue problems: a practical guide.* Society for Industrial and Applied Mathematics, 2000.

[132] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, *SIAM review* **51** (2009), no. 3 455–500.

[133] A. E. Gelfand, A. M. Schmidt, S. Banerjee, and C. Sirmans, *Nonstationary multivariate process modeling through spatially varying coregionalization*, *Test* **13** (2004), no. 2 263–312.

[134] M. E. Tipping and C. M. Bishop, *Probabilistic principal component analysis*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **61** (1999), no. 3 611–622.

[135] S. Banerjee, A. E. Gelfand, A. O. Finley, and H. Sang, *Gaussian predictive process models for large spatial data sets*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70** (2008), no. 4 825–848.

[136] N. Cressie and G. Johannesson, *Fixed rank kriging for very large spatial data sets*, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70** (2008), no. 1 209–226.

[137] M. L. Stein, *Limitations on low rank approximations for covariance matrices of spatial data*, *Spatial Statistics* **8** (2014) 1–19.

[138] S. H. Sørbye and H. Rue, *Simultaneous credible bands for latent Gaussian models*, *Scandinavian Journal of Statistics* **38** (2011), no. 4 712–725.

[139] F. Gerber, R. Furrer, G. Schaepman-Strub, R. de Jong, and M. Schaepman, *Predicting missing values in spatio-temporal satellite data*, *IEEE Transactions on Geoscience and Remote Sensing* **56** (2018) 2841–2853.

[140] D. Nychka, S. Bandyopadhyay, D. Hammerling, F. Lindgren, and S. Sain, *A Multiresolution Gaussian Process Model for the Analysis of Large Spatial Datasets*, *Journal of Computational and Graphical Statistics* **24** (2015), no. 2 579–599.

[141] M. J. Heaton, W. F. Christensen, and M. A. Terres, *Nonstationary Gaussian Process Models Using Spatial Hierarchical Clustering from Finite Differences*, *Technometrics* **59** (2017), no. 1 93–101.

[142] S. S. Shen, *R programming for climate data analysis and visualization: computing and plotting for NOAA data applications.* San Diego State University, USA., 2017.

[143] A. Zammit-Mangion and N. Cressie, *FRK: An R Package for Spatial and Spatio-Temporal Prediction with Large Datasets*, *Journal of Statistical Software* **98** (2021), no. 4 1–48.

[144] R. B. Gramacy, *laGP: large-scale spatial modeling via local approximate Gaussian processes in R*, *Journal of Statistical Software* **72** (2016), no. 1 1–46.

[145] J. Wu and M. Gu, *Perfecting liquid-state theories with machine intelligence*, *The Journal of Physical Chemistry Letters* **14** (2023), no. 47 10545–10552.

[146] S. Eftekharnejad, V. Vittal, G. T. Heydt, B. Keel, and J. Loehr, *Impact of increased penetration of photovoltaic generation on power systems*, *IEEE Transactions on Power Systems* **28** (2013), no. 2 893–901.

[147] T. Liu, Y. Song, L. Zhu, and D. J. Hill, *Stability and control of power grids*, *Annual Review of Control, Robotics, and Autonomous Systems* **5** (2022), no. 1 689–716.

[148] E. Torre, S. Marelli, P. Embrechts, and B. Sudret, *A general framework for data-driven uncertainty quantification under complex input dependencies using vine copulas*, *Probabilistic Engineering Mechanics* **55** (2019) 1–16.

[149] C. Canizares *et. al.*, *Benchmark models for the analysis and control of small-signal oscillatory dynamics in power systems*, *IEEE Transactions on Power Systems* **32** (2017), no. 1 715–722.

[150] M. N. Kurt, Y. Yılmaz, and X. Wang, *Real-time nonparametric anomaly detection in high-dimensional settings*, *IEEE transactions on pattern analysis and machine intelligence* **43** (2020), no. 7 2463–2479.

[151] M. Gu and H. Li, *Gaussian orthogonal latent factor processes for large incomplete matrices of correlated data*, *Bayesian Analysis* **17** (2022), no. 4 1219–1244.

[152] R. Killick, I. A. Eckley, K. Ewans, and P. Jonathan, *Detection of changes in variance of oceanographic time-series using changepoint analysis*, *Ocean Engineering* **37** (2010), no. 13 1120–1126.

[153] Y. Li, R. Lund, and A. Hewaarachchi, *Multiple changepoint detection with partial information on changepoint times*, .

[154] R. Duan, Y. Ning, and Y. Chen, *Heterogeneity-aware and communication-efficient distributed statistical inference*, *Biometrika* **109** (2022), no. 1 67–83.

[155] A. Shojaie and E. B. Fox, *Granger causality: A review and recent advances*, *Annual Review of Statistics and Its Application* **9** (2022) 289–319.

[156] H. Zou, T. Hastie, and R. Tibshirani, *Sparse principal component analysis*, *Journal of computational and graphical statistics* **15** (2006), no. 2 265–286.

[157] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[158] W. Gu, J. Choi, M. Gu, H. Simon, and K. Wu, *Fast change point detection for electricity market analysis*, in *2013 IEEE International Conference on Big Data*, pp. 50–57, IEEE, 2013.

[159] Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, R. Ren, K. S. Leung, E. H. Lau, J. Y. Wong, *et. al.*, *Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia*, *New England Journal of Medicine* (2020) [https://doi.org/10.1056/NEJMoa2001316].

[160] C. for Disease Control, Prevention, *et. al.*, *Duration of Isolation and Precautions for Adults with COVID-19*, *Atlanta, GA: CDC* (2020).