THE UNIVERSITY OF CHICAGO

SPARSITY, NONCONVEXITY AND GEOMETRIC CONSTRAINTS ON STATISTICAL INVERSE PROBLEMS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

COMMITTEE ON COMPUTATIONAL AND APPLIED MATHEMATICS

BY
HWANWOO KIM

CHICAGO, ILLINOIS
JUNE 2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Daniel Sanz-Alonso, for his guidance, and patience throughout my PhD journey. His consistent support since my arrival in Chicago made this work possible and contributed to my personal and professional growth.

I am also grateful to my committee members, Alexander Strang and Yuxin Chen for their guidance on research projects and serving as a committee member. Their constructive critiques, suggestions, and encouragement contributed to the refinement of my work. I also thank Panos Toulis for his guidance on various research projects, intellectual discussions, and career advice.

I would like to thank many fantastic PhD students who were around me. A special thanks to my cohort members Phillip, Richard, and Alex, from whom I learned a lot in the early stage of my PhD. I would like to thank Ruiyi and Yuming for their support during my Ph.D., especially in my job search, and for their detailed explanations of their research. Thanks to Omar, Nathan, Jiajun, and Jiaheng, I was exposed to many theoretical aspects of statistics and computational mathematics.

I also thank Mary Silber, Guillaume Bal, Zellencia Harris, and Jonathan Rodriguez who have handled many administrative tasks behind the scenes to make my PhD journey possible. Also special thanks to many collaborators including Jerry Chee, Chong Liu, Shixiao Jiang, John Harlim, and Shiv Agrawal, with whom I had interacted during my time at UChicago and got help on various projects. I also thank all my tennis mates and Korean friends during my time in Chicago.

Most importantly, heartfelt appreciation goes to my parents, whom I admire the most, my brother, and my lovely partner. Their love, support, and sacrifice provided me with the strength I needed to pursue this endeavor.

ABSTRACT

This dissertation delves into statistical inverse problems with a focus on Bayesian approaches for parameter estimation and uncertainty quantification under sparsity, nonconvexity, and geometric constraints. The dissertation covers innovative methodologies for addressing these challenges across various contexts, including compressed sensing, dynamical systems learning, and parameter estimation of differential equations on Euclidean space and manifolds. The work encompasses various methodologies based on mean-field variational inference, ensemble Kalman methods, Bayesian optimization, and graph Gaussian process to obtain point estimates for the quantity of interest as well as comprehensive uncertainty quantification associated with it. The dissertation effectively showcases how the introduced methods improve computational efficiency and accuracy in parameter estimation and uncertainty analysis across complex models. This is achieved through a blend of theoretical insights and numerical studies, inspired by a wide array of practical scenarios.

CHAPTER 1

INTRODUCTION

1.1 Statistical Inverse Problems

Inverse problems are ubiquitous in the realm of scientific discovery and technological innovation, presenting a set of computational and methodological challenges that stem from the need to infer quantities of interest from indirect observations. In short, the goal of inverse problems is to uncover underlying parameters of interest from observed outcomes, often in the presence of noise and uncertainty. The field of statistical inverse problems [O'Sullivan, 1986, Evans and Stark, 2002, Kaipo and Somersalo, 2006] focuses on the application of statistical principles and computational methods to these challenges, offering a structured approach to quantify uncertainty, incorporate prior knowledge, and derive meaningful inferences about the latent processes governing observed data. The significance of statistical inverse problems extends across a wide range of fields, from geophysical exploration, where they are used to understand the subsurface properties of the Earth, to biomedical engineering, for enhancing image reconstruction techniques in medical imaging [Somersalo et al., 1992, Kaipo and Somersalo, 2006, Stuart, 2010, Iglesias et al., 2016]. Each application not only underscores the pervasiveness of statistical inverse problems but also highlights the critical role of statistical and computational tools in advancing our understanding and capabilities within these domains.

In this thesis, we adopt the Bayesian framework to solve statistical inverse problems [Kaipo and Somersalo, 2006, Stuart, 2010, Dashti and Stuart, 2017]. This framework is rooted in Bayes' theorem, which updates prior beliefs about unknown parameters based on observations, thereby yielding a posterior distribution that encapsulates both the uncertainties in the measurements and the prior information. This approach is particularly powerful for dealing with complex systems where direct measurements of the parameters of interest are not possible, and where uncertainty plays a critical role in the modeling process.

Throughout this thesis, we assume the following data-generating mechanism.

$$y = \mathcal{G}(u) + \eta, \quad u \in \mathcal{U}, \quad y \in \mathcal{Y},$$

where u is the parameter of interest, $\mathcal{U}$ is the parameter space, y is the observed data, $\mathcal{Y}$ is the data space, $\mathcal{G}$ is the forward map between the parameter space and the data space, and η is random measurement error. Based on the data-generating mechanism given above, Bayes' theorem can be succinctly expressed as

$$\pi(u|y) \propto \mathcal{L}(y - \mathcal{G}(u))\pi(u),$$

where $\mathcal{L}$ is the data-likelihood induced by the probabilistic assumption on the measurement errors, and $\pi(u), \pi(u|y)$ are respectively the prior/posterior distribution of u. The Bayesian approach to inverse problems is then characterized by providing a pointwise estimate for the parameter of interest that reflects observed data and its comprehensive quantification of uncertainty through the posterior distribution. Each of the following chapters in this thesis will deal with various assumptions on the parameter u, the forward map $\mathcal{G}$, and the parameter space $\mathcal{U}$. We will provide novel statistical and computational tools that exploit imposed assumptions to facilitate statistical inversion.

1.2 Sparsity

In many scientific and engineering applications, the underlying physical phenomena can be represented as sparse signals meaning that their significant information can be captured with a relatively small number of non-zero coefficients (or parameters) in a suitable basis or representation. Leveraging such sparse structures in data, fields like compressed sensing, and high-dimensional statistics, have revolutionized the ability to recover or reconstruct these sparse signals from a limited number of observations [Candès and Wakin, 2008, Foucart et al., 2013, Bühlmann and Van De Geer, 2011, Hastie et al., 2015]. This principle has found widespread applications, ranging from medical imaging, where it enables high-resolution reconstructions from fewer measurements, thereby reducing exposure to harmful radiation, to genetics and genomics by enabling the identification of relevant genetic markers associated with various traits or diseases while simultaneously handling the high-dimensional nature of genomic data. Exploiting sparse structures not only enhances computational efficiency and accuracy but also notably mitigates challenges coming from ill-posedness and noise in estimating parameters (or coefficients), marking a notable advancement in the field of inverse problems.

In the Bayesian framework, numerous priors that leverage sparse structures in data have been introduced [Carvalho et al., 2010, Brown and Griffin, 2010, Ročková and George, 2018, Ročková, 2018, Calvetti et al., 2019b]. Of particular interest is the hierarchical Bayesian model introduced in [Brown and Griffin, 2010, Calvetti et al., 2019b]. Reflecting the sparse structures in the parameter of a linear model, they impose a zero-centered Gaussian prior with gamma hyperpriors on the variance parameter. Through zero-mean Gaussian prior, one is implicitly reflecting the underlying sparse structure in the parameter and the variance parameter effectively controls the level of sparsity.

To be more specific, consider the following model,

$$y = Au + \eta,$$

where u is the parameter of interest, y is the observations, and η is a Gaussian noise with zero mean and covariance Γ. [Brown and Griffin, 2010, Calvetti et al., 2019b] assume that u follows a zero-centered Gaussian prior with gamma hyperpriors on its variance parameter

θ. Then one can show that

$$p(u, \theta|y) \propto \exp\bigl(-J(u, \theta)\bigr),$$

where

$$J(u, \theta) := \frac{1}{2}\|y - Au\|_{\Gamma}^2 + \frac{1}{2}\|u\|_{D_\theta}^2 + \sum_{i=1}^{d} \left[\frac{\theta_i}{\alpha} - \Bigl(\beta - \frac{3}{2}\Bigr) \log \frac{\theta_i}{\alpha} \right]. \tag{1.1}$$

Here α, β are parameters for the gamma hyperpriors. Although the posterior distribution is intractable, thanks to the form of the Gibbs energy function J given in (1.1), [Calvetti et al., 2019b] introduced a coordinate-wise optimization strategy known an Iterative Alternating Scheme (IAS) to obtain a point estimate, the maximum a posteriori (MAP) estimate of the parameter u. They have shown that as the parameter β tends to $\frac{3}{2}$, the optimum of (1.1) converges to the optimum of ℓ_1-regularized least squares solution, justifying a Bayesian framework that subsumes Laplace prior set up, which corresponds to the famous Least Absolute Shrinkage and Selection Operator (LASSO) [Tibshirani, 1996] algorithm in the frequentist setup.

In the meantime, [Calvetti et al., 2019b] did not provide a full Bayesian solution, in the sense that the uncertainty associated with their MAP estimate was not quantified. To acquire a full Bayesian solution to parameter u, instead of utilizing Markov Chain Monte Carlo (MCMC) algorithms like in [Brown and Griffin, 2010], in Chapter 2, we utilize techniques from variational inference [Blei et al., 2017] to approximate target posterior distribution of u with a tractable probability distribution that belongs to the mean-field family. Such an approach scales better in the dimension of the parameter u and allows one to acquire both MAP estimate and credible interval for the target parameter.

Furthermore, in Chapter 3, we extend the aforementioned hierarchical model framework to the nonlinear forward map setting through methodologies based on ensemble Kalman filter algorithms. Viewing variants of ensemble Kalman filter methods as particle-based stochastic nonlinear optimization techniques, we iteratively optimize

$$J(u,\theta) := \frac{1}{2}\|y-\mathcal{G}(u)\|_\Gamma^2 + \frac{1}{2}\|u\|_{D_\theta}^2 + \sum_{i=1}^{d}\left[\frac{\theta_i}{\alpha} - \left(\beta - \frac{3}{2}\right)\log\frac{\theta_i}{\alpha}\right], \tag{1.2}$$

with respect to u and θ, like in the IAS procedure. In the setting where the forward map is nonlinear with sparsity assumption on the parameter u, the introduced algorithms will quantify uncertainties of the output through particles propagated according to some iterative update rules.

1.3 Nonconvexity

In statistical inverse problems, the objective functions optimized to obtain point estimates are often nonconvex. Nonconvexity has become increasingly common due to the surge in the complexity of models and data in various science and engineering disciplines. This rise is primarily driven by the need to capture the intricate relationships between variables in real-world data, which often necessitate models with nonconvex objective functions to accurately represent the underlying phenomena. As the ambition to solve more sophisticated problems grows, so does the reliance on models that introduce nonconvex landscapes and lack closed-form mathematical expressions, presenting both a challenge and an opportunity for innovation in optimization techniques. The shift towards embracing these complicated models is fueled by their potential to provide deeper insights, more accurate predictions, and solutions to previously intractable problems, pushing the boundaries of what can be achieved across scientific and engineering disciplines. Consequently, the development and refinement of algorithms capable of effectively navigating nonconvex landscapes have become a crucial area of research, aiming to unlock the full capabilities of these advanced models.

In numerous Bayesian inverse problems, the forward map $\mathcal{G}$ often involves a solution of differential equations with various types of incomplete observations. To acquire a point estimate, i.e., maximum a posteriori estimate, one naturally needs to seek an optimum of an objective function with nonconvex landscapes [Cleary et al., 2021, Lan et al., 2023, Schneider et al., 2022]. To give a concrete example, consider the following Rossler dynamical system given by

$$
\begin{aligned}
\frac{dx}{dt} &= -y - z, \\
\frac{dy}{dt} &= x + 0.2y, \\
\frac{dz}{dt} &= 0.2 + z(x - u),
\end{aligned}
$$

with an initial condition $x(0) = 1, y(0) = 0$, and $z(0) = 1$. One can attempt to estimate the parameter u from the observed data. Following the setup in [Cleary et al., 2021, Schneider et al., 2022], if the available data is partial information about the solution trajectory of the Rossler dynamical system, for instance, the first and second moments of the solution trajectory over a time window [30, 50], the landscape of objective functions for the parameter can be non-convex as one can see from Figure 1.1. Furthermore, there is no closed-form expression for the objective function, due to the lack of an analytic solution of the Rossler dynamical system. The objective function can only be numerically evaluated based on numerical differential equation solvers, which could be expensive.

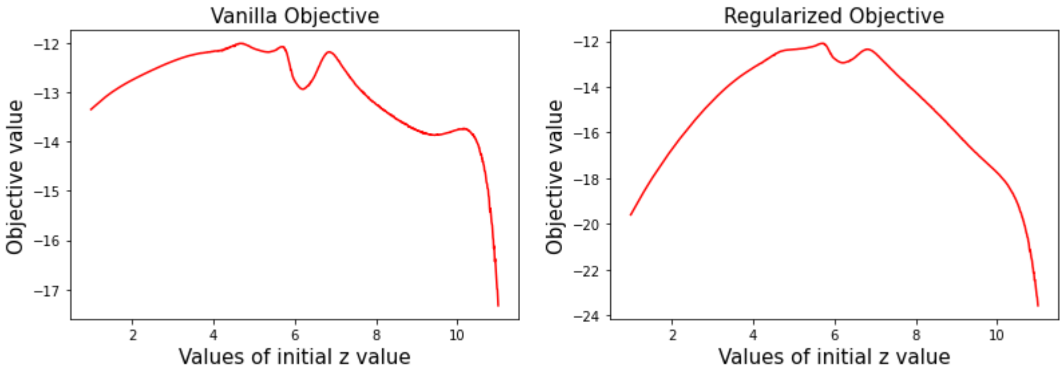


Figure 1.1: Objective function: $\frac{1}{2}\|y - \mathcal{G}(u)\|^2_{\hat{\Gamma}}$, $\hat{\Gamma}$: estimated noise covariance (more details in [Cleary et al., 2021, Schneider et al., 2022]), Left: with no regularization, Right: with ℓ_2 regularization

To address this challenge, one can leverage popular black-box global optimization techniques, generically labeled as Bayesian optimization methods [Jones et al., 1998, Shahriari et al., 2015, Frazier, 2018]. Loosely speaking, for some pre-specified number of iterations T, the Bayesian optimization methodologies based on Gaussian process [Stein, 2012, Williams and Rasmussen, 2006] can be summarized as follows: into the following procedures [Frazier, 2018]:

- Place a Gaussian process prior to the target objective function f.
- For $t = 1, \ldots, T$:
    1. Derive Gaussian process posterior using all available query locations $X_{t-1} \coloneqq \{x_1, \ldots, x_{t-1}\}$, and function evaluations $F_{t-1} \coloneqq \{f(x_1), \ldots, f(x_{t-1})\}$.
    2. Obtain the next query location x_t as the maximizer of the acquisition function α_t which depends on the Gaussian process posterior.
    3. Observe $y_t = f(x_t)$ or $y_t = f(x_t) + \epsilon_t$, where ϵ_t is a random observation noise.
    4. Set $X_t = X_{t-1} \cup \{x_t\}$ and $F_t = F_{t-1} \cup \{f(x_t)\}$
    5. Go back to Step 2 and repeat.

From the step-by-step description provided above, unlike many existing optimization algorithms that rely on first-order or second-order information about the objective function, Bayesian optimization algorithms do not necessarily require derivative information or formulaic expressions of the objective function. The method only requires evaluative functions and is thus a natural optimization tool for solving statistical inverse problems with complex forward maps. Furthermore, even for a non-convex function, there have been numerous asymptotic convergence results obtained for Bayesian optimization algorithms depending on the choice of kernel function for the Gaussian process, the acquisition function, and the assumptions placed on the objective functions [Bull, 2011, Srinivas et al., 2010, Chowdhury and Gopalan, 2017, Vakili et al., 2021, Russo and Van Roy, 2014].

Typically, the convergence of the algorithm is established by showing a sublinear growth rate of the quantify known as the cumulative regret bound, given by $R_T = \sum_{t=1}^{T} f(x^*) - f(x_t)$, where x^* is the maximizer of f and x_t is the t-th query location chosen by the Bayesian optimization algorithm. As the simple regret $r_t = f(x^*) - \max_{t=1,\cdots,T} f(x_t)$ is bounded above by $\frac{R_T}{T} = \frac{1}{T}\sum_{t=1}^{T} r_t$, a sublinear growth rate of the Bayesian optimization algorithm naturally translates to the convergence to the global maximum value. Most of the aforementioned works established regret bounds through the quantity known as the maximum information gain, which can only be defined in the setting where the function evaluation is corrupted by some random noise. Therefore, there has been a natural gap between theory and optimization practice. In Chapter 4 of this thesis, we introduce two novel Bayesian optimization methodologies that nearly achieve the optimal simple regret bound shown in [Bull, 2011].

1.4 Geometric constraints

Many real-world systems exhibit complex behavior that cannot be adequately captured by traditional mathematical models defined on Euclidean spaces. Instead, these systems often evolve over manifolds—geometric spaces with intrinsic curvature and structure. Mathematical models defined on manifolds offer a powerful framework for describing such systems, enabling a more succinct representation of their underlying dynamics. Understanding these models is essential for tackling challenges in fields such as brain imaging [Mémoli et al., 2004], biomembranes [Elliott and Stinner, 2010], robotics [Jaquier et al., 2020, 2022], and liquid crystals [Virga, 2018], where systems are inherently constrained by the geometry of their environment. Incorporating such geometric constraints on the input of the mathematical models and parameter space, researchers can develop tailored mathematical models that account for the manifold's geometric properties, yielding deeper insights into the behavior of complex systems.

For partial differential equations defined on manifolds, several works have studied how to estimate parameters of the differential equation in Bayesian fashion [Harlim et al., 2020, Garcia Trillos and Sanz-Alonso, 2018]. To build probabilistic tools to facilitate Bayesian inference, a central object described in Chapther 5 and 6 is the graph Laplacian, which encodes the underlying geometric information through a point cloud of manifold data. Given a set of points $\{x_1, \cdots, x_N\}$ lying on underlying manifold $\mathcal{M}$, the (unnormalized) graph Laplacian can be constructed through $N \times N$ similarity matrix W, which measures closeness between elements in point cloud under suitable notion of distance. The (unnormalized) graph Laplacian Δ_N is then $D - W$, where D is a diagonal matrix whose elements are simply the row sum of W [Von Luxburg, 2007]. Thanks to spectral convergence results of the graph Laplacian to the Laplace-Beltrami operator on the manifold $\mathcal{M}$ [Burago et al., 2015, Garcia Trillos et al., 2020a], the eigenvectors of the graph Laplacian provide a natural basis to represent functions on a manifold.

Utilizing the graph Laplacian, [Harlim et al., 2020, Garcia Trillos and Sanz-Alonso, 2018] imposed a graph Laplacian-based Gaussian process prior to the parameter of interest and leveraged graph-based discretized forward map to facilitate Bayesian inference. The strength of these approaches is that they do not require explicit geometric information of the manifold such as chart, tangent space, Riemannian gradient, and retraction maps. Just through the point cloud, they implicitly learn the underlying manifold to express mathematical objects defined on it.

Borrowing the ideas in previous works, in Chapter 5, we introduce a novel Bayesian optimization method for objective functions defined on an unknown manifold when the only source of available information about it is given through the point cloud. Based on the graph Gaussian process [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2020, 2021], we lift the Bayesian optimization strategy described in Chapter 4 to the unknown manifold set up and establish a cumulative regret bound of the algorithm. The algorithm is especially useful when the objective function is expensive to evaluate, complementing the weakness of the graph-based MCMC algorithms in previous works. We demonstrate its effectiveness in estimating the parameter of a heat equation defined on the sphere and finding an optimum of a function defined on a cow-shaped manifold.

Lastly, in Chapter 6, we extend the existing graph-based MCMC methodologies to the boundary-existing manifold setup. As the differential equation models are accompanied by boundary conditions of the solution, we introduce a novel graph-based Gaussian process that reflects the impact of boundary conditions. We demonstrate its effectiveness in parameter estimation problems for elliptic and parabolic partial differential equations defined on manifolds with boundaries. We supplement the point estimate with an MCMC-based posterior credible interval, which serves as the full Bayesian inference framework for differential equation models defined on manifolds with boundaries.

1.5 Outline and Main Contributions

We provide an outline of the upcoming chapters and summarize their main contributions.

- Chapter 2 is based on [Agrawal et al., 2022]. We first introduce the Laplace approximation approach to build credible intervals for IAS output. Next, utilizing results from the mean-field variational inference, we propose a variational iterative alternating scheme (VIAS), which serves as an iterative algorithm to find a probability distribution closest to the target posterior distribution under Kullback-Leibler divergence. The effectiveness of the VIAS algorithm over LASSO and Laplace approximation is demonstrated in sparse regression problems, deconvolution tasks, and Lorenz 63 dynamical systems learning.

- Chapter 3 is based on [Kim et al., 2023]. We provide an optimization perspective of ensemble Kalman methods and utilize it within the hierarchical Bayesian framework introduced in Chapter 2. Through particles generated from ensemble Kalman methods, one can build approximate credible intervals of the point estimate for the parameter of interest, in a nonlinear forward map setup. We demonstrate its effectiveness in a compressed sensing problem, a PDE parameter estimation with a known explicit forward map, and an elliptic inverse problem with a numerical forward map.

- Chapter 4 is based on [Kim and Sanz-Alonso, 2024]. In this work, we explore Bayesian optimization with noise-free observations, presenting novel algorithms that leverage results from scattered data approximation. These algorithms incorporate a random exploration step, ensuring a near-optimal reduction in the fill distance between query points. The algorithms maintain the simplicity of existing Bayesian optimization methods and achieve cumulative regret bounds close to those proposed in [Vakili, 2022]. Moreover, the proposed methodologies demonstrate superior performance over other Bayesian optimization techniques across various tasks, including optimization of benchmark non-convex functions, machine learning model hyperparameter tuning, and engineering design challenges for garden sprinkler systems.

- Chapter 5 is based on [Kim et al., 2024]. We introduce a novel Bayesian optimization algorithm for objective functions defined on an unknown manifold. Utilizing graph Gaussian process [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2021], we build a surrogate of the objective function along the point cloud and seek the optimum of the objective function over the same point cloud. We show that as the iteration number and the size of the point cloud grow to infinity, the algorithm successfully converges to the global optimum of the objective function defined on the underlying true manifold. We demonstrate the effectiveness of the proposed algorithm on benchmark functions defined on one-dimensional manifolds, an objective function defined on a cow manifold, and parameter estimation of the heat equation defined on a sphere.

- Chapter 6 is based on [Harlim et al., 2022]. We provide a graph-based Bayesian approach to proceed with statistical inversion for parameters of the partial differential equation models defined on manifolds with boundaries. Unlike the existing methodology [Harlim et al., 2020], where authors assumed the underlying closed manifold is boundary-free, we relax such assumption and augment the graph based Mateŕn type Gaussian field [Sanz-Alonso and Yang, 2022a, Harlim et al., 2020] with boundary components to reflect partial differential equations' boundary conditions. With the help of the Ghost point diffusion map algorithm [Jiang and Harlim, 2023], we demonstrate the effectiveness of the proposed methodology for estimating parameters of elliptic and parabolic partial differential equations defined on a manifold with boundaries.

CHAPTER 2

A VARIATIONAL INFERENCE APPROACH TO INVERSE PROBLEMS WITH GAMMA HYPERPRIORS

2.1 Introduction

This chapter introduces a variational inference approach that enables uncertainty quantification for hierarchical Bayesian inverse problems with gamma hyperpriors. The hierarchical model that we consider, along with an iterative alternating scheme (IAS) to compute the *maximum a posteriori* (MAP) estimate, were introduced and analyzed in [Calvetti et al., 2020b, 2019b, 2020a, 2019a, 2015]. These papers provide strong evidence of the flexibility of the hierarchical model and show that the IAS algorithm is easy to implement and globally convergent. However, despite the Bayesian motivation for the hierarchical model, previous work has only considered MAP estimation and the potential to perform uncertainty quantification has not yet been realized. Using the general framework of variational inference, we introduce a variational iterative alternating scheme (VIAS) that shares the flexibility and ease of implementation of IAS, while enabling uncertainty quantification and model selection.

The hierarchical Bayesian model that we consider gives a posterior density $p(u, \theta \,|\, y)$ for the unknown quantity of interest $u \in \mathbb{R}^d$ and parameters $\theta \in \mathbb{R}^d$ given observed data $y \in \mathbb{R}^n$. The goal of IAS is to find the MAP estimator, that is, the pair (u^*, θ^*) that maximizes the posterior density. This leads to an optimization problem, which IAS solves by producing iterates (u^k, θ^k), $k \geq 1$, satisfying

$$\begin{aligned} u^{k+1} &= \arg\max_{u} p(u, \theta^k \,|\, y), \\ \theta^{k+1} &= \arg\max_{\theta} p(u^{k+1}, \theta \,|\, y). \end{aligned} \tag{2.1}$$

In contrast, the goal of our proposed VIAS method is to find the density $q^*(u, \theta)$ that is closest to the posterior $p(u, \theta \,|\, y)$ in Kullback-Leibler divergence, within the mean-field family of distributions of the form $q(u, \theta) = q(u)\, q(\theta)$. This leads to an optimization problem over densities, which VIAS solves by producing iterates $q^k(u, \theta) = q^k(u)\, q^k(\theta)$, $k \geq 1$, satisfying

$$\begin{aligned} q^{k+1}(\theta) &= \arg\min_{q(\theta)} d_{\text{KL}}\big(q^k(u)\, q(\theta) \,\|\, p(u, \theta \,|\, y)\big), \\ q^{k+1}(u) &= \arg\min_{q(u)} d_{\text{KL}}\big(q(u)\, q^{k+1}(\theta) \,\|\, p(u, \theta \,|\, y)\big), \end{aligned} \tag{2.2}$$

where d_{KL} denotes the Kullback-Leibler divergence. Approximate Bayesian inference can then be performed using the variational distribution $q^*(u, \theta)$, which will be shown to be tractable, rather than the posterior $p(u, \theta \,|\, y)$. Due to the tractability of $q^*(u, \theta)$, point estimates and credible intervals can be efficiently computed with the variational distribution, while doing so with the true posterior would be computationally challenging.

Central to the implementation of IAS is the fact that the maximizers u^{k+1} and θ^{k+1} in (2.1) can be obtained in closed form, by exploiting the structure of the hierarchical model with gamma hyperpriors. A similar property is satisfied by VIAS. Indeed, our choice of mean-field admissible densities ensures that the minimizers $q^{k+1}(u)$ and $q^{k+1}(\theta)$ in (2.2) are, respectively, Gaussian and generalized inverse Gaussian densities. We will derive closed formulas for the iterative updating of the parameters of these distributions.

Despite their shared structure, there are some fundamental differences between IAS and VIAS. While IAS only gives a point estimate i.e the MAP, VIAS gives a variational distribution that approximates the posterior. This variational distribution can be used to understand the covariance structure and find credible intervals for the estimates. However, it is worth emphasizing that VIAS only provides an approximation to the posterior, and therefore point estimates or credible intervals constructed with VIAS will only give approximate posterior inference. In contrast, IAS converges to the true MAP estimate. The primary advantage of VIAS is its potential to provide meaningful uncertainty quantification. We will also show that the variational perspective lends itself naturally to model selection for the choice of hyperparameters. An advantage of IAS is that it converges globally to the MAP estimate due to the convexity of the log-posterior density, while VIAS is, in general, only guaranteed to converge to a local maximizer of the optimization problem (2.2). We will demonstrate the potential emergence of spurious local maxima in the VIAS objective function for extreme data realizations and hyperparameter values, and describe how convergence to the global maximizer can be achieved in practice by suitable initialization of the variational algorithm.

2.1.1 Related Work

This chapter, among others, introduces variational inference techniques [Bishop, 2006, Jordan et al., 1999, Wainwright and Jordan, 2008, Blei et al., 2017] to Bayesian inverse problems [Tarantola, 2015, Kaipo and Somersalo, 2006, Calvetti and Somersalo, 2007, Stuart, 2010, Sanz-Alonso et al., 2023], where computational approaches are often based on MAP estimation [Kaipo and Somersalo, 2006], Monte Carlo and measure transport sampling [Liu, 2008, Agapiou et al., 2017, Marzouk et al., 2016], or iterative Kalman methods [Chada et al., 2021]. Some recent works that have investigated the use of variational inference for inverse problems include [Maestrini et al., 2021, Tonolini et al., 2020]. Variational inference has a comparable computational cost to MAP estimation, but has two main advantages: (i) it can provide uncertainty quantification, and (ii) it lends itself naturally to conduct model selection. In addition, the variational distribution can be used as a proposal mechanism for Monte Carlo sampling algorithms. A simple but popular alternate way to probe the posterior is to find its Laplace approximation, namely the Gaussian centered at the MAP whose covariance is given by the inverse Hessian of the negative log-posterior density. The Kullback-Leibler accuracy of Laplace approximations was investigated in [Dehaene, 2017] and the Hellinger accuracy in inverse problems with small noise was established in [Schillings et al., 2020]. However, Laplace approximations can be ineffective in large-noise or small-data regimes, where the posterior may not be well approximated by a Gaussian measure. In addition, computing the inverse Hessian can be prohibitively expensive in high dimensional nonlinear inverse problems. Monte Carlo methods can provide accurate posterior inference while variational inference is based on an approximation to the posterior; however, Monte Carlo methods often require a large number of samples and hence a large number of forward model evaluations, which can be costly. In addition, tuning Monte Carlo methods and assessing their convergence can be challenging. For hierarchical Bayesian models, the Gibbs sampler alleviates the need of tuning [Damlen et al., 1999], but the chain may still converge slowly for highly anisotropic target densities [Agapiou et al., 2014, Roberts and Sahu, 1997].

As mentioned above, we will consider a hierarchical Bayesian model with gamma hyperpriors introduced and analyzed in [Calvetti et al., 2020b, 2019b, 2020a, 2019a, 2015]. The paper [Calvetti et al., 2020b] investigates generalized gamma hyperpriors and [Calvetti et al., 2020a] discusses hybrid solvers for MAP estimation that can improve on IAS. The hierarchical model and IAS algorithm have been shown to be successful in realistic inverse problems including brain activity mapping from MEG [Calvetti et al., 2015, 2019a]. These papers also emphasize the flexibility of the model and its ability to provide useful regularization for sparse signals [Calvetti et al., 2019b, 2020b,a]. As described in [Calvetti et al., 2019b], the IAS algorithm is inspired by classical iterative reweighted least squares [Green, 1984] and related work [Gorodnitsky and Rao, 1997, Daubechies et al., 2010] on signal processing with emphasis on sparsity. Sparsity-promoting algorithms and models are key in statistics applications [Tibshirani, 1996, Carvalho et al., 2009]. Our hierarchical approach is closely related to empirical Bayes statistical methods [Robbins, 1992] and to bilevel and data-driven methods for inverse problems [Bard, 2013, Arridge et al., 2019].

In a similar spirit as IAS, there is a vast literature on sparsity-promoting priors. These priors typically yield log-posterior densities containing L^p-regularization terms with $p \in (0, 1]$ [Park and Casella, 2008, Polson et al., 2014]. A variational approximation to the posterior under such sparsity-promoting priors is given in [Armagan, 2009]. Alternatively, [Carvalho et al., 2010] proposed a standard half-Cauchy prior on positive reals for standard deviations of u, which were assumed to follow a conditional Gaussian prior centered at zero. A recent work that utilizes a variational inference technique under the same conditional Gaussian prior but with scale-mixtures of generalized inverse Gaussians can be found in [Law and Zankin, 2022]. Also popular are spike and slab priors, defined by a mixture distribution with a Dirac mass at zero and a continuous distribution, see e.g. [Ročková and George, 2018].

2.1.2 Outline and Main Contributions

- Section 2.2 formalizes the problem of interest and reviews the hierarchical model with gamma hyperpriors.

- Section 2.3 describes the IAS algorithm. Building on previous work on IAS [Calvetti et al., 2019b], we derive and show the convergence of an iterative Laplace approximation to the posterior, used in Section 2.5 for numerical comparisons with our proposed VIAS.

- Section 2.4 introduces the novel VIAS and discusses its convergence. We will give all necessary background on variational inference.

- Section 2.5 demonstrates the accuracy of VIAS and its ability to provide meaningful uncertainty quantification in four computed examples. These examples include a deconvolution problem from [Calvetti et al., 2020b] and a new application of IAS and VIAS for data-driven sparse identification of dynamical systems [Brunton et al., 2016] from time series data. We also introduce and show the effectiveness of a model selection approach for the choice of hyperparameters. The Python code to reproduce our numerical results can be found at https://github.com/Hwanwoo-Kim/VIAS.

- We close in Section 2.6 with some research directions that stem from this work.

Notation For matrix P, we write $P \succ 0$ if P is positive definite. For $P \succ 0$, we denote by $\|\cdot\|_P^2 := |P^{-1/2}\cdot|^2$ the squared Mahalanobis norm induced by the matrix P, where $|\cdot|$ denotes the Euclidean norm.

2.2 Hierarchical Bayesian Model

In this section, we formalize the inverse problem of interest and the hierarchical model with gamma hyperpriors. We consider the following linear discrete inverse problem of recovering an unknown u from data y related by

$$y = Au + \eta, \tag{2.3}$$

where $A \in \mathbb{R}^{n\times d}$ is a given, possibly ill-conditioned, matrix and typically $d \geq n$. We assume that the noise term η is Gaussian distributed $\eta \sim (0, \Gamma)$ with given $\Gamma \succ 0$. Following [Calvetti et al., 2020b, 2019b, 2020a, 2019a, 2015], we adopt the following hierarchical Bayesian model, where the prior on u is conditionally Gaussian given a prior variance vector $\theta \in \mathbb{R}^d$:

$$\begin{aligned} y \,|\, u &\sim (Au, \Gamma), \\ u \,|\, \theta &\sim (0, D_\theta), \qquad\qquad D_\theta = \operatorname{diag}(\theta), \\ \theta_i &\sim \operatorname{Gamma}(\alpha_i, \beta), \quad 1 \leq i \leq d. \end{aligned} \tag{2.4}$$

Here α_i and β denote the shape and rate parameters, respectively. Our aim is to estimate $z := (u, \theta)$ given the observed data y. In the Bayesian approach to inverse problems [Kaipo and Somersalo, 2006, Stuart, 2010, Sanz-Alonso et al., 2023], inference is based on the posterior distribution which, for the hierarchical model (2.4), takes the form

$$p(u\,, \theta \,|\, y) = \frac{p(y \,|\, u, \theta) p(u \,|\, \theta) p(\theta)}{p(y)} \propto \exp\big(-\mathsf{J}(u, \theta)\big), \tag{2.5}$$

where

$$\mathsf{J}(u,\theta) := \overbrace{\frac{1}{2}\|y - Au\|_{\Gamma}^2 + \underbrace{\frac{1}{2}\|u\|_{D_\theta}^2}_{}}^{(a)} \underbrace{{}+ \textstyle\sum_{i=1}^{d}\left[\frac{\theta_i}{\alpha_i} - \left(\beta - \frac{3}{2}\right)\log\frac{\theta_i}{\alpha_i}\right]}_{(b)}. \tag{2.6}$$

Here (a) and (b) identify the two objectives that will be minimized iteratively by IAS. The α_i's act as scale parameters that control the expected size of θ_i, and, as a result, u_i^2 [Calvetti et al., 2020b]. They can be chosen automatically using the signal to noise ratio and expected cardinality of the support [Calvetti et al., 2019b]. Previous work [Calvetti et al., 2020b] has analyzed a whitening of the problem, setting all $\alpha_i = 1$, which simply amounts to a change of coordinates. In contrast, in our variational algorithm we will not perform such whitening, and the α_i's will determine the degree of shrinkage towards zero off the support of the unknown. For simplicity, we assume $\alpha_i = \alpha$ for all $i = 1, \ldots, d$, indicating a uniform shrinkage effect towards zero. The hyperparameter β controls how sharply $\mathsf{J}(u,\theta)$ penalizes non-sparse inputs for small but non-zero values off the support. In the limit as β converges to 3/2, the MAP estimator given by the minimizer of $\mathsf{J}(u,\theta)$ converges to the solution to an L^1 penalized problem [Calvetti et al., 2019b]. We refer to [Calvetti et al., 2015] for further background and motivation on the use of hierarchical gamma hyperpriors in Bayesian inverse problems which, contrary to common practice in statistics, do not lead to a conjugate model.

2.3 MAP Estimation and Laplace Approximation

This section is organized as follows. Subsection 2.3.1 overviews the IAS algorithm for MAP estimation and introduces an iterative Laplace approximation method. Subsection 2.3.2 reviews a convergence result for the IAS algorithm, from which we deduce convergence of the iterative Laplace approximation method.

2.3.1 The Iterative Alternating Scheme and Laplace Approximation

The MAP estimate of $z = (u, \theta)$ is, by definition, the maximizer of the posterior $p(z \,|\, y)$ or, equivalently, the minimizer of $\mathsf{J}(z)$. The papers [Calvetti et al., 2020b, 2019b, 2020a, 2019a, 2015] proposed, analyzed, and implemented an Iterative Alternating Scheme (IAS) for MAP estimation in a variety of inverse problems. The IAS consists of two separate minimization steps:

1. **Initialize** θ^0, $k = 0$.
2. **Iterate until convergence:**
    - (i) Update $u^{k+1} = \arg\min_u \mathsf{J}(u, \theta^k)$.
    - (ii) Update $\theta^{k+1} = \arg\min_\theta \mathsf{J}(u^{k+1}, \theta)$.
    - (iii) $k \to k + 1$.

Let $z^k := (u^k, \theta^k)$. Clearly $\mathsf{J}(z^k)$ is monotonically decreasing in k. Under suitable assumptions on the hyperparameters, to be made precise in Proposition 2.3.2 below, J is convex and IAS converges to the global minimizer of J. In other words, z^k convereges to the MAP estimator. In addition to this convergence guarantee, the IAS algorithm is simple to implement because of the structure of the energy functional J. Indeed, in step (i) only the u-dependent part (a) in (2.6) needs to be considered, and in step (ii) only the θ-dependent part (b) is needed. This results in straightforward implementation of both steps, as we describe next.

Updating u

The update of the u component boils down to solving a standard linear least-squares problem, which admits a closed form solution

$$\begin{aligned}\arg\min_u \mathsf{J}(u,\theta) &= \arg\min_u \frac{1}{2}\|y - Au\|_\Gamma^2 + \frac{1}{2}\|u\|_{D_\theta}^2 \\ &= (A^\top \Gamma^{-1} A + D_\theta^{-1})^{-1} A^\top \Gamma^{-1} y.\end{aligned} \tag{2.7}$$

In practice, when the dimension d of the unknown is much larger than the dimension n of the data, this linear least-squares problem can be effectively solved using conjugate gradient together with an early stopping based on Morozov's discrepancy principle. This approach has been applied in [Calvetti et al., 2020b, 2019b, 2020a, 2019a, 2015], and further analyzed in [Calvetti et al., 2018]. In such underdetermined problems, inverting in d-dimensional space can also be avoided using the Sherman-Morrison-Woodbury lemma, which gives the following equivalent Kalman-type update

$$\arg\min_u \mathsf{J}(u,\theta) = Gy, \qquad G := D_\theta A^\top (A D_\theta A^\top + \Gamma)^{-1}, \tag{2.8}$$

where the matrix G is called the Kalman gain.

Note that (2.7) can be rewritten as

$$\arg\min_u \mathsf{J}(u,\theta) = \arg\min_u \frac{1}{2}\|y - Au\|_\Gamma^2 + \frac{1}{2}\sum_{i=1}^{d} \frac{u_i^2}{\theta_i},$$

which shows that D_θ controls the sparsity of the solution u, with smaller θ_i leading to more shrinkage of u_i towards zero. Therefore, in the hierarchical Bayesian model setup, the variance parameter θ not only determines the variation of the parameter u but also the level of sparsity of u.

Algorithm 2.3.1 Iterative Alternating Scheme (IAS)

```
Input: Data $y$, matrix $A$. Prior hyperparameters: $\alpha, \beta$.
Initialize $\theta^0$, $k = 0$.
For $k = 0, 1, \dots$ until convergence do:
    (i) Set $D_\theta = \mathrm{diag}(\theta^k)$ and update
        $u^{k+1} = (A^\top \Gamma^{-1} A + D_\theta^{-1})^{-1} A^\top \Gamma^{-1} y.$
    (ii) Update
        $\theta_i^{k+1} = \alpha\left(\frac{\tilde{\beta}}{2} + \sqrt{\frac{\tilde{\beta}^2}{4} + \frac{(u_i^{k+1})^2}{2\alpha}}\right), \quad \tilde{\beta} = \beta - 3/2.$
end for
Output: Approximation to the MAP estimator $(u^{k+1}, \theta^{k+1}) \approx \arg\max p(z \mid y)$.
```

Updating θ

As shown in [Calvetti et al., 2019b], the update of the θ-component part (b) in (2.6) can be obtained by direct computation of a critical point of $\mathsf{J}(z)$ as follows

$$\arg\min_{\theta} \mathsf{J}(u, \theta) = \alpha\left(\frac{\tilde{\beta}}{2} + \sqrt{\frac{\tilde{\beta}^2}{4} + \frac{u_i^2}{2\alpha}}\right), \quad \tilde{\beta} = \beta - 3/2.$$

A pseudo-code for the IAS algorithm is given in Algorithm 2.3.1.

Remark 2.3.1. A variety of stopping rules have been considered. For instance, the relative change in u (or u and θ) being below some threshold. As an alternative, the decrease in the two terms (a) and (b) in (2.6) can be monitored to direct stopping. We note again that the u update in step (i) can be implemented using conjugate gradient together with a stopping criteria given by Morozov's discrepancy principle.

IAS Laplace Approximation

Here we show that the IAS iterates can be used to obtain a Laplace approximation to the posterior. Recall that the Laplace approximation $q_{\text{LP}}(z) = (z_{\text{LP}}, C_{\text{LP}})$ to the posterior $p(z \,|\, y)$ is the Gaussian distribution whose mean z_{LP} is the MAP estimator and whose precision C_{LP}^{-1} is the Hessian of the objective $\mathsf{J}(z)$ evaluated at z_{LP}, that is,

$$z_{\text{LP}} = \arg\min_{z} \mathsf{J}(z), \qquad C_{\text{LP}}^{-1} = \nabla\nabla \mathsf{J}(z_{\text{LP}}). \tag{2.9}$$

Thus, the sequence $z^k = (u^k, \theta^k)$ can be used to approximate $q_{\text{LP}}(z)$ by the Gaussian $q_{\text{LP}}^k(z) = (z_{\text{LP}}^k, C_{\text{LP}}^k)$, where

$$z_{\text{LP}}^k = (u^k, \theta^k), \qquad C_{\text{LP}}^k = \Big(\nabla\nabla \mathsf{J}(z_{\text{LP}}^k)\Big)^{-1}. \tag{2.10}$$

Partitioning the Hessian $H(z) = \nabla\nabla \mathsf{J}(z)$ into four blocks of size $d \times d$ gives

$$H(z) = \begin{bmatrix} H_{uu}(z) & H_{u\theta}(z) \\ H_{\theta u}(z) & H_{\theta\theta}(z) \end{bmatrix},$$

with

$$\begin{aligned} H_{uu}(z) &= A^\top \Gamma^{-1} A + \operatorname{diag}(1/\theta), \\ H_{u\theta}(z) &= -\operatorname{diag}(u/\theta^2), \\ H_{\theta\theta}(z) &= \operatorname{diag}(u^2/\theta^3 + \tilde{\beta}/\theta^2), \end{aligned}$$

where multiplication and division operations are defined in element-wise. This explicit characterization of the Hessian, together with Algorithm 2.3.1 and (2.10) yield an iterative Laplace approximation method.

2.3.2 Convergence of IAS and Laplace Approximation

The following result was proved in [Calvetti et al., 2015].

Proposition 2.3.2. *For $\beta > 3/2$ and $\alpha_i = \alpha > 0$ for all $1 \leq i \leq d$, the energy functional* (2.6) *defined over $\mathbb{R}^d \times \mathbb{R}^d_+$ is strictly convex, thus has a unique global minimizer $z^* = (u^*, \theta^*)$. The IAS algorithm produces a sequence $z^k = (u^k, \theta^k)$ that converges to the global minimizer.*

The convergence analysis of IAS was further developed in [Calvetti et al., 2019b], where rates of convergence were established. As a consequence of Proposition 2.3.2 we have the following corollary:

Corollary 2.3.3. *For $\beta > 3/2$ the IAS Laplace approximation $q^k_{\text{LP}}(z) = (z^k_{\text{LP}}, C^k_{\text{LP}})$ given by* (2.10) *converges weakly to the Laplace approximation $q_{\text{LP}}(z) = (z_{\text{LP}}, C_{\text{LP}})$ given by* (2.9).

Proof. Weak convergence of Gaussians is equivalent to convergence of their means and covariances [Bogachev, 1998]. The result follows from Proposition 2.3.2 and continuity of the Hessian. □

2.4 Variational Inference

In this section, we introduce our variational approach for posterior approximation. We provide the necessary background on variational inference in Subsection 2.4.1. The main algorithm is described in Subsection 2.4.2, and convergence guarantees are discussed in Subsection 2.4.3. Our presentation is parallel to that of the previous section.

2.4.1 Background and Mean-field Assumption

Variational inference is a popular technique [Bishop, 2006, Jordan et al., 1999, Wainwright and Jordan, 2008, Blei et al., 2017, Garcia Trillos and Sanz-Alonso, 2020] for approximating the posterior distribution $p(z \,|\, y)$ of some unknown parameter z given data y. We will be concerned with approximating the posterior $p(z \mid y)$ given by (2.5) with $z = (u, \theta)$. The goal of variational inference is to find an approximating distribution $q^*(z)$ which is close to the posterior, but tractable. Then, approximate Bayesian inference can be performed using $q^*(z)$ rather than $p(z \mid y)$. The approximating distribution $q^*(z)$ is defined as the (numerical) solution to an optimization problem. Precisely, one specifies a family $\mathcal{D}$ of tractable distributions and sets

$$q^*(z) := \arg\min_{q \in \mathcal{D}} d_{\mathrm{KL}}\big(q(z) \| p(z \mid y)\big). \tag{2.11}$$

The above minimization can be reformulated as maximizing the *evidence lower-bound* (ELBO) given by:

$$\begin{aligned} \textsc{elbo}(q) &:= \mathbb{E}_q\left[\log p(z, y)\right] - \mathbb{E}_q\left[\log q(z)\right] && (2.12) \\ &= \log p(y) - d_{\mathrm{KL}}\big(q(z) \| p(z \mid y)\big) && (2.13) \\ &= \mathbb{E}_q\left[\log p(y \mid z)\right] - d_{\mathrm{KL}}\big(q(z) \| p(z)\big). && (2.14) \end{aligned}$$

Note from Equation (2.12) that $\textsc{elbo}(q)$ can be evaluated without computing the evidence $p(y)$, which is often intractable. Since the Kullback-Leibler divergence is non-negative, Equation (2.13) shows that $\textsc{elbo}(q)$ indeed provides a lower-bound on the log-evidence. This property can be used for model selection, since larger ELBO indicates a higher probability of the data being generated by a particular model. Finally, Equation (2.14) shows that the optimal $q^*(z)$ finds a compromise between maximizing the expected log-likelihood and minimizing the Kullback-Leibler divergence to the prior $p(z)$.

For reasons discussed below, we will choose the variational family to be

$$\mathcal{D} := \Big\{ q(z) : q(u, \theta) = q(u) q(\theta), \quad q(\theta) = \prod_{i=1}^{d} q(\theta_i) \Big\}. \tag{2.15}$$

This *mean-field* family is a popular choice in variational inference because it enables efficient numerical optimization of the ELBO using the coordinate ascent variational inference (CAVI) algorithm [Bishop, 2006]. Note, however, that under the mean-field approximation the variational distribution is unable to capture the dependence structure between u and θ. This is not an assumption on the data model, but rather is implied by the choice of the variational family $\mathcal{D}$. More flexible variational families and methods may be considered, including structured variational inference [Saul and Jordan, 1995, Barber and Wiegerinck, 1998]. However, the resulting optimization problems do not admit the closed form updates of our proposed approach, and the loss in computational efficiency may not be justified in view of the successful reconstructions achieved in Section 2.5. Another potential weakness of variational inference is that it may underestimate the marginal posterior variance. In fact, our results in Section 2.5 indicate a slight underestimation, which may be potentially alleviated using the ideas in [Giordano et al., 2015]. In the next subsection, we derive a CAVI algorithm for the hierarchical Bayesian model (2.4). We shall see that this variational algorithm shares the ease of implementation of the IAS algorithm.

2.4.2 The Variational Iterative Alternating Scheme (VIAS)

The variational distribution $q^*(z)$ is, by definition, the closest distribution in $\mathcal{D}$ to the posterior $p(z \,|\, y)$, where closeness is quantified using the Kullback-Leibler divergence. Equivalently, $q^*(z)$ is the distribution that maximizes the ELBO in $\mathcal{D}$. Here we propose and analyze a variational iterative alternating scheme (VIAS) to maximize the ELBO, consisting of the following two separate maximization steps:

1. **Initialize** $q^0(u)$, $k = 0$.

2. **Iterate until convergence:**

   (i) Update $q^{k+1}(\theta) = \arg\max_{q(\theta)} \text{ELBO}\big(q^k(u)q(\theta)\big)$.

   (ii) Update $q^{k+1}(u) = \arg\max_{q(u)} \text{ELBO}\big(q(u)q^{k+1}(\theta)\big)$.

(iii) $k \to k+1$.

Note that the structure of VIAS is identical to that of IAS, replacing the energy $\mathsf{J}(z)$ over unknown u and parameters θ with the energy $\textsc{elbo}(q(z))$ over their joint distribution. Let $q^k(z) := q^k(u)q^k(\theta)$. By construction, $\textsc{elbo}(q^k(z))$ is monotonically increasing with k. In other words, the Kullback-Leibler divergence between $q^k(z)$ and the posterior decreases monotonically.

VIAS also shares with IAS its ease of implementation. The following well-known result [Bishop, 2006] gives a characterization for the maximizing distributions in steps (i) and (ii).

Proposition 2.4.1 (Optimization of ELBO in Mean-field Variational Inference)**.** *It holds that*

$$\arg\max_{q(u)} \mathit{ELBO}\big(q(u)q(\theta)\big) \propto \exp\Big(\mathbb{E}_{q(\theta)}\left[\log p(y,z)\right]\Big), \tag{2.16}$$

$$\arg\max_{q(\theta)} \mathit{ELBO}\big(q(u)q(\theta)\big) \propto \exp\Big(\mathbb{E}_{q(u)}\left[\log p(y,z)\right]\Big). \tag{2.17}$$

We next describe how these characterizations, which are a consequence of the mean-field assumption, imply that the maximizing distributions $q(u)$ and $q(\theta)$ in steps (i) and (ii) belong to certain parametric families. Precisely, we shall see in Subsection 2.4.2 that (2.16) implies that $q(u) = \mathcal{N}(m, C)$ and in Subsection 2.4.2 that (2.17) implies that $q(\theta_i) = \text{GIG}(b, r_i, s)$, where GIG denotes the generalized inverse Gaussian distribution. These considerations will reduce the implementation of steps (i) and (ii) to an explicit recursion in the variational parameters.

Before delving into the derivations, we recall for convenience, and later reference, that a random variable $\theta_i \sim \text{GIG}(b, r_i, s)$ has probability density function

$$q(\theta_i \,|\, b, r_i, s) = \frac{(b/r_i)^{s/2}}{2K_s(\sqrt{r_i b_i})}\theta_i^{s-1} e^{-(b\theta_i + r_i/\theta_i)/2}, \tag{2.18}$$

where K_s denotes the modified Bessel function of the second kind. Moreover, the following identities hold

$$\begin{aligned}\mathbb{E}_{q(\theta)}\left[\theta_i\right] &= \frac{K_{s+1}(\sqrt{r_i b})}{K_s(\sqrt{r_i b})} \cdot \sqrt{r_i/b}\,, \\ \mathbb{V}_{q(\theta)}[\theta_i] &= \frac{K_{s+2}(\sqrt{r_i b})}{K_s(\sqrt{r_i b})} \cdot (r_i/b) - (\mathbb{E}_{q(\theta)}\left[\theta_i\right])^2, \\ \mathbb{E}_{q(\theta)}\left[1/\theta_i\right] &= \frac{K_{s-1}(\sqrt{r_i b})}{K_s(\sqrt{r_i b})} \cdot \sqrt{b/r_i}\ ,\end{aligned} \tag{2.19}$$

where $\mathbb{V}$ denotes the variance. The first and second identities can be used to compute credible intervals with the variational distribution, while the third identity will be used to derive the update for $q(\theta)$. For further properties of the generalized inverse Gaussian distribution, we refer to [Lemonte and Cordeiro, 2011].

Updating $q(u)$

To derive the update for $q(u)$, we use (2.16). Note that

$$\begin{aligned}\log q(u) &\propto \mathbb{E}_{q(\theta)}\left[\log p(y,u,\theta)\right] \\ &\propto \mathbb{E}_{q(\theta)}\left[\log p(y \mid u) + \log p(u \mid \theta) + \log p(\theta)\right] \\ &\propto \mathbb{E}_{q(\theta)}\left[-\frac{1}{2}\|Au-y\|_\Gamma^2 - \frac{1}{2}\sum_i \frac{u_i^2}{\theta_i}\right] \\ &\propto -\frac{1}{2}\|Au-y\|_\Gamma^2 - \frac{1}{2}u^\top L u,\end{aligned}$$

where $L = \operatorname{diag}\Big(\mathbb{E}_{q(\theta)}\left[1/\theta\right]\Big)$. This implies that $q(u)$ is Gaussian with mean m and covariance C given by

$$\begin{aligned}m &= (A^\top \Gamma^{-1} A + L)^{-1} A^\top \Gamma^{-1} y, \\ C &= (A^\top \Gamma^{-1} A + L)^{-1}.\end{aligned}$$

The expectations $\mathbb{E}_{q(\theta)}\left[1/\theta_i\right]$ in the diagonal of L can be obtained analytically using the fact (derived in the next subsection) that $q(\theta_i) = \mathrm{GIG}(b, r_i, s)$, together with the third identity in (2.19). Similar to IAS, the sparsity of the VIAS estimate of the parameter u is controlled by the regularization coefficient matrix L. The larger the diagonal component of L is, the smaller the corresponding component of m will be.

Updating $q(\theta)$

To update $q(\theta) = \prod_{i=1}^{d} q(\theta_i)$, we use independence and (2.17) to obtain updates for each $q(\theta_i)$. Note that

$$\begin{aligned}
\log q(\theta_i) &\propto \mathbb{E}_{q(u)}\left[\log p(y \,|\, u) + \log p(u \,|\, \theta) + \log p(\theta)\right] \\
&\propto \log p(\theta_i) + \mathbb{E}_{q(u)}\left[-\log\sqrt{\theta_i} - \frac{u_i^2}{2\theta_i}\right] \\
&\propto \left(\alpha - \frac{3}{2}\right)\log\theta_i - \beta\theta_i - \frac{1}{2\theta_i}\left(m_i^2 + C_{ii}\right),
\end{aligned}$$

where $\propto$ denotes equality up to an additive constant which does not depend on θ_i. In the last displayed line, we used that $\mathbb{E}_{q(u)}\left[u_i^2\right] = m_i^2 + C_{ii}$ for $q(u) = (m, C)$. Recalling (2.18), this implies that $q(\theta_i) = \mathrm{GIG}(b, r_i, s)$ where $b = 2\beta$, $s = \alpha - 0.5$ and $r_i = m_i^2 + C_{ii}$.

Together, the update rule for $q(u)$ given $q(\theta)$, and $q(\theta)$ given $q(u)$, specify the VIAS. A pseudo-code for VIAS is given in Algorithm 2.4.1.

Remark 2.4.2. Using the Woodbury matrix identity, the Kalman-type expression in (2.8) can also be used to obtain VIAS updates for m and C without computing high dimensional matrix inversions

$$\begin{aligned}
m^{k+1} &= Gy, \qquad G := L^{-1}A^\top(AL^{-1}A^\top + \Gamma)^{-1}, \\
C^{k+1} &= (I - GA)L^{-1}.
\end{aligned}$$

Algorithm 2.4.1 Variational Iterative Alternating Scheme (VIAS)

```
Input: Data $y$, matrix $A$. Prior hyperparameters: $\alpha, \beta$.
Initialize: $m^0, C^0, k = 0$. Set $b = 2\beta$, $s = \alpha - 0.5$.
For $k = 0, 1, \dots$ until convergence do:
    (i) Update $r_i^{k+1} = (m_i^k)^2 + C_{ii}^k$ for each $i = 1, \dots, d$.
    (ii) Set
        $L = \mathrm{diag}(\ell), \qquad \ell_i = \frac{K_{s-1}\left(\sqrt{r_i^{k+1} b}\right)}{K_s\left(\sqrt{r_i^{k+1} b}\right)} \cdot \sqrt{\frac{b}{r_i^{k+1}}}, \qquad (2.20)$
        and update
        $m^{k+1} = (A^\top \Gamma^{-1} A + L)^{-1} A^\top \Gamma^{-1} y,$
        $C^{k+1} = (A^\top \Gamma^{-1} A + L)^{-1}.$
end for
Output: Variational approximation $p(z \mid y) \approx q^{k+1}(z) = q^{k+1}(u) q^{k+1}(\theta)$,
    where
        $q^{k+1}(u) = (m^{k+1}, C^{k+1}), \quad q^{k+1}(\theta_i) = \mathrm{GIG}(b, r_i^{k+1}, s).$
```

The update for m could also be implemented using conjugate gradient for least squares and an early stopping condition.

Remark 2.4.3. The ELBO can be computed at each iteration, and the relative change in the ELBO can be used as a stopping criteria, since this algorithm maximizes the ELBO. The relative change in the variational parameters along VIAS iterates could also be monitored to determine stopping. $\square$

Variational Parameters and VIAS

On deriving the CAVI updates, we obtain that $q(u) = (m, C)$ and $q(\theta_i) = \mathrm{GIG}(b, r_i, s)$, for $i = 1, \ldots, d$. The parameters $m, C, b, \{r_i\}_{i=1}^{d}$, and s are known as the variational parameters.

The parameters m and C are of the greatest interest to us, since they will determine the prediction of the unknown quantity of interest u. While m gives the approximate posterior mean, C will allow us to obtain credible intervals on the prediction, and to understand the correlation between different components of u.

In the distribution of θ, the values of the parameters b and s are directly related to the hyperparameters describing the prior gamma distribution: $b = 2\beta, s = \alpha - 0.5$. Thus, the choice of hyperparameters directly affects the variational posterior. Note that each diagonal component of the matrix L in (2.20) will be large if b or β are large. On the other hand, each diagonal element is a decreasing function in $s \in (-0.5, 0.5)$ and therefore in $\alpha \in (0, 1)$. As a consequence, if one expects sparse structure in the true parameter u, choosing a small α value with a moderately large β value would lead to adequate shrinkage. Each diagonal component of the matrix L diverges to infinity as β increases. So, to avoid shrinking each component of the parameter estimate too close to zero, one should not use extremely large β value. The mean of the prior α/β should be chosen to be close to the expected size of the unknown θ, if prior information on this size is available. In addition to these heuristics, we will illustrate in Section 2.5 how the hyperparameters can be learned by a simple model selection procedure. Our numerical experiments show that the reconstructions are not sensitive to perturbation of the model hyperparameters, but that obtaining appropriate ballpark values for the hyperparameters through model selection can substantially improve the reconstruction.

All the information about the variational distribution is stored in the five parameters; two of them are fixed, and the other three are interdependent. The CAVI algorithm updates these three parameters iteratively as follows:

- Keeping m and C constant, update each r_i with the formula:

$$r_i = m_i^2 + C_{ii}.$$

- Keeping r constant, update m and C:

$$\begin{aligned} C &= (A^\top \Gamma^{-1} A + L)^{-1} \text{ with } L_{ii} = \mathbb{E}\Big[\frac{1}{\theta_i}\Big], \\ m &= (A^\top \Gamma^{-1} A + L)^{-1} A^\top \Gamma^{-1} y. \end{aligned}$$

2.4.3 Convergence and Initialization of VIAS

The ELBO in (2.16) and (2.17) can now be written as a function of the variational parameters rather than the variational distribution, i.e., $\text{ELBO}\big(q(u)q(\theta)\big) = \text{ELBO}(m, C, r)$. Then, the CAVI updates can be rewritten

$$\begin{aligned} \arg\max_{q(\theta)} \text{ELBO}\big(q(u)q(\theta)\big) &= \arg\max_{r} \text{ELBO}\big(m, C, r\big), \\ \arg\max_{q(u)} \text{ELBO}\big(q(u)q(\theta)\big) &= \arg\max_{m,C} \text{ELBO}\big(m, C, r\big). \end{aligned}$$

As the parameters updated through VIAS are m, C and r, we will ignore terms in the ELBO that do not depend on them. We will still denote the remaining expression as $\text{ELBO}(m, C, r)$.

A straightforward calculation shows that

$$\mathrm{ELBO}(m,C,r) = -\frac{1}{2}\Big(\mathrm{tr}(\Gamma^{-\frac{1}{2}}AC(\Gamma^{-\frac{1}{2}}A)^\top) + \|Am-y\|_\Gamma^2 - \log \det C\Big) \\ + \sum_{i=1}^{d}\left(\log 2K_s\left(\sqrt{r_i b}\right) - \frac{s}{2}\log\frac{b}{r_i}\right).$$

The following result, which follows from Theorem 2.2 of [Bezdek et al., 1987], shows local convergence of VIAS.

Proposition 2.4.4. *Suppose that* $\mathrm{ELBO}(m,C,r)$ *has a local maximum at* (m^*,C^*,r^*) *and that the Hessian of* $\mathrm{ELBO}(m,C,r)$ *is negative definite at* (m^*,C^*,r^*)*. Then there is a neighborhood* $\mathcal{U}$ *of* (m^*,C^*,r^*) *such that, for any initialization* $(m^0,C^0,r^0)\in\mathcal{U}$*, VIAS converges to* (m^*,C^*,r^*)*.*

Unfortunately, the convergence to the global maximum of $\mathrm{ELBO}(m,C,r)$ is not guaranteed. This is because the function $\mathrm{ELBO}(m,C,r)$ can have multiple local maxima. We illustrate this phenomenon in the univariate case $u\in\mathbb{R}$, where $y = Au+\eta$, $\eta\sim(0,I_n)$. Denoting $C = c\in(0,\infty)$ and setting $b=1$, we then have

$$\mathrm{ELBO}(m,c,r) = -\frac{A^\top A}{2}(m^2+c) + m^\top A^\top y + \frac{1}{2}\log c + \frac{s}{2}\log r + \log\left(2K_s(\sqrt{r})\right). \tag{2.21}$$

VIAS maximizes (2.21) along the manifold $\mathcal{M}$ given by

$$\mathcal{M} = \{(m,c,r)|\; m = cA^\top y,\; r = (A^\top y)^2c^2 + c\}.$$

Denoting $y_A := A^\top y$, the expression (2.21) on this manifold becomes

$$-\frac{A^\top A}{2}(c + y_A^2c^2) + y_A^2 c + \frac{1}{2}\log c + \frac{s}{2}\log\left(y_A^2c^2+c\right) + \log\left(2K_s\left(\sqrt{y_A^2c^2+c}\right)\right). \tag{2.22}$$

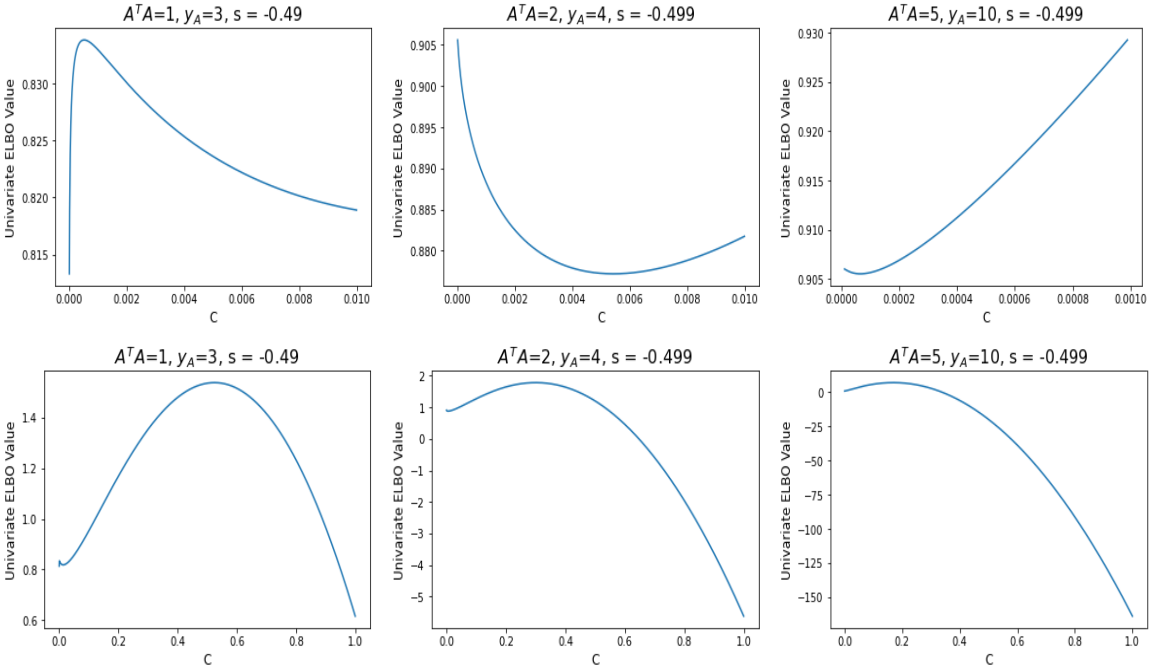


Figure 2.1: Top row: plots of (2.22) near zero. Bottom row: plots of (2.22) for [0,1].

To gain a better understanding of the $\text{ELBO}(m, c, r)$ on the manifold $\mathcal{M}$, we provide plots of (2.22) with $n = 50$ for the following three cases: 1) $A^\top A = 1, y_A = 3, s = -0.49$; 2) $A^\top A = 2, y_A = 4, s = -0.499$; and 3) $A^\top A = 5, y_A = 10, s = -0.499$. Figure 2.1 shows that there are multiple local maxima, which suggests that the initialization of VIAS can have an effect on the final point of convergence. The plots in Figure 2.1 indicate that, in each case, the global maximizer is the local maximizer farthest away from zero. For this reason, we recommend initializing VIAS with a covariance $C^0 \succ \lambda I_d$, with $\lambda > 0$ far away from zero. In addition, we expect $|y_A|$ to be large when the noise level is high. In such a case the global maximum of (2.22) will be far from the origin, as seen in Figure 2.2, which further justifies the suggested initialization. From the perspective of quantifying uncertainties in u, initializing at a large covariance ensures convergence of the VIAS iterates C^k to a matrix C that gives conservative credible intervals for the reconstruction.

In Section 2.5, we introduce hyperparameter tuning based on the ELBO values. Typically, the calibrated s values were near -0.5. Accordingly, we have characterized a region

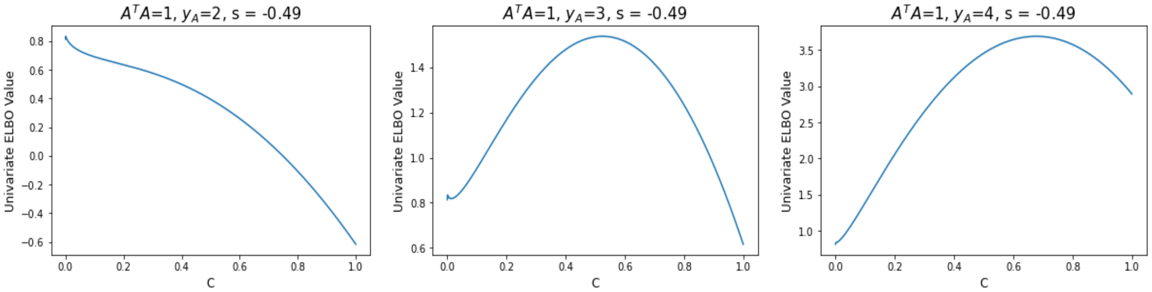


Figure 2.2: Plots of (2.22) for $y_A = 2, 3, 4$ with $A^\top A = 1, s = -0.49$.

of $(|y_A|, A^\top A)$ values where the ELBO has more than one local maximum for $s = -0.499$. We used a grid-search to find a local maximum with a mesh step size of 10^{-7} ranging from zero to one. Multiple local maxima occurred in the beige-colored region in Figure 2.3. In the case where multiple local maxima exist, we observed that the global maximum is the farthest away from zero. For smaller s values, i.e., closer to -0.5, which would promote sparse structure in the VIAS estimate, we observed a similar pattern to the one in the left plot of Figure 2.3.

To assess if data realizations that give multiple local maxima are likely to occur, we ran an empirical study to estimate $\mathbb{P}(|y_A| \geq k)$ for $k \in \{0, \ldots, 10\}$. To do so, we first sampled θ from the gamma distribution with shape parameter 0.001 and rate parameter 1. In addition, each component of the vector A satisfying the prespecified $A^\top A$ value was obtained from a uniform distribution. Next we generated a scalar u from the Gaussian distribution with mean zero and variance θ. Then we randomly sampled $y = Au + \eta$, where $\eta \sim N(0, I_n)$ for 10^4 times and obtained the proportion of times when the event $\{|y_A| \geq k\}$ occurs. From Figure 2.3, we can observe that such an event can occur with a zero probability.

In all our experiments, the global maximum of the ELBO was the local maximum farthest away from zero. Based on such experiments and on the computed examples in the next section, we believe VIAS is very likely to converge to the global maximum of the ELBO in most practical applications, as long as it is initialized as suggested above.

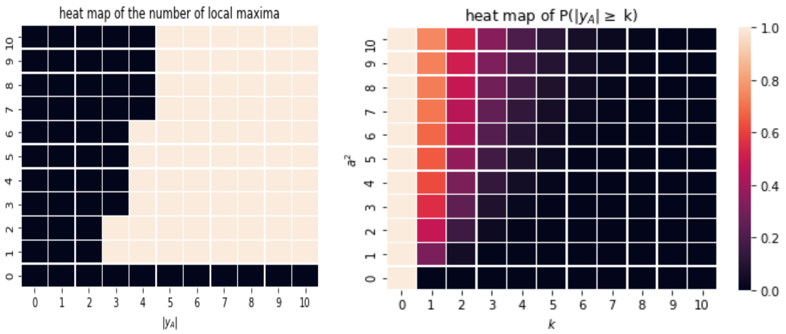


Figure 2.3: Common parameter: $s = -0.499$. Left: heat map of the number of local maxima of the ELBO (black: one maximum, beige: more than one maxima). Right: heat map of the approximated $\mathbb{P}(|y_A| \geq k)$. Together, the two plots show that it is unlikely to observe data that gives an ELBO with more than two local maxima.

2.5 Computed Examples

In this section, we report the performance of VIAS in four computed examples, assessing its accuracy and its ability to provide meaningful uncertainty quantification. We also explore how to exploit the variational inference framework to guide the choice of model hyperparameters.

2.5.1 Truth and Data from Hierarchical Model

We first apply VIAS to data generated from the hierarchical model (2.4). This serves to illustrate the role of the hyperparameters in the hierarchical model, and also the application of our proposed variational inference technique. We compare the accuracy of point estimates constructed with VIAS and IAS, as well as the uncertainty quantification given by VIAS and the iterative Laplace approximation in Section 2.3. Finally, we show how the ELBO can be used to select the model hyperparameters, and we demonstrate that the accuracy of the reconstruction obtained with this model selection approach is comparable to the accuracy achieved using the true hyperparameters.

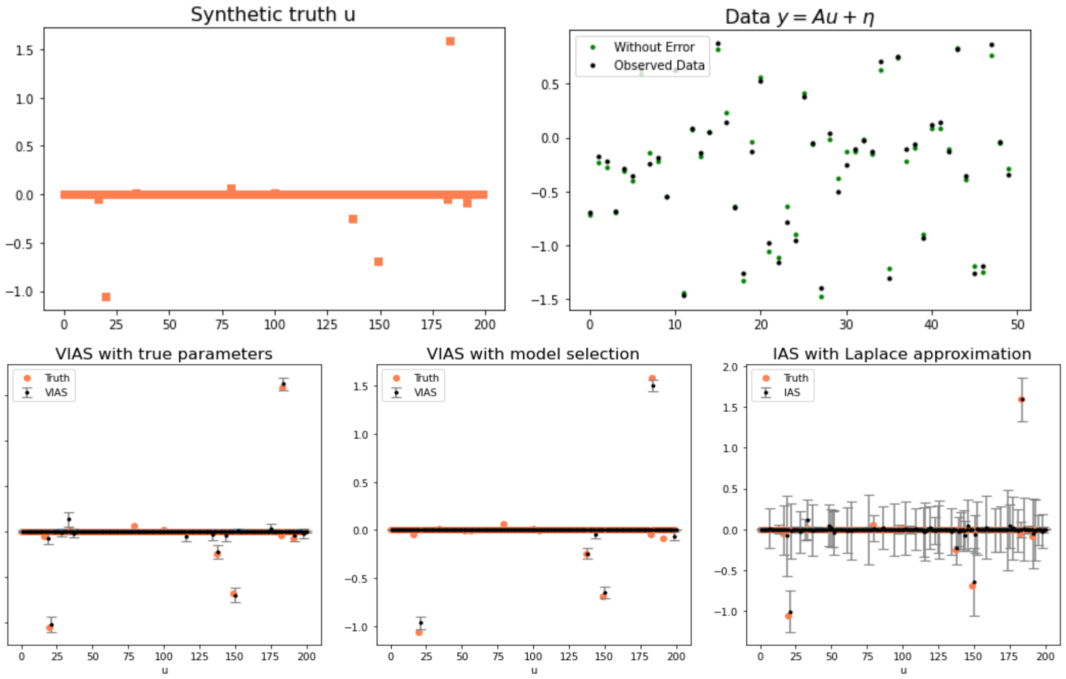


Figure 2.4: First row: synthetic truth and data (Subsection 2.5.1). Second row: computed results with VIAS and IAS. The VIAS 95% credible intervals are shorter while providing suitable coverage.

Setting

We sample θ_i values from a gamma distribution with $\alpha = 0.005\,,\ \beta = 0.05$ (mean 0.1 and variance 2). Conditional on these θ_i values, we generate the synthetic truth $u \in \mathbb{R}^d$ by sampling independently $u_i \sim (0, \theta_i)$. The data $y \in \mathbb{R}^n$ is generated by $y = Au + \eta$, where $A \in \mathbb{R}^{n \times d}$ is randomly generated with each entry being uniformly distributed between 0 and 1. We choose $d = 200$ and $n = 50$ so that the problem is severely underdetermined. The error term η is sampled from a normal distribution $(0, \gamma^2 I_{50})$, where γ is chosen to be 5% of the max-norm of Au. Figure 2.4 shows the generated synthetic truth u and data y. The generated $u \in \mathbb{R}^{200}$ is very sparse with only 4 distinctly large components of varying sizes.

Numerical Results with True Hyperparameters

Here we report numerical results for VIAS and IAS. For VIAS, we set hyperparameters α and β to be exactly those used to generate the data, namely $\alpha = 0.005\,,\ \beta = 0.05$. This determines the choice of variational parameters $s = -0.495$ and $b = 0.1$. For IAS, as we expect sparse structure in the parameter of interest on a unit scale, we set the parameter $\alpha = 1$ with $\tilde{\beta} = 0.00001$. Figure 2.4 displays the results for VIAS and IAS. For the initializations, we set both θ^0, $m^0 \in \mathbb{R}^{200}$ to be the all-ones vector, and $C^0 \in \mathbb{R}^{200\times 200}$ to be the identity matrix. Both algorithms yielded successful reconstruction of u. The credible intervals for the VIAS estimates are significantly shorter than the intervals from the IAS using Laplace approximation. The implication is that IAS may not give sufficient shrinkage off the support, while VIAS may underestimate the uncertainty in the reconstruction. Reducing the total number of VIAS iterations alleviates the underestimation of uncertainty but leads to less accurate reconstructions. To quantify the accuracy of these credible intervals, we conduct repeated simulations fixing the matrix A and synthetic truth u, while resampling the error term η to generate different y values. On conducting 1000 such simulations and generating 200 credible intervals for each component of u, we observe that the 95% credible interval for the components of u covers the true values 96.06% of the time for VIAS with the true hyperparameters, and 98.67% for IAS. Thus, VIAS maintains similar accuracy to IAS with much narrower credible intervals.

The left and the middle plots in Figure 2.5 show the convergence of VIAS using the true hyperparameters. The ELBO value stabilizes after 100-200 iterations. We also illustrate the decay of the relative change in max-norm of the variational parameters along the VIAS iterations, which can be seen in the middle plot. The number of iterations that IAS needs to stabilize is significantly lower, of the order of 10.

Numerical Results with Model Selection

In this subsection, we investigate the learning of the hyperparameters α, β. For this purpose, we use the ELBO as a model selection tool, choosing the hyperparameters which maximize the ELBO. Since the ELBO is a lower bound for the marginal probability of the data, larger ELBO values suggest a better fit to the data. In practice, we obtained ELBO values after 300 iterations of VIAS for each choice of (α, β) values in a two-dimensional grid. The choice of (α, β) value which led to the maximal ELBO value was used as our hyperparameter values.

One would expect the hyperparameters which maximize the ELBO to be close to the true model hyperparameters. However, on conducting the model selection, we find that this is not the case. The hyperparameters found using model selection are $\alpha = 0.001$ and $\beta = 1623$, whose corresponding ELBO value after 1000 iterations was roughly around 3899, a significantly larger value than the ELBO with the true hyperparameters, which is 3243, which can be seen in Figure 2.5. Despite this large difference in the hyperparameters and the ELBO, the resulting reconstructions are similar, as displayed in Figure 2.4. Relative to the results obtained with the true parameters, the ELBO-selected model slightly underestimates the signal due to overshrinkage induced by using large β value, see Figure 2.4. On conducting repeated simulations in the manner mentioned previously, the 95% credible intervals for this ELBO-selected model contain the true values around 91-92% of the time. The model selected by maximizing the ELBO provides credible intervals with good accuracy. If no prior knowledge on hyperparemeters α and β is available, we propose calibrating hyperparameters based on the ELBO as a way to suitably balance between shrinkage and estimation of parameters.

2.5.2 Fixed Sparse Truth

In this second example, we consider a fixed truth which is less sparse than the one in the previous example. Moreover, the truth is chosen rather than sampled generatively. The

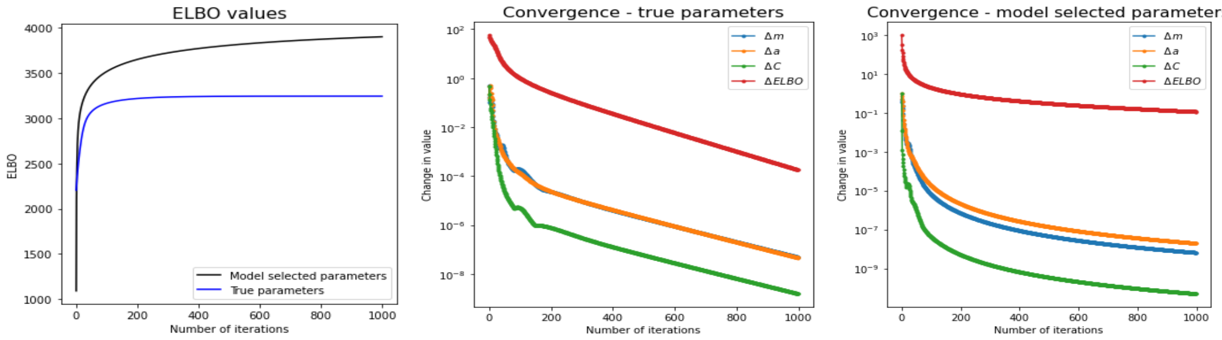


Figure 2.5: Convergence of the ELBO and the variational parameters along VIAS iterates, with truth and data generated from the hierarchical model (Subsection 2.5.1).

model hyperparameters α and β are chosen according to the model selection procedure to maximize the ELBO, which gave $\alpha = 0.0001$ and $\beta = 33.59$.

Setting

We generate a random matrix $A \in \mathbb{R}^{50\times 100}$ with each entry drawn uniformly between 0 and 1. The to-be-reconstructed parameter $u \in \mathbb{R}^{100}$ is chosen so that only 10 components are non-zero, see Figure 2.6. The data y is generated by multiplying A with u and adding a randomly sampled Gaussian with standard deviation taken as 2% of the max-norm of Au. As in the previous example, we set both θ^0, $m^0 \in \mathbb{R}^{100}$ to be the all-ones vector, and $C^0 \in \mathbb{R}^{100\times 100}$ to be the identity matrix.

Numerical Results

Figure 2.6 shows the VIAS results and a comparison to other techniques. The VIAS predictions are close to the true values and even when the prediction was not accurate, credible intervals successfully captured true values. Compared to IAS, VIAS point estimates are much closer to the true values. The IAS reconstruction is less sparse than the VIAS reconstruction and typically underestimates the non-zero components of the signal. We have also obtained LASSO estimates with tuning parameter calibration based on cross validation (CV), Akaike

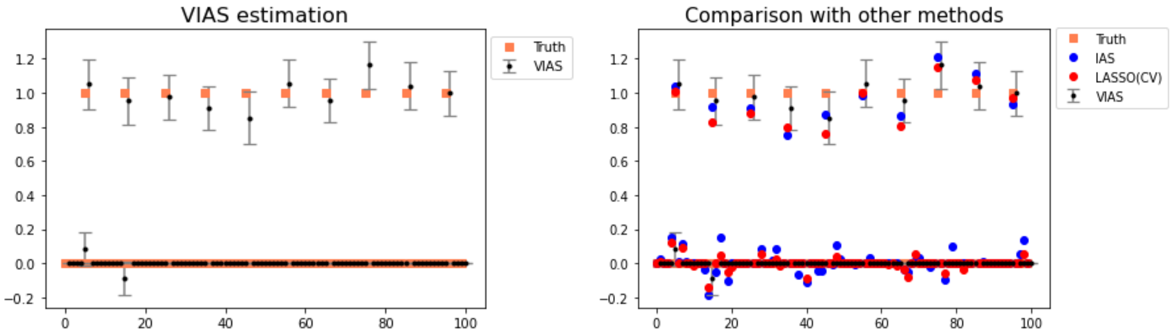


Figure 2.6: Fixed sparse truth (Subsection 2.5.2). VIAS reconstruction and credible intervals (left). Comparison with IAS and LASSO with cross validation (right).

information criterion, and Bayesian information criterion using Python's `LassoLarsIC` and `LassoLarsCV` functions. We only report in Figure 2.6 the result based on CV, which was the most accurate. From Figure 2.6, we can observe that the VIAS estimate was superior to IAS and LASSO in terms of estimating zero components of the parameter, while also maintaining a good accuracy in non-zero components.

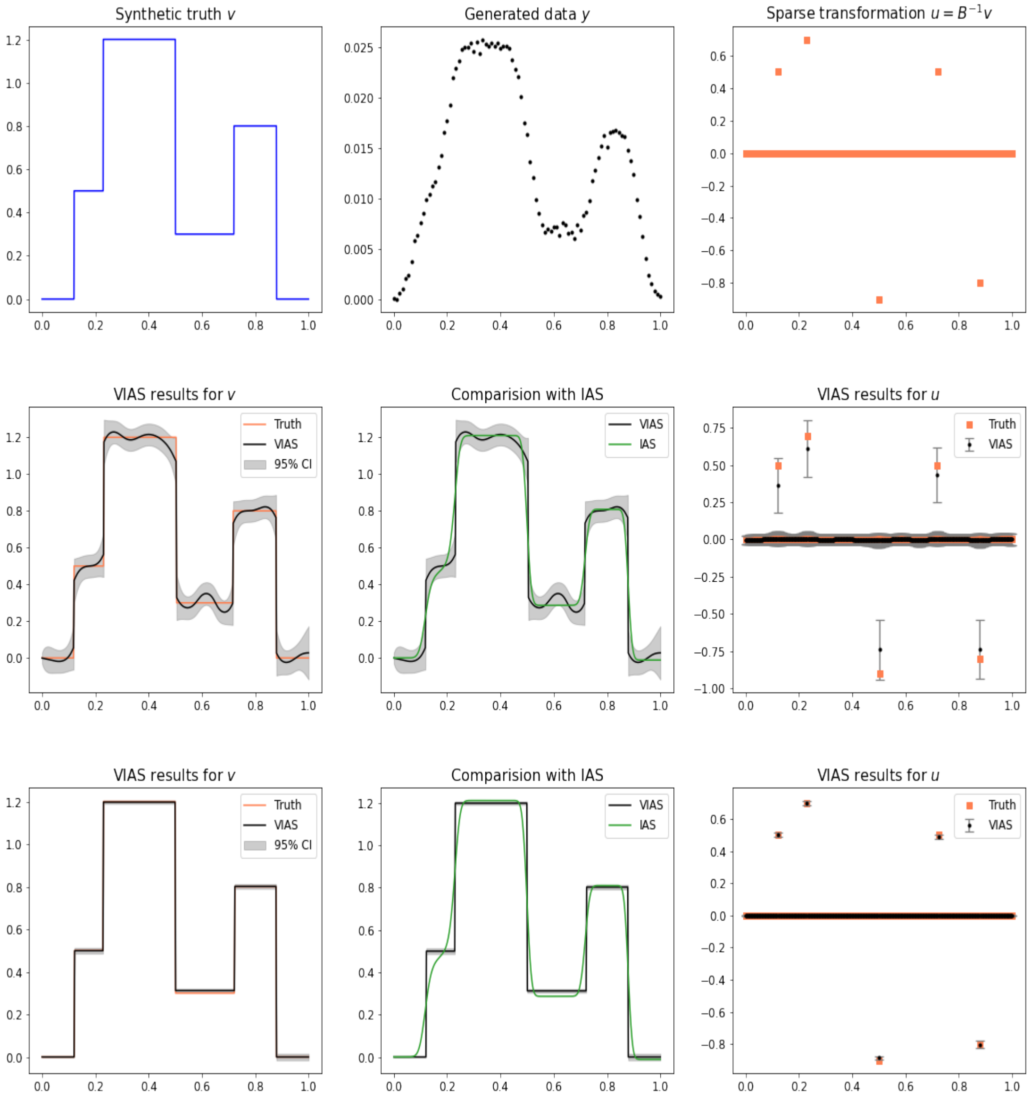
2.5.3 Deconvolution

In this example, we consider the 1D deconvolution problem in [Calvetti et al., 2020b], where the goal is to reconstruct a piecewise constant signal convolved with an Airy kernel. We compare the results obtained with IAS and VIAS. We demonstrate the high accuracy achieved by VIAS with ELBO-selected model hyperparameters and show that the VIAS signal covariance provides meaningful uncertainty quantification on the reconstruction.

Setting

Let $f : [0, 1] \to \mathbb{R}$ be a piecewise constant function with $f(0) = 0$. The data y is generated by the following convolution:

$$y_j = \int_0^1 A(s_j - t) f(t) dt + \eta_j, \quad 1 \le j \le n, \quad A(t) = \left(\frac{J_1(\kappa|t|)}{\kappa|t|} \right)^2,$$


Synthetic truth v

Generated data y

Sparse transformation $u = B^{-1}v$

VIAS results for v

Comparision with IAS

VIAS results for u

VIAS results for v

Comparision with IAS

VIAS results for u

Figure 2.7: Deconvolution problem (Subsection 2.5.3). First row: truth, data, and sparse representation. Second row: VIAS reconstruction of the signal and its sparse representation with user-chosen model hyperparameters $\alpha = 0.12$ and $\beta = 50$. Third row: same as second row, but with ELBO-selected model.

where J_1 is the Bessel function of the first kind, κ is a scalar controlling the width of the kernel that we set to $\kappa = 40$, and $s_j = (4 + j)/100$. The above integral can be discretized, leading to the linear equation

$$y = Av + \eta\,, \quad A_{jk} = w_k A(s_j - t_k)\,, \quad \eta \sim (0, \gamma^2 I_n)\,, \tag{2.23}$$

where $v \in \mathbb{R}^d$ has components $v_k = f(t_k)$ with $t_k = (k-1)/(n-1)$, and the w_k are quadrature weights for discretization of the integral. The standard deviation γ is set to be 1% of the max-norm of the noiseless signal.

The unknown parameter v is not a sparse vector, but can be written in sparse form in a suitable basis. To that end, define $u_j = v_j - v_{j-1}$ with $u_0 = 0$. Since v is piecewise constant, u is sparse. Note that we can write $u = B^{-1}v$, where

$$B^{-1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ -1 & 1 & \dots & 0 \\ & & \ddots & \\ 0 & \dots & -1 & 1 \end{bmatrix} \in \mathbb{R}^{d \times d}.$$

Thus, we can rewrite (2.23) in terms of this sparse unknown vector u as follows:

$$y = ABu + \eta, \quad \eta \sim (0, \gamma^2 I_n). \tag{2.24}$$

Our inverse problem is to estimate the vector u, assumed to be sparse, from the data vector y. Figure 2.7 shows the piecewise constant function v to be reconstructed, its sparse transformation u, and the data y. We take $d = 500$ and $n = 91$ so that the problem is underdetermined.

Reconstruction Accuracy and Model Selection

The results of applying VIAS to this problem are displayed in Figure 2.7. As in the previous example, we set $m^0 \in \mathbb{R}^{500}$ to be the all-ones vector, and $C^0 \in \mathbb{R}^{500\times 500}$ to be the identity matrix. We consider two implementations of VIAS. In the first (Figure 2.7, middle row), we use hyperparameter values $\alpha = 0.12$ and $\beta = 50$. In the second (Figure 2.7, bottom row), we adopt the ELBO approach for model selection, which gives hyperparameter values $\alpha = 0.0001$ and $\beta \approx 7742$. Both VIAS implementations produce sparse solutions and the true signal lies within the obtained credible intervals. The signal reconstructed with VIAS has sharp jumps whereas the IAS reconstruction has much smoother jumps. For instance, the first two jumps are treated as a smooth transition by IAS, while VIAS successfully detects them as separate jumps. While VIAS with hyperparameters $\alpha = 0.12$ and $\beta = 50$ detects the presence of five distinct jumps, uncertainty remains in the location of the jumps. Moreover, the reconstruction exhibits oscillatory artifacts in the constant regions, where IAS remains accurate. The VIAS reconstruction with model selection is significantly more accurate; it detects the five jump locations and it does not show oscillatory artifacts.

Uncertainty Quantification and Covariance Structure

The credible intervals obtained with both VIAS implementations provide additional insight into the nature of uncertainty quantification in this problem. VIAS detects that the main source of uncertainty is the location of the jumps. We observe in Figure 2.7 that the uncertainty in the reconstruction spikes near the jumps, while it remains relatively small around the constant regions. Moreover, the enhanced accuracy of VIAS with model selection is accompanied by narrower confidence intervals. Notice that we have imposed the condition $v_0 = 0$, hence we are certain that the value at 0 is 0, and uncertainty is expected to increase from left to right. This overall trend is also successfully identified by both VIAS implementations.

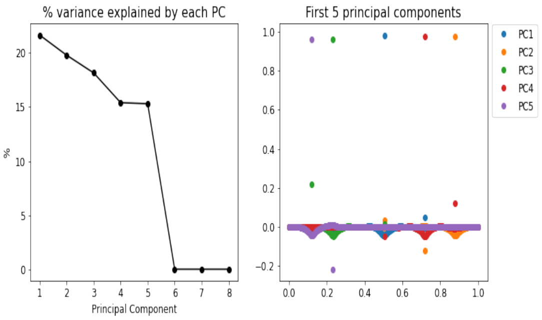


Figure 2.8: Principal component analysis of VIAS uncertainty with hyperparameters $\alpha = 0.12, \beta = 50$.

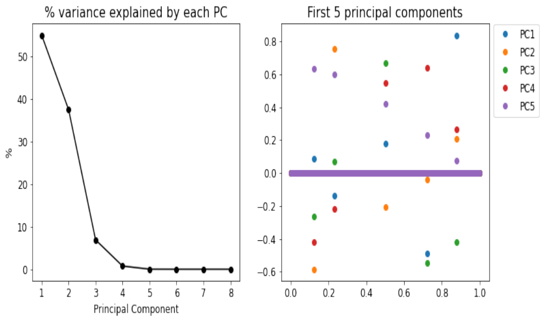


Figure 2.9: Principal component analysis of VIAS uncertainty with ELBO-selected hyperparameters.

An additional benefit of VIAS is that it not only gives an approximation to the component-wise variances on the signal reconstruction, but also an approximate covariance matrix. This matrix contains information on the dependencies in the reconstruction of various components of the signal. To illustrate this point, Figures 2.8 and 2.9 show a principal component analysis of the covariance matrix C obtained with the two VIAS implementations considered above. For VIAS with user-chosen hyperparameters $\alpha = 0.12$ and $\beta = 50$, the first five principal components each explain 15-20% of the variance, and this drops to less than 0.1% after the 5th component. In contrast, the three principal components already explain most of the variance for VIAS with model selection. In both VIAS implementations, each principal component is localized around a jump in the signal. With model selection, the principal components are fully localized at the jumps, reflecting that no uncertainty remains in the location of the jumps. On the other hand, with the first VIAS implementation the localization of principal components at the jumps is not perfect, reflecting that there is non-negligible uncertainty in both the location and magnitude of the jumps. Each principal component also contains components of its nearby jumps, which suggests that nearby jumps may be correlated. For instance, this is noticeable in the first VIAS implementation PC3 and PC5 —corresponding to the first and second jumps— likely because the first two jumps are very close to each other and it is harder to distinguish the two. Overall, the principal components obtained with VIAS successfully identify that most of the variance in the signal reconstruction lies around the jumps, that nearby jumps are correlated, and that uncertainty in the problem propagates from left to right. Finally, this example demonstrates the effect of the hyperparameters in the reconstruction. Poor choice of the hyperparameters gives a less accurate, more uncertain reconstruction.

2.5.4 Learning Dynamics of Lorenz-63 System

In this section, we illustrate the use of IAS and VIAS for sparse identification of dynamical systems. Our problem setting is motivated by [Brunton et al., 2016].

Setting

Consider the Lorenz-63 system [Lorenz, 1963]

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \zeta z, \end{aligned} \tag{2.25}$$

with the classical parameter values $\sigma = 10, \rho = 28, \zeta = 8/3$. Our goal is to recover the right-hand side of (2.25) from time series data. We assume to have observations of a trajectory and its derivative along 2000 equidistant time points in the time-interval $[0, 40]$; thus the time between observations is $\Delta t = 0.02$. Note that in this example y denotes the second component of the dynamics rather than the observed data.

Following [Brunton et al., 2016], we adopt a dictionary-learning strategy and construct, from the given trajectory data, a matrix of the form:

$$A = \begin{bmatrix} | & | & | & | & | & | & | & | & | & & | & \\ \mathbf{x} & \mathbf{y} & \mathbf{z} & \mathbf{x}^2 & \mathbf{y}^2 & \mathbf{z}^2 & \mathbf{xy} & \mathbf{xz} & \mathbf{yz} & \ldots & \mathbf{x}^5 & \ldots \\ | & | & | & | & | & | & | & | & | & & | & \end{bmatrix} \in \mathbb{R}^{2000\times 55}.$$

We then obtain, as in [Brunton et al., 2016], synthetic data on the derivatives by setting

$$\begin{aligned} \dot{\mathbf{x}} &= A\Phi_1 + \eta_1, \quad \Phi_1 = [-10, 10, 0, 0, 0, 0, 0, 0, \ldots, 0]^\top, \\ \dot{\mathbf{y}} &= A\Phi_2 + \eta_2, \quad \Phi_2 = [28, -1, 0, 0, -1, 0, 0, \ldots, 0]^\top, \\ \dot{\mathbf{z}} &= A\Phi_3 + \eta_3, \quad \Phi_3 = [0, 0, -8/3, 0, 0, 0, 1, \ldots, 0]^\top, \end{aligned}$$

where $\eta_i \sim \mathcal{N}(0, 0.3 I_{2000})$ are independent. Our goal is to recover Φ_1, Φ_2, Φ_3 based on A, $\dot{\mathbf{x}}$, $\dot{\mathbf{y}}$, $\dot{\mathbf{z}}$. Note that Φ_1, Φ_2, Φ_3 are sparse. More generally, the data-driven learning of dynamical systems in [Brunton et al., 2016] relies on the underlying assumption that only a few terms of a given dictionary (in our example made of polynomials of degree five) govern the dynamics; sparsity-promoting VIAS is hence a natural algorithm for identification of dynamical systems.

Numerical Results

The recovery of the Lorenz-63 model via VIAS is shown in Figure 2.10. Since we expect sparse structure in the parameter of interest, same as in our first example, we use $s = -0.495$ and $b = 0.1$ for the variational parameters. For the initializations, we set $\theta^0, m^0 \in \mathbb{R}^{55}$ to be the all-ones vector, and $C^0 \in \mathbb{R}^{55\times 55}$ to be the identity matrix. We observe that VIAS accurately recovers the true parameter values. As VIAS can quantify uncertainties of our estimates, we provide the true dynamics in the blue line in Figure 2.11 with the shaded regions determined by dynamics obtained from 2.5 and 97.5 percentile credible levels of the true parameters. We point out that despite the chaotic behavior of the Lorenz-63 system, the uncertainty in the dynamics remains moderate due to the high accuracy of the recovered coefficients. Moreover, we note that the relative larger error in the coefficients of the y-trajectory in Figure 2.10 translates into wider credible intervals for the reconstructed trajectories of y in Figure 2.11. Therefore, VIAS correctly identifies that there is more uncertainty in the reconstruction of the y-component.

Compared to VIAS, IAS showed inferior performance in estimating parameters of the Lorenz-63 model as one can see in Figure 2.10. Furthermore, we provide plots of true dynamics with shaded regions determined by dynamics recovered from 95 percent credible intervals obtained from a Laplace approximation to the posterior. As shown in Section 2.5.1, credible intervals based on Laplace approximation tend to be larger than the ones obtained from VIAS. In the context of the Lorenz-63 model, the uncertainty in the dynamics is amplified by the mismatch between the estimated coefficients and the true coefficients. From Figure 2.11, we can see that quantifying uncertainty of dynamics based on Laplace approximation gives little information as the constructed shaded regions are often too wide, which highlights the strength of VIAS in uncertainty quantification tasks.

2.6 Conclusion and Future Directions

This chapter introduced VIAS, a variational inference computational framework for linear inverse problems with gamma hyperpriors. The proposed VIAS shares the flexibility and ease of implementation of IAS for MAP estimation. We have shown the accuracy of VIAS in several computed examples, and we have explored its potential to provide meaningful uncertainty quantification and perform model selection. There are several research directions that stem from this work:

- We have established a local convergence result for VIAS, but we have not provided an analysis of convergence rates. Moreover, it would be interesting to study the approxi-

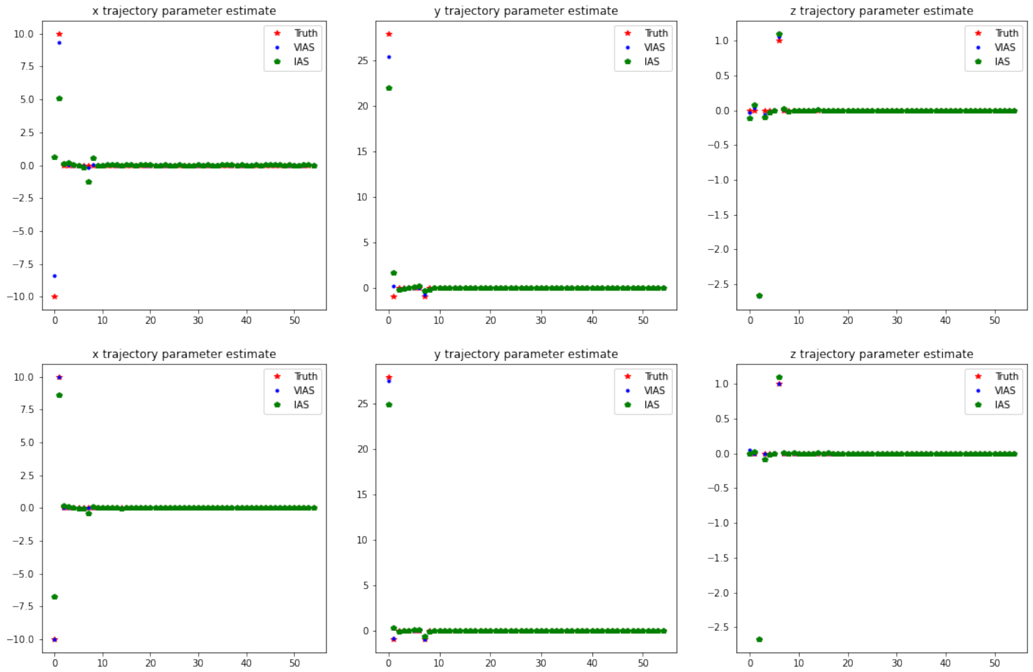


Figure 2.10: Recovery of dictionary coefficients for x-trajectory (first column), y-trajectory (second column), and z-trajectory (third column) using IAS and VIAS. Top: two iterations. Bottom: five iterations.

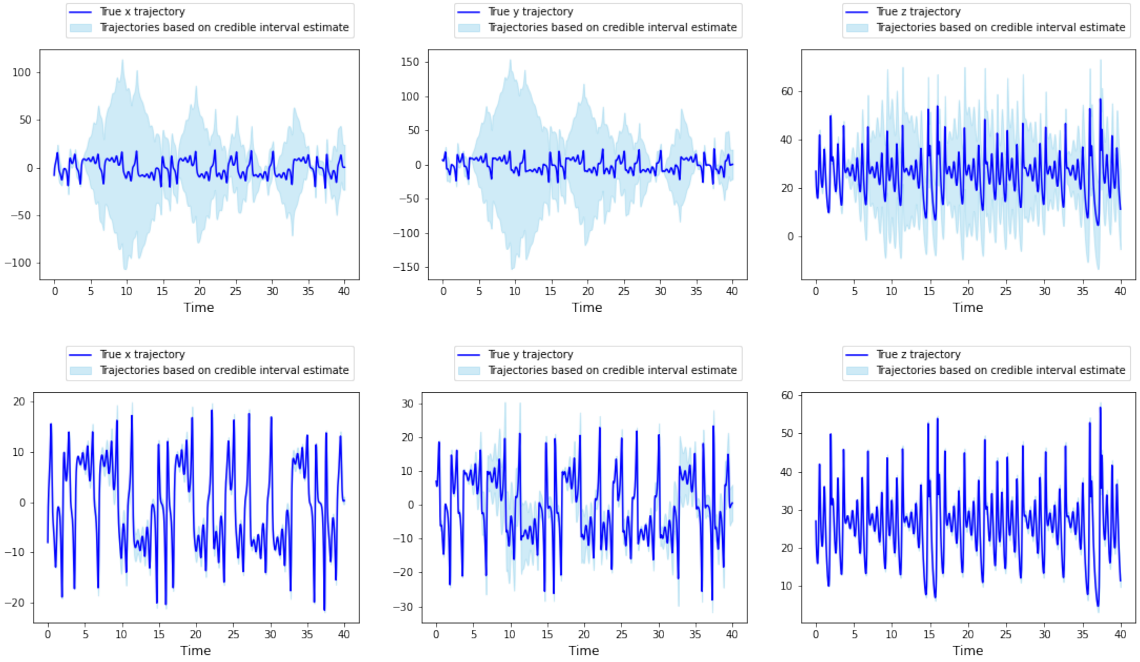


Figure 2.11: True Lorenz-63 trajectory and VIAS estimation. Top: two VIAS iterations. Bottom: five VIAS iterations. Blue line is the true dynamics. Shaded regions are constructed from 2.5 and 97.5 credible levels of coefficients.

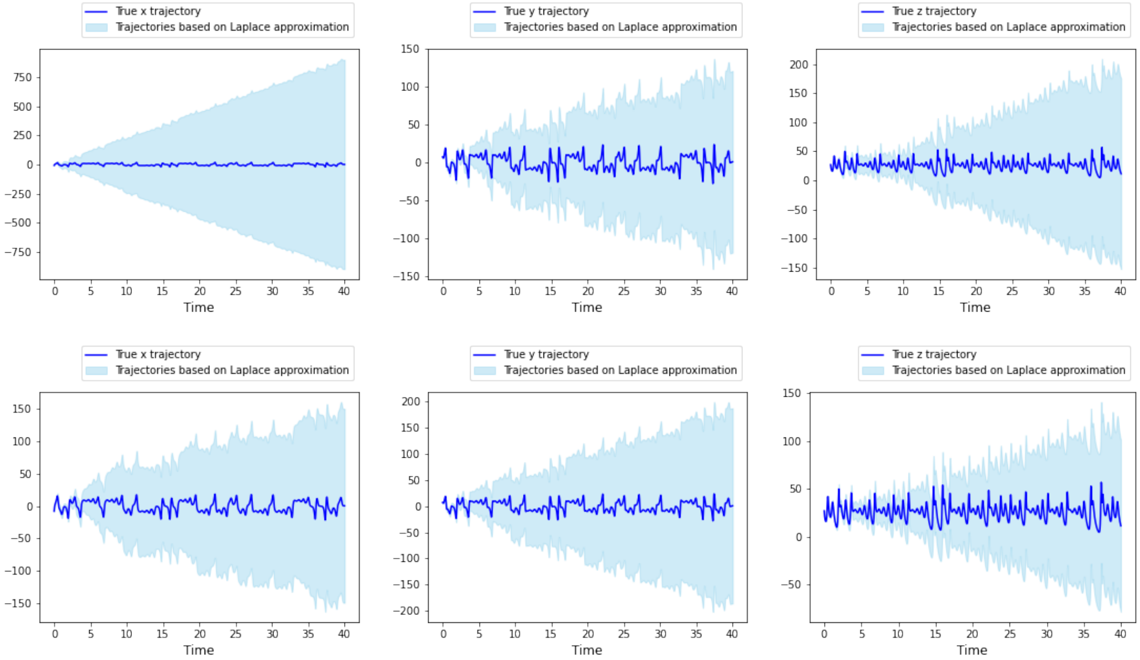


Figure 2.12: True Lorenz-63 trajectory and IAS estimation. Top: two IAS iterations. Bottom: five IAS iterations. Blue line is the true dynamics. Shaded regions are constructed from 2.5 and 97.5 credible levels of coefficients based on Laplace approximation.

mation error between the variational distribution and the true posterior.

- Combining VIAS with iterative ensemble Kalman methods [Chada et al., 2021] may allow us to extend the current variational framework to nonlinear inverse problems, and to enhance the scalability to high dimensional linear and nonlinear inverse problems. In addition, we also envision that VIAS may provide a natural way to promote sparsity in iterative ensemble Kalman methods that are based on L^2 penalties.

- We have explored the potential of VIAS to perform approximate Bayesian inference and provide meaningful uncertainty quantification. In future work, our variational approach will be combined with Markov chain Monte Carlo [de Freitas et al., 2001] and sequential Monte Carlo [Naesseth et al., 2018] for fully-Bayesian inference.

- More general hyperpriors could be considered within our variational framework. In this direction, the work [Calvetti et al., 2020b] has investigated more flexible generalized gamma hyperpriors in the context of MAP estimation.

- A consequence of the mean-field restriction on the variational family is underestimation of the variance of the components of the unknown. An interesting and practically useful direction for future research is to alleviate such variance underestimation phenomenon following the ideas in [Giordano et al., 2015].

CHAPTER 3

HIERARCHICAL ENSEMBLE KALMAN METHODS WITH SPARSITY-PROMOTING GENERALIZED GAMMA HYPERPRIORS

3.1 Introduction

Ensemble Kalman methods are a family of derivative-free, black-box optimization algorithms that rely on Kalman-based formulas to propagate an ensemble of interacting particles. Propagating an ensemble, rather than a single estimate, is advantageous when solving high-dimensional inverse problems with complex forward maps: the ensemble provides preconditioners and surrogate forward map derivatives to accelerate the optimization. This chapter sets forth a new computational framework to leverage ensemble Kalman methods for the numerical solution of nonlinear inverse problems with flexible regularizers beyond standard ℓ_2-penalties. In order to minimize objective functions that cannot be expressed in the usual nonlinear least-squares form, our approach introduces auxiliary variables which enable the use of ensemble Kalman methods within a reweighted least-squares procedure. In so doing, our framework generalizes the *iterative alternating scheme* (IAS), a MAP estimation strategy for sparse linear inverse problems, to nonlinear inverse problems. We show the effectiveness of our framework in illustrative examples, including compressed sensing and subsurface flow inverse problems.

We will adopt a Bayesian viewpoint to derive our methodology. Specifically, the minimizer approximated by our main algorithm corresponds to the *maximum a posteriori* (MAP) estimate of a hierarchical Bayesian model with conditionally Gaussian prior and generalized gamma hyperpriors. We propose an iterative algorithm for MAP estimation, which alternates between updating the unknown with an ensemble Kalman method and updating the hyperparameters in the regularization. The resulting method imposes sparsity while preserving the computational benefits of ensemble Kalman methods. In particular, for linear settings and under suitable conditions on the prior hyperparameters, our iterative scheme is globally convergent. Furthermore, our algorithm can provide uncertainty quantification in linear or mildly nonlinear settings. Moreover, our approach retains convexity of the objective for certain nonlinear forward maps. Most importantly, numerical experiments demonstrate the usefulness of prior hyperparameters that result in non-convex objectives but strongly promote sparsity.

3.1.1 Related Work

Ensemble Kalman methods, overviewed in [Evensen, 2009, Chada et al., 2021], were first developed as scalable filtering schemes for high-dimensional state estimation in numerical weather forecasting [Evans, 1995, Evans and Leeuwen, 1996]. Since then, they have become popular algorithms in data assimilation, inverse problems, and machine learning. The papers [Gu and Oliver, 2007, Li and Reynolds, 2007, Reynolds et al., 2006] pioneered the development of ensemble Kalman methods for inverse problems in petroleum engineering and the geophysical sciences. Similar algorithms were introduced in [Iglesias et al., 2014, Iglesias, 2016] inspired by classical regularization schemes [Hanke, 1997]. Ensemble Kalman methods have grown into a rich family of computational tools for the numerical solution of inverse problems; our computational framework can incorporate sparsity-promoting regularization into any of the numerous existing variants [Chada et al., 2021]. In this chapter, we consider two implementations based on the *iterative ensemble Kalman filter* (IEKF) and the *iterative ensemble Kalman filter with statistical linearization* (IEKF-SL). All necessary methodological background will be provided in Section 3.3 below. Recent theoretical work on ensemble Kalman methods has established continuous-time and mean-field limits, as well as various convergence results, e.g. [Schillings and Stuart, 2017, Blömker et al., 2019, Blömker et al., 2018, Chada and Tong, 2022, Herty and Visconti, 2019, Ding and Li, 2021, Kovachki and

Stuart, 2019, Huang et al., 2022, Al-Ghattas and Sanz-Alonso, 2024].

Despite the effectiveness of ensemble Kalman methods in high-dimensional nonlinear inverse problems, efforts to broaden their scope by accommodating a wider range of regularization techniques are only starting to emerge. The paper [Chada et al., 2020] introduced Tikhonov ℓ_2-regularization and [Lee, 2021] generalized this idea using a transformation to turn ℓ_2-penalties into ℓ_p-penalties. This latter work highlights the key importance of suitably regularizing the inverse problem to achieve sparse reconstructions with ensemble Kalman methods. Both approaches in [Chada et al., 2020, Lee, 2021] operate in an augmented space of dimension $(k+d)$ where k is the dimension of the data and d is the dimension of the parameter space. Our method, in contrast, operates in d-dimensional space. Furthermore, the transformation in [Lee, 2021] involves a term that is exponential in $\frac{2}{p}$, which causes overflow for small p and limits the choice of penalties that can be implemented; this issue prevents, in particular, accurate approximation of ℓ_0-regulatization, which is important in applications [Pan et al., 2014, Louizos et al., 2017, Natarajan, 1995]. Our method is composed of bi-level iterations where the first level is given by standard ensemble Kalman iterations, while the second level modifies the covariance matrix of the particles to induce a regularization effect. In a similar spirit, [Armbruster et al., 2022] proposed stabilizing ensemble Kalman methods by modifying the ensemble covariance matrix to induce regularization and achieve a faster convergence rate. Another approach for imposing sparsity-promoting regularization through thresholding was introduced in [Schneider et al., 2022]. Cross-entropy loss [Kovachki and Stuart, 2019] and logistic loss [Pidstrigach and Reich, 2022] have also been recently considered.

The hierarchical prior model used to derive our methodology, along with an *iterative alternating scheme* (IAS) for MAP estimation, were introduced in [Calvetti et al., 2020b] for sparse linear inverse problems. In fact, our computational framework builds on a series of articles [Calvetti et al., 2019b, 2020a, 2019a, 2015] that considered (generalized) gamma hyperpriors for sparse solution of *linear* inverse problems. This line of work also developed the IAS algorithm to compute the MAP estimate, which alternates between updating the unknown using conjugate gradient with early stopping and updating the regularization using closed formulas. Extensions to tackle both parameter estimation and uncertainty quantification tasks were introduced in [Agrawal et al., 2022, Law and Zankin, 2022] using variational inference. The IAS algorithm has been successfully implemented in applied linear inverse problems, including brain activity mapping from magnetoencephalography and identification of dynamics from time series data [Calvetti et al., 2015, 2019a, Agrawal et al., 2022]. These hierarchical Bayesian techniques [Congdon, 2010, Wikle and Berliner, 2007, Gelman et al., 1995] are rooted in a broader literature on signal processing with emphasis on sparsity [Gorodnitsky and Rao, 1997, Daubechies et al., 2010] and are inspired by the classical reweighted least-squares algorithm [Green, 1984]. Our framework is reminiscent of the centered hierarchical method for ensemble Kalman inversion in [Chada et al., 2018], but with an additional layer of (generalized) gamma hyperprior. Sparsity-promoting algorithms and models are also essential in statistical science [Tibshirani, 1996, Carvalho et al., 2009]; our hierarchical approach has connections with empirical Bayes statistical methods [Robbins, 1992] and with bi-level and data-driven methods for inverse problems [Bard, 2013, Arridge et al., 2019].

3.1.2 Main Contributions

The main contributions of this chapter can be summarized as follows:

- We introduce a flexible computational framework to incorporate regularization techniques in ensemble Kalman methods, including sparsity-promoting ℓ_p-penalties with $p \in (0, 1)$.
- Our framework generalizes the IAS algorithm [Calvetti et al., 2018, 2019b, 2020b,a] to nonlinear inverse problems.

- For linear or mildly nonlinear inverse problems, our framework provides uncertainty quantification; therefore, our methodology complements and generalizes variational inference techniques [Agrawal et al., 2022] that are only applicable for linear inverse problems with gamma hyperpriors.

- Our presentation gives a Bayesian interpretation to statistical methods that rely on ℓ_p-penalties, such as *least absolute shrinkage and selection operator* (LASSO) and adaptive LASSO [Tibshirani, 1996, Zou, 2006].

- We demonstrate the effectiveness of our methods in three computed examples: compressed sensing, a PDE-constrained inverse problem with convex forward map, and an elliptic inverse problem in subsurface flow. We present two implementations of our framework based on IEKF and IEKF-SL ensemble Kalman methods. The code used to reproduce our results can be found in https://github.com/hwkim12/Lp-regularized-IEKF.

3.1.3 Outline

Section 3.2 overviews the hierarchical model used to derive our computational framework. Section 3.3 introduces our methodology and discusses its theoretical underpinnings. Section 3.4 contains numerical results. Section 3.5 closes with a summary of our work.

Notation For matrix P, we write $P \succ 0$ if P is symmetric positive definite. For $P \succ 0$, we denote by $\|\cdot\|_P^2 := |P^{-1/2}\cdot|^2$ the squared Mahalanobis norm induced by the matrix P, where $|\cdot|$ denotes the Euclidean norm.

3.2 Hierarchical Bayesian Model

Consider the inverse problem of reconstructing an unknown $u \in \mathbb{R}^d$ from noisy data $y \in \mathbb{R}^k$, related by

$$y = \mathcal{G}(u) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \Gamma). \tag{3.1}$$

Here $\mathcal{G} : \mathbb{R}^d \to \mathbb{R}^k$ is a given forward map and ε represents a Gaussian measurement error with a given covariance $\Gamma \succ 0$. In particular, we are interested in the case when k is substantially smaller than d. Such problem setting arises in numerous applications in data assimilation, inverse problems, machine learning, and statistics —see, for instance, [Sanz-Alonso et al., 2023, Chada et al., 2021, Stuart, 2010] and references therein.

To encode the prior belief that u is sparse, we adopt the hierarchical model introduced in [Calvetti et al., 2020b]. First, let the components of u be independent random variables

$$u_i \sim \mathcal{N}(0, \theta_i), \quad \theta_i > 0, \quad 1 \le i \le d, \tag{3.2}$$

with unknown variances θ_i. Then, small θ_i yields shrinkage in the estimation of the corresponding unknown component u_i. Equation (3.2) defines a conditional Gaussian prior

$$\pi(u \,|\, \theta) \propto \frac{1}{\prod_{i=1}^d \sqrt{\theta_i}} \exp\left(-\frac{1}{2}\|u\|_{D_\theta}^2\right), \quad D_\theta := \operatorname{diag}(\theta_1, \ldots, \theta_d). \tag{3.3}$$

To modulate the level of sparsity, we set a hyperprior on θ from the generalized gamma distribution [Korolev and Zeifman, 2019] given by

$$\pi_{\text{hyper}}(\theta) := \prod_{i=1}^d \frac{|r|}{\Gamma(\beta)\vartheta_i^\beta} \theta_i^{r\beta-1} \exp\left(-\frac{\theta_i^r}{\vartheta_i}\right) = \frac{|r|^d}{\Gamma(\beta)^d} \prod_{i=1}^d \left(\frac{\theta_i^{r\beta-1}}{\vartheta_i^\beta}\right) \exp\left(-\frac{\theta_i^r}{\vartheta_i}\right), \tag{3.4}$$

where $\beta \in \mathbb{R}_{>0}$, $\{\vartheta_i\}_{i=1}^d \subset \mathbb{R}_{>0}$, $r \in \mathbb{R} \setminus \{0\}$, and $\Gamma(\cdot)$ is the Gamma function. Notice that if $r = 1$, the hyperprior becomes a product of gamma distributions with a common shape parameter β and an individual scale parameter ϑ_i. Similarly, if $r = -1$, the hyperprior becomes a product of inverse gamma distributions with a common shape parameter β and an individual scale parameter $\frac{1}{\vartheta_i}$. We refer to r as the regularization parameter. In Section 3.3 we will demonstrate that the value of r determines the level of regularization in the reconstruction of u.

Bayes's formula combines the likelihood function implied by the data model (3.1)

$$\pi(y \,|\, u, \theta) = \pi(y \,|\, u) \propto \exp\left(-\frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2\right), \tag{3.5}$$

together with the hierarchical prior $\pi_{\text{prior}}(u,\theta) := \pi(u \,|\, \theta)\pi_{\text{hyper}}(\theta)$ defined by (3.3)–(3.4), to give the posterior distribution

$$\begin{aligned} \pi(u, \theta \,|\, y) &\propto \pi(y \,|\, u, \theta)\pi_{\text{prior}}(u,\theta) \\ &= \pi(y \,|\, u, \theta)\pi(u \,|\, \theta)\pi_{\text{hyper}}(\theta) \propto \exp\big(-\mathsf{J}(u,\theta)\big), \end{aligned}$$

where

$$\mathsf{J}(u,\theta) := \overbrace{\frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + \rlap{$\underbrace{\phantom{\frac{1}{2}\|u\|_{D_\theta}^2 - \Big(r\beta - \frac{3}{2}\Big)\sum_{i=1}^d \log\frac{\theta_i}{\vartheta_i} + \sum_{i=1}^d \frac{\theta_i^r}{\vartheta_i}}}_{(b)}$}\frac{1}{2}\|u\|_{D_\theta}^2}^{(a)} - \Big(r\beta - \frac{3}{2}\Big)\sum_{i=1}^d \log\frac{\theta_i}{\vartheta_i} + \sum_{i=1}^d \frac{\theta_i^r}{\vartheta_i}. \tag{3.6}$$

The MAP estimate is the maximizer of the posterior density or, equivalently, the minimizer of the objective (3.6). The computational framework developed in the next section will minimize $\mathsf{J}(u,\theta)$ iteratively, alternating between minimizing the term (a) with ensemble Kalman methods and minimizing the term (b) using closed formulas for suitable choices of hyperparameters.

In the following section we 1) propose a generic optimization method for the objective function in (3.6) and build a connection with IAS [Calvetti et al., 2015, 2019b], which is only applicable when the forward map $\mathcal{G}$ is linear; 2) characterize a region of hyperparameter values (r,β) for which the objective function is convex under suitable assumptions on $\mathcal{G}$; and

3) elucidate the relationship between the parameters (r, β) and the regularization imposed on the reconstruction of u.

3.3 Iterative Alternating Ensemble Kalman Filters

In this section we propose an iterative optimization method to compute the MAP estimate. Our approach attempts to minimize the objective (3.6) in a block coordinate-wise fashion. The first term (a) can be minimized over u using ensemble Kalman methods and the second term (b) can be easily minimized over θ for suitable choices of hyperparameters. Motivated by these observations, we initialize θ^0 and compute iteratively the following updates:

$$\begin{aligned} u^{\ell+1} &= \arg\min_u \mathsf{J}(u, \theta^\ell) = \arg\min_u \frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + \frac{1}{2}\|u\|_{D_{\theta^\ell}}^2 \ , \\ \theta^{\ell+1} &= \arg\min_\theta \mathsf{J}(u^{\ell+1}, \theta) = \arg\min_\theta \frac{1}{2}\|u^{\ell+1}\|_{D_\theta}^2 - \left(r\beta - \frac{3}{2}\right)\sum_{i=1}^d \log\frac{\theta_i}{\vartheta_i} + \sum_{i=1}^d \frac{\theta_i^r}{\vartheta_i} \ , \end{aligned} \tag{3.7}$$

until a stopping criterion is satisfied. We will demonstrate that, for suitable choices of $(r, \beta, \{\vartheta_i\}_{i=1}^d)$, the above iterative procedure agrees with minimizing the ℓ_p-penalized objective

$$\mathsf{J}_p(u) := \frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + \frac{1}{2}\sum_{i=1}^d w_i |u_i|^p,$$

where the regularization level p is determined by r, and the weight w_i is determined by p and the scale parameter ϑ_i. In other words, we will show that the minimizer u found by the iterative procedure (3.7) solves an ℓ_p-regularized nonlinear optimization problem, provided appropriate hyperparameters $(r, \beta, \{\vartheta_i\}_{i=1}^d)$. The following two subsections describe the numerical implementation of each update rule and the choice of hyperparameters. Subsection 3.3.3 contains the main algorithms, and Subsection 3.3.4 provides sufficient conditions on the forward map and hyperarameters that ensure convexity of the objective (3.6).

3.3.1 Updating u

For a generic forward map $\mathcal{G}$, updating u in (3.7) requires solving a nonlinear least-squares optimization problem. To this end, we will use ensemble Kalman methods designed to minimize Tikhonov-Phillips (TP) objectives of the form

$$\mathsf{J}_{\text{TP}}(u) = \frac{1}{2}\|y - \mathcal{G}(u)\|_{\Gamma}^2 + \frac{1}{2}\|u - m\|_{P}^2, \tag{3.8}$$

where $\Gamma, P \succ 0$, and m are given. We remark, for later reference, that minimizing (3.8) can be interpreted as maximizing the posterior density with Gaussian likelihood and prior given by

$$\pi(y|u) = \mathcal{N}\big(\mathcal{G}(u), \Gamma\big), \tag{3.9}$$

$$\pi(u) = \mathcal{N}(m, P). \tag{3.10}$$

Since we are interested in the update (3.7), we will take $m = 0$ throughout; $P = D_{\theta^\ell}$ will be iteratively updated in subsequent developments.

Starting from an *initial ensemble* $\{u_0^{(n)}\}_{n=1}^N$, ensemble Kalman methods update the ensemble in an artificial discrete-time index t

$$\{u_t^{(n)}\}_{n=1}^N \mapsto \{u_{t+1}^{(n)}\}_{n=1}^N$$

using Kalman formulas that promote fitness of the ensemble with data and with the prior distribution (3.10) implied by the Tikhonov-Phillips regularization. The goals of fitting data and fitting the prior are balanced using an ensemble-based Kalman gain matrix, as well as certain additional random perturbation terms that control the long-time distribution of the ensemble in the large N limit. We view the iteration subscript t as a discrete-time index because the evolution of the ensemble may arise from the discretization of a system of stochastic differential equations, coupled by the ensemble-based Kalman gain. Once the ensemble reaches statistical equilibrium, we report the ensemble mean as an approximate solution of the optimization problem of interest. In this subsection we introduce two types of ensemble Kalman methods, IEKF and IEKF-SL, which we employ in the update of u. These two algorithms differ in how they construct the Kalman gain and in how they introduce random perturbations in the ensemble update. To formulate these algorithms we need some notation. Given the ensemble $\{u_t^{(n)}\}_{n=1}^N$ at time t, we denote ensemble empirical means by

$$m_t = \frac{1}{N}\sum_{n=1}^N u_t^{(n)}, \qquad g_t = \frac{1}{N}\sum_{n=1}^N \mathcal{G}\big(u_t^{(n)}\big),$$

and we denote empirical covariances and cross-covariances by

$$\begin{aligned} P_t^{uu} &= \frac{1}{N}\sum_{n=1}^N (u_t^{(n)} - m_t)(u_t^{(n)} - m_t)^\top, \\ P_t^{uy} &= \frac{1}{N}\sum_{n=1}^N \big(u_t^{(n)} - m_t\big)\big(\mathcal{G}(u_t^{(n)}) - g_t\big)^\top, \\ P_t^{yy} &= \frac{1}{N}\sum_{n=1}^N \big(\mathcal{G}(u_t^{(n)}) - g_t\big)\big(\mathcal{G}(u_t^{(n)}) - g_t\big)^\top. \end{aligned}$$

We will use repeatedly the principle of *statistical linearization* —see [Ungarala, 2012, Chada et al., 2021]— which we recall briefly. Notice that if $\mathcal{G}$ is linear, i.e., $\mathcal{G}(u) = Gu$, we have

$$P_t^{uy} = P_t^{uu} G^\top.$$

The principle of statistical linearization is to approximate the Jacobian of a generic nonlinear map $\mathcal{G}$ at time t using the above identity, namely

$$\mathcal{G}'(u_t^{(n)}) \approx (P_t^{uy})^\top (P_t^{uu})^{-1} =: G_t^N, \qquad n = 1, \dots, N.$$

Here and henceforth, $(P_t^{uu})^{-1}$ denotes the pseudoinverse of P_t^{uu}. We next present IEKF, IEKF-SL, and a unified framework providing more insights on these algorithms.

Iterative Ensemble Kalman Filter (IEKF)

Here we present the IEKF method introduced in [Chada et al., 2021]. The pseudocode is given in Algorithm 3.3.1. We refer to [Ungarala, 2012, Reynolds et al., 2006] for other variants of IEKF.

Algorithm 3.3.1 Iterative Ensemble Kalman Filter (IEKF)

```
Input: Number $T$ of iterations, step-size $\alpha$, covariance $P \succ 0$.
Initialization: Draw initial ensemble $u_0^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, P), \quad 1 \le n \le N$.
For $t = 0, 1, \ldots T$ do:
    1. Set $G_t^N = (P_t^{uy})^\top (P_t^{uu})^{-1}$.
    2. Update the Kalman gain $K_{t,0}^N = P_0^{uu}(G_t^N)^\top \big(G_t^N P_0^{uu}(G_t^N)^\top + \Gamma\big)^{-1}$.
    3. For $1 \le n \le N$, draw $y_t^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(y, \alpha^{-1}\Gamma)$.
    4. For $1 \le n \le N$, update
       $$u_{t+1}^{(n)} = u_t^{(n)} + \alpha\Big\{K_{t,0}^N\big(y_t^{(n)} - \mathcal{G}(u_t^{(n)})\big) + (I - K_{t,0}^N G_t^N)\big(u_0^{(n)} - u_t^{(n)}\big)\Big\}.$$
Output: Final ensemble mean $m_T$.
```

The Kalman gain $K_{t,0}^N$ is defined using the empirical covariance P_0^{uu} of the initial ensemble, the approximated Jacobian G_t^N of the forward map, and the covariance Γ of the measurement error. In the update of ensemble members, random perturbations are introduced only to the term which measures the discrepancy between data and the image of current ensemble members under the nonlinear map $\mathcal{G}$. Furthermore, the measure of fitness of ensemble members to the prior distribution is assessed by comparing each ensemble member with the corresponding initial ensemble member, drawn from the prior.

Employing P_0^{uu} in the construction of the Kalman gain $K_{t,0}^N$ and $u_0^{(n)}$ in the update of

$u_t^{(n)}$ implicitly regularizes the ensemble by forcing all its members to remain in the linear span of the initial ensemble. This *initial subspace property* holds for several ensemble Kalman methods [Iglesias et al., 2014, Chada et al., 2020, 2021]. In addition, [Chada et al., 2021] has shown that, in a linear forward map setting with step-size $\alpha = 1$, the ensemble mean computed by the IEKF algorithm converges (as $N \to \infty$) in a single step ($T = 1$) to the posterior mean with likelihood and prior given by (3.9)–(3.10). The next three remarks discuss the choice of prior covariance, stopping criteria, and step-size with pointers to the literature.

Remark 3.3.1. In ensemble Kalman methods, the prior covariance P typically incorporates application-specific knowledge. For instance, the initial ensemble may be drawn from a uniform or log-normal distribution whose support reflects prior information [Iglesias and Yang, 2021, Schneider et al., 2017]. Instead of sampling the initial ensemble, one can specify it deterministically using the first principal components of a suitable covariance model [Iglesias et al., 2014]. These considerations may be used to determine a suitable initialization $P = D_{\theta^0}$ for our main algorithms in Subsection 3.3.3.

Remark 3.3.2. Instead of providing a total number T of iterations, a stopping rule can be used. For instance, one could use *Morozov's discrepancy principle* [Morozov, 1966, Groetsch, 1993]; continue the iteration until the discrepancy between the data and the forward mapping of m_t falls below the noise level, i.e.,

$$|y - \mathcal{G}(m_t)| \le \sqrt{\mathrm{tr}(\Gamma)}.$$

Remark 3.3.3. For simplicity we will consider a constant and fixed step-size α. The step-size can be chosen on-line with a line search method based on *Wolfe*'s condition or *ad hoc* procedures introducing additional hyperparameters [Gu and Oliver, 2007]. Non-constant and adaptive step-sizes have been employed in [Chada et al., 2020, Chada and Tong, 2022, Iglesias and Yang, 2021].

Iterative Ensemble Kalman Filter with Statistical Linearization (IEKF-SL)

Here we present the IEKF-SL introduced in [Chada et al., 2021]. The pseudocode is provided in Algorithm 3.3.2.

Algorithm 3.3.2 IEKF with Statistical Linearization (IEKF-SL)

```
Input: Number $T$ of iterations, step-size $\alpha$, covariance $P \succ 0$.
Initialization: Draw initial ensemble $u_0^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, P), \quad 1 \le n \le N$.
For $t = 0, 1, \dots T$ do:
    1. Set $G_t^N = (P_t^{uy})^\top (P_t^{uu})^{-1}$.
    2. Update the Kalman gain $K_t^N = P(G_t^N)^\top \left(G_t^N P (G_t^N)^\top + \Gamma\right)^{-1}$.
    3. For $1 \le n \le N$, draw $y_t^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(y, 2\alpha^{-1}\Gamma)$, $m_t^{(n)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 2\alpha^{-1}P)$.
    4. For $1 \le n \le N$, set
       $u_{t+1}^{(n)} = u_t^{(n)} + \alpha\Big\{K_t^N\big(y_t^{(n)} - \mathcal{G}(u_t^{(n)})\big) + (I - K_t^N G_t^N)\big(m_t^{(n)} - u_t^{(n)}\big)\Big\}$.
Output: Final ensemble mean $m_T$.
```

Unlike IEKF, IEKF-SL constructs the Kalman gain K_t^N using the prior covariance P, the approximated Jacobian G_t^N of the forward map, and the covariance Γ of the measurement error. Furthermore, in the update of ensemble members, it introduces random perturbations to both the data and prior terms. The measure of fitness of ensemble members to the prior distribution is assessed by comparing each member with a perturbed prior mean.

Although IEKF-SL does not have the initial subspace property, it has been shown to achieve superior performance in a variety of inverse problems [Chada et al., 2021]. In contrast to IEKF, the ensemble empirical mean and covariance of IEKF-SL converge, as $\alpha \to 0$ and $N, T \to \infty$, to the true posterior mean and covariance under the likelihood and prior model (3.9)–(3.10) when the forward map is linear. Therefore, for mildly nonlinear problems, IEKF-

SL can be used to build approximate credible intervals for the reconstruction, allowing us to quantify uncertainties.

Unified Framework through Stochastic Differential Equations

Both IEKF and IEKF-SL can be viewed as ensemble-based stochastic approximations of the deterministic extended Kalman filter. The extended Kalman filter finds the minimum of the objective function given in (3.8) by sequentially updating the initial guess u_0 according to the following rule:

$$u_{t+1} = u_t + \alpha\Big(K_t\big(y - \mathcal{G}(u_t)\big) + (I - K_t G_t)\big(m - u_t\big)\Big),$$

where $\alpha > 0$ is a step-size, $G_t = \mathcal{G}'(u_t)$ is the Jacobian of $\mathcal{G}$, and $K_t = PG_t^\top(G_t P G_t^\top + \Gamma)^{-1}$ is the Kalman gain matrix [Chada et al., 2021]. Setting $C_t = (I - K_t G_t)P$, we get from Woodbury's matrix inversion lemma that $K_t = C_t G_t^\top \Gamma^{-1}$. Hence we can rewrite the preceding update rule as

$$u_{t+1} = u_t + \alpha C_t\Big(G_t^\top \Gamma^{-1}\big(y - \mathcal{G}(u_t)\big) + P^{-1}\big(m - u_t\big)\Big), \tag{3.11}$$

which agrees with a Gauss-Newton iteration applied on the Tikhonov-Phillips objective (3.8), see [Bell and Cathey, 1993]. For small step-size α, one can view (3.11) as a discretization of the following differential equation, which describes the continuum version of the discrete trajectories of the iterates from the extended Kalman filter:

$$\frac{du_s}{ds} = C_s\Big(\mathcal{G}'(u_s)^\top \Gamma^{-1}\big(y - \mathcal{G}(u_s)\big) + P^{-1}(m - u_s)\Big),$$

where C_s acts as a preconditioner. Using the identities $C_t = (I - K_t G_t)P$ and $K_t = C_t G_t^\top \Gamma^{-1}$, one can show that

$$C_t^{-1} = G_t^\top \Gamma^{-1} G_t + P^{-1}, \tag{3.12}$$

which leads, in the continuum limit, to

$$C_s = \big(\mathcal{G}'(u_s)^\top \Gamma^{-1} \mathcal{G}'(u_s) + P^{-1}\big)^{-1}. \tag{3.13}$$

To give rise to an ensemble of random particles that roughly follow the continuous trajectory we defined, we employ the above deterministic differential equation as our drift term and introduce two different diffusion terms to obtain, for $1 \le n \le N$,

$$\begin{aligned} du_s^{(n)} &= C_s\Big(\mathcal{G}'(u_s)^\top \Gamma^{-1}\big(y - \mathcal{G}(u_s)\big) + P^{-1}(m - u_s)\Big)ds + C_s \mathcal{G}'(u_s)^\top \Gamma^{-\frac{1}{2}} dW_s^{(n)}, \\ du_s^{(n)} &= C_s\Big(\mathcal{G}'(u_s)^\top \Gamma^{-1}\big(y - \mathcal{G}(u_s)\big) + P^{-1}(m - u_s)\Big)ds + \sqrt{2C_s} dW_s^{(n)}, \end{aligned}$$

where the first and second stochastic differential equation will respectively correspond to IEKF and IEKF-SL. More specifically, discretization of the above stochastic differential equations, together with ensemble-based approximation of the Jacobian of $\mathcal{G}$, gives IEKF and IEKF-SL. First, applying Euler-Maruyama we get, for $1 \le n \le N$,

$$\begin{aligned} u_{t+1}^{(n)} &= u_t^{(n)} + \alpha C_t\Big(G_t^\top \Gamma^{-1}\big(y - \mathcal{G}(u_t^{(n)})\big) + P^{-1}(m - u_t^{(n)})\Big) + \sqrt{\alpha} C_t G_t^\top \Gamma^{-\frac{1}{2}} Z_t, \\ u_{t+1}^{(n)} &= u_t^{(n)} + \alpha C_t\Big(G_t^\top \Gamma^{-1}\big(y - \mathcal{G}(u_t^{(n)})\big) + P^{-1}(m - u_t^{(n)})\Big) + \sqrt{\alpha}\sqrt{2C_t} Z_t, \end{aligned}$$

where $Z_t \sim \mathcal{N}(0, I)$. From the form of preconditioner in (3.12), one can show that

$$\sqrt{\alpha}\sqrt{2C_t} Z_t \overset{\mathrm{d}}{=} \sqrt{2\alpha} C_t G_t^\top \Gamma^{-\frac{1}{2}} Z_t^y + \sqrt{2\alpha} C_t P^{-\frac{1}{2}} Z_t^m,$$

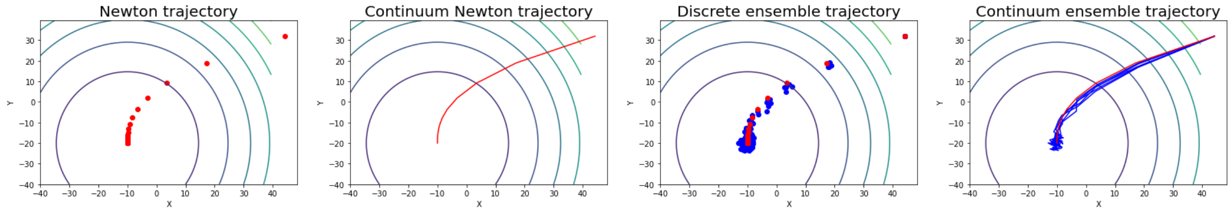


Figure 3.1: Leftmost: Newton iteration. Middle-left: continuum Newton trajectory. Middle-right: ensemble Kalman iteration. Rightmost: continuum ensemble trajectory.

where $Z_t^y, Z_t^m \sim^{\text{i.i.d.}}\sim \mathcal{N}(0, I)$. By introducing randomness to y through Z_i^y and m through Z_i^m, we get the following update rules, for $1 \le n \le N$,

$$\begin{aligned} u_{t+1}^{(n)} &= u_t^{(n)} + \alpha C_t \Big(G_t^\top \Gamma^{-1} \big(y_t^{(n)} - \mathcal{G}(u_t^{(n)}) \big) + P^{-1}(m - u_t^{(n)}) \Big), \qquad (3.14) \\ &\quad \text{with } y_t^{(n)} \sim \mathcal{N}(y, \alpha^{-1}\Gamma), \\ u_{t+1}^{(n)} &= u_t^{(n)} + \alpha C_t \Big(G_t^\top \Gamma^{-1} \big(y_t^{(n)} - \mathcal{G}(u_t^{(n)}) \big) + P^{-1}(m_t^{(n)} - u_t^{(n)}) \Big), \qquad (3.15) \\ &\quad \text{with } y_t^{(n)} \sim \mathcal{N}(y, 2\alpha^{-1}\Gamma), m_t^{(n)} \sim \mathcal{N}(y, 2\alpha^{-1}P). \end{aligned}$$

With random ensemble updates of the form (3.14) and (3.15), one can avoid computing the Jacobian of $\mathcal{G}$ by using statistical linearization, which leads to a coupling of the stochastic dynamics. Doing so, we will derive IEKF and IEKF-SL.

We first consider (3.14). If we approximate m by $u_0^{(n)}$, P by P_0^{uu} and use statistical linearization G_t^N in place of the Jacobian G_t in (3.14) we have, for $1 \le n \le N$,

$$u_{t+1}^{(n)} = u_t^{(n)} + \alpha C_{t,0} \Big((G_t^N)^\top \Gamma^{-1} \big(y_t^{(n)} - \mathcal{G}(u_t^{(n)}) \big) + (P_0^{uu})^{-1} (u_0^{(n)} - u_t^{(n)}) \Big),$$

where

$$C_{t,0} = \Big((G_t^N)^\top \Gamma^{-1} G_t^N + (P_0^{uu})^{-1} \Big)^{-1},$$

which leads to the IEKF scheme in Algorithm 3.3.1.

For IEKF-SL, we consider (3.15) and get, for $1 \le n \le N$,

$$u_{t+1}^{(n)} = u_t^{(n)} + \alpha C_t^N \Big((G_t^N)^\top \Gamma^{-1} \big(y_t^{(n)} - \mathcal{G}(u_t^{(n)}) \big) + P^{-1}(m_t^{(n)} - u_t^{(n)}) \Big),$$

where

$$C_t^N = \Big((G_t^N)^\top \Gamma^{-1} (G_t^N) + P^{-1} \Big)^{-1},$$

which leads to the IEKF-SL scheme in Algorithm 3.3.2.

3.3.2 Updating θ: Generalized Gamma and ℓ_p-Regularization

Once we solve the optimization problem for $u^{\ell+1}$, the proposed coordinate-wise minimization strategy updates θ by setting

$$\theta^{\ell+1} = \arg\min_\theta \frac{1}{2}\|u^{\ell+1}\|_{D_\theta}^2 + \Big(r\beta - \frac{3}{2}\Big)\sum_{i=1}^d \log\frac{\theta_i}{\vartheta_i} + \sum_{i=1}^d \frac{\theta_i^r}{\vartheta_i}. \tag{3.16}$$

In this subsection we demonstrate an update rule for θ based on a particular choice of hyperparameter values in the hyperprior $\pi_{\text{hyper}}(\theta)$ in (3.4). Specifically, we focus on r, β satisfying $r\beta = \frac{3}{2}$. This choice will suffice to illustrate how the prior directly relates to the level of regularization one imposes. Then, the general form of the objective function for θ in (3.16) becomes

$$\frac{1}{2}\|u\|_{D_\theta}^2 + \sum_{i=1}^d \frac{\theta_i^r}{\vartheta_i}. \tag{3.17}$$

In order to minimize this function with respect to θ_i, we can restrict our attention to minimizing

$$\frac{u_i^2}{2\theta_i} + \frac{\theta_i^r}{\vartheta_i}. \tag{3.18}$$

For $r > 0$, one can observe that

$$\lim_{\theta_i \to 0} \left(\frac{u_i^2}{2\theta_i} + \frac{\theta_i^r}{\vartheta_i} \right) = \infty \quad \text{and} \quad \lim_{\theta_i \to \infty} \left(\frac{u_i^2}{2\theta_i} + \frac{\theta_i^r}{\vartheta_i} \right) = \infty.$$

Therefore, the update for θ_i amounts to solving for the first order optimality condition

$$-\frac{1}{2}\frac{u_i^2}{\theta_i^2} + \frac{r}{\vartheta_i}\theta_i^{r-1} = 0,$$

which leads to the following update rule

$$\theta_i = \left(\frac{\vartheta_i u_i^2}{2r} \right)^{\frac{1}{r+1}}. \tag{3.19}$$

For $r = 1$ the update precisely agrees with the update rule for IAS [Calvetti et al., 2019b] with the sparsity parameter $\eta = 0$ and ith scale parameter $\theta_i^* = \frac{1}{\vartheta_i}$ in their notation.

Plugging the update rule (3.19) back to the objective (3.17), we get

$$\sum_{i=1}^{d} \left(\frac{u_i^2}{2\theta_i} + \frac{\theta_i^r}{\vartheta_i^r} \right) = (r+1) \sum_{i=1}^{d} \vartheta_i^{-\frac{1}{r+1}} \left(\frac{u_i^2}{2r} \right)^{\frac{r}{r+1}}.$$

By setting $p := \frac{2r}{r+1} \in (0, 2)$, the objective function given in (3.6) becomes

$$\mathsf{J}_p(u) := \frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + C_r \sum_{i=1}^{d} w_{i,r}|u_i|^p, \; C_r = \frac{r+1}{(2r)^{\frac{r}{r+1}}}, \; w_{i,r} = \vartheta_i^{-\frac{1}{r+1}}, \tag{3.20}$$

which can be viewed as an ℓ_p-regularized problem for any $p \in (0, 2)$. By adjusting r, one can impose different types of regularization. In particular, for $r = 1$, we have an ℓ_1-regularized problem, whose natural Bayesian interpretation sets the gamma distribution with shape parameter $\beta = \frac{3}{2}$ and scale parameter ϑ_i as the prior for θ_i. Furthermore, prior knowledge on the units/scales of each u_i is captured by the $w_{i,r}$ terms, determined by the hyperparameters

ϑ_i and the regularization parameter r. Such hyperprior-based component reweighting resembles the well-known adaptive LASSO [Zou, 2006], which includes component-wise weights when solving ℓ_1-minimization LASSO problems to remove the bias induced from the ℓ_1-regularization term. In contrast to adaptive LASSO, our reweighting is motivated as a step towards finding the MAP estimate of a hierarchical Bayesian model.

3.3.3 Main Algorithms

Our proposed iterative methodology combines the algorithmic ideas introduced in Subsections 3.3.1 and 3.3.2 to compute the u and θ updates in (3.7). The procedure is summarized in Algorithm 3.3.3.

Algorithm 3.3.3 ℓ_p-IEKF and ℓ_p-IEKF-SL

```
Input: Initial $\theta^0$, hyperparameters $r$ and $\{\vartheta_i\}_{i=1}^{d}$, step-size $\alpha$, number $T$ of inner iterations.
Iterate (outer iteration) until convergence:
    1. Update $u^{\ell+1} := m_T$ running IEKF/IEKF-SL (inner iteration) with a step size $\alpha$ and an initial covariance $P := D_{\theta^\ell}$.
    2. Update $\theta_i^{\ell+1} := \left(\frac{\vartheta_i}{2r}\right)^{\frac{1}{r+1}} |u_i^{\ell+1}|^p$, where $p := \frac{2}{r+1}$.
    3. $\ell \to \ell + 1$.
Output: Final ensemble mean $m_T$.
```

The two coordinate-wise optimization steps serve two distinct purposes. When optimizing for u, the reconstruction of the unknown is updated with a given regularization; when optimizing for θ, the regularization is updated. The algorithm involves two nested iterations. First, each u update runs IEKF/IEKF-SL for T iterations. Second, u and θ are iteratively updated, in alternating fashion, until convergence. We call the first type of iteration *inner iteration* and the latter type *outer iteration.* As outer iterations update θ, we will also refer to them as outer regularization iterations.

The proposed method utilizes an ensemble to minimize

$$\mathsf{J}(u,\theta) = \frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + \frac{1}{2}\|u\|_{D_\theta}^2 + \sum_{i=1}^{d} \frac{\theta_i^r}{\vartheta_i} \tag{3.21}$$

in a coordinate-wise fashion to obtain the minimum of the lower-dimensional objective

$$\mathsf{J}_p(u) = \frac{1}{2}\|y - \mathcal{G}(u)\|_\Gamma^2 + C_r \sum_{i=1}^{d} w_{i,r}|u_i|^p. \tag{3.22}$$

One can show that $\mathsf{J}(u,\theta)$ agrees with $\mathsf{J}_p(u)$ along the manifold

$$\left\{ (u,\theta) : \theta_i = \frac{\vartheta_i^{\frac{1}{r+1}}}{(2r)^{\frac{1}{r+1}}} |u_i|^{\frac{2}{r+1}} \right\}.$$

Hence by element-wise optimizing $\mathsf{J}(u,\theta)$, we can recover the minimizer of $\mathsf{J}_p(u)$.

Algorithm 3.3.3 contains the pseudocode for ℓ_p-IEKF and ℓ_p-IEKF-SL. For $r \in [1,2)$ under a linear forward map setting, the objective function in (3.21) is globally convex, guaranteeing the existence and uniqueness of a global minimizer of (3.22). For $r \in (0,1)$, the objective function in (3.21) may not be convex thus may admit multiple local minimizers, even for linear forward maps. Sufficient conditions for convexity are stated and proved in Subsection 3.3.4. The practical efficiency of ℓ_p-IEKF and ℓ_p-IEKF-SL for $r \in (0,1)$ is demonstrated in Section 3.4.

Remark 3.3.4. In building the Kalman gain matrix, the IEKF and IEKF-SL Algorithms 3.3.1 and 3.3.2 require the inversion of a $k \times k$ matrix, where k is the dimension of the data y. In contrast, the implementation in [Lee, 2021] requires inversion of a $(k+d) \times (k+d)$ matrix, where d is the dimension of the unknown u (see Remark 3.3. in [Lee, 2021] for further details). However, the Algorithm 3.3.3 contains additional outer iterations when compared to [Lee, 2021], which partly offset the computational gain of working with smaller matrices.

Remark 3.3.5. In contrast to IEKF, ℓ_p-IEKF partially preserves the initial subspace property: the output of ℓ_p-IEKF lies in the span of the initial ensemble of the last u update step, rather than the span of the initial ensemble of the first u update step. Outer regularization iterations can be viewed as a way to adaptively modify the prior covariance for the u update to reflect the sparse structure of the true parameter. As each outer iteration modifies the prior covariance, the initial subspace for each u update step will change accordingly.

Remark 3.3.6. For the u update step, one may use other ensemble Kalman methods such as *Ensemble Kalman Inversion* (EKI) [Iglesias et al., 2014] or *Tikhonov Ensemble Kalman Inversion* (TEKI) [Chada et al., 2020]. In contrast to IEKF and IEKF-SL, for linear forward maps the ensemble obtained using EKI and TEKI collapse to a single point in the long time limit [Chada et al., 2021]. Consequently, EKI and TEKI do not provide uncertainty quantification, even in linear or mildly nonlinear settings, unless suitably stopped. The algorithm IEKF-SL was designed so that in linear settings the empirical covariance of the ensemble approximates the true posterior covariance in the long time asymptotic. We point out that the method in [Lee, 2021] inherits the ensemble collapse from EKI and TEKI, as it leverages a bijective mapping of each particle.

Remark 3.3.7. To determine when to terminate the outer iteration for ℓ_p-IEKF and ℓ_p-IEKF-SL, one may monitor the relative change of iterates. For instance, one may set a small tolerance $\tau > 0$ and terminate if

$$\frac{\|u^{\ell+1} - u^{\ell}\|_\infty}{\|u^{\ell}\|_\infty} < \tau.$$

Remark 3.3.8. Although we have focused on a particular choice of hyperparameter values, namely $r\beta = \frac{3}{2}$, the general framework extends beyond this choice to $r\beta > \frac{3}{2} + \delta$ for arbitrary δ, and to alternative r. In those cases, the update function which maximizes θ given u is expressed implicitly as the solution to an initial value problem which can be easily solved to update θ [Calvetti et al., 2020b]. One can also consider $r = -1$, which produces effective penalty terms corresponding to prior distributions whose tails decay as power laws in u. For $r = -1$ —which corresponds to imposing an inverse gamma prior— the objective function for the θ_i update is given by

$$\theta_i = \frac{1}{\beta + \frac{3}{2}} \left(\frac{u_i^2}{2} + \frac{1}{\vartheta_i} \right).$$

Assuming $\vartheta_i = 1$ for all $i = 1, \dots, d$ with $\kappa = \beta + \frac{3}{2}$, the prior component for u is given by

$$\pi(u) \propto \exp \left(-\sum_{i=1}^{d} \log(u_i^2 + 2)^\kappa \right) = \prod_{i=1}^{d} \frac{1}{(u_i^2 + 2)^\kappa}.$$

In the limit as $\beta \to 0$, i.e. $\kappa \to \frac{3}{2}$, the prior distribution for each component u_i approaches the Student distribution with two degrees of freedom, a heavy-tailed distribution which favors outliers.

3.3.4 Convexity

Proposition 3.3.9 below gives conditions on the forward map $\mathcal{G}$ and the parameters (r, β) that ensure convexity of the objective function in (3.6). A concrete numerical example where the assumptions in the following proposition hold is given in Subsection 3.4.2. For convex objectives, our methodology can be viewed as an ensemble approximation of a coordinate descent scheme that is globally convergent under mild assumptions [Tseng, 2001].

Proposition 3.3.9. *Let $\beta, r > 0$. Then the following holds:*

1. *If $r \geq 1$ or $r \leq 0$, $\eta = r\beta - \frac{3}{2} \geq 0$ and $\sum_{i=1}^{n} \mathcal{G}_i(u) \nabla^2 \mathcal{G}_i(u) \succcurlyeq 0$, then the objective function $\mathsf{J}(u, \theta)$ in (3.6) is convex everywhere.*

2. *If $0 < r < 1$, $\eta = r\beta - \frac{3}{2} \geq 0$ and $\sum_{i=1}^{n} \mathcal{G}_i(u) \nabla^2 \mathcal{G}_i(u) \succcurlyeq 0$, then the objective function*

$\mathsf{J}(u,\theta)$ *in* (3.6) *is convex provided that, for all* $i \in \{1,\dots,d\}$,

$$\frac{\theta_i}{\vartheta_i} \le \left(\frac{\eta}{r(1-r)}\right)^{\frac{1}{r}}. \tag{3.23}$$

Proof. The Hessian of $\mathsf{J}(u,\theta)$ is given by

$$H = \nabla^2 \mathsf{J}(u,\theta) = \begin{bmatrix} \nabla_u \nabla_u \mathsf{J}(u,\theta) & \nabla_\theta \nabla_u \mathsf{J}(u,\theta) \\ \nabla_u \nabla_\theta \mathsf{J}(u,\theta) & \nabla_\theta \nabla_\theta \mathsf{J}(u,\theta) \end{bmatrix},$$

where

$$\begin{aligned} \nabla_u \nabla_u \mathsf{J}(u,\theta) &= \nabla \mathcal{G} \nabla \mathcal{G}^\top + \sum_{i=1}^{n} \mathcal{G}_i(u) \nabla^2 \mathcal{G}_i(u) + D_\theta^{-1}, \\ \nabla_u \nabla_\theta \mathsf{J}(u,\theta) &= \nabla_\theta \nabla_u \mathsf{J}(u,\theta) = \operatorname{diag}\left(-\frac{u_i}{\theta_i^2}\right), \\ \nabla_\theta \nabla_\theta \mathsf{J}(u,\theta) &= \operatorname{diag}\left(\frac{u_i^2}{\theta_i^3} + \frac{r(r-1)}{\vartheta_i^2}\left(\frac{\theta_i}{\vartheta_i}\right)^{r-2} + \frac{\eta}{\theta_i^2}\right). \end{aligned}$$

For any vector $q = \begin{bmatrix} v \\ w \end{bmatrix}$, we have

$$\begin{aligned} q^\top H q = \|\nabla \mathcal{G}^\top v\|^2 \quad &+ \sum_{i=1}^{n} \mathcal{G}_i(u)(v^\top \nabla^2 \mathcal{G}_i(u) v) \\ &+ \sum_{i=1}^{d} \frac{v_i^2}{\theta_i} + \sum_{i=1}^{d} \left(\frac{u_i^2}{\theta_i^3} + \frac{r(r-1)}{\vartheta_i^2}\left(\frac{\theta_i}{\vartheta_i}\right)^{r-2} + \frac{\eta}{\theta_i^2}\right) w_i^2 - 2\sum_{i=1}^{d} \frac{u_i}{\theta_i^2} v_i w_i \\ = \|\nabla \mathcal{G}^\top v\|^2 + \sum_{i=1}^{n} &\mathcal{G}_i(u)(v^\top \nabla^2 \mathcal{G}_i(u) v) \\ &+ \sum_{i=1}^{d} \frac{1}{\theta_i}\left(v_i - \frac{u_i w_i}{\theta_i}\right)^2 + \sum_{i=1}^{d} \left(\frac{r(r-1)}{\vartheta_i^2}\left(\frac{\theta_i}{\vartheta_i}\right)^{r-2} + \frac{\eta}{\theta_i^2}\right) w_i^2. \end{aligned}$$

From the assumption, the first three terms are always non-negative and the remaining term is non-negative if, for all $i \in \{1, \dots, d\}$,

$$\frac{r(r-1)}{\vartheta_i^2}\left(\frac{\theta_i}{\vartheta_i}\right)^{r-2} + \frac{\eta}{\theta_i^2} \geq 0,$$

which implies the conditions of the two different cases. □

3.4 Numerical Experiments

In this section we demonstrate the effectiveness of the proposed methodology in three examples: 1) underdetermined linear inverse problem; 2) nonlinear inverse problem with an explicit forward map that gives a convex objective for certain hyperparameter values; and 3) nonlinear elliptic inverse problem. For all three examples we assumed that only a few components of the unknown u are nonzero and compared $\ell_{0.5}/\ell_1$-IEKF/IEKF-SL with the vanilla IEKF/IEKF-SL. Throughout, the step-size of ensemble Kalman methods is set to be $\alpha = 0.5$ and the scale parameters are set to be $\vartheta_i = 1$ for all $1 \leq i \leq d$. These choices suffice to illustrate the successful regularization achieved by our method when compared to vanilla ensemble Kalman methods with ℓ_2-regularization. In order to clearly compare different algorithms and regularization techniques, we report the evolution of the corresponding ensembles instead of using a stopping criterion.

3.4.1 Linear Inverse Problem

Our first example explores the performance of our methods in two tasks: point estimation and uncertainty quantification. For the point estimation task, we provide comparisons with the *iterative alternating scheme* (IAS) [Calvetti et al., 2019b], *least absolute shrinkage and selection operator* (LASSO) [Tibshirani, 1996], and *Tikhonov ensemble Kalman inversion* (TEKI) [Chada et al., 2020]. For the uncertainty quantification task, we compare credible intervals constructed using the empirical distribution of the ensembles produced by our algorithm with credible intervals constructed using the *variational iterative alternating scheme* (VIAS) [Agrawal et al., 2022].

Setting

Consider the linear inverse problem

$$y = Gu + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, 0.01 I_{30}),$$

where each component of $G \in \mathbb{R}^{30\times 300}$ is independently sampled from the standard Gaussian distribution. We assume that the true parameter $u \in \mathbb{R}^{300}$ has four nonzero components. Our goal is to recover u from $y \in \mathbb{R}^{30}$.

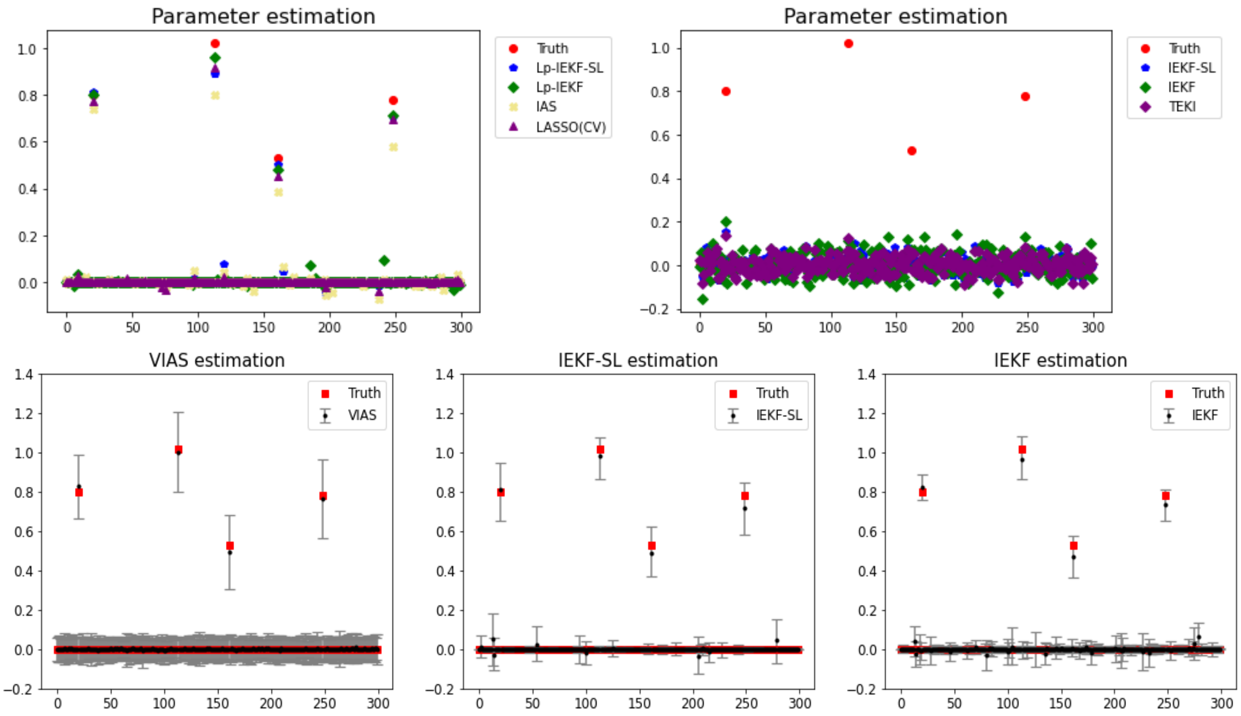


Figure 3.2: Parameter estimation and uncertainty quantification in linear example with $\ell_{0.5}$-regularizations on IEKF and IEKF-SL. Top row: parameter estimation. Bottom row: uncertainty quantification via approximate credible intervals.

For point estimation, we compare results obtained with $\ell_{0.5}$-IEKF, $\ell_{0.5}$-IEKF-SL, IAS, and LASSO. For both $\ell_{0.5}$-IEKF and $\ell_{0.5}$-IEKF-SL, a total of $N = 300$ ensemble members were used with $T = 30$ inner iterations and 10 outer iterations. The particles $u_0^{(n)}$ of the initial ensemble were independently sampled from a centered Gaussian with covariance $0.1I_{300}$. The sparsity parameter of IAS, denoted by η in [Calvetti et al., 2019b], was set to be zero. For LASSO, the regularization coefficient was chosen based on 10-fold cross-validation (CV).

In addition to obtaining a point estimate for the parameter of interest, one can build approximate credible intervals based on the ensemble members employed in IEKF/IEKF-SL. Like in Markov chain Monte Carlo, after enough iterations ensemble members serve as good proxies for samples from the true posterior distribution in linear or mildly nonlinear settings. From these ensemble members one can obtain 2.5/97.5th sample percentiles to construct approximate 95 percentile credible intervals for each component of the parameter. We provide such approximate credible intervals for $\ell_{0.5}$-IEKF/IEKF-SL with $N = 300$ ensemble members. A total of eight outer iterations and $T = 30$ inner IEKF/IEKF-SL iterations were run. In order to demonstrate the effectiveness of these approximate credible intervals, we also present approximate credible intervals based on VIAS [Agrawal et al., 2022] in Figure 3.2. The parameters of VIAS, were set to be $b = 0.1$ and $s = -0.495$, with a total of 40 iterations.

From Figure 3.2, we can observe the effectiveness of our regularized methods in estimating the true parameter. The proposed $\ell_{0.5}$-IEKF/IEKF-SL clearly outperforms the vanilla IEKF/IEKF-SL in the estimation task. It is shown to be competitive with LASSO and IAS, which are only applicable in linear settings. Although the convexity of (3.6) is not guaranteed for $p = 0.5$, the numerical result in Figure 3.2 demonstrates successful regularization. In terms of uncertainty quantification, $\ell_{0.5}$-IEKF/IEKF-SL showed comparable performance in constructing approximate credible intervals to the recently proposed VIAS. These results clearly show that $\ell_{0.5}$-IEKF/IEKF-SL can preserve desirable properties of ensemble-based derivative-free optimization methods and IAS. To further investigate the convergence and the regularization effect of the parameter r, we also conducted extensive simulations using a less severely underdetermined system with $y \in \mathbb{R}^{100}$ to avoid possible instabilities caused by a small number of observations. To evaluate the regularization effect, we varied r values ranging from 0.1 to 2 and computed the ℓ_2-norm of the ℓ_1-IEKF/IEKF-SL estimates whose indices correspond to entries off the support of the true signal.

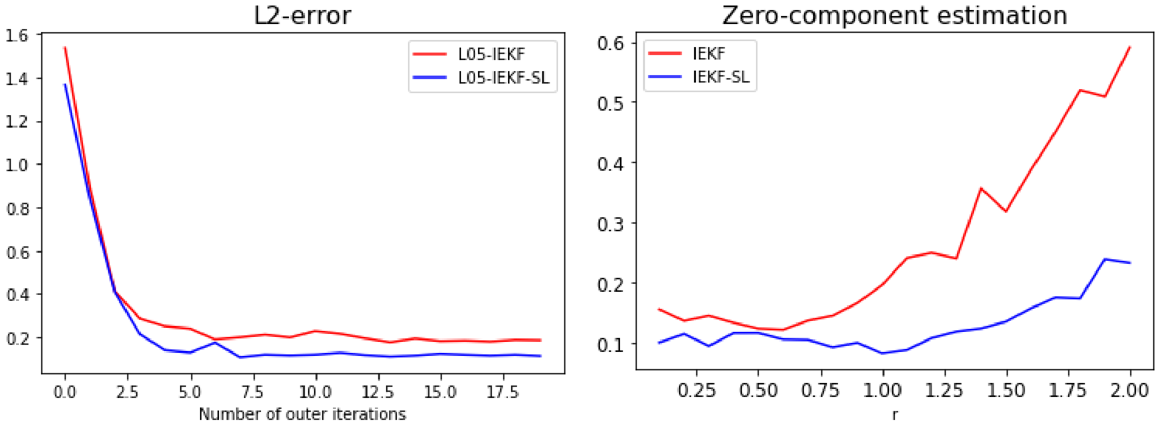


Figure 3.3: Left: ℓ_2-convergence comparison. Right: regularization effect of r.

The left plot in Figure 3.3 shows the ℓ_2 norm of the error $|u^\ell - u^*|$ between the true parameter u^* and the estimates obtained by ℓ_1-IEKF/IEKF-SL over 30 outer iterations. From the plot, ℓ_1-IEKF-SL is more accurate than ℓ_1-IEKF. The right plot in Figure 3.3 clearly demonstrates stronger regularization at small r values, as expected.

3.4.2 Nonlinear Inverse Problem with Explicit Forward Map

Here we study a nonlinear inverse problem introduced in [Kabanikhin, 2008], which has a closed-form forward map that satisfies the condition in Proposition 3.3.9. Our results demonstrate that strongly sparsity-promoting regularization techniques, for which convexity of the objective is lost, provide more accurate reconstruction.

Setting

Consider the following first-order partial differential equation (PDE)

$$\begin{aligned} \partial_{x_1} v - \partial_{x_2} v - u(x_1)v = 0, &\qquad (x_1, x_2) \in (0,1) \times (0,1), \\ v(x_1, 0) = \varphi(x_1), &\qquad x_1 \in [0,1]. \end{aligned} \tag{3.24}$$

If u is continuous and φ is continuously differentiable, then (3.24) admits the solution

$$v(x_1, x_2) = \varphi(x_1 + x_2) \exp\left(\int_{x_1+x_2}^{x_1} u(z)dz \right), \qquad (x_1, x_2) \in [0,1] \times [0,1].$$

For a given φ, the data y is sampled according to

$$y(x_1, x_2) = v(x_1, x_2) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.1^2), \quad (x_1, x_2) \in [0,1] \times [0,1].$$

The domain of interest, $[0,1] \times [0,1]$, was discretized with a 21×21 uniform grid. The solution of the PDE was observed on the grid. Our goal is to recover the function u given the data $y \in \mathbb{R}^{21 \times 21}$. We further assume that u admits a representation

$$u(x) = \sum_{j=1}^{30} u_j \sin(j\pi x) + \sum_{j=1}^{30} \tilde{u}_j \cos(j\pi x), \ x \in [0,1],$$

which reduces the function recovery problem to a parameter estimation problem. Therefore, for each $(x_1, x_2) \in [0,1] \times [0,1]$, the forward map is given by

$$\mathcal{G} : (u_j, \tilde{u}_j)_{j=1}^{30} \in \mathbb{R}^{60} \mapsto v \in \mathbb{R}^{441},$$

which is convex/concave if the function φ is positive/negative. Hence, from the proposition (3.3.9), the objective function given in (3.6) is convex for $r \geq 1$. The following simulations illustrate the effectiveness of the proposed methodology for $r = \frac{1}{3}$ —which amounts to imposing an $\ell_{0.5}$-regularization— despite the possible loss of convexity for $r \in (0,1)$. For the numerical simulation, we set $\varphi(x) = \cos(x)$ and assumed that only three components of each u_j and $\tilde{u}_j$ are nonzero. Specifically, we set

$$u(x) = 1.2\big(\sin(\pi x) + \sin(3\pi x) - \sin(6\pi x) - \cos(3\pi x)\big) - 0.6\big(\cos(\pi x) - \cos(6\pi x)\big).$$

Numerical Results

We compared the performance of IEKF, IEKF-SL, $\ell_{0.5}$-IEKF, $\ell_{0.5}$-IEKF-SL, ℓ_1-IEKF, and ℓ_1-IEKF-SL in terms of their accuracy. For all of our methods, we used $N = 100$ ensemble members with three outer regularization iterations and $T = 20$ inner iterations of ensemble Kalman methods. The initial ensemble members $u_0^{(n)}$ were sampled from a centered Gaussian with covariance matrix $0.04I_{60}$. We compare the reconstructions to the true target function u.

In both Figures 3.4 and 3.5, the top and bottom rows correspond to ℓ_1 and $\ell_{0.5}$-regularization, respectively. The blue curves represent our function recovery based on the corresponding ensemble Kalman method. The shaded regions represent elementwise 2.5/97.5 percentile values of the recovery results. The recovery improves with additional outer iterations. As in the linear example, $\ell_{0.5}$-regularization worked as effectively as, or better than, ℓ_1-regularization. In particular, $\ell_{0.5}$-IEKF-SL recovered the true function almost perfectly after three outer iterations.

	# of outer iteration		
Method	0th	1st	3rd
ℓ_1-IEKF	0.238	0.168	0.094
$\ell_{0.5}$-IEKF	0.238	0.148	0.057
ℓ_1-IEKF-SL	0.205	0.141	0.067
$\ell_{0.5}$-IEKF-SL	0.205	0.132	0.037

Table 3.4.1: ℓ_2-error between parameter estimate and true value.

	# of outer iteration		
Method	0th	1st	3rd
ℓ_1-IEKF	3.678	5.371	4.228
$\ell_{0.5}$-IEKF	3.678	4.945	2.959
ℓ_1-IEKF-SL	4.660	3.649	2.898
$\ell_{0.5}$-IEKF-SL	4.660	3.060	2.400

Table 3.4.2: Average width of credible intervals for recovery.

Table 3.4.1 shows the ℓ_2-norm error between the parameter estimate and the truth. The results demonstrate the effectiveness of sparsity-promoting regularization. In all four methods, the ℓ_2-error decreased with additional outer iterations. In both IEKF and IEKF-SL, $\ell_{0.5}$-regularization produced more accurate (ℓ_2) recovery than ℓ_1-regularization.

Table 3.4.2 contains the average widths of the credible intervals along the number of outer iterations. The widths of the credible intervals tend to decrease as more outer iterations are performed. This is expected since, as regularization effects accumulate, ensembles are more likely to center about their mean and credible bands around the target function become narrower. We can also see the stronger regularizing effect by comparing the width of the credible intervals corresponding to $\ell_{0.5}$-regularization and that of ℓ_1-regularization.

3.4.3 2D-Elliptic Inverse Problem

Finally, following [Lee, 2021] we consider a two-dimensional elliptic inverse problem under a sparsity assumption. We show that our methodology can achieve accurate reconstructions with ℓ_p-regularization with $p = 0.5$, while the approach in [Lee, 2021] could not be implemented with $p < 0.7$.

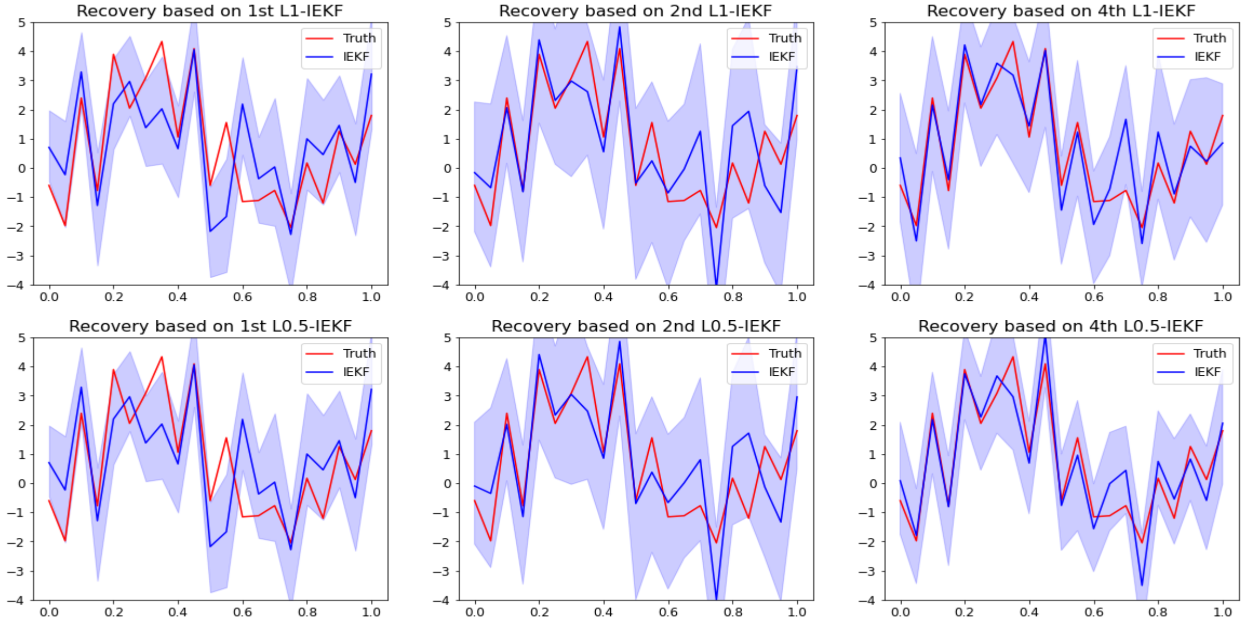


Figure 3.4: Example in Subsection 3.4.2. Red: target function to recover. Blue: ℓ_p-IEKF recovery. Top row: ℓ_1-IEKF. Bottom row: $\ell_{0.5}$-IEKF. Left column: vanilla IEKF. Middle column: ℓ_p-IEKF after one outer iteration. Right column: ℓ_p-IEKF after three outer iterations. Shaded: 2.5/97.5 percentile of the recovery.

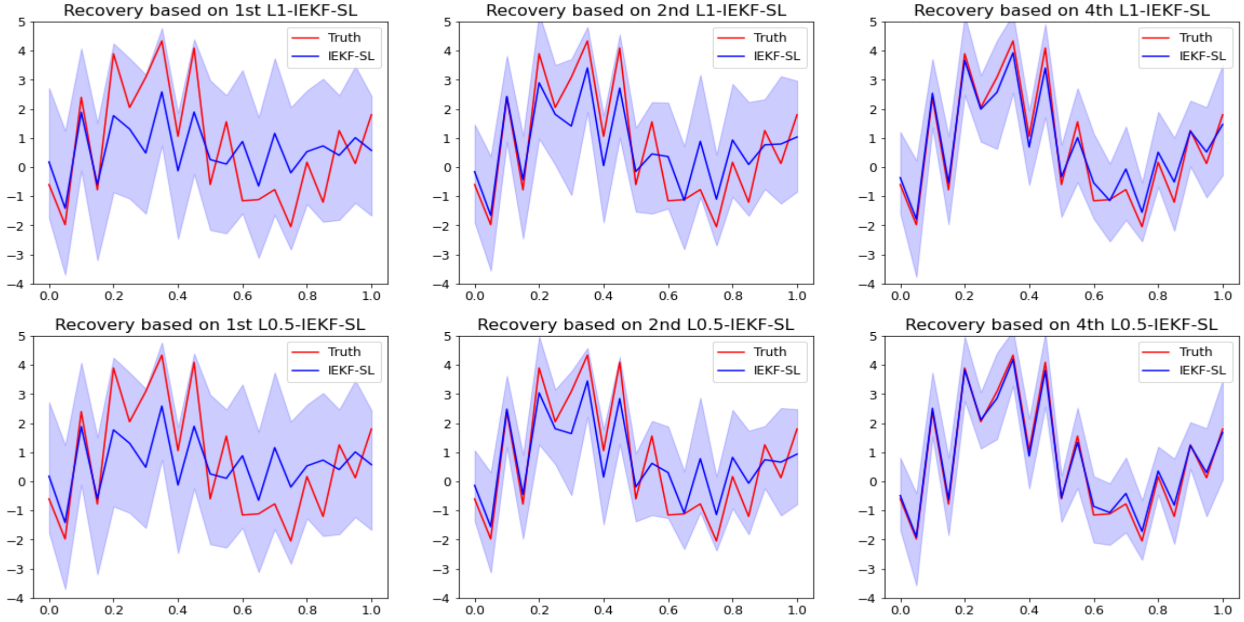


Figure 3.5: Example in Subsection 3.4.2. Red: target function to recover. Blue: ℓ_p-IEKF-SL recovery. Top row: ℓ_1-IEKF-SL. Bottom row: $\ell_{0.5}$-IEKF-SL. Left column: vanilla IEKF-SL. Middle column: ℓ_p-IEKF-SL after one outer iteration. Right column: ℓ_p-IEKF-SL after three outer iterations. Shaded: 2.5/97.5 percentile of the recovery.

Setting

Consider the elliptic PDE

$$-\mathrm{div}\big(e^{u(x)}\nabla v(x)\big) = f(x),\ x = (x_1, x_2) \in [0,1] \times [0,1]$$

with boundary conditions

$$v(x_1, 0) = 100,\ \frac{\partial v}{\partial x_1}(1, x_2) = 0,\ \ -e^{u(x)}\frac{\partial v}{\partial x_1}(0, x_2) = 500,\ \frac{\partial v}{\partial x_2}(x_1, 1) = 0,$$

and source term

$$f(x) = f(x_1, x_2) = \begin{cases} 0 & 0 \leq x_2 \leq \frac{4}{6}, \\ 137 & \frac{4}{6} < x_2 \leq \frac{5}{6}, \\ 274 & \frac{5}{6} < x_2 \leq 1. \end{cases}$$

Following [Lee, 2021], the equation is discretized in a uniform 15×15 grid in $[0,1] \times [0,1]$. We assumed that the log diffusion coefficient can be expressed as

$$u(x_1, x_2) = \sum_{i=0}^{19}\sum_{j=0}^{19} u_{ij}\varphi_{ij}(x_1, x_2),$$

where $\varphi_{ij}(x_1, x_2) = \cos(i\pi x_1)\cos(j\pi x_2)$. Using the first boundary condition, the forward map is given by

$$\mathcal{G} : \{u_{i,j}\}_{i,j=0}^{19} \in \mathbb{R}^{400} \mapsto v \in \mathbb{R}^{14\times 15},$$

which we implemented using the five-point stencil finite-difference method. To ensure sparsity, only six of the 400 components of $\{u_{i,j}\}_{i,j=0}^{19}$ were chosen to be nonzero. We aim to recover $\{u_{i,j}\}_{i,j=0}^{19} \in \mathbb{R}^{400}$ from the data

$$y(x_1, x_2) = \mathcal{G}\big(u(x_1, x_2)\big) + \varepsilon = v(x_1, x_2) + \varepsilon,\ \varepsilon \sim \mathcal{N}(0, 0.1^2).$$

Numerical Results

We compare the performance of IEKF, IEKF-SL, $\ell_{0.5}$-IEKF, $\ell_{0.5}$-IEKF-SL, ℓ_1-IEKF, and ℓ_1-IEKF-SL in terms of their parameter estimation accuracy. For all our methods, we used $N = 400$ ensemble members with six outer regularization iterations and $T = 30$ inner iterations. The initial ensemble $u_0^{(n)}$ was sampled from a centered Gaussian with covariance matrix $0.1 I_{400}$. We present comparisons to the true parameters $\{u_{i,j}\}_{i,j=0}^{19}$.

In Figures 3.6 and 3.7, the top and bottom rows respectively correspond to ℓ_1 and $\ell_{0.5}$-regularization. We also provide approximate credible intervals, constructed from elementwise 2.5/97.5 percentile values of the empirical distribution of the ensemble. In both figures, we observe that estimates improve with more outer iterations. In addition, $\ell_{0.5}$-regularization acted more strongly off the support than ℓ_1-regularization. Within three outer regularization iterations, both $\ell_{0.5}$-IEKF/IEKF-SL yielded parameter estimates very close to the true value.

As in the previous numerical example, Table 3.4.3 provides the ℓ_2-norm errors. In all cases, errors decreased with the additional outer iterations. Note that $\ell_{0.5}$-regularization is an order of magnitude more accurate than ℓ_1-regularization whether using IEKF or IEKF-SL. Table 3.4.4 shows that the length of approximate credible intervals decreased with the number of outer iterations. After only six iterations, credible intervals produced by $\ell_{0.5}$-regularization are an order of magnitude or more smaller than those produced by ℓ_1-regularization.

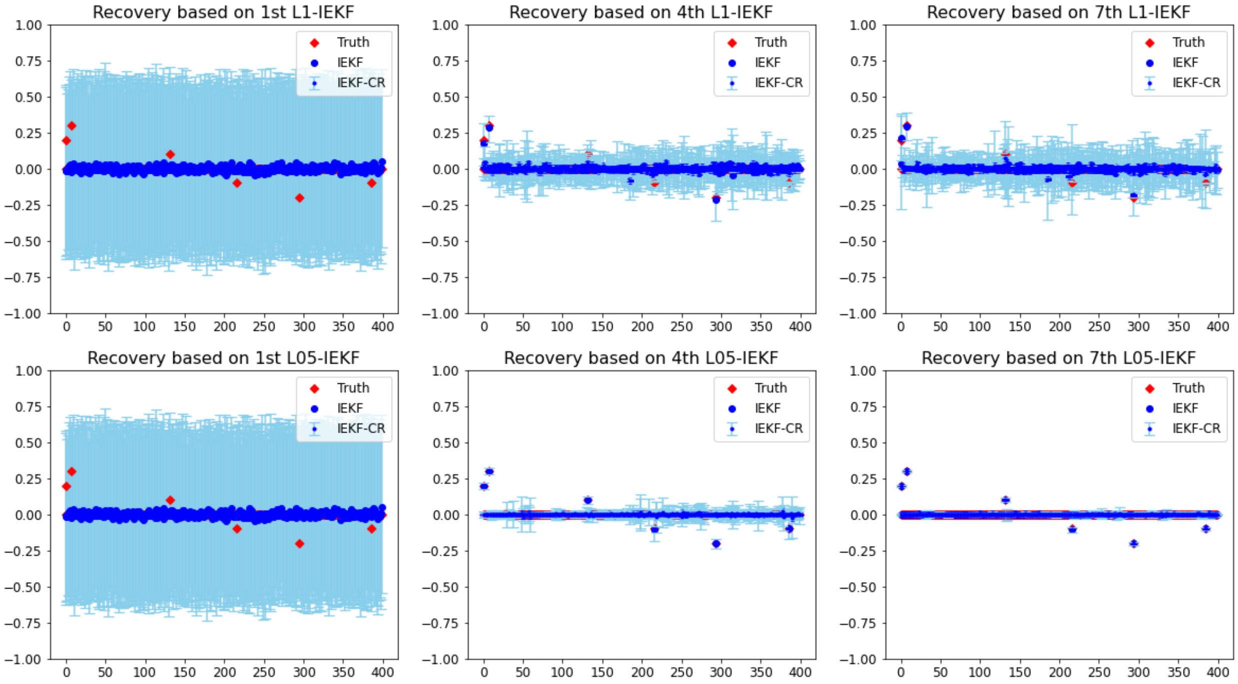


Figure 3.6: Parameter recovery for 2D-elliptic inverse problem based on $\ell_1/\ell_{0.5}$-IEKF. Red: Truth. Blue: ℓ_p-IEKF estimate. Left column: vanilla (non-regularized) IEKF. Middle column: ℓ_p-IEKF after three outer iterations. Right column: ℓ_p-IEKF after six outer iterations. Shaded: elementwise 2.5/97.5 percentile for parameter estimate.

	# of outer iteration		
Method	0th	3rd	6th
ℓ_1-IEKF	0.030	0.014	0.012
$\ell_{0.5}$-IEKF	0.030	0.004	0.002
ℓ_1-IEKF-SL	0.030	0.020	0.014
$\ell_{0.5}$-IEKF-SL	0.030	0.008	0.007

Table 3.4.3: ℓ_2-error between parameter estimate and true value.

	# of outer iteration		
Method	0th	3rd	6th
ℓ_1-IEKF	1.211	0.198	0.211
$\ell_{0.5}$-IEKF	1.211	0.055	0.017
ℓ_1-IEKF-SL	1.399	0.259	0.169
$\ell_{0.5}$-IEKF-SL	1.399	0.041	0.007

Table 3.4.4: Average width of credible intervals for recovery.

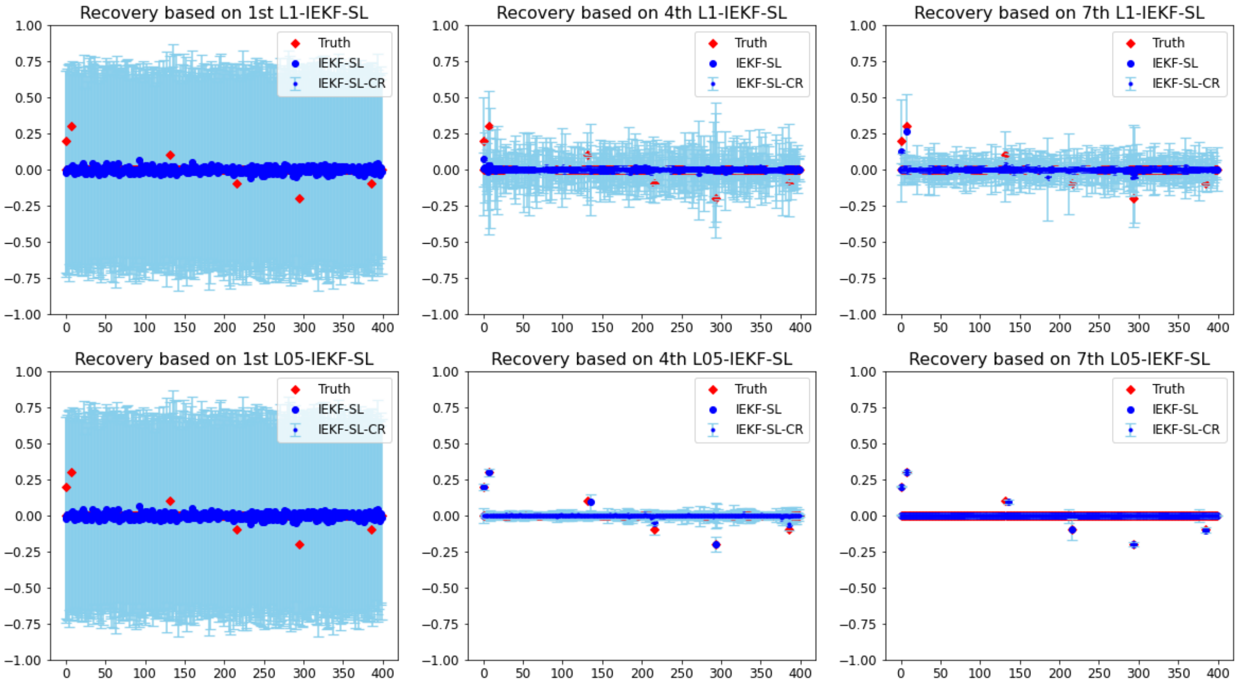


Figure 3.7: Parameter recovery for 2D-elliptic inverse problem based on $\ell_1/\ell_{0.5}$-IEKF-SL. Red: Truth. Blue: ℓ_p-IEKF-SL estimate. Left column: vanilla (non-regularized) IEKF-SL. Middle column: ℓ_p-IEKF-SL after three outer iterations. Right column: ℓ_p-IEKF-SL after six outer iterations. Shaded: elementwise 2.5/97.5 percentile for parameter estimate.

3.5 Conclusion

This chapter introduced a flexible and computationally efficient framework to incorporate a wide range of regularization techniques in ensemble Kalman methods. We have adopted a hierarchical Bayesian perspective to derive our methodology and shown that suitable choices of hyperparameters yield sparsity-promoting regularization. The effectiveness of our procedure was demonstrated in three numerical examples. While we have focused on sparsity-promoting ℓ_p-penalties, our framework extends beyond sparse models. In particular, heavy-tailed Student prior regularization and relaxed ℓ_p-penalties with $r\beta > 3/2$ could be considered for applications in nonlinear regression and in learning dynamical systems from time-averaged data [Schneider et al., 2022]. Finally, this chapter focused on ensemble Kalman methods for inverse problems; future work will investigate regularization of ensemble Kalman filters [Sanz-Alonso et al., 2023, Chen et al., 2022] in data assimilation.

CHAPTER 4

BAYESIAN OPTIMIZATION WITH NOISE-FREE OBSERVATIONS: IMPROVED REGRET BOUNDS VIA RANDOM EXPLORATION

4.1 Introduction

Bayesian optimization [Jones et al., 1998, Mockus, 1998, Frazier, 2018] is an attractive strategy for global optimization of black-box objective functions. Bayesian optimization algorithms sequentially acquire information on the objective by observing its value at carefully selected query points. In some applications, these observations are noisy, but in many others the objective can be noiselessly observed; examples include hyperparameter tuning for machine learning algorithms [Burges and Schölkopf, 1996], parameter estimation for computer models [Clark Jr et al., 2016, Pourmohamad, 2020], goal-driven dynamics learning [Bansal et al., 2017], and alignment of density maps in Wasserstein distance [Singer and Yang, 2023]. While most Bayesian optimization algorithms can be implemented with either noisy or noise-free observations, few methods and theoretical analyses are tailored to the noise-free setting.

This paper introduces two new algorithms rooted in scattered data approximation for Bayesian optimization with noise-free observations. The first algorithm, which we call GP-UCB+, supplements query points obtained via the classical GP-UCB algorithm [Srinivas et al., 2010] with randomly sampled query points. The second algorithm, which we call EXPLOIT+, supplements query points obtained by maximizing the posterior mean of a Gaussian process surrogate model with randomly sampled query points. Both algorithms retain the simplicity and ease of implementation of the GP-UCB algorithm, but introduce an additional random exploration step to ensure that the fill-distance of query points decays at a near-optimal rate, thus enhancing the accuracy of surrogate models for the objective function. The new random exploration step has a profound impact on both theoretical guarantees and empirical performance. On the one hand, the convergence rate of GP-UCB+ and EXPLOIT+ improve upon existing and refined rates for the GP-UCB algorithm. Indeed, the new algorithms nearly achieve the optimal convergence rate in [Bull, 2011] and the decaying rate of simple regret implied by the conjectured cumulative regret bounds in [Vakili, 2022]. On the other hand, GP-UCB+ and EXPLOIT+ explore the state space faster, which leads to an improvement in numerical performance across a range of benchmark and real-world examples.

4.1.1 Main Contributions

- We introduce two new algorithms, GP-UCB+ and EXPLOIT+, whose simple regret convergence rate nearly matches with the optimal minimax convergence rate in [Bull, 2011]. The proposed algorithms far improve existing and refined rates for the classical GP-UCB algorithm. En route to comparing the cumulative regret bounds for our new algorithms with those for GP-UCB, we establish in Theorem 4.3.1 a regret bound for GP-UCB with squared exponential kernels that refines the one in [Lyu et al., 2019].

- We numerically demonstrate that GP-UCB+ and EXPLOIT+ outperform GP-UCB and other popular Bayesian optimization algorithms across many examples, including optimization of several 10-dimensional benchmark objective functions, hyperparameter tuning for random forests, and optimal parameter estimation of a garden sprinkler computer model.

- We showcase that both GP-UCB+ and EXPLOIT+ share the simplicity and ease of implementation of the GP-UCB algorithm. In addition, EXPLOIT+ requires fewer input parameters than GP-UCB or GP-UCB+, and achieves competitive empirical performance without any tuning.

4.1.2 Outline

Section 4.2 formalizes the problem of interest and provides necessary background. We review related work in Section 4.3. Our new algorithms are introduced in Section 4.4, where we establish regret bounds under a deterministic assumption on the objective function. Section 4.5 contains numerical examples, and we close in Section 4.6. Proofs and additional numerical experiments are deferred to an appendix.

4.2 Preliminaries

4.2.1 Problem Statement

We want to find the global maximizer of an objective function $f : \mathcal{X} \to \mathbb{R}$ by leveraging the observed values of f at carefully chosen query points. We are interested in the setting where the observations of the objective are noise-free, i.e. for query points $X_t = \{x_1, \ldots, x_t\}$ we can access observations $F_t = [f(x_1), \ldots, f(x_t)]^\top$. The functional form of f is not assumed to be known. For simplicity, we assume throughout that $\mathcal{X} \subset \mathbb{R}^d$ is a d-dimensional hypercube. We assume that $f \in \mathcal{H}_k(\mathcal{X})$ belongs to the Reproducing Kernel Hilbert Space (RKHS) associated with a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.

4.2.2 Gaussian Processes and Bayesian Optimization

Many Bayesian optimization algorithms, including the ones introduced in this paper, rely on a Gaussian process surrogate model of the objective function to guide the choice of query points. Here, we review the main ideas. Denote generic query locations by $X_t = \{x_1, \ldots, x_t\}$ and the corresponding noise-free observations by $F_t = [f(x_1), \ldots, f(x_t)]^\top$. Gaussian process interpolation with a prior $\mathcal{GP}(0, k)$ yields the following posterior predictive mean and variance:

$$\mu_{t,0}(x) = k_t(x)^\top K_{tt}^{-1} F_t,$$
$$\sigma_{t,0}^2(x) = k(x,x) - k_t(x)^\top K_{tt}^{-1} k_t(x),$$

where $k_t(x) = [k(x,x_1),\dots,k(x,x_t)]^\top$ and K_{tt} is a $t \times t$ matrix with entries $(K_{tt})_{i,j} = k(x_i, x_j)$.

Our interest lies in Bayesian optimization with noise-free observations. However, we recall for later reference that if the observations are noisy and take the form $y_i = f(x_i) + \eta_i$, $1 \leq i \leq t$, where $\eta_i \overset{\text{i.i.d.}}{\sim} N(0,\lambda)$, then the posterior predictive mean and variance are given by

$$\mu_{t,\lambda}(x) = k_t(x)^\top \left(K_{tt} + \lambda I\right)^{-1} Y_t,$$
$$\sigma_{t,\lambda}^2(x) = k(x,x) - k_t(x)^\top (K_{tt} + \lambda I)^{-1} k_t(x),$$

where $Y_t = [y_1,\dots,y_t]^\top$.

To perform Bayesian optimization, one can sequentially select query points by optimizing a Gaussian Process Upper Confidence Bound (GP-UCB) acquisition function. Let $X_{t-1} = \{x_1,\dots,x_{t-1}\}$ denote the query points at the $(t-1)$-th iteration of the algorithm. Then, at the t-th iteration, the classical GP-UCB algorithm [Srinivas et al., 2010] sets

$$x_t = \arg\max_{x \in \mathcal{X}} \mu_{t-1,\lambda}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1,\lambda}(x), \tag{4.1}$$

where β_t is a user-chosen positive parameter. The posterior predictive mean provides a surrogate model for the objective; hence, one expects the maximum of f to be achieved at a point $x \in \mathcal{X}$ where $\mu_{t-1,\lambda}(x)$ is large. However, the surrogate model $\mu_{t-1,\lambda}(x)$ may not be accurate at points $x \in \mathcal{X}$ where $\sigma_{t-1,\lambda}^2(x)$ is large, and selecting query points with large predictive variance helps improve the accuracy of the surrogate model. The GP-UCB algorithm finds a compromise between exploitation (maximizing the mean) and exploration (maximizing the variance). The weight parameter β_t balances this exploitation-exploration trade-off. For later discussion, Algorithm 4.2.1 below summarizes the approach with noise-free observations F_t and $\lambda = 0$.

Algorithm 4.2.1 GP-UCB with noise-free observations.

```
Input: Kernel $k$; Total number of iterations $T$; Initial design points $X_0$; Initial noise-free observations $F_0$. Weights $\{\beta_t\}_{t=1}^{T}$.
Construct $\mu_{0,0}(x)$ and $\sigma_{0,0}(x)$ using $X_0$ and $F_0$.
For $t = 1, \dots, T$ do:
    1. Set
           $x_t = \arg\max_{x \in \mathcal{X}} \mu_{t-1,0}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1,0}(x).$
    2. Set $X_t = X_{t-1} \cup \{x_t\}$ and $F_t = F_{t-1} \cup \{f(x_t)\}$.
    3. Update $\mu_{t,0}(x)$ and $\sigma_{t,0}(x)$ using $X_t$ and $F_t$.
Output: optimization iterates $\{x_1, x_2, \dots, x_T\}$.
```

4.2.3 Performance Metric

The performance of Bayesian optimization algorithms is often analyzed through bounds on their *simple regret*, S_T, given by

$$S_T = f^* - \max_{t=1,\dots,T} f(x_t)$$

or their *cumulative regret*, R_T, given by

$$R_T = \sum_{t=1}^{T} r_t, \qquad r_t = f^* - f(x_t),$$

where f^* is the maximum of the objective f, x_t is the t-th iterate of the optimization algorithm, and r_t is called the *instantaneous regret.* Notice that from the definition, we observe $S_T \leq \frac{R_T}{T}$ and $S_T \leq r_T$. Naturally an upper bound on $\frac{R_T}{T}$ or r_T serves as an upper bound for S_T and the convergence of algorithm is implied by showing $S_T \to 0$ as $T \to \infty$.

Several convergence results have been established directly through bounding the simple regret or the instantaneous regret [Bull, 2011, De Freitas et al., 2012]. On the other hand, the goal of theoretical analyses of optimization algorithms based on the cumulative regret R_T is to show sublinear growth rate of R_T to ensure convergence to the global maximum. A bound on the rate of convergence (with respect to simple regret) is then given by the decaying rate of R_T/T. In this context, R_T serves as a useful intermediate quantity to establish convergence rates for popular Bayesian optimization strategies including algorithms based on upper confidence bounds [Srinivas et al., 2010, Chowdhury and Gopalan, 2017, Bogunovic and Krause, 2021] and Thompson sampling [Chowdhury and Gopalan, 2017]. As noted in [Bull, 2011], a caveat of using R_T to analyze Bayesian optimization algorithms is that the fastest rate of convergence one can obtain is T^{-1}. In addition, R_T accounts for costs that are not incurred by the optimization algorithm. For these reasons, we analyze our new algorithms using simple and instantaneous regret, and additionally compare our simple regret bounds with those implied by existing bounds on R_T.

4.2.4 Choice of Kernel

We will consider the well-specified setting where Gaussian process interpolation for surrogate modeling is implemented using the same kernel k which specifies the deterministic or probabilistic assumptions on f, namely $f \in \mathcal{H}_k$ or $f \sim \mathcal{GP}(0, k)$. The impact of kernel misspecification on Bayesian optimization algorithms is studied in [Bogunovic and Krause, 2021, Kim et al., 2024].

For concreteness, we focus on *Matérn kernels* with smoothness parameter ν and length-scale parameter ℓ, given by

$$k(x, x') = \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\frac{\|x - x'\|}{\ell} \right)^{\nu} B_{\nu} \left(\frac{\|x - x'\|}{\ell} \right),$$

where B_ν is a modified Bessel function of the second kind, and on *squared exponential kernels* with lengthscale parameter ℓ, given by

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right).$$

We recall that the Matérn kernel converges to the squared exponential kernel in the large ν asymptotic. Both types of kernel are widely used in practice, and we refer to [Williams and Rasmussen, 2006, Wendland, 2004, Stein, 2012] for further background.

4.3 Related Work

4.3.1 Existing Regret Bounds: Noisy Observations

Numerous works have established cumulative regret bounds for Bayesian optimization with noisy observations under both deterministic assumption on the objective function [Srinivas et al., 2010, Chowdhury and Gopalan, 2017, Vakili et al., 2021, Bogunovic and Krause, 2021, Russo and Van Roy, 2014, Kandasamy et al., 2018]. These bounds involve a quantity known as the *maximum information gain*, which under a Gaussian noise assumption is given by $\gamma_t = \frac{1}{2}\log|I + \lambda^{-1}K_t t|$, where $\lambda > 0$ represents the noise level. In particular, under a deterministic objective function assumption, [Chowdhury and Gopalan, 2017] showed a cumulative regret bound for GP-UCB of the form $\mathcal{O}\left(\gamma_T\sqrt{T}\right)$, which improves the one obtained in [Srinivas et al., 2010] by a factor of $\mathcal{O}\big(\log^{3/2}(T)\big)$. By tightening existing upper bounds on the maximum information gain, [Vakili et al., 2021] established a cumulative regret bound for GP-UCB of the form

$$R_T = \begin{cases} \mathcal{O}\left(T^{\frac{2\nu+3d}{4\nu+2d}} \log^{\frac{2\nu}{2\nu+d}} T\right), \\ \mathcal{O}\left(T^{\frac{1}{2}} \log^{d+1} T\right), \end{cases}$$

for Matérn and squared exponential kernels.

4.3.2 Existing Regret Bounds: Noise-Free Observations

In contrast to the noisy setting, few works have obtained regret bounds with noise-free observations. With an expected improvement acquisition function and Matérn kernel, [Bull, 2011] provided a simple regret bound of the form $\tilde{\mathcal{O}}\left(T^{-\min\{\nu,1\}/d}\right)$, where $\tilde{\mathcal{O}}$ suppresses logarithmic factors, under deterministic objective function assumption. On the other hand, [De Freitas et al., 2012] introduced a branch and bound algorithm that achieves an exponential rate of convergence for the instantaneous regret, under the probabilistic assumption on the objective function. However, unlike the standard GP-UCB algorithm, the algorithm in [De Freitas et al., 2012] requires many observations in each iteration to reduce the search space, and it further requires solving a constrained optimization problem in the reduced search space.

To the best of our knowledge, [Lyu et al., 2019] presents the only cumulative regret bound available for GP-UCB with noise-free observations under a deterministic assumption on the objective. Specifically, they consider Algorithm 4.2.1, and, noticing that $\sigma_{t,0}(x) \leq \sigma_{t,\lambda}(x)$ for any $\lambda \geq 0$, they deduce that existing cumulative regret bounds for Bayesian optimization with noisy observations remain valid with noise-free observations. Furthermore, in the noise-free setting, the cumulative regret bound is improved by a factor of $\sqrt{\gamma_T}$, which comes from using a constant weight parameter $\beta_t := \|f\|^2_{\mathcal{H}_k(\mathcal{X})}$ given by the squared RKHS norm of the objective. This leads to a cumulative regret bound with rate $\mathcal{O}(\sqrt{\gamma_T T})$, which gives

$$R_T = \begin{cases} \mathcal{O}\left(T^{\frac{\nu+d}{2\nu+d}} \log^{\frac{\nu}{2\nu+d}} T\right), \\ \mathcal{O}\left(T^{\frac{1}{2}} \log^{\frac{d+1}{2}} T\right), \end{cases} \tag{4.2}$$

for Matérn and squared exponential kernels.

4.3.3 Tighter Cumulative Regret Bound for Squared Exponential Kernels

[Vakili, 2022] sets as an open problem whether one can improve the cumulative regret bounds in (4.2) for the GP-UCB algorithm with noise-free observations. For squared exponential kernels, we claim that one can further improve the cumulative regret bound in (4.2) by a factor of $\sqrt{\log T}$.

Theorem 4.3.1. *Let $f \in \mathcal{H}_k(\mathcal{X})$, where k is a squared exponential kernel. GP-UCB with noise-free observations and $\beta_t := \|f\|^2_{\mathcal{H}_k}$ satisfies the cumulative regret bound*

$$R_T = \mathcal{O}\left(T^{\frac{1}{2}} \log^{\frac{d}{2}} T\right),$$

which yields the convergence rate of

$$S_T = \mathcal{O}\left(T^{-\frac{1}{2}} \log^{\frac{d}{2}} T\right).$$

Remark 4.3.2. Our improvement in the bound comes from a constant term $\frac{1}{\log(1+\lambda^{-1})}$, which was ignored in existing analyses with noisy observations. By letting $\lambda \to 0$, the constant offsets a $\sqrt{\log T}$ growth in the cumulative regret bound.

Remark 4.3.3. For Matérn kernels, a similar approach to improve the rate is not feasible. A state-of-the-art, near-optimal upper bound on the maximum information gain with Matérn kernels obtained in [Vakili et al., 2021] introduces a polynomial growth factor as the noise variance λ decreases to zero. Minimizing the rate of an upper bound in [Vakili et al., 2021] one can match the rate obtained in [Lyu et al., 2019].

Remark 4.3.4. Unlike cumulative regret bounds with noisy observations, Theorem 4.3.1 and the results in [Lyu et al., 2019] are deterministic.

4.3.4 Optimal Simple Regret Bounds

Under the deterministic objective function assumption, Theorem 4.3.1 refines the rate bound in (4.2) for GP-UCB with noise-free observations using squared exponential kernels. In the rest of the paper, we will design new algorithms that achieve drastically faster convergence rates. In particular, for Matérn kernels, our algorithms nearly achieve the optimal convergence rate in [Bull, 2011] of the form $S_T = \Theta\left(T^{-\frac{\nu}{d}}\right)$. Furthermore, our algorithms' convergence rate is competitive to algorithms that satisfy the conjectured cumulative regret bound in [Vakili, 2022] of the form

$$R_T = \begin{cases} \mathcal{O}(T^{\frac{d-\nu}{d}}), & \text{for } d > \nu \\ \mathcal{O}(\log T), & \text{for } d = \nu \\ \mathcal{O}(1), & \text{for } d < \nu \end{cases}$$

which translates to the convergence rate of

$$S_T = \begin{cases} \mathcal{O}\left(T^{-\frac{\nu}{d}}\right), & \text{for } d > \nu \\ \mathcal{O}\left(T^{-1}\log T\right), & \text{for } d = \nu \\ \mathcal{O}\left(T^{-1}\right), & \text{for } d < \nu. \end{cases}$$

Our new algorithms nearly achieve the optimal convergence rate while preserving the ease of implementation of GP-UCB algorithms. The recent preprint [Salgia et al., 2023] proposes an alternative batch-based approach, which combines random sampling with domain shrinking to attain the conjectured cumulative regret bounds for $d \geq \nu$ with a high probability.

4.4 Exploitation with Accelerated Exploration

4.4.1 How Well Does GP-UCB Explore?

The GP-UCB algorithm selects query points by optimizing an acquisition function which incorporates the posterior mean to promote exploitation and the posterior standard deviation to promote exploration. Our new algorithms are inspired by the desire to improve the exploration of GP-UCB. Before introducing the algorithms in the next subsection, we heuristically explain why such an improvement may be possible.

A natural way to quantify how well data $X_t = \{x_1, \ldots, x_t\}$ cover the search space $\mathcal{X}$ is via the *fill-distance*, given by

$$h(\mathcal{X}, X_t) \coloneqq \sup_{x \in \mathcal{X}} \inf_{x_i \in X_t} \|x - x_i\|.$$

The fill-distance appears in error bounds for Gaussian process interpolation and regression [Wendland, 2004, Teckentrup, 2020, Stuart and Teckentrup, 2018, Tuo and Wang, 2020]. For quasi-uniform points, it holds that $h(\mathcal{X}, X_t) = \Theta\left(t^{-\frac{1}{d}}\right)$, which is the fastest possible decay rate for any sequence of design points. The fill-distance of the query points selected by our new Bayesian optimization algorithms will (nearly) decay at this rate.

[Wenzel et al., 2021] introduced a stabilized greedy algorithm to obtain query points by maximizing the posterior predictive standard deviation at each iteration. Their algorithm sequentially generates a set of query points whose fill-distance decays at a rate $\Theta\left(t^{-\frac{1}{d}}\right)$ by sequentially solving constrained optimization problems, which can be computationally demanding. Since GP-UCB simply promotes exploration through the posterior predictive standard deviation term in the UCB acquisition function, one may heuristically expect the fill-distance of query points selected by the standard GP-UCB algorithm to decay at a slower rate. On the other hand, a straightforward online approach to obtain a set of query points whose fill-distance nearly decays at a rate $\Theta\left(t^{-\frac{1}{d}}\right)$ is to sample randomly from a probability

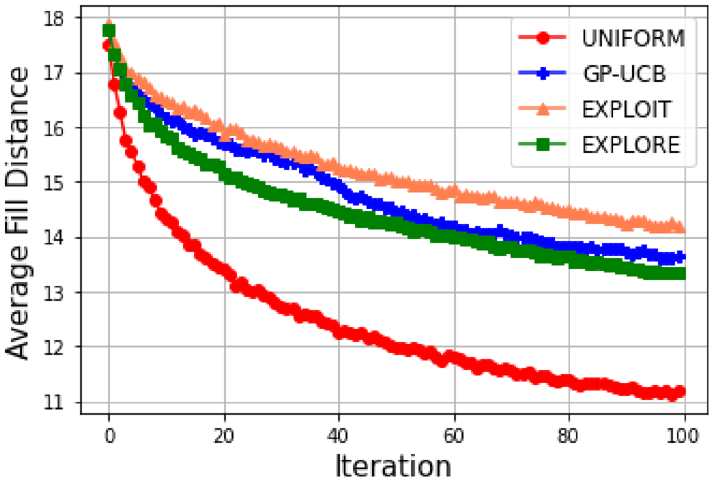


Figure 4.1: Average fill-distance of a set of query points obtained using four different algorithms over 100 independent experiments. The results are based on a 10-dimensional Rastrigin function. The discrete subset $\mathcal{X}_D$ consists of 100 Latin hypercube samples.

measure P with a strictly positive Lebesgue density on $\mathcal{X}$. Specifically, [Oates et al., 2019] shows that, in expectation, the fill-distance of independent samples from such a measure decays at a near-optimal rate: for any $\epsilon > 0$, $\mathbb{E}_P\big[h(\mathcal{X}, X_t)\big] = \mathcal{O}(t^{-\frac{1}{d}+\epsilon})$.

Figure 4.1 compares the decay of the fill-distance for query points selected using four strategies. For a 10-dimensional Rastrigin function, we consider: (i) GP-UCB; (ii) EXPLOIT (i.e., maximizing the posterior mean at each iteration); (iii) EXPLORE (i.e., maximizing the posterior variance); and (iv) UNIFORM (i.e. independent uniform random samples on $\mathcal{X}$). The results were averaged over 100 independent experiments. The fill-distance for GP-UCB lies in between those for EXPLORE and EXPLOIT; whether it lies closer to one or the other depends on the choice of weight parameter, which here we choose based on a numerical approximation of the max-norm of the objective, $\beta_t^{\frac{1}{2}} = \max_{x \in \mathcal{X}_D} |f(x)|$ where $\mathcal{X}_D$ is a discretization of the search space $\mathcal{X}$. Note that UNIFORM yields a drastically smaller fill-distance even when compared with EXPLORE. Our new algorithms will leverage random sampling to enhance exploration in Bayesian optimization and achieve improved regret bounds.

4.4.2 Improved Exploration via Random Sampling

In this subsection, we introduce two Bayesian optimization algorithms that leverage random sampling as a tool to facilitate efficient exploration of the search space and enhance the accuracy of surrogate models of the objective function with which to acquire new optimization iterates. While the GP-UCB algorithm selects a single query point x_t per iteration, our algorithms select two query points $\{x_t, \tilde{x}_t\}$ to produce a single optimization iterate x_t.

The first algorithm we introduce, which we call GP-UCB+, selects a query point x_t using the GP-UCB acquisition function and another query point $\tilde{x}_t$ by random sampling. We outline the pseudocode in Algorithm 4.4.1. The second algorithm we introduce, which we call EXPLOIT+, decouples the exploitation and exploration goals, selecting one query point x_t by maximizing the posterior mean to promote exploitation, and another query point $\tilde{x}_t$ by random sampling to promote exploration. We outline the pseudocode in Algorithm 4.4.2. As noted above, both GP-UCB+ and EXPLOIT+ produce a single optimization iterate x_t; the role of the additional query point $\tilde{x}_t$ is to enhance the surrogate model of the objective with which x_t is acquired. Since our new algorithms require two query points per iteration, in our numerical experiments in Section 4.5 we ensure a fair comparison by running them for has as many iterations as used for algorithms that require one query point per iteration. For the convergence rate analysis in Subsection 4.4.3, the fact that our algorithms require twice as many iterations is inconsequential, since halving the number of iterations does not affect the convergence rate.

Notably, EXPLOIT+ does not require input weight parameters $\{\beta_t\}_{t=1}^T$. As mentioned in Section 4.3, many regret bounds for GP-UCB algorithms rely on choosing the weight parameters as the squared RKHS norm of the objective or in terms of a bound on it. The performance of GP-UCB and GP-UCB+ can be sensitive to this choice, which in practice is often based on empirical tuning or heuristic arguments rather than guided by the theory. In contrast, EXPLOIT+ achieves the same regret bounds as GP-UCB+ and drastically

Algorithm 4.4.1 GP-UCB+.

```
1: Input: Kernel $k$; Total number of iterations $T$; Initial design points $X_0^{\text{full}}$; Initial noise-free observations $F_0^{\text{full}}$; Probability distribution $P$ on $\mathcal{X}$. Weights $\{\beta_t\}_{t=1}^T$.
2: Construct posterior mean $\mu_{0,0}^{\text{full}}(x)$ and standard deviation $\sigma_{0,0}^{\text{full}}(x)$ using $X_0^{\text{full}}$ and $F_0^{\text{full}}$.
3: For $t = 1, \ldots, T$ do:
    1. Exploitation + Exploration: Set
       $$x_t = \arg\max_{x \in \mathcal{X}} \mu_{t-1,0}^{\text{full}}(x) + \beta_t^{\frac{1}{2}} \sigma_{t-1,0}^{\text{full}}(x).$$
    2. Exploration: Sample $\tilde{x}_t \sim P$.
    3. Set
       $$X_t^{\text{full}} = X_{t-1}^{\text{full}} \cup \{x_t, \tilde{x}_t\},$$
       $$F_t^{\text{full}} = F_{t-1}^{\text{full}} \cup \{f(x_t), f(\tilde{x}_t)\}.$$
    4. Update $\mu_{t,0}^{\text{full}}(x)$ and $\sigma_{t,0}^{\text{full}}(x)$ using $X_t^{\text{full}}$ and $F_t^{\text{full}}$.
4: Output: optimization iterates $\{x_1, x_2, \ldots, x_T\}$.
```

Algorithm 4.4.2 EXPLOIT+.

```
1: Input: Kernel $k$; Total number of iterations $T$; Initial design points $X_0^{\text{full}}$; Initial noise-free observations $F_0^{\text{full}}$; Probability distribution $P$ on $\mathcal{X}$.
2: Construct posterior mean $\mu_{0,0}^{\text{full}}(x)$ and standard deviation $\sigma_{0,0}^{\text{full}}(x)$ using $X_0^{\text{full}}$ and $F_0^{\text{full}}$.
3: For $t = 1, \ldots, T$ do:
    1. Exploitation: Set $x_t = \arg\max_{x \in \mathcal{X}} \mu_{t-1,0}^{\text{full}}(x)$.
    2. Exploration: Sample $\tilde{x}_t \sim P$.
    3. Set
       $$X_t^{\text{full}} = X_{t-1}^{\text{full}} \cup \{x_t, \tilde{x}_t\},$$
       $$F_t^{\text{full}} = F_{t-1}^{\text{full}} \cup \{f(x_t), f(\tilde{x}_t)\}.$$
    4. Update $\mu_{t,0}^{\text{full}}(x)$ using $X_t^{\text{full}}$ and $F_t^{\text{full}}$.
4: Output: optimization iterates $\{x_1, x_2, \ldots, x_T\}$.
```

faster rates than GP-UCB without requiring the practitioner to specify weight parameters. Additionally, EXPLORE+ shows competitive empirical performance.

Remark 4.4.1. In the exploration step, one can acquire a batch of points to further enhance the exploration of GP-UCB+ and EXPLOIT+. As long as the number of points sampled at each iteration does not grow with respect to the iteration index t, the regret bounds stated in Theorem 4.4.4 below remain valid.

Remark 4.4.2. A common heuristic strategy to expedite the performance of Bayesian optimization algorithms is to acquire a moderate number of initial design points by uniformly sampling the search space. Since the order of the exploration and exploitation steps can be swapped in our algorithms, such heuristic strategy can be interpreted as an initial batch exploration step.

Remark 4.4.3. A natural choice for P is the uniform distribution on the search space $\mathcal{X}$. Our theory, which utilizes bounds on the fill-distance of randomly sampled query points from [Oates et al., 2019], holds as long as P has a strictly positive Lebesgue density on $\mathcal{X}$. In what follows, we assume throughout that P satisfies this condition.

4.4.3 Regret Bounds

We now obtain regret bounds under the deterministic assumption that f belongs to the RKHS of a kernel k. Our algorithms are random due to sampling from P, and we show cumulative regret bounds in expectation with respect to such randomness.

Theorem 4.4.4. *Let $f \in \mathcal{H}_k(\mathcal{X})$. Suppose $t \in \mathbb{N}$ is large enough. GP-UCB+ with $\beta_t := \|f\|^2_{\mathcal{H}_k(\mathcal{X})}$ and EXPLOIT+ attain the following instantaneous regret bounds. For Matérn kernels with parameter $\nu > 0$,*

$$\mathbb{E}_P[r_t] = \mathcal{O}\left(t^{-\frac{\nu}{d}+\varepsilon}\right)$$

where $\varepsilon > 0$ can be arbitrarily small. For squared exponential kernels,

$$\mathbb{E}_P[r_t] = \mathcal{O}\left(\exp\left(-Ct^{\frac{1}{d}-\varepsilon}\right)\right),$$

for some constant $C > 0$ with an arbitrarily small $\varepsilon > 0$.

Remark 4.4.5. In particular, these results imply that

$$\mathbb{E}_P[S_T] = \begin{cases} \mathcal{O}\left(T^{-\frac{\nu}{d}+\varepsilon}\right), & \text{for Matérn kernel} \\ \mathcal{O}\left(\exp\left(-CT^{\frac{1}{d}-\varepsilon}\right)\right), & \text{for square exponential kernel} \end{cases}$$

Remark 4.4.6. For Matérn kernels, in expectation, the proposed algorithms nearly attain the optimal convergence rate established by [Bull, 2011] and the convergence rate implied by the cumulative regret bound conjectured in [Vakili, 2022]. Moreover, one can further obtain the exact optimal convergence rate by replacing the random sampling step in GP-UCB+ and EXPLOIT+ with a more computationally expensive quasi-uniform sampling scheme.

Remark 4.4.7. Compared with the GP-UCB algorithm with noise-free observations, the proposed algorithms attain improved convergence rates in expectation for both Matérn and squared exponential kernels. For the Matérn kernel, the new convergence rate has a faster polynomial decaying factor with a removal of the logarithmic growth factor. For the squared exponential kernel, the proposed algorithms have an exponential convergence rate, whereas the improved bound for the GP-UCB algorithm in Theorem 4.3.1 has a convergence rate of $\mathcal{O}(T^{-\frac{1}{2}}\log^{\frac{d}{2}}(T))$.

Remark 4.4.8. For Matérn kernels, compared with the recent preprint [Salgia et al., 2023], which attains $\mathcal{O}(T^{-1}\log^{\frac{3}{2}} T)$ convergence rate when $d < \nu$, our algorithms attains exponential convergence rate. When $d \geq \nu$, [Salgia et al., 2023] attains the convergence rate implied by the conjecture in [Vakili, 2022] up to a logarithmic factor, while we attain the implied convergence rate up to a factor of $\mathcal{O}(T^{\varepsilon})$, for arbitrarily small $\varepsilon > 0$. Compared to existing works, we additionally establish exponential convergence rates with the squared exponential kernel.

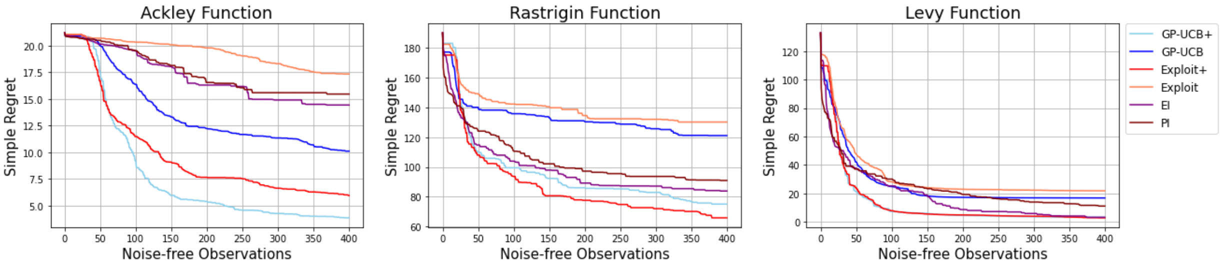


Figure 4.2: Simple regret vs number of noise-free observations.

4.5 Numerical Experiments

This section explores the empirical performance of our methods on three benchmark objective functions, on hyperparameter tuning for a machine learning model, and on optimizing a black-box objective function designed to guide engineering decisions. We compare the new algorithms (GP-UCB+, EXPLOIT+) with GP-UCB and two other popular Bayesian optimization strategies: Expected Improvement (EI) and Probability of Improvement (PI). We also compare with the EXPLOIT approach outlined in Subsection 4.4.1, but not with EXPLORE as this method did not achieve competitive performance. Throughout, we choose the distribution P which governs random exploration in the new algorithms to be uniform on $\mathcal{X}$. For the weight parameter of UCB acquisition functions, we considered a well-tuned constant value $\beta_t^{1/2} = 2$ that achieves good performance in our examples, and the approach in [Chowdhury and Gopalan, 2019], which sets $\beta_t^{1/2} = \max_{x \in \mathcal{X}_D} |f(x)|$ where $\mathcal{X}_D$ is a discretization of the search space. All the hyperparameters of the kernel function were iteratively updated through maximum likelihood estimation. Since the new algorithms need two noise-free observations per iteration but the methods we compare with only need one, we run the new algorithms for half as many iterations to ensure a fair comparison.

4.5.1 Benchmark Objective Functions

We consider three 10-dimensional benchmark objective functions: Ackley, Rastrigin, and Levy. Each of them has a unique global maximizer but many local optima, posing a challenge to standard first and second-order convex optimization algorithms. Following the virtual library of simulation experiments: `https://www.sfu.ca/ ssurjano/`, we respectively set the search space to be $[-32.768, 32.768]^{10}, [-5.12, 5.12]^{10}$, and $[-10, 10]^{10}$. We used a Matérn kernel with the default initial smoothness parameter $\nu = 2.5$ and initial length scale parameter $\ell = 1$. For each method and objective, we obtain 400 noise-free observations and average the results over 20 independent experiments. For GP-UCB and GP-UCB+, we set $\beta_t^{1/2} = 2$. Figure 4.2 shows the average simple regrets, given by $f^* - \max_{t=1,\dots,T} f(x_t)$. We report the regret as a function of the number of observations rather than the number of iterations to ensure a fair comparison. For all three benchmark functions, GP-UCB+ and EXPLOIT+ outperform the other methods. To further demonstrate the strength of the proposed algorithms, Table 4.5.1 shows the average simple regret at the last iteration, normalized so that for each benchmark objective the worst-performing algorithm has unit simple regret. Table 4.7.1 in Appendix 4.7.3 shows results for the standard deviation, indicating that the new methods are not only more accurate, but also more precise.

To illustrate the sensitivity of UCB algorithms to the choice of weight parameters, we include numerical results with $\beta_t^{1/2} = \max_{x \in \mathcal{X}_D} |f(x)|$ in Appendix 4.7.3. In particular, since GP-UCB+ has an additional exploration step through random sampling, using a smaller weight parameter for GP-UCB+ than for GP-UCB tends to work more effectively. Remarkably, the parameter-free EXPLOIT+ algorithm achieves competitive performance compared with UCB algorithms with well-tuned weight parameters.

Table 4.5.1: Normalized average simple regret with 400 function evaluations for benchmark objectives in dimension $d = 10$.

METHOD	ACKLEY	RASTRIGIN	LEVY
GP-UCB+	**0.222**	0.576	0.146
GP-UCB	0.583	0.930	0.768
EXPLOIT+	0.342	**0.505**	**0.126**
EXPLOIT	1.000	1.000	1.000
EI	0.832	0.644	0.142
PI	0.891	0.698	0.507

4.5.2 Random Forest Hyperparameter Tuning

Here we use Bayesian optimization to tune four hyperparameters of a random forest regression model for the California housing dataset [Pace and Barry, 1997]. The parameters of interest are (i) three integer-valued quantities: the number of trees in the forest, the maximum depth of the tree, and the minimum number of samples required to split the internal node; and (ii) a real-valued quantity between zero and one: the transformed maximum number of features to consider when looking for the best split. For the discrete quantities, instead of optimizing over a discrete search space, we performed the optimization over a continuous domain and truncated the decimal values when evaluating the objective function. We split the dataset into training (80%) and testing (20%) sets. To define a deterministic objective function, we fixed the random state parameter for the `RandomForestRegressor` function from the Python scikit-learn package and built the model using the training set. We then defined our objective function to be the negative mean-squared test error of the built model. We used a Matérn kernel with initial smoothness parameter $\nu = 2.5$ and initial lengthscale parameter $\ell = 1$. For the GP-UCB and GP-UCB+ algorithms, we set $\beta_t^{1/2} = \max_{x \in \mathcal{X}_D} |f(x)|$ where $\mathcal{X}_D$ consists of 40 Latin hypercube samples. We conducted 20 independent experiments with 80 noise-free observations. From Table 4.5.2 and Figure 4.3, we see that both GP-UCB+ and EXPLOIT+ algorithms led to smaller cumulative test errors. An instantaneous test error plot with implementation details can be found in Appendix 4.7.3.

Table 4.5.2: Cumulative test error averaged over 20 experiments.

METHOD	MEAN ± SD
GP-UCB+	28.552 ± 1.971
GP-UCB	35.226 ± 1.467
EXPLOIT+	$\mathbf{26.346} \pm \mathbf{1.404}$
EXPLOIT	35.026 ± 1.092
EI	34.294 ± 0.987
PI	33.438 ± 1.085

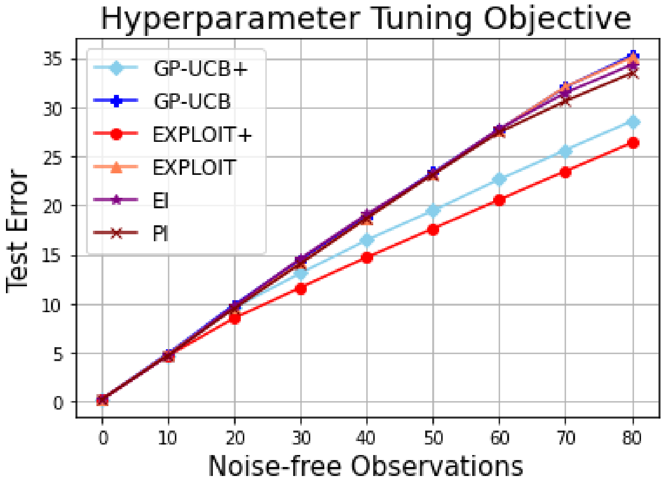


Figure 4.3: Cumulative test error vs number of noise-free observations.

4.5.3 Garden Sprinkler Computer Model

The Garden Sprinkler computer model simulates the range of a garden sprinkler that sprays water. The model contains eight physical parameters that represent vertical nozzle angle, tangential nozzle angle, nozzle profile, diameter of the sprinkler head, dynamic friction moment, static friction moment, entrance pressure, and diameter flow line. First introduced in [Siebertz et al., 2010] and later formulated into a deterministic black-box model by [Pourmohamad, 2020], the goal is to maximize the accessible range of a garden sprinkler over the domain of the eight-dimensional parameter space. In this problem, the observations of the objective are noise-free. Following [Pourmohamad and Lee, 2021], for GP-UCB and GP-UCB+ we set $\beta_t^{1/2} = 2$ and used a squared exponential kernel with an initial lengthscale parameter $\ell^2 = 50$. We ran 30 independent experiments, each with 100 noise-free observations. The results in Table 4.5.3 and Figure 4.4 demonstrate that the new algorithms achieve competitive performance. In particular, EXPLOIT+ attains on average the largest maximum value, while also retaining a moderate standard deviation across experiments.

Table 4.5.3: Maximum attained value of the garden sprinkler objective function averaged over 30 experiments.

Method	Mean ± SD
GP-UCB+	17.511 ± 1.603
GP-UCB	18.038 ± 2.026
EXPLOIT+	**18.427 ± 1.825**
EXPLOIT	17.352 ± 2.537
EI	18.061 ± 1.657
PI	17.105 ± 2.329

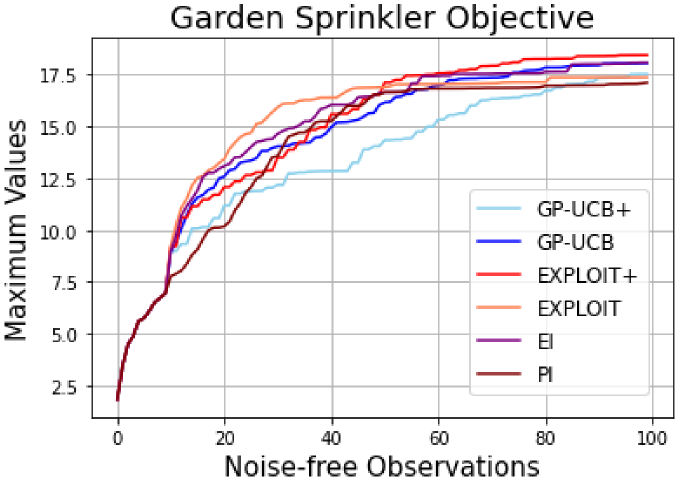


Figure 4.4: Maximum attained value of the garden sprinkler objective function vs number of noise-free observations.

4.6 Conclusion

This chapter has introduced two Bayesian optimization algorithms, GP-UCB+ and EXPLOIT+, that supplement query points obtained via UCB or posterior mean maximization with query points obtained via random sampling. The additional sampling step in our algorithms promotes search space exploration and ensures that the fill-distance of the query points decays at a nearly optimal rate. From a theoretical viewpoint, we have shown that GP-UCB+ and EXPLOIT+ satisfy near-optimal convergence rate that improve upon existing and refined rates for the classical GP-UCB algorithm with noise-free observations. Indeed, at the price of a higher computational cost, one can obtain the optimal convergence rate from [Bull, 2011] as well as the convergence rate implied by the conjectured cumulative regret bound in [Vakili, 2022] by replacing the random sampling step in GP-UCB+ and EXPLOIT+ with a quasi-uniform sampling scheme. From an implementation viewpoint, both GP-UCB+ and EXPLOIT+ retain the appealing simplicity of the GP-UCB algorithm; moreover, EXPLOIT+ does not require specifying input weight parameters. From an empirical viewpoint, we have demonstrated that the new algorithms outperform existing ones in a wide range of examples.

4.7 Appendix

4.7.1 Proof of Theorem 4.3.1

Let $f^* = f(x^*) = \max_{x \in \mathcal{X}} f(x)$ and let $r_t = f^* - f(x_t)$ be the instantaneous regret. Then,

$$\begin{aligned} r_t &= f^* - \mu_{t-1,0}(x^*) + \mu_{t-1,0}(x^*) - \mu_{t-1,0}(x_t) + \mu_{t-1,0}(x_t) - f(x_t) \\ &\overset{\text{(i)}}{\leq} \|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x^*) + \mu_{t-1,0}(x^*) - \mu_{t-1,0}(x_t) + \|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x_t) \\ &\overset{\text{(ii)}}{\leq} \|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x_t) + \mu_{t-1,0}(x_t) - \mu_{t-1,0}(x_t) + \|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x_t) \\ &= 2\|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x_t), \end{aligned} \tag{4.3}$$

where for (i) we use twice that, for any $x \in \mathcal{X}$, it holds that $|f(x) - \mu_{t-1,0}(x)| \leq \|f\|_{\mathcal{H}_k(\mathcal{X})} \sigma_{t-1,0}(x)$ —see for instance Corollary 3.11 in [Kanagawa et al., 2018]— and for (ii) we use the definition of x_t in the GP-UCB algorithm. Thus, for any $\lambda > 0$,

$$R_T^2 \overset{\text{(i)}}{\leq} T \sum_{t=1}^{T} r_t^2 \overset{\text{(ii)}}{\leq} 4T \|f\|^2_{\mathcal{H}_k(\mathcal{X})} \sum_{t=1}^{T} \sigma^2_{t-1,0}(x_t) \overset{\text{(iii)}}{\leq} 4T \|f\|^2_{\mathcal{H}_k(\mathcal{X})} \sum_{t=1}^{T} \sigma^2_{t-1,\lambda}(x_t),$$

where (i) follows by the Cauchy-Schwarz inequality, (ii) from the bound on r_t, and (iii) from the fact that $\sigma_{t-1,0}(x_t) \leq \sigma_{t-1,\lambda}(x_t)$ for any $\lambda > 0$. Since the function $\frac{x}{\log(1+x)}$ is strictly increasing in x and for the squared exponential kernel it holds that $\lambda^{-1}\sigma^2_{t-1,\lambda}(x_t) \leq \lambda^{-1}$, we have that $\lambda^{-1}\sigma^2_{t-1,\lambda}(x_t) \leq \frac{\lambda^{-1}}{\log(1+\lambda^{-1})} \log\bigl(1 + \lambda^{-1}\sigma^2_{t-1,\lambda}(x_t)\bigr)$. Therefore,

$$R_T^2 \leq \frac{8T\|f\|^2_{\mathcal{H}_k(\mathcal{X})}}{\log(1+\lambda^{-1})} \left(\frac{1}{2} \sum_{t=1}^{T} \log\Bigl(1 + \lambda^{-1}\sigma^2_{t-1,\lambda}(x_t)\Bigr) \right) \leq \frac{8T\|f\|^2_{\mathcal{H}_k(\mathcal{X})}}{\log(1+\lambda^{-1})} \gamma_{T,\lambda}, \tag{4.4}$$

where the last inequality follows from Lemma 5.3 in [Srinivas et al., 2010]. Since (4.4) holds for any $\lambda > 0$, by plugging $\lambda = T^{-\alpha}$, for some $\alpha > 0$, we conclude that

$$R_T^2 \leq \frac{8T\|f\|^2_{\mathcal{H}_k(\mathcal{X})}}{\log(1+T^\alpha)}\gamma_{T,T^{-\alpha}}. \tag{4.5}$$

For squared exponential kernels, Corollary 1 in [Vakili et al., 2021] implies that

$$\gamma_{T,T^{-\alpha}} \leq \left(\left(2(1+\alpha)\log T + \tilde{C}(d)\right)^d + 1\right)\log\left(1+T^{1+\alpha}\right) \lesssim \log^d(T)\log\left(1+T^{1+\alpha}\right),$$

where $\tilde{C}(d) = \mathcal{O}(d\log d)$ is independent of T and λ. Hence, using that $\frac{\log(1+T^{\alpha+1})}{\log(1+T^\alpha)} \leq \frac{\alpha+1}{\alpha}$ for $\alpha > 0, T \geq 1$, we obtain

$$R_T^2 \lesssim T\log^d(T)\frac{\log(1+T^{1+\alpha})}{\log(1+T^\alpha)} \lesssim T\log^d(T),$$

concluding the proof.

4.7.2 Proof of Theorem 4.4.4

We first prove the cumulative regret bound for GP-UCB+. As in (4.3), one can show that

$$r_t \leq 2\|f\|_{\mathcal{H}_k(\mathcal{X})}\sigma^{\text{full}}_{t-1,0}(x_T) \leq 2\|f\|_{\mathcal{H}_k(\mathcal{X})}\sup_{x\in\mathcal{X}}\sigma^{\text{full}}_{t-1,0}(x).$$

For Matérn kernels, for large $t \in \mathbb{N}$, [Wu and Schaback, 1993] shows that $\sup_{x\in\mathcal{X}}\sigma^{\text{full}}_{t-1,0}(x) \leq h(\mathcal{X}, X_t^{\text{full}})^\nu$ —see also Lemma 2 in [Wang et al., 2020]. Moreover, we have the trivial bound

$$h(\mathcal{X}, X_t^{\text{full}}) \leq h_t(\mathcal{X}) \coloneqq \sup_{x\in\mathcal{X}}\inf_{\tilde{x}_i\in\{\tilde{x}_1,\ldots,\tilde{x}_t\}}\|x-\tilde{x}_i\|.$$

Hence, for any $\epsilon > 0$,

$$\mathbb{E}_P[r_t] \lesssim \mathbb{E}_P\Big[\sup_{x\in\mathcal{X}} \sigma^{\text{full}}_{t-1,0}(x)\Big] \lesssim \mathbb{E}_P\Big[h_t(\mathcal{X})^{\nu}\Big] \overset{(\star)}{\lesssim} t^{-\frac{\nu}{d}+\varepsilon},$$

where $(\star)$ follows from Proposition 4 in [Helin et al., 2023] —see also Lemma 2 in [Oates et al., 2019].

For squared exponential kernels, Theorem 11.22 in [Wendland, 2004] shows that, for some $C > 0$, $\sup_{x\in\mathcal{X}} \sigma^{\text{full}}_{t-1,0}(x) \leq \exp\big(-C/h(\mathcal{X}, X^{\text{full}}_t)\big)$. Hence, for any $\epsilon \leq \frac{1}{2d}$,

$$\mathbb{E}_P[r_t] \lesssim \mathbb{E}_P\Big[\sup_{x\in\mathcal{X}} \sigma^{\text{full}}_{t-1,0}(x)\Big] \lesssim \mathbb{E}_P\Big[\exp\big(-C/h_t(\mathcal{X})\big)\Big] \overset{(\star)}{\lesssim} \exp\Big(-Ct^{\frac{1}{d}-\varepsilon}\Big),$$

where $(\star)$ follows from Proposition 4 in [Helin et al., 2023] —see also Lemma 2 in [Oates et al., 2019].

For the EXPLOIT+ algorithm, we have that

$$\begin{aligned}
r_t &= f^* - f(x_t) \\
&= f^* - \mu^{\text{full}}_{t-1,0}(x^*) + \mu^{\text{full}}_{t-1,0}(x^*) - \mu^{\text{full}}_{t-1,0}(x_t) + \mu^{\text{full}}_{t-1,0}(x_t) - f(x_t) \\
&\overset{\text{(i)}}{\leq} \|f\|_{\mathcal{H}_k}\sigma^{\text{full}}_{t-1,0}(x^*) + \mu^{\text{full}}_{t-1,0}(x^*) - \mu^{\text{full}}_{t-1,0}(x_t) + \|f\|_{\mathcal{H}_k}\sigma^{\text{full}}_{t-1,0}(x_t) \\
&\overset{\text{(ii)}}{\leq} \|f\|_{\mathcal{H}_k(\mathcal{X})}\sigma^{\text{full}}_{t-1,0}(x^*) + \|f\|_{\mathcal{H}_k(\mathcal{X})}\sigma^{\text{full}}_{t-1,0}(x_t) \\
&\leq 2\|f\|_{\mathcal{H}_k(\mathcal{X})} \sup_{x\in\mathcal{X}} \sigma^{\text{full}}_{t-1,0}(x),
\end{aligned}$$

where for (i) we use twice that, for any $x \in \mathcal{X}$, $|f(x) - \mu_{t-1,0}(x)| \leq \|f\|_{\mathcal{H}_k(\mathcal{X})}\sigma_{t-1,0}(x)$ holds, and for (ii) we use the definition of x_t in the EXPLOIT+ algorithm. The rest of the proof proceeds exactly as the one for GP-UCB+, and we hence omit the details.

4.7.3 Additional Experiments and Implementation Details

Benchmark Functions

This appendix provides detailed descriptions of the numerical experiments conducted in Section 4.5.1. The functional forms of the three objective functions we considered and their respective search space are provided below. For all three benchmark functions we denote $x = (x^1, \ldots, x^d)$ and set $d = 10$.

- Ackley function: for $x \in [-32.768, 32.768]^d$,

$$f(x) = -20 \exp\left(-\frac{1}{5}\sqrt{\frac{1}{d}\sum_{i=1}^{d}(x^i)^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x^i)\right) + 20 + \exp(1).$$

- Rastrigin function: for $x \in [-5.12, 5.12]^d$,

$$f(x) = 10d + \sum_{i=1}^{d}[(x^i)^2 - 10\cos(2\pi x^i)].$$

- Levy function: With $\omega_i = 1 + \frac{x^i - 1}{4}$, for all $i \in \{1, \ldots, d\}$, for $x \in [-10, 10]^d$,

$$f(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{d-1}(\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)] + (\omega_d - 1)^2[1 + \sin^2(2\pi\omega_d)].$$

Recall that Figure 4.2 portrayed the average simple regret of the six Bayesian optimization strategies we consider: GP-UCB+ (proposed algorithm), GP-UCB ([Srinivas et al., 2010]) (both with the choice of $\beta_t = 2$), EXPLOIT+ (proposed algorithm), EXPLOIT (GP-UCB with $\beta_t = 0$), EI (Expected Improvement), and PI (Probability of Improvement). The simple regret values at the last iteration were displayed in Table 4.5.1. Furthermore, Table 4.7.1 shows the standard deviations of the last simple regret values over 20 independent experiments. From Figure 4.2 and Table 4.7.1, one can see that not only were the proposed

Table 4.7.1: Normalized average standard deviation of simple regret with 400 function evaluations for different benchmark objectives in dimension $d = 10$.

METHOD	ACKLEY	RASTRIGIN	LEVY
GP-UCB+	**0.075**	0.797	**0.131**
GP-UCB	1.000	1.000	0.719
EXPLOIT+	0.306	0.577	0.127
EXPLOIT	0.733	0.976	1.000
EI	0.466	0.609	0.160
PI	0.329	**0.360**	0.312

methods (GP-UCB+ and EXPLOIT+) able to yield superior simple regret performance, but also their standard deviations were substantially smaller than those of the other methods, indicating superior stability.

Additionally, Figure 4.5 shows the cumulative regret for GP-UCB+ and GP-UCB with different choices of β_t. All results were averaged over 20 independent experiments. We considered $\beta_t^{1/2} = 2$ and $\beta_t^{1/2} = \max_{x \in \mathcal{X}_D} |f(x)|$ where $\mathcal{X}_D$ is a set of 100 Latin hypercube samples. In all experiments, $\max_{x \in \mathcal{X}_D} |f(x)|$ was significantly larger than 2. Figure 4.5 demonstrates that the choice of β_t can significantly influence the cumulative regret. In particular, we have observed that the GP-UCB+ algorithm tends to work better with smaller β_t values, as the algorithm contains additional exploration steps through random sampling; this behavior can also be seen in Figure 4.5. In all three benchmark functions, GP-UCB exhibits sensitivity to the choice of parameter β_t; in contrast, our EXPLOIT+ algorithm does not require specifying weight parameters and consistently achieves competitive or improved performance across all our experiments.

Hyperparameter Tuning

To train the random forest regression model for California housing dataset [Pace and Barry, 1997], we first divided the dataset into test and train datasets. 80 percent of (feature vector,

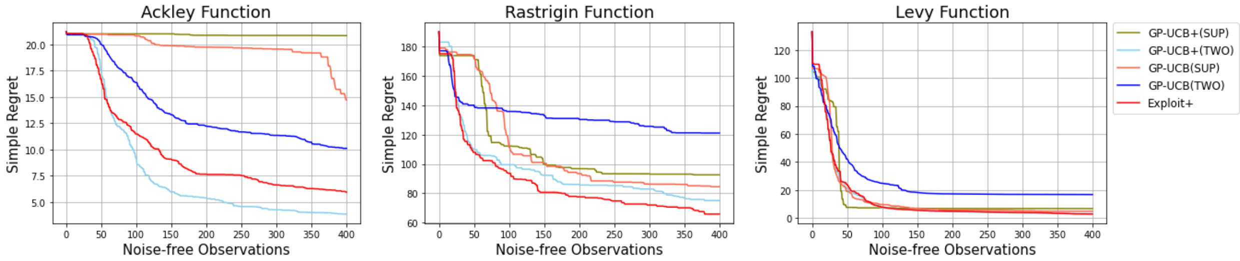


Figure 4.5: Simple regret plots for benchmark functions with $\beta_t^{1/2} = 2$ (TWO) and $\beta_t^{1/2} = \max_{x \in \mathcal{X}_D} |f(x)|$ (SUP).

response) pairs were assigned to be the training set, while the remaining 20 percent were treated as a test set. In constructing the deterministic objective function, we defined it to be a mapping from the vector of four hyperparameters to a negative test error of the model built based on the input and training set. As the model construction may involve randomness coming from the bootstrapped samples, we fixed the random state parameter to remove any such randomness in the definition of the objective. We tuned the following four hyperparameters:

- Number of trees in the forest $\in [10, 200]$.
- Maximum depth of the tree $\in [1, 20]$.
- Minimum number of samples requires to split the internal node $\in [2, 10]$.
- Maximum proportion of the number of features to consider when looking for the best split $\in [0.1, 0.999]$.

For the first three parameters we conducted the optimization task in the continuous domain and rounded down to the nearest integers. Figure 4.6 shows that the proposed algorithms attained smaller cumulative and instantaneous test errors.

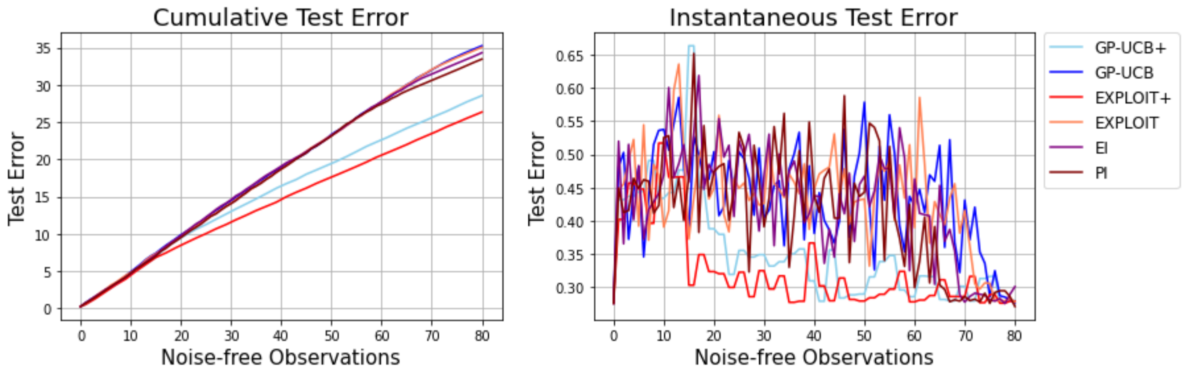


Figure 4.6: Test errors vs number of noise-free observations.

Garden Sprinkler Computer Model

For the Garden Sprinkler computer model, the eight-dimensional search space we considered was given by:

- Vertical nozzle angle $\in [0, 90]$.
- Tangential nozzle angle $\in [0, 90]$.
- Nozzle profile $\in [2 \times 10^{-6}, 4 \times 10^{-6}]$.
- Diameter of the sprinkler head $\in [0.1, 0.2]$.
- Dynamic friction moment $\in [0.01, 0.02]$.
- Static friction moment $\in [0.01, 0.02]$.
- Entrance pressure $\in [1, 2]$.
- Diameter flow line $\in [5, 10]$.

CHAPTER 5

OPTIMIZATION ON MANIFOLDS VIA GRAPH GAUSSIAN PROCESSES

5.1 Introduction

Optimization problems on manifolds are ubiquitous in science and engineering. For instance, low-rank matrix completion and rotational alignment of 3D bodies can be formulated as optimization problems over spaces of matrices that are naturally endowed with manifold structures. These matrix manifolds belong to agreeable families [Ye et al., 2022] for which Riemannian gradients, geodesics, and other geometric quantities have closed-form expressions that facilitate the use of Riemannian optimization algorithms [Edelman et al., 1998, Absil et al., 2009, Boumal, 2020]. In contrast, this chapter is motivated by optimization problems where the search space is a manifold that the practitioner can only access through a discrete point cloud representation, preventing direct use of Riemannian optimization algorithms. Moreover, the hidden manifold may not belong to an agreeable family, further hindering the use of classical methods. Illustrative examples where manifolds are represented by point cloud data include computer vision, robotics, and shape analysis of geometric morphometrics [Hein and Audibert, 2005, Gao et al., 2019, Garcia Trillos et al., 2019]. Additionally, across many applications in data science, high-dimensional point cloud data contains low-dimensional structure that can be modeled as a manifold for algorithmic design and theoretical analysis [Coifman and Lafon, 2006, Belkin et al., 2006, Garcia Trillos et al., 2020b]. Motivated by these problems, this chapter introduces a Bayesian optimization method with convergence guarantees to optimize an expensive-to-evaluate function on a point cloud of manifold samples.

To formalize our setting, consider the optimization problem

$$\text{maximize } f(x), \qquad x \in \mathcal{M}_N, \tag{5.1}$$

where $\mathcal{M}_N = \{x_i\}_{i=1}^N$ is a collection of samples from a compact manifold $\mathcal{M} \subset \mathbb{R}^d$. We assume that the manifold $\mathcal{M}$ is unknown to the practitioner, but that they have access to the samples $\mathcal{M}_N$. The objective function f in (5.1) is defined on the hidden manifold $\mathcal{M}$; however, since $\mathcal{M}$ is unknown, we restrict the search domain to the given point cloud $\mathcal{M}_N$. Motivating examples include locating the portion of busiest traffic along a highway (idealized as a one-dimensional manifold), or finding the point of highest temperature on an artificial surface for material design. In these and other applications, the search domains are manifolds for which only a discrete representation may be available. As a result, Riemannian optimization methods [Edelman et al., 1998, Absil et al., 2009, Boumal, 2020, Hu et al., 2020, Ye et al., 2022] that require Riemannian gradients or geodesics are not directly applicable.

While being discrete, the optimization problem (5.1) is challenging when the objective function f is expensive to evaluate due to computational, monetary, or opportunity costs. For instance, querying f may involve numerically solving a system of partial differential equations, placing a sensor at a new location, or time-consuming human labor. In such cases, solving (5.1) by exhaustive search over $\mathcal{M}_N$ is unfeasible for large N, and it is important to design optimization algorithms that provably require fewer evaluations of the objective than the size N of the point cloud. Solving (5.1) is also challenging in applications where the objective function does not satisfy structural assumptions (e.g. concavity or linearity) other than a sufficient degree of smoothness, and in applications where f is a *black-box* in that one has only access to noisy output from f rather than to an analytic expression of this function. We refer to [Frazier, 2018] for a survey of problems where these conditions arise.

Motivated by these geometric and computational challenges, we introduce an approach to solve (5.1) that works directly on the point cloud $\mathcal{M}_N$ and necessitates few evaluations of the objective. In particular, we show that in the large N limit and under suitable smoothness assumptions, our method provably requires far fewer evaluations of the objective than the size N of the point cloud. Our algorithm falls in the general framework of Bayesian optimization and is specifically designed to achieve such a convergence guarantee. The main focus will be on the mathematical analysis of the proposed approach, but we also present simulation studies to illustrate and complement our theory.

5.1.1 Overview of our Approach

We adopt a Bayesian optimization (BO) approach to solve (5.1) because the problem features that gradients are not available and evaluation of the objective is expensive. BO is an iterative procedure that relies on solving a sequence of surrogate optimization problems to sidestep the need of gradient information on f. At each iteration, the surrogate problem is to optimize an *acquisition function* defined using a *probabilistic model* of the objective function conditioned to previous iterates. The acquisition function should be inexpensive to evaluate and optimize, and at the same time provide useful information about where the optimizer of f is most likely to lie. The probabilistic model should be sufficiently rich to adequately represent the objective function. Many choices of acquisition function have been proposed in the literature, including expected improvement, entropy search, and knowledge gradient (see [Frazier, 2018] for a review). Popular probabilistic models for f include Gaussian processes [Williams and Rasmussen, 2006, Gramacy, 2020] and Bayesian additive regression trees [Chipman et al., 2010]. Adequately choosing the acquisition function and the probabilistic model is essential to the success of BO algorithms.

The BO method that we propose and analyze has the distinctive feature that both the probabilistic model and the acquisition function are carefully chosen to ensure convergence of the returned solution to a global maximizer of f under suitable smoothness assumptions. A natural way to characterize the smoothness of f is to assume it is a sample path from a Gaussian process (GP) defined on $\mathcal{M}$. Under this smoothness assumption, we adopt a graph GP model [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2021] for $f|_{\mathcal{M}_N}$, the restriction of f to the point cloud. The graph GP is designed to be a discretely indexed GP that approximates a Matérn or squared exponential GP on the hidden manifold $\mathcal{M}$ as the size of the point cloud grows to infinity. Applications of graph GPs in Bayesian inverse problems, spatial statistics, and semi-supervised learning are discussed in [Sanz-Alonso and Yang, 2022a, Garcia Trillos et al., 2022, Harlim et al., 2020, 2022]. In this chapter, we extend the convergence analysis for Matérn graph GP models in [Sanz-Alonso and Yang, 2022a,b, Garcia Trillos et al., 2020b, Garcia Trillos and Sanz-Alonso, 2018] to also cover squared exponential kernels, see Proposition 5.2.3.

Such error analysis is important since it allows us to quantify the misspecification error when modeling $f|_{\mathcal{M}_N}$ with a graph GP. In particular, the model that we use for computation does not necessarily match the true distribution of $f|_{\mathcal{M}_N}$ due to the missing information about $\mathcal{M}$; to obtain convergence guarantees, this geometric misspecification needs to be corrected by suitably choosing the acquisition function. We accomplish this goal by applying the framework developed in [Bogunovic and Krause, 2021]. In so doing, we adapt their formulation to cover our problem setting, where f is a sample path from a GP instead of an element of a reproducing kernel Hilbert space.

5.1.2 Contributions and Related Work

Our careful choice of probabilistic model and acquisition function allows us to establish a bound on the simple regret (see (5.4) for its definition) that converges to zero as the number L of evaluations of the objective and the size N of the point cloud converge to infinity while keeping the relation $L \ll N$ (see Theorem 5.2.5, Remark 5.2.6, and Corollary 5.2.7). In other words, our algorithm can provably find a global maximizer of f as we acquire more samples from the compact manifold $\mathcal{M}$ while still keeping the number of evaluations of the objective much smaller than the size of the point cloud. We are not aware of an existing algorithm to solve (5.1) that enjoys a similar convergence guarantee. Synthetic computed examples will complement the theory, illustrate the applicability of our method, and showcase the importance of incorporating geometric information in the probabilistic model.

As noted in [Frazier, 2018], BO algorithms have been most popular in continuous Euclidean domains. Methods that are tailored to manifold settings [Jaquier et al., 2020, 2022] and discrete spaces [Baptista and Poloczek, 2018, Luong et al., 2019, Swersky et al., 2020, Deshwal et al., 2021] have received less attention. On the one hand, the search domain in our setting (5.1) is a discrete subset of a manifold, but naive application of discrete BO (e.g. using a standard Euclidean GP on the ambient space $\mathbb{R}^d$) would fail to adequately exploit the geometric information contained in the point cloud; in particular, it would fail to suitably encode smoothness of the probabilistic model for f along the hidden manifold $\mathcal{M}$. The empirical advantage of our graph GPs over Euclidean kernels will be illustrated in our numerical experiments (see Subsection 5.4.2). On the other hand, the manifold in our setting is only available as a point cloud, which precludes the use of manifold BO approaches [Jaquier et al., 2020, 2022] that require access to geodesic distances and eigenpairs of the Laplace-Beltrami operator on $\mathcal{M}$ for modeling f, and to Riemannian gradients for optimizing the acquisition function. Therefore, our algorithm solves a practical problem for which limited tools with theoretical guarantees are available. In the context of Riemannian optimization, our algorithm is still applicable when the differential geometric quantities necessary for gradient-based methods are not readily available. A closely related work in this direction is [Shustin et al., 2022], which also assumes a point cloud representation of the manifold but instead reconstructs from it tangent spaces, gradients, and retractions, followed by an approximate Riemannian gradient descent. This chapter differs from [Shustin et al., 2022] in that our algorithm is based on Bayesian optimization and no gradient approximation is carried out, as a result of which we do not need to assume the point cloud to be quasi-uniform. Going beyond manifold constraints, optimization of functions with low effective dimensionality has been addressed in [Wang et al., 2016, Kirschner et al., 2019, Cartis et al., 2023a,b] employing subspace methods (see also the references therein).

5.1.3 Outline

- Section 5.2 introduces the *graph Gaussian process upper confidence bound* (GGP-UCB) algorithm and describes the choice of surrogate model and acquisition function. Our main result, Theorem 5.2.5, establishes convergence rates.
- Section 5.3 discusses important practical considerations such as estimating the parameters of the surrogate model and tuning the acquisition function.
- Section 5.4 contains numerical examples that illustrate and complement the theory.
- Section 5.5 closes with a summary of this chapter and directions for further research.
- The proofs of our main results can be found in the appendices.

5.1.4 Notation

For a, b two real numbers, we denote $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. The symbol $\lesssim$ will denote less than or equal to up to a universal constant. For two real sequences $\{a_i\}$ and $\{b_i\}$, we denote (i) $a_i \ll b_i$ if $\lim_i (a_i/b_i) = 0$; (ii) $a_i = O(b_i)$ if $\limsup_i (a_i/b_i) \leq C$ for some positive constant C; and (iii) $a_i \asymp b_i$ if $c_1 \leq \liminf_i (a_i/b_i) \leq \limsup_i (a_i/b_i) \leq c_2$ for some positive constants c_1, c_2.

5.2 The GGP-UCB Algorithm

In this section we introduce our algorithm and establish convergence guarantees. We start in Subsection 5.2.1 by formalizing the problem setting. Subsection 5.2.2 describes the main

GGP-UCB algorithm. The choice of surrogate model and acquisition function are discussed in Subsections 5.2.3 and 5.2.4, respectively. Finally, Subsection 5.2.5 presents our main theoretical result, Theorem 5.2.5.

5.2.1 *Problem Formulation*

Let f be a function defined over a compact Riemannian submanifold $\mathcal{M} \subset \mathbb{R}^d$ of dimension m. Suppose that a full representation of $\mathcal{M}$ is not available and we are only given the dimension m and a point cloud of manifold samples $\{x_i\}_{i=1}^N =: \mathcal{M}_N \subset \mathcal{M}$. We are interested in solving the optimization problem

$$\max_{x \in \mathcal{M}_N} f(x) \tag{5.2}$$

in applications where the objective f is expensive to evaluate and we may only collect $L \ll N$ noisy measurements y_ℓ of the form

$$y_\ell = f(z_\ell) + \eta_\ell, \qquad \eta_\ell \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2), \quad 1 \leq \ell \leq L, \tag{5.3}$$

where $\{z_\ell\}_{\ell=1}^L$ are query points and σ is a given noise level. The goal is then to solve (5.2) with $L \ll N$ queries of f.

Let $\mathcal{Z}_L := \{z_\ell\}_{\ell=1}^L \subset \mathcal{M}_N$ denote the query points sequentially found by our algorithm, introduced in Subsection 5.2.2 below. We shall quantify the performance of our approach using the *simple regret*, defined as

$$r_{N,L} := f(z^*_{\mathcal{M}_N}) - f(z^*_L), \qquad z^*_{\mathcal{M}_N} = \arg\max_{z \in \mathcal{M}_N} f(z), \quad z^*_L = \arg\max_{z \in \mathcal{Z}_L} f(z). \tag{5.4}$$

Note that the simple regret depends both on the number L of queries and on the size N of the point cloud, since $z^*_{\mathcal{M}_N}$ and z^*_L both depend implicitly on N. One should interpret N as a large fixed number and L as the running index. The dependence on N of the query points z_ℓ's will be omitted for notational simplicity.

Remark 5.2.1. The optimizer $z^*_{\mathcal{M}_N}$ over the point cloud $\mathcal{M}_N$ is not necessarily the global optimizer of f over $\mathcal{M}$. Since we only have access to $\mathcal{M}_N$, finding the maximizer over $\mathcal{M}_N$ is the best we can hope for without reconstructing or estimating the hidden manifold $\mathcal{M}$. Nevertheless, we will show in Corollary 5.2.7 that the *continuum regret*, defined as

$$r^{\text{cont}}_{N,L} := f(z^*_{\mathcal{M}}) - f(z^*_L), \qquad z^*_{\mathcal{M}} = \arg\max_{z \in \mathcal{M}} f(z), \quad z^*_L = \arg\max_{z \in \mathcal{Z}_L} f(z), \tag{5.5}$$

also converges to zero as both N and L approach infinity while keeping $L \ll N$ if the x_i's satisfy Assumption 5.2.2. In other words, the maximizer z^*_L returned by our algorithm is an approximate global maximizer of f over $\mathcal{M}$ despite the fact that $z^*_L \in \mathcal{M}_N$. □

5.2.2 Main Algorithm

The Bayesian approach to optimization starts by constructing a GP model for the function to be optimized. We recall that a GP with mean $\mu(\cdot)$ and covariance $c(\cdot,\cdot)$ is a stochastic process where the joint distribution over any finite set of indices $s_1, \ldots, s_n$ is a multivariate Gaussian with mean vector $[\mu(s_i)]_{i=1}^n$ and covariance matrix $[c(s_i, s_j)]_{i,j=1}^n$ [Williams and Rasmussen, 2006]. The mean and covariance functions together encode information about the values of the function, their correlation, and their uncertainty.

In our setting, we need to construct a GP surrogate prior model π_N for f_N, where π_N would simply be an N-dimensional multivariate Gaussian. A natural requirement is that, for $u_N \sim \pi_N$, $u_N(x_i)$ and $u_N(x_j)$ should be highly correlated iff x_i and x_j are close *along the manifold*, that is, if the geodesic distance $d_{\mathcal{M}}(x_i, x_j)$ is small. We shall discuss in Subsection 5.2.3 prior models π_N that fulfill this requirement. Defining the covariance matrix of π_N by using a standard covariance function in the Euclidean space $\mathbb{R}^d$ would in general fail to meet this requirement, since two points may be close in Euclidean space but far apart in terms of the geodesic distance $d_{\mathcal{M}}$ in $\mathcal{M}$.

Once a choice of surrogate prior model is made, the next step is to sequentially find query points by maximizing an acquisition function [Srinivas et al., 2010]. Suppose we have picked query points $z_1, \ldots, z_{\ell-1}$ in the first $\ell - 1$ iterations and obtained noisy measurements

$$y_k = f(z_k) + \eta_k, \qquad \eta_k \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2), \qquad 1 \leq k \leq \ell - 1. \tag{5.6}$$

At the ℓ-th iteration, we will pick the next query point z_ℓ by maximizing an upper confidence bound acquisition function [Srinivas et al., 2010, Bogunovic and Krause, 2021] of the form

$$A_{N,\ell}(z) = \mu_{N,\ell-1}(z) + B_{N,\ell}\sigma_{N,\ell-1}(z), \qquad z \in \mathcal{M}_N, \tag{5.7}$$

where $B_{N,\ell}$ is a user-chosen parameter, and $\mu_{N,\ell-1}$, $\sigma_{N,\ell-1}$ are the mean and standard deviation of the posterior distribution $\pi_N(\cdot \,|\, y_1, \ldots, y_{\ell-1})$. Denoting by $c_N(\cdot,\cdot)$ the covariance function of the surrogate prior π_N, i.e., $c_N(x_i, x_j)$ is the covariance between $u_N(x_i)$ and $u_N(x_j)$ for $u_N \sim \pi_N$, we have the expressions

$$\begin{aligned} \mu_{N,\ell-1}(z) &= c_{N,\ell-1}(z)^\top (C_{N,\ell-1} + \sigma^2 I)^{-1} Y_{\ell-1}, \\ \sigma^2_{N,\ell-1}(z) &= c_N(z,z) - c_{N,\ell-1}(z)^\top (C_{N,\ell-1} + \sigma^2 I)^{-1} c_{N,\ell-1}(z), \end{aligned} \qquad z \in \mathcal{M}_N, \tag{5.8}$$

where $Y_{\ell-1} = (y_1, \ldots, y_{\ell-1})^\top \in \mathbb{R}^{\ell-1}$, $c_{N,\ell-1}(z) \in \mathbb{R}^{\ell-1}$ is a vector whose i-th entry is $c_N(z, z_i)$, and $C_{N,\ell-1} \in \mathbb{R}^{\ell-1\times\ell-1}$ is a matrix with entries $(C_{N,\ell-1})_{ij} = c_N(z_i, z_j)$.

The GGP-UCB method is summarized in Algorithm 5.2.1. The intuition is that maximizing the acquisition function (5.7) represents a compromise between choosing points where the mean of the surrogate is large (exploitation) and where the variance is large (exploration). The parameter $B_{N,\ell}$ balances these two competing goals and its choice is crucial to the performance of the algorithm. In particular, we will discuss in Subsection 5.2.4 a choice

Algorithm 5.2.1 The GGP-UCB Algorithm

```
Require: Point cloud $\mathcal{M}_N$; prior $\pi_N$; initialization $z_0$; total iterations $L$; parameters $\{B_{N,\ell}\}_{\ell=1}^{L}$.
  for $\ell = 1, \dots, L$ do
    Observe $y_{\ell-1} = f(z_{\ell-1}) + \eta_{\ell-1}$, with $\eta_{\ell-1} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2)$.
    Compute $\mu_{N,\ell-1}$ and $\sigma_{N,\ell-1}$ based on $\{(z_k, y_k)\}_{k=0}^{\ell-1}$.
    Choose $z_\ell = \arg\max_{z \in \mathcal{M}_N} \Big\{\mu_{N,\ell-1}(z) + B_{N,\ell}\sigma_{N,\ell-1}(z)\Big\}$.
  end for
Ensure: $z_1, \dots, z_L$.
```

of $B_{N,\ell}$ that helps correct for misspecification arising from the point cloud representation of $\mathcal{M}$, and we will discuss in Subsection 5.3.3 a practical approach for tuning $B_{N,\ell}$ empirically. Finally, we point out that in practice one may choose to return as output of the algorithm the candidate z_ℓ that leads to the largest observation y_ℓ when the noise is small, or, otherwise, the z_ℓ that maximizes the posterior mean at the L-th iteration, i.e., the mean $\mu_{N,L}$ of $\pi_N(\cdot \,|\, y_1, \dots, y_L)$.

5.2.3 *Choice of Prior: Graph Gaussian Processes (GGPs)*

In this subsection we review the construction of GGP models for f_N, the restriction of f to the x_i's. We first give a brief overview of manifold GPs before describing GGPs. Manifold GPs will be used in our theoretical analysis, but are not implementable in our setting since the manifold $\mathcal{M}$ is unknown to the practitioner. The presentation in this subsection follows [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2020] and readers familiar with manifold GPs and GGPs can skip to Proposition 5.2.3.

Manifold GP Models

Since f is a function over $\mathcal{M}$, it will be useful to start by recalling the construction of GPs over $\mathcal{M}$. A naive approach would be to simply use geodesic distances instead of Euclidean ones in covariance functions such as the Matérn and squared exponential (SE)

$$c^{\mathrm{Ma}}_{\nu,\kappa}(x,\tilde{x}) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\kappa|x-\tilde{x}|\right)^{\nu} K_{\nu}\left(\kappa|x-\tilde{x}|\right), \qquad c^{\mathrm{SE}}_{\tau}(x,\tilde{x}) = \exp\left(-\frac{|x-\tilde{x}|^2}{4\tau}\right), \tag{5.9}$$

where $|\cdot|$ denotes the Euclidean distance, Γ is the gamma function, and K_ν is the modified Bessel function of the second kind. The parameters ν and κ in the Matérn covariance control the smoothness of sample paths and the inverse length scale of the field, while the parameter τ in the squared exponential covariance controls the length scale. (Note that we are not including the variance parameter that usually appears as a multiplicative constant in the covariances.) Unfortunately, the naive idea of plugging in geodesic distances often leads to failure of positive definiteness of the resulting covariance matrix [Gneiting, 2013, Feragen et al., 2015].

To circumvent this challenge, the seminal paper [Lindgren et al., 2011] exploits the stochastic partial differential equation (SPDE) representation of Euclidean GPs with the Matérn covariance function. More precisely, it is shown in [Whittle, 1963] that the GP with covariance function $c^{\mathrm{Ma}}_{\nu,\kappa}$ over a Euclidean space $\mathbb{R}^m$ is the unique stationary solution to the following equation (up to a multiplicative constant independent of κ)

$$(\kappa^2 - \Delta)^{\frac{\nu}{2}+\frac{m}{4}} u(x) = \kappa^{\nu} \mathcal{W}(x), \qquad x \in \mathbb{R}^m, \tag{5.10}$$

where Δ is the usual Laplacian on $\mathbb{R}^m$ and $\mathcal{W}$ is a spatial white noise with unit variance. The equation (5.10) can then be lifted to the manifold case to construct Matérn GPs over manifolds [Lindgren et al., 2011]. Based on this idea, the papers [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2020] study the following series definition of GPs over compact manifolds:

$$\text{(Matérn manifold-GP)} \quad u^{\mathrm{Ma}} = \kappa^{s-\frac{m}{2}} \sum_{i=1}^{\infty} (\kappa^2 + \lambda_i)^{-\frac{s}{2}} \xi_i \psi_i, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.11}$$

where (λ_i, ψ_i)'s are eigenvalue-eigenfunction pairs of the negative Laplace-Beltrami operator $-\Delta_{\mathcal{M}}$ on $\mathcal{M}$. Compactness of $\mathcal{M}$ ensures that $\Delta_{\mathcal{M}}$ admits a countable eigenbasis so that the solution to the analog equation of (5.10) over $\mathcal{M}$ can be represented as the series (5.11). The parameters $s, \kappa > 0$ in (5.11) control the smoothness and the inverse length scale as in the Euclidean case: $s = \nu + m/2$ controls the spectrum decay, while κ acts as a cutoff on the essential frequencies. The scaling factor $\kappa^{s-\frac{m}{2}}$ ensures that samples from different κ's have L^2-norms on the same order (see. e.g. [Sanz-Alonso and Yang, 2022a, Remark 2.1]), which is essential in applications where κ needs to be inferred.

As the smoothness parameter $\nu \to \infty$, it can be shown that the Matérn covariance converges (after a suitable normalization) to the SE covariance (see e.g. [Williams and Rasmussen, 2006, Section 4.2]). Accordingly, there is a similar SPDE to (5.10) that characterizes the SE GP on a Euclidean space $\mathbb{R}^m$ [Borovitskiy et al., 2020]:

$$e^{-\frac{\tau\Delta}{2}} u(x) = \tau^{\frac{m}{4}} \mathcal{W}(x), \qquad x \in \mathbb{R}^m,$$

which motivates its manifold analog as the series expansion

$$\text{(SE manifold-GP)} \quad u^{\text{SE}} = \tau^{\frac{m}{4}} \sum_{i=1}^{\infty} e^{-\frac{\lambda_i \tau}{2}} \xi_i \psi_i, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.12}$$

where (λ_i, ψ_i)'s are eigenvalue-eigenfunction pairs of $-\Delta_{\mathcal{M}}$. Here the factor $\tau^{\frac{m}{4}}$ is again interpreted as balancing the magnitude of samples from different τ's (see Lemma 5.6.4). Furthermore, the induced covariance function has the form

$$c^{\text{SE}}(x, \tilde{x}) = \tau^{\frac{m}{2}} \sum_{i=1}^{\infty} e^{-\lambda_i \tau} \psi_i(x) \psi_i(\tilde{x}). \tag{5.13}$$

Notice that this is also known as the heat kernel (up to the scaling factor $\tau^{m/2}$), which is a natural generalization of the SE kernel over the manifold. A similar expression holds for the induced covariance function of u^{Ma}:

$$c^{\mathrm{Ma}}(x, \tilde{x}) = \kappa^{2s-m} \sum_{i=1}^{\infty} (\kappa^2 + \lambda_i)^{-s} \psi_i(x) \psi_i(\tilde{x}). \tag{5.14}$$

Besides the connection with their Euclidean counterparts, notice that the random fields (5.11) and (5.12) are series expansions of the eigenfunctions of the Laplace-Beltrami operator, which form an orthonormal basis for $L^2(\mathcal{M})$ and carry rich information about the geometry of $\mathcal{M}$; therefore, (5.11) and (5.12) are natural GP models for functions over $\mathcal{M}$. However, computing the pairwise covariances (5.13) and (5.14) between any two points would require knowledge of the Laplace-Beltrami eigenvalues and eigenfunctions, which are only known analytically for a few manifolds such as the sphere and the torus, and can otherwise be expensive to approximate. More importantly, in applications where only a point cloud representation of $\mathcal{M}$ is available we need an empirical way to approximate the manifold GPs (5.11) and (5.12). To that end, we will adopt a manifold learning approach using graph Laplacians.

GGP Models

The construction in this subsection follows [Sanz-Alonso and Yang, 2022a]. Given a point cloud $\mathcal{M}_N = \{x_1, \dots, x_N\} \subset \mathcal{M}$, recall that our goal is to build a GP model for f_N, the restriction of f to the x_i's. It then suffices to construct an N-dimensional Gaussian that approximates the manifold GPs (5.11) and (5.12); in particular, we need to construct a suitable covariance matrix.

To start with, observe that the manifold Matérn GP (5.11) can be seen as the Karhunen-Loève expansion of the Gaussian measure [Bogachev, 1998] (the infinite-dimensional analog of multivariate Gaussian) $\mathcal{N}(0,\mathcal{C})$, where $\mathcal{C}$ is the covariance operator

$$\mathcal{C} = \kappa^{2s-m}(\kappa^2 I - \Delta_{\mathcal{M}})^{-s},$$

with I denoting the identity operator. Therefore a natural candidate for an N-dimensional approximation is to consider the multivariate Gaussian $\mathcal{N}(0,\mathcal{C}_N)$, where

$$\mathcal{C}_N = \kappa^{2s-m}(\kappa^2 I_N + \Delta_N)^{-s} \tag{5.15}$$

for some $\Delta_N \in \mathbb{R}^{N\times N}$ constructed with the x_i's that approximates $-\Delta_{\mathcal{M}}$ with I_N denoting the N-dimensional identity matrix. We shall set Δ_N to be a suitable *graph Laplacian*, as we describe next.

Let $\mathcal{M}_N = \{x_i\}_{i=1}^N$ be a collection of points on $\mathcal{M}$. One can construct a weighted graph over the x_i's by introducing a weight matrix $W \in \mathbb{R}^{N\times N}$ whose entry W_{ij} represents the similarity between points x_i and x_j. The *unnormalized graph Laplacian* is then defined as $\Delta_N = D - W$, where D is a diagonal matrix whose entries are $D_{ii} = \sum_{j=1}^N W_{ij}$. One can immediately check that Δ_N is symmetric and positive semi-definite using the relation

$$v^\top \Delta_N v = \frac{1}{2}\sum_{i=1}^N\sum_{j=1}^N W_{ij}|v_i - v_j|^2, \quad v \in \mathbb{R}^N,$$

implying that Δ_N admits a spectral decomposition with nonnegative eigenvalues $\{\lambda_{N,i}\}_{i=1}^N$ (ordered increasingly) and the associated eigenvectors $\{\psi_{N,i}\}_{i=1}^N$ form an orthonormal basis for $\mathbb{R}^N$. Several normalizations of Δ_N have also been considered, including the *random walk graph Laplacian* $\Delta_N^{\mathrm{rw}} = D^{-1}\Delta_N$ and *symmetric graph Laplacian* $\Delta_N^{\mathrm{sym}} = D^{-1/2}\Delta_N D^{-1/2}$, see [Von Luxburg, 2007]. We focus on the unnormalized version due to its symmetry, which makes it a valid choice in the covariance matrix (5.15), and its convergence properties that we will describe now.

As its name suggests, Δ_N approximates the Laplace-Beltrami operator in a suitable sense. Indeed, if we set the pairwise similarity to be

$$W_{ij} = \frac{2(m+2)}{N\nu_m h_N^{m+2}} \mathbf{1}\{|x_i - x_j| < h_N\}, \tag{5.16}$$

where $|\cdot|$ denotes the Euclidean distance, ν_m is the volume of the $m-$dimensional unit ball and h_N is a graph connectivity parameter, then for suitable choices of h_N it can be shown (see e.g. [Garcia Trillos et al., 2020a] or Proposition 5.6.2) that the eigenpair $(\lambda_{N,i}, \psi_{N,i})$ of Δ_N approximates the corresponding eigenpair (λ_i, ψ_i) of $-\mathrm{vol}(\mathcal{M})^{-1}\Delta_{\mathcal{M}}$. Based on this fact, we shall now define two GGPs as follows

$$\text{(Matérn GGP)} \quad u_N^{\mathrm{Ma}} = \kappa^{s-\frac{m}{2}} \sum_{i=1}^{k_N} (\kappa^2 + \lambda_{N,i})^{-\frac{s}{2}} \xi_i \psi_{N,i}, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.17}$$

$$\text{(SE GGP)} \quad u_N^{\mathrm{SE}} = \tau^{\frac{m}{4}} \sum_{i=1}^{k_N} e^{-\frac{\lambda_{N,i}\tau}{2}} \xi_i \psi_{N,i}, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.18}$$

where $k_N \leq N$ is a truncation level to be determined. Notice that Matérn and SE GGPs can be interpreted as discretely indexed GPs over the graph $(\mathcal{M}_N, W)$, hence the name GGP. Similar objects have also been studied by [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2021, Dunson et al., 2022]. When $k_N = N$, we see that (5.17) is nothing but the multivariate Gaussian $\mathcal{N}\big(0, \kappa^{2s-m}(\kappa^2 I_N + \Delta_N)^{-s}\big)$, matching our goal (5.15) at the beginning. The motivation for introducing the truncation is that the spectral approximation accuracy degrades quickly when we go to higher modes (see e.g. Proposition 5.6.2), where the error bounds are only meaningful when $h_N\sqrt{\lambda_i} \ll 1$. Therefore (5.17) can be seen as a low rank approximation of (5.15) that keeps only the low and accurate frequencies. By Weyl's law (see e.g. [Canzani, 2013, Theorem 72]), $\lambda_i \asymp i^{2/m}$ and in particular $\lambda_i \to \infty$, which suggests a necessary condition $k_N \ll h_N^{-m}$. In Subsection 5.3.2 we discuss an empirical way of choosing

k_N. The induced covariance functions take the form

$$\left.\begin{aligned} c_N^{\mathrm{Ma}}(x,\tilde{x}) &= \kappa^{2s-m}\sum_{i=1}^{k_N}(\kappa^2+\lambda_{N,i})^{-s}\psi_{N,i}(x)\psi_{N,i}(\tilde{x}), \\ c_N^{\mathrm{SE}}(x,\tilde{x}) &= \tau^{\frac{m}{2}}\sum_{i=1}^{k_N}e^{-\lambda_{N,i}\tau}\psi_{N,i}(x)\psi_{N,i}(\tilde{x}), \end{aligned}\right. \qquad x,\tilde{x}\in\mathcal{M}_N. \tag{5.19}$$

Notice that the definitions (5.17) and (5.18) are completely parallel with (5.11) and (5.12); hence the spectral convergence of Δ_N leads to convergence of GGPs to their manifold counterparts. We will rely on the following assumption:

Assumption 5.2.2. $\mathcal{M}$ is a smooth, compact and connected submanifold of dimension $m \geq 2$ in $\mathbb{R}^d$ that has no boundary and bounded sectional curvature, normalized so that $\mathrm{vol}(\mathcal{M}) = 1$. Assume the x_i's are i.i.d. samples from the uniform distribution on $\mathcal{M}$.

The following result provides a simplified statement of the convergence analysis for Matérn GGPs in [Sanz-Alonso and Yang, 2022a,b] and in addition covers SE GGPs. The proof can be found in the Appendix 5.6.1.

Proposition 5.2.3. *Let $0 < \iota < 1$ be arbitrary. Define $\alpha_m = (m+4+\iota)\vee(2m)$ and $\beta_{m,s} = \frac{2s-3m+1}{6m+6}\wedge 1$. Let $p_m = \frac{3}{4}$ when $m=2$ and $p_m = \frac{1}{m}$ otherwise. For $s > \frac{3}{2}m - \frac{1}{2}$, set*

$$\begin{aligned} &\text{(Matérn GGP)} \qquad h_N \asymp N^{-\frac{1}{\alpha_m}}(\log N)^{\frac{p_m}{2}}, \quad k_N \asymp N^{\frac{m\beta_{m,s}}{(2s-3m+1)\alpha_m}}(\log N)^{-\frac{mp_m\beta_{m,s}}{(4s-6m+2)}}, \\ &\text{(SE GGP)} \qquad h_N \asymp N^{-\frac{1}{\alpha_m}}(\log N)^{\frac{p_m}{2}}, \quad (\log N)^{\frac{m}{2}} \ll k_N \ll N^{\frac{m}{(3m+3)\alpha_m}}(\log N)^{-\frac{mp_m}{6m+6}}. \end{aligned}$$

Under Assumption 5.2.2, with probability $1 - O(N^{-c})$ for some $c > 0$, there exists $T_N : \mathcal{M} \to \{x_1, \dots, x_N\}$ satisfying $T_N(x_i) = x_i$ such that

$$\left.\begin{aligned} \mathbb{E}\|u_N^{Ma}\circ T_N - u^{Ma}\|_\infty &\lesssim N^{-\frac{\beta_{m,s}}{2\alpha_m}}(\log N)^{\frac{\beta_{m,s}p_m}{4}} \\ \mathbb{E}\|u_N^{SE}\circ T_N - u^{SE}\|_\infty &\lesssim N^{-\frac{1}{2\alpha_m}}(\log N)^{\frac{p_m}{4}} \end{aligned}\right\} =: \epsilon_N. \tag{5.20}$$

The fact that we can study L^∞-norms of these random fields follows from their almost sure continuity established in [Sanz-Alonso and Yang, 2022b, Lemma 3] and Lemma 5.6.4. Proposition 5.2.3 will be a key ingredient in establishing regret bounds for GGP-UCB.

5.2.4 Choice of Acquisition Function

When the GGP prior π_N matches the truth f_N, i.e., when f_N is a sample from π_N, [Srinivas et al., 2010] gives a choice of $B_{N,L}$ for the acquisition function (5.7) that ensures vanishing regret. However, this is not necessarily true in our case since f_N is the restriction of a function f over $\mathcal{M}$ whereas the GGP π_N is only constructed with $\mathcal{M}_N$. A mismatch is possible and below we address this issue following ideas in [Bogunovic and Krause, 2021].

Suppose that the function f to be optimized is a sample from the manifold GP (5.11) (or (5.12)) and we adopt the corresponding GGP prior π_N given by (5.17) (resp. (5.18)) for f_N. Proposition 5.2.3 then imples that if $u_N \sim \pi_N$, we have with probability $1-\delta$

$$\|u_N - f_N\|_\infty \leq \delta^{-1}\epsilon_N, \tag{5.21}$$

where here $\|\cdot\|_\infty$ denotes the entry-wise maximum and ϵ_N is a placeholder for the approximation error defined in (5.20). In other words, there is potentially a *misspecification error* coming from the fact that we are using an approximate GP to model f_N. With the understanding of such error obtained in Proposition 5.2.3, we can follow the approach in [Bogunovic and Krause, 2021] and set

$$B_{N,\ell} = \sqrt{2\log\left(\frac{\pi^2\ell^2 N}{6\delta}\right)} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}, \tag{5.22}$$

where we recall that σ is the noise standard deviation. Notice that this differs from the plain GP-UCB in [Srinivas et al., 2010] by the additional term $\epsilon_N\sqrt{\ell-1}/\delta\sigma$ that aims to correct for the misspecification. Intuitively, such correction leads to an increase of the weight on the posterior standard deviation, which accounts for the increased uncertainty due to the approximate modeling. Therefore at the ℓ-th iteration, we shall pick the candidate z_ℓ as

$$z_\ell = \underset{z \in \mathcal{M}_N}{\arg\max} \left\{ \mu_{N,\ell-1}(z) + \left[\sqrt{2 \log \left(\frac{\pi^2 \ell^2 N}{6\delta} \right)} + \frac{\epsilon_N \sqrt{\ell - 1}}{\delta \sigma} \right] \sigma_{N,\ell-1}(z) \right\}, \tag{5.23}$$

where $\mu_{N,\ell-1}$ and $\sigma_{N,\ell-1}$ are defined as in (5.8) but with $c_N(\cdot,\cdot)$ being the graph covariance functions (5.19).

Remark 5.2.4. In our setting we do not have access to the underlying manifold $\mathcal{M}$ and hence continuous optimization is not applicable. As a result, (5.23) is optimized over the discrete set $\mathcal{M}_N$ and would require evaluation of the acquisition function over the entire point cloud. If N is large and evaluating the acquisition function over the full point cloud is costly, then one can, for practical purposes, approximately optimize (5.23) using a subsample of the point cloud $\mathcal{M}_N$. Optimizing the acquisition function approximately is common practice in BO. It is important to emphasize, however, that in the applications that motivate our work the objective function f is much more expensive to evaluate than the acquisition function. □

5.2.5 Main Result: Regret Bounds

Now we are ready to state our main result. Its proof can be found in Appendix 5.6.2.

Theorem 5.2.5. *Suppose f is a sample from the Matérn manifold-GP* (5.11) *with parameters κ, s (resp. SE manifold-GP* (5.12) *with parameter τ). Let π_N be the Matérn (resp. SE) GGP constructed as in Proposition 5.2.3 with the same parameters. Apply Algorithm 5.2.1 with π_N and with $B_{N,\ell}$ given by* (5.22)*. Under Assumption 5.2.2, for N large enough, we* *have with probability* $1-2\delta-O(N^{-c})$ *that*

$$r_{N,L} \le C\left[\frac{\sqrt{2\log(\pi^2 L^2 N/6\delta)}}{\sqrt{L}} + \frac{\epsilon_N}{\delta\sigma}\right]\sqrt{k_N \log L}\,, \qquad \forall\ L \ge 1,$$

where $c, C > 0$ *are universal constants. Here we recall that* σ *is the observation noise standard deviation,* k_N *is the truncation parameter in Proposition 5.2.3, and* ϵ_N *is the approximation error as in* (5.20).

Remark 5.2.6. By plugging the scaling in Proposition 5.2.3, we get

$$r_{N,L} = \widetilde{O}\Big((L^{-\frac{1}{2}} + \epsilon_N)\sqrt{k_N}\Big) = \widetilde{O}\begin{cases} L^{-\frac{1}{2}} N^{\frac{m\beta_{m,s}}{(4s-6m+2)\alpha_m}} + N^{-\frac{(2s-4m+1)\beta_{m,s}}{(4s-6m+2)\alpha_m}} & \text{(Matérn)} \\ L^{-\frac{1}{2}} + N^{-\frac{1}{2\alpha_m}} & \text{(SE)} \end{cases}. \tag{5.24}$$

Here the notation $\widetilde{O}(\cdot)$ means that we have dropped all dependence on logarithmic factors. The regret goes to zero as both N and L approach infinity in both cases (when $s > \frac{7}{4}m + \frac{1}{2}$ for the Matérn case), although we recall that N should be treated as a fixed large number and L is the running index. The two terms in the above upper bound can be understood as the error incurred by Bayesian optimization and by misspecification, respectively. For a fixed N, the regret will decrease as $L \to \infty$ to a threshold imposed by the misspecification error, which itself will go to zero with more data points from $\mathcal{M}$ as $N \to \infty$. Notice that the two terms are balanced at $L \asymp N^{\beta_{m,s}/\alpha_m}$ for the Matérn case and $L \asymp N^{1/\alpha_m}$ for the SE case. Since $\beta_{m,s} \le 1$, for a fixed large enough N, number of queries of the order $L \ll N$ would be sufficient in both cases because otherwise the error coming from misspecification will dominate. We shall demonstrate by simulations in Section 5.4 that the algorithm is able to find the optimizer (or an almost optimizer) after a number L of queries that is significantly smaller than the size N of the point cloud. $\square$

We end this section with a bound on the continuum regret $r_{N,L}^{\text{cont}}$ (see its definition in

(5.5)).

Corollary 5.2.7. *Under the same assumptions as in Theorem 5.2.5,* $r_{N,L}^{\mathrm{cont}}$ *follows the same bound as* (5.24)*.*

Therefore we can recover a global maximizer of f over $\mathcal{M}$ as both N and L tend to infinity while keeping $L \ll N$.

5.3 Estimation and Tuning of GGP-UCB Parameters

This section discusses important considerations for the practical implementation of the GGP-UCB algorithm. Subsections 5.3.1, 5.3.2 and 5.3.3 describe respectively the estimation of prior GGP parameters, the choice of graph connectivity h_N and truncation level k_N, and the empirical tuning of the acquisition function.

5.3.1 Parameter Estimation

Theorem 5.2.5 holds under the assumption that the GGP model uses the same parameters κ, s, τ as those for the truth. However, these parameters are typically unavailable in practice and need to be estimated. In this subsection we give a possible empirical solution.

Recall that at the ℓ-th iteration we pick the next query point z_ℓ based on (5.23) and observe a noisy function value

$$y_\ell = f(z_\ell) + \eta_\ell,$$

where f is assumed to be a sample from the manifold GP (5.11) or (5.12) with parameter θ ($\theta = (\kappa, s)$ for the Matérn case and $\theta = \tau$ for the SE case). We shall obtain an estimate θ_ℓ of

θ in each iteration of the above procedure using a maximum likelihood estimation approach:

$$\theta_\ell = \arg\max_{\theta} \mathbb{P}(Y_\ell \,|\, \theta), \tag{5.25}$$

where $Y_\ell = (y_1, \ldots, y_\ell)^\top$. Exact maximization of (5.25) would require knowing the covariance structure of the underlying manifold GP, in particular the eigenpairs of the Laplace-Beltrami operator, the lack of which is precisely the reason why we introduced our graph-based approach. However, since the GGPs (5.17) and (5.18) are what we actually use for modeling f, a natural idea is then to seek for parameters of these surrogate models that can best fit the data. Therefore we shall consider the following "surrogate" data model by pretending that the y_ℓ's are generated from the GGPs:

$$\begin{aligned} &y_k = u_N(z_k) + \eta_k, \qquad \eta_k \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma^2), \qquad k = 1, \ldots, \ell, \\ &u_N \sim \mathcal{N}(0, \mathcal{C}_N^\theta), \end{aligned}$$

where $\mathcal{C}_N^\theta$ is the covariance matrix associated with (5.19). It follows that

$$Y_\ell \sim \mathcal{N}(0, \Sigma_N^\theta), \qquad \Sigma_N^\theta = A\mathcal{C}_N^\theta A^\top + \sigma^2 I_\ell, \tag{5.26}$$

where $A \in \mathbb{R}^{\ell \times N}$ is a matrix of 0's and 1's whose entries indicate the indices of the z_ℓ's among $\mathcal{M}_N = \{x_i\}_{i=1}^N$. Maximization of the likelihood of Y_ℓ under (5.26) gives the estimate θ_ℓ.

5.3.2 Determining the Truncation Level $\boldsymbol{k_N}$ and the Graph Connectivity $\boldsymbol{h_N}$

As mentioned in Subsection 5.2.3, the truncation level k_N is crucial in that the higher frequencies obtained from the graph Laplacian give poor approximations to their manifold counterparts and can have a negative impact on approximating manifold GPs. Proposition

5.2.3 gives a scaling for k_N that is based on the asymptotic behavior of the graph Laplacian. Empirically, one can simply choose k_N by plotting the spectrum of Δ_N.

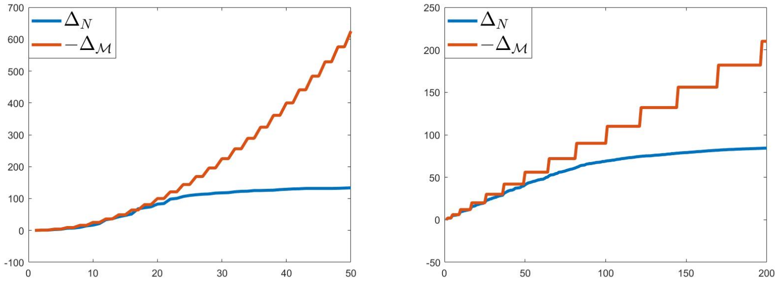


Figure 5.1: Spectrum of Δ_N versus spectrum of $-\Delta_{\mathcal{M}}$ for the unit circle (left) and the unit sphere (right).

Proposition 5.6.2 in the appendix gives an upper bound on the eigenvalue approximation, where the error is small only when $h_N\sqrt{\lambda_i} \ll 1$. In practice, what we usually observe is not only such poor spectral approximation for large i's, but also a "saturation" of the graph Laplacian eigenvalues after certain threshold. Figure 5.1 shows the first several eigenvalues of the Laplace-Beltrami operator $-\Delta_{\mathcal{M}}$ and the graph Laplacian Δ_N constructed with (a) $N = 500$ points from the unit circle; and (b) $N = 3000$ points from the unit sphere. We can see from both plots that for small index i, the eigenvalues $\lambda_{N,i}$ of Δ_N approximate well the eigenvalues λ_i of $-\Delta_{\mathcal{M}}$; however, the spectrum of Δ_N is essentially flat for large i. Therefore one can choose k_N to be around the point of saturation in the spectrum of Δ_N. Such saturation phenomenon, noted for instance in [Garcia Trillos et al., 2020b, Garcia Trillos and Sanz-Alonso, 2018, Garcia Trillos et al., 2019], also helps to explain the need for truncation. Indeed, the eigenvalues $\lambda_{N,i}$ determine the decay of the coefficients in the series (5.17) and (5.18) defining our GGPs. Without a truncation, too much weight would be given to the high frequencies, which would lead to overly rough sample paths.

Another key parameter in the construction of our GGPs is the graph connectivity h_N in the definition of the weights (5.16). A common choice [Garcia Trillos et al., 2020a, Sanz-Alonso and Yang, 2022b] is $h_N \propto \sqrt{\rho_N}$, where ρ_N defined in (5.36) can be interpreted as the maximum distance between any two nearby x_i's or the "resolution" of $\mathcal{M}_N$. In particular, the choice $h_N \propto \sqrt{\rho_N}$ ensures that the neighborhood of each x_i in the graph is local but rich enough to capture the local geometry. Moreover, this choice balances the two terms in the error bound $\rho_N/h_N + h_N\sqrt{\lambda_i}$ in Proposition 5.6.2. The scaling of ρ_N is shown in [Garcia Trillos et al., 2020a, Theorem 2] and recorded in Proposition 5.6.1, which leads to the choice $h_N = CN^{-1/2m}$ (the logarithmic factor can be absorbed into the proportion constant). The proportion constant can be determined again by plotting the corresponding spectrum of Δ_N. Starting with a large C, one can keep decreasing the value of C while observing the point of saturation becoming larger, until one hits a point where the spectrum is no longer meaningful. This latter case will happen when h_N is too small so that the graph is disconnected and the graph Laplacian has repeated zero eigenvalues.

5.3.3 Empirical Tuning of the Acquisition Function

Recall that the selection rule (5.23) incorporates information on the level of misspecification ϵ_N incurred by the GGPs. Proposition 5.2.3 gives such a bound on ϵ_N, which goes to zero as $N \to \infty$. However, for practical considerations, the upper bound may not be small for certain ranges of δ and N, especially since there is a possibly non-sharp proportion constant in ϵ_N. Therefore this could cause the term $\epsilon_N\sqrt{\ell-1}/\delta\sigma$ in $B_{N,\ell}$ to be overly large, so that the acquisition function puts too much weight on the posterior standard deviation, as a result of which exploration overwhelms exploitation. For this reason, we shall consider instead setting $B_{N,\ell}$ as

$$B_{N,\ell} = a\sqrt{2\log\left(\frac{\pi^2\ell^2 N}{6\delta}\right)}\,, \tag{5.27}$$

with a tuning parameter $a > 0$. As noticed in the simulation studies in [Srinivas et al., 2010], setting $a = 1/5$ in practice leads to the best performance in well-specified cases, i.e., when $\epsilon_N = 0$ (although their theoretical results are proved for $a = 1$). Motivated by this observation, we shall set $a = 1/2$ throughout for our simulation studies in Section 5.4 to account for the case $\epsilon_N \neq 0$. The idea is that the original correction term $\epsilon_N \sqrt{\ell - 1}/\delta\sigma$ for misspecification is now absorbed as the increment $(1/2 - 1/5)\sqrt{2\log(\pi^2 \ell^2 N/6\delta)}$.

Finally, the selection rule (5.23) searches for the query points over the entire $\mathcal{M}_N$ at each iteration, which could return points that have already been picked and get stuck at local optima in practice. We shall modify (5.23) slightly by maximizing it over $\mathcal{M}_N \backslash \{z_1, \ldots, z_{\ell-1}\}$ at the ℓ-th iteration, i.e., by asking the algorithm to output a query point that has not been chosen in previous iterations.

5.4 Numerical Examples

This section contains preliminary numerical experiments that complement the theory. The main focus will be to illustrate the performance of our method within the scope of Bayesian optimization rather than conduct an exhaustive comparison with existing discrete optimization algorithms.

In Subsection 5.4.1 we give a detailed investigation of our approach over the unit circle, where eigenvalues and eigenfunctions of the Laplace-Beltrami operator are analytically known and manifold GPs are computable. The goal of this example is to show that our discrete GGP-UCB algorithm, which only requires point cloud data from the unit circle, achieves comparable performance to a UCB algorithm with manifold GPs. We also illustrate the parameter estimation technique discussed in Subsection 5.3.1. In Subsection 5.4.2 we consider an artificial manifold for which the spectrum of its Laplace-Beltrami operator is not available, showcasing a typical application of our framework when the manifold is only accessed through a point cloud. The goal of this example is to show the empirical advantage of using our geometry-informed GGPs over Euclidean GPs. Finally, in Subsection 5.4.3 we apply Algorithm 5.2.1 to solve an inverse problem —heat source detection over the sphere, which is only represented as a point cloud. Here the objective function is defined in terms of a partial differential equation that needs to be numerically solved. The goal of this example is to illustrate the applicability of our algorithm with expensive-to-evaluate objective functions that need to be approximated using graph-based techniques.

Throughout all the examples in Subsections 5.4.1 and 5.4.2, we set $\sigma = 0.05 \cdot \|f_N\|_2/\sqrt{N}$, which corresponds to a noise level of roughly 5%. We adopt the selection rule (5.27) and set $\delta = 0.1$ in the choice of $B_{N,\ell}$.

5.4.1 The Unit Circle

Let $\mathcal{M}$ be the unit circle in $\mathbb{R}^2$ and $\mathcal{M}_N = \{x_i\}_{i=1}^{N=500}$ be i.i.d. samples from the uniform distribution over $\mathcal{M}$. The fact that the eigenvalues and eigenfunctions of the Laplace-Beltrami operator are available in closed form allows us to carry out —for comparison purposes— computation on the continuum level. In particular, we can compute the manifold GP covariance functions defined in (5.13) and (5.14).

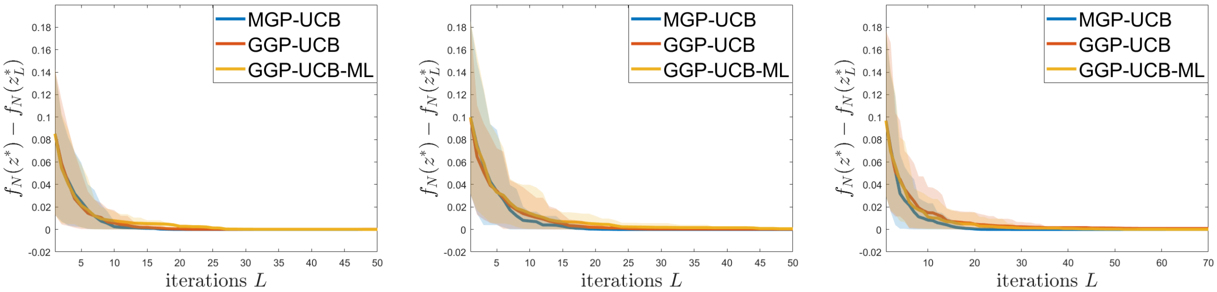
To start with, suppose first that f is a sample from the manifold Matérn GP (5.11) with parameters τ_* and s_*, which can be generated from (5.11) with a sufficiently high truncation. We shall compare the performance of Algorithm 5.2.1 with three different choices of the prior: (i) (5.11) with true parameters, (ii) (5.17) with true parameters, and (iii) (5.17) with inferred parameters, i.e.,

$$\text{(MGP-UCB)} \qquad u^{\mathcal{M}} = \kappa_*^{s_*-\frac{m}{2}} \sum_{i=1}^{K} (\kappa_*^2 + \lambda_i)^{-\frac{s_*}{2}} \xi_i \psi_i, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.28}$$

$$\text{(GGP-UCB)} \qquad u^{\mathcal{M}_N} = \kappa_*^{s_*-\frac{m}{2}} \sum_{i=1}^{k_N} (\kappa_*^2 + \lambda_{N,i})^{-\frac{s_*}{2}} \xi_i \psi_{N,i}, \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.29}$$

$$\text{(GGP-UCB-ML)} \quad u^{\text{MLE}} = \kappa_\ell^{s_\ell-\frac{m}{2}} \sum_{i=1}^{k_N} (\kappa_\ell^2 + \lambda_{N,i})^{-\frac{s_\ell}{2}} \xi_i \psi_{N,i}, \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.30}$$

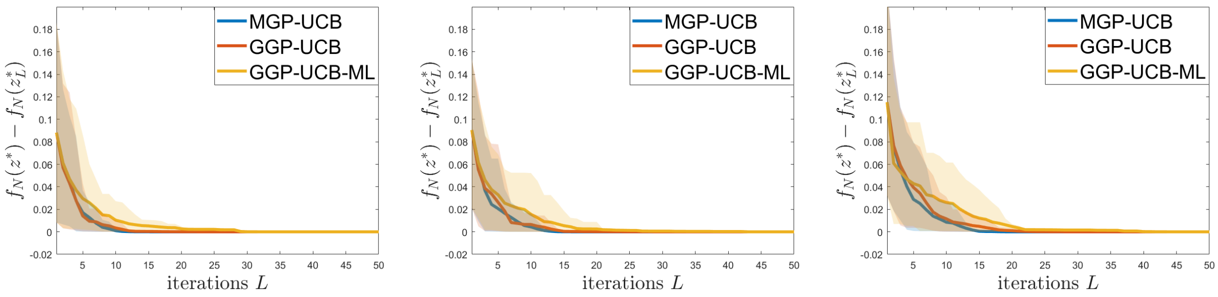
where $K = 100$ is a truncation for computing $u^{\mathcal{M}}$, and κ_ℓ and s_ℓ are the estimated parameters as discussed in Subsection 5.3.1. Specifically, we shall view MGP-UCB as an oracle algorithm whose performance serves as a benchmark, since for the graph-based algorithms we assume to be only given the point cloud $\{x_i\}_{i=1}^{N=500}$ and to have no access to the λ_i's and ψ_i's. We set $k_N = 20$ and $h_N = 4 \times N^{-1/2}$ in the construction of Δ_N.



(a) $\kappa_*^2 = 5, s_* = 2.$ (b) $\kappa_*^2 = 10, s_* = 2.$ (c) $\kappa_*^2 = 15, s_* = 2.$

Figure 5.2: Comparisons of the simple regrets obtained from MGP-UCB (prior with (5.28)), GGP-UCB (prior with (5.29)), and GGP-UCB-ML (prior with (5.30)) when f is a Matérn GP (5.11). The curves represent the average regrets over 50 trials and the shaded regions represent the $10\% \sim 90\%$ percentiles.

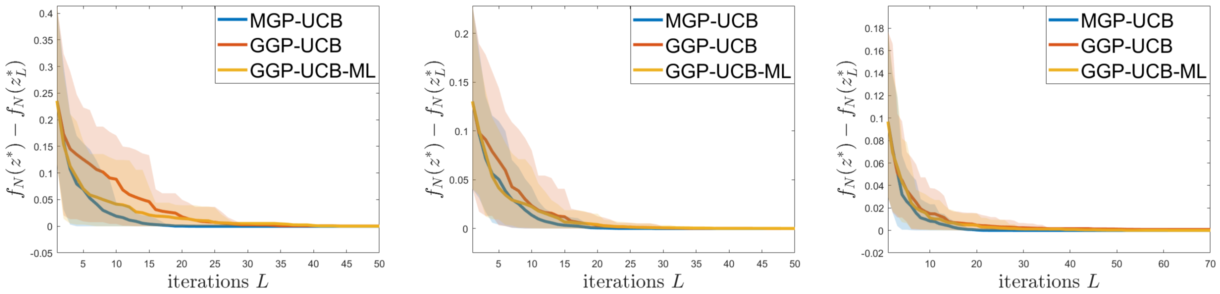
Figure 5.2 shows the comparison for three sets of parameters (κ_*, s_*), representing increasingly oscillatory true objective f. In all cases, the oracle MGP-UCB approach achieves the smallest regret, which is expected since it assumes complete knowledge of the unit circle. Meanwhile, the other two approaches show competitive performance and find the maximizer in less than $L = 50$ iterations, which is much smaller than the size $N = 500$ of the point cloud. In particular, incorporating maximum likelihood estimation of the parameters gives similar performance compared to the case when the parameters are assumed to be known. In a parallel setting, we also perform a similar comparison when the truth is a SE GP (5.12), where the graph SE GP (5.18) is used for modeling. Figure 5.3 shows the comparison, which is qualitatively similar to the Matérn case except that the approach incorporating maximum likelihood gives a slightly worse performance. Nevertheless, it is still able to find a near optimizer within 50 iterations.



(a) $\tau_* = 0.15$. (b) $\tau_* = 0.1$. (c) $\tau_* = 0.05$.

Figure 5.3: Comparisons of the simple regrets obtained from MGP-UCB (prior with (5.28)), GGP-UCB (prior with (5.29)), and GGP-UCB-ML (prior with (5.30)) when f is a SE GP (5.12). The curves represent the average regrets over 50 trials and the shaded regions represent the 10% $\sim$ 90% percentiles.

Next, we investigate the effect of the number N of point cloud samples on the algorithmic performance. We generate the truth from (5.28) as before and apply our graph-based algorithms with N=100, 300, 500 uniform samples from the unit circle. Figure 5.4 shows the results, suggesting improved performance as N increases, in agreement with the qualitative behavior predicted by our regret bounds in (5.24).



(a) $N = 100$. (b) $N = 300$. (c) $N = 500$.

Figure 5.4: Comparisons of the simple regrets obtained from MGP-UCB (prior with (5.28)), GGP-UCB (prior with (5.29)), and GGP-UCB-ML (prior with (5.30)) with different size N of the point cloud when f is a Matérn GP (5.11) with parameters $\kappa_*^2 = 15$ and $s_* = 2$. The curves represent the average regrets over 50 trials and the shaded regions represent the 10% $\sim$ 90% percentiles.

Finally, to further investigate the performance of our GGP-UCB algorithm, we consider optimizing three benchmark functions —the Levy, Ackley, and Rastrigin functions defined over the circle (with suitable rescaling):

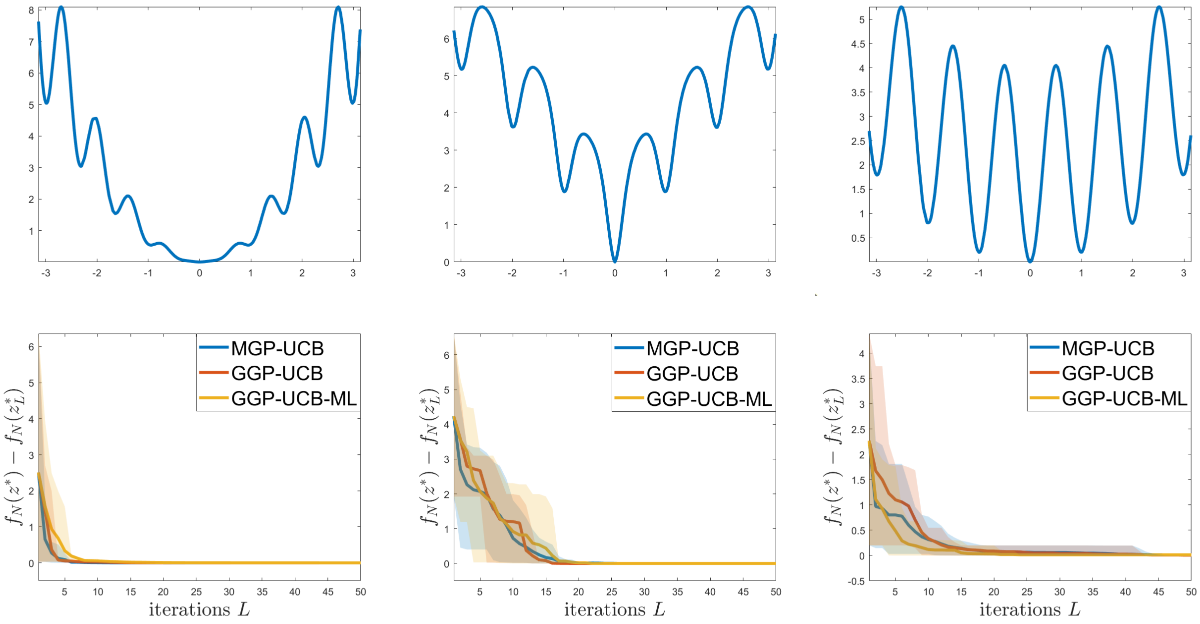
$$f^{\text{Levy}}(\theta) = \Big(\frac{3\theta}{4}\Big)^2\bigg(1+\sin^2\Big(\frac{\pi(3\theta+3)}{2}\Big)\bigg), \qquad \theta\in[-\pi,\pi), \tag{Levy}$$

$$f^{\text{Ackley}}(\theta) = -20\exp(-0.1\theta)-\exp(\cos(2\pi\theta))+20+\exp(1), \qquad \theta\in[-\pi,\pi), \tag{Ackley}$$

$$f^{\text{Rastrigin}}(\theta) = 2+\theta^2-2\cos(2\pi\theta), \qquad \theta\in[-\pi,\pi), \tag{Rastrigin}$$

where we identify points on the circle with their angle $\theta \in [-\pi, \pi)$. The top row of Figure 5.5 shows plots of the functions f^{Levy}, f^{Ackley}, and $f^{\text{Rastrigin}}$, all of which admit many sharp local minima. These benchmark functions will serve as examples where the truth to be optimized is not generated from a GP. As before, we shall compare the performance of Algorithm 5.2.1 with three different choices of prior (5.28), (5.29), (5.30), by manually setting $\kappa_* = 15$ and $s_* = 1$ for the first two. The results are shown in the bottom row of Figure 5.5, where all algorithms can find the global optimizer with very few iterations (much fewer than the total number $N = 500$ of the point cloud), including GGP-UCB-ML which infers the covariance parameters. This illustrates the applicability of our algorithm when the truth is not necessarily a sample path from the same GP model that we use for the algorithm.

Remark 5.4.1. We end this example with a remark on inferring the GP parameters with maximum likelihood. For the Matérn case, our experience suggests that joint estimation of κ_ℓ, s_ℓ for (5.30) turns out to be unstable, and hence in the simulations above we have fixed κ_ℓ to be 1 throughout and only estimated s_ℓ. Such an observation may be related to the fact that not all parameters for the Matérn model but only a certain combination of them are identifiable (see e.g. [Zhang, 2004, Bolin and Kirchner, 2020, Li et al., 2023]). This issue may be exacerbated by the fact that the graph Matérn GP we adopt is only an approximation of the Matérn model, and similarly for the squared exponential model. A detailed investigation of maximum likelihood for GGPs is an interesting direction for future research. Our focus on the remaining experiments will be however on illustrating other important aspects of our



(a) Levy. (b) Ackley. (c) Rastrigin.

Figure 5.5: Top row: plots of the Levy, Ackley, and Rastrigin functions. Bottom row: Comparisons of the simple regrets obtained from MGP-UCB (prior with (5.28)), GGP-UCB (prior with (5.29)), and GGP-UCB-ML (prior with (5.30)) for optimizing the three functions respectively. The curves represent the average regrets over 50 trials and the shaded regions represent the $10\% \sim 90\%$ percentiles.

GGP-UCB algorithm, and for this reason we henceforth assume the GP parameters to be known or tune them empirically. □

5.4.2 *Two-Dimensional Artificial Manifold*

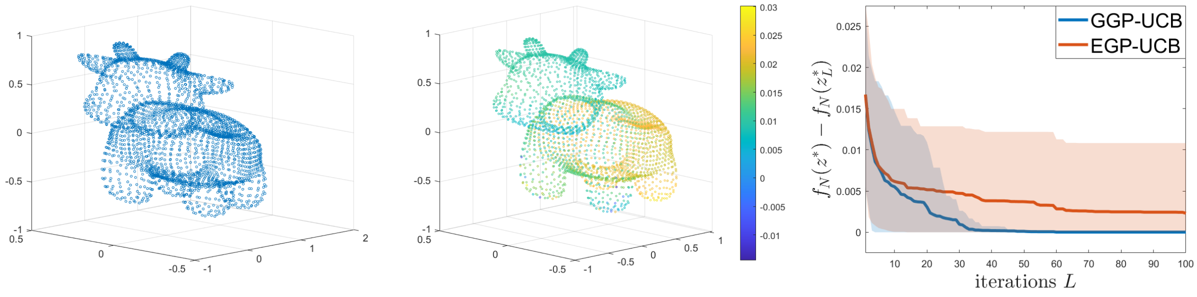
In this subsection we consider an artificial two-dimensional manifold, whose point cloud representation—taken from Keenan Crane's 3D repository [Crane]—is shown in Figure 5.6a. This example is motivated by an application to locate the point of highest temperature [Srinivas et al., 2010] on a surface where an explicit parameterization is not given. Unlike the unit circle case in Subsection 5.4.1, the eigenvalues and eigenfunctions of the Laplace-Beltrami operator over this new manifold are no longer known analytically, which prevents us from computing manifold GP covariances. The goal of this example is to demonstrate the superior performance of GGPs over Euclidean GPs.

First, we shall generate our truth using a finer point cloud than the one given for optimization. More precisely, the original dataset $\mathcal{M}_{\bar{N}}$ provided by [Crane] consists of $\bar{N} = 2930$ points, but we only assume to be given a subsample of $N = 2000$ points as our $\mathcal{M}_N$. The truth is then generated as a sample defined on the finer point cloud $\mathcal{M}_{\bar{N}}$:

$$f_{\bar{N}} = \kappa_*^{s_* - \frac{m}{2}} \sum_{i=1}^{k_{\bar{N}}} (\kappa_*^2 + \lambda_{\bar{N},i})^{-\frac{s_*}{2}} \xi_i \psi_{\bar{N},i}, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \tag{5.31}$$

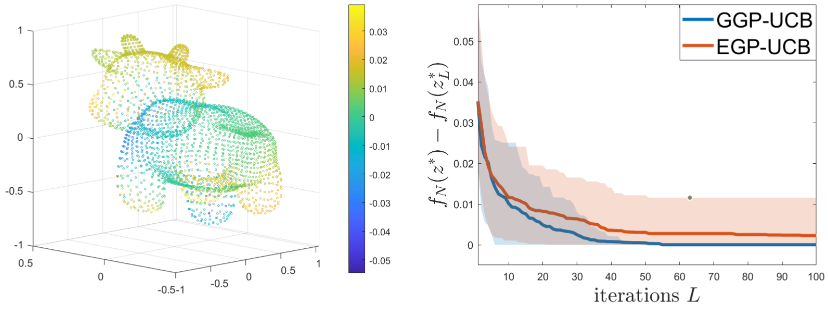
where $\lambda_{\bar{N},i}$ and $\psi_{\bar{N},i}$'s are the eigenpairs of the graph Laplacian $\Delta_{\bar{N}}$ constructed with all $\bar{N}$ points. Here the graph connectivity is taken to be $h_{\bar{N}} = 4 \times \bar{N}^{-1/2}$ and $k_{\bar{N}}$ is set to be 50 based on the eigenvalue saturation of $\Delta_{\bar{N}}$. Figure 5.6b shows one realization of $f_{\bar{N}}$ with parameters $\kappa_*^2 = 5$ and $s_* = 2.5$

Since the manifold GP is not available in this example, we shall compare the performance of Algorithm 5.2.1 with prior taken as a GGP (cf.(5.29) with graph connectivity $h_N = 4 \times N^{-1/2}$ and truncation $k_N = 50$) or a Euclidean GP (EGP). As the truth (5.31) is of Matérn type, it is natural to take the EGP as defined by the usual Matérn covariance function (5.9) by viewing points in $\mathcal{M}_N$ as elements of $\mathbb{R}^3$. As discussed in Remark 5.4.1, we shall use the true parameters in GGP modeling, but point out that the true parameters are not necessarily the ones that lead to the best performance since the truth is generated based on $\Delta_{\bar{N}}$, whose eigenpairs are only close to but different from those of Δ_N used for computation. For EGP modeling, we tune the parameters empirically and report the one that leads to the smallest regret. The results are presented in Figure 5.6c, suggesting that GGP modeling outperforms EGP and can find the optimizer with far fewer queries than the size $N = 2000$ of the given point cloud. In a parallel setup, Figure 5.7b compares the performance of GGP-UCB with EGP-UCB when the truth and the associated prior models are of squared exponential type (cf. (5.18) and (5.9)), where qualitatively similar behavior is observed.



(a) Point cloud. (b) Matérn GGP sample. (c) Simple regrets.

Figure 5.6: (a) Point cloud. (b) A random sample $f_{\bar{N}}$ defined as (5.31) with $\kappa_*^2 = 5, s_* = 2.5$; values of $f_{\bar{N}}$ vary smoothly along the point cloud. (c) Comparison of simple regrets as a function of L between GGP-UCB and EGP-UCB. The curves represent the average regrets over 50 trials and the shaded regions represent the $10\% \sim 90\%$ percentiles.



(a) SE GGP sample. (b) Simple regrets.

Figure 5.7: (a) A random sample from (5.18) based on the graph Laplacian $\Delta_{\bar{N}}$ with $\tau_* = 0.05$. (b) Comparison of simple regrets as a function of L between GGP-UCB and EGP-UCB. The curves represent the average regrets over 50 trials and the shaded regions represent the $10\% \sim 90\%$ percentiles.

5.4.3 Heat Source Detection on the Sphere

In this subsection we employ Algorithm 5.2.1 on a heat source detection problem on the two-dimensional unit sphere S^2, which is given only as a point cloud. The goal of this example is to demonstrate the applicability of our BO framework in inverse problem settings, where the objective function to be optimized involves an expensive-to-evaluate *forward map* that usually needs to be approximated.

Consider the heat equation

$$\begin{cases} \varphi_t = \Delta_{S^2}\varphi, & (x,t) \in S^2 \times [0,\infty), \\ \varphi(x,0) = \varphi_0(x), & x \in S^2, \end{cases} \tag{5.32}$$

where Δ_{S^2} is the Laplace-Beltrami operator on S^2 and φ_0 is an initial heat configuration. The solution of the heat equation for some time $t > 0$ is given by

$$\varphi(x,t) = \sum_{i=1}^{\infty} \langle \varphi_0, \psi_i \rangle_{S^2} \cdot e^{-\lambda_i t} \psi_i(x), \quad x \in S^2, \tag{5.33}$$

where $\{(\lambda_i, \psi_i)\}_{i=1}^{\infty}$ are the eigenpairs of $-\Delta_{S^2}$ and $\langle \cdot, \cdot \rangle_{S^2}$ is the Riemannian inner product associated to S^2. The initial heat configuration is given by

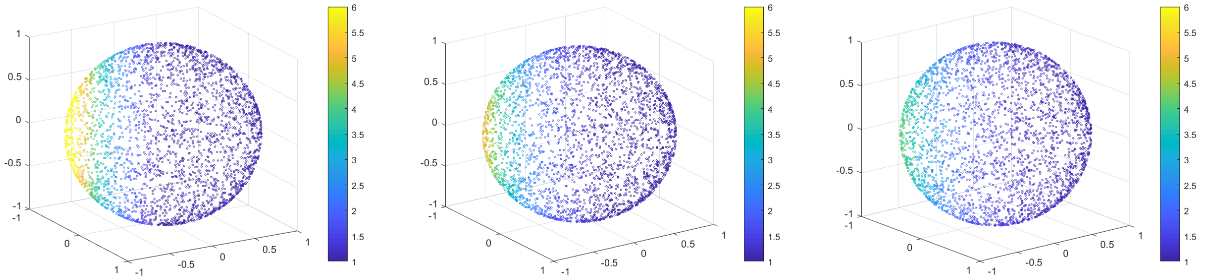
$$\varphi_0(x) = \exp\left(\zeta {z^*}^\top x\right), \quad \zeta > 0,\ x \in S^2, \tag{5.34}$$

which can be viewed as an unnormalized density of the von-Mises Fisher distribution [Fisher, 1953] on S^2. A larger concentration parameter ζ leads to more probability mass centered around its mean z^*.

Our goal is to recover the center z^* of the initial heat configuration, assuming we are only given a point cloud $\mathcal{M}_N = \{x_i\}_{i=1}^{N=3000}$ but not $\mathcal{M}$ directly, and noisy heat measurements at some positive time t of the form

$$\mathsf{d} = \varphi(\mathbf{x}) + \eta, \qquad \varphi(\mathbf{x}) = \big(\varphi(x_1,t), \ldots, \varphi(x_N,t)\big)^\top,$$

where $\eta \sim \mathcal{N}(0, 0.01 I_N)$. To generate $\varphi(x,t)$, we truncate (5.33) at $i = 36$, by keeping only the terms with $\lambda_i \leq 30$ (the sixth repeated eigenvalue of $-\Delta_{S^2}$). Figure 5.8 contains plots



(a) $t = 0$.

(b) $t = 0.25$.

(c) $t = 0.4$.

Figure 5.8: (a) Initial heat over the point cloud. (b) Noisy evaluation of heat at $t = 0.25$. (c) Noisy evaluation of heat at $t = 0.4$.

of an example of initial heat configuration with $\zeta = 2$ and the corresponding noisy data for times $t = 0.25$ and $t = 0.4$. Assuming that the center $z^* \in \mathcal{M}_N$, we adopt an optimization perspective to this inverse problem [Sanz-Alonso et al., 2023] and attempt to maximize the objective function

$$f(z) = -\log \|\mathsf{d} - \mathcal{G}(z)\|_\infty, \qquad z \in \mathcal{M}_N,$$

along the point cloud $\mathcal{M}_N$, where $\mathcal{G}(z) \in \mathbb{R}^N$ is the forward map given by

$$[\mathcal{G}(z)]_k = \sum_{i=1}^{\infty} \langle \varphi_0^z, \psi_i \rangle_{S^2} \cdot e^{-\lambda_i t} \psi_i(x_k), \qquad x_k \in \mathcal{M}_N, \tag{5.35}$$

with $\varphi_0^z(x) = \exp\left(\zeta z^\top x\right)$ for $x \in S^2$. However, since $\mathcal{M}$ is only known through $\mathcal{M}_N$, the eigenvalue and eigenfunctions should be also treated as unknown to us. Therefore, we shall instead maximize the approximate objective function

$$f_N(z) = -\log \|\mathsf{d} - \mathcal{G}_N(z)\|_\infty, \qquad z \in \mathcal{M}_N,$$

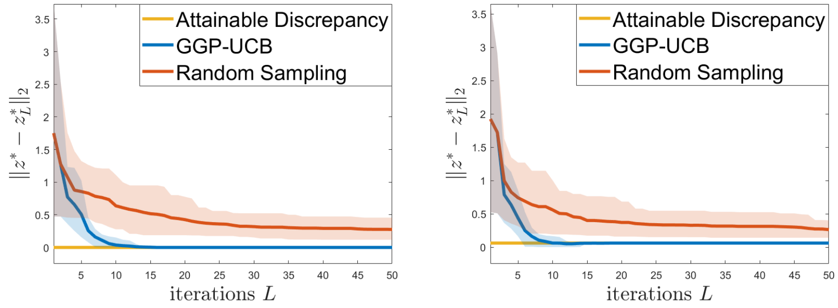
where

$$\mathcal{G}_N(z) = \sum_{i=1}^{k_N} \langle \varphi_{0,N}^z, \psi_{N,i} \rangle \cdot e^{-\lambda_{N,i} t} \psi_{N,i}, \quad \varphi_{0,N}^z = \Big(\exp\big(\zeta z^\top x_1\big), \cdots, \exp\big(\zeta z^\top x_N\big)\Big)^\top$$

with the hope that the optimizer of f_N agrees with, or at least is close to, that of f. Here, as before, $\{(\lambda_{N,i}, \psi_{N,i})\}_{i=1}^N$ are eigenpairs of the unnormalized graph Laplacian and $\langle \cdot, \cdot \rangle$ is the standard Euclidean inner product. For the truncation level, we set $k_N = 70$ to account for the discrepancy —shown in Figure 5.1— between the spectrum of the graph Laplacian and that of the negative Laplace-Beltrami operator.

To optimize f_N, we apply Algorithm 5.2.1 with a graph Matérn prior (5.29) with parameters $s = 4$, $\kappa = 1$. There is no observation noise in this case since f_N can be computed exactly, so that $\mu_{N,\ell}$ and $\sigma_{N,\ell}$ in the acquisition function will be computed using (5.8) with $\sigma = 0$ and $Y_\ell = (f_N(z_1), \cdots, f_N(z_\ell))^\top$. Since we are interested in the recovery of z^*, we shall report the distance measure $\|z^* - z_L^*\|_2$, where z_L^* is the query point returned by GGP-UCB or random sampling that maximizes f_N in the first L iterations. The results are shown in Figure 5.9 for observations d collected at two different times $t = 0.25$ and $t = 0.4$. Qualitatively similar performance as in previous examples is achieved. However, notice that in Figure 5.9b the recovery is not exact, as the distance $\|z^* - z_L^*\|_2$ does not decrease to zero. This is because we are searching for the maximizer of the approximate objective f_N, which differs from the true heat source z^* when t is large due to the approximation error of $\mathcal{G}_N$ to $\mathcal{G}$. In other words, the attainable discrepancy, defined as the distance between z^* and the maximizer of f_N, is nonzero in this case. Besides this effect caused by an error in the approximation of the objective, the simulation results suggest that our GGP-UCB algorithm correctly finds the maximizer of the approximate objective f_N with a significantly smaller number L of queries than the total number N of points in $\mathcal{M}_N$.

We remark that there are two intertwined aspects which make source detection difficult for intermediate to large t values. The smoothing effect of the forward map $\mathcal{G}$ implies that a larger observation time will lead to a more flattened (homogeneous) temperature configuration, as shown in Figure 5.8. In other words, two rather different initial heat configurations will yield almost identical heat configurations after a large time $t > 0$. Such ill-posedness



(a) $t = 0.25$.

(b) $t = 0.4$.

Figure 5.9: Recovery error $\|z^* - z^*_L\|_2$, where z^* is the true source in (5.34) and z^*_L is the query point returned by GGP-UCB or random sampling that maximizes f_N in the first L iterations. Heat measurements are collected at times (a) $t = 0.25$ and (b) $t = 0.4$. The curves represent the average regrets over 50 trials and the shaded regions represent the 10% $\sim$ 90% percentiles.

hinders the recovery of the true heat source location for large t. In addition, the forward map $\mathcal{G}$ and its approximation $\mathcal{G}_N$ are defined in terms of an exponential transformation of the eigenvalues of the Laplace-Beltrami operator and the graph Laplacian. Therefore, for moderate t, any small inaccuracy in the eigenvalue estimation can lead to significantly different forward models $\mathcal{G}$ and $\mathcal{G}_N$, so that f_N is a poor approximation to f. If one had access to the true forward map, this issue would not be present.

5.5 Discussion

This chapter introduced GGP-UCB, a manifold learning technique to optimize an objective function on a hidden compact manifold. Our regret bounds and numerical experiments demonstrate the effectiveness of our method.

Curse of Dimensionality Similar to other Bayesian nonparametric techniques, we expect GGP-UCB to be particularly effective when the dimension m of the manifold $\mathcal{M} \subset \mathbb{R}^d$ is small or moderate. In particular, our regret bounds in Theorem 5.2.5 suffer from the standard curse of dimension with m, while they do not depend on the dimension d of the ambient space.

Estimating the Intrinsic Dimension For simplicity, we have assumed that the dimension m of $\mathcal{M}$ is known, that we have access to clean samples from $\mathcal{M}$, and that $\mathcal{M}$ has no boundary. If the dimension of $\mathcal{M}$ is unknown, classical manifold learning techniques can be used to estimate it [Hein and Audibert, 2005, Harlim et al., 2020]. Similarly, if the given point cloud is noisy in that it consists of random perturbations of points sampled from $\mathcal{M}$, a denoising method can be employed to uncover the underlying geometry [Garcia Trillos et al., 2019]. Finally, if $\mathcal{M}$ has a boundary, our GGP-UCB method may be combined with a ghost point diffusion map algorithm to remove boundary artifacts [Harlim et al., 2022, Peoples and Harlim, 2021, Jiang and Harlim, 2023].

Other Acquisition Functions Our focus on UCB acquisition functions was motivated by the desire to establish convergence guarantees under misspecification, as well as by their simplicity and successful empirical performance. However, there is no algorithmic roadblock to employ other acquisition functions such as expected improvement and Thompson sampling. An interesting direction for future research is to investigate how to provably correct for geometric misspecification when using these alternative acquisition functions.

Beyond the Manifold Setting We have focused on GGP surrogate models defined via a specific choice of unnormalized graph-Laplacian; other graph constructions (e.g. based on nearest neighbors or self-tuning kernels) and graph-Laplacian (e.g. symmetric and random walk) could be considered [Von Luxburg, 2007]. Furthermore, the proposed BO framework can be extended beyond the manifold setting. In particular, similar constructions of the GGPs can be carried out over any point cloud (not necessarily embedded in a Euclidean space) as long as a graph Laplacian encoding pairwise similarities of the point cloud can be formed [Sanz-Alonso and Yang, 2022a, Borovitskiy et al., 2021]. Together with suitable choices of acquisition functions, the resulting framework can be used to solve discrete optimization problems by endowing the search space with a graph structure, which could facilitate the search of optimizers. This is an interesting direction for future research.

5.6 Appendix

These appendices contain the proofs of Proposition 5.2.3, Theorem 5.2.5, and Corollary 5.2.7. The proofs build on the theory of spectral convergence of graph Laplacians and regret analysis of Bayesian optimization algorithms. To make our presentation self-contained, we will introduce necessary background and previous results whenever needed.

5.6.1 Proof of Proposition 5.2.3

Let $\mathcal{M}_N$ be i.i.d. samples from a distribution μ supported on a smooth, connected, and compact m-dimensional submanifold $\mathcal{M} \subset \mathbb{R}^d$ without boundary. For simplicity, we shall assume that μ is the uniform distribution on $\mathcal{M}$. The first result [Garcia Trillos et al., 2020a, Theorem 2] states that with high probability, the x_i's form a ρ_N-net over $\mathcal{M}$ and characterizes ρ_N.

Proposition 5.6.1. *For any $c > 1$, with probability $1-O(N^{-c})$, there exists a transportation map $T_N : \mathcal{M} \to \{x_1, \cdots, x_N\}$ so that*

$$\rho_N := \sup_{x\in\mathcal{M}} d_{\mathcal{M}}\big(x, T_N(x)\big) \lesssim \frac{(\log N)^{p_m}}{N^{1/m}}, \tag{5.36}$$

where $p_m = 3/4$ when $m = 2$ and $p_m = 1/m$ otherwise. We recall that $d_{\mathcal{M}}$ is the geodesic distance on $\mathcal{M}$.

Proposition 5.6.1 implies that the point cloud $\mathcal{M}_N$ is "well-structured" with high probability and is the building block for the spectral approximation results below [Sanz-Alonso and Yang, 2022b, Proposition 10 and Lemma 15]. Recall that the graph-Laplacian Δ_N constructed in Subsection 5.2.3 admits a spectral decomposition, in analogy to the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$.

Proposition 5.6.2. *Suppose there exists $\delta > 0$ such that, for N sufficiently large,*

$$h_N \gtrsim N^{-\frac{1}{m+4+\delta}}, \qquad k_N \lesssim N^{\frac{1-\delta}{m}}, \qquad h_N k_N^{\frac{2}{m}} \lesssim 1. \tag{5.37}$$

Then with probability $1-O(N^{-c})$ for some $c > 0$, there exists orthonormalized eigenfunctions $\{\psi_{N,i}\}_{i=1}^N$ for Δ_N, $\{\psi_i\}_{i=1}^\infty$ for $\Delta_{\mathcal{M}}$, and $T_N : \mathcal{M} \to \{x_1, \ldots, x_N\}$ satisfying $T_N(x_i) = x_i$ so that, for $i = 1, \ldots, k_N$,

$$\begin{aligned} |\lambda_{N,i} - \lambda_i| &\lesssim \lambda_i \left(\frac{\rho_N}{h_N} + h_N \sqrt{\lambda_i} \right), \\ \|\psi_{N,i} \circ T_N - \psi_i\|_\infty &\lesssim \lambda_i^{m+1} i^{\frac{3}{2}} \sqrt{\frac{\rho_N}{h_N} + h_N \sqrt{\lambda_i}}\ . \end{aligned}$$

We also need a result on the growth of the L^∞-norm of the Laplace-Beltrami eigenfunctions and their gradients from [Donnelly, 2006, Theorem 1.2] and [Xu, 2006, equation (2.10)].

Proposition 5.6.3. *Let ψ be an L^2-normalized eigenfunction of $-\Delta_{\mathcal{M}}$ associated with $\lambda \neq 0$. Then $\|\psi\|_\infty \le C\lambda^{(m-1)/4}$ and $\|\nabla\psi\|_\infty \le C\lambda^{(m+1)/2}$ for a universal constant C.*

Lemma 5.6.4. *The random field u^{SE} defined in* (5.12) *satisfies $\mathbb{E}\|u^{SE}\|_{L^2}^2 \asymp 1$ and has a modification that is locally Hölder continuous of order α for all $\alpha < \frac{1}{2}$. The random field u^{Ma} defined in* (5.11) *has a modification that is locally Hölder continuous of order γ for all $\gamma < \frac{2s-2m+1}{m+3} \wedge \frac{1}{2}$.*

Proof. By Weyl's law that $\lambda_i \asymp i^{\frac{2}{m}}$ (see e.g. [Canzani, 2013, Theorem 72]), we have

$$\mathbb{E}\left\|\sum_{i=1}^{\infty} e^{-\frac{\lambda_i \tau}{2}} \xi_i \psi_i\right\|_{L^2}^2 = \sum_{i=1}^{\infty} e^{-\lambda_i \tau} \asymp \sum_{i=1}^{\infty} e^{-Ci^{2/m}\tau} \asymp \int_0^{\infty} e^{-Cx^{2/m}\tau}\, dx.$$

By a change of variable, the last expression is equal to

$$\frac{m}{2}\int_0^{\infty} e^{-C\tau x} x^{\frac{m}{2}-1}\, dx = \frac{m}{2} \cdot \frac{\Gamma(\frac{m}{2})}{(C\tau)^{m/2}} \asymp \tau^{-\frac{m}{2}}.$$

For the second claim, by [Lang et al., 2016, Corollary 4.5] it suffices to show that

$$\mathbb{E}|u^{\text{SE}}(x) - u^{\text{SE}}(y)|^2 \le C d_{\mathcal{M}}(x,y)^{\eta} \tag{5.38}$$

for all $\eta \in (0,1)$, $C > 0$, and for all x, y satisfying $d_{\mathcal{M}}(x,y) < 1$. Notice that

$$\begin{aligned}
\mathbb{E}|u^{\text{SE}}(x) - u^{\text{SE}}(y)|^2 &= \tau^{\frac{m}{2}} \mathbb{E}\Big|\sum_{i=1}^{\infty} e^{-\frac{\lambda_i \tau}{2}} \xi_i (\psi_i(x) - \psi_i(y))\Big|^2 \\
&= \tau^{\frac{m}{2}} \sum_{i=1}^{\infty} e^{-\lambda_i \tau} |\psi_i(x) - \psi_i(y)|^2 \\
&\le \tau^{\frac{m}{2}} \sum_{i=1}^{\infty} e^{-\lambda_i \tau} \|\nabla \psi_i\|_{\infty}^2 d_{\mathcal{M}}(x,y)^2 \\
&\le C\tau^{\frac{m}{2}} \sum_{i=1}^{\infty} e^{-\lambda_i \tau} \lambda_i^{m+1} d_{\mathcal{M}}(x,y)^2,
\end{aligned}$$

where we have used Proposition 5.6.3 in the last step. Now by Weyl's law,

$$\sum_{i=1}^{\infty} e^{-\lambda_i \tau} \lambda_i^{m+1} \le C \sum_{i=1}^{\infty} e^{-c\tau i^{2/m}} i^{\frac{2(m+1)}{m}} \le C \int_1^{\infty} e^{-c\tau x^{2/m}} x^{\frac{2(m+1)}{m}}\, dx < \infty.$$

Therefore,

$$\mathbb{E}|u^{\text{SE}}(x) - u^{\text{SE}}(y)|^2 \leq C d_{\mathcal{M}}(x,y)^2 \leq C d_{\mathcal{M}}(x,y)^{\eta}$$

for any $\eta \in (0,1)$ when $d_{\mathcal{M}}(x,y) < 1$, thereby verifying (5.38).

To show local Hölder continuity of u^{Ma}, we need a more careful analysis. Similarly as above, we have

$$\begin{aligned}\mathbb{E}|u^{\text{Ma}}(x) - u^{\text{Ma}}(y)|^2 &= \kappa^{2s-m}\mathbb{E}\Big|\sum_{i=1}^{\infty}(\kappa^2+\lambda_i)^{-\frac{s}{2}}\xi_i(\psi_i(x)-\psi_i(y)\Big|^2 \\ &= \kappa^{2s-m}\sum_{i=1}^{\infty}(\kappa^2+\lambda_i)^{-s}|\psi_i(x)-\psi_i(y)|^2.\end{aligned}$$

Now by Proposition 5.6.3, we shall control $|\psi_i(x) - \psi_(y)|$ by the smaller quantity of the following two bounds

$$\begin{aligned}|\psi_i(x)-\psi_i(y)| &\leq C\lambda_i^{\frac{m-1}{4}}, \\ |\psi_i(x)-\psi_i(y)| &\leq C\lambda_i^{\frac{m+1}{2}} d_{\mathcal{M}}(x,y).\end{aligned}$$

Precisely, we have

$$\begin{aligned}\mathbb{E}|u^{\text{Ma}}(x) - u^{\text{Ma}}(y)|^2 &\leq C\sum_{i=1}^{\infty}(\kappa^2+\lambda_i)^{-s}\min\left\{\lambda_i^{\frac{m-1}{2}}, \lambda_i^{m+1} d_{\mathcal{M}}(x,y)^2\right\} \\ &\leq C\sum_{i=1}^{\infty} i^{-\frac{2s}{m}}\min\left\{i^{\frac{m-1}{m}}, i^{\frac{2m+2}{m}} d_{\mathcal{M}}(x,y)^2\right\} \\ &\leq C\sum_{i=1}^{K} i^{-\frac{2s}{m}} i^{\frac{2m+2}{m}} d_{\mathcal{M}}(x,y)^2 + C\sum_{i=K+1}^{\infty} i^{-\frac{2s}{m}} i^{\frac{m-1}{m}}, \end{aligned} \tag{5.39}$$

where $K = d_{\mathcal{M}}(x,y)^{-\frac{2m}{m+3}}$. Therefore we have

$$\begin{aligned}(5.39) &\leq C d_{\mathcal{M}}(x,y)^2 \int_1^K z^{\frac{-2s+2m+2}{m}} dz + C \int_K^\infty z^{\frac{-2s+m-1}{m}} dz \\ &\leq C d_{\mathcal{M}}(x,y)^{\frac{4s-4m+2}{m+3}} \leq C d_{\mathcal{M}}(x,y).\end{aligned}$$

The result follows again by [Lang et al., 2016, Corollary 4.5].

□

Now we are ready to prove Proposition 5.2.3. The first statement on the approximation error of u_N^{Ma} follows from [Sanz-Alonso and Yang, 2022b, Theorem 4.6]. To show the second, recall that

$$\begin{aligned} u_N^{\text{SE}} &= \tau^{\frac{m}{4}} \sum_{i=1}^{k_N} e^{-\frac{\lambda_{N,i}\tau}{2}} \xi_i \psi_{N,i}, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \\ u^{\text{SE}} &= \tau^{\frac{m}{4}} \sum_{i=1}^{\infty} e^{-\frac{\lambda_i\tau}{2}} \xi_i \psi_i, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \end{aligned}$$

and introduce two intermediate random processes

$$\begin{aligned} \widetilde{u}_N^{\text{SE}} &= \tau^{\frac{m}{4}} \sum_{i=1}^{k_N} e^{-\frac{\lambda_i\tau}{2}} \xi_i \psi_{N,i}, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1), \\ \widehat{u}_N^{\text{SE}} &= \tau^{\frac{m}{4}} \sum_{i=1}^{k_N} e^{-\frac{\lambda_i\tau}{2}} \xi_i \psi_i, \qquad \xi_i \overset{i.i.d.}{\sim} \mathcal{N}(0,1). \end{aligned}$$

We then have

$$\mathbb{E}\|u_N^{\text{SE}} \circ T_N - u^{\text{SE}}\|_\infty \leq \mathbb{E}\|u_N^{\text{SE}} \circ T_N - \widetilde{u}_N^{\text{SE}} \circ T_N\|_\infty + \mathbb{E}\|\widetilde{u}_N^{\text{SE}} \circ T_N - \widehat{u}_N^{\text{SE}}\|_\infty + \mathbb{E}\|\widehat{u}_N^{\text{SE}} - u^{\text{SE}}\|_\infty$$

and we shall proceed by bounding each of the three terms on the right. First, note that

$$\begin{aligned}\mathbb{E}\|\widehat{u}_N^{\mathrm{SE}} - u^{\mathrm{SE}}\|_\infty &= \mathbb{E}\Bigg\|\tau^{\frac{m}{4}} \sum_{i=k_N+1}^{\infty} e^{-\frac{\lambda_i\tau}{2}}\xi_i\psi_i\Bigg\|_\infty \\ &\lesssim \sum_{i=k_N+1}^{\infty} e^{-\frac{\lambda_i\tau}{2}}\mathbb{E}|\xi_i|\|\psi_i\|_\infty \lesssim \sum_{i=k_N+1}^{\infty} e^{-\frac{\lambda_i\tau}{2}}\lambda_i^{\frac{m-1}{4}}, \end{aligned} \tag{5.40}$$

where we have used Proposition 5.6.3 in the last step. Now by Weyl's law, $\lambda_i \asymp i^{2/m}$ so that we can further bound

$$\begin{aligned}(5.40) \lesssim \sum_{i=k_N+1}^{\infty} e^{-c_0\tau i^{2/m}} i^{\frac{2}{m}\frac{m-1}{4}} &\lesssim \int_{k_N}^{\infty} e^{-c_0\tau x^{2/m}} x^{\frac{m-1}{2m}}\, dx \\ &= \int_{k_N^{2/m}}^{\infty} e^{-c_0\tau z} z^{\frac{3m-1}{4}-1} dz \end{aligned} \tag{5.41}$$

after a change of variable, where c_0 is a universal constant. Notice that the rightmost term (5.41) is equal up to a multiplicative constant to $\mathbb{P}(X \geq k_N^{2/m})$ with X being a Gamma random variable with shape parameter $\frac{3m-1}{4}$ and scale parameter $\frac{1}{c_0\tau}$. Now by the tail bound of sum-Gamma distributions (cf. [Zhang and Chen, 2020, Lemma 5.1]) applied to $X - \mathbb{E}X \in \mathrm{sub}\Gamma\Big(\frac{3m-1}{4c_0^2\tau^2}, \frac{1}{c_0\tau}\Big)$, we have

$$\mathbb{E}\|\widehat{u}_N^{\mathrm{SE}} - u^{\mathrm{SE}}\|_\infty \lesssim (5.41) \lesssim \mathbb{P}(X - \mathbb{E}X \geq k_N^{2/m} - \mathbb{E}X) \lesssim e^{-Ck_N^{2/m}} \tag{5.42}$$

for some constant C when $k_N^{2/m} \gg \mathbb{E}X = \frac{3m-1}{4c_0\tau}$. Similarly, we have

$$\mathbb{E}\|u_N^{\mathrm{SE}} \circ T_N - \widetilde{u}_N^{\mathrm{SE}} \circ T_N\|_\infty \lesssim \sum_{i=1}^{k_N} \Big|e^{-\frac{\lambda_{N,i}\tau}{2}} - e^{-\frac{\lambda_i\tau}{2}}\Big|\|\psi_{N,i} \circ T_N\|_\infty. \tag{5.43}$$

By the mean value theorem, we have that $|e^{-x} - e^{-y}| = e^{-\zeta}|x-y| \leq \max\{e^{-x}, e^{-y}\}|x-y|$

for some $\zeta \in \big(\min(x,y), \max(x,y)\big)$ where $x, y > 0$. Thus, we have

$$\left|e^{-\frac{\lambda_{N,i}\tau}{2}} - e^{-\frac{\lambda_i\tau}{2}}\right| \leq \max\Big\{e^{-\frac{\lambda_{N,i}\tau}{2}}, e^{-\frac{\lambda_i\tau}{2}}\Big\}\frac{\tau}{2}|\lambda_{N,i} - \lambda_i| \leq \frac{\tau}{2}e^{-\frac{\lambda_i\tau}{4}}\lambda_i\left(\frac{\rho_N}{h_N} + h_N\sqrt{\lambda_i}\right),$$

where in the last step we have used Proposition 5.6.2 which also implies $\lambda_{N,i} \geq \lambda_i/2$ when N is large. Moreover, Proposition 5.6.2 implies that, for $i = 1, \ldots, k_N$,

$$\begin{aligned}\|\psi_{N,i} \circ T_N\|_\infty &\leq \|\psi_{N,i} \circ T_N - \psi_i\|_\infty + \|\psi_i\|_\infty \\ &\lesssim \lambda_i^{m+1} i^{\frac{3}{2}}\sqrt{\frac{\rho_N}{h_N} + h_N\sqrt{\lambda_i}} + \|\psi_i\|_\infty \lesssim \Big(\frac{\rho_N}{h_N} + h_N\Big)i^{\frac{7m+5}{2m}} + \|\psi_i\|_\infty. \quad (5.44)\end{aligned}$$

Proposition 5.6.3 implies that $\|\psi_i\|_\infty \lesssim \lambda_i^{\frac{m-1}{4}} \lesssim i^{\frac{m-1}{2m}}$. Therefore we would like to set h_N and k_N to satisfy

$$\Big(\frac{\rho_N}{h_N} + h_N\Big)k_N^{\frac{7m+5}{2m}} \lesssim k_N^{\frac{m-1}{2m}} \tag{5.45}$$

so that (5.44) grows like $\|\psi_i\|_\infty$ for all $i = 1, \ldots, k_N$. We shall keep (5.45) in mind together with those conditions in (5.37) and proceed by assuming that we have made such choices. Now we can bound

$$\begin{aligned}(5.43) &\lesssim \sum_{i=1}^{k_N}\frac{\tau}{2}e^{-\frac{\lambda_i\tau}{4}}\lambda_i^{\frac{m+3}{4}}\left(\frac{\rho_N}{h_N} + h_N\sqrt{\lambda_i}\right) \\ &\lesssim \frac{\tau}{2}\left(\frac{\rho_N}{h_N} + h_N\right)\sum_{i=1}^{k_N}e^{-\frac{\lambda_i\tau}{4}}\lambda_i^{\frac{m+5}{4}} \lesssim \frac{\rho_N}{h_N} + h_N, \quad (5.46)\end{aligned}$$

where we used the fact that

$$\sum_{i=1}^{k_N}e^{-\frac{\lambda_i\tau}{4}}\lambda_i^{\frac{m+5}{4}} \lesssim \sum_{i=1}^{k_N}e^{-Ci^{2/m}}i^{\frac{m+5}{2m}} \lesssim \int_1^\infty e^{-Cx^{2/m}}x^{\frac{m+5}{2m}}\,dx < \infty.$$

Lastly, we have by Proposition 5.6.2

$$\begin{aligned}\mathbb{E}\|\widetilde{u}_N^{\text{SE}} \circ T_N - \widehat{u}_N^{\text{SE}}\|_\infty &\lesssim \sum_{i=1}^{k_N} e^{-\frac{\lambda_i \tau}{2}} \|\psi_{N,i} \circ T_N - \psi_i\|_\infty \\ &\lesssim \sum_{i=1}^{k_N} e^{-\frac{\lambda_i \tau}{2}} \lambda_i^{m+1} i^{\frac{3}{2}} \sqrt{\frac{\rho_N}{h_N} + h_N \sqrt{\lambda_i}} \lesssim \sqrt{\frac{\rho_N}{h_N} + h_N} \,. \qquad (5.47)\end{aligned}$$

Combining (5.42), (5.46),(5.47), we get

$$\mathbb{E}\|u_N^{\text{SE}} \circ T_N - u^{\text{SE}}\|_\infty \lesssim e^{-Ck_N^{2/m}} + \sqrt{\frac{\rho_N}{h_N} + h_N} \,.$$

Now it remains to set h_N and k_N and we remark that the approximation error will be dominated by the second term $\sqrt{\rho_N/h_N + h_N}$ when N is large. It can be checked that the following scaling satisfies the conditions imposed by (5.37) and (5.45).

Case 1: $m \le 4$ Setting for some arbitrarily small $\delta > 0$

$$h_N \asymp N^{-\frac{1}{m+4+\delta}} (\log N)^{\frac{p_m}{2}}, \qquad (\log N)^{\frac{m}{2}} \ll k_N \ll N^{\frac{m}{(m+4+\delta)(3m+3)}} (\log N)^{-\frac{mp_m}{6m+6}},$$

we obtain that, for large N,

$$\mathbb{E}\|u_N^{\text{SE}} \circ T_N - u^{\text{SE}}\|_\infty \lesssim N^{-\frac{1}{2(m+4+\delta)}} (\log N)^{\frac{p_m}{4}} .$$

Case 2: $m \ge 5$ Setting

$$h_N \asymp N^{-\frac{1}{2m}} (\log N)^{-\frac{p_m}{2}}, \qquad (\log N)^{\frac{m}{2}} \ll k_N \ll N^{\frac{1}{6m+6}} (\log N)^{-\frac{mp_m}{6m+6}},$$

we obtain

$$\mathbb{E}\|u_N^{\mathrm{SE}} \circ T_N - u^{\mathrm{SE}}\|_\infty \lesssim N^{-\frac{1}{4m}} (\log N)^{\frac{pm}{4}}.$$

5.6.2 Proof of Theorem 5.2.5

We start by introducing the key ingredients of the regret analysis of Bayesian optimization algorithms, in particular the GGP-UCB algorithm. Most of the preliminary results in this section can be found in [Srinivas et al., 2010, Bogunovic and Krause, 2021].

Recall that our goal is to bound the simple regret defined as in (5.4). But a typical strategy in the BO literature is to look at the *cumulative regret*, defined as

$$R_{N,L} = \sum_{\ell=1}^{L} \Big(f(z^*) - f(z_\ell) \Big), \qquad z^* = \arg\max_{z \in \mathcal{M}_N} f(z). \tag{5.48}$$

Then using the fact that

$$f(z_L^*) \geq \frac{1}{L} \sum_{\ell=1}^{L} f(z_\ell), \qquad z_L^* = \arg\max_{z \in \{z_\ell\}_{\ell=1}^{L}} f(z),$$

one can bound the simple regret as

$$r_{N,L} = f(z^*) - f(z_L^*) \leq \frac{1}{L} \sum_{\ell=1}^{L} \Big(f(z^*) - f(z_\ell) \Big) = \frac{R_{N,L}}{L}. \tag{5.49}$$

The key to bounding the cumulative regret consists of two steps. The first is a concentration-type result that constructs confidence bands which f lies in with high probability based on the observed samples. More precisely, we have the following result.

Lemma 5.6.5. *Let $\delta \in (0,1)$ and set $b_{N,\ell} = \sqrt{2\log(\pi^2 \ell^2 N / 6\delta)}$. Then with probability $1-\delta$,* *we have*

$$|u_N(z) - \widetilde{\mu}_{N,\ell-1}(z)| \le b_{N,\ell}\sigma_{N,\ell-1}(z) \quad \forall z \in \mathcal{M}_N, \quad \forall \ell \ge 1,$$

where

$$\widetilde{\mu}_{N,\ell}(z) = c_{N,\ell}(z)^\top (C_{N,\ell} + \sigma^2 I)^{-1} \widetilde{Y}_{N,\ell}$$

and $\widetilde{Y}_{N,\ell} \in \mathbb{R}^\ell$ *is vector with entry* $(\widetilde{Y}_{N,\ell})_i = u_N(z_i) + \eta_i$. *See* (5.8) *for the definition of* $c_{N,\ell}$ *and* $C_{N,\ell}$.

Proof. This is [Srinivas et al., 2010, Lemma 5.1] applied to the graph GP u_N, with the "surrogate" data $\widetilde{Y}_{N,\ell}$. □

Here and below, we shall use $c_N(\cdot,\cdot)$ as a placeholder for either the Matérn or SE graph-based covariance function (5.19). Notice that the "surrogate" data $\widetilde{Y}_{N,\ell}$ is introduced only for the purpose of analysis and the algorithm only has access to the real data $y_\ell = f(z_\ell) + \eta_\ell$. An important follow-up question is on the difference between the surrogate-data posterior mean $\widetilde{\mu}_{N,\ell}$ and the true posterior mean $\mu_{N,\ell} = c_{N,\ell}(x)^\top (C_{N,\ell} + \sigma^2 I)^{-1} Y_\ell$ that is actually used in the algorithm, answered by the following result.

Lemma 5.6.6. *In the event of* (5.21), *we have*

$$|\mu_{N,\ell}(z) - \widetilde{\mu}_{N,\ell}(z)| \le \frac{\epsilon_N \sqrt{\ell}}{\delta\sigma} \sigma_{N,\ell}(z), \quad \forall z \in \mathcal{M}_N \quad \forall \ell \ge 1,$$

where we recall σ *is the standard deviation of the noise* η_ℓ *and* $\sigma_{N,\ell}$ *is defined in* (5.8).

Proof. This follows by setting the misspecification error to be ϵ_N/δ in [Bogunovic and Krause, 2021, Lemma 2]. □

Now with these preparations, we are ready to start the proof of Theorem 5.2.5. In the event of (5.21) that

$$\max_{z\in\mathcal{M}_N} |u_N(z) - f(z)| \le \delta^{-1}\epsilon_N,$$

which holds with probability $1-\delta$ by Proposition 5.2.3 (with ϵ_N the corresponding error bounds (5.20)), we can shift our focus to the following cumulative regret

$$\widetilde{R}_{N,L} = \sum_{\ell=1}^{L} u_N(z^*) - u_N(z_\ell), \qquad z^* = \arg\max_{z\in\mathcal{M}_N} f(z),$$

which differs from $R_{N,L}$ (5.48) at most by $2\epsilon_N L/\delta$. Under the further event where Lemma 5.6.5 holds, we have by Lemma 5.6.6 that for all $z \in \mathcal{M}_N$,

$$\begin{aligned} \mu_{N,\ell-1}(z) - \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z) \le u_N(z) \\ u_N(z) \le \mu_{N,\ell-1}(z) + \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z). \end{aligned}$$

Therefore

$$\begin{aligned} \widetilde{R}_{N,L} &\le \sum_{\ell=1}^{L}\Bigg(\mu_{N,\ell-1}(z^*) + \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z^*) \\ &\qquad - \left[\mu_{N,\ell-1}(z_\ell) - \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z_\ell)\right]\Bigg) \\ &\le 2\sum_{\ell=1}^{L}\left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z_\ell) \\ &\le 2\left(b_{N,L} + \frac{\epsilon_N\sqrt{L-1}}{\delta\sigma}\right)\sum_{\ell=1}^{L}\sigma_{N,\ell-1}(z_\ell), \end{aligned}$$

where in the second step we have used our definition of z_ℓ in (5.23) that for all $z \in \mathcal{M}_N$

including z^*

$$\mu_{N,\ell-1}(z_\ell) + \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z_\ell) \geq \mu_{N,\ell-1}(z) + \left(b_{N,\ell} + \frac{\epsilon_N\sqrt{\ell-1}}{\delta\sigma}\right)\sigma_{N,\ell-1}(z).$$

Therefore we have arrived at the conclusion that

$$R_{N,L} \leq \frac{2\epsilon_N L}{\delta} + 2\left(b_{N,L} + \frac{\epsilon_N\sqrt{L-1}}{\delta\sigma}\right)\sum_{\ell=1}^{L}\sigma_{N,\ell-1}(z_\ell). \tag{5.50}$$

Here comes the second key ingredient in the regret analysis, which is to relate the sum of posterior standard deviations $\sum_{\ell=1}^{L}\sigma_{N,\ell-1}(z_\ell)$ to the so-called *maximum information gain.* The following result is taken from [Srinivas et al., 2010, Lemma 5.3].

Lemma 5.6.7. *Let* $I(y;v)$ *denote the mutual information between two random vectors* y *and* v *of the same size. We have*

$$I\big(\widetilde{Y}_{N,L}; \{u_N(z_\ell)\}_{\ell=1}^{L}\big) = \frac{1}{2}\sum_{\ell=1}^{L}\log\left(1+\sigma^{-2}\sigma_{N,\ell-1}^2(z_\ell)\right),$$

where $\widetilde{Y}_{N,L}$ *is the surrogate data defined in Lemma 5.6.5.*

As a corollary, we have the following result.

Lemma 5.6.8. *For* N *large, there exists a universal constant* B *such that* $c_N(x,\tilde{x}) \leq B$*. Moreover,*

$$\sum_{\ell=1}^{L}\sigma_{N,\ell-1}(z_\ell) \leq \sqrt{2(\sigma^2+B^2)L\gamma_L}\,,$$

where

$$\gamma_L = \max_{S\subset\mathcal{M}_N,|S|=L} I\big(\widetilde{Y}_{N,S}; u_N(S)\big)$$

is the maximum information gain. Here $u_N(S)$ *denotes the vector* $\{u_N(s)\}_{s\in S}$ *and* $\widetilde{Y}_{N,S}$ *is the associated vector of observations as in Lemma 5.6.5.*

Proof. The first statement can be proved in a similar fashion as Proposition 5.2.3 by bounding the difference $|c_N(z_1,z_2)-c(z_1,z_2)|$ between the graph and manifold covariance functions, and using the fact that the manifold covariance function $c(\cdot,\cdot)$ is uniformly upper bounded (which follows by the control of growth of the Laplace-Beltrami eigenfunctions in Proposition 5.6.3).

For the second statement, notice that $\sigma_{N,\ell-1}(z_\ell)\le c_N(z_\ell,z_\ell)\le B$. Using the fact that $(1+\sigma^{-2}B^2)\log(1+x)\ge x$ over $[0,\sigma^{-2}B^2]$, we have

$$\begin{aligned}\sum_{\ell=1}^{L}\sigma^2_{N,\ell-1}(z_\ell)&\le(\sigma^2+B^2)\sum_{\ell=1}^{L}\log\left(1+\sigma^{-2}\sigma^2_{N,\ell-1}(z_\ell)\right)\\&=2(\sigma^2+B^2)I(\{y_\ell\}_{\ell=1}^L;\{f_N(z_\ell)\}_{\ell=1}^L)\le 2(\sigma^2+B^2)\gamma_L,\end{aligned}$$

where the equality in the second step follows from Lemma 5.6.7. Finally, by Cauchy-Schwarz inequality we have that $\sum_{\ell=1}^L\sigma_{N,\ell-1}(z_\ell)\le\sqrt{L\sum_{\ell=1}^L\sigma_{N,\ell-1}(z_\ell)^2}$ and the result follows. ☐

Applying Lemma 5.6.8 to (5.50), we get

$$R_{N,L}\le C\left(b_{N,L}\sqrt{L}+\frac{\epsilon_N L}{\delta\sigma}\right)\sqrt{\gamma_L}, \tag{5.51}$$

where C is a universal constant. Upper bounds on γ_L have been studied extensively in the literature and by [Vakili et al., 2021, Theorem 3 or equation (7)] with $D=k_N$ and $\delta_D=0$ in our case (which holds because our graph kernel only has k_N nonzero eigenvalues), we get

$$R_{N,L}\le C\left(b_{N,L}\sqrt{L}+\frac{\epsilon_N L}{\delta\sigma}\right)\sqrt{k_N\log L}.$$

Finally, we return to bounding the simple regret using (5.49):

$$r_{N,L} \le \frac{R_{N,L}}{L} \le C\left(\frac{b_{N,L}}{\sqrt{L}} + \frac{\epsilon_N}{\delta\sigma}\right)\sqrt{k_N \log L}.$$

5.6.3 Proof of Corollary 5.2.7

Denote $\hat{z}_N^* = \arg\min_{z \in \mathcal{M}_N} d_{\mathcal{M}}(z^*, z)$, i.e., the point in $\mathcal{M}_N$ closest to z^*. Then by Proposition 5.6.1 we necessarily have

$$d_{\mathcal{M}}(z^*, \hat{z}_N^*) \le d_{\mathcal{M}}(z^*, T_N(z^*)) \le \rho_N.$$

Now notice that

$$f(z^*) - f(z_N^*) = [f(z^*) - f(\hat{z}_N^*)] + [f(\hat{z}_N^*) - f(z_N^*)] \le f(z^*) - f(\hat{z}_N^*)$$

since z_N^* being the maximizer of f over $\mathcal{M}_N$ implies $f(\hat{z}_N^*) - f(z_N^*) \le 0$. By local α-Hölder continuity of f at z^*, we conclude that

$$f(z^*) - f(\hat{z}_N^*) \le C_f d_{\mathcal{M}}(z^*, \hat{z}_N^*)^\alpha \le C\rho_N^\alpha.$$

By Lemma 5.6.4 and (5.36), we get

$$f(z^*) - f(z_N^*) = \widetilde{O}\begin{cases} N^{-\left[\frac{2s-2m+1}{m(m+3)} \wedge \frac{1}{2m}\right]} & \text{(Matérn)} \\ N^{-\frac{1}{2m}} & \text{(SE)} \end{cases}, \tag{5.52}$$

where we have dropped all dependence on logarithmic factors in the notation $\widetilde{O}$. The results follows by the identity

$$f(z^*) - f(z_L^*) = f(z^*) - f(z_N^*) + r_{N,L}$$

and the observation that the error in (5.52) would be absorbed by that of $r_{N,L}$ as shown in (5.24).

CHAPTER 6

GRAPH-BASED PRIOR AND FORWARD MODELS FOR INVERSE PROBLEMS ON MANIFOLDS WITH BOUNDARIES

6.1 Introduction

This chapter develops manifold learning techniques to address two tasks in the numerical solution of PDE-constrained Bayesian inverse problems on manifolds with boundaries: (1) the design and approximation of Gaussian field priors for spatially-distributed PDE parameters; and (2) the approximation of forward maps from PDE parameters to PDE solutions. We introduce graph-based approximations of prior and forward models and numerically show their effectiveness in two test problems. The first one concerns the recovery of the diffusion coefficient of an elliptic PDE from pointwise noisy measurements of the PDE solution; the second one concerns the recovery of the initial condition of a heat equation from noisy measurements of the PDE solution at some positive time. Both of these problems have been widely used as test cases for Bayesian inversion on manifolds and Euclidean domains [Stuart, 2010, Franklin, 1970, Garcia Trillos and Sanz-Alonso, 2017, Harlim et al., 2020, Garcia Trillos et al., 2020b, Chada et al., 2021, Bigoni et al., 2020], but previous work has largely ignored the boundary effects that are the focus of this chapter. The applied significance of elliptic and heat inverse problems is exemplified by [Zimmerman et al., 1998, Beck et al., 1985] and references therein. PDEs on manifolds arise in many applications, including granular flow [Rauter and Tuković, 2018], liquid crystals [Virga, 2018], biomembranes [Elliott and Stinner, 2010], computer graphics [Bertalmıo et al., 2001, Macdonald and Ruuth, 2010], and brain imaging [Mémoli et al., 2004].

In the Bayesian approach to inverse problems [Tarantola, 2015, Kaipo and Somersalo, 2006, Stuart, 2010], overviewed in Section 6.2 below, inference on the PDE parameters is performed using a posterior distribution obtained by conditioning a user-chosen prior distribution to observed data. When the parameter of interest is a function, employing an adequate choice of prior is crucial: the prior determines the support of the posterior, and hence the space of parameters that can be recovered given sufficient data. In this work, we will focus on Matérn-type Gaussian field priors [Matérn, 2013, Stein, 2012] that have been widely adopted in inverse problems, statistics and machine learning due to their flexibility and computational efficiency, e.g. [Lindgren et al., 2011, Bolin and Kirchner, 2019, Bolin et al., 2020, 2018, Roininen et al., 2019, Wiens et al., 2020, Bolin, 2014]. We consider Matérn-type models on manifolds [Lindgren et al., 2011] and investigate their discretization using graphs, as well ass their generalization to enhance flexibility near the boundaries. To our knowledge, all previous work on graph-based Matérn models, e.g. [Bertozzi et al., 2018, Sanz-Alonso and Yang, 2022a, Garcia Trillos et al., 2020b, Sanz-Alonso and Yang, 2022b], disregarded boundary effects, despite their importance in the discretization of Gaussian field priors in Euclidean settings [Khristenko et al., 2019, Daon and Stadler, 2018, Calvetti et al., 2006]. Our numerical results will confirm that accounting for boundary conditions is important in the design of priors and in their graph-based approximation. While our emphasis is on PDE-constrained inverse problems, we expect that the boundary-aware graph-based Matérn priors that we introduce will also find applications in graph-based machine learning [Sanz-Alonso and Yang, 2022b].

Approximation of the forward map requires solving PDEs on manifolds with a range of PDE parameters. There are numerous techniques to solve PDEs on manifolds, including the finite element method [Dziuk and Elliott, 2013], the level set method [Bertalmıo et al., 2001], closest point representation [Ruuth and Merriman, 2008], and the mesh-free radial basis function method [Fuselier and Wright, 2013]. In contrast to all of these methods, the kernel approach we will consider avoids the need to have some parameterization on the manifolds; we refer to [Jiang and Harlim, 2023] for a detailed discussion of the advantages and disadvantages of these PDE solvers. In this work, we introduce graph-based forward map approximations for elliptic and heat inversion on manifolds with boundaries. Our graph-based approach leverages the ghost point diffusion map (GPDM) algorithm [Jiang and Harlim, 2023] to approximate second-order elliptic operators with classical boundary conditions, which are used to define elliptic and heat forward models, and also the Matérn prior covariance. The main idea behind the GPDM algorithm is to extend the domain of the PDE through a set of artificially constructed *ghost points*. By extending the underlying manifold where the PDE was defined, one can treat the boundary of the original manifold as the interior of the extended manifold. The implication is that the kernel-based approximations that underpin the design of the GPDM method remain valid close to the boundary. The GPDM algorithm has been generalized to solve time-dependent advection-diffusion equations on manifolds [Yan et al., 2023].

We close this introduction with an outline of the chapter and a summary of our contributions.

- In Section 6.2, we overview the Bayesian approach to inverse problems and provide a high-level summary of the proposed procedure. We also review the Bayesian approach in [Harlim et al., 2020] for elliptic inversion on a closed (i.e. boundary-free and compact) manifold, and we lay out the forward map discretization for the heat inverse problem. We present elliptic and heat inverse problems in a parallel way, emphasizing that both can be treated in the same way under the proposed approach except for the necessary distinction in the discretization of the forward map.

- In Section 6.3, we review the GPDM algorithm and introduce our boundary-aware Matérn-type priors and forward map approximations for elliptic and heat inversion.

- In Section 6.4, we showcase implementations of the proposed methodology for elliptic and heat inversion on one and two-dimensional manifolds with boundaries. In addition, we demonstrate the enhanced flexibility of our proposed prior relative to existing graph-based approaches that ignore boundary effects.

- In Section 6.5, we provide conclusions and some open directions for future work.

6.2 Background

In this section, we first give the necessary background on the Bayesian formulation of inverse problems [Tarantola, 2015, Kaipo and Somersalo, 2006, Sanz-Alonso et al., 2023] and the function space perspective [Stuart, 2010, Dashti and Stuart, 2017]. Next, we formulate the problem of interest, namely, PDE-constrained Bayesian inversion on manifolds. Lastly, we review the graph-based approach in [Harlim et al., 2020] which is only applicable for closed manifolds.

6.2.1 Bayesian Formulation of Inverse Problems

Suppose we have a parameter of interest $\theta \in \Theta$, observed data $y \in \mathbb{R}^M$ and a forward model $\mathcal{G}$, satisfying

$$y = \mathcal{G}(\theta) + \eta, \tag{6.1}$$

where $\eta \in \mathbb{R}^M$ is an observation noise. Our goal is to study the inverse problem of recovering θ from the observed data y. In the Bayesian framework, one endows probabilistic structure to any unknown quantity. Here this involves specifying a *prior* distribution μ for the parameter θ and a distribution ρ for the noise η. To facilitate our presentation, we assume throughout that $\eta \sim \rho = \mathcal{N}(0, \Gamma)$, where $\Gamma \in \mathbb{R}^{M \times M}$ is a given positive-definite covariance matrix. We also assume that θ and η are independent random variables. The solution of the inverse problem under the Bayesian framework is then the *posterior* distribution of θ given y, denoted μ^y. The posterior distribution allows uncertainty quantification in the parameters in addition to point estimates of the parameters. In practice, this often requires to resort to sampling techniques such as Markov chain Monte Carlo (MCMC).

If the parameter space is finite-dimensional, i.e. $\Theta \subseteq \mathbb{R}^N$, and the prior μ has Lebesgue density π, then the posterior μ^y has Lebesgue-density π^y given by

$$\pi^y(\theta) = \frac{1}{Z}\rho\big(y - \mathcal{G}(\theta)\big)\pi(\theta), \qquad Z := \int \rho\big(y - \mathcal{G}(\theta)\big)\pi(\theta)\, d\theta, \tag{6.2}$$

where $\rho\big(y - \mathcal{G}(u)\big)$ is the data *likelihood*. Under suitable assumptions [Stuart, 2010], the characterization (6.2) of the posterior can be generalized to infinite dimensional parameter space Θ by writing the posterior as a change of measure with respect to the prior:

$$\frac{d\mu^y}{d\mu}(\theta) = \frac{1}{Z}\,\rho\big(y - \mathcal{G}(\theta)\big) \propto \exp\big(-\Phi(\theta; y)\big), \tag{6.3}$$

where we have defined $\Phi(\theta; y) := \frac{1}{2}\|y - \mathcal{G}(\theta)\|_\Gamma^2$ and we set $\|\cdot\|_\Gamma := \|\Gamma^{-\frac{1}{2}} \cdot \|$.

6.2.2 General Setting and Approach

We are interested in the inverse problem of recovering a parameter function $\theta \in \Theta$ of a PDE defined on a manifold $\mathcal{M} \subset \mathbb{R}^D$ from noisy measurements of the PDE solution $u \in \mathcal{U}$ at given locations $\{\tilde{x}_m\}_{m=1}^M \subset \mathcal{M}$. Here both Θ and $\mathcal{U}$ are suitable function spaces. The data model (6.1) is therefore given by

$$y_m = u(\tilde{x}_m) + \eta_m, \quad m = 1, \ldots, M, \qquad \eta = \{\eta_m\}_{m=1}^M \sim \mathcal{N}(0, \Gamma), \tag{6.4}$$

with the *forward model* $\mathcal{G} : \theta \mapsto \big(u(\tilde{x}_1), \ldots, u(\tilde{x}_M)\big)$ defined as the composition of a *forward map* $\mathcal{F} : \theta \mapsto u$ from PDE input to PDE solution with an *observation map* $\mathcal{O} : u \mapsto \big(u(\tilde{x}_1), \ldots, u(\tilde{x}_M)\big)$ from PDE solution to observed quantities. In this chapter, we solely focus on pointwise observations and tacitly assume throughout that the solutions to the PDEs we consider can be evaluated pointwise. However, our methodology can deal with other types of observation map, e.g. defined by bounded linear functionals in L^2 [Harlim et al., 2020]. We will discretize the forward map using a point cloud in $\mathcal{M}$, which is denoted by

$\{x_n\}_{n=1}^N \supseteq \{\tilde{x}_m\}_{m=1}^M$, where $M \le N$. Our approach for the computational implementation of the inverse problem is then summarized in the following four steps:

(i) **Prior Specification**: Specify a prior distribution for the infinite dimensional parameter $\theta \in \Theta$.

(ii) **Prior and Forward Model Discretization**:

  - Discretize using graph-based techniques the prior distribution μ to obtain μ_N, a prior distribution over $\theta_N := \big(\theta(x_1), \cdots, \theta(x_N)\big) \in \mathbb{R}^N$. Note that μ_N is a prior distribution over discrete functions θ_N defined on the point cloud $\{x_n\}_{n=1}^N$.
  - Discretize using graph-based techniques the given forward map $\mathcal{F} : \theta \to u$ to obtain $\mathcal{F}_N : \theta_N \mapsto u_N := \mathcal{F}_N(\theta_N) \in \mathbb{R}^N$, where u_N is an approximation to the solution of the PDE evaluated along the point cloud, i.e., $u_N \approx \big(u(x_1), \ldots, u(x_N)\big)$. Furthermore, we define $\mathcal{G}_N(\theta_N) := \big(u_N(\tilde{x}_1), \ldots, u_N(\tilde{x}_M)\big) \in \mathbb{R}^M$ where $u_N(\tilde{x}_i)$ denotes the component of the vector $u_N \in \mathbb{R}^N$ that corresponds to $\tilde{x}_i \in \{x_n\}_{n=1}^N$.

(iii) **Sampling**: Use an MCMC algorithm to obtain samples from the posterior distribution over θ_N, given by

$$\frac{d\mu_N^y}{d\mu_N}(\theta_N) \propto \exp(-\Phi_N(\theta_N; y)), \text{ where } \Phi_N(\theta_N; y) := \frac{1}{2}\|y - \mathcal{G}_N(\theta_N)\|_\Gamma^2. \tag{6.5}$$

(iv) **Interpolation**: If desired, extend the samples to functions on $\mathcal{M}$ with an interpolation map.

This four-step approach was introduced in [Garcia Trillos et al., 2020b, Garcia Trillos and Sanz-Alonso, 2018] and previous works on the discretization process include [Harlim et al., 2020]. The primary focus of this chapter is to contribute to the specification and discretization steps for PDE-constrained inverse problems on manifolds with boundaries.

We propose (1) priors that accommodate for prescribed boundary conditions, (2) graph-based discretizations of such priors; and (3) graph-based discretizations of the forward PDEs supplemented with boundary conditions. These procedures will be introduced in Section 6.3. In the remainder of this section, we address the sampling and interpolation steps, which are based on existing methodologies that are applicable in wide generality.

For the sampling step, we will use the graph pCN method [Bertozzi et al., 2018, Garcia Trillos et al., 2020b], summarized in Algorithm 6.2.1.

Algorithm 6.2.1 Graph pCN

```
1: Input: initial value $\theta_N^{(0)}$, number of samples $J$, parameter value $\zeta \in (0,1)$.
2: For $j = 1, \dots, J$ do:
    1. Propose $\tilde{\theta}_N^{(j)} = (1-\zeta^2)^{\frac{1}{2}}\theta_N^{(j)} + \zeta\,\xi_N^{(j)}$, where $\xi_N^{(j)} \sim \mathcal{N}(0, V_N)$.
    2. Set
       $$a(\theta_N^{(j)}, \tilde{\theta}_N^{(j)}) := \min\Big\{1, \exp\Big(\Phi_N(\theta_N^{(j)}; y) - \Phi_N(\tilde{\theta}_N^{(j)}; y)\Big)\Big\},$$
       and let
       $$\theta_N^{(j+1)} = \begin{cases} \tilde{\theta}_N^{(j)} & \text{with probability } a(\theta_N^{(j)}, \tilde{\theta}_N^{(j)}), \\ \theta_N^{(j)} & \text{with probability } 1 - a(\theta_N^{(j)}, \tilde{\theta}_N^{(j)}). \end{cases}$$
3: Output: Samples of discrete functions $\{\theta_N^{(j)}\}_{j=1}^J$.
```

The covariance matrix V_N in the pCN proposal is tightly connected with the prior construction. In fact, this matrix will be precisely the covariance matrix of the discretized prior distribution. The parameter ζ controls the size of the proposed moves of the chain. For a large ζ value, we explore a wider region of the state space with higher number of rejections. The motivation for using the graph pCN method is that it shows robustness with respect to the level of discretization refinement determined by the value of N. We refer to [Beskos et al., 2008, Cotter et al., 2010, Bertozzi et al., 2018, Garcia Trillos et al., 2020b] for the theoretical and empirical justification of this robustness.

For the interpolation step, once we obtain finite-dimensional samples $\theta_N \in \mathbb{R}^N$ representing a parameter function evaluated on the point cloud $\{x_n\}_{n=1}^N$, we can extend the samples into functions on $\mathcal{M}$ using the K-NN interpolation map defined by

$$\theta(x) = \frac{1}{\mathsf{K}} \sum_{x_i \in N_{\mathsf{K}}(x)} \theta_N(x_i), \; x \in \mathcal{M}, \; x \notin \{x_n\}_{n=1}^N,$$

where $N_{\mathsf{K}}(x)$ is the set of K-nearest points in the point cloud $\{x_n\}_{n=1}^N$ to the point x. To find the K-nearest points, one can use the Euclidean distance in $\mathbb{R}^D$ or the geodesic distance on $\mathcal{M} \subset \mathbb{R}^D$, if available. We refer to [Garcia Trillos et al., 2020b] for more details and we note that other interpolation methods are possible.

6.2.3 Elliptic Inverse Problems and Heat Inversion on Closed Manifolds

In this section, we overview existing procedures for the prior specification and the discretization of prior and forward models on *closed* (i.e. compact and boundary-free) manifolds. We start by describing the two inverse problems used as running examples. Throughout this section, $\mathcal{M}$ will denote a d-dimensional smooth closed manifold isometrically embedded in $\mathbb{R}^D$.

Elliptic Inverse Problem Consider the elliptic PDE

$$\mathcal{L}u := -\mathrm{div}(\kappa \nabla u) = f, \; x \in \mathcal{M}, \tag{6.6}$$

where the divergence and gradient operators are defined with respect to the Riemannian metric inherited by $\mathcal{M}$ from $\mathbb{R}^D$. The goal of the elliptic inverse problem is to recover the diffusion coefficient κ given the right-hand side f (assumed to be smooth) and noisy pointwise observation of the solution u at M spatial locations $\{\tilde{x}_m\}_{m=1}^M \subset \mathcal{M}$. The data are therefore given by $\{y_m\}_{m=1}^M = \{u(\tilde{x}_m) + \eta_m\}_{m=1}^M$. We cast this problem into the general framework (6.1) defining the forward map $\mathcal{F} : \theta \mapsto u$, where $\theta := \log \kappa \in (-\infty, \infty)$. Previous kernel-based methods for elliptic inverse problems on manifolds [Harlim et al., 2020] assumed $\mathcal{M}$ to be closed so that the elliptic PDE given in (6.6) is not supplemented with a boundary condition. Bayesian elliptic inverse problems on Euclidean domains are one of the standard model problems in uncertainty quantification [Stuart, 2010, Dashti and Stuart, 2017, Garcia Trillos and Sanz-Alonso, 2017]. We will refer to the operator $\mathcal{L}$ as the weighted Laplacian operator.

Heat Inversion Consider the heat equation

$$\begin{cases} u_t = -\Delta_{\mathcal{M}} u, & (x,t) \in \mathcal{M} \times [0,\infty), \\ u(x,0) = \theta(x), & x \in \mathcal{M}, \end{cases} \tag{6.7}$$

where $\Delta_{\mathcal{M}} := -\mathrm{div}(\nabla \cdot)$ is the Laplace-Beltrami operator on $\mathcal{M}$, which reduces to the negative of the conventional Laplace operator in Euclidean space. The goal of the inverse heat problem is to recover the initial heat θ defined on $\mathcal{M}$ from noisy pointwise observation of the heat at time $t^* > 0$ along M spatial locations $\{\tilde{x}_m\}_{m=1}^M \subset \mathcal{M}$. The data are therefore given by $\{y_m\}_{m=1}^M = \{u(\tilde{x}_m, t^*) + \eta_m\}_{m=1}^M$ where $u(\,\cdot\,, t^*)$ is the solution of the heat equation (6.7) at time t^*. The corresponding forward map is a heat equation solver, namely $\mathcal{F} : \theta \mapsto u$, where $u(x) := u(x, t^*)$ and θ is the initial condition. The function space formulation of Bayesian heat inversion was introduced in [Franklin, 1970] on Euclidean domains, and graph-based formulations on closed manifolds were studied in [Garcia Trillos et al., 2020b]. Other than its natural physical interpretation, this inverse problem can also be viewed as a standard deconvolution problem arising in imaging applications.

Prior Specification and Discretization

Here we describe the specification of Matérn Gaussian field priors for the log-diffusion coefficient and the initial heat on closed manifolds. We will also overview their graph-based discretization. Recall that the Matérn model on a closed manifold $\mathcal{M}$ is defined [Lindgren et al., 2011] as the Gaussian measure

$$\mu = \mathcal{N}(0, V), \quad V = c(\tau I + \Delta_{\mathcal{M}})^{-s}, \tag{6.8}$$

where $\Delta_{\mathcal{M}} := -\mathrm{div}(\nabla \cdot)$ is the Laplace-Beltrami operator on $\mathcal{M}$ and $\tau > 0, s > \frac{d}{2}$ are two free parameters. The choice of normalizing constant

$$c = \frac{1}{\sum_{i=1}^{\infty}(\tau + \lambda_i)^{-s}},$$

where $(\lambda_i)_{i=1}^{\infty}$ are the increasingly ordered eigenvalues of $\Delta_{\mathcal{M}}$, ensures that $\theta \sim \mu$ has unit marginal variance. Samples can be represented using the Karhunen-Loève (KL) expansion

$$\theta(x) = \sqrt{c}\sum_{i=1}^{\infty}(\tau + \lambda_i)^{-\frac{s}{2}}\zeta_i\varphi_i(x), \quad x \in \mathcal{M}, \tag{6.9}$$

where $(\zeta_i)_{i=1}^{\infty}$ are i.i.d. standard normal random variables and $(\varphi_i)_{i=1}^{\infty}$ are eigenfunctions of $\Delta_{\mathcal{M}}$ with corresponding eigenvalues $(\lambda_i)_{i=1}^{\infty}$. The parameter $\tau^{\frac{1}{2}}$ represents the inverse length-scale and allows to discern the significant terms in the KL expansion (6.9). The parameter s characterizes the almost-sure regularity of the samples. The requirement $s > \frac{d}{2}$ is motivated by Weyl's law, see [Canzani, 2013, Theorem 72] and [Colbois et al., 2015], which asserts that $\lambda_i \asymp i^{\frac{2}{d}}$, i.e., the asymptotic behavior of λ_i is equivalent to that of $i^{\frac{2}{d}}$. Thus, $s > \frac{d}{2}$ ensures that μ is a well-defined Gaussian measure in $L^2(\mathcal{M})$. Further increasing s allows to ensure higher-order sample path differentiability and Sobolev regularity [Stuart, 2010, Sanz-Alonso and Yang, 2022c], and thereby the well-posedness of elliptic and heat inverse problems [Stuart, 2010, Harlim et al., 2020]. The flexibility of the Matérn model can be enhanced by letting the inverse length-scale be a spatially varying function rather than a scalar parameter [Lindgren et al., 2011, Sanz-Alonso and Yang, 2022a].

To define a prior distribution over discrete functions defined along the point cloud $\{x_n\}_{n=1}^N \subset \mathcal{M}$, the paper [Harlim et al., 2020] proposed to replace the Laplace-Betrami operator $\Delta_{\mathcal{M}}$ in (6.8) by a graph Laplacian Δ_N constructed using the point cloud $\{x_n\}_{n=1}^N$. To be more specific, the discretized prior distribution is given by

$$\mu_N = \mathcal{N}(0, V_N), \quad V_N = c_N(\tau I + \Delta_N)^{-s},$$

where $\Delta_N \in \mathbb{R}^{N\times N}$ is a graph Laplacian constructed using $\{x_n\}_{n=1}^N$ and $\tau > 0$, $s > \frac{d}{2}$ are two free parameters. The role of the parameters τ and s is analogous to the infinite-dimensional case and these can be manually tuned, or learned from data using a hierarchical Bayesian approach [Harlim et al., 2020, Sanz-Alonso and Yang, 2022a]. In practice, among different choices of graph Laplacian [Von Luxburg, 2007], the use of self-tuning graph Laplacian [Zelnik-Manor and Perona, 2005] was recommended in [Harlim et al., 2020]. More specifically, in our numerical experiments we use a symmetric graph Laplacian given by

$$\Delta_N = I_N - A^{-\frac{1}{2}} S A^{-\frac{1}{2}}. \tag{6.10}$$

Here S is a similarity matrix and A is a diagonal matrix whose entries are respectively given by

$$S_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{2d_i d_j}\right), \ A_{ii} = \sum_{j=1}^N S_{ij},$$

where d_i is the distance between x_i and its k-th closest neighbor. Same definition applies for d_j. Typically, the parameter k is tuned empirically. We refer to [Von Luxburg, 2007] for a review of graph Laplacians and to [Sanz-Alonso and Yang, 2022a] for generalizations of this graph-based prior model and its connection to the Matérn family [Stein, 2012]. To obtain samples whose variance per node is one, the normalizing constant c_N can be set to be

$$c_N = \frac{N}{\sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-s}},$$

where $(\lambda_n^{(N)})_{n=1}^{N}$ are the increasingly ordered eigenvalues of Δ_N. Samples θ_N from this prior can be represented via the KL expansion

$$\theta_N(x_i) = \sqrt{c_N} \sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-\frac{s}{2}} \zeta_n \varphi_n^{(N)}(x_i), \quad 1 \le i \le N, \tag{6.11}$$

where $(\zeta_n)_{n=1}^{N}$ are i.i.d. standard normal random variables and $(\varphi_n^{(N)})_{n=1}^{N}$ are eigenvectors of Δ_N with corresponding eigenvalues $(\lambda_n^{(N)})_{n=1}^{N}$.

Forward Map Discretization

Elliptic Inverse Problem For the discretization of the elliptic forward map, one can approximate the operator $\mathcal{L}$ in equation (6.6) by an integral operator [Harlim et al., 2020], which can be subsequently approximated using a Monte Carlo sum. To be more specific, let

$$G_\epsilon\big(u(x)\big) := \epsilon^{-\frac{d}{2}} \int_{\mathcal{M}} h\left(\frac{|x-y|^2}{4\epsilon}\right) u(y)\, d\mathcal{V}(y),$$

where $h(s) = 2^{-d}\pi^{-\frac{d}{2}} \exp(-s)$ and $\mathcal{V}$ denotes the volume form inherited by $\mathcal{M}$ from the ambient space $\mathbb{R}^D$. For a smooth u, it was shown in [Coifman and Lafon, 2006] that

$$G_\epsilon\big(u(x)\big) = u(x) + \epsilon\big(\omega u(x) - \Delta_{\mathcal{M}} u(x)\big) + \mathcal{O}(\epsilon^2),\ x \in \mathcal{M}, \tag{6.12}$$

where ω is a function that depends only on the parametrization of the manifold $\mathcal{M}$.

Recall that $\Delta_{\mathcal{M}} \coloneqq -\mathrm{div}(\nabla \cdot)$ and from (6.6),

$$\begin{aligned} \mathcal{L}u &:= -\mathrm{div}(\kappa \nabla u) = \kappa \Delta_{\mathcal{M}} u - \nabla u \cdot \nabla \kappa && (6.13) \\ &= \sqrt{\kappa}\Big(\Delta_{\mathcal{M}}(u\sqrt{\kappa}) - u\Delta_{\mathcal{M}}(\sqrt{\kappa})\Big). && (6.14) \end{aligned}$$

Use (6.12) on $\sqrt{\kappa}$ and $u\sqrt{\kappa}$ to obtain,

$$\begin{aligned} uG_\epsilon \sqrt{\kappa} &= u\sqrt{\kappa} + \epsilon\big(\omega u \sqrt{\kappa} - u\Delta_{\mathcal{M}}\sqrt{\kappa}\big) + \mathcal{O}(\epsilon^2), \\ G_\epsilon(u\sqrt{\kappa}) &= u\sqrt{\kappa} + \epsilon\big(\omega u\sqrt{\kappa} - \Delta_{\mathcal{M}}\big(u\sqrt{\kappa}\big)\big) + \mathcal{O}(\epsilon^2). \end{aligned}$$

This yields

$$uG_\epsilon\sqrt{\kappa} - G_\epsilon(u\sqrt{\kappa}) = \epsilon\Big(\Delta_{\mathcal{M}}(u\sqrt{\kappa}) - u\Delta_{\mathcal{M}}(\sqrt{\kappa})\Big) + \mathcal{O}(\epsilon^2) = \frac{\epsilon}{\sqrt{\kappa}}\mathcal{L}u + \mathcal{O}(\epsilon^2).$$

Motivated from this equation, we define

$$\mathcal{L}_\epsilon u(x) \coloneqq \frac{\sqrt{\kappa(x)}}{\epsilon}\big(u(x)G_\epsilon\big(\sqrt{\kappa(x)}\big) - G_\epsilon(u(x)\sqrt{\kappa(x)})\big),$$

which can be rewritten as the following integral operator

$$\mathcal{L}_\epsilon u(x) \coloneqq \frac{1}{\epsilon^{\frac{d}{2}+1}} \int_{\mathcal{M}} h\left(\frac{|x-y|^2}{4\epsilon}\right) \sqrt{\kappa(x)\kappa(y)}\big(u(x) - u(y)\big)\, d\mathcal{V}(y) \tag{6.15}$$

satisfying

$$\mathcal{L}_\epsilon u(x) = \mathcal{L}u(x) + \mathcal{O}(\epsilon),\ x \in \mathcal{M}.$$

The kernel operator $\mathcal{L}_\epsilon$ can be approximated by Monte Carlo viewing the point cloud $\{x_n\}_{n=1}^N$ as manifold samples. Using importance sampling with approximate density q_ϵ, we have an approximation of $\mathcal{L}_\epsilon$ given by, for $i = 1, \ldots, N$,

$$\mathcal{L}_{\epsilon,N} u(x_i) := \frac{1}{\epsilon^{\frac{d}{2}+1}} \left(\frac{1}{N} \sum_{j=1}^{N} h\left(\frac{|x_i - x_j|^2}{4\epsilon} \right) \sqrt{\kappa(x_i)\kappa(x_j)}\, q_\epsilon(x_j)^{-1} \big(u(x_i) - u(x_j) \big) \right), \tag{6.16}$$

$$q_\epsilon(x_j) := \frac{1}{2^d \pi^{\frac{d}{2}} N \epsilon^{\frac{d}{2}}} \sum_{k=1}^{N} \exp\left(-\frac{|x_j - x_k|^2}{4\epsilon} \right)$$

is a kernel estimator of the density of the point cloud. One can write the discretized weighted Laplacian operator in (6.16) in a matrix form. Specifically, define a kernel matrix H with entries $H_{ij} := \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon} \right)$ and define a vector Q with entries $Q_i = \sum_{j=1}^{N} H_{ij}$. Set the matrix W with entries $W_{ij} = \sqrt{\kappa(x_i)\kappa(x_j)}\, H_{ij} Q_j^{-1}$ and the diagonal matrix D with diagonal entries $D_{ii} = \sum_{j=1}^{N} W_{ij}$. Then the discretized weighted Laplacian $\mathcal{L}_{\epsilon,N}$ can be written as

$$\mathcal{L}_{\epsilon,N} = \frac{1}{\epsilon}(D - W). \tag{6.17}$$

For practical implementation, the bandwith parameter ϵ can be empirically chosen so that it lies in the region where

$$\log\big(T(\epsilon)\big) = \log \left(\sum_{i,j=1}^{N,K} \exp\left(-\frac{|x_i - x_j(i)|^2}{4\epsilon} \right) \right) \tag{6.18}$$

is approximately linear [Harlim et al., 2020]. Here $(x_j(i))_{j=1}^{K}$ are the K closest points to the point x_i. In practice, it was further observed that the maximum slope of $\log\left(T(\epsilon)\right)$ often coincided with $\frac{d}{2}$, where d is the dimension of the underlying manifold $\mathcal{M}$ (see [Berry and Harlim, 2016]).

Using the above discretization, one can obtain a discretized forward map, $\mathcal{F}_{\epsilon,N}$, which maps $\theta_N = \big(\log(\kappa(x_1)), \ldots, \log(\kappa(x_N))\big)$ to an N-dimensional vector u_N, which represents an approximate solution to (6.6) restricted to the point cloud. In other words, u_N is the minimal norm least-squares solution of

$$\mathcal{L}_{\epsilon,N} u_N = f_N,$$

where $f_N = \big(f(x_1), \cdots, f(x_N)\big)$. Therefore, the discretized forward map is given by

$$\mathcal{F}_{\epsilon,N} : \theta_N \mapsto u_N = \mathcal{L}_{\epsilon,N}^{-1} f_N,$$

where $\mathcal{L}_{\epsilon,N}^{-1}$ denotes the pseudo-inverse. This allows us to write the discretized posterior distribution μ_N^y as a change of measure with respect to the discretized prior distribution μ_N in the following way:

$$\frac{d\mu_N^y}{d\mu_N}(\theta_N) \propto \exp\left(-\frac{1}{2}|y - \mathcal{G}_{\epsilon,N}(\theta_N)|_\Gamma^2\right),$$

where $\mathcal{G}_{\epsilon,N}(\theta_N) = \big(u_N(\tilde{x}_1), \ldots, u_N(\tilde{x}_M)\big)$. The Lebesgue density of the posterior is given by $\pi_N^y(\theta_N) \propto \rho\big(y - \mathcal{G}_{\epsilon,N}(\theta_N)\big)\pi_N(\theta_N)$, where ρ and π_N are Gaussian densities $\mathcal{N}(0,\Gamma)$ and $\mathcal{N}(0, V_N)$ in $\mathbb{R}^M$ and $\mathbb{R}^N$, respectively. The graph pCN algorithm can then be used to obtain samples $\theta_N \sim \pi_N^y$ that can be extended into the underlying manifold using a K-NN map.

Heat Inversion To discretize the heat forward map, we replace the Laplace-Beltrami operator in (6.7) with the graph Laplacian Δ_N in (6.10) and solve

$$\begin{cases} \frac{\partial}{\partial t} u_N = -\Delta_N u_N, & t \in [0,\infty), \\ u_N(0) = \theta_N, \end{cases}$$

where $\theta_N \in \mathbb{R}^N$ represents the initial heat function restricted to the point cloud. Then the solution $u_N := u_N(t^*)$ at time t^* can be expressed as

$$u_N = \sum_{n=1}^{N} \langle \theta_N, \varphi_n^{(N)} \rangle e^{-\lambda_n^{(N)} t^*} \varphi_n^{(N)},$$

where $(\lambda_n^{(N)}, \varphi_n^{(N)})_{n=1}^N$ are the ordered eigenpairs of the graph Laplacian Δ_N. From this, we naturally obtain a discretization of the forward map given by

$$\mathcal{F}_N : \theta_N \mapsto u_N = \sum_{n=1}^{N} \langle \theta_N, \varphi_n^{(N)} \rangle e^{-\lambda_n^{(N)} t^*} \varphi_n^{(N)}.$$

We can then write the discretized posterior distribution μ_N^y as a change of measure with respect to the prior distribution μ_N

$$\frac{d\mu_N^y}{d\mu_N}(\theta_N) \propto \exp\left(-\frac{1}{2} |y - \mathcal{G}_N(\theta_N)|_\Gamma^2 \right),$$

where $\mathcal{G}_N(\theta_N) = \bigl(u_N(\tilde{x}_1), \dots, u_N(\tilde{x}_M)\bigr)$. The Lebesgue density of the posterior is given by $\pi_N^y(\theta_N) \propto \rho\bigl(y - \mathcal{G}_N(\theta_N)\bigr)\pi_N(\theta_N)$, and sampling and interpolation can be performed with the general methodology described previously.

6.3 Bayesian Inverse Problems on Manifolds with Boundaries

In this section, we propose a novel methodology for Bayesian inversion on manifolds with boundaries, addressing the design of priors that can reflect the given boundary conditions, the graph-based discretization of these priors, and the graph-based approximation of PDE-constrained forward maps supplemented with boundary conditions. To be concrete, we will focus on the following elliptic and heat inverse problems on one and two-dimensional manifolds with Dirichlet boundary conditions. Throughout this section and the rest of this chapter, $\mathcal{M}$ will denote a smooth compact d-dimensional manifold isometrically embedded in $\mathbb{R}^D$ with boundary $\partial\mathcal{M}$. We denote $\mathcal{M}^o = \mathcal{M} \setminus \partial\mathcal{M}$.

Elliptic Inverse Problem Consider the following elliptic PDE with Dirichlet boundary conditions,

$$\begin{cases} \mathcal{L}u := -\mathrm{div}(\kappa \nabla u) = f, \quad x \in \mathcal{M}^o, \\ u(x) = h(x), \quad x \in \partial \mathcal{M}, \end{cases} \tag{6.19}$$

where the divergence and gradient operators are defined with respect to the Riemannian metric inherited by the manifold $\mathcal{M}$ from $\mathbb{R}^D$. In our numerical examples, we will consider a one-dimensional semi-ellipse where $\partial\mathcal{M} = \{x_0, x_N\}$ and a two-dimensional semi-torus where $\partial\mathcal{M} = \mathcal{B}_1 \cup \mathcal{B}_2$ and $\mathcal{B}_i$, $i = 1, 2$ are circles. We will then denote by h_1 and h_2 the Dirichlet boundary conditions at $\mathcal{B}_1$ and $\mathcal{B}_2$.

The goal of the elliptic inverse problem is to recover the diffusion coefficient κ given the right-hand side f and noisy pointwise observation of the solution u at M spatial locations $\{\tilde{x}_m\}_{m=1}^M \subset \mathcal{M}$. The data are given by $\{y_m\}_{m=1}^M = \{u(\tilde{x}_m) + \eta_m\}_{m=1}^M$ with the forward map $\mathcal{F} : \theta \mapsto u$, where $\theta := \log \kappa \in (-\infty, \infty)$. We refer to [Jiang and Harlim, 2023] for sufficient conditions on $\mathcal{M}$, κ, f and h to guarantee that the solution to (6.19) can be evaluated pointwise.

Heat Inversion Consider the following heat equation with Dirichlet boundary conditions,

$$\begin{cases} u_t(x,t) = -\Delta_{\mathcal{M}} u(x,t), \quad x \in \mathcal{M}^o, t > 0, \\ u(x,t) = h(x), \quad x \in \partial\mathcal{M}, \, t \geq 0, \\ u(x,0) = \theta(x), \quad x \in \mathcal{M}, \end{cases} \tag{6.20}$$

Again the goal of the inverse heat problem is to recover the initial heat θ defined on $\mathcal{M}$ from noisy pointwise observation of the heat at time $t^* > 0$ along M spatial locations $\{\tilde{x}_m\}_{m=1}^M \subset \mathcal{M}$. The data are therefore given by $\{y_m\}_{m=1}^M = \{u(\tilde{x}_m, t^*) + \eta_m\}_{m=1}^M$. The corresponding forward map is a heat equation solver, namely $\mathcal{F} : \theta \mapsto u$, where $u(x) \equiv u(x, t^*)$ denotes the solution of (6.20) at time t^* and θ is the initial condition. For our numerical examples we will use a one-dimensional semi-ellipse and a two-dimensional semi-torus, and we will adopt the same notations as in the elliptic inverse problem.

6.3.1 Ghost Point Diffusion Maps for Dirichlet Boundary Conditions

In this section, we give a short discussion on the construction of ghost points and the ghost point diffusion map (GPDM) algorithm, which will be used in the discretizations of the prior and forward models. In this chapter, we focus on a specific GPDM algorithm to approximate the weighted Laplacian operator, $\mathcal{L} = -\mathrm{div}(\kappa\nabla\cdot)$ whose inputs are functions $u : \mathcal{M} \to \mathbb{R}$ that satisfy the Dirichlet boundary condition, $u(x) = h(x)$, for all $x \in \partial\mathcal{M}$. The discussion will focus on the algorithmic aspect that will be used in the forward map discretization. For other types of (possibly non-symmetric) second-order diffusion operators and boundary conditions, and the convergence analysis, we refer to [Jiang and Harlim, 2023, Yan et al., 2023].

The key idea of the GPDM algorithm comes from the classical ghost point method [LeVeque, 2007] for solving PDEs with Neumann boundary condition using the finite-difference method. Particularly, the ghost points are constructed to improve the convergence rate in approximating the normal derivative at the boundary points. In the context of the GPDM algorithm, the ghost points are employed to overcome the biases induced by the graph Laplacian discretization near the boundary. Numerically, solving PDEs with the ghost point method requires the following two steps: (1) specification of ghost points; and (2) specification of function values at the ghost points. While these two steps are trivial when the geometry is Euclidean or known, they require nontrivial numerical algorithms and theoretical justification when the manifold is unknown in the sense that it can only be identified with finitely sampled point cloud data.

The GPDM algorithm addresses step (1) above by augmenting the sampled point cloud data on the manifold with a set of ghost points specified on the outer normal collar of the

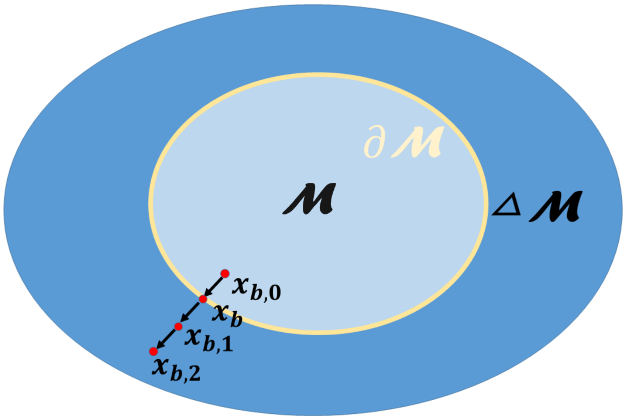


Figure 6.1: Construction of ghost points.

boundary. Theoretically, the GPDM algorithm extends the embedded manifold $\mathcal{M} \overset{\iota}{\hookrightarrow} \mathbb{R}^D$ with its collar neighbor $\Delta\mathcal{M}$ of a sufficiently large radius, such that the extended manifold $\mathcal{M} \cup \Delta\mathcal{M} \overset{\tilde{\iota}}{\hookrightarrow} \mathbb{R}^D$ is isometrically embedded and does not change the geometry of $\mathcal{M}$, i.e., $\tilde{\iota}|_{\mathcal{M}} = \iota$ (see Lemma 3.5 in [Jiang and Harlim, 2023]). With this modification, the graph Laplacian construction in (6.16) is a consistent pointwise estimator of $\mathcal{L}$ for all points in $\mathcal{M}$, even for the points that are very close to the boundary $\partial\mathcal{M}$ since they are sufficiently far away from the boundary of the extended manifold, $\partial(\mathcal{M} \cup \Delta\mathcal{M})$, as illustrated in Figure 6.1. Since our goal is to construct a forward map on the manifold $\mathcal{M}$, we need to specify the additional unknowns (the function values at these ghost points as we noted in the step (2)) by adding more equations. Specifically, we will impose a set of linear extrapolation equations, whose solution specifies the function values on the ghost points through function values on the point cloud.

While there are various ways to realize the two steps above, in the following we will provide a simple numerical procedure for well-ordered data that is used in the numerical examples in this chapter. For randomly sampled data or a higher convergence rate method, we refer interested readers to [Jiang and Harlim, 2023]. We should also point out that the presentation below uses much simpler notations compared to those in [Jiang and Harlim, 2023, Yan et al., 2023] since we are only interested in the algorithmic aspect of the GPDM method. Given a point cloud $\{x_n\}_{n=1}^N \subset \mathcal{M}$, the GPDM estimator for $\mathcal{L}$ is constructed as follows:

1. Specification of the ghost points. At each boundary point, $x_b \in \partial\mathcal{M}$:

   (a) Apply the secant line to approximate the normal vector at each boundary point. Following the notation in the illustration in Figure 6.1, the unit normal vector is estimated via
$$v_b = \frac{x_b - x_{b,0}}{\|x_b - x_{b,0}\|}.$$

   (b) We specify K ghost points along the normal vector at each boundary point, $x_b \in \partial\mathcal{M} \cap \{x_n\}_{n=1}^N$,
$$x_{b,k} := x_b + \delta k v_b, \quad k = 1, \ldots, K,$$
   where $\delta = \|x_b - x_{b,0}\|$ and $x_{b,0}$ is the closest point to x_b in $\mathcal{M}^o \cap \{x_n\}_{n=1}^N$.

2. Specification of the function values at the ghost points. For each boundary point x_b, we impose the following extrapolation condition,
$$\begin{aligned} u(x_{b,1}) - 2u(x_b) + u(x_{b,0}) &= 0, \\ u(x_{b,2}) - 2u(x_{b,1}) + u(x_b) &= 0, \\ u(x_{b,k}) - 2u(x_{b,k-1}) + u(x_{b,k-2}) &= 0, \qquad k = 3, \ldots, K. \end{aligned} \tag{6.21}$$
   These algebraic equations are discrete analogs of matching the first-order derivatives along the estimated normal direction, v_b.

3. Construction of the GPDM discrete estimator. Construct the graph Laplacian estimator as in (6.16) for the extended points, $\{x_n\}_{n=1}^N \cup \{x_{b,k}\}_{b,k=1}^{B,K}$. We point out that we also use a set of algebraic equations identical to (6.21) to determine the function value of κ at the ghost points $\{x_{b,k}\}_{b,k=1}^{B,K}$. Let $\bar{N} = N + BK$ and obtain the corresponding weighted Laplacian matrix, $\mathcal{L}_{\epsilon,\bar{N}}$, whose construction is analogous to (6.17). The

weighted Laplacian matrix can be rewritten as

$$\mathcal{L}_{\epsilon,\bar{N}} = \begin{pmatrix} \mathcal{L}^{(1)}_{\epsilon,\bar{N}} & \mathcal{L}^{(2)}_{\epsilon,\bar{N}} \\ \mathcal{L}^{(3)}_{\epsilon,\bar{N}} & \mathcal{L}^{(4)}_{\epsilon,\bar{N}} \end{pmatrix} \in \mathbb{R}^{\bar{N}\times\bar{N}}. \tag{6.22}$$

We will construct the GPDM matrix based on the sub-matrices $\mathcal{L}^{(1)}_{\epsilon,\bar{N}} \in \mathbb{R}^{N\times N}$, whose entries are constructed based on the affinity between pairs of the point cloud points in $\{x_n\}_{n=1}^{N} \subset \mathcal{M}$, and $\mathcal{L}^{(2)}_{\epsilon,\bar{N}} \in \mathbb{R}^{N\times BK}$, whose entries represent the affinity between an element of $\{x_n\}_{n=1}^{N}$ and a ghost point in $\{x_{b,k}\}_{b,k=1}^{B,K}$. Let $u_G = (u(x_{1,1}),\ldots,u(x_{B,K})) \in \mathbb{R}^{BK}$ and $u_N = (u(x_1),\ldots,u(x_{b,0}),\ldots,u(x_N)) \in \mathbb{R}^{N}$. Then we can write the solution of (6.21) in a compact form as $u_G = Gu_N$ such that $\mathcal{L}^{(1)}_{\epsilon,\bar{N}}u_N + \mathcal{L}^{(2)}_{\epsilon,\bar{N}}u_G = (\mathcal{L}^{(1)}_{\epsilon,\bar{N}} + \mathcal{L}^{(2)}_{\epsilon,\bar{N}}G)u_N$. Based on this observation, we define the GPDM estimator without boundary condition as

$$\tilde{\mathcal{L}}_{\epsilon,N} := \mathcal{L}^{(1)}_{\epsilon,\bar{N}} + \mathcal{L}^{(2)}_{\epsilon,\bar{N}}G \in \mathbb{R}^{N\times N}. \tag{6.23}$$

Denoting

$$\tilde{\mathcal{L}}_{\epsilon,N} = \begin{pmatrix} \tilde{\mathcal{L}}_{\epsilon,(N-B)\times(N-B)} & \tilde{\mathcal{L}}_{\epsilon,(N-B)\times B} \\ \tilde{\mathcal{L}}_{\epsilon,B\times(N-B)} & \tilde{\mathcal{L}}_{\epsilon,B\times B}, \end{pmatrix} \tag{6.24}$$

and splitting $u_N := (u_{N-B}, u_B)$ into function values at the interior and boundary points, respectively, the GPDM algorithm discretizes the Dirichlet problem in (6.19) as follows:

$$\begin{aligned} \tilde{\mathcal{L}}_{\epsilon,(N-B)\times(N-B)}u_{N-B} + \tilde{\mathcal{L}}_{\epsilon,(N-B)\times B}u_B = {} & f_{N-B}, \\ u_B = {} & h_B. \end{aligned} \tag{6.25}$$

Here, the components of the vector $f_{N-B} \in \mathbb{R}^{N-B}$ are the function values at the interior points, whereas the components of the vector $h_B \in \mathbb{R}^{B}$ are the function values at the boundary points.

6.3.2 Prior Specification and Discretization

In this section, we propose novel prior constructions to facilitate elliptic and heat Bayesian inversion on manifolds with boundaries. The priors we propose contain two terms. The first one is a Matérn-type Gaussian field with Dirichlet homogeneous boundary conditions, whose primary role is to capture the uncertainty on the parameter of interest in the interior of the manifold. The second term accounts for prior uncertainty along the boundary of the manifold, and can be omitted if the boundary values of the parameter are known.

For a one-dimensional manifold $\mathcal{M}$ with boundary $\partial\mathcal{M} = \{x_1, x_N\}$, i.e. a semi-ellipse in our numerical experiments, we propose using a prior defined as the law of

$$\theta = \underbrace{\frac{1}{\sqrt{\sum_{i=1}^{\infty}(\tau+\lambda_i)^{-s}}}\sum_{i=1}^{\infty}(\tau+\lambda_i)^{-\frac{s}{2}}\zeta_i\varphi_i}_{\text{Interior term}} + \underbrace{\mu_1\psi_1 + \mu_2\psi_2}_{\text{Boundary term}}. \tag{6.26}$$

In the interior term, τ, s, and $(\zeta_i)_{i=1}^{\infty}$ play the same role as in (6.9), but now $(\lambda_i, \varphi_i)_{i=1}^{\infty}$ are the ordered eigenpairs of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ equipped with homogeneous Dirichlet boundary conditions, so that $\varphi_i(x_1) = \varphi_i(x_N) = 0$ for all i. Nonhomogeneous Dirichlet boundary conditions could also be considered. In the boundary term, $\mu_1, \mu_2 \sim \mathcal{N}(0,1)$ are independent of all other randomness and ψ_1, ψ_2 are harmonic functions defined by

$$\begin{cases} \Delta_{\mathcal{M}}\psi_1 = 0, \\ \psi_1(x_1) = 1, \\ \psi_1(x_N) = 0, \end{cases} \quad \text{and} \quad \begin{cases} \Delta_{\mathcal{M}}\psi_2 = 0, \\ \psi_2(x_1) = 0, \\ \psi_2(x_N) = 1. \end{cases} \tag{6.27}$$

Thus the random coefficients μ_1 and μ_2 in (6.26) respectively represent the uncertainty on the parameter value of interest at the two boundary points x_1 and x_N. Overall, the prior defined by (6.26) combines the flexibility of the Matérn model with homogeneous Dirichlet boundary conditions for interior reconstruction with additional flexibility in the boundary reconstruction. A simulation study that illustrates the increased flexibility afforded by the boundary term and our prior construction is given in Section 6.4.

Following the same idea, for a two-dimensional manifold $\mathcal{M}$ with boundary $\partial\mathcal{M} = \mathcal{B}_1 \cup \mathcal{B}_2$, where $\mathcal{B}_1$ and $\mathcal{B}_2$ are disjoint regular closed curves, e.g. in our numerical examples $\mathcal{B}_1$ and $\mathcal{B}_2$ are boundary circles of a semi-torus, we define a prior as the law of

$$\theta = \underbrace{\frac{1}{\sqrt{\sum_{i=1}^{\infty}(\tau+\lambda_i)^{-s}}}\sum_{i=1}^{\infty}(\tau+\lambda_i)^{-\frac{s}{2}}\zeta_i\varphi_i}_{\text{Interior term}} + \underbrace{\sum_{\ell=1}^{L}\mu_{1,\ell}\psi_{1,\ell} + \sum_{\ell=1}^{L}\mu_{2,\ell}\psi_{2,\ell}}_{\text{Boundary term}} . \tag{6.28}$$

The interior term is defined analogously as in the one-dimensional case, using the spectrum of the Laplace-Beltrami operator with homogeneous Dirichlet boundary conditions. For the boundary term, $\mu_{1,\ell}, \mu_{2,\ell} \sim \mathcal{N}(0,1)$ are independent of all other randomness, and $\{\psi_{1,\ell}\}_{\ell=1}^{L}$ and $\{\psi_{2,\ell}\}_{\ell=1}^{L}$ are harmonic functions on $\mathcal{M}$ defined by

$$\begin{cases} \Delta_{\mathcal{M}}\psi_{1,\ell} = 0, \\ \psi_{1,\ell}(x) = \tilde{\psi}_{1,\ell}(x), \quad x \in \mathcal{B}_1, \\ \psi_{1,\ell}(x) = 0, \quad x \in \mathcal{B}_2, \end{cases} \qquad \text{and} \qquad \begin{cases} \Delta_{\mathcal{M}}\psi_{1,\ell} = 0, \\ \psi_{2,\ell}(x) = 0, \quad x \in \mathcal{B}_1, \\ \psi_{2,\ell}(x) = \tilde{\psi}_{2,\ell}(x), \quad x \in \mathcal{B}_2, \end{cases} \qquad \text{for } \ell = 1, \dots, L, \tag{6.29}$$

where $\tilde{\psi}_{1,\ell}(x)$ and $\tilde{\psi}_{2,\ell}(x)$ are ordered eigenfunctions of the Laplace-Beltrami operator defined on the closed curves $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. The number L of basis-type functions for each boundary controls the flexibility of the prior along the boundary. Larger L allows to recover more frequencies of the parameter of interest along the boundary, but at the expense of introducing additional model parameters.

To discretize the prior defined in (6.26), we simply replace the role of the Laplace-Betrami operator $\Delta_{\mathcal{M}}$ with a graph Laplacian as we did in Subsection 6.2.3, but now taking care of boundary conditions. Specifically, for one-dimensional inverse problems, our discretized prior distribution would have samples of the form

$$\theta_N \sim \frac{\sqrt{N}}{\sqrt{\sum_{n=1}^{N}(\tau+\lambda_n^{(N)})^{-s}}}\sum_{n=1}^{N}(\tau+\lambda_n^{(N)})^{-\frac{s}{2}}\zeta_n\varphi_n^{(N)} + \mu_1\psi_1^{(N)} + \mu_2\psi_2^{(N)}, \tag{6.30}$$

where $(\lambda_n^{(N)}, \varphi_n^{(N)})$ are ordered eigenpairs of $\tilde{\Delta}_N$ defined as the $N \times N$ submatrix corresponding to the point cloud portion of the graph Laplacian (6.10) constructed using both the point cloud $\{x_n\}_{n=1}^{N}$ and ghost points $\{x_{b,k}\}_{b,k=1}^{B,K}$. From now on, we will refer to $\tilde{\Delta}_N$ as a *truncated graph Laplacian*. More specifically, given the point cloud and ghost points, we define $\tilde{\Delta}_N$ as an $N \times N$ submatrix of the $\bar{N} \times \bar{N}$ matrix $\Delta_{\bar{N}}$ with $\bar{N} := N + BK$. Indices of the submatrix $\tilde{\Delta}_N$ correspond to that of the elements in the point cloud. We have observed such a construction provided an effective approximation to the Laplace-Beltrami operator on $\mathcal{M}$ with homogeneous Dirichlet boundary conditions in our numerical experiments. This construction avoids any potential idiosyncratic boundary behavior of eigenvectors when using a graph Laplacian constructed solely with point cloud data. Indeed, in our numerical experiments (see Figure 6.6 and the associated discussion in Section 6.4 below) we note that the eigenvectors of the self-tuned graph Laplacian (6.10) have spikes and oscillations near the boundary. Finally, $\psi_1^{(N)}$ and $\psi_2^{(N)}$ are the solutions of (6.27) evaluated along the point cloud. These solutions can be obtained by the GPDM algorithm as described in Section 6.3.1.

An alternative attempt to discretize the homogeneous Dirichlet boundary condition was introduced in [Thiede et al., 2019] where they employed truncation on the original point clouds without adding ghost points. In fact, the spectral convergence of the truncated graph Laplacian to the Dirichlet Laplacian on manifold with boundaries has recently been reported

[Peoples and Harlim, 2021]. We should also point out that there were two discretizations of the Laplace-Beltrami operator introduced in this chapter: 1) self-tuned normalized symmetric graph Laplacian given in (6.10) and 2) ghost point diffusion map(GPDM) matrix given in (6.23). The eigenvectors of the former are orthogonal as the matrix is symmetric, whereas the eigensolutions of the latter are not the case since the GPDM matrix in (6.23) is not symmetric. With this fact, we clarify that we only use GPDM to solve linear problems (6.27) and do not use the eigenvectors of the corresponding GPDM matrix elsewhere.

For the two-dimensional graph-based prior discretization, we obtain a finite number of discretized functions for each boundary, denoted by $\{\psi_{1,\ell}^{(N)}\}_{\ell=1}^{L}$ and $\{\psi_{2,\ell}^{(N)}\}_{\ell=1}^{L}$, by solving (6.29) along the point cloud. One can again use the GPDM algorithm to obtain $\{\psi_{1,\ell}^{(N)}\}_{\ell=1}^{L}$ and $\{\psi_{2,\ell}^{(N)}\}_{\ell=1}^{L}$ where the boundary basis functions $\{\tilde{\psi}_{1,\ell}\}_{\ell=1}^{L}$ and $\{\tilde{\psi}_{2,\ell}\}_{\ell=1}^{L}$ are discretized using a self-tuned graph Laplacian constructed with all the point cloud elements that lie on the boundaries, which are two disjoint closed curves in our numerical examples. For instance, to obtain $\{\tilde{\psi}_{1,\ell}\}_{\ell=1}^{L}$ one can construct the self-tuned graph Laplacian solely using points in $\mathcal{B}_1 \cap \{x_n\}_{n=1}^{N}$, where $\{x_n\}_{n=1}^{N} \subset \mathcal{M}$ is the point cloud of the manifold. Then $\{\tilde{\psi}_{1,\ell}\}_{\ell=1}^{L}$ can be chosen to be the first L eigenvectors of this self-tuned graph Laplacian. To summarize, in the two-dimensional case, samples from the proposed graph-based prior are defined by

$$\theta_N \sim \frac{\sqrt{N}}{\sqrt{\sum_{n=1}^{N}(\tau+\lambda_n^{(N)})^{-s}}}\sum_{n=1}^{N}(\tau+\lambda_n^{(N)})^{-\frac{s}{2}}\zeta_n\varphi_n^{(N)} + \sum_{\ell=1}^{L}\mu_{1,\ell}\psi_{1,\ell}^{(N)} + \sum_{\ell=1}^{L}\mu_{2,\ell}\psi_{2,\ell}^{(N)}, \quad (6.31)$$

where $(\lambda_n^{(N)}, \varphi_n^{(N)})_{n=1}^{N}$ are ordered eigenpairs of a truncated graph Laplacian.

6.3.3 Forward Map Discretization

Using the tools introduced in previous sections, here we propose graph-based discretizations of forward maps for elliptic and heat inverse problems on one and two-dimensional manifolds with boundary.

Elliptic Inverse Problem For the inversion problem involving the elliptic PDE in (6.19), our goal is to learn the diffusion coefficient κ on $\mathcal{M}$. Correspondingly, analogous to the boundary-free setting, a forward map was given by $\mathcal{F} : \theta \mapsto u$, where $\theta = \log \kappa \in (-\infty, \infty)$ and u solves (6.19). In this case, we can use the GPDM algorithm introduced in Section 6.3.1 to obtain the discretization of the forward map. Precisely, the discretized forward map is given by

$$\mathcal{F}_{\epsilon,N} : \theta_N \mapsto u_N = (u_{N-B}, u_B) = \left(\tilde{\mathcal{L}}^{-1}_{\epsilon,(N-B)\times(N-B)} (f_{N-B} - \tilde{\mathcal{L}}_{\epsilon,(N-B)\times B} h_B), h_B \right),$$

where $\theta_N = \big(\log(\kappa(x_1)), \ldots, \log(\kappa(x_N))\big)$ and $\tilde{\mathcal{L}}^{-1}_{\epsilon,(N-B)\times(N-B)}$ denotes the pseudo-inverse. From this discretization process, we arrive at the relationship between the discretized posterior distribution μ_N^y and the discretized prior distribution μ_N, which is given by

$$\frac{d\mu_N^y}{d\mu_N}(\theta_N) \propto \exp\left(-\frac{1}{2} |y - \mathcal{G}_{\epsilon,N}(\theta_N)|_\Gamma^2 \right),$$

where $\mathcal{G}_{\epsilon,N}(\theta_N) = \big(u_N(\tilde{x}_1), \ldots, u_N(\tilde{x}_M)\big)$. For the sampling and interpolation steps, we follow the general methodology in Section 6.2.2.

Heat Inversion Consider first the one-dimensional heat equation with Dirichlet boundary conditions given by (6.20) with $\partial\mathcal{M} = \{x_1, x_N\}$. Analogous to the prior construction, we introduce two time-independent functions ψ_1 and ψ_2 defined on $\mathcal{M}$ satisfying

$$\begin{cases} \Delta_{\mathcal{M}} \psi_1 = 0, \\ \psi_1(x_1) = 1, \\ \psi_1(x_N) = 0, \end{cases} \quad \text{and} \quad \begin{cases} \Delta_{\mathcal{M}} \psi_2 = 0, \\ \psi_2(x_1) = 0, \\ \psi_2(x_N) = 1. \end{cases}$$

Then we can obtain the solution of (6.20) by solving for w that satisfies,

$$\begin{cases} w_t(x,t) = -\Delta_{\mathcal{M}} w(x,t), \quad x \in \mathcal{M}^o, t > 0, \\ w(x_1,t) = 0,\ w(x_N,t) = 0, \quad t \geq 0, \\ w(x,0) = \theta(x) - h(x_1)\psi_1(x) - h(x_N)\psi_2(x), \quad x \in \mathcal{M}. \end{cases} \tag{6.32}$$

To see this, notice that the function

$$u^*(x,t) \coloneqq w(x,t) + h(x_1)\psi_1(x) + h(x_N)\psi_2(x) \tag{6.33}$$

is the solution of (6.20).

For the two-dimensional heat equation, let $\tilde{\psi}_{1,\ell}$ and $\tilde{\psi}_{2,\ell}$ be ordered eigenfunctions of the Laplace-Beltrami operator defined on $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. We can then write

$$h_1(x) = \sum_{\ell=1}^{\infty} a_\ell \tilde{\psi}_{1,\ell}(x), \quad x \in \mathcal{B}_1, \quad \text{and} \quad h_2(x) = \sum_{\ell=1}^{\infty} b_\ell \tilde{\psi}_{2,\ell}(x), \quad x \in \mathcal{B}_2,$$

for some real coefficients $\{a_\ell\}_{\ell=1}^{\infty}$ and $\{b_\ell\}_{\ell=1}^{\infty}$. The solution to (6.20) is then given by

$$u^*(x,t) \coloneqq w(x,t) + \sum_{\ell=1}^{\infty} a_\ell \psi_{1,\ell}(x) + \sum_{\ell=1}^{\infty} b_\ell \psi_{2,\ell}(x), \tag{6.34}$$

where w is the solution of

$$\begin{cases} w_t(x,t) = -\Delta_{\mathcal{M}} w(x,t), \quad x \in \mathcal{M}^o, \quad t > 0, \\ w(x,t) = 0, \quad x \in \mathcal{B}_1, \quad t \geq 0, \\ w(x,t) = 0, \quad x \in \mathcal{B}_2, \quad t \geq 0, \\ w(x,0) = \theta(x) - \sum_{\ell=1}^{\infty} a_\ell \psi_{1,\ell}(x) - \sum_{\ell=1}^{\infty} b_\ell \psi_{2,\ell}(x), \quad x \in \mathcal{M}. \end{cases} \tag{6.35}$$

Here, $\psi_{1,\ell}$ is the harmonic function with boundary condition $\tilde{\psi}_{1,\ell}$ in $\mathcal{B}_1$ and 0 in $\mathcal{B}_2$, while $\psi_{2,\ell}$ is the harmonic function with boundary condition 0 in $\mathcal{B}_1$ and $\tilde{\psi}_{2,\ell}$ in $\mathcal{B}_2$. We have hence expressed the solution of the heat equation with non-homogeneous boundary conditions as a linear superposition of basis-like functions and the solution of homogeneous heat equation. Such a decomposition will play a key role in the discretization procedures.

Now suppose that in the one-dimensional case we are given an N-dimensional discrete representation θ_N of the initial heat distribution of the form

$$\theta_N = \psi_3^{(N)} + \mu_1 \psi_1^{(N)} + \mu_2 \psi_2^{(N)}, \text{ where } \psi_3^{(N)} = \frac{\sqrt{N}}{\sqrt{\sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-s}}} \sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-\frac{s}{2}} \zeta_n \varphi_n^{(N)},$$

which reflects our prior construction given in the previous section. To discretize the forward map of the one-dimensional heat equation, we replace the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ by $\tilde{\Delta}_N$ and solve for $w^{(N)} = (w_1, \ldots, w_N)$ satisfying

$$\begin{cases} \frac{\partial}{\partial t} w^{(N)} = -\tilde{\Delta}_N w^{(N)}, \\ w^{(N)}(0) = \psi_3^{(N)}, \end{cases} \tag{6.36}$$

This equation can be viewed as a discrete analog of (6.32) as the eigenvectors of $\tilde{\Delta}_N$ approximate those of the Laplace-Beltrami operator with homogeneous Dirichlet boundary condition. Given an initial condition $\psi_3^{(N)} = \theta_N - \mu_1 \psi_1^{(N)} - \mu_2 \psi_2^{(N)}$, the solution of the above initial value problem can be expressed as a linear combination of the eigenvectors of $\tilde{\Delta}_N$. In other words, the solution at time t is given by

$$w^{(N)}(t) = \sum_{n=1}^{N} \langle \varphi_n^{(N)}, \psi_3^{(N)} \rangle e^{-\lambda_n^{(N)} t} \varphi_n^{(N)}.$$

From this, and the previous observations we made in (6.33), we naturally obtain the discrete approximation for the solution of (6.20) given by

$$u_N = w^{(N)} + \mu_1 \psi_1^{(N)} + \mu_2 \psi_2^{(N)},$$

which defines the discretized forward map $\mathcal{F}_N : \theta_N \mapsto u_N$.

Using the same argument, we can discretize the forward map for the two-dimensional heat equation. Suppose we are given an N-dimensional discrete representation of the initial heat function for two-dimensional heat inverse problem, denoted by θ_N, of the form

$$\begin{aligned}\theta_N &= \psi_3^{(N)} + \sum_{\ell=1}^{L} \mu_{1,\ell} \psi_{1,\ell}^{(N)} + \sum_{\ell=1}^{L} \mu_{2,\ell} \psi_{2,\ell}^{(N)}, \\ \text{where } \psi_3^{(N)} &= \frac{\sqrt{N}}{\sqrt{\sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-s}}} \sum_{n=1}^{N} (\tau + \lambda_n^{(N)})^{-\frac{s}{2}} \zeta_n \varphi_n^{(N)}.\end{aligned}$$

Similarly as in the one-dimensional case, with the observation (6.34), the discrete approximation for the solution of (6.20) in the two-dimensional case is given by

$$u_N = w^{(N)} + \sum_{\ell=1}^{L} \mu_{1,\ell} \psi_{1,\ell}^{(N)} + \sum_{\ell=1}^{L} \mu_{2,\ell} \psi_{2,\ell}^{(N)},$$

where $w^{(N)}$ denotes the solution of (6.36). Accordingly, the discretized forward map is given by $\mathcal{F}_N : \theta_N \mapsto u_N$. In both the one and two-dimensional cases the discretized posterior distribution μ_N^y and the discretized prior distribution μ_N satisfy the relationship

$$\frac{d\mu_N^y}{d\mu_N}(\theta_N) \propto \exp\left(-\frac{1}{2}|y - \mathcal{G}_N(\theta_N)|_\Gamma^2\right),$$

where $\mathcal{G}_N(\theta_N) = \big(u_N(\tilde{x}_1), \ldots, u_N(\tilde{x}_M)\big)$. For the sampling and interpolation steps, we again follow the general methodology described in Section 6.2.2.

6.4 Numerical Results

In this section, we provide simulation results for the numerical solution of Bayesian inverse problems on manifolds with boundaries. We showcase our methodology for the elliptic inverse problem in Section 6.4.1 and for the heat inverse problem in Section 6.4.2. For each inverse problem, we consider one-dimensional and two-dimensional examples. In addition to validating our approach and providing implementation details, our numerical results will demonstrate the enhanced flexibility of our proposed priors near the boundary when compared with previous graph representations of Matérn priors. Specifically, we show the improved reconstruction achieved by our prior in a one-dimensional elliptic inverse problem, and we illustrate in a two-dimensional setting the emergence of artifacts near the boundary for the eigenfunctions of the graph Laplacian (6.10) used to define graph Matérn priors on closed manifolds in [Harlim et al., 2020].

Our one-dimensional examples are set on a semi-ellipse and the two-dimensional examples are set on a semi-torus. For the semi-ellipse the embedding was given by

$$\iota(\alpha) = \begin{pmatrix} \cos\alpha \\ 3\sin\alpha \end{pmatrix}, \ \alpha \in [0, \pi], \tag{6.37}$$

with Riemannian metric

$$g = \sin^2(\alpha) + 9\cos^2(\alpha). \tag{6.38}$$

The embedding for the semi-torus was given by

$$\iota(\alpha, \beta) = \begin{pmatrix} (2+\cos\alpha)\cos\beta \\ (2+\cos\alpha)\sin\beta \\ \sin\alpha \end{pmatrix}, \ \alpha \in [0, 2\pi], \ \beta \in [0, \pi], \tag{6.39}$$

where (α, β) are the intrinsic coordinates and the corresponding Riemannian metric is given by

$$g = \begin{pmatrix} 1 & 0 \\ 0 & (2+\cos\alpha)^2 \end{pmatrix}. \tag{6.40}$$

We use uniform grids to define point clouds over the given manifolds. For the one-dimensional semi-ellipse we used $N = 630$ points and for the two-dimensional semi-torus $N = N_1 \times N_2 = 1296$ points in a 36 by 36 grid. The boundary of the semi-ellipse corresponds to $\alpha = 0$ or $\alpha = \pi$, which necessitates to model boundary values of the parameter on $\alpha = 0, \pi$. For the semi-torus the boundary corresponds to $\beta = 0$ or $\beta = \pi$, which implies the need to model boundary values of the parameter along two boundary curves. In all of our numerical experiments we choose $N = M$, that is, we assume to have observations along the entire point cloud.

6.4.1 Elliptic Inverse Problem

For the one-dimensional elliptic inverse problem, we set the true PDE solution $u^\dagger$ to be

$$u^\dagger(\alpha) = \sin(\alpha),$$

and for the two-dimensional elliptic inverse problem, the true PDE solution was given by

$$u^\dagger(\alpha, \beta) = 10\sin(2\alpha)\cos(\beta).$$

While the one-dimensional solution satisfies homogeneous Dirichlet boundary condition at $\alpha = 0, \pi$, the two-dimensional solution has non-trivial Dirichlet boundary condition at $\beta = 0$ and π. According to the above true solution, the observations were given by

$$y_n = u^\dagger(\alpha_n) + \eta_n, \ \alpha_n = \iota^{-1}(x_n), \ n = 1, \dots, N,$$

where $\eta_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.01)$. We consider several choices of true input parameter κ, and for each choice we define the right-hand side of the PDE using the identity

$$f = -\text{div}(\kappa \nabla u) = -\frac{1}{\sqrt{\text{det} g}} \partial_i \left(\kappa g^{ij} \partial_j u \sqrt{\text{det} g} \right).$$

Note that here and henceforth we abuse notation by referring to $u^\dagger \circ \iota$ as $u^\dagger$.

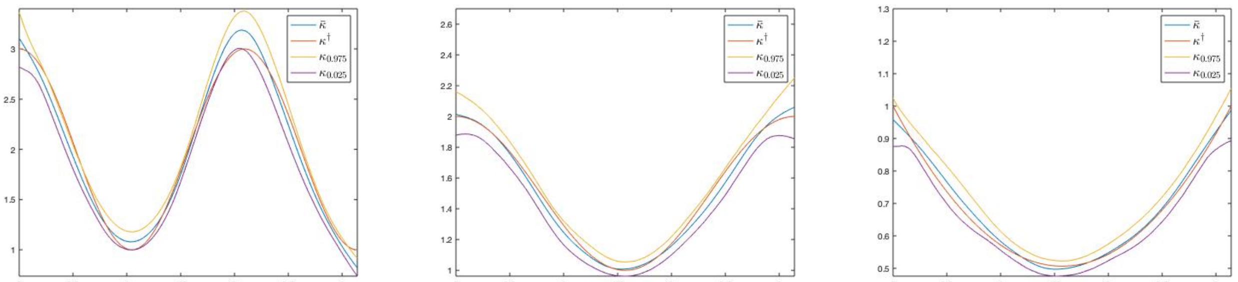
One-dimensional Manifold

For the one-dimensional elliptic inverse problem, we first augmented the given manifold point cloud data, adding 10 ghost points at each boundary point. To model the interior term from the proposed prior, we constructed a self-tuned graph Laplacian using both the manifold point cloud and ghost points. After obtaining the graph Laplacian, we truncated it to obtain a submatrix whose indices correspond to the manifold point cloud elements. Recall that this matrix was referred to as a truncated graph Laplacian in Section 6.3.2. We used two nearest neighbors to construct the self-tuned graph Laplacian. For the two boundary terms, we used the GPDM algorithm to obtain two harmonic functions whose boundary values were either one or zero. These harmonic functions will allow us to model boundary values, i.e. values at $\alpha = 0, \pi$, of the diffusion coefficient. When constructing the weighted Laplacian matrix, we used 51 closest points in (6.18) and chose the value of ϵ which attained the maximum slope of $\log\big(T(\epsilon)\big) \approx \frac{1}{2}$.

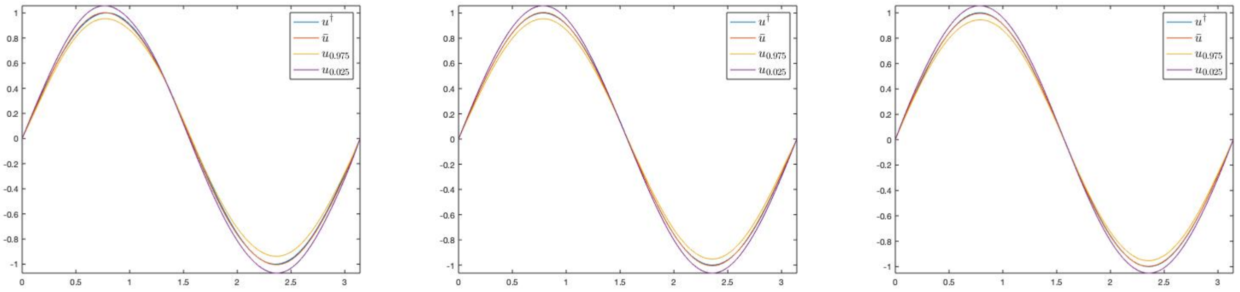
We set the smoothness prior parameter to be $s = 4$, and the inverse length-scale parameter to be $\tau = 0.2$, where the choice of these values is based on empirical experiments. Hierarchical Bayesian formulations to learn these parameters could be considered [Sanz-Alonso and Yang, 2022a]. In practice, instead of using all eigenvectors of the truncated graph Laplacian, one can use a sufficiently large subset of eigenvectors. In our simulation study, we used 20 eigenvectors of the truncated graph Laplacian to represent the interior term. Furthermore, to attain an acceptance rate between 40 and 60 percent, we used $\zeta = 0.01$ for the graph pCN algorithm. We computed a total of 10000 MCMC iterations with an initial 5000 burn-in period. The results are shown in Figure 6.2, where three different choices of diffusion coefficients are considered.

Observe from Figure 6.2 that the three true diffusion coefficients considered lie, for the most part, inside of the 95 percent credible intervals. Moreover, the PDE solutions obtained using the recovered coefficients were all very close to the PDE solution with the true coefficients. To showcase the flexibility of our prior compared to the one proposed in [Harlim et al., 2020], additional numerical experiments were conducted. While using the same forward map approximation given by the GPDM algorithm, we employed priors proposed in [Harlim et al., 2020] where the graph Laplacian is constructed solely from the point cloud on the manifold. We first present the recovery results for $\kappa_1(\alpha) = 2 + \cos(3\alpha)$ and $\kappa_2(\alpha) = 1 + \cos^2(\alpha)$ with the same semi-ellipse manifold as in Figure 6.2. All the parameter values for priors and pCN algorithms were set to be identical as before. The results are shown in Figure 6.3.

Figures 6.3a and 6.3b show that the recovered diffusion coefficients using the priors in [Harlim et al., 2020] have artificial spikes near the boundary. The prior in [Harlim et al., 2020] is only effective when the underlying manifold is closed, in which case the graph Laplacian used in [Harlim et al., 2020] approximates the Laplace-Beltrami operator on the underlying manifold. However, when the underlying manifold has a boundary, the prior in [Harlim et al., 2020] leads to the formation of spikes near the boundaries and it does not allow for flexible modeling of boundary conditions. To illustrate this point further, we performed the Bayesian inversion procedure for the one-dimensional elliptic equation on an ellipse restricted to the first quadrant (i.e. $\alpha \in \left[0, \frac{\pi}{2}\right]$) instead of the semi-ellipse. The true diffusion coefficient was set to be $\kappa = 2 + \cos(3\alpha)$. All the parameter values were again the same as before. Figure 6.4 demonstrates the flexibility of our proposed prior, while the reconstruction using the prior in [Harlim et al., 2020] exhibits artifacts near the boundary

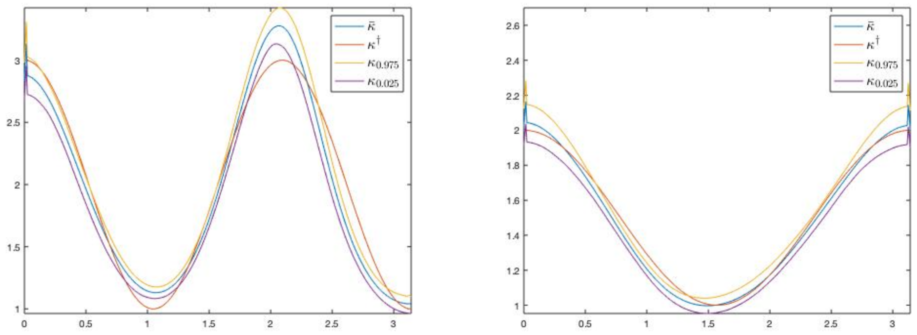


(a) Recovery of $\kappa_1(\alpha) = 2 + \cos(3\alpha)$ (b) Recovery of $\kappa_2(\alpha) = 1 + \cos^2(\alpha)$ (c) Recovery of $\kappa_3 = 1 + \frac{\alpha(\alpha-\pi)}{5}$



(d) Recovered u_N of κ_1 (e) Recovered u_N of κ_2 (f) Recovered u_N of κ_3

Figure 6.2: One-dimensional elliptic inverse problem on a semi-ellipse: Top row: reconstruction of κ. Bottom row: solution of an elliptic PDE corresponding to the κ given right above.



(a) Recovery of $\kappa_1(\alpha) = 2 + \cos(3\alpha)$ (b) Recovery of $\kappa_2(\alpha) = 1 + \cos^2(\alpha)$

Figure 6.3: Elliptic inverse problem on a semi-ellipse with Matérn-type prior in [Harlim et al., 2020].

and appears to incorrectly suggest a homogeneous Neumann boundary condition for κ. In the next subsection we provide further understanding of these phenomena by illustrating the different terms involved in the definition of our prior and the one in [Harlim et al., 2020] in a two-dimensional setting.

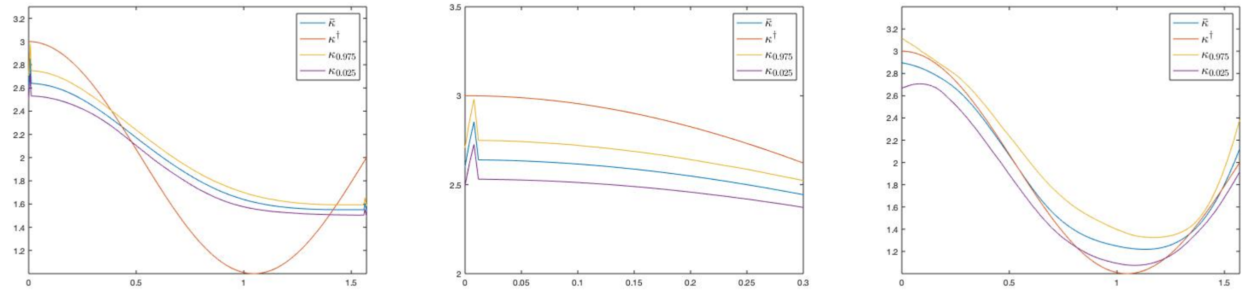


Figure 6.4: Comparison of the performance of two priors in the reconstruction of $\kappa_1(\alpha) = 2 + \cos(3\alpha)$ with $\alpha \in \left[0, \frac{\pi}{2}\right]$. Left: using the prior proposed in [Harlim et al., 2020]. Middle: zoom-in of the left figure near the boundary. Right: using our proposed prior.

Two-dimensional Manifold

Similarly, as in the one-dimensional elliptic PDE, we augmented the given point cloud with 216 ghost points for each boundary. To define the boundary part of our prior, for each boundary (which is a circle in this example) we construct a self-tuned graph Laplacian using only the observations on the boundary with two-nearest neighbors. Then we obtain the 10 eigenvectors corresponding to the smallest 10 eigenvalues for each boundary. These eigenvectors will serve the role of discretized basis functions for each boundary. Solving 20 different Laplace-type equations in (6.29) using the GPDM algorithm, one can obtain 10 basis-like functions for each boundary, which would correspond to $\{\psi_{1,\ell}^{(N)}\}_{\ell=1}^{10}$ and $\{\psi_{2,\ell}^{(N)}\}_{\ell=1}^{10}$ in (6.31). These basis-like functions allow flexible model of functions in the vicinity of each boundary curve. In particular, in our example the true diffusion coefficient was set to $\kappa(\alpha, \beta) = 10 + 8\sin(\alpha)\cos(\beta)$. Therefore, the values we would like to capture along each boundary would be $\kappa(\alpha,0) = 10 + 8\sin(\alpha)$ and $\kappa(\alpha,\pi) = 10 - 8\sin(\alpha)$. When constructing the weighted Laplacian matrix, we used 128 closest points in (6.18) and chose the value of ϵ which attained the maximum slope of $\log\big(T(\epsilon)\big) \approx 1$.

For the interior part of the proposed prior, analogously to the one-dimensional case, we truncated a self-tuned graph Laplacian constructed using the augmented dataset and obtained a submatrix whose indices correspond to the elements in the point cloud. Taking eigenvalues and eigenvectors of this truncated graph Laplacian would give $(\lambda_n^{(N)}, \varphi_n^{(N)})$ in (6.31). We used four-nearest neighbors to construct the self-tuned graph Laplacian. We set the smoothness prior parameter to be $s = 4$, and the inverse length-scale parameter to be $\tau = 0.24$, where the choice of these values is based on empirical experiments. To attain an acceptance rate between 40 and 60 percent, we used $\zeta = 0.001$ for the graph pCN algorithm. A total of 150000 MCMC iterations with initial 75000 burn-in iterations were run to obtain our results.

Starting from the top row of Figure 6.5, the panels in the first row represent the true diffusion coefficient and the solution of the elliptic PDE based on the true diffusion coefficient. The second row represents, from left to right, the posterior mean of the MCMC samples, 2.5 percentile of MCMC samples and 97.5 percentile of MCMC samples. In other words, the second row portrays the credible interval for the true diffusion coefficient function. The third row corresponds to the solution of the elliptic PDE equation based on the diffusion coefficients given in the second row. The last two plots in the fourth row depict the difference between the true diffusion coefficient and the posterior mean of the MCMC samples, and the difference between the true solution with the one based on the posterior mean of the MCMC samples.

For the two-dimensional elliptic inverse problem, there is a larger error in the recovery of the true diffusion coefficient. This is unsurprising, since as one can see from the solutions corresponding to the MCMC samples of the diffusion coefficients, one can find several different diffusion coefficients that lead to approximate solutions that are close to the solution of the elliptic PDE, which is a manifestation of the ill-posedness of this elliptic inverse problem.

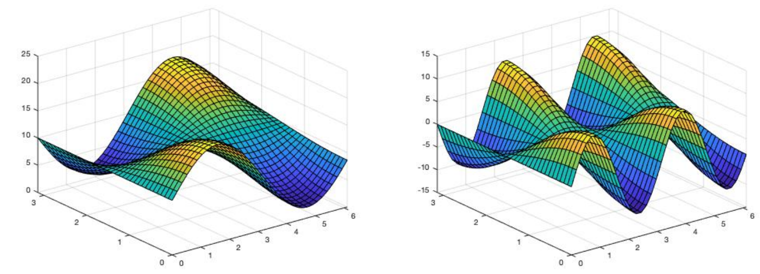
To illustrate the strength of the proposed prior relative to [Harlim et al., 2020], Figure 6.6 includes surface plots of (i) the first two eigenfunctions of graph Laplacian used to define the prior in [Harlim et al., 2020]; (ii) the first two eigenfunctions, i.e., φ_1^N, φ_2^N of the truncated graph Laplacian used to define the interior term of our prior; and (iii) the boundary terms $\psi_{1,2}^N, \psi_{1,3}^N$ in (6.31). As one can see in the leftmost column in Figure 6.6 the graph Laplacian used in the prior construction in [Harlim et al., 2020] is not appropriate for manifolds with boundary. In particular, the spikes in the first eigenfunction can be explained by the use of a symmetric graph Laplacian and the fact that the degree of nodes close to the boundary is significantly different than the degree of nodes in the interior. However, the samples we propose do not possess such undesirable behavior as the boundary values are solely modeled by boundary components which are in the middle column of Figure 6.6. The interior term is modeled by superposition of eigenfunctions of the homogeneous Dirichlet eigenvalue problem (rightmost column in Figure 6.6) approximated using a truncated graph Laplacian.

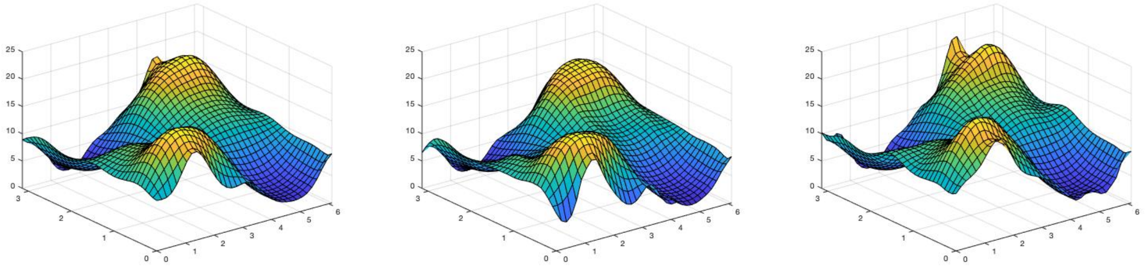
6.4.2 Heat Inverse Problem

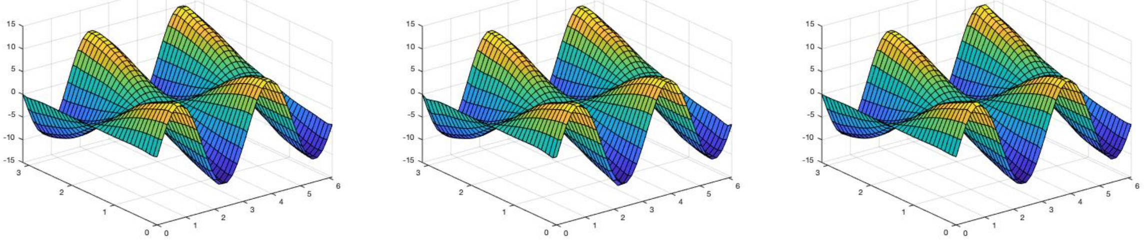
For the one-dimensional numerical simulations of heat inversion, given an initial heat function u_0, observations were obtained by

$$y_n = \tilde{u}(\alpha_n, t^*) + \eta_n,\ \alpha_n = \iota^{-1}(x_n),\ n = 1, \ldots, N,$$

where $\eta_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.01)$ and $\tilde{u}$ represents an approximate solution of the heat equation given in (6.20). We assume that the boundary is given by $\{x_1, x_N\}$. For the one-dimensional semi-ellipse, we used the following explicit formula to compute the approximate solution over the


(a) True diffusion coefficient κ (b) True PDE solution u


(c) Posterior mean for κ (d) 2.5% post. percentile for κ (e) 97.5% post. percentile for κ


(f) u_N with κ in Figure 6.5c (g) u_N with κ in Figure 6.5d (h) u_N with κ in Figure 6.5e

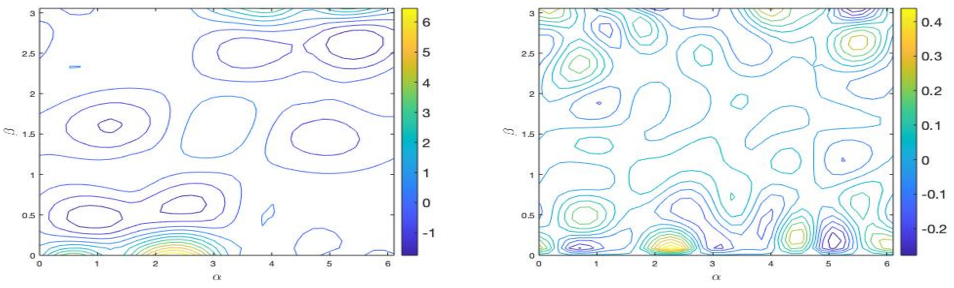
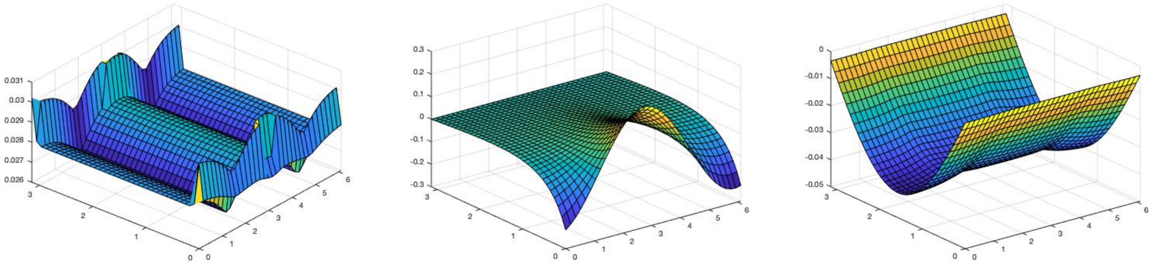
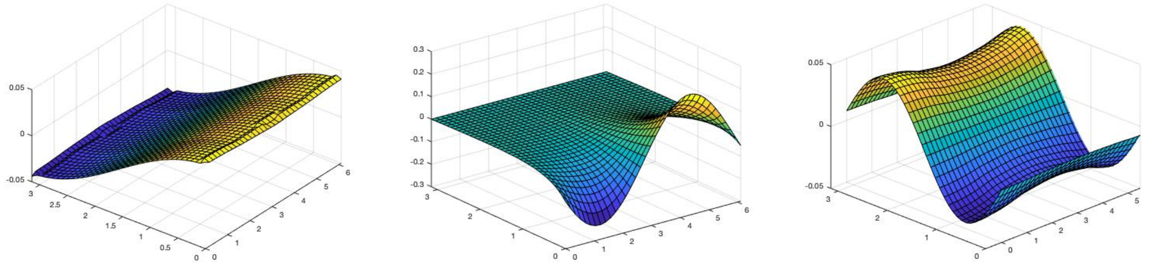

(i) Error in diffusion coefficient (j) Error in PDE solution

Figure 6.5: Elliptic inverse problem on a semi-torus for $\kappa(\alpha, \beta) = 10 + 8\sin(\alpha)\cos(\beta)$.



(a) First eigenfunction (b) First boundary component (c) First interior component



(d) Second eigenfunction (e) Second boundary component (f) Second interior component

Figure 6.6: Representation of terms used to define prior draws on a semi-torus. Left column: first two eigenfunctions of the covariance matrix for the prior proposed by [Harlim et al., 2020]. Middle column: first two (excluding the constant one) boundary components in our proposed prior. Right column: first two interior terms in our proposed prior.

point cloud at time $t^* > 0$, denoted by $\tilde{u}_N = (\tilde{u}(\alpha_1, t^*), \cdots, \tilde{u}(\alpha_N, t^*))$,

$$\tilde{u}_N \coloneqq \sum_{n=1}^{N} \langle \varphi_n^{(N)}, \psi_3^{(N)} \rangle e^{-\lambda_n^{(N)} t^*} \varphi_n^{(N)} + u_0(\alpha_1)\psi_1^{(N)} + u_0(\alpha_N)\psi_2^{(N)},$$

where $(\lambda_n^{(N)}, \varphi_n^{(N)})_{n=1}^N$ are eigenpairs of the truncated graph Laplacian constructed using point cloud $\{x_n\}_{n=1}^N$ with two boundary points $\{x_1, x_N\}$. The definitions of $\psi_1^{(N)}$, $\psi_2^{(N)}$ and $\psi_3^{(N)}$ are given in Section 6.3.2. Recall that $\psi_1^{(N)}$, $\psi_2^{(N)}$ are responsible for modelling values of the parameter at $\{x_1, x_N\}$ while $\psi_3^{(N)}$ determines the interior values of the parameter.

Similarly, for the two-dimensional semi-torus, given an initial heat function u_0, observations were obtained by

$$y_n = \tilde{u}((\alpha_{n_1}, \beta_{n_2}), t^*) + \eta_n, \ (\alpha_{n_1}, \beta_{n_2}) =: \iota^{-1}(x_n), \ n = 1, \ldots, N,$$

where $\eta_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 0.01)$ and $\tilde{u}$ represents the approximate solution for (6.20). For the two-dimensional setting, the approximate solution

$$\tilde{u}_N = (\tilde{u}(\alpha_1, \beta_1, t^*), \cdots, \tilde{u}(\alpha_{N_1}, \beta_{N_2}, t^*))$$

over the point cloud at time $t^* > 0$, with $N = N_1 \times N_2$, is given by

$$\tilde{u}_N \coloneqq \sum_{n=1}^{N} \langle \varphi_n^{(N)}, \psi_3^{(N)} \rangle e^{-\lambda_n^{(N)} t^*} \varphi_n^{(N)} + \sum_{\ell=1}^{L} \tilde{\mu}_{1,\ell} \psi_{1,\ell}^{(N)} + \sum_{\ell=1}^{L} \tilde{\mu}_{2,\ell} \psi_{2,\ell}^{(N)},$$

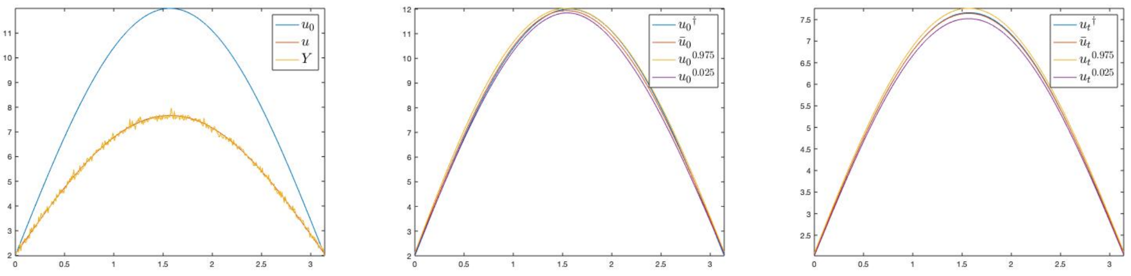
where the leftmost term in the right-hand side is defined similarly as in the one-dimensional manifold setting. The remaining finite summation terms involving $\{\varphi_n^{(N)}\}_{n=1}^N, \{\psi_{1,l}^{(N)}\}_{l=1}^L$ and $\{\psi_{2,l}^{(N)}\}_{l=1}^L$ are responsible for modelling parameter values along each boundary curve. For the implementation, the true coefficients $\{\tilde{\mu}_{1,\ell}\}_{l=1}^L$ and $\{\tilde{\mu}_{1,\ell}\}_{l=1}^L$ were obtained by regressing $u_0^{(N)} = (u_0(x_1), \cdots, u_0(x_N))$ on $\{\varphi_n^{(N)}\}_{n=1}^N, \{\psi_{1,l}^{(N)}\}_{l=1}^L$ and $\{\psi_{2,l}^{(N)}\}_{l=1}^L$. In our numerical experiments, we constructed 10 basis-like functions for each boundary of the two-dimensional semi-torus, i.e., $L = 10$, and used 20 eigenvectors of the truncated graph Laplacian.

One-dimensional Manifold

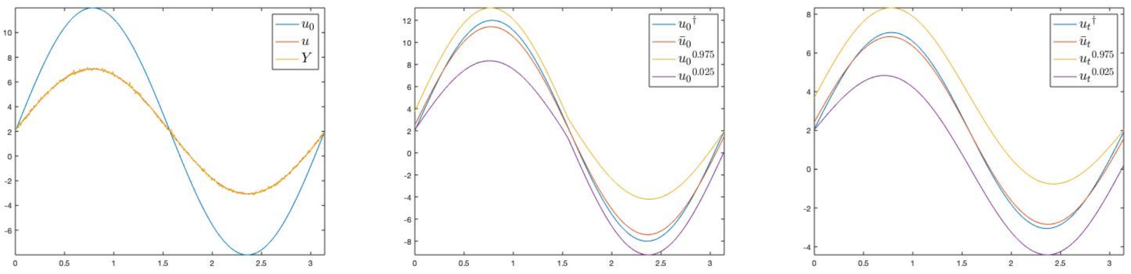
The prior construction was exactly the same as in the one-dimensional elliptic problem. And hence, the boundary values were only defined on two points i.e. values at $\alpha = 0, \pi$. We set the smoothness prior parameter to be $s = 6$, and the inverse length-scale parameter to be $\tau = 0.3$, where the choice of these values is based on empirical experiments. Furthermore, to attain an MCMC acceptance rate of roughly 50 percent, we used $\zeta = 0.005$ for the graph pCN algorithm. A total of 20000 MCMC iterations with initial 10000 burn-in iterations were run to obtain our results. For the heat equation, the degree of ill-posedness of the inverse problem is closely related to the time $t^* > 0$ at which we observe the data. The smaller the time $t^* > 0$ is, the easier the inversion.

We considered three different choices of initial heat functions. Starting from the left-most column of Figure 6.7, plots in each column respectively represent initial heat function with observed data, true initial function with the posterior mean/2.5th and 97.5th percentile of MCMC samples, and the true solution with the solutions corresponding to the posterior mean, 2.5th and 97.5 percentile of MCMC samples. Specifically, the leftmost column in Figure 6.7 represents the initial heat function u_0 with its corresponding noise-free observation u and noisy observation y. The middle column represents the true initial function $u_0^\dagger$ with the posterior mean estimate $\bar{u}_0$, 2.5th percentile $u_0^{0.025}$, and 97.5th percentile $u_0^{0.975}$ of MCMC samples. The rightmost column portrays the approximate solution based on the true initial heat function at time t, denoted by $u_t^\dagger$ with approximate solutions based on the posterior mean estimate, 2.5th percentile and 97.5th percentile, respectively denoted by $\bar{u}_t$, $u_t^{0.025}$ and $u_t^{0.975}$.

(a) $u_0 = 10\sin(\alpha) + 2$ (b) Recovery of $u_0 = 10\sin(\alpha) + 2$ (c) Solution $u(t)$ at $t = 10$



(d) $u_0 = 10\sin(2\alpha) + 2$ (e) Recovery of $u_0 = 10\sin(2\alpha) + 2$ (f) Solution $u(t)$ at $t = 3$



(g) $u_0 = 10\cos(\alpha) + 2$ (h) Recovery of $u_0 = 10\cos(\alpha) + 2$ (i) Solution $u(t)$ at $t = 5$

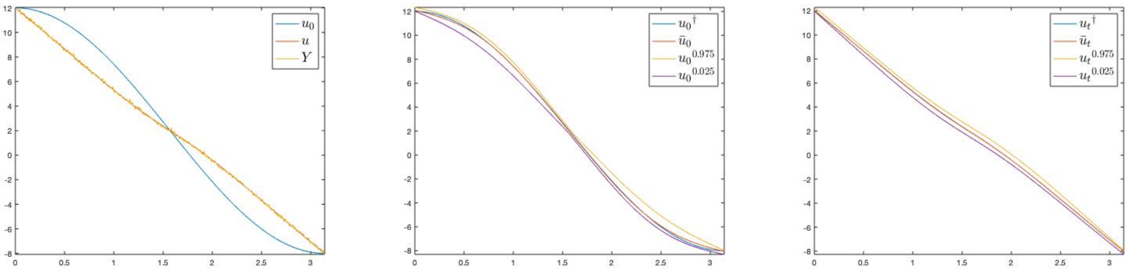


Figure 6.7: Heat inverse problem on a semi-ellipse.

Figure 6.7 shows that all of the true initial heat functions were captured inside of the 95 percent credible intervals for all three cases. Moreover, the corresponding solutions were very close to the true solution.

Two-dimensional Manifold

The practical implementation of the two-dimensional prior was analogous to that of the two-dimensional elliptic inverse problem. We set the smoothness prior parameter to be $s = 4$, and the inverse length-scale parameter to be $\tau = 0.3$ or 0.012, where the choice of these values is based on empirical experiments. Furthermore, to attain an MCMC acceptance rate of roughly 50 percent, we used $\zeta = 0.006$ for the graph pCN algorithm. A total of 100000 number of MCMC iterations with initial 50000 burn-in iterations were run to obtain these results. We considered two initial heat functions: $u_0(\alpha, \beta) = 10\sin(\alpha)\cos(2\beta)$ in Figure 6.8 and $u_0(\alpha, \beta) = 2 + \sin(\alpha)\cos(\beta)$ in Figure 6.9. For $u_0(\alpha, \beta) = 10\sin(\alpha)\cos(2\beta)$, the initial heat function values along each boundary curve we would like to recover is $10\sin(\alpha)$. Similarly for $u_0(\alpha, \beta) = 2 + \sin(\alpha)\cos(\beta)$, our goal is to recover an initial heat function with values $2 + \sin(\alpha)$ or $2 - \sin(\alpha)$ corresponding to each boundary.

Starting from the top row of Figures 6.8 and 6.9, the panels in the first row represent, from left to right, the true initial heat function, approximate solution of the heat equation based on the true heat initial function, noise-incorporated approximate solution, i.e. observation data. The second row represents from left to right, posterior mean of the MCMC samples, 2.5 percentile of MCMC samples, and 97.5 percentile of MCMC samples. In other words, the second row portrays the credible interval for the true initial heat function. The third row corresponds to the approximate solution of the heat equation based on the initial heat functions given in the second row. Finally, the last two plots in the fourth row depict the difference between the true initial function and the posterior mean of the MCMC samples, and the difference between the approximate solution based on the true initial function with the one based on the posterior mean of the MCMC samples. Figures 6.8 and Figure 6.9 show that our proposed methodology led to reasonably successful recovery of the parameter of interest.

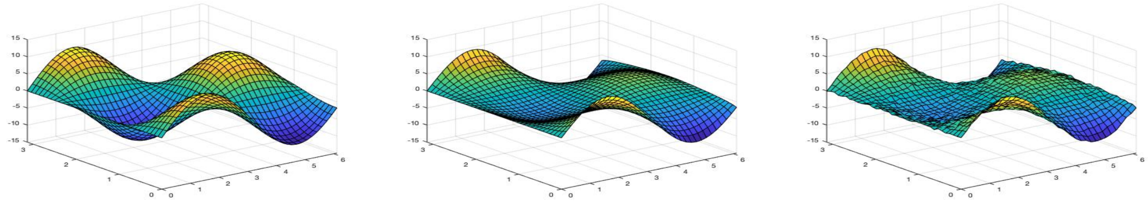
6.5 Conclusions and Open Directions

In this chapter, we developed graph-based Matérn priors for solving Bayesian inverse problems on manifolds with boundaries. Our idea is to extend the Matérn priors introduced in [Harlim et al., 2020], developed for elliptic PDEs on closed manifolds, by representing the boundary conditions via a set of functions obtained from solving Laplace equations on manifolds with appropriate Dirichlet boundary conditions. To solve PDEs on manifolds with boundaries, we employed the recently developed GPDM algorithm [Jiang and Harlim, 2023], which uses fictitious ghost points to remove the bias induced by integrating radial type kernels near the boundaries.

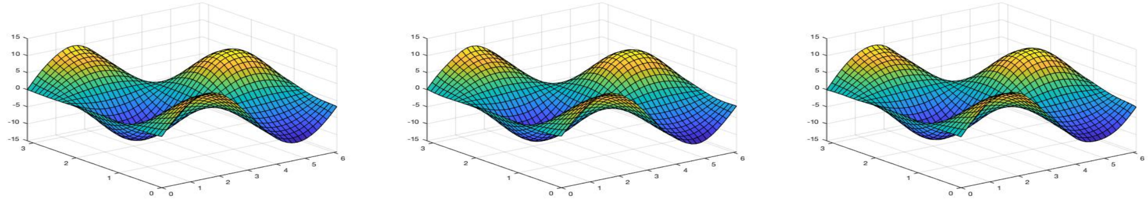
We validated this approach on two test problems. The first problem is an inversion of the diffusion coefficient of an elliptic PDE from the solution of the PDE corrupted by noise. The second problem is an inversion of the initial condition of a heat equation from noisy observation of the solution at a positive time. Based on our numerical simulations, we found positive results given the ill-posedness of the inverse problems we considered.

While the proposed method produces encouraging results, there are many open questions. First, we should point out that while the computational cost is independent of the ambient dimension, it scales exponentially as a function of intrinsic dimension (see [Jiang and Harlim, 2023] for detailed convergence rates for the forward maps). Since Bayesian inversion often requires to evaluate the forward map numerous times, it is of interest to improve on our graph-based approximations of the forward map by using computationally cheaper surrogate forward models and/or faster numerical solvers. Second, the method represents the hidden variables by a vector whose components are the function values of the variable of interest

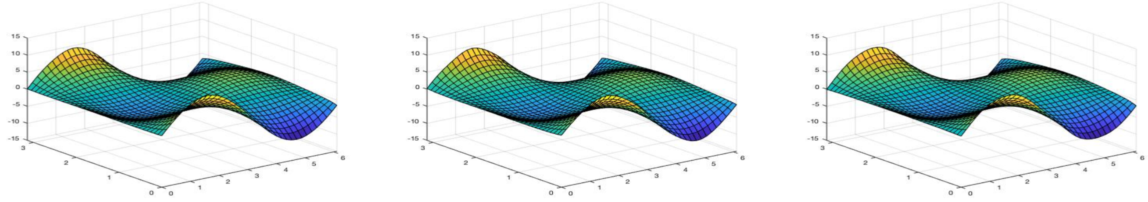
(a) True initial heat u_0 (b) PDE solution $u(t^*)$ based on u_0 (c) Observed data y



(d) Posterior mean for u_0 (e) 2.5% post. percentile for u_0 (f) 97.5% post. percentile for u_0



(g) $u_N(t^*)$ with u_0 in Figure 6.8d (h) $u_N(t^*)$ with u_0 in Figure 6.8e (i) $u_N(t^*)$ with u_0 in Figure 6.8f



(j) Error in initial condition (k) Error in PDE solution

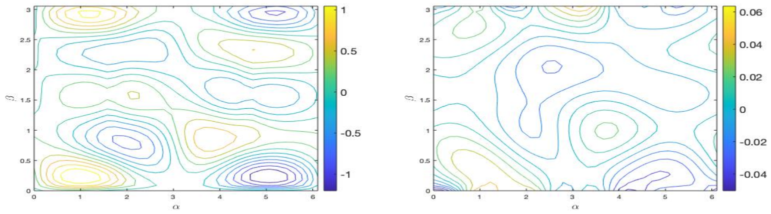
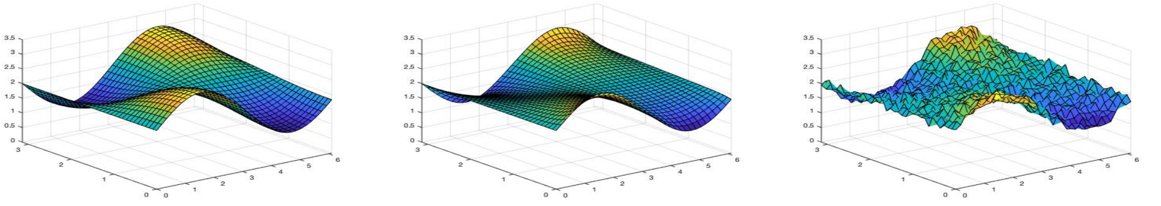
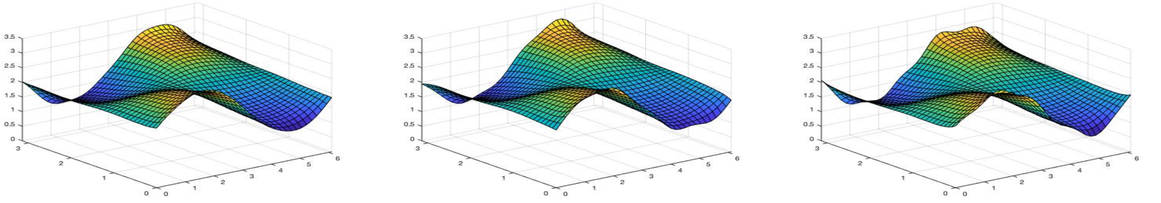
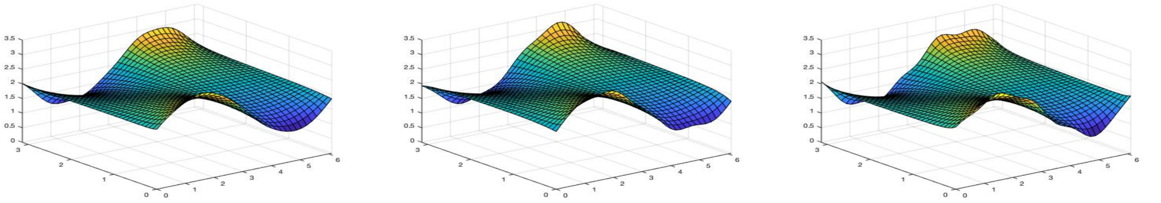


Figure 6.8: Heat inverse problem on a semi-torus: $u_0(\alpha, \beta) = 10\sin(\alpha)\cos(2\beta)$ with $\tau = 0.3$, $t^* = 5$.


(a) True initial heat u_0 (b) PDE solution $u(t^*)$ based on u_0 (c) Observed data y


(d) Posterior mean for u_0 (e) 2.5% post. percentile for u_0 (f) 97.5% post. percentile for u_0


(g) $u_N(t^*)$ with u_0 in Figure 6.9d (h) $u_N(t^*)$ with u_0 in Figure 6.9e (i) $u_N(t^*)$ with u_0 in Figure 6.9f

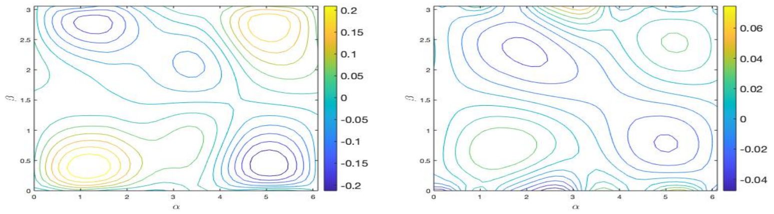

(j) Error in initial condition (k) Error in PDE solution

Figure 6.9: Heat inverse problem on a semi-torus: $u_0(\alpha, \beta) = 2 + \sin(\alpha)\cos(\beta)$ with $\tau = 0.012$, $t^* = 5$.

(e.g., diffusion coefficients) on the given point clouds. How to extend this to other points on the domain is of practical interest. Beyond these practical considerations, it is also of interest to understand the theoretical aspect of such an approach, especially the effects of boundaries, extending the theoretical convergence result for closed manifolds in [Harlim et al., 2020].

REFERENCES

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

S. Agapiou, J. M. Bardsley, O. Papaspiliopoulos, and A. M. Stuart. Analysis of the Gibbs sampler for hierarchical inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 2(1):511–544, 2014.

S. Agapiou, O. Papaspiliopoulos, D. Sanz-Alonso, and A. M. Stuart. Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, 32(3):405–431, 2017.

S. Agrawal, H. Kim, D. Sanz-Alonso, and A. Strang. A variational inference approach to inverse problems with gamma hyperpriors. *SIAM/ASA Journal on Uncertainty Quantification*, 10(4):1533–1559, 2022.

O. Al-Ghattas and D. Sanz-Alonso. Non-asymptotic analysis of ensemble kalman updates: effective dimension and localization. *Information and Inference: A Journal of the IMA*, 13(1):iaad043, 2024.

A. Armagan. Variational bridge regression. In *Artificial Intelligence and Statistics*, pages 17–24. PMLR, 2009.

D. Armbruster, M. Herty, and G. Visconti. A stabilization of a continuous limit of the ensemble Kalman inversion. *SIAM Journal on Numerical Analysis*, 60(3):1494–1515, 2022.

S. Arridge, P. Maass, O. Öktem, and C. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

S. Bansal, R. Calandra, T. Xiao, S. Levine, and C. J. Tomlin. Goal-driven dynamics learning via Bayesian optimization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 5168–5173. IEEE, 2017.

R. Baptista and M. Poloczek. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471. PMLR, 2018.

D. Barber and W. Wiegerinck. Tractable variational structures for approximating graphical models. *Advances in Neural Information Processing Systems*, 11, 1998.

J. F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*, volume 30. Springer Science & Business Media, 2013.

J. V. Beck, B. Blackwell, and C. R. St Clair Jr. *Inverse heat conduction: Ill-posed problems.* James Beck, 1985.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006.

B. M. Bell and F. W. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, 1993.

T. Berry and J. Harlim. Variable bandwidth diffusion kernels. *Applied and Computational Harmonic Analysis*, 40(1):68–96, 2016.

M. Bertalmıo, L-T Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174(2):759–780, 2001.

A. L. Bertozzi, X. Luo, A. M. Stuart, and K. C. Zygalakis. Uncertainty quantification in graph-based classification of high dimensional data. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):568–595, 2018.

A. Beskos, G. Roberts, A. Stuart, and J. Voss. MCMC methods for diffusion bridges. *Stochastics and Dynamics*, 8(03):319–350, 2008.

J. C. Bezdek, R. J. Hathaway, R. E. Howard, C. A. Wilson, and M. P. Windham. Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications*, 54(3):471–477, 1987.

D. Bigoni, Y. Chen, N. Garcia Trillos, Y. Marzouk, and D. Sanz-Alonso. Data-driven forward discretizations for bayesian inversion. *Inverse Problems*, 36(10):105008, 2020.

C. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

D. Blömker, C. Schillings, and P. Wacker. A strongly convergent numerical scheme from ensemble Kalman inversion. *SIAM Journal on Numerical Analysis*, 56(4):2537–2562, 2018.

D. Blömker, C. Schillings, P. Wacker, and S. Weissmann. Well posedness and convergence analysis of the ensemble Kalman inversion. *Inverse Problems*, 35(8):085007, 2019.

V. I. Bogachev. *Gaussian Measures.* Number 62. American Mathematical Soc., 1998.

I. Bogunovic and A. Krause. Misspecified Gaussian process bandit optimization. *Advances in Neural Information Processing Systems*, 34:3004–3015, 2021.

D. Bolin. Spatial Matérn fields driven by non-Gaussian noise. *Scandinavian Journal of Statistics*, 41(3):557–579, 2014.

D. Bolin and K. Kirchner. The rational SPDE approach for Gaussian random fields with general smoothness. *Journal of Computational and Graphical Statistics*, pages 1–27, 2019.

D. Bolin and K. Kirchner. The rational SPDE approach for Gaussian random fields with general smoothness. *Journal of Computational and Graphical Statistics*, 29(2):274–285, 2020.

D. Bolin, K. Kirchner, and M. Kovács. Weak convergence of Galerkin approximations for fractional elliptic stochastic PDEs with spatial white noise. *BIT Numerical Mathematics*, 58(4):881–906, 2018.

D. Bolin, K. Kirchner, and M. Kovács. Numerical solution of fractional elliptic stochastic PDEs with spatial white noise. *IMA Journal of Numerical Analysis*, 40(2):1051–1073, 2020.

V. Borovitskiy, A. Terenin, P. Mostowsky, et al. Matérn Gaussian processes on Riemannian manifolds. *Advances in Neural Information Processing Systems*, 33:12426–12437, 2020.

V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. Deisenroth, and N. Durrande. Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pages 2593–2601. PMLR, 2021.

N. Boumal. *An Introduction to Optimization on Smooth Manifolds*. To appear in Cambridge University Press, Available online, 2020.

P. J. Brown and J. E. Griffin. Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis*, 2010.

S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media, 2011.

A. D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12(10), 2011.

D. Burago, S. Ivanov, and Y. Kurylev. A graph discretization of the Laplace–Beltrami operator. *Journal of Spectral Theory*, 4(4):675–714, 2015.

C. J. Burges and B. Schölkopf. Improving the accuracy and speed of support vector machines. *Advances in Neural Information Processing Systems*, 9, 1996.

D. Calvetti and E. Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, volume 2. Springer Science & Business Media, 2007.

D. Calvetti, J. P Kaipio, and E. Somersalo. Aristotelian prior boundary conditions. *International Journal of Mathematics and Computer Science*, 1(1):63–81, 2006.

D. Calvetti, A. Pascarella, F. Pitolli, E. Somersalo, and B. Vantaggi. A hierarchical Krylov–Bayes iterative inverse solver for MEG with physiological preconditioning. *Inverse Problems*, 31(12):125005, 2015.

D. Calvetti, F. Pitolli, E. Somersalo, and B. Vantaggi. Bayes meets Krylov: Statistically inspired preconditioners for CGLS. *SIAM Review*, 60(2):429–461, 2018.

D. Calvetti, A. Pascarella, F. Pitolli, E. Somersalo, and B. Vantaggi. Brain activity mapping from MEG data via a hierarchical Bayesian algorithm with automatic depth weighting. *Brain topography*, 32(3):363–393, 2019a.

D. Calvetti, E. Somersalo, and A. Strang. Hierachical Bayesian models and sparsity: L^2-magic. *Inverse Problems*, 35(3):035003, 2019b.

D. Calvetti, M. Pragliola, and E. Somersalo. Sparsity promoting hybrid solvers for hierarchical Bayesian inverse problems. *SIAM Journal on Scientific Computing*, 42(6):A3761–A3784, 2020a.

D. Calvetti, M. Pragliola, E. Somersalo, and A. Strang. Sparse reconstructions from few noisy data: analysis of hierarchical Bayesian models with generalized gamma hyperpriors. *Inverse Problems*, 36(2):025010, 2020b.

E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE signal processing magazine*, 25(2):21–30, 2008.

Y. Canzani. Analysis on manifolds via the Laplacian. *Lecture Notes available at: http://www. math. harvard. edu/canzani/docs/Laplacian. pdf*, 2013.

C. Cartis, E. Massart, and A. Otemissov. Bound-constrained global optimization of functions with low effective dimensionality using multiple random embeddings. *Mathematical Programming*, 198(1):997–1058, 2023a.

C. Cartis, E. Massart, and A. Otemissov. Global optimization using random embeddings. *Mathematical Programming*, 200(2):781–829, 2023b.

C. M. Carvalho, N. G Polson, and J. G. Scott. Handling sparsity via the horseshoe. In *Artificial Intelligence and Statistics*, pages 73–80. PMLR, 2009.

C. M. Carvalho, N. G. Polson, and J. G. Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

N. K. Chada and X. Tong. Convergence acceleration of ensemble Kalman inversion in nonlinear settings. *Mathematics of Computation*, 91(335):1247–1280, 2022.

N. K. Chada, M. A. Iglesias, L. Roininen, and A. M. Stuart. Parameterizations for ensemble Kalman inversion. *Inverse Problems*, 34(5):055009, 2018.

N. K. Chada, A. M. Stuart, and X. T. Tong. Tikhonov regularization within ensemble Kalman inversion. *SIAM Journal on Numerical Analysis*, 58(2):1263–1294, 2020.

N. K. Chada, Y. Chen, and D. Sanz-Alonso. Iterative ensemble Kalman methods: A unified perspective with some new variants. *Foundations of Data Science*, 3(3):331–369, 2021.

Y. Chen, D. Sanz-Alonso, and R. Willett. Autodifferentiable ensemble Kalman filters. *SIAM Journal on Mathematics of Data Science*, 4(2):801–833, 2022.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

S. R. Chowdhury and A. Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pages 844–853. PMLR, 2017.

S. R. Chowdhury and A. Gopalan. Bayesian optimization under heavy-tailed payoffs. *Advances in Neural Information Processing Systems*, 32, 2019.

D. L. Clark Jr, H. Bae, K. Gobal, and R. Penmetsa. Engineering design exploration using locally optimized covariance kriging. *AIAA Journal*, 54(10):3160–3175, 2016.

E. Cleary, A. Garbuno-Inigo, S. Lan, T. Schneider, and A. M. Stuart. Calibrate, emulate, sample. *Journal of Computational Physics*, 424:109716, 2021.

R. R. Coifman and S. Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

B. Colbois, A. El Soufi, and A. Savo. Eigenvalues of the Laplacian on a compact manifold with density. *Communications in Analysis and Geometry*, 23(3):639–670, 2015.

P. D. Congdon. *Applied Bayesian Hierarchical Methods*. Chapman and Hall/CRC, 2010.

S. Cotter, M. Dashti, and A. M. Stuart. MCMC methods for functions: modifying old algorithms to make them faster. *SIAM Journal on Numerical Analysis*, 48(1):322–345, 2010.

K. Crane. Keenan's 3d model repository. URL `http://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository`.

P. Damlen, J. Wakefield, and S. Walker. Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(2):331–344, 1999.

Y. Daon and G. Stadler. Mitigating the influence of the boundary on PDE-based covariance operators. *Inverse Problems and Imaging*, 12(5):1083–1102, 2018.

M. Dashti and A. M. Stuart. Bayesian approach to inverse problems. *Handbook of Uncertainty Quantification*, pages 311–428, 2017.

I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 63(1):1–38, 2010.

N. de Freitas, P. Højen-Sørensen, M. I. Jordan, and S. Russell. Variational MCMC. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 120–127, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

N. De Freitas, A. J. Smola, and M. Zoghi. Exponential regret bounds for Gaussian process bandits with deterministic observations. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 955–962, 2012.

G. P. Dehaene. Computing the quality of the Laplace approximation. *Neural Information Processing Systems*, 2017.

A. Deshwal, S. Belakaria, and J. R. Doppa. Bayesian optimization over hybrid spaces. In *International Conference on Machine Learning*, pages 2632–2643. PMLR, 2021.

Z. Ding and Q. Li. Ensemble Kalman sampler: Mean-field limit and convergence analysis. *SIAM Journal on Mathematical Analysis*, 53(2):1546–1578, 2021.

H. Donnelly. Eigenfunctions of the Laplacian on compact Riemannian manifolds. *Asian Journal of Mathematics*, 10(1):115–126, 2006.

D. B. Dunson, H.-T. Wu, and N. Wu. Graph based Gaussian processes on restricted domains. *Journal of the Royal Statistical Society Series B*, 84(2):414–439, 2022.

G. Dziuk and C. M. Elliott. Finite element methods for surface PDEs. *Acta Numerica*, 22: 289–396, 2013.

A. Edelman, T. A Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

C. M. Elliott and B. Stinner. Modeling and computation of two phase geometric biomembranes using surface finite elements. *Journal of Computational Physics*, 229(18):6585–6612, 2010.

G. Evans. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99 (c5):10143–10162, 1995.

G. Evans and P. Van Leeuwen. Assimilation of Geosat altimeter data for the Agulhas current using the ensemble Kalman filter with a quasigeostrophic model. *Monthly Weather Review*, 124(1):85–96, 1996.

S. N. Evans and P. B. Stark. Inverse problems as statistics. *Inverse problems*, 18(4):R55, 2002.

G. Evensen. *Data Assimilation: the Ensemble Kalman Filter*, volume 2. Springer, 2009.

A. Feragen, F. Lauze, and S. Hauberg. Geodesic exponential kernels: When curvature and linearity conflict. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3032–3042, 2015.

R. A. Fisher. Dispersion on a sphere. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 217(1130):295–305, 1953.

S. Foucart, H. Rauhut, S. Foucart, and H. Rauhut. *An invitation to compressive sensing.* Springer, 2013.

J. Franklin. Well-posed stochastic extensions of ill-posed linear problems. *Journal of mathematical analysis and applications*, 31(3):682–716, 1970.

P. I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

E. J. Fuselier and G. B. Wright. A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. *Journal of Scientific Computing*, 56(3):535–565, 2013.

T. Gao, S. Z. Kovalsky, and I. Daubechies. Gaussian process landmarking on manifolds. *SIAM Journal on Mathematics of Data Science*, 1(1):208–236, 2019.

N. Garcia Trillos and D. Sanz-Alonso. The Bayesian formulation and well-posedness of fractional elliptic inverse problems. *Inverse Problems*, 33(6):065006, 2017.

N. Garcia Trillos and D. Sanz-Alonso. Continuum limits of posteriors in graph Bayesian inverse problems. *SIAM Journal on Mathematical Analysis*, 50(4):4020–4040, 2018.

N. Garcia Trillos and D. Sanz-Alonso. The Bayesian update: variational formulations and gradient flows. *Bayesian Analysis*, 15(1):29–56, 2020.

N. Garcia Trillos, D. Sanz-Alonso, and R. Yang. Local regularization of noisy point clouds: Improved global geometric estimates and data analysis. *Journal of Machine Learning Research*, 20(136):1–37, 2019. URL `http://jmlr.org/papers/v20/19-261.html`.

N. Garcia Trillos, M. Gerlach, M. Hein, and D. Slepčev. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs toward the Laplace–Beltrami operator. *Foundations of Computational Mathematics*, 20(4):827–887, 2020a.

N. Garcia Trillos, Z. Kaplan, T. Samakhoana, and D. Sanz-Alonso. On the consistency of graph-based Bayesian semi-supervised learning and the scalability of sampling algorithms. *Journal of Machine Learning Research*, 21(28):1–47, 2020b.

N. Garcia Trillos, D. Sanz-Alonso, and R. Yang. Mathematical foundations of graph-based Bayesian semi-supervised learning. *arXiv preprint arXiv:2207.01093*, 2022.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis.* Chapman and Hall/CRC, 1995.

R. J. Giordano, T. Broderick, and M. I. Jordan. Linear response methods for accurate covariance estimates from mean field variational bayes. *Advances in Neural Information Processing Systems*, 28, 2015.

T. Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19 (4):1327–1349, 2013.

I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm. *IEEE Transactions on signal processing*, 45(3):600–616, 1997.

R. B. Gramacy. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences.* Chapman and Hall/CRC, 2020.

P. J. Green. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society: Series B (Methodological)*, 46(2):149–170, 1984.

C. W. Groetsch. *Inverse Problems in the Mathematical Sciences*, volume 52. Springer, 1993.

Y. Gu and D. S. Oliver. An iterative ensemble Kalman filter for multiphase fluid flow data assimilation. *SPE Journal*, 12(04):438–446, 2007.

M. Hanke. A regularizing Levenberg-Marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse Problems*, 13(1):79–95, 1997.

J. Harlim, D. Sanz-Alonso, and R. Yang. Kernel methods for bayesian elliptic inverse problems on manifolds. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1414–1445, 2020.

J. Harlim, S. W. Jiang, H. Kim, and D. Sanz-Alonso. Graph-based prior and forward models for inverse problems on manifolds with boundaries. *Inverse Problems*, 38(3):035006, 2022.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations.* CRC press, 2015.

M. Hein and J.-Y. Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In *Proceedings of the 22nd international conference on Machine learning*, pages 289–296. ACM, 2005.

T. Helin, A. M. Stuart, A. L. Teckentrup, and K. Zygalakis. Introduction to Gaussian process regression in Bayesian inverse problems, with new results on experimental design for weighted error measures. *arXiv preprint arXiv:2302.04518*, 2023.

M. Herty and G. Visconti. Kinetic methods for inverse problems. *Kinetic & Related Models*, 12(5):1109, 2019.

J. Hu, X. Liu, Z.-W. Wen, and Y.-X. Yuan. A brief introduction to manifold optimization. *Journal of the Operations Research Society of China*, 8(2):199–248, 2020.

D. Z. Huang, J. Huang, S. Reich, and A. M. Stuart. Efficient derivative-free bayesian inference for large-scale inverse problems. *Inverse Problems*, 38(12):125006, 2022.

M. A. Iglesias. A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems. *Inverse Problems*, 32(2):025002, 2016.

M. A. Iglesias and Y. Yang. Adaptive regularisation for ensemble Kalman inversion. *Inverse Problems*, 37(2):025008, 2021.

M. A. Iglesias, K. J. H. Law, and A. M. Stuart. Ensemble Kalman methods for inverse problems. *Inverse Problems*, 29(4):045001, 2014.

M. A. Iglesias, Y. Lu, and A. M. Stuart. A bayesian level set method for geometric inverse problems. *Interfaces and free boundaries*, 18(2):181–217, 2016.

N. Jaquier, L. Rozo, S. Calinon, and M. Bürger. Bayesian optimization meets Riemannian manifolds in robot learning. In *Conference on Robot Learning*, pages 233–246. PMLR, 2020.

N. Jaquier, V. Borovitskiy, A. Smolensky, A. Terenin, T. Asfour, and L. Rozo. Geometry-aware Bayesian optimization in robotics using Riemannian Matérn kernels. In *Conference on Robot Learning*, pages 794–805. PMLR, 2022.

S. W. Jiang and J. Harlim. Ghost point diffusion maps for solving elliptic pdes on manifolds with classical boundary conditions. *Communications on Pure and Applied Mathematics*, 76(2):337–405, 2023.

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455, 1998.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999.

S. I. Kabanikhin. Definitions and examples of inverse and ill-posed problems. *Journal of Inverse and Ill-posed Problems*, 2008.

J. Kaipo and E. Somersalo. Statistical and Computational Inverse Problems. *Springer Science & Business Media*, 160, 2006.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos. Parallelised Bayesian optimisation via Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, pages 133–142. PMLR, 2018.

U. Khristenko, L. Scarabosio, P. Swierczynski, E. Ullmann, and B. Wohlmuth. Analysis of boundary effects on PDE-based sampling of Whittle–Matérn Random Fields. *SIAM/ASA Journal on Uncertainty Quantification*, 7(3):948–974, 2019.

H. Kim and D. Sanz-Alonso. Bayesian optimization with noise-free observations: Improved regret bounds via random exploration. *Under Review*, 2024.

H. Kim, D. Sanz-Alonso, and A. Strang. Hierarchical ensemble kalman methods with sparsity-promoting generalized gamma hyperpriors. *Foundation of Data Science*, 2023.

H. Kim, D. Sanz-Alonso, and R. Yang. Optimization on manifolds via graph gaussian processes. *SIAM Journal on Mathematics of Data Science*, 2024.

J. Kirschner, M. Mutny, N. Hiller, R. Ischebeck, and A. Krause. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *International Conference on Machine Learning*, pages 3429–3438. PMLR, 2019.

V. Y. Korolev and A. I. Zeifman. Generalized negative binomial distributions as mixed geometric laws and related limit theorems. *Lithuanian Mathematical Journal*, 59(3):366–388, 2019.

N. B. Kovachki and A. M. Stuart. Ensemble Kalman inversion: a derivative-free technique for machine learning tasks. *Inverse Problems*, 35(9):095005, 2019.

S. Lan, S. Li, and M. Pasha. Bayesian spatiotemporal modeling for inverse problems. *Statistics and Computing*, 33(4):89, 2023.

A. Lang, J. Potthoff, M. Schlather, and D. Schwab. Continuity of random fields on Riemannian manifolds. *Communications on Stochastic Analysis*, 10(2):4, 2016.

K. J. H. Law and V. Zankin. Sparse online variational Bayesian regression. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1070–1100, 2022.

Y. Lee. ℓ_p regularization for ensemble Kalman inversion. *SIAM Journal on Scientific Computing*, 43(5):A3417–A3437, 2021.

A. J. Lemonte and G. M. Cordeiro. The exponentiated generalized inverse Gaussian distribution. *Statistics & Probability Letters*, 81(4):506–517, 2011.

R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-state and Time-Dependent Problems*, volume 98. Siam, 2007.

D. Li, W. Tang, and S. Banerjee. Inference for gaussian processes with matérn covariogram on compact riemannian manifolds. *Journal of Machine Learning Research*, 24(101):1–26, 2023.

G. Li and A. C. Reynolds. An iterative ensemble Kalman filter for data assimilation. In *SPE annual technical conference and exhibition*. Society of Petroleum Engineers, 2007.

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(4):423–498, 2011.

J. S. Liu. *Monte Carlo Strategies in Scientific Computing.* Springer Science & Business Media, 2008.

E. N. Lorenz. Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20(2): 130–141, 1963.

C. Louizos, M. Welling, and D. P. Kingma. Learning sparse neural networks through ℓ_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

P. Luong, S. Gupta, D. Nguyen, S. Rana, and S. Venkatesh. Bayesian optimization with discrete variables. In *Australasian Joint Conference on Artificial Intelligence*, pages 473–484. Springer, 2019.

Y. Lyu, Y. Yuan, and I. W. Tsang. Efficient batch black-box optimization with deterministic regret bounds. *arXiv preprint arXiv:1905.10041*, 2019.

C. B. Macdonald and S. J. Ruuth. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing*, 31(6):4330–4350, 2010.

L. Maestrini, R. G. Aykroyd, and M. P. Wand. A variational inference framework for inverse problems. *arXiv preprint arXiv:2103.05909*, 2021.

Y. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: an introduction. *Handbook of Uncertainty Quantification*, pages 1–41, 2016.

B. Matérn. *Spatial Variation*, volume 36. Second Edition, Springer Science & Business Media, 2013.

F. Mémoli, G. Sapiro, and P. Thompson. Implicit brain imaging. *NeuroImage*, 23:S179–S188, 2004.

J. Mockus. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117, 1998.

V. A. Morozov. On the solution of functional equations by the method of regularization. In *Doklady Akademii Nauk*, volume 167, pages 510–512. Russian Academy of Sciences, 1966.

C. Naesseth, S. Linderman, R. Ranganath, and D. Blei. Variational sequential Monte Carlo. In *International conference on artificial intelligence and statistics*, pages 968–977. PMLR, 2018.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

C. J. Oates, J. Cockayne, F.-X. Briol, and M. Girolami. Convergence rates for a class of estimators based on Stein's method. *Bernoulli*, 25(2):1141–1159, 2019.

F. O'Sullivan. A statistical perspective on ill-posed inverse problems. *Statistical science*, pages 502–518, 1986.

R. K. Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33 (3):291–297, 1997.

J. Pan, Z. Hu, Z. Su, and M.-H. Yang. Deblurring text images via ℓ_0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908, 2014.

T. Park and G. Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

J. W. Peoples and J. Harlim. Spectral convergence of symmetrized graph Laplacian on manifolds with boundary. *arXiv preprint arXiv:2110.06988*, 2021.

J. Pidstrigach and S. Reich. Affine-invariant ensemble transform methods for logistic regression. *Foundations of Computational Mathematics*, pages 1–34, 2022.

N. G. Polson, J. G. Scott, and J. Windle. The bayesian bridge. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4):713–733, 2014.

T. Pourmohamad. Compmodels: Pseudo computer models for optimization. *R package version 0.2. 0*, 2020.

T. Pourmohamad and H. KH. Lee. *Bayesian Optimization with Application to Computer Experiments*. Springer, 2021.

M. Rauter and Ž. Tuković. A finite area scheme for shallow granular flows on three-dimensional surfaces. *Computers & Fluids*, 166:184–199, 2018.

A. C. Reynolds, M. Zafari, and G. Li. Iterative forms of the ensemble Kalman filter. In *ECMOR X-10th European conference on the mathematics of oil recovery*, pages cp–23. European Association of Geoscientists & Engineers, 2006.

H. E. Robbins. An empirical Bayes approach to statistics. In *Breakthroughs in Statistics*, pages 388–394. Springer, 1992.

G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2):291–317, 1997.

V. Ročková. Bayesian estimation of sparse signals with a continuous spike-and-slab prior. *The Annals of Statistics*, 2018.

V. Ročková and E. I. George. The spike-and-slab lasso. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

L. Roininen, M. Girolami, S. Lasanen, and M. Markkanen. Hyperpriors for Matérn fields with applications in Bayesian inversion. *Inverse Problems & Imaging*, 13(1):1–29, 2019.

D. Russo and B. Van Roy. Learning to optimize via information-directed sampling. *Advances in Neural Information Processing Systems*, 27, 2014.

S. J. Ruuth and B. Merriman. A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3):1943–1961, 2008.

S. Salgia, S. Vakili, and Q. Zhao. Random exploration in Bayesian Optimization: order-optimal regret and computational efficiency. *arXiv preprint arXiv:2310.15351*, 2023.

D. Sanz-Alonso and R. Yang. The SPDE approach to Matérn fields: Graph representations. *Statistical Science*, 37(4):519–540, 2022a.

D. Sanz-Alonso and R. Yang. Unlabeled data help in graph-based semi-supervised learning: a Bayesian nonparametrics perspective. *Journal of Machine Learning Research*, 23(97): 1–28, 2022b.

D. Sanz-Alonso and R. Yang. Finite element representations of Gaussian processes: Balancing numerical and statistical accuracy. *To appear in SIAM/ASA Journal on Uncertainty Quantification*, 2022c.

D. Sanz-Alonso, A. Stuart, and A. Taeb. *Inverse Problems and Data Assimilation*, volume 107. Cambridge University Press, 2023.

L. Saul and M. Jordan. Exploiting tractable substructures in intractable networks. *Advances in neural information processing systems*, 8, 1995.

C. Schillings and A. M. Stuart. Analysis of the ensemble Kalman filter for inverse problems. *SIAM Journal on Numerical Analysis*, 55(3):1264–1290, 2017.

C. Schillings, B. Sprungk, and P. Wacker. On the convergence of the Laplace approximation and noise-level-robustness of Laplace-based Monte Carlo methods for Bayesian inverse problems. *Numerische Mathematik*, 145(4):915–971, 2020.

T. Schneider, S. Lan, A. M. Stuart, and J. Teixeira. Earth system modeling 2.0: A blueprint for models that learn from observations and targeted high-resolution simulations. *Geophysical Research Letters*, 44(24):12–396, 2017.

T. Schneider, A. M. Stuart, and J. Wu. Ensemble kalman inversion for sparse learning of dynamical systems from time-averaged data. *Journal of Computational Physics*, 470: 111559, 2022.

B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

B. Shustin, H. Avron, and B. Sober. Manifold free riemannian optimization. *arXiv preprint arXiv:2209.03269*, 2022.

K. Siebertz, T. Hochkirchen, and D. van Bebber. *Statistische Versuchsplanung.* Springer, 2010.

A. Singer and R. Yang. Alignment of density maps in Wasserstein distance. *arXiv preprint arXiv:2305.12310*, 2023.

E. Somersalo, M. Cheney, and D. Isaacson. Existence and uniqueness for electrode models for electric current computed tomography. *SIAM Journal on Applied Mathematics*, 52(4): 1023–1040, 1992.

N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1015–1022, 2010.

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging.* Springer Science & Business Media, 2012.

A. M. Stuart. Inverse problems: a bayesian perspective. *Acta numerica*, 19:451–559, 2010.

A. M. Stuart and A. Teckentrup. Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. *Mathematics of Computation*, 87(310):721–753, 2018.

K. Swersky, Y. Rubanova, D. Dohan, and K. Murphy. Amortized Bayesian optimization over discrete spaces. In *Conference on Uncertainty in Artificial Intelligence*, pages 769–778. PMLR, 2020.

A. Tarantola. Inverse Problem Theory and Methods for Model Parameter Estimation. *SIAM*, 2015.

A. L. Teckentrup. Convergence of Gaussian process regression with estimated hyperparameters and applications in Bayesian inverse problems. *SIAM/ASA Journal on Uncertainty Quantification*, 8(4):1310–1337, 2020.

E. H. Thiede, D. Giannakis, A. R. Dinner, and J. Weare. Galerkin approximation of dynamical quantities using trajectory data. *The Journal of Chemical Physics*, 150(24):244111, 2019.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

F. Tonolini, J. Radford, A. Turpin, D. Faccio, and R. Murray-Smith. Variational inference for computational imaging inverse problems. *Journal of Machine Learning Research*, 21 (179):1–46, 2020.

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494, 2001.

R. Tuo and W. Wang. Kriging prediction with isotropic Matérn correlations: Robustness and experimental designs. *Journal of Machine Learning Research*, 21(1):7604–7641, 2020.

S. Ungarala. On the iterated forms of Kalman filters using statistical linearization. *Journal of Process Control*, 22(5):935–943, 2012.

S. Vakili. Open problem: Regret bounds for noise-free kernel-based bandits. In *Conference on Learning Theory*, pages 5624–5629. PMLR, 2022.

S. Vakili, K. Khezeli, and V. Picheny. On information gain and regret bounds in Gaussian process bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 82–90. PMLR, 2021.

E. G. Virga. *Variational Theories for Liquid Crystals.* Chapman and Hall/CRC, 2018.

U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference.* Now Publishers Inc, 2008.

W. Wang, R. Tuo, and J. CF. Wu. On prediction properties of kriging: Uniform error bounds and robustness. *Journal of the American Statistical Association*, 115(530):920–930, 2020.

Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55: 361–387, 2016.

H. Wendland. *Scattered Data Approximation*, volume 17. Cambridge University Press, 2004.

T. Wenzel, G. Santin, and B. Haasdonk. A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability and uniform point distribution. *Journal of Approximation Theory*, 262:105508, 2021.

P. Whittle. Stochastic-processes in several dimensions. *Bulletin of the International Statistical Institute*, 40(2):974–994, 1963.

A. Wiens, D. Nychka, and W. Kleibe. Modeling spatial data using local likelihood estimation and a mat\'ern to sar translation. *arXiv preprint arXiv:2002.01124*, 2020.

C. K. Wikle and L. M. Berliner. A Bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):1–16, 2007.

C. K. I. Williams and C. E. Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT press Cambridge, MA, 2006.

Z. Wu and R. Schaback. Local error estimates for radial basis function interpolation of scattered data. *IMA Journal of Numerical Analysis*, 13(1):13–27, 1993.

B. Xu. Asymptotic behavior of L^2-normalized eigenfunctions of the Laplace-Beltrami operator on a closed Riemannian manifold. *Harmonic Analysis and its Applications*, pages 99–117, 2006.

Q. Yan, S. W. Jiang, and J. Harlim. Kernel-based methods for solving time-dependent advection-diffusion equations on manifolds. *Journal of Scientific Computing*, 94(1):5, 2023.

K. Ye, K. S.-W. Wong, and L.-H. Lim. Optimization on flag manifolds. *Mathematical Programming*, 194(1):621–660, 2022.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2005.

H. Zhang. Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association*, 99(465):250–261, 2004.

H. Zhang and S. X. Chen. Concentration inequalities for statistical inference. *arXiv preprint arXiv:2011.02258*, 2020.

D. A. Zimmerman, G. De Marsily, C. A. Gotway, M. G. Marietta, C. L. Axness, R. L. Beauheim, R. L. Bras, J. Carrera, G. Dagan, P. B. Davies, et al. A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow. *Water Resources Research*, 34(6):1373–1413, 1998.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.