Bayesian Spatiotemporal Modeling and Uncertainty Quantification

by

Shuyi Li

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved September 2023 by the
Graduate Supervisory Committee:

Shiwei Lan, Chair
Cheng Dan
Paul Richard Hahn
Robert McCulloch
Hedibert Lopes

ARIZONA STATE UNIVERSITY

December 2023

ABSTRACT

Uncertainty Quantification (UQ) is crucial in assessing the reliability of predictive models that make decisions for human experts in a data-rich world. The Bayesian approach to UQ for inverse problems has gained popularity. However, addressing UQ in high-dimensional inverse problems is challenging due to the intensity and inefficiency of Markov Chain Monte Carlo (MCMC) based Bayesian inference methods. Consequently, the first primary focus of this thesis is enhancing efficiency and scalability for UQ in inverse problems.

On the other hand, the omnipresence of spatiotemporal data, particularly in areas like traffic analysis, underscores the need for effectively addressing inverse problems with spatiotemporal observations. Conventional solutions often overlook spatial or temporal correlations, resulting in underutilization of spatiotemporal interactions for parameter learning. Appropriately modeling spatiotemporal observations in inverse problems thus forms another pivotal research avenue.

In terms of UQ methodologies, the calibration-emulation-sampling (CES) scheme has emerged as effective for large-dimensional problems. I introduce a novel CES approach by employing deep neural network (DNN) models during the emulation and sampling phase. This approach not only enhances computational efficiency but also diminishes sensitivity to training set variations. The newly devised "Dimension-Reduced Emulative Autoencoder Monte Carlo (DREAM)" algorithm scales Bayesian UQ up to thousands of dimensions in physics-constrained inverse problems. The algorithm's effectiveness is exemplified through elliptic and advection-diffusion inverse problems.

In the realm of spatiotemporal modeling, I propose to use Spatiotemporal Gaussian processes (STGP) in likelihood modeling and Spatiotemporal Besov processes (STBP)

in prior modeling separately. These approaches highlight the efficacy of incorporating spatial and temporal information for enhanced parameter estimation and UQ. Additionally, the superiority of STGP is demonstrated compared to static and time-averaged methods in time-dependent advection-diffusion partial differential equation (PDE) and three chaotic ordinary differential equations (ODE). Expanding upon Besov Process (BP), a method known for sparsity-promotion and edge-preservation, STBP is introduced to capture spatial data features and model temporal correlations by replacing the random coefficients in the series expansion with stochastic time functions following Q-exponential process(Q-EP). This advantage is showcased in dynamic computerized tomography (CT) reconstructions through comparison with classic STGP and a time-uncorrelated approach.

ACKNOWLEDGMENTS

I would like to express my deep appreciation to Professor Shiwei Lan for his exceptional mentorship, which has been both nurturing and transformative. Throughout my journey as a graduate student, his care, friendship, and brilliant collaboration have had a profound impact on my personal and intellectual growth. Beyond imparting knowledge in statistics, he has played an instrumental role in shaping my character and guiding my overall development. Professor Lan's unwavering support and unwavering belief in my potential have provided me with a solid foundation to explore my ideas and push the boundaries of my capabilities. His relentless pursuit of knowledge has been a constant source of inspiration and motivation for me.

Furthermore, I am deeply thankful to my thesis committee members—Dr. Paul Richard Hahn, Dr. Robert McCulloch, Dr. Cheng Dan and Dr. Hedibert Lopes—for their invaluable input, meaningful discussions, and accessibility. Thanks to them, I have developed a profound appreciation for this field of study through participation in statistical courses. These courses have provided me with the essential tools necessary to establish a strong and solid foundation in statistics. I have acquired knowledge on a multitude of research avenues from them, each of which carries significant worth.

Equally importantly, I extend my heartfelt gratitude to my family, whose consistent support and steadfast belief in me have been constant sources of strength. Their enduring presence in my life has been a guiding force, and for this, I am truly thankful. Equally deserving of my deepest appreciation is my boyfriend, Jin Lu. His unwavering companionship and relentless encouragement have been instrumental throughout my PhD journey. His presence has been a beacon of motivation, and I am deeply thankful for his steadfast support.

The School of Mathematical and Statistical Sciences at Arizona State University

deserves my utmost appreciation for providing the necessary resources and financial assistance that have been crucial in the successful completion of my doctoral study. Especially TA coordinator Katie Kolossa, academic advisor Joelle Park and Jennie Burel, Whenever I require assistance regarding my program or assignment, they are consistently available and willing to provide guidance and support. Their willingness to assist me has been priceless and has contributed significantly to my success in completing my tasks efficiently. Without their support, none of this would have been possible. Additionally, I would like to express my gratitude to the ASU Advanced Computing Center for enabling the computational aspects of my research, which have been vital to this progress.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Living in an era of data explosion, spatiotemporal data like traffic data and climate forecasting are ubiquitous and have been a trending topic in research. Traditional solutions for these problems often ignore the spatial or temporal correlations in the data (static model) or simply model the data summarized over time (time-averaged model). In either case, the data information that contains the spatiotemporal interactions is not fully utilized for parameter learning, which leads to insufficient modeling in these problems.

Inverse problems involving spatiotemporal observations are pervasive in scientific research and engineering applications. These problems necessitate spatiotemporal modeling due to the reliance on observed multivariate time series for inferring parameters of physical or biological significance. This work applies Bayesian models with spatiotemporal likelihood and prior to various inverse problems. Specifically, I utilize Spatiotemporal Gaussian processes (STGP) as a likelihood function for inverse problems, showcasing the effectiveness of spatial and temporal information in parameter estimation and uncertainty quantification. The superiority of Bayesian spatiotemporal likelihood modeling is demonstrated compared to traditional static and time-averaged methods, using a time-dependent advection-diffusion partial differential equation (PDE) and three chaotic ordinary differential equations (ODE). The theoretical justification for the efficacy of spatiotemporal modeling, even in complex scenarios like chaotic dynamics, is also provided. Regarding spatiotemporal prior modeling, the Spatiotemporal Besov Process (STBP) is employed as a regularization

method to address ill-posedness and model constraints. Two limited-angle CT reconstruction examples showcase the advantage of the proposed STBP in preserving spatial features while accounting for temporal changes compared with classic STGP and a time-uncorrelated approach.

There is a burgeoning interest in UQ within applied mathematics, physics, and engineering. This interest is driven by the necessity to calibrate model inadequacies, conduct sensitivity analyses, and engage in optimal control under conditions of uncertainty. The Bayesian methodology has garnered substantial attention for its dual capacity to facilitate parameter estimation and address the pivotal facet of UQ by deriving insights from the standard deviation of posterior samples.

In pursuing high-dimensional UQ, the Calibration-Emulation-Sampling (CES) scheme, as elaborated in [28], emerges as a promising framework due to its inherent structure. However, its practical implementation necessitates substantial computational resources to manage the expansive dimensions effectively. I propose incorporating deep neural networks (DNNs) during the emulation phase to enhance the CES framework. This substitution arises from the DNNs' computational efficiency and reduced susceptibility to training set variations. Additionally, using an autoencoder (AE) during the sampling stage helps by mapping to a lower dimension, reducing complexities from high dimensions.

1.1 Background

Uncertainty Quantification(UQ)

UQ is essential for assessing and managing uncertainties in predictions and decisions, ensuring informed choices and strategies in complex systems. It enhances reliability by

acknowledging potential variations and risks, contributing to effective decision-making. As a result, Bayesian methods for inverse problems, such as reservoir modeling [39, 40, 167, 161] and weather forecasting [155, 61], have become increasingly popular. Models in these application domains are usually constrained to physical laws and are typically represented as ODE/PDE systems, which involve expensive numerical simulations. Therefore, Bayesian UQ for such physics-constrained inverse problems is quite difficult because they involve 1) computationally intensive simulations for solving ODE/PDE systems and 2) sampling from the resulting high dimensional posterior distributions. To address these issues, we follow the work of [29] and propose a scalable framework for Bayesian UQ that combines ensemble Kalman methods and Markov Chain Monte Carlo (MCMC) algorithms.

Ensemble Kalman methods, originated from geophysics [49], have achieved significant success in state estimation for complex dynamical systems with noisy observations [50, 70, 1, 48, 46, 80, 86, 71]. More recently, these methods have been used to solve inverse problems to estimate the model parameters instead of the states [119, 27, 44, 74, 73, 47, 55]. As a gradient-free optimization algorithm based on a few ensembles, these methods gained popularity for solving inverse problems since they can be implemented non-intrusively and in parallel. However, due to the collapse of ensembles [135, 136, 159, 24], they tend to underestimate the posterior variances and thus fail to provide a rigorous basis for systematic UQ. Combining Kalman methods with MCMC can alleviate this issue. This approach consists of three stages: (i) calibrating models with ensemble Kalman methods, (ii) emulating the parameter-to-data map using evaluations of forward models, and (iii) sampling posteriors with MCMC based on cheaper emulators. We refer to this approach as *Calibration-Emulation-Sampling (CES)* [29]. Two immediate benefits of such framework are 1) the reuse of expensive

forward evaluations, and 2) the computationally cheap surrogates in the MCMC procedure.

Spatiotemporal Modeling for Inverse Problems

Spatiotemporal data are ubiquitous in inverse problems, typically recorded as multivariate time series. There are examples in fluid dynamics that describe the flow of liquid (e.g., petroleum) or gas (e.g., flame jet) [79]. Other examples include dynamical systems with chaotic behavior prevalent in weather prediction [106], biology [104], economics [15] etc., where small perturbation of the initial condition could lead to a significant deviation from what is observed/calculated in time. Such inverse problems aim to recover the parameters from given observations and knowledge of the underlying physics. The spatiotemporal information is crucial and should be respected when considering proper statistical models for parameter learning. This is not only of interest in statistics but also beneficial for practical applications of physics and biology to obtain inverse solutions and UQ more effectively.

Traditional methods for these spatiotemporal inverse problems often ignore the time dependence in the data for a simplified solution [154, 29, 94]. They either treat the observed time series statically as independent identically distributed (i.i.d.) observations across times [154, 94] (hence refer to it as "static" model), or summarize them by taking time average or higher order moments [114, 29, 72] (referred as "time-averaged" approach). The former is prevalent in Bayesian inverse problems with time series observations [94]. The latter is especially common in parameter learning of chaotic dynamics, e.g., Lorenz systems [106, 29], due to their sensitivity to the initial conditions and the system parameters, which in turn causes a rough landscape

of the objective function. In both scenarios, the spatiotemporal information is not fully integrated into the statistical modeling.

Regularization on Function Spaces

Regularization on function spaces is one of the fundamental questions in statistics and machine learning. High-dimensional objects such as images can be viewed as evaluation of proper functions. Statistical models for these objects on function spaces demand regularization to induce sparsity, prevent over-fitting, produce meaningful reconstruction, etc. The Gaussian process has been a popular choice for the L_2 penalty or function space prior. However, this approach can result in over-smoothed candidate functions when modeling objects with sharp edges, such as images. To address this issue, researchers have proposed a class of L_1 penalty based priors including Laplace random field [125, 108, 88] and Besov process(BP) [98, 36, 38]. Particularly, BP defined by wavelet expansions with random coefficients has been proposed as a more appropriate prior due to its discretization-invariant property for this type of Bayesian inverse problems. They have been extensively applied in spatial modeling [125], signal processing [88] and imaging analysis [152, 108].

1.2 Inverse Problem

1.2.1 Introduction

'Inverse problem' refers to using the results of actual observations to infer the values of the parameters characterizing the system under investigation. In detail, the inverse problem is composed of 3 elements. We call objects of interest *"parameters"*,

information collected about these objects like actual observations *"measurements"* or *"data"* and the mapping(forward problem) *"measurement operator(MO)"*. Taking Computed Tomography(CT) scan[1] as example, *MO* refers to a rotating X-ray tube and a row of detectors which are placed in a gantry, then the tomographic (cross-sectional) images(virtual "slices") of a body would be *measurements* processed on a computer using tomographic reconstruction algorithms from those multiple X-ray operator taken from different angles. Finally, *parameters* refer to varying tissues inside the body that could interact with X-ray detectors and record X-ray attenuations.

Inverse problems are typically challenging to solve for at least two different reasons: (1) different values of the model parameters may be consistent with the data (knowing the height of the main-mast is not sufficient for calculating the age of the captain), and (2) discovering the values of the model parameters may require the exploration of a vast parameter space (finding a needle in a 100-dimensional haystack is difficult).

The Bayesian framework has been introduced to solve challenges in inverse problems like reservoir modeling and weather forecasting. This framework utilizes previous knowledge of unknown parameters to optimize the entire process and account for uncertainty through posterior distribution.

1.2.2 Bayesian Inverse Problem

In many physics-constrained inverse problems, we are interested in finding an unknown parameter function u based on observations y. A forward mapping $\mathcal{G} : \mathbb{X} \mapsto \mathbb{Y}$ from a separable Hilbert space $\mathbb{X}$ to the data space $\mathbb{Y}$ (e.g. $\mathbb{Y} = \mathbb{R}^m$ for $m \geq 1$) connects

[1]CT scan, also known as X-ray scan, a medical imaging technique used to obtain detailed internal images of the body.

u to y as follows:

$$y = \mathcal{G}(u) + \eta \tag{1.1}$$

where $\eta \in \mathbb{Y}$ is assumed to be Gaussian noise $\eta \sim \mathcal{N}(0, \Gamma)$. We can define the potential function (negative log-likelihood), $\Phi : \mathbb{X} \times \mathbb{Y} \to \mathbb{R}$ as:

$$\Phi(u; y) = \frac{1}{2}\|y - \mathcal{G}(u)\|_{\Gamma}^2 = \frac{1}{2}\langle y - \mathcal{G}(u), \Gamma^{-1}(y - \mathcal{G}(u))\rangle \tag{1.2}$$

The solution could be computationally demanding due to non-linearity. Consequently, repeated forward evaluations of $\mathcal{G}(u)$ could be expensive for different u's, which could appear as coefficients or boundary conditions in these systems.

In the Bayesian setting, a prior measure μ_0 is imposed on u, independent of η. For example, we could assume a Gaussian prior $\mu_0 = \mathcal{N}(0, \mathcal{C})$ with the covariance $\mathcal{C}$ being a positive, self-adjoint, and trace-class operator on $\mathbb{X}$. Then we can obtain the posterior of u, denoted as μ^y, using Bayes' theorem [147, 37]:

Theorem 1.2.1. *(**Bayes' theorem**) Assume that*

$$0 < Z := \int_{\mathbb{X}} exp(-\Phi(u; y))\mu_0(du) < +\infty$$

Then $\mu|y$ is a random variable with Lebesgue density μ^y given by

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z}\, exp(-\Phi(u; y)) \tag{1.3}$$

Notice that the posterior μ^y can exhibit strongly non-Gaussian behavior, posing considerable challenges for efficiently sampling the distribution. For simplicity, I drop y from terms involved, so we denote the posterior as $\mu(du)$ and the potential function as $\Phi(u)$.

1.3 Outline

This dissertation is organized as follows. Chapter 1.2 introduces inverse problems and how to incorporate Bayesian ideas. Chapter 2 discusses UQ with a proposed class of hybrid MCMC algorithms named Dimension-Reduced Emulative Autoencoder Monte Carlo (DREAM) to improve the computational efficiency and also demonstrates advantage via simulation and two high-dimensional inverse problems. Chapter 3 provides an overview of Bayesian spatiotemporal likelihood modeling and its application in Advection-diffusion inverse problems and chaotic dynamic inverse problems such as Lorenz, Rossler, and Chen dynamics. Chapter 4 extends BP to the spatiotemporal domain (STBP) by substituting the random coefficients in the series expansion with stochastic time functions following the Q-exponential process which governs the temporal correlation strength, which could better preserve edges and capture sharp changes. Two limited-angle CT reconstruction examples are used to demonstrate the advantage of the proposed STBP. The last chapter 5 provides conclusions and discusses future research directions.

Chapter 2

UNCERTAINTY QUANTIFICATION(UQ)

This chapter is adapted from: "Lan, S., Li, S., &Shahbaba, B. (2022). Scaling Up Bayesian Uncertainty Quantification for Inverse Problems Using Deep Neural Networks. In SIAM/ASA Journal on Uncertainty Quantification (Vol. 10, Issue 4, pp. 1684–1713). Society for Industrial &Applied Mathematics (SIAM). https://doi.org/10.1137/21m1439456".

In this chapter, I will implement a novel UQ method to the inverse problems. First, I will introduce the details of this method. Afterwards, I will demonstrate how my innovative implementation of deep neural networks could scale up thousands of dimensions in subsections 2.3.1 and 2.3.2. Section 2.4 will showcase a comparison between my innovative approach, DREAM, and the conventional *Calibration-Emulation-Sampling* (CES) scheme in numerical experiments on elliptic inverse problem and advection-diffusion problem. This will undoubtedly highlight the superior efficiency and advantages of my approach.

2.1 Introduction

There is a growing interest in UQ to calibrate models and have a perception of to what extent we could trust the model. Furthermore, in the inference stage I could recover the distribution of parameters or make prediction with mean estimate and variance(uncertainty).

Bayesian approach has recently gained popularity for UQ in applied mathematics,

physics, and engineering, especially in ordinary and partial differential equation (ODE/PDE) systems, which involve expensive numerical simulation, since it makes it possible to obtain samples following desired posterior distribution and afterwards people would take variance as the uncertainty of those samples.

2.2 Calibration-Emulation-Sampling(CES)

From section 1.2.2, we knew the two major challenges(expensive forward evaluation and sampling high dimensional non-Gaussian posterior) when trying to work on UQ for inverse problems. Moreover, the high dimensionality of the discretized parameter of u makes the forward evaluation computationally intensive and challenges the robustness of sampling algorithms.

To address all these issues, [29] proposes CES as a favorable framework for high dimensional UQ and approximate Bayesian parameter learning since it combines Kalman methods with MCMC to alleviate the long-standing issue of underestimating posterior variances due to the collapse of ensemble in Kalman-based methods. The CES framework consists of the following three stages [29]:

1. **Calibration**: using optimization-based algorithms (ensemble Kalman) to obtain parameter estimation and collect expensive forward evaluations for the emulation step;
2. **Emulation**: recycling forward evaluations in the calibration stage to build an emulator for sampling;
3. **Sampling**: sampling the posterior approximately based on the emulator, which is much cheaper than the original forward mapping.

The CES scheme is promising for high-dimensional Bayesian UQ in inverse problems.

Emulation bypasses the expensive evaluation of original forward models (dominated by the cost of repeated forward solving of ODE/PDE systems) and reduces the cost of sampling to a small computational overhead. The sampling also benefits from the calibration, which provides MCMC algorithms with a good initial point in the high-density region to reduce the burning time.

2.2.1 Calibration: Ensemble Kalman Methods

Initializing J ensemble particles $\{u^{(j)}\}_{j=1}^{J}$ with, for example, prior samples, the basic ensemble Kalman inversion (EKI) method uses the following iterative equation to estimate the unknown function u:

$$u_{n+1}^{(j)} = u_n^{(j)} + C_n^{up}(C_n^{pp} + h^{-1}\Gamma)^{-1}(y_{n+1}^{(j)} - \mathcal{G}(u_n^{(j)})), \quad j = 1, \cdots, J,\ n = 1, \cdots, N-1 \tag{2.1}$$

where $h = 1/N$, $y_{n+1}^{(j)} = y + \xi_{n+1}^{(j)}$ with $\xi_{n+1}^{(j)} \sim \mathcal{N}(0, h^{-1}\Sigma)$, $\overline{u}_n := \frac{1}{J}\sum_{j=1}^{J} u_n^{(j)}$, $\overline{\mathcal{G}}_n := \frac{1}{J}\sum_{j=1}^{J} \mathcal{G}(u_n^{(j)})$, and

$$C_n^{pp} = \frac{1}{J}\sum_{j=1}^{J}(\mathcal{G}(u_n^{(j)}) - \overline{\mathcal{G}}_n) \otimes (\mathcal{G}(u_n^{(j)}) - \overline{\mathcal{G}}_n),$$
$$C_n^{up} = \frac{1}{J}\sum_{j=1}^{J}(u_n^{(j)} - \overline{u}_n) \otimes (\mathcal{G}(u_n^{(j)}) - \overline{\mathcal{G}}_n)$$

It can be shown [135] that Equation (2.1) has the following time-continuous limit as $h \to 0$:

$$\frac{du^{(j)}}{dt} = \frac{1}{J}\sum_{k=1}^{J}\left\langle \mathcal{G}(u^{(k)}) - \overline{\mathcal{G}}, y - \mathcal{G}(u^{(j)}) + \sqrt{\Sigma}\frac{dW^{(j)}}{dt}\right\rangle_{\Gamma}(u^{(k)} - \overline{u}) \tag{2.2}$$

where $\{W^{(j)}\}$ are independent cylindrical Brownian motions on $\mathbb{Y}$. I can set $\Sigma = 0$ to remove noise for an optimization algorithm or set $\Sigma = \Gamma$ to add noise for a dynamics

that transforms the prior to the posterior in one-time unit for linear forward maps [135, 55].

A variant of EKI to approximate sample from the posterior is the ensemble Kalman sampler (EKS) [55]. This is obtained by adding a prior-related damping term as in [24] and modifying the position-dependent noise in Equation (2.2):

$$\frac{du^{(j)}}{dt} = \frac{1}{J}\sum_{k=1}^{J} \left\langle \mathcal{G}(u^{(k)}) - \overline{\mathcal{G}}, y - \mathcal{G}(u^{(j)}) \right\rangle_{\Gamma} (u^{(k)} - \overline{u}) - C(u)\mathcal{C}^{-1}u^{(j)} + \sqrt{2C(u)}\frac{dW^{(j)}}{dt} \tag{2.3}$$

where $C(u) := \frac{1}{J}\sum_{j=1}^{J}(u^{(j)} - \overline{u}) \otimes (u^{(j)} - \overline{u})$. The time discretization using a linearly implicit split-step scheme is given by [55] with an adaptive time scheme Δt_n as in [87].

$$\begin{aligned} u_{n+1}^{(*,j)} &= u_n^{(j)} + \Delta t_n \frac{1}{J}\sum_{k=1}^{J} \left\langle \mathcal{G}(u_n^{(k)}) - \overline{\mathcal{G}}, y - \mathcal{G}(u_n^{(j)}) \right\rangle_{\Gamma} u_n^{(k)} - \Delta t_n C(u_n)\mathcal{C}^{-1}u_{n+1}^{(*,j)} \\ u_{n+1}^{(j)} &= u_{n+1}^{(*,j)} + \sqrt{2\Delta t_n C(u_n)}\xi_n^{(j)} \end{aligned} \tag{2.4}$$

However, due to the collapse of ensembles [135, 136, 159, 24], sample variance estimated by ensembles tends to underestimate the true uncertainty. Therefore, these methods do not provide a rigorous basis for systematic UQ. Figure 1 illustrates that both EKI and EKS severely underestimate the posterior standard deviation (plot in the 2d domain) of the parameter function in an elliptic inverse problem (see more details in Section 2.4.1). In what follows, we discuss scalable (dimension robust) inference methods and propose using them in the CES sampling step.

2.2.2 Emulation

After calibration, input-output pairs ($\{u, \mathcal{G}(u)\}$) are given from calibration step EKI or EKS, then GP is a preferred way to emulate the forward mapping $\mathcal{G}(u)$ in

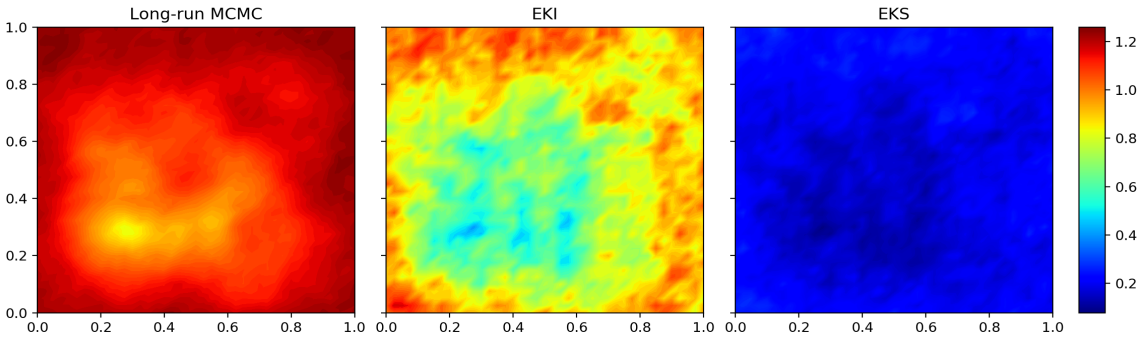


Figure 1. Comparing the estimation of standard deviation by MCMC (left) and ensemble Kalman methods (right two) in an elliptic inverse problem (Section 2.4.1).

equation (1.1) of inverse problems, avoiding the repeated computationally expensive forward evaluation. More importantly, from [28] GP leads to a more practical Bayesian inference during Markov Chain Monte Carlo (MCMC) sampling since it serves to remove the noise. Check subsection 3.2.1 for a detailed review of GP.

2.2.3 Sampling

Traditional Metropolis-Hastings algorithms are characterized by deteriorating mixing times upon mesh-refinement or with increasing dimensions. In contrast, a new class of dimension-robust algorithms, including *preconditioned Crank-Nicolson (pCN)* [31], *infinite-dimensional MALA (∞-MALA)* [13], *infinite-dimensional HMC (∞-HMC)* [10] and *infinite-dimensional manifold MALA (∞-mMALA)* [11], are well-defined on the infinite-dimensional Hilbert space, and thus yield the important computational benefit of dimension-independent mixing times for finite but high-dimensional problems in practice.

Consider the following continuous-time Hamiltonian dynamics:

$$\frac{d^2u}{dt^2} + \mathcal{K}\left\{\mathcal{C}^{-1}u + D\Phi(u)\right\} = 0, \quad \left(v := \frac{du}{dt}\right)\Big|_{t=0} \sim \mathcal{N}(0, \mathcal{K}) \ . \tag{2.5}$$

If we let $\mathcal{K} \equiv \mathcal{C}$, Equation (2.5) preserves the total energy $H(u, v) = \Phi(u) + \frac{1}{2}\|v\|^2_{\mathcal{K}}$. HMC algorithm [115] solves the dynamics (2.5)

using the following Störmer-Verlet symplectic integrator [153]:

$$\begin{aligned} v^- &= v_0 - \tfrac{\alpha\varepsilon}{2}\,\mathcal{C}D\Phi(u_0) \ ; \\ \begin{bmatrix} u_\varepsilon \\ v^+ \end{bmatrix} &= \begin{bmatrix} \cos\varepsilon & \sin\varepsilon \\ -\sin\varepsilon & \cos\varepsilon \end{bmatrix} \begin{bmatrix} u_0 \\ v^- \end{bmatrix} \ ; \\ v_\varepsilon &= v^+ - \tfrac{\alpha\varepsilon}{2}\,\mathcal{C}D\Phi(u_\varepsilon) \ . \end{aligned} \tag{2.6}$$

Equation (2.6) gives rise to the leapfrog map $\Psi_\varepsilon : (u_0, v_0) \mapsto (u_\varepsilon, v_\varepsilon)$. Given a time horizon τ and current position u, the MCMC mechanism proceeds by concatenating $I = \lfloor \tau/\varepsilon \rfloor$ steps of leapfrog map consecutively,

$$u' = \mathcal{P}_u\big\{\Psi^I_\varepsilon(u, v)\big\} \ , \quad v \sim \mathcal{N}(0, \mathcal{K}) \ .$$

where $\mathcal{P}_u$ denotes the projection onto the u-argument. Then, the proposal u' is accepted with probability $a(u, u') = 1 \wedge \exp(-\Delta H(u, v))$, where

$$\begin{aligned} \Delta H(u, v) =& H(\Psi^I_\varepsilon(u, v)) - H(u, v) \\ =& \Phi(u_I) - \Phi(u_0) - \frac{\alpha^2\varepsilon^2}{8}\left\{\|\mathcal{C}^{\frac{1}{2}}D\Phi(u_I)\|^2 - \|\mathcal{C}^{\frac{1}{2}}D\Phi(u_0)\|^2\right\} \\ & - \frac{\alpha\varepsilon}{2}\sum_{i=0}^{I-1}(\langle v_i, D\Phi(u_i)\rangle + \langle v_{i+1}, D\Phi(u_{i+1})\rangle) \end{aligned}$$

This yields ∞-HMC [10]. We can use different step-sizes in (2.6): ε_1 for the first and third equations, and ε_2 for the second equation and let $I = 1$, $\varepsilon_1^2 = h, \cos\varepsilon_2 = \frac{1-h/4}{1+h/4}, \sin\varepsilon_2 = \frac{\sqrt{h}}{1+h/4}$. Then, ∞-HMC reduces to ∞-MALA, which can also be derived

from Langevin dynamics [13, 12]. When $\alpha = 0$, ∞-MALA further reduces to pCN [12], which does not use gradient information and can be viewed as an infinite-dimensional analogy of random walk Metropolis.

After exhaustive summarization of the CES scheme, let me highlight multiple contributions as follows, which apply deep neural networks in emulation and sampling stage:

1. apply CNN to train emulators for Bayesian inverse problems,
2. embed AE in CES to significantly improve its computational efficiency,
3. scale Bayesian UQ for physics-constrained inverse problems up to thousands of dimensions with DREAM.

2.3 Dimension Reduced Emulative Autoencoder MCMC(DREAM)

Combining CNN and AE, I propose a class of hybrid MCMC algorithms named *Dimension-Reduced Emulative Autoencoder Monte Carlo (DREAM)* that can improve and scale up the application of the CES framework for Bayesian UQ from hundreds of dimensions (with GP emulation) [29] to thousands of dimensions. Details of emulating functions, extracting gradients, and reducing dimensions will be discussed in subsection 2.3.1 and 2.3.2.

2.3.1 Scaling Up Bayesian UQ with CNNs-Emulation

There are two main challenges that limit the scalability of Bayesian UQ for inverse problems: the intensive computation required for repeated evaluations of likelihood (potential), $\Phi(u)$, and the large dimensionality of the discretized space. When using

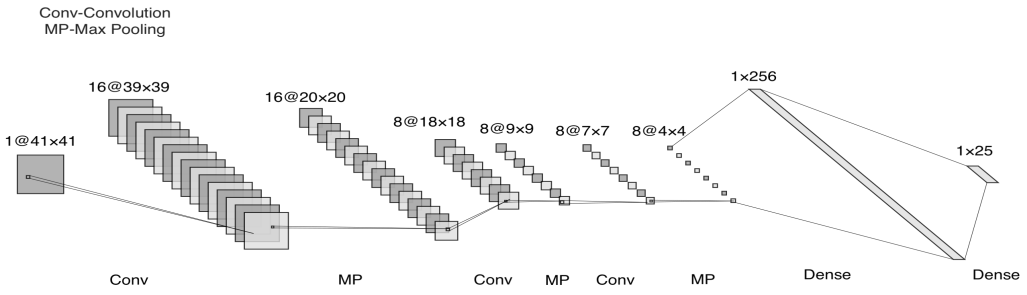


Figure 2. A Typical Architecture of Convolutional Neural Network (CNN)

∞-MALA or ∞-HMC, the gradient $D\Phi(u)$ is required but often unavailable. To effectively tackle the issues at hand, I plan to employ the use of neural networks. More specifically, training CNN to emulate the forward evaluation and AE to reduce the parameter dimensionality. In the following, I discretize the parameter function u and denote its dimension as d. I still denote the discretized parameter as u when there is no confusion. Usually, $d \gg 1$.

The ensemble-based algorithms in the calibration phase produce parameters and forward solutions $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{j=1}^J$ for $n = 0, \cdots, N$. These input-output pairs can be used to train a DNN as an emulator $\mathcal{G}^e$ of the forward mapping $\mathcal{G}$ [58]:

$$\mathcal{G}^e(u;\theta) = F_{K-1} \circ \cdots \circ F_0(u), \quad F_k(\cdot) = g_k(W_k \cdot + b_k) \in C(\mathbb{R}^{d_k}, \mathbb{R}^{d_{k+1}}) \tag{2.7}$$

where $d_0 = d$, $d_K = m$; $W_k \in \mathbb{R}^{d_{k+1} \times d_k}$, $b_k \in \mathbb{R}^{d_{k+1}}$, $\theta_k = (W_k, b_k)$, $\theta = (\theta_0, \cdots, \theta_{K-1})$; and g_k's are activation functions. There are multiple choices of activation functions, e.g. $g_k(x) = (\sigma(x_1), \cdots, \sigma(x_{d_{k+1}}))$ with $\sigma \in C(\mathbb{R}, \mathbb{R})$ including rectified linear unit (ReLU, $\sigma(x_i) = 0 \vee x_i$), leaky ReLU ($\sigma(x_i;\alpha) = x_i I(x_i \geq 0) + \alpha x_i I(x_i < 0)$), tanh ($\sigma(x_i) = (e^{2x_i} - 1)/(e^{2x_i} + 1)$); or alternatively, $g_k(x) = (\sigma_1(x), \cdots, \sigma_{d_{k+1}}(x)) \in C(\mathbb{R}^{d_{k+1}}, \mathbb{R}^{d_{k+1}})$ such as softmax ($\sigma_i(x) = e^{x_i} / \sum_{i'=1}^{d_{k+1}} e^{x_{i'}}$).

In many physics-constrained inverse problems, the parameter function u is defined over a 2-d or 3-d field, which possesses unique spatial features resembling an image. This has motivated the choice of CNN for emulators. Inspired by biological processes, where the connectivity pattern between neurons resembles the organization of the visual cortex [54], CNN has become a powerful tool in image recognition and classification [89]. As a regularized neural network with varying depth and width, CNN has fewer connections and thus fewer training parameters than standard fully connected DNNs of similar size. Therefore, CNN is preferred to DNN in the CES framework due to its flexibility and computational efficiency.

In general, CNN consists of a series of convolution layers with filters (kernels), pooling layers to extract features, and fully connected layers to connect to outputs. The convolutional layer is introduced for sparse interaction, parameter sharing, and equivalent representations [58]. Therefore, instead of full matrix multiplication, I consider the following discrete convolution [100]

$$F_k(\cdot) = g_k(w_k * \cdot + b_k) \in C(\mathbb{R}^{d_k}, \mathbb{R}^{d_{k+1}}) \tag{2.8}$$

where w_k is a kernel function with discrete format defined by (multiplying) a circulant matrix W_k^*. Convolution is the first layer to extract spatial features (corresponding to filters) from an input image. Each image could be seen as a $c \times h \times w$ array of pixels ($c = 3$ for RGB images and $c = 1$ for grayscale images; h, w are image size in height and width respectively). For convolution of the image with the kernel, CNN slides a pre-specified size (kernel size) window with certain stride (step size) over the image. The resulting operation typically reduces dimension.

After the convolutional layer, a pooling layer is added to reduce the number of parameters by generating summary statistics of the nearby outputs. Such operation is a form of non-linear down-sampling that sparsifies the neural network but retains

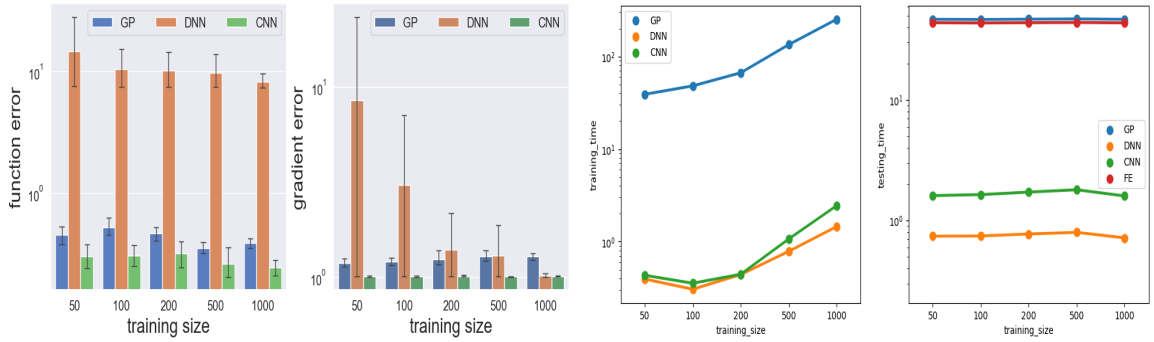


Figure 3. Comparing the emulation $\mathcal{G}^e : \mathbb{R}^{1681} \to \mathbb{R}^{25}$ in an elliptic inverse problem(Section 2.4.1) by GP, DNN and CNN in terms of error (left) and time (right).

the essential input image information. Spatial pooling can be different, such as max-pooling, average-pooling, and sum-pooling. Finally, we use a dense layer to generate forward outputs $\{\mathcal{G}(u)\}$. Figure 2 illustrates the structure of a CNN used in the elliptic inverse problem (Section 2.4.1).

CES [29] uses GP for the emulation step. However, CNN has several advantages over GP for building the emulator: 1) it is computationally more efficient for large training sets, 2) it is less sensitive to the locations of training samples, and 3) we could take advantage of all the ensemble samples collected by EKI or EKS to train CNN without the need to carefully "design" a training set of controlled size as it is common in GP. After the emulator is trained, we could approximate the potential function using the prediction of CNN:

$$\Phi(u^*) \approx \Phi^e(u^*) = \frac{1}{2}\|y - \mathcal{G}^e(u^*)\|_\Gamma^2 \tag{2.9}$$

In the sampling stage, significant computation will be saved if we use Φ^e instead of Φ in the accept/reject step of MCMC. If the emulator is a good representation of the forward mapping, then the difference between Φ^e and Φ is small. Thus the samples

by such emulative MCMC have the stationary distribution with a slight discrepancy compared to the true posterior $\mu(du)$.

In gradient-based MCMC algorithms, we need to calculate $D\Phi(u)$ – the derivatives of (log) density function $\Phi(u)$ with respective to parameter function u. However, it can be obtained by using the network emulator:

$$D\Phi^e(u^*) = -\langle y - \mathcal{G}^e(u^*), D\mathcal{G}^e(u^*)\rangle_\Gamma \tag{2.10}$$

where $D\mathcal{G}^e(u^*)$ can be the output from CNN's back-propagation, e.g., implemented in `GradientTape` of `TensorFlow`. We can see that $D\Phi(u^*) \approx D\Phi^e(u^*)$ if $D\Phi(u^*)$ exists.

The following theorem generalizes [169] and gives the error bound of the CNN emulator in approximating the true potential Φ and its gradient $D\Phi$.

Theorem 2.3.1. *Let $2 \le s \le d$ and $\Omega \subset [-1,1]^d$. Assume $\mathcal{G}_j \in H^r(\mathbb{R}^d)$ for $r > 2 + d/2$, $j = 1, \cdots, m$. If $K \ge 2d/(s-1)$, then there exist $\mathcal{G}^e$ by CNN with ReLU activation function such that*

$$\|\Phi - \Phi^e\|_{H^1(\Omega)} \le c\|\mathcal{G}\|\sqrt{\log K}K^{-\frac{1}{2}-\frac{1}{2d}} \tag{2.11}$$

where we have $\|\Phi\|_{H^1(\Omega)} = \left(\|\Phi\|^2_{L^2(\Omega)} + \|D\Phi\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}$, c is an absolute constant and $\|\mathcal{G}\| = \max_{1\le j\le m}\|\mathcal{G}_j\|_{H^r(\mathbb{R}^d)}$ with $\|\mathcal{G}_j\|_{H^r(\mathbb{R}^d)} := \|(1+|\omega|^2)^{r/2}\widehat{\mathcal{G}}_j(\omega)\|_{L^2(\mathbb{R}^d)}$.

Proof. See Appendix A.1. □

Remark 1. *Based on Theorem 3 of [109], we have a weaker bound with sup-norm:*

$$\|\Phi - \Phi^e\|_{W^{1,\infty}(\Omega)} \le \tilde{c}\|\mathcal{G}\|K^{-\frac{1}{2}} \tag{2.12}$$

where $\|\Phi\|_{W^{1,\infty}(\Omega)} = \max_{0\le i\le d}\|D_i\Phi\|_{L^\infty(\Omega)}$ ($D_0\Phi = \Phi$). Since $H^r(\Omega) \hookrightarrow W^{1,\infty}(\Omega) \hookrightarrow C(\Omega)$, we can show that:

- *For any continuous forward mapping $\mathcal{G}$, a CNN emulator, $\mathcal{G}^e$, with depth K can be built such that $\|\Phi - \Phi^e\|_{L^\infty(\Omega)} \to 0$ as $K \to \infty$.*
- *If $\mathcal{G}$ is further continuously differentiable, a CNN, $\mathcal{G}^e$, with depth K exists such that $\|\Phi - \Phi^e\|_{W^{1,\infty}(\Omega)} \to 0$ as $K \to \infty$.*

Even if $D\Phi(u^*)$ does not exist, such gradient information $D\Phi^e(u^*)$ can still be extracted from the emulator $\mathcal{G}^e$ to inform the landscape of Φ in the vicinity of u^*. Note that we train CNN only on $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}$ as opposed to $\{u_n^{(j)}, D\mathcal{G}(u_n^{(j)})\}$. That is, no gradient information is used for training. This is similar to extracting geometric information from GP emulator [146, 96]. Figure 3 compares GP, DNN, and CNN in emulating a forward map that takes a $1681(41 \times 41)$ dimensional discretized parameter function with 25 observations taken from the solution of an elliptic PDE as the output (see more details in Section 2.4.1). Given limited training data, CNN outperforms both GP and DNN in rendering smaller approximation errors with less time for consumption.

2.3.2 Scaling Up Bayesian UQ with AE-Sampling

Although emulation can reduce computation, the MCMC algorithms used for Bayesian inference are still defined in high-dimensional spaces. In this section, I use AE to reduce the dimensions and further speed up the UQ process [140]. AE is a special type of feed-forward neural network for latent representation learning. The input is encoded into a low-dimensional latent representation (code). The code is then decoded into a reconstruction of the original input (see Figure 4 for the structure of an AE). The model is trained to minimize the difference between the input and the

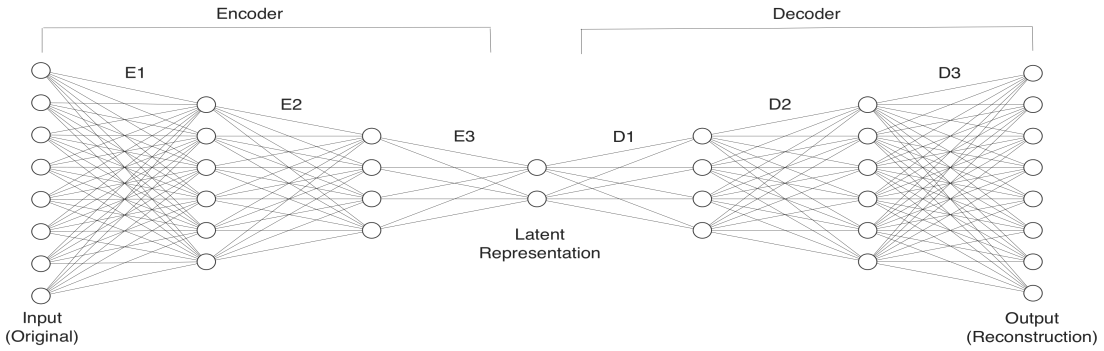


Figure 4. A Typical Architecture of Autoencoder (AE) Neural Network

reconstruction. The AE could learn complicated nonlinear dimensionality reduction and thus is widely used in many challenging tasks such as image recognition and artificial data generation [66].

While AE is commonly used to reduce the dimensionality of the data, here we use it to reduce the dimensionality of the parameter space. Denote the latent space as $\mathbb{X}_L$ with dimensionality $d_L \ll d$. Let $u_L \in \mathbb{X}_L$ be the latent representation of parameter u. Then the encoder ϕ and the decoder ψ are defined respectively as follows

$$\begin{aligned} \phi &: \mathbb{X} \to \mathbb{X}_L, \quad u \mapsto u_L \\ \psi &: \mathbb{X}_L \to \mathbb{X}, \quad u_L \mapsto u_R \end{aligned} \tag{2.13}$$

where $u_R \in \mathbb{X}$ is a reconstruction of u; ϕ and ψ can be chosen as multilayer neural networks as in Equation (2.7). Depending on the layers and structures, we could have convolutional AE (CAE) [59, 131], variational AE (VAE) [84, 83], etc.

According to the universal approximation theorem [34, 124, 107], a feed-forward artificial neural network can approximate any continuous function given some mild assumptions about the activation functions. Theoretically, an AE with suitable activation functions could represent an identity map, i.e., $\psi \circ \phi = id$. An accurate reconstruction of the input implies a good low-dimensional representation encoded in

ϕ. In practice, the algorithm's success heavily relies on the quality of the trained AE. Note that we train the AE with ensembles $\{u_n^{(j)}, j = 1, \cdots J, n = 0, \cdots, N\}$ from the calibration stage. Even though there is a difference between $\psi \circ \phi$ and the identity map id, AE could provide a reconstruction $\psi \circ \phi(u)$ very close to the original parameter u. See Figure 7b (Section 2.4.1) and Figure 12b (Section 2.4.2) for examples.

The potential function $\Phi(u)$ and its derivative $D\Phi(u)$ can be projected to the latent space $\mathbb{X}_L$ – denoted as $\Phi_r(u_L)$ and $D\Phi_r(u_L)$ respectively – as follows:

$$\begin{aligned} \Phi_r(u_L) &= \Phi(u) = \Phi(\psi(u_L)) \\ D\Phi_r(u_L) &= \left(\frac{\partial u}{\partial u_L}\right)^T \frac{\partial \Phi(u)}{\partial u} = (d\psi(u_L))^T D\Phi(\psi(u_L)) \end{aligned} \tag{2.14}$$

where $d\psi = \frac{\partial u}{\partial u_L}$ is the Jacobian matrix of size $d \times d_L$ for the decoder ψ. The derivative information $D\Phi_r(u_L)$ needed in the gradient-based MCMC, ∞-MALA and ∞-HMC will be discussed in Section 2.3.

In practice, I avoid explicit computation of the Jacobian matrix $d\psi$ by calculating the Jacobian-vector action altogether:

$$D\Phi_r(u_L) = \frac{\partial}{\partial u_L}[\psi(u_L)^T D\Phi(\psi(u_L))] \tag{2.15}$$

which is the output by AE's back-propagation.

The implementation of emulation merging is a critical strategy for minimizing the computational workload necessary for optimal performance. The resulting approximate MCMC algorithms in the latent space involve potential function and its derivative, denoted as $\Phi_r^e(u_L)$ and $D\Phi_r^e(u_L)$ respectively, which are defined by replacing Φ with Φ^e in Equation (2.14).

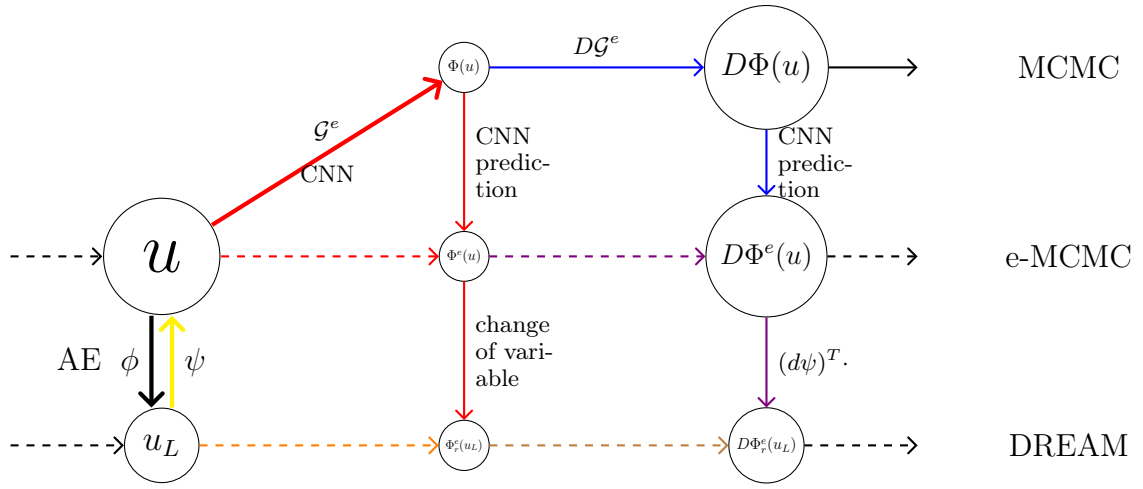


Figure 5. Relationship among quantities in various MCMC algorithms: node sizes indicate relative dimensions of these quantities. Thick solid arrows mean training neural networks. Dashed arrows with colors represent mappings that are not directly calculated but have equivalent compositions indicated by the same color, e.g., $u \mapsto \Phi^e(u)$ (dashed red arrow) obtained by training CNN (thick solid red arrow) followed by network prediction (solid red arrow); or by color mixing, e.g., $u_L \mapsto \Phi_r^e(u_L)$ (dashed orange arrow) as a result of combing the decoder ψ (thick solid yellow arrow), $u \mapsto \Phi^e(u)$ (dashed red arrow), and the change of variable (solid red arrow).

2.3.3 DREAM

Next, I combine all the abovementioned techniques to speed up Bayesian UQ for inverse problems. More specifically, the method is composed of the following three stages:

1. **Calibration**: collect JN samples $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{j,n}$ from EKI or EKS procedure;
2. **Emulation**: build an emulator of the forward mapping $\mathcal{G}^e$ based on $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{j,n}$ (and extract $D\mathcal{G}^e$) using CNN; train an AE (ϕ, ψ) based on $\{u_n^{(j)}\}_{j,n}$;
3. **Sampling**: run approximate MCMC based on emulation to propose u' from u:

   i) obtain the projection of u by $u_L = \phi(u)$;

ii) propose u'_L from u_L by ∞-MCMC (with Φ^e_r and $D\Phi^e_r$) in the latent space $\mathbb{X}_L$;

iii) obtain the sample $u' = \psi(u'_L)$

In the class of ∞-MCMC, I can use the emulated potential and its derivative instead of the faithful evaluations. Refer to the resulting algorithms as *emulative ∞-MCMC (e-MCMC)*. Further, AE is adopted to project these approximate MCMC into low-dimensional latent space. I denote these algorithms as *dimension-reduced emulative autoencoder ∞-MCMC (DREAM)*. Figure 5 illustrates the relationship among the quantities involved in these MCMC algorithms. For example, the mapping $u_L \mapsto D\Phi^e_r(u_L)$ (dashed brown arrow) is not directly calculated. Still, I could combine the decoder ψ (thick solid yellow arrow), emulated gradient $u \mapsto D\Phi^e(u)$ (dashed violet arrow), and left multiplying Jacobian matrix $(d\psi)^T$ (solid violet arrow).

Note that if we accept/reject proposals u'_L in the latent space with Φ^e_r, there is no need to constantly traverse between the original and the latent space. The chain can mainly stay in the latent space $\mathbb{X}_L$ to collect samples $\{u_L\}$, as shown in the bottom level of Figure 5, and only needs to go back to the original space $\mathbb{X}$ when relevant emulated quantities are required. In the following, the details of DREAM algorithms would be described.

For the convenience of following disposition, I first whiten the coordinates by the transformation $u \mapsto \tilde{u} := \mathcal{C}^{-\frac{1}{2}}u$. The whitened variable $\tilde{u}$ has the prior $\tilde{\mu}_0 = \mathcal{N}(0, \mathcal{I})$, where the identity covariance operator is not a trace-class on $\mathbb{X}$. However, random draws from $\tilde{\mu}_0$ are square-integrable in the weighted space $\mathrm{Im}(\mathcal{C}^{-\frac{1}{2}})$. It is important to note that even after the implementation of the transformation, I am still capable of generating an exceedingly accurate proposal for the function space of parameter u through the inversion technique mentioned in [33, 90].In the whitened coordinates $\tilde{u}$,

the Langevin and Hamiltonian (2.5) dynamics (with algorithmic parameter $\alpha \equiv 1$) become the following respectively

$$\frac{d\tilde{u}}{dt} = -\frac{1}{2}\left\{\mathcal{I}\tilde{u} + \alpha D\Phi(\tilde{u})\right\} + \frac{dW}{dt} \tag{2.16}$$

$$\frac{d^2\tilde{u}}{dt^2} + \left\{\mathcal{I}\tilde{u} + \alpha D\Phi(\tilde{u})\right\} = 0, \quad \left(\tilde{v} := \frac{d\tilde{u}}{dt}\right)\Big|_{t=0} \sim \mathcal{N}(0, \mathcal{I}) \ . \tag{2.17}$$

where $D\Phi(\tilde{u}) = \mathcal{C}^{\frac{1}{2}} D\Phi(u)$. Then I can train CNN based on $\{\tilde{u}_n^{(j)}, \mathcal{G}(\tilde{u}_n^{(j)})\}_{j,n}$ and AE based on $\{\tilde{u}_n^{(j)}\}_{j,n}$.

On the other hand, since the AE does not preserve the volume ($\psi \circ \phi \approx id$ but $\psi \circ \phi \neq id$), the acceptance of proposals in the latent space needs to be adjusted with a volume correction term $\frac{V'}{V}$ to maintain the ergodicity [97, 140]. Note, the volume adjustment term $\frac{V'}{V}$ breaks into the product of Jacobian determinants of the encoder ϕ and the decoder ψ that can be calculated with Gramian function as follows [140]:

$$\frac{V'}{V} = \det(d\psi(\tilde{u}'_L))\det(d\phi(\tilde{u})) = \sqrt{\det\left[\left(\frac{\partial\tilde{u}'}{\partial\tilde{u}'_L}\right)^T\left(\frac{\partial\tilde{u}'}{\partial\tilde{u}'_L}\right)\right]}\sqrt{\det\left[\left(\frac{\partial\tilde{u}_L}{\partial\tilde{u}}\right)\left(\frac{\partial\tilde{u}_L}{\partial\tilde{u}}\right)^T\right]} \tag{2.18}$$

where terms under square root are determinants of matrices with small size $d_L \times d_L$, which can be obtained by the Jacobian matrices' singular value decomposition, respectively. In practice, we can exclude $\frac{V'}{V}$ from the acceptance probability and use it as a resampling weight as in importance sampling [95]. Alternatively, we can ignore the accept/reject step for an approximate Bayesian UQ [158, 140].

2.3.3.1 DREAM-pCN

In the whitened latent space, pCN proposal becomes

$$\tilde{u}'_L = \rho\,\tilde{u}_L + \sqrt{1-\rho^2}\,\tilde{\xi}_L \ , \quad \tilde{\xi}_L \sim \mathcal{N}(0, I_{d_L}) \tag{2.19}$$

If using emulated potential energy, then the acceptance probability with volume adjustment (2.18) of the resulting DREAM-pCN algorithm is

$$a(\tilde{u}_L, \tilde{u}'_L) = 1 \wedge \exp\{-\Phi_r^e(\tilde{u}'_L) + \Phi_r^e(\tilde{u}_L) + \log\det(d\psi(\tilde{u}'_L)) + \log\det(d\phi(\tilde{u}))\}$$

2.3.3.2 DREAM-∞-MALA

Based on the Langevin dynamics in the whitened coordinates, ∞-MALA proposal in the latent space with emulated gradient becomes

$$\tilde{u}'_L = \rho\,\tilde{u}_L + \sqrt{1-\rho^2}\,\tilde{v}_L\ , \quad \tilde{v}_L = \tilde{\xi}_L - \tfrac{\alpha\sqrt{h}}{2}\,D\Phi_r^e(\tilde{u}_L)\ , \quad \rho = (1-\tfrac{h}{4})/(1+\tfrac{h}{4})\ . \tag{2.20}$$

The resulting DREAM-∞-MALA has adjusted acceptance probability $a(\tilde{u}_L, \tilde{u}'_L) = 1 \wedge \frac{\kappa(\tilde{u}'_L, \tilde{u}_L)}{\kappa(\tilde{u}_L, \tilde{u}'_L)}\frac{V'}{V}$ with $\frac{V'}{V}$ as in (2.18) and

$$\kappa(\tilde{u}_L, \tilde{u}'_L) = \exp(-\Phi_r^e(\tilde{u}_L)) \cdot \exp\left\{-\frac{\alpha^2 h}{8}\|D\Phi_r^e(\tilde{u}_L)\|^2 - \frac{\alpha\sqrt{h}}{2}\langle D\Phi_r^e(\tilde{u}_L), \tilde{v}_L(\tilde{u}_L, \tilde{u}'_L)\rangle\right\}$$

2.3.3.3 DREAM-∞-HMC

To derive ∞-HMC in the latent space based on the Hamiltonian dynamics in the whitened coordinates (2.17), I also need to project $\tilde{v} \sim \mathcal{N}(0, I_d)$ into d_L-dimensional latent space. I could have used the same encoder ϕ as in [140]; however, since $\tilde{v}_i \overset{iid}{\sim} \mathcal{N}(0,1)$ for $i = 1, \cdots, d$, I just set $\tilde{v}_L \sim \mathcal{N}(0, I_{d_L})$ for simplicity. Then, the ∞-HMC proposal $\Psi_\varepsilon : (\tilde{u}_{L,0}, \tilde{v}_{L,0}) \mapsto (\tilde{u}_{L,\varepsilon}, \tilde{v}_{L,\varepsilon})$ in the whitened augmented latent

Algorithm 1 Dimension Reduced Emulative Autoencoder ∞-dimensional HMC (DREAM-∞-HMC)

Require: Collect NJ samples $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{j,n}$ from EKI or EKS procedure; whiten coordinates $\{\tilde{u}_n^{(j)} = \mathcal{C}^{-\frac{1}{2}} u_n^{(j)}\}_{j,n}$.

Require: Build an emulator of the forward mapping $\mathcal{G}^e$ based on $\{\tilde{u}_n^{(j)}, \mathcal{G}^{(j)}\}_{j,n}$ (and extract $D\mathcal{G}^e$) using CNN; train an AE (ϕ, ψ) based on $\{\tilde{u}_n^{(j)}\}_{j,n}$;

Initialize current state $\tilde{u}^{(0)}$ and project it to the latent space by $\tilde{u}_L^{(0)} = \phi(\tilde{u}^{(0)})$

Sample velocity $\tilde{v}_L^{(0)} \sim \mathcal{N}(0, I_{d_L})$

Calculate current energy $E_0 = \Phi_r^e(\tilde{u}_L^{(0)}) - \frac{\alpha^2\varepsilon^2}{8}\|D\Phi_r^e(\tilde{u}_L^{(0)})\|^2 + \log\det(d\phi(\tilde{u}^{(0)}))$

for $i = 0$ to $I - 1$ **do**

  Run $\Psi_\varepsilon : (\tilde{u}_L^{(i)}, \tilde{v}_L^{(i)}) \mapsto (\tilde{u}_L^{(i+1)}, \tilde{v}_L^{(i+1)})$ according to (2.21).

  Update the energy $E_0 \leftarrow E_0 + \frac{\alpha\varepsilon}{2}(\langle \tilde{v}_{L,i}, D\Phi_r^e(\tilde{u}_{L,i})\rangle + \langle \tilde{v}_{L,i+1}, D\Phi_r^e(\tilde{u}_{L,i+1})\rangle)$

end for

Calculate new energy $E_1 = \Phi_r^e(\tilde{u}_L^{(I)}) - \frac{\alpha^2\varepsilon^2}{8}\|D\Phi_r^e(\tilde{u}_L^{(I)})\|^2 - \log\det(d\psi(\tilde{u}_L^{(I)}))$

Calculate acceptance probability $a = \exp(-E_1 + E_0)$

Accept $\tilde{u}_L^{(I)}$ with probability a for the next state $\tilde{u}_L'$ or set $\tilde{u}_L' = \tilde{u}_L^{(0)}$ in the latent space.

Record the next state $u' = \mathcal{C}^{\frac{1}{2}}\psi(\tilde{u}_L')$ in the original space.

space with emulated gradient becomes

$$\begin{aligned} \tilde{v}_L^- &= \tilde{v}_{L,0} - \tfrac{\alpha\varepsilon}{2} D\Phi_r^e(\tilde{u}_{L,0}) \ ; \\ \begin{bmatrix} \tilde{u}_{L,\varepsilon} \\ \tilde{v}_L^+ \end{bmatrix} &= \begin{bmatrix} \cos\varepsilon & \sin\varepsilon \\ -\sin\varepsilon & \cos\varepsilon \end{bmatrix} \begin{bmatrix} \tilde{u}_{L,0} \\ \tilde{v}_L^- \end{bmatrix} \ ; \\ \tilde{v}_{L,\varepsilon} &= \tilde{v}_L^+ - \tfrac{\alpha\varepsilon}{2} D\Phi_r^e(\tilde{u}_{L,\varepsilon}) \ . \end{aligned} \tag{2.21}$$

The acceptance probability for the resulting DREAM-∞-HMC algorithm involves $H(\tilde{u}_L, \tilde{v}_L) = \Phi_r^e(\tilde{u}_L) + \frac{1}{2}\|\tilde{v}_L\|^2$ and becomes $a(\tilde{u}_L, \tilde{u}_L') = 1 \wedge \exp(-\Delta H(\tilde{u}_L, \tilde{v}_L))\frac{V'}{V}$ with $\frac{V'}{V}$ as in (2.18) and

$$\begin{aligned} \Delta H(\tilde{u}_L, \tilde{v}_L) =& H(\Psi_\varepsilon^I(\tilde{u}_L, \tilde{v}_L)) - H(\tilde{u}_L, \tilde{v}_L) \\ =& \Phi(\tilde{u}_{L,I}) - \Phi(\tilde{u}_{L,0}) - \frac{\alpha^2\varepsilon^2}{8}\left\{\|D\Phi_r^e(\tilde{u}_{L,I})\|^2 - \|D\Phi_r^e(\tilde{u}_{L,0})\|^2\right\} \\ & - \frac{\alpha\varepsilon}{2}\sum_{i=0}^{I-1}(\langle \tilde{v}_{L,i}, D\Phi_r^e(\tilde{u}_{L,i})\rangle + \langle \tilde{v}_{L,i+1}, D\Phi_r^e(\tilde{u}_{L,i+1})\rangle) \end{aligned} \tag{2.22}$$

I summarize DREAM-∞-HMC in Algorithm 1, which includes DREAM-∞-MALA with $I = 1$ and DREAM-pCN with $\alpha = 0$.

2.4 Numerical Experiments

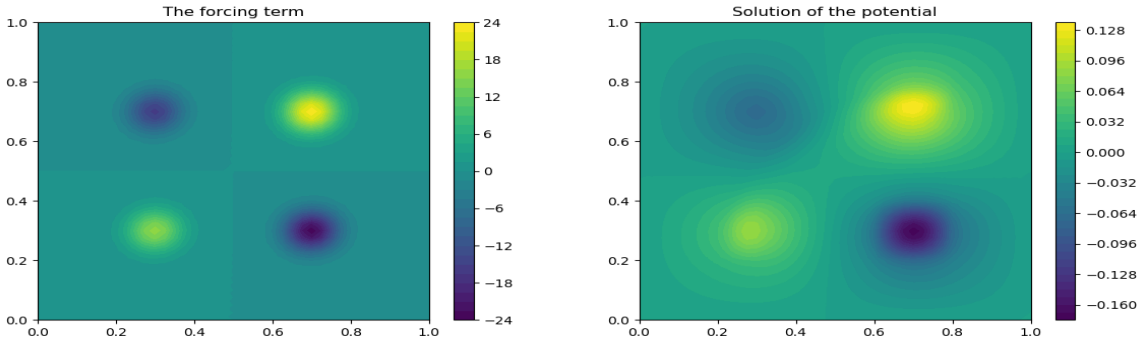
This section considers two high-dimensional inverse problems involving elliptic PDE and advection-diffusion equation. In both problems, the forward parameter-to-observation mappings are nonlinear, and the posterior distributions are non-Gaussian. The high dimensionality of the discretized parameter imposes a big challenge on Bayesian UQ. The second inverse problem involving advection-diffusion equation is even more difficult because it is based on spatiotemporal observations. I demonstrate substantial numerical advantages of proposed methods and show that they indeed can scale up the Bayesian UQ for PDE-constrained inverse problems to thousands of dimensions. Python codes are publicly available at https://github.com/lanzithinking/DREAM-BUQ.

2.4.1 Elliptic Inverse Problem

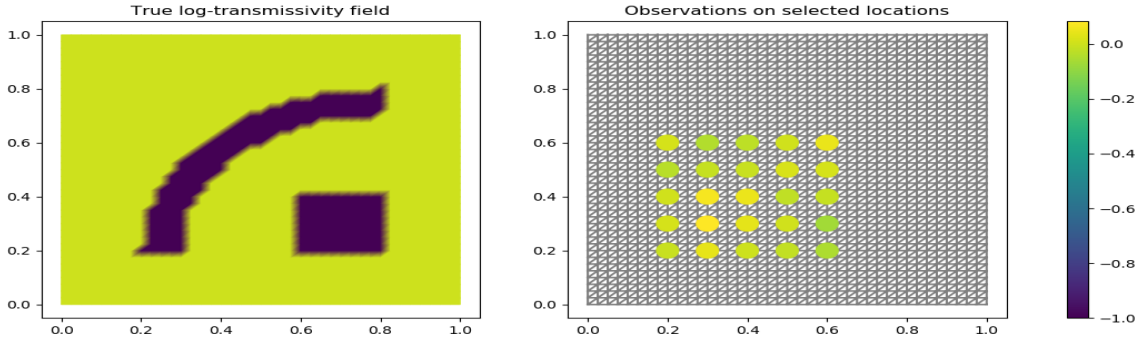
The following elliptic PDE [33, 90] is defined on the unit square domain $\Omega = [0, 1]^2$:

$$\begin{aligned} -\nabla \cdot (k(s)\nabla p(s)) &= f(s), \quad s \in \Omega \\ \langle k(s)\nabla p(s), \vec{n}(s)\rangle &= 0, \quad s \in \partial\Omega \\ \int_{\partial\Omega} p(s)dl(s) &= 0 \end{aligned} \tag{2.23}$$

where $k(s)$ is the transmissivity field, $p(s)$ is the potential function, $f(s)$ is the forcing term, and $\vec{n}(s)$ is the outward normal to the boundary. The source/sink term $f(s)$ is



(a) Forcing field $f(s)$ (left), and the solution $p(s)$ with true transmissivity field $k_0(s)$ (right).



(b) True log-transmissivity field $u^\dagger(s)$ (left), and 25 observations on selected locations indicated by circles (right), with color indicating their values.

Figure 6. Elliptic Inverse Problem

defined by the superposition of four weighted Gaussian plumes with standard deviation 0.05, centered at $[0.3, 0.3]$, $[0.7, 0.3]$, $[0.7, 0.7]$, $[0.3, 0.7]$, with weights $\{2, -3, 3, -2\}$ respectively, as shown in the left panel of Figure 6a.

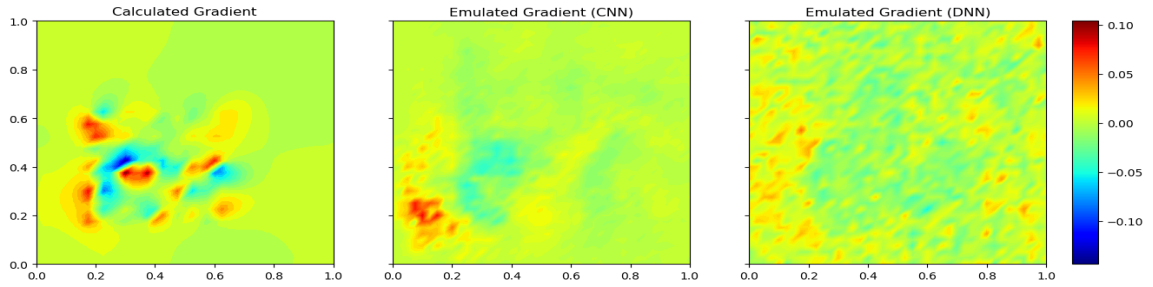
The transmissivity field is endowed with a log-Gaussian prior, i.e.

$$k(s) = \exp(u(s)), \quad u(s) \sim \mathcal{N}(0, \mathcal{C})$$

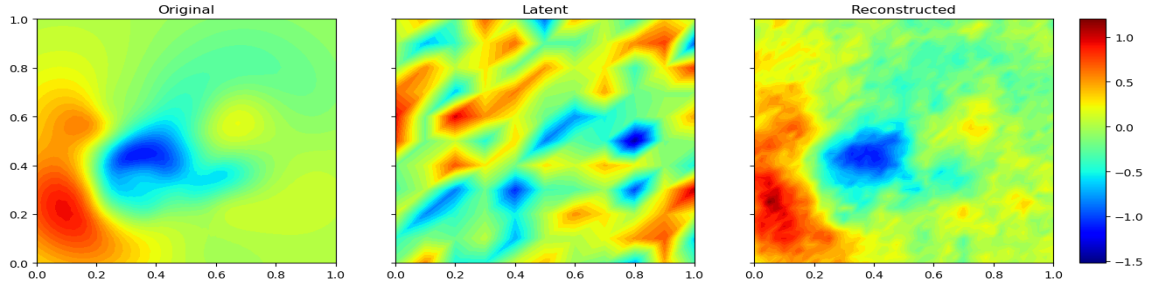
where the covariance operator $\mathcal{C}$ is defined through an exponential kernel function

$$\mathcal{C} : \mathbb{X} \to \mathbb{X},\ u(s) \mapsto \int c(s, s')u(s')ds', \quad c(s, s') = \sigma_u^2 \exp\left(-\frac{\|s - s'\|}{2s_0}\right), \text{ for } s, s' \in \Omega$$

with the prior standard deviation $\sigma_u = 1.25$ and the correlation length $s_0 = 0.0625$. To make the inverse problem more challenging, I follow [33] to use a true log transmissivity



(a) CNN (middle) and DNN (right) emulation ($\mathcal{G}^e : \mathbb{R}^{1681} \to \mathbb{R}^{25}$) extracting gradients $D\Phi^e(u^{\text{MAP}})$ compared with the true gradient $D\Phi(u^{\text{MAP}})$ (left).



(b) AE compressing the original function u^{MAP} (left) into latent space u_r^{MAP} (middle) and reconstructing it in the original space $u^{\text{MAP}'}$ (right).

Figure 7. Elliptic Inverse Problem: Outputs by Neural Networks Viewed as 2d Images

field $u^\dagger(s)$ that is not drawn from the prior, as shown on the left panel of Figure 6b. The right panel of Figure 6a shows the potential function, $p(s)$, solved with $u^\dagger(s)$, which is also used for generating noisy observations. Partial observations are obtained by solving $p(s)$ on an 81×81 mesh and then collecting at 25 measurement sensors as shown by the circles on the right panel of Figure 6b. The corresponding observation operator $\mathcal{O}$ yields the data

$$y = \mathcal{O}p(s) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma_\eta^2 I_{25})$$

where considering the signal-to-noise ratio $\text{SNR} = \max_s\{u(s)\}/\sigma_\eta = 50$ in this example.

The inverse problem involves sampling from the posterior of the log-transmissivity field $u(s)$, which becomes a vector with a dimension of 1681 after being discretized on

41×41 mesh (with Lagrange degree 1). Implement the CES framework described in Section 2.3. In the calibration stage, I collect $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{n=1,j=1}^{N,J}$ from $N = 10$ iterations of EKS runs with ensemble size $J = 500$. For the emulation, DNN and CNN are trained with 75% of these 5000 ensembles and test/validate them with the remaining 25%. The DNN has three layers with 'softplus' activation function for the hidden layers and 'linear' activation for the output layer, and 40% nodes dropped out. The structure of CNN is illustrated in Figure 2 with 'softplus' activation for the convolution layers, 'softmax' activation for the latent layer (dimension 256), and 'linear' activation for the output layer. The trained CNN has a dropout rate of 50% on all its nodes. Figure 7a compares the true gradient function $D\Phi(u^{\text{MAP}})$ (left) and its emulations $D\Phi^e(u_{\text{MAP}})$ (right two) as in Equation (2.10) extracted from two types of neural networks. These gradient functions are plotted on the 2d domain $[0, 1]^2$. We can see that even trained on forward outputs without any gradient information, these extracted gradients from the neural network model provide decent approximations to the true gradient that captures its main graphical feature viewed as a 2d image. The result by CNN is qualitatively better than DNN, which is supported by the numeric evidence of error comparison illustrated in the left panel of Figure 3.

In the sampling stage, I train AE with the structure illustrated in Figure 4. The latent dimension is $d_L = 121$ (11×11) and the node sizes of hidden layers between input and latent, between latent and output, are linearly interpolated. All the activation functions are chosen as 'LeakyReLU($\alpha = 2$)'. Figure 7b plots the original u^{MAP} (left), the latent representation $u_r^{\text{MAP}} = \phi(u^{\text{MAP}})$ (middle) and the reconstruction $u^{\text{MAP}'} = \psi(u_r^{\text{MAP}})$ (right). Even though the latent representation is not intuitive, the output function (image) decoded from the latent space can be viewed as a 'faithful' reconstruction of the original function (image), indicating a sufficiently good AE that

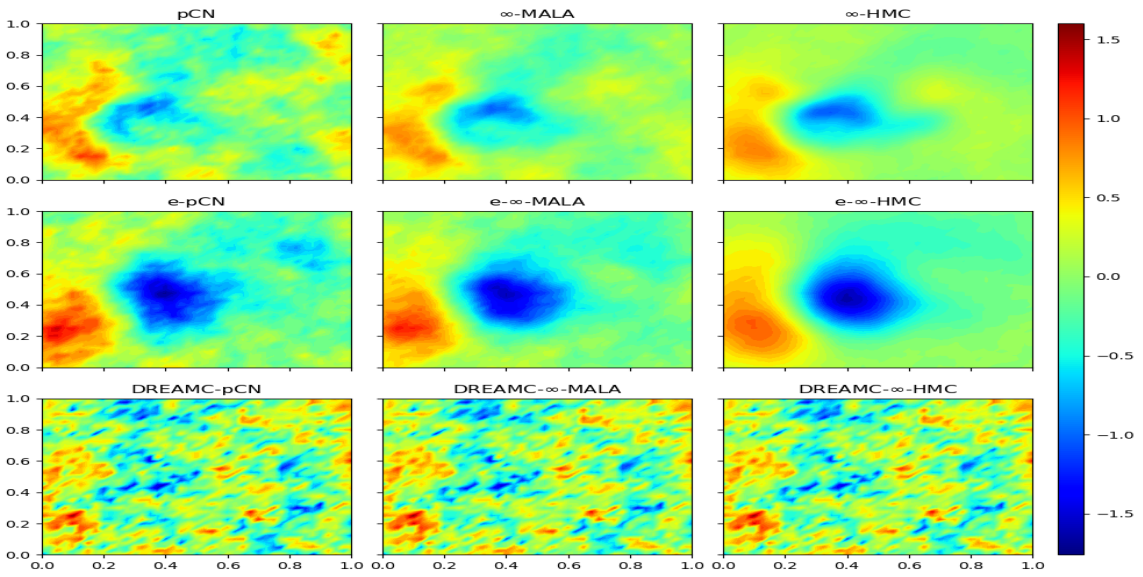


Figure 8. Elliptic inverse problem: Bayesian posterior mean estimates of the log-transmissivity field estimate $u(s)$ based on 5000 samples by various MCMC algorithms.

compresses and restores information. Therefore, the proposed MCMC algorithms, defined on the latent space, generate samples that can be projected back to the original space without losing too much accuracy in representing the posterior distribution.

I compare the performance of algorithms including vanilla pCN, ∞-MALA, ∞-HMC, their emulative versions and corresponding DREAM algorithms. Run 6000 iterations for each algorithm and burn in the first 1000. For HMC algorithms, set $I = 5$. I tune the step sizes for each algorithm so that they have similar acceptance rates around $60 \sim 70\%$. Figure 8 compares their posterior mean estimates and Figure 9 compares their estimates of posterior standard deviation. We can see that emulative MCMC algorithms generate results very close to those by the original MCMC methods. DREAM algorithms introduce more errors due to the information loss in AE but still provide estimates that reasonably resemble those generated by the original MCMC algorithms.

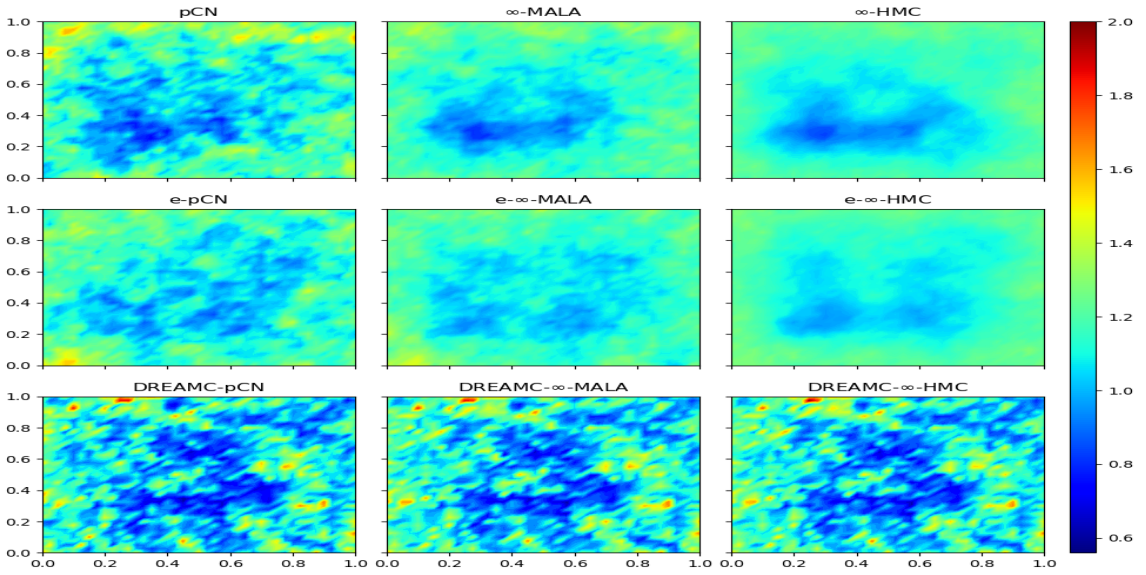


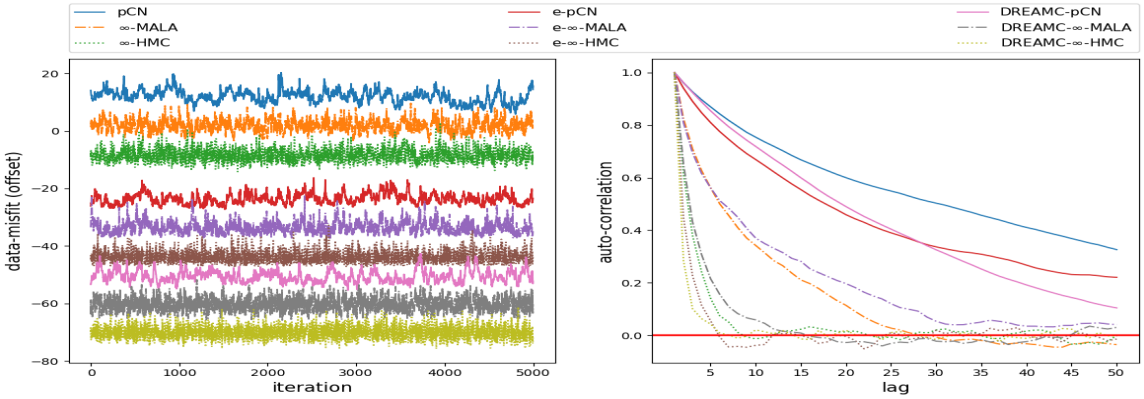
Figure 9. Elliptic inverse problem: Bayesian posterior standard deviation estimates of the log-transmissivity field $u(s)$ based on 5000 samples by various MCMC algorithms.

Method	h [a]	AP [b]	s/iter [c]	ESS(min,med,max) [d]	minESS/s [e]	spdup [f]	PDEsolns [g]
pCN	0.03	0.65	0.49	(7.8,28.93,73.19)	0.0032	1.00	6001
∞-MALA	0.15	0.61	0.56	(29.21,120.79,214.85)	0.0105	3.30	12002
∞-HMC	0.10	0.70	1.65	(547.62,950.63,1411.6)	0.0663	20.82	36210
e-pCN	0.05	0.60	0.02	(10.07,43.9,93.62)	0.0879	27.60	0
e-∞-MALA	0.15	0.67	0.03	(33.23,133.54,227.71)	0.2037	63.95	0
e-∞-HMC	0.10	0.77	0.07	(652.54,1118.08,1455.56)	1.9283	**605.47**	0
DREAM-pCN	0.10	0.67	0.02	(36.78,88.36,141.48)	0.3027	95.03	0
DREAM-∞-MALA	1.00	0.66	0.04	(391.53,782.06,927.08)	2.0988	**659.01**	0
DREAM-∞-HMC	0.60	0.64	0.11	(2289.86,3167.03,3702.4)	4.1720	**1309.97**	0

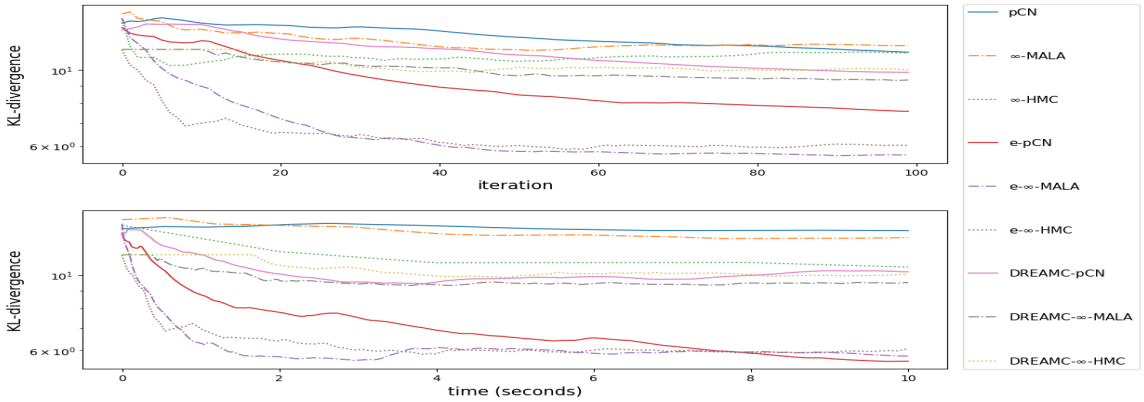
[a] step size [b] acceptance probability [c] seconds per iteration [d] (minimum, median, maximum) effective sample size [e] minimal ESS per second [f] comparison of minESS/s with pCN as benchmark [g] number of PDE solutions

Table 1. Elliptic Inverse Problem: Sampling Efficiency of Various MCMC Algorithms.

Table 1 summarizes the sampling efficiency of various MCMC algorithms measured by minimum effective sample size (ESS) normalized by the total time consumption, i.e., minESS/s. With this standard, emulative ∞-HMC and DREAM ∞-MALA achieve more than 600 times speed-up in sampling efficiency, and DREAM ∞-HMC attains three orders of magnitude improvement compared to the benchmark pCN. Such comparison focuses on the cost of obtaining uncertainty estimates. It does not



(a) The trace plots of data-misfit function evaluated with each sample (left, values have been offset to be better compared with) and the auto-correlation of data-misfits as a function of lag (right).



(b) The KL divergence between the posterior and the prior as a function of iteration (upper) and time (lower).

Figure 10. Elliptic Inverse Problem: Analysis of Posterior Samples

include the time for training CNN and AE, which is relatively much smaller compared with the overall sampling time.

Figure 10a shows the traceplots of the potential function (data-misfit) on the left panel and autocorrelation functions on the right panel. HMC algorithms make distant proposals with the least autocorrelation, followed by MALA algorithms and then pCN algorithms with the highest autocorrelation. This is also supported numerically

by ESS of parameters (the lower autocorrelation, the higher ESS) in Table 1. Note DREAM ∞-MALA has similar autocorrelation as HMC algorithms.

Finally, I plot the Kullback–Leibler (KL) divergence between the posterior and the prior in terms of iteration (upper) and time (lower) in Figure 10b. Among all the MCMC algorithms, emulative MCMC algorithms stabilize such measurements the fastest and attain smaller values for given iterations and time.

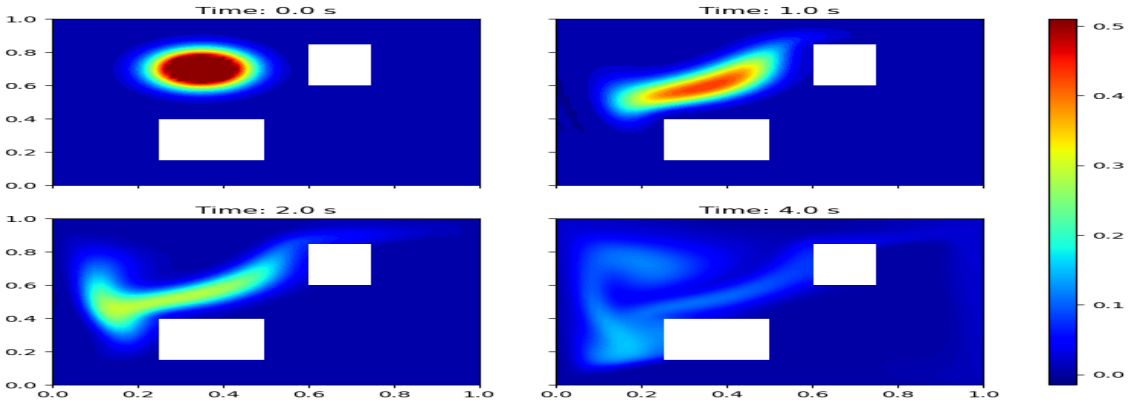
2.4.2 Advection-Diffusion Inverse Problem

In the following example, I quantify the uncertainty in solving an inverse problem governed by a parabolic PDE via the Bayesian inference framework. The underlying PDE is a time-dependent advection-diffusion equation in which I seek to infer an unknown initial condition from spatiotemporal point measurements.

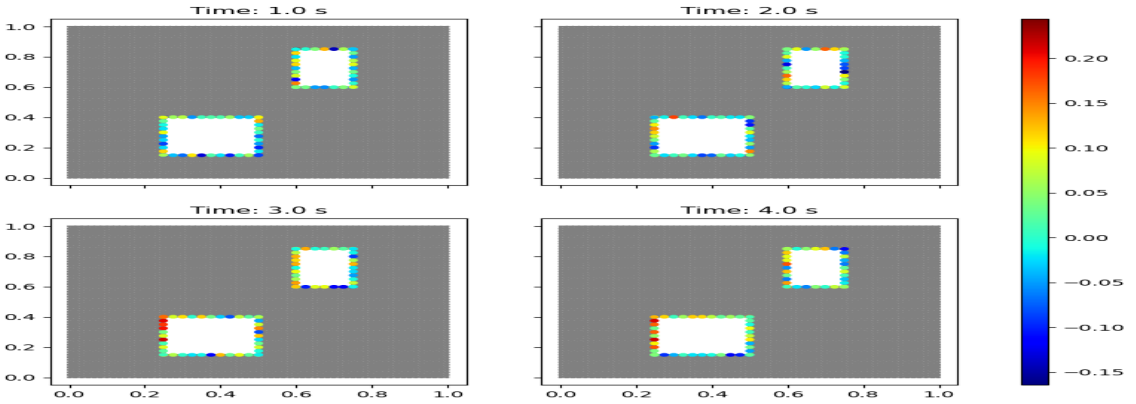
The parameter-to-observable forward mapping $\mathcal{G} : u_0 \to \mathcal{O}u$ maps an initial condition $u_0 \in L^2(\Omega)$ to pointwise spatiotemporal observations of the concentration field $u(\mathbf{x}, t)$ through the solution of the following advection-diffusion equation [123, 154]:

$$\begin{aligned} u_t - \kappa \Delta u + \mathbf{v} \cdot \nabla u = 0 \quad & in\ \Omega \times (0, T) \\ u(\cdot, 0) = u_0 \quad & in\ \Omega \\ \kappa \nabla u \cdot \vec{n} = 0, \quad & on\ \partial\Omega \times (0, T) \end{aligned} \tag{2.24}$$

where $\Omega \subset [0, 1]^2$ is a bounded domain shown in Figure 11a, $\kappa > 0$ is the diffusion coefficient (set to 10^{-3}), and $T > 0$ is the final time. The velocity field $\mathbf{v}$ is computed by solving the following steady-state Navier-Stokes equation with the side walls driving



(a) True initial condition (top left), and the solutions $u(s)$ at different time points.



(b) Spatiotemporal observations at 80 selected locations indicated by circles across different time points, with color indicating their values.

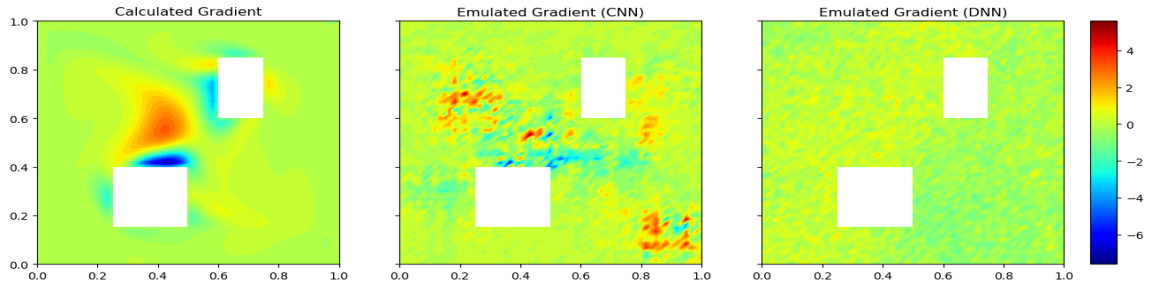
Figure 11. Advection-diffusion Inverse Problem

the flow [123]:

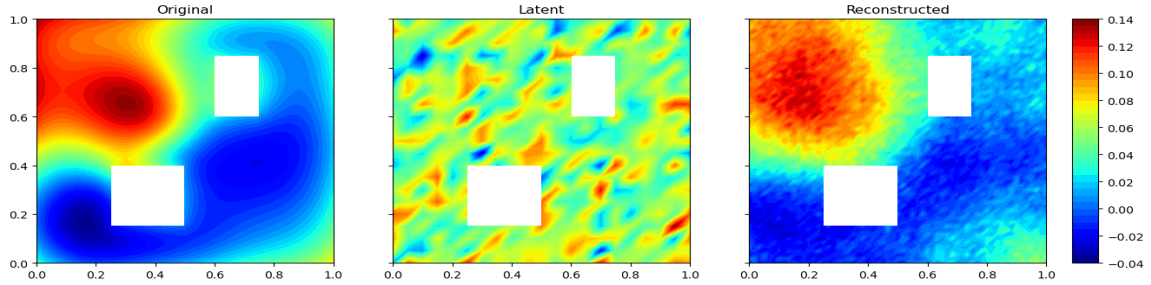
$$\begin{aligned} -\frac{1}{\mathrm{Re}}\Delta \mathbf{v} + \nabla q + \mathbf{v}\cdot\nabla \mathbf{v} &= 0 \quad in\ \Omega \\ \nabla \cdot \mathbf{v} &= 0 \quad in\ \Omega \\ \mathbf{v} &= \mathbf{g}, \quad on\ \partial\Omega \end{aligned} \tag{2.25}$$

Here, q is the pressure, Re is the Reynolds number, which is set to 100 in this example. The Dirichlet boundary data $\mathbf{g} \in \mathbb{R}^d$ is given by $\mathbf{g} = \mathbf{e}_2 = (0, 1)$ on the left wall of the domain, $\mathbf{g} = -\mathbf{e}_2$ on the right wall, and $\mathbf{g} = \mathbf{0}$ everywhere else.

Set the true initial condition $u_0^\dagger = 0.5 \wedge \exp\{-100[(x_1 - 0.35)^2 + (x_2 - 0.7)^2]\}$,



(a) CNN (middle) and DNN (right) emulation ($\mathcal{G}^e : \mathbb{R}^{3413} \to \mathbb{R}^{1280}$) extracting gradients $D\Phi^e(u^{\text{MAP}})$ compared with the true gradient $D\Phi(u^{\text{MAP}})$ (left).



(b) AE compressing the original function u^{MAP} (left) into latent space u_r^{MAP} (middle) and reconstructing it in the original space $u^{\text{MAP}'}$ (right).

Figure 12. Advection-diffusion Inverse Problem: Outputs by Neural Networks Viewed as 2d Images

illustrated in the top left panel of Figure 11a, which also shows a few snapshots of solutions u at other time points on a regular grid mesh of size 61×61. To obtain spatiotemporal observations, I collect solutions $u(\mathbf{x}, t)$ solved on a refined mesh at 80 selected locations across 16 time points evenly distributed between 1 and 3 seconds (thus denoted as $\mathcal{O}u$) and inject some Gaussian noise $\mathcal{N}(0, \sigma_\eta^2)$ such that the relative noise standard deviation is 0.5 ($\sigma_\eta / \max \mathcal{O}u = 0.5$); that is,

$$y = \mathcal{O}u(\mathbf{x}, t) + \eta, \quad \eta \sim \mathcal{N}(0, \sigma_\eta^2 I_{1280})$$

Figure 11b plots four snapshots of these observations at 80 locations along the inner boundary. In the Bayesian setting, I adopt the following Gaussian process prior with the covariance kernel $\mathcal{C}$ defined through the Laplace operator Δ:

$$u \sim \mu_0 = \mathcal{N}(0, \mathcal{C}), \quad \mathcal{C} = (\delta \mathcal{I} - \gamma \Delta)^{-2}$$

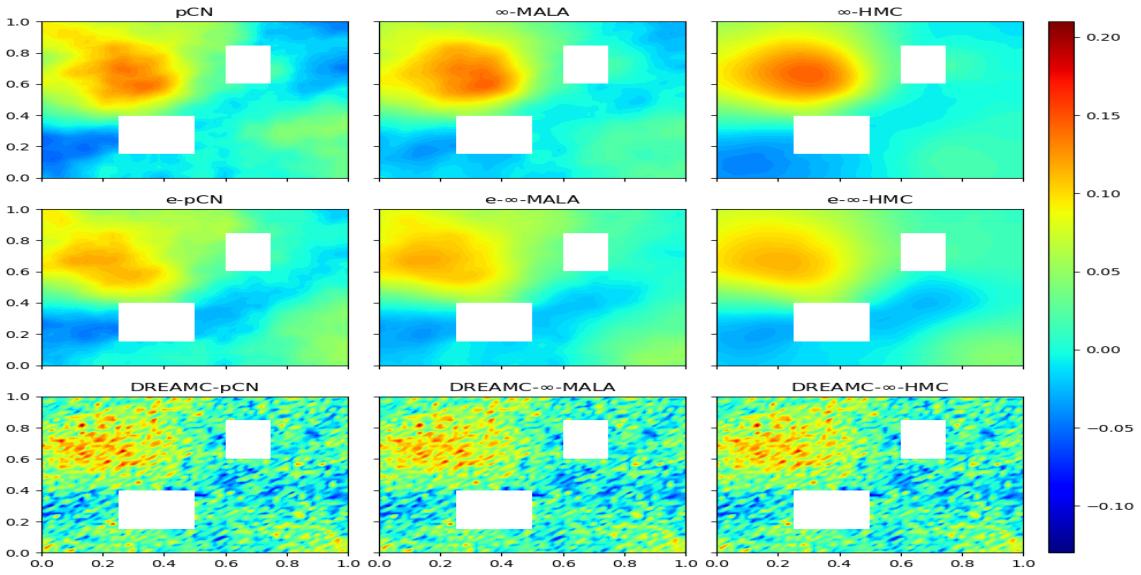


Figure 13. Advection-diffusion inverse problem: Bayesian posterior mean estimates of the initial concentration field $u(\mathbf{x})$ based on 5000 samples by various MCMC algorithms.

where δ governs the variance of the prior and γ/δ controls the correlation length. We set $\gamma = 2$ and $\delta = 10$ in this example.

The Bayesian inverse problem estimates the initial condition u_0 and quantifies its uncertainty based on the 80×16 spatiotemporal observations. For the notational convenience, I still denote $u_0(\mathbf{x})$ as $u(\mathbf{x})$ when it is not confused with the general concentration field $u(\mathbf{x}, t)$. The Bayesian UQ in this example is incredibly challenging not only because of its large dimensionality (3413) of spatially discretized u (Lagrange degree 1) at each time t but also due to the spatiotemporal interactions in these observations. Following the CES framework as in Section 2.3, we collect $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{n=1,j=1}^{N,J}$ by running EKS runs with ensemble size $J = 500$ for $N = 10$ iterations in the calibration stage. For the emulation, I train DNN and CNN with the same $3:1$ splitting of these 5000 ensembles for training/testing data. The DNN

has five layers with the activation function 'LeakyReLU($\alpha = 0.01$)' for the hidden layers and 'linear' activation for the output layer, and 25% nodes dropped out. The structure of CNN is illustrated in Figure 2 with four filters in the last convolution layer, activation 'LeakyReLU($\alpha = 0.2$)' for the convolution layers, 'PReLU' activation for the latent layer (dimension 1024) and 'linear' activation for the output layer. The trained CNN has a dropout rate of 50% on all its nodes. Figure 12a compares the true gradient function $D\Phi(u^{\text{MAP}})$ (left) and its emulations $D\Phi^e(u_{\text{MAP}})$ (right two) as in Equation (2.10) extracted from two types of neural networks. As before, we can see better-extracted gradient output by CNN as an approximation to the true gradient compared with DNN. Due to the large dimensionality of inputs and outputs ($\mathcal{G}^e : \mathbb{R}^{3413} \to \mathbb{R}^{1280}$) and memory requirement, GP failed to fit and output gradient extraction.

In the sampling stage, I adopt AE with the same structure as in Figure 4, the latent dimension $d_L = 417$, and the activation functions chosen as 'elu'. Figure 12b plots the original u^{MAP} (left), the latent representation $u_r^{\text{MAP}} = \phi(u^{\text{MAP}})$ (middle) and the reconstruction $u^{\text{MAP}'} = \psi(u_r^{\text{MAP}})$ (right). Again the autoencoder has successfully reconstructed the original image despite the latent representation being less intuitive.

I compare the performance of ∞-MCMC algorithms (pCN, ∞-MALA, ∞-HMC), their emulative versions and corresponding DREAM algorithms. Run 6000 iterations for each algorithm and burn in the first 1000. For HMC algorithms, set $I = 5$. I tune the step sizes for each algorithm so that they have similar acceptance rates around $60 \sim 70\%$. Figure 13 compares their posterior mean estimates, and Figure 14 compares their estimates of posterior standard deviation. It is evident that emulative MCMC algorithms produce outcomes comparable to those of the original MCMC techniques. DREAM algorithms yield estimates close enough to those by the original

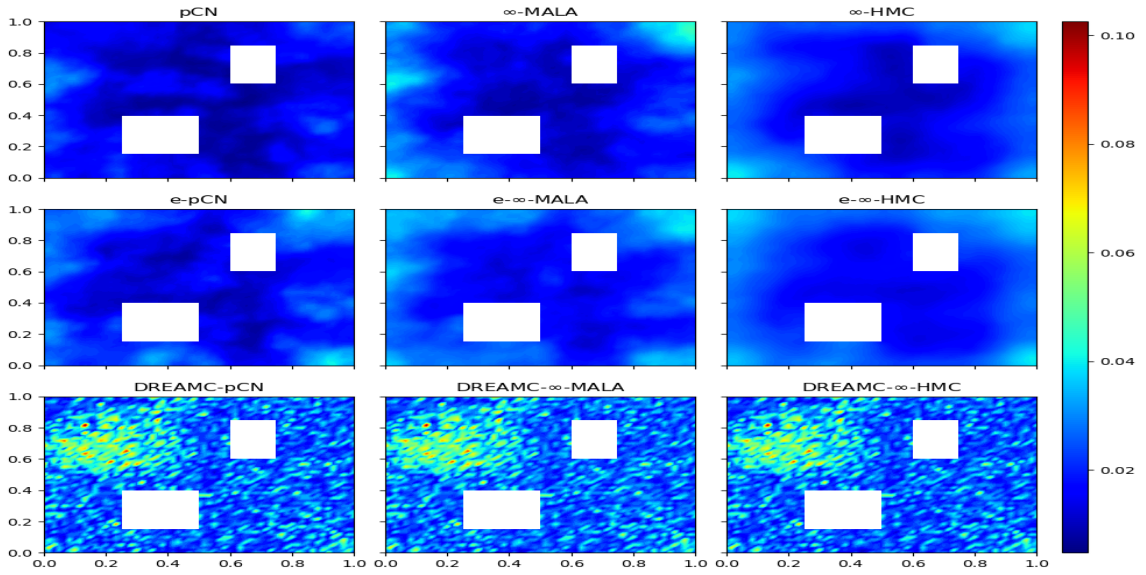


Figure 14. Advection-diffusion inverse problem: Bayesian posterior standard deviation estimates of the initial concentration field $u(\mathbf{x})$ based on 5000 samples by various MCMC algorithms.

MCMC. Although there are some deviations in the uncertainty estimates, the results by DREAM algorithms are significantly better than those by ensemble Kalman methods, which severely underestimate the posterior standard deviations.

Method	h [a]	AP [b]	s/iter [c]	ESS(min,med,max) [d]	minESS/s [e]	spdup [f]	PDEsolns [g]
pCN	0.00	0.69	0.03	(3.16,6.37,40.7)	0.0222	1.00	6001
∞-MALA	0.01	0.68	0.06	(3.78,11.6,51.5)	0.0122	0.55	12002
∞-HMC	0.01	0.78	0.12	(31.55,83.54,240.34)	0.0507	2.29	35872
e-pCN	0.00	0.69	0.02	(3.33,7.19,58.2)	0.0324	1.46	0
e-∞-MALA	0.01	0.72	0.05	(4.28,14.3,62)	0.0157	0.71	0
e-∞-HMC	0.01	0.72	0.11	(25.41,113.11,270.79)	0.0475	2.14	0
DREAM-pCN	0.02	0.68	0.02	(8.88,16.99,53.35)	0.0727	**3**.28	0
DREAM-∞-MALA	0.10	0.83	0.06	(37.65,66.58,157.09)	0.1310	**5**.91	0
DREAM-∞-HMC	0.10	0.72	0.17	(564.12,866.72,1292.11)	0.6791	**30**.64	0

[a] step size [b] acceptance probability [c] seconds per iteration [d] (minimum, median, maximum) effective sample size [e] minimal ESS per second [f] comparison of minESS/s with pCN as benchmark [g] number of PDE solutions

Table 2. Advection-diffusion Inverse Problem: Sampling Efficiency of MCMC Algorithms

Table 2 compares the sampling efficiency of various MCMC algorithms measured by minESS/s. The three most efficient sampling algorithms are all DREAM algorithms.

DREAM ∞-HMC attains up to 30 times faster than the benchmark pCN. This is a significant achievement considering the complexity of this inverse problem with spatiotemporal observations. Again, the training time of CNN and AE is excluded since it is relatively negligible compared with the overall sampling time.

Figure B.1a verifies DREAM ∞-HMC is the most efficient MCMC algorithm with the smallest autocorrelation shown on the right panel. It follows by other HMC algorithms and DREAM ∞-MAMA, which is even better than ∞-HMC. Figure B.1b plots the KL divergence between the posterior and the prior in terms of iteration (upper) and time (lower). From the figure, it appears that ∞-HMC has the fastest convergence.

Chapter 3

SPATIOTEMPORAL LIKELIHOOD MODELING

This chapter is adapted from: "Lan, S., Li, S., &Pasha, M. (2023). Bayesian spatiotemporal modeling for inverse problems. In Statistics and Computing (Vol. 33, Issue 4). Springer Science and Business Media LLC. https://doi.org/10.1007/s11222-023-10253-z".

Spatiotemporal data are ubiquitous nowadays in our daily life. They can be viewed as either multiple time series observed across various locations or geographic data recorded at different time points[91]. Emerging research area arises due to the development and application of novel computational techniques allowing for the analysis of large spatiotemporal databases. Researchers are interested in the intricate relationship between space and time in this data type, e.g., the dynamic brain connectivity study in neuroscience and shipping movements across a geographic area over time.

Statistical models have been popular when dealing with spatiotemporal data due to their flexibility and adaptability. The following section briefly introduces several wildly used statistical models.

3.1 Introduction

From [17], Generalized Linear Model(GLM), Generalized Additive Model(GAM) are two basic models that expand Linear Model(LM) with an additional dimension assuming the independence between observations in space and time. The GLM's

systematic component involves selecting a link function to change the average response, which is then expressed as a linear function of the time and space-related covariates. Based on GLM, GAM assumes a more flexible function which could be parametric(polynomial), semi-parametric, or non-parametric of the covariates rather than identity map in the GLM and keep other properties the same. When assuming independence, it becomes challenging to factor in important information about space and time dependency. This can lead to predictions that are unreliable when dealing with spatiotemporal data that has a high correlation at the same spatial position during adjacent times or between neighboring locations at the same time. To erase the limitation and allow the dependence structure involved, the Hierarchical Spatiotemporal model and STGP are proposed to perform parameter inference when there are dependent errors. The Hierarchical Spatiotemporal model includes at least two stages, while the first stage decomposes the observation into a true (latent) spatiotemporal process and independent error term. What's more, the true process could be modeled in flexible ways:

1. Spatiotemporal fixed effects as a consequence of covariates plus a spatiotemporally dependent random process which is the key part to account for the dependence.
2. A single Spatiotemporal Gaussian Process(STGP) and the covariance of which includes spatiotemporal information.

3.2 Spatiotemporal Gaussian Process(STGP)

My research focuses on STGP to model spatiotemporal data in the inverse problem due to its rich choices of covariance kernel which could be fully parameterized by the

Bayesian approach when selecting hyperparameters in the kernel. Furthermore, it allows uncertainty quantification when performing parameter inference. Therefore, I would like to discuss GP and then move on to STGP.

3.2.1 Gaussian Process(GP)

From [128], GP describes a distribution over functions and inference taking place directly in the functions space, enabling us to work in infinite dimensions.

Definition 1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [128].*

GP is specified by 2 components: mean function $m(\mathbf{x})$ and covariance function $\mathcal{C}_{\mathbf{x}}$, I define them as

$$m(\mathbf{x}) = \mathrm{E}[f(\mathbf{x})]$$
$$\mathcal{C}_{\mathbf{x}} = \mathrm{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$$

and could write GP as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), \mathcal{C}_{\mathbf{x}}) \tag{3.1}$$

where $\mathbf{x}$ refers to a random variable. For the notational simplicity, I will take the mean function to be zero, $\mathcal{C}_{\mathbf{x}}$ then could be modeled in many ways depending on how to formulate $f(\mathbf{x})$. Many covariance function candidates like squared exponential(SE) and Matern could be applied in any scenario to obtain the covariance element $\gamma(\mathbf{x}, \mathbf{x}')$, taking SE as an example:

$$\mathcal{C}_{\mathbf{x}} = cov(\mathbf{x}, \mathbf{x}') = \exp(-\frac{1}{2}|\mathbf{x} - \mathbf{x}'|^2)$$

3.2.2 STGP

After understanding how to construct GP, STGP could be easily obtained by adding another dimension. Suppose t is another time variable and our observation would be $y(\mathbf{x}, t) = f(\mathbf{x}, t)$, then our observations could be modelled with STGP:

$$y(\mathbf{x}, t) \sim \mathcal{GP}(m(\mathbf{x}, t), \Gamma(\mathbf{x}, t)) = \mathcal{GP}(m(\mathbf{x}, t), \mathcal{C}_{\mathbf{x}} \otimes \mathcal{C}_t) \tag{3.2}$$

where $\mathcal{C}_{\mathbf{x}}$ and $\mathcal{C}_t$ are spatial and temporal kernel respectively.

STGP could be applied in any spatiotemporal data like economics, biologic, etc., whereas there is no previous work on implementing STGP on Bayesian inverse problems. Thus, I'd like to introduce how to extend it to Bayesian inverse problem in the following section 3.3 and compare it with the previously used traditional methods like static models(subsection 3.3.1) and time-averaged model(subsection 3.3.2).

3.3 Spatiotemporal Inverse Problems (STIP)

Spatiotemporal modeling was introduced to inverse problems. However, it was either qualitatively applied to specific domains such as functional magnetic resonance imaging (fMRI) [160], electroencephalography (EEG) [143] and electrocardiography (ECG) [141], or to a simplified Gauss-linear problem [105, 117, 30, 163]. Spatiotemporal information was also used to construct prior [168] and regularization [164, 122], or to reduce the number of parameters [42]. However, none formulates the spatiotemporal modeling in the general framework of Bayesian inverse problems with spatiotemporal observations.

When the observations are taken from a spatiotemporal process, $y(\mathbf{x}, t)$, simple

Gaussian likelihood function as (1.2) with $\Gamma = \sigma^2 I$, for example, may not be sufficient to describe the space-time interactions. To address this issue, I propose to rewrite the data model (1.1) in terms of a GP with spatiotemporal kernel $\Gamma(\mathbf{x}, t)$:

$$y(\mathbf{x}, t) = \mathcal{G}(u)(\mathbf{x}, t) + \eta(\mathbf{x}, t), \qquad \eta(\mathbf{x}, t) \sim \mathcal{GP}(0, \Gamma(\mathbf{x}, t)) \tag{3.3}$$

Before delving into the STGP structure for $\Gamma(\mathbf{x}, t)$, I would like to first discuss the traditional methods of the static model (subsection 3.3.1) and the time-averaged model (subsection 3.3.2). These methods are commonly used by researchers when dealing with Bayesian inverse problems.

3.3.1 Static Model

In the literature of Bayesian inverse problems, the noise η is often assumed i.i.d. over time in (3.3), i.e. $\eta(\mathbf{x}, t_j) \overset{iid}{\sim} \mathcal{N}(0, \mathcal{C}_{\mathbf{x}})$. This leads to the following static model where the temporal correlation is ignored:

$$\begin{aligned} y(\mathbf{x}, t)|u, \Gamma &\sim \mathcal{GP}(\mathcal{G}(u)(\mathbf{x}, t), \Gamma(\mathbf{x}, t)) \\ \text{static}: \qquad \Gamma(\mathbf{x}, t) &= \mathcal{C}_{\mathbf{x}} \otimes \mathcal{I}_t \end{aligned} \tag{3.4}$$

where $\mathcal{I}_t$ is the Dirac operator such that $\mathcal{I}_t(t, t') = 1$ only if $t = t'$. When the spatial dependence is also suppressed (as in the advection-diffusion example of Section 3.5.1 and in [154, 94]), it becomes $\mathcal{C}_{\mathbf{x}} = \sigma_\varepsilon^2 \mathcal{I}_{\mathbf{x}}$.

Temporal correlation is disregarded in the static model (3.4). When there is (spatio-)temporal effect in the residual η, the static model (3.4) may be insufficient to account for the spatiotemporal relationships in the data. For illustration, I consider an inverse problem involving advection-diffusion (Section 3.5.1) equation [154, 94] of an evolving concentration field $u(\mathbf{x}, t)$ and seek the solution to the initial condition,

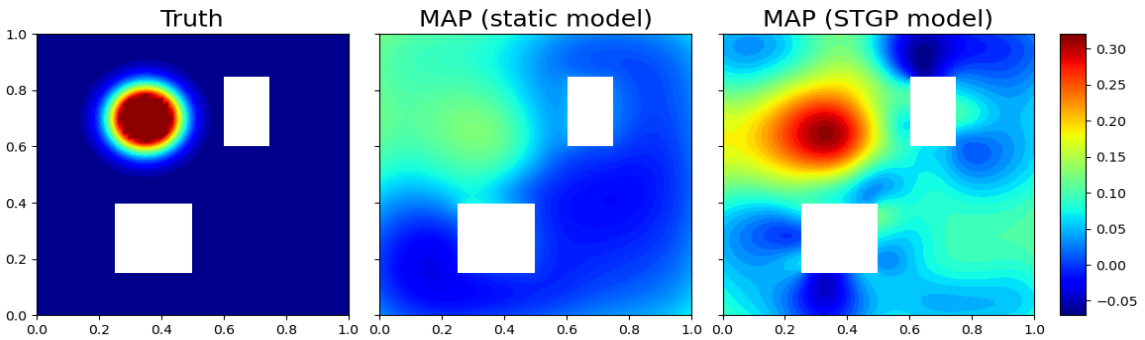


Figure 15. Advection-diffusion inverse problem: comparing maximum *a posteriori* (MAP) estimates of parameter $u_0 = u(\mathbf{x}, 0)$ by the static model (middle) and the STGP model (right) with the truth $u_0^\dagger$ (left).

$u_0 = u(\mathbf{x}, 0)$, based on spatiotemporal solutions observed (through an observation operator $\mathcal{O}$) on the boundaries of two boxes (Figure 15, left panel) for a given time period, i.e. $y = \mathcal{O}u(\mathbf{x}, t) + \eta,\ \eta \sim N(0, \sigma_\eta^2)$. As shown in Figure 15, the simple static model (3.4) used in [94] does not account for space-time interactions hence yields the result underestimating the true function $u_0^\dagger$ (left panel). On the contrary, the estimate by the spatiotemporal model (3.13) (right panel) is much closer to the truth.

3.3.2 Time-averaged model

In many chaotic dynamics, people observe the trajectories as multivariate time series that are very sensitive to the initial condition and the parameters. This usually results in a complex objective function with multiple local minima [2]. They, in turn, form a rough landscape of the objective and pose extreme difficulties on parameter learning [29] (See also Figure 19). The time-averaged approach is commonly used to extract sufficient statistics from the raw data [53].

Consider the same data model as in (3.3) with $\mathcal{G}(u)$ being the observed solution $\mathbf{x}(t; u, \mathbf{x}_0)$ of the following chaotic dynamics (r-th order ODE) for a given parameter

$u \in \mathbb{R}^p$:

$$\dot{\mathbf{x}} := \frac{d\mathbf{x}}{dt} = f(t, \mathbf{x}, \mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(r)}; u), \qquad \mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^I \tag{3.5}$$

That is, $\mathcal{G}(u) = \mathcal{O}\mathbf{x}(t; u, \mathbf{x}_0)$ with an observation operator $\mathcal{O}$. At each time t, the observed vector could include components of $\mathbf{x}$ and up to their k-th order interactions for $k \geq 1$. For example, if $\mathbf{x} = [x_1, \cdots, x_I]$, after including all the first and second-order terms in the observation vector, $\mathcal{O}\mathbf{x} = [x_1, \cdots, x_I, x_1^2, x_1x_2, \cdots, x_ix_j, \cdots, x_I^2]$. Because the trajectories of $\mathcal{G}(u)$ are usually complex, it is often to average them over time and consider the following forward mapping instead:

$$\mathcal{G}_T(u; \mathbf{x}_0) := \frac{1}{T} \int_{t_0}^{t_0+T} \mathcal{O}\mathbf{x}(t; u, \mathbf{x}_0) dt \tag{3.6}$$

where t_0 is the spin-up time, and T is the window length for averaging the observed trajectories of the dynamics.

Following [29], I make the same assumption regarding the dynamical system (3.5):

Assumption 1. *1. For $u \in \mathbb{X}$, (3.5) has a compact attractor $\mathcal{A}$, supporting an invariant measure $\mu(d\boldsymbol{x}; u)$. The system is ergodic, and the following limit of the Law of Large Numbers (LLN) is satisfied: for $\boldsymbol{x}_0 \sim \mu(\cdot; u)$ fixed, with probability one,*

$$\lim_{T\to\infty} \mathcal{G}_T(u; \boldsymbol{x}_0) = \mathcal{G}_\infty(u) := \int_{\mathcal{A}} \mathcal{O}\boldsymbol{x}(t; u, \boldsymbol{x}_0) \mu(d\boldsymbol{x}; u) \tag{3.7}$$

2. The Central Limit Theorem (CLT) holds quantifying the ergodicity: for $\boldsymbol{x}_0 \sim \mu(\cdot; u)$,

$$\mathcal{G}_T(u; \boldsymbol{x}_0) \dot{\sim} \mathcal{N}\left(\mathcal{G}_\infty(u), T^{-1}\Sigma(u)\right) \tag{3.8}$$

The limit $\mathcal{G}_\infty(u)$ becomes independent of the initial condition $\mathbf{x}_0$. However, the finite-time truncation in $\mathcal{G}_T(u; \mathbf{x}_0)$, with different random initializations $\mathbf{x}_0$, generates random errors from the limit $\mathcal{G}_\infty(u)$, which are assumed approximately Gaussian.

Assume the data y can be observed with a true parameter $u^\dagger$, i.e. $y = \mathcal{G}_T(u^\dagger; \mathbf{x}_0)$. The following time-averaged model is usually adopted for the inverse problems involving chaotic dynamics [29]:

$$\begin{aligned} y|u, \Sigma(u) &\sim \mathcal{N}(\mathcal{G}_\infty(u), T^{-1}\Sigma(u)) \\ \text{time-average}: \qquad T^{-1}\Sigma(u) &\approx \Gamma_{\text{obs}} \end{aligned} \tag{3.9}$$

where the empirical covariance Γ_{obs} can be estimated with $\mathcal{G}_\tau(u; \mathbf{x}_0)$ for $\tau \gg T$.

In practice, I replace $\mathcal{G}_\infty(u)$ with $\mathcal{G}_T(u; \mathbf{x}_0)$ in (3.9) and define the potential $\Phi_{\text{T}}(u)$ of parameter u for the time-averaged model (3.9) as follows:

$$\Phi_{\text{T}}(u) = \frac{1}{2}\|y - \mathcal{G}_T(u; \mathbf{x}_0)\|^2_{\Gamma_{\text{obs}}} \tag{3.10}$$

If observing the trajectories (without component interaction terms, i.e., $\mathcal{O}\mathbf{x} = \mathbf{x}$) at discrete time points $\mathbf{t}$ with $t_{J-1} = t_0 + T$, then $\mathcal{O}\mathbf{x}(t; u)$ yields multivariate time series, denoted as $\mathbf{X}(u)_{I\times J} = \mathbf{x}(\mathbf{t}; u) = [\mathbf{x}(t_0; u), \cdots, \mathbf{x}(t_{J-1}; u)]$. Then we have

$$\mathcal{G}_T(u; \mathbf{x}_0) = \overline{\mathbf{X}}(u) := \mathbf{X}(u)\frac{\mathbf{1}_J}{J}, \quad y = \mathbf{X}(u^\dagger)\frac{\mathbf{1}_J}{J}, \quad \Gamma_{\text{obs}} = \mathbf{X}(u^\dagger)\left[\mathbf{I}_J - \frac{\mathbf{1}_J\mathbf{1}_J^\intercal}{J}\right]\mathbf{X}(u^\dagger)^\intercal \tag{3.11}$$

Denote $\mathbf{X}_0 = \mathbf{X}(u) - \mathbf{X}(u^\dagger)$. Therefore the potential Φ_{T} becomes

$$\Phi_{\text{T}}(u) = \frac{1}{2}\frac{\mathbf{1}_J^\intercal}{J}\mathbf{X}_0^\intercal\Gamma_{\text{obs}}^{-1}\mathbf{X}_0\frac{\mathbf{1}_J}{J} = \frac{1}{2}\text{tr}\left[\frac{\mathbf{1}_J\mathbf{1}_J^\intercal}{J^2}\mathbf{X}_0^\intercal\Gamma_{\text{obs}}^{-1}\mathbf{X}_0\right] \tag{3.12}$$

Note averaging the trajectories over time does not ease the difficulty of rough landscapes; refer to Figure 19 for a visual representation. However, the potential function for the following STGP model (3.13) is more convex around the true values $u^\dagger$ compared with the time-averaged approach (3.9).

The aforementioned two approaches, the static model (3.4) and the time-averaged model (3.9), can be recognized as special cases of a more general framework of spatiotemporal modeling based on STGP, to be discussed in the following section.

3.3.3 Spatiotemporal GP model

For the spatiotemporal data $y(\mathbf{x}, t)$ in the inverse problems, I consider the following likelihood model based on STGP:

$$\begin{aligned} y(\mathbf{x}, t)|u, \Gamma &\sim \mathcal{GP}(\mathcal{G}(u)(\mathbf{x}, t), \Gamma(\mathbf{x}, t)) \\ \text{STGP}: \qquad \Gamma(\mathbf{x}, t) &= \mathcal{C}_{\mathbf{x}} \otimes \mathcal{C}_t \end{aligned} \tag{3.13}$$

where $\mathcal{C}_{\mathbf{x}}$ and $\mathcal{C}_t$ are spatial and temporal kernel respectively.

If observing the process $y(\mathbf{x}, t)$ according to (3.13), the resulted data matrix $\mathbf{Y} = \mathcal{G}(u)(\mathbf{X}, \mathbf{t})$ follows the matrix normal distribution (denoted as '$\mathcal{MN}$') [60] for which I can also specify the above-mentioned three models

$$\mathbf{Y}|\mathbf{M}, \mathbf{U}, \mathbf{V} \sim \mathcal{MN}(\mathbf{M}, \mathbf{U}, \mathbf{V}), \quad \mathbf{M} = \mathcal{G}(u^\dagger)(\mathbf{X}, \mathbf{t})$$

$$\text{static}: \qquad \mathbf{U}_{\text{S}} = \sigma_\varepsilon^2 \mathbf{I}_{\mathbf{x}}, \quad \mathbf{V}_{\text{S}} = \mathbf{I}_t \tag{3.14a}$$

$$\text{time-average}: \qquad \mathbf{U}_{\text{T}} = \Gamma_{\text{obs}}, \quad \mathbf{V}_{\text{T}} = J^2(\mathbf{1}_J \mathbf{1}_J^{\text{T}})^- \tag{3.14b}$$

$$\text{STGP}: \qquad \mathbf{U}_{\text{ST}} = \mathbf{C}_{\mathbf{x}}, \quad \mathbf{V}_{\text{ST}} = \mathbf{C}_t \tag{3.14c}$$

where $\mathbf{Y} = \mathcal{O}\mathbf{x}(t; u) = \mathbf{X}(u)$ for the static model and M^- is the pseudo-inverse of M.

In all of the three models mentioned above (3.14), I assume $\mathbf{Y}$ i.i.d. over u's. Denote Φ_* and $\mathcal{I}_*$ as potential function and Fisher information matrix with $*$ being 'S' for the static model (3.14a), "T" for the time-averaged model (3.14b) and "ST" for the STGP model (3.14c) respectively. The following theorem compares the convexity of their likelihoods and indicates that the STGP model (3.14c) with proper configuration has the advantage of parameter learning with the most convex likelihood among the three models.

Theorem 3.3.1. *If we set the maximal eigenvalues of $\boldsymbol{C}_{\boldsymbol{x}}$ and $\boldsymbol{C}_t$ such that $\lambda_{\max}(\boldsymbol{C}_{\boldsymbol{x}})\lambda_{\max}(\boldsymbol{C}_t) \leq \sigma_\varepsilon^2$, then the following inequality holds regarding the Fisher*

information matrices, $\mathcal{I}_S$ and $\mathcal{I}_{ST}$, of the static model and the STGP model respectively:

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_S(u) \tag{3.15}$$

If we control the maximal eigenvalues of $\boldsymbol{C_x}$ and $\boldsymbol{C}_t$ such that $\lambda_{\max}(\boldsymbol{C_x})\lambda_{\max}(\boldsymbol{C}_t) \leq J\lambda_{\min}(\Gamma_{obs})$, then the following inequality holds regarding the Fisher information matrices, $\mathcal{I}_T$ and $\mathcal{I}_{ST}$, of the time-averaged model and the STGP model respectively:

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_T(u) \tag{3.16}$$

Proof. See Appendix A.2.1. □

The following theorem considers a special case, $\mathbf{C_x} = \Gamma_{\text{obs}}$, under milder condition in comparing the likelihood convexity of the time-averaged model and the STGP model.

Theorem 3.3.2. *If we choose $\boldsymbol{C_x} = \Gamma_{obs}$ and require the maximal eigenvalue of $\boldsymbol{C}_t$, $\lambda_{\max}(\boldsymbol{C}_t) \leq J$, then the following inequality holds regarding the Fisher information matrices, $\mathcal{I}_T$ and $\mathcal{I}_{ST}$, of the time-averaged model and the STGP model respectively:*

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_T(u) \tag{3.17}$$

Proof. See Appendix A.2.2. □

Remark 2. *In general, $\Phi_*(u)$ is not the potential of a Gaussian distribution because of the possible non-linearity of $\mathcal{G}(u)$. Theorems 3.3.1 and 3.3.2 indicate that for each $u \in \mathbb{X}$, the STGP model can have a more convex Gaussian proxy in the Laplace approximation.*

Remark 3. *If we view Fisher information as a measurement of (statistical) convexity, the above theorems 3.3.1 and 3.3.2 indicate that the STGP model can have a likelihood*

more convex around the true parameter value than either the static model or the time-averaged model does. This implies that the parameter learning method based on the STGP model could be more effective in the sense that it converges faster.

3.4 Inference

Often our attention is directed towards predicting the underlying process $y(\mathbf{x}, t)$ at future time t_* given the spatiotemporal observations $\mathbf{Y}$. Based on the STGP model (3.13), the following posterior predictive distribution could be used:

$$p(y(\mathbf{x}, t_*)|\mathbf{Y}) = \int p(y(\mathbf{x}, t_*)|u, \mathbf{Y})p(u|\mathbf{Y})du \tag{3.18}$$

Denote the conditional prediction $\mathrm{E}[y(\mathbf{x}, t_*)|u, \mathbf{Y}]$ as

$$\mathcal{G}^*(u)(\mathbf{x}, t_*) = \underbrace{\mathcal{G}(u)(\mathbf{x}, t_*)}_{Physical} + \underbrace{\Gamma_{t_*\mathbf{t}}\Gamma_{\mathbf{tt}}^{-1}(\mathbf{Y} - \mathcal{G}(u)(\mathbf{X}, \mathbf{t}))}_{Statistical} \tag{3.19}$$

Then I predict $y(\mathbf{x}, t_*)$ with the following predicative mean

$$\begin{aligned} \mathrm{E}[y(\mathbf{x}, t_*)|\mathbf{Y}] &= \mathrm{E}_{u|\mathbf{Y}}[\mathrm{E}_{y_*|u,\mathbf{Y}}[y(\mathbf{x}, t_*)]] \\ &= \mathrm{E}_{u|\mathbf{Y}}[\mathcal{G}^*(u)(\mathbf{x}, t_*)] \\ &\approx \overline{\mathcal{G}}(\mathbf{x}, t_*) + \Gamma_{t_*\mathbf{t}}\Gamma_{\mathbf{tt}}^{-1}(\mathbf{Y} - \overline{\mathcal{G}}(\mathbf{X}, \mathbf{t})) \end{aligned} \tag{3.20}$$

where $\overline{\mathcal{G}}(\mathbf{x}, t_*) := \frac{1}{S}\sum_{s=1}^{S}\mathcal{G}(u^{(s)})(\mathbf{x}, t_*)$ with $u^{(s)} \sim p(u|\mathbf{Y})$. The uncertainty can be measured through the law of total conditional variance.

$$\begin{aligned} \mathrm{Var}[y(\mathbf{x}, t_*)|\mathbf{Y}] &= \mathrm{E}_{u|\mathbf{Y}}[\mathrm{Var}_{y_*|u,\mathbf{Y}}[y(\mathbf{x}, t_*)]] + \mathrm{Var}_{u|\mathbf{Y}}[\mathrm{E}_{y_*|u,\mathbf{Y}}[y(\mathbf{x}, t_*)]] \\ &= \Gamma_{t_*t_*} - \Gamma_{t_*\mathbf{t}}\Gamma_{\mathbf{tt}}^{-1}\Gamma_{\mathbf{t}t_*} + \mathrm{Var}_{u|\mathbf{Y}}[\mathcal{G}^*(u)(\mathbf{x}, t_*)] \\ &\approx \Gamma_{t_*t_*} - \Gamma_{t_*\mathbf{t}}\Gamma_{\mathbf{tt}}^{-1}\Gamma_{\mathbf{t}t_*} + s^2_{\mathcal{G}^*}(\mathbf{x}, t_*) \end{aligned} \tag{3.21}$$

where $s^2_{\mathcal{G}^*}(\mathbf{x}, t_*) := \frac{1}{S}\sum_{s=1}^{S}[\mathcal{G}^*(u^{(s)})(\mathbf{x}, t_*) - \overline{\mathcal{G}^*}(\mathbf{x}, t_*)]^2$ with $u^{(s)} \sim p(u|\mathbf{Y})$.

Assume $t_* \notin \mathbf{t}$. In the static model (3.4), when $\Gamma_{t_*\mathbf{t}} = 0$, It is evident that $\mathcal{G}^*(u)(\mathbf{x}, t_*) = \mathcal{G}(u)(\mathbf{x}, t_*)$. This leads to simplified results.

$$\mathrm{E}[y(\mathbf{x}, t_*)|\mathbf{Y}] \approx \overline{\mathcal{G}}(\mathbf{x}, t_*), \qquad \mathrm{Var}[y(\mathbf{x}, t_*)|\mathbf{Y}] \approx \sigma^2_\varepsilon + s^2_{\mathcal{G}}(\mathbf{x}, t_*) \tag{3.22}$$

This may underestimate the uncertainty compared with the more general STGP model (3.13). If only interested in predicting the forward map $\mathcal{G}(u)$ to new time $t = t_*$, similar results would be obtained:

$$\mathrm{E}[\mathcal{G}(u)(\mathbf{x}, t_*)|\mathbf{Y}] \approx \overline{\mathcal{G}}(\mathbf{x}, t_*), \qquad \mathrm{Var}[\mathcal{G}(u)(\mathbf{x}, t_*)|\mathbf{Y}] \approx s^2_{\mathcal{G}}(\mathbf{x}, t_*) \tag{3.23}$$

Note all the above prediction is feasible only if we can solve ODE/PDE systems to time t_*, i.e., the ability to evaluate $\mathcal{G}(u^{(s)})(\mathbf{x}, t)$ at $t = t_*$. If the necessary computer codes are not available, another GP $\mathcal{GP}(0, \Gamma^{\mathcal{G}})$ can be utilized to model $\mathcal{G}(u)(\mathbf{x}, t)$ and make predictions about the forward mapping.

$$\mathcal{G}(u)(\mathbf{x}, t_*)|\mathcal{G}(u)(\mathbf{X}, \mathbf{t}) \sim \mathcal{N}(\Gamma^{\mathcal{G}}_{t_*\mathbf{t}}(\Gamma^{\mathcal{G}}_{\mathbf{tt}})^{-1}\mathcal{G}(u)(\mathbf{X}, \mathbf{t}), \Gamma^{\mathcal{G}}_{t_*t_*} - \Gamma^{\mathcal{G}}_{t_*\mathbf{t}}(\Gamma^{\mathcal{G}}_{\mathbf{tt}})^{-1}\Gamma^{\mathcal{G}}_{\mathbf{t}t_*}) \tag{3.24}$$

3.5 Numerical Experiments

In this section, I demonstrate the numerical advantage of spatiotemporal modeling in parameter estimation and UQ. More specifically, I compare the STGP model (3.13) with the static model (3.4) using an advection-diffusion inverse problem (Section 3.5.1) previously considered in [154, 94] with the static method. Then I compare the STGP model (3.13) with the time-averaged model (3.9) using three chaotic dynamical inverse problems (Section 3.5.2) of which the Lorenz problem (Section 3.5.2.1) was studied by [29] with the time-averaged approach. Numerical evidence is

presented to support that the STGP model (3.13) is preferable to the other two models. All the computer codes are publicly available at https://github.com/lanzithinking/Spatiotemporal-inverse-problem.

3.5.1 Advection-diffusion inverse problem

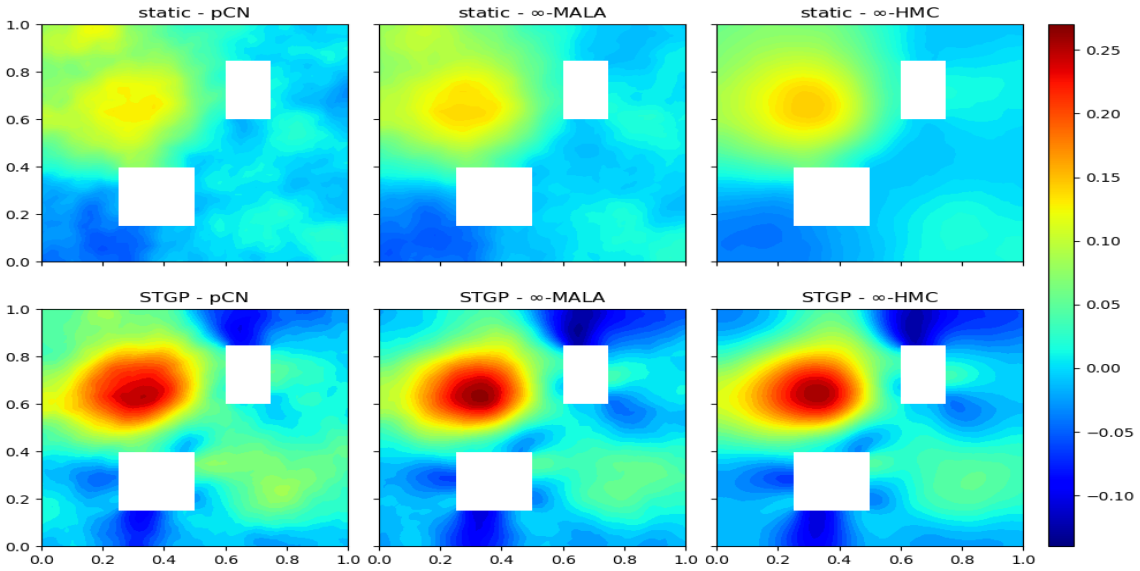
In addition, I work on the Advection-diffusion inverse problem, which was previously discussed in subsection 2.4.2. Also I use the same setup with equation (2.24) and the same initial condition $u_0^\dagger = 0.5 \wedge \exp\{-100[(x-0.35)^2 + (y-0.7)^2]\}$, illustrated in the top left panel of Figure 11a, which also shows a few snapshots of solutions $u(\mathbf{x}, t)$ at other time points on a regular grid mesh of size 61×61.

In the Bayesian setting, I adopt a GP prior for $u_0 \sim \mu_0 = \mathcal{GP}(0, \mathcal{C})$ with the covariance kernel $\mathcal{C} = (\delta\mathcal{I} - \gamma\Delta)^{-2}$ defined through the Laplace operator Δ,

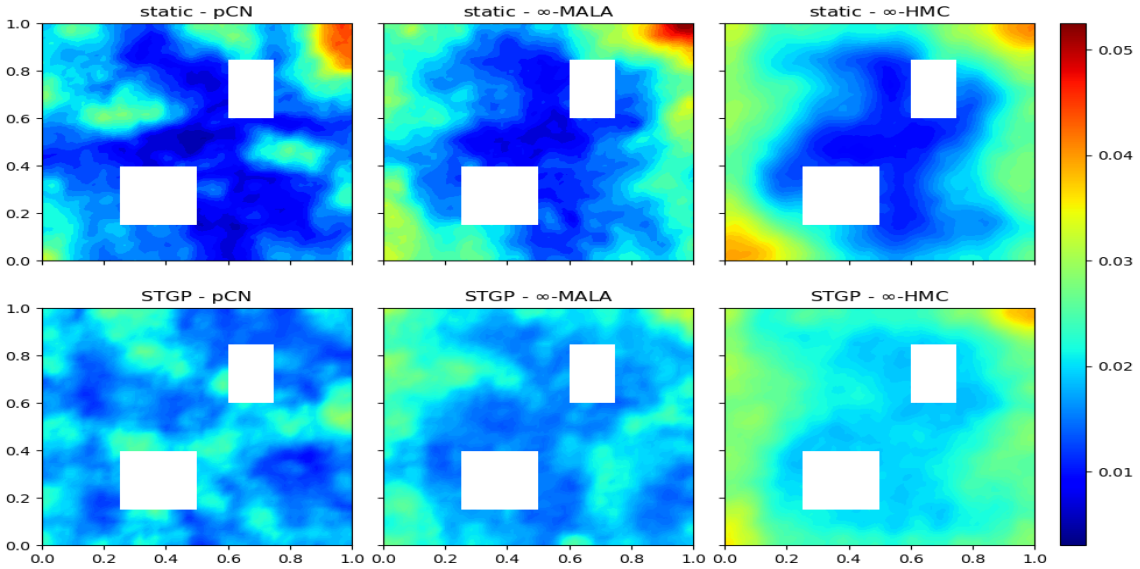
where δ governs the variance of the prior and γ/δ controls the correlation length. We set $\gamma = 2$ and $\delta = 10$ in this example.

The Bayesian inverse problem estimates the initial condition u_0 and quantifies its uncertainty based on the 80×16 spatiotemporal observations. The Bayesian UQ in this example is incredibly challenging not only because of its large dimensionality (3413) of spatially discretized u (Lagrange degree 1) at each time t but also due to the spatiotemporal correlations in these observations.

I compare two likelihood models (3.4) and (3.13). The static model (3.4) is commonly used in the literature of Bayesian inverse problems [90, 154, 94]. Here the STGP model (3.13) is considered to better account for the spatiotemporal relationships in the data. I estimate the variance parameter of the joint kernel from data. The



(a) Posterior mean estimates of the initial concentration field $u_0(\mathbf{x})$.



(b) Posterior standard deviation estimates of the initial concentration field $u_0(\mathbf{x})$.

Figure 16. Advection-diffusion inverse problem: comparing posterior estimates of parameter u_0 in the static model (upper row) and the STGP model (lower row) based on 5000 samples by various MCMC algorithms.

	Estimation			Prediction		
Models	pCN	∞-MALA	∞-HMC	pCN	∞-MALA	∞-HMC
static	0.83 (0.023)	0.81 (0.011)	0.79 (0.005)	0.43 (0.013)	0.4 (0.006)	0.4 (0.003)
STGP	0.74 (0.021)	0.73 (0.012)	0.73 (0.003)	0.44 (0.068)	0.32 (0.016)	0.31 (0.005)

Table 3. Advection-diffusion inverse problem: comparing (i) posterior estimates of parameter u_0 in terms of relative error of mean REM $= \frac{\|\hat{u}_0 - u_0^\dagger\|}{\|u_0^\dagger\|}$ and (ii) the forward predictions $\mathcal{G}(u)(\mathbf{x}, t_*)$ in terms of relative error $\frac{\|\overline{\mathcal{G}}(\mathbf{x},t_*) - \mathcal{G}(u_0^\dagger)(\mathbf{x},t_*)\|}{\|\mathcal{G}(u_0^\dagger)(\mathbf{x},t_*)\|}$ by two likelihood models (static and STGP). Each experiment is repeated for 10 runs of MCMC (pCN, ∞-MALA, and ∞-HMC), and the numbers in the bracket are standard deviations of these repeated experiments.

correlation length parameters are determined ($\ell_{\mathbf{x}} = 0.5$ and $\ell_t = 0.2$) by investigating their autocorrelations as in Figure B.4. Figure 15 compares the maximum a posterior (MAP) of the parameter u_0 by the two likelihood models (right two panels) with the true parameter $u_0^\dagger$ (left panel). The STGP model yields a better MAP estimate closer to the truth than the static model.

I also run MCMC algorithms (pCN, ∞-MALA, and ∞-HMC) to estimate u_0. I run 6000 iterations for each algorithm and burn in the first 1000. The remaining 5000 samples are used to obtain the posterior estimate $\hat{u}_0$ (Figure 16a) and posterior standard deviation (Figure 16b). The STGP model (3.13) consistently generates estimates closer to the true values (refer to Figure 15) with smaller posterior standard deviation than the static model (3.4) using various MCMC algorithms. Such improvement of parameter estimation by the STGP model (3.13) is also verified by smaller relative error of mean estimates REM $= \frac{\|\hat{u}_0 - u_0^\dagger\|}{\|u_0^\dagger\|}$ reported in Table 3, which summarizes the results of 10 repeated experiments with their standard deviations in the brackets.

Finally, I consider the forward prediction (3.23) over the time interval $[0, 5]$. I substitute each of the 5000 samples $\{u^{(s)}\}_{s=1}^{5000}$ generated by ∞-HMC into $\mathcal{G}(u^{(s)})(\mathbf{x}, t_*)$

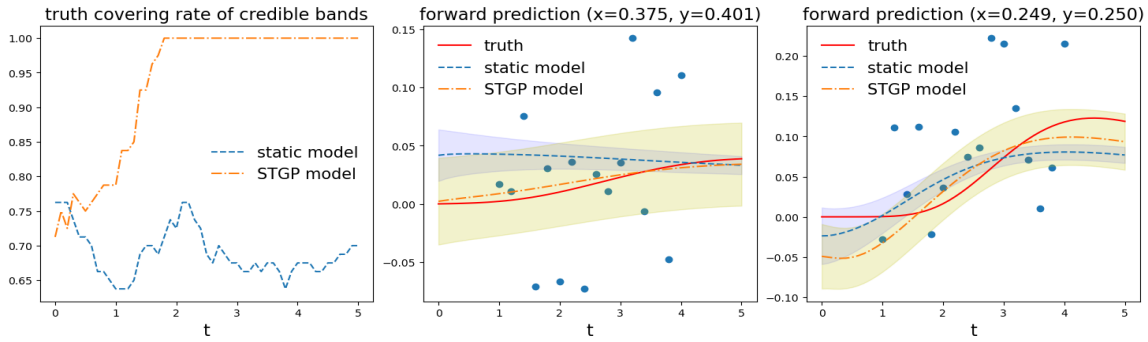


Figure 17. Advection-diffusion inverse problem: comparing forward predictions, $\overline{\mathcal{G}}(\mathbf{x}, t_*)$, based on the static model and the STGP model. The left panel plots the curves representing the percentage of 80 (corresponding to the selected locations) credible bands that cover the true solution $\mathcal{G}(u_0^\dagger)(\mathbf{x}, t_*)$ at each time $t_* \in [0, 5]$. The right two panels show the predicted time series (blue dashed and orange dot-dashed lines) along with the credible bands (shaded regions) by the two models compared with the truth (red solid line) at two selective locations $\mathbf{x} = (0.375, 0.401)$ and $\mathbf{x} = (0.249, 0.250)$. Blue dots are observations.

to solve the advection-diffusion equation (2.24) for $t_* \in [0, 5]$. We observe each of these 5000 solutions at the 80 locations (Figure 11b) for 50 points equally spaced in $[0, 5]$. Then I obtain the prediction by $\overline{\mathcal{G}}(\mathbf{x}, t_*)_{80\times 50} = \frac{1}{5000}\sum_{s=1}^{5000} \mathcal{G}(u^{(s)})(\mathbf{x}, t_*)$, and compute the relative errors in terms of the Frobenius norm of the difference between the prediction and the true solution $\mathcal{G}(u_0^\dagger)(\mathbf{x}, t_*)$: $\frac{\|\overline{\mathcal{G}}(\mathbf{x},t_*) - \mathcal{G}(u_0^\dagger)(\mathbf{x},t_*)\|}{\|\mathcal{G}(u_0^\dagger)(\mathbf{x},t_*)\|}$. Table 3 shows the STGP model (3.13) provides more accurate predictions with smaller errors compared with the static model (3.4). Figure 17 depicts the predicted time series $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ at two selective locations based on the static (blue dashed line) and the STGP (orange dot-dashed line) models along with their credible bands (shaded regions) compared with the truth (red solid lines) in the two right panels. Note that with smaller credible bands, the static model is more certain about its prediction, which is far from the truth. The STGP model provides wider credible bands that cover more of the true trajectories, indicating a more appropriate uncertainty being quantified. Therefore, on the left panel of Figure 17, the STGP model has a higher truth covering rate for its

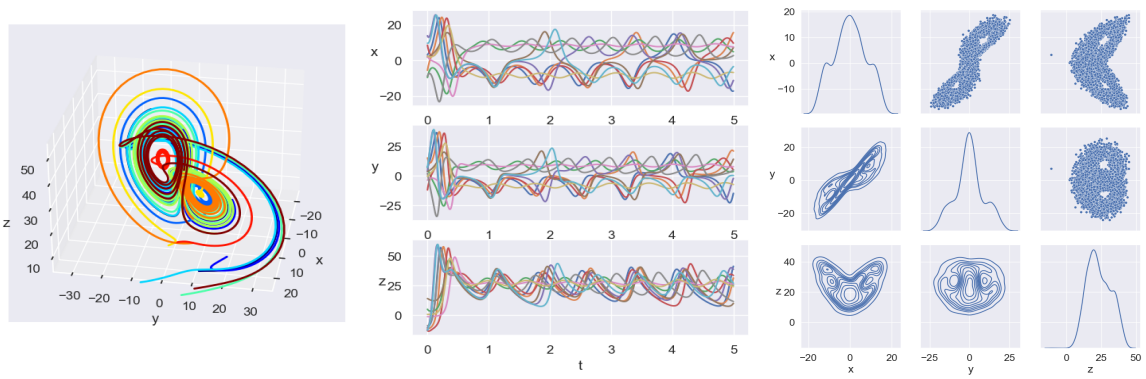


Figure 18. Lorenz63 Dynamics: Two-lobe Orbits (Left), Chaotic Solutions (Middle), and Coordinates' Distributions (Right)

credible intervals among these 80 locations on most of $t_* \in [0, 5]$. Note these models are trained on $t \in [1, 4]$, so the STGP model does not show much advantage initially but quickly outperforms the static model after $t_* = 1$.

3.5.2 Chaotic dynamical inverse problems

Chaos refers to the behavior of a dynamical system that appears to be random in the long term, even if the initial condition entirely determines its evolution. Many physical systems are characterized by the presence of chaos that has been extensively demonstrated [106, 76, 14]. The main challenges of analyzing chaotic dynamical systems include the stability, the transitivity, and the sensitivity to the initial conditions (which contributes to the seeming randomness) [43]. One of the interests in the study of chaotic dynamical systems is determining the essential system parameters given the observed data. In this section, I will investigate three chaotic dynamical systems, Lorenz63 [106], Rössler [4], and Chen [165], that can be summarized as the first-order ODE: $\dot{\mathbf{x}} = f(\mathbf{x}; u)$. I will apply the CES framework (Section 2.2) to learn the system

parameter u and quantify its associated uncertainty based on the observed trajectories. I find the spatiotemporal models numerically more advantageous by fitting the whole trajectories than the common approach by averaging the trajectories over time [137, 29, 72].

3.5.2.1 Lorenz system

The most popular example of chaotic dynamics is the Lorenz63 system [106], which represents a simplified model of atmospheric convection for the chaotic behavior of the weather. The following ODE gives the governing equations of the Lorenz system

$$\begin{cases} \dot{x} &= \sigma(y - x), \\ \dot{y} &= x(\rho - z) - y, \\ \dot{z} &= xy - \beta z, \end{cases} \tag{3.25}$$

where x, y, and z denote variables proportional to convective intensity, horizontal and vertical temperature differences and $u := (\sigma, \rho, \beta)$ represents the model parameters known as Prandtl number (σ), Rayleigh number (ρ), and an unnamed parameter (β) used for physical proportions of the regions [120].

The behavior of Lorenz63 system (3.25) strongly relies on the parameters. In many studies, the parameter ρ varies in $(0, \infty)$ and the other parameters σ and β are held constant. In particular, (3.25) has a stable equilibrium point at the origin for $\rho \in (0, 1)$. For $\rho \in (1, \gamma)$ with $\gamma = \sigma\frac{\sigma+\beta+3}{\sigma-\beta-1}$, (3.25) has three equilibrium points, one unstable equilibrium point at the origin and two stable equilibrium points at $(\sqrt{\beta(\rho-1)}, \sqrt{\beta(\rho-1)}, \rho-1)^\mathsf{T}$ and $(-\sqrt{\beta(\rho-1)}, -\sqrt{\beta(\rho-1)}, \rho-1)^\mathsf{T}$. When $\rho > \gamma$, the equilibrium points become unstable, resulting in erratic spiral-shaped

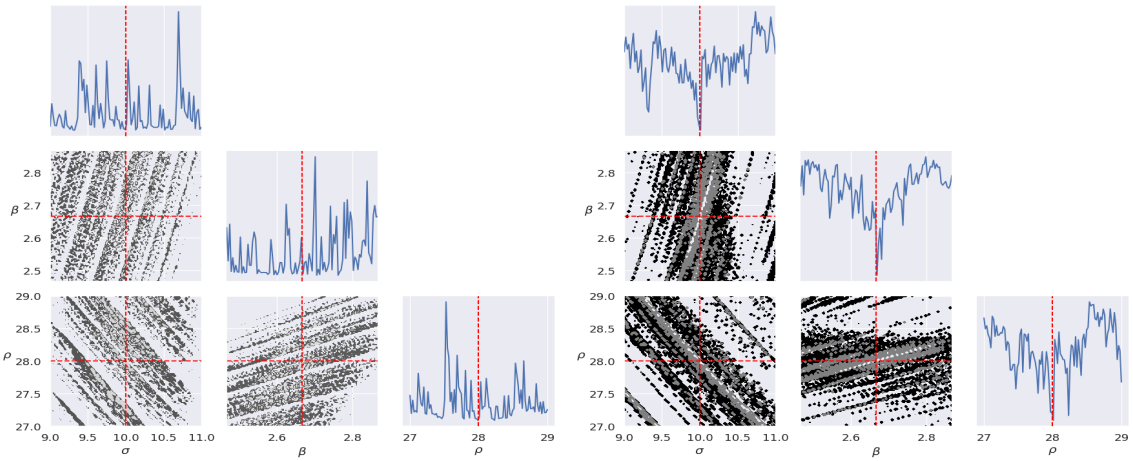


Figure 19. Lorenz inverse problem: marginal (diagonal) and pairwise (lower triangle) sections of the joint density $p(u)$ by the time-averaged model (left) and the STGP model (right) respectively.

trajectories. One classical configuration for the parameter in (3.25) is $\sigma = 10$, $\beta = \frac{8}{3}$, $\rho = 28$ when the system exhibits two-lobe orbits, also known as the butterfly effect [162] (See the left panel of Figure 18). In this example, I seek to infer such parameter $u^\dagger = (\sigma^\dagger, \beta^\dagger, \rho^\dagger) = (10, 8/3, 28)$ based on the observed chaotic trajectories demonstrated in the middle panel of Figure 18. Note the solutions $(x(t), y(t), z(t))$ highly depend on the initial conditions $(x(0), y(0), z(0))$, we hence fix $(x(0), y(0), z(0))$ in the following.

Due to the chaotic nature of the states $\{(x(t), y(t), z(t)) : t \in [0, \tau]\}$, I can treat these coordinates as random variables. In the right panel of Figure 18, I demonstrate their marginal and pairwise distributions (diagonal and lower triangle) estimated by a collection of states (upper triangle) along a long-time trajectory solved with $u^\dagger$. For a given parameter $u = (\sigma, \beta, \rho)$, The trajectory $\mathcal{G}(u)$ is represented by the following map.:

$$\mathcal{G}(u) : \mathbb{R}_+ \to \mathbb{R}^3, \quad t \mapsto (x(t; u), y(t; u), z(t; u)) \tag{3.26}$$

where $(x(t; u), y(t; u), z(t; u))$ is the solution of (3.25) for given parameter u. I generate

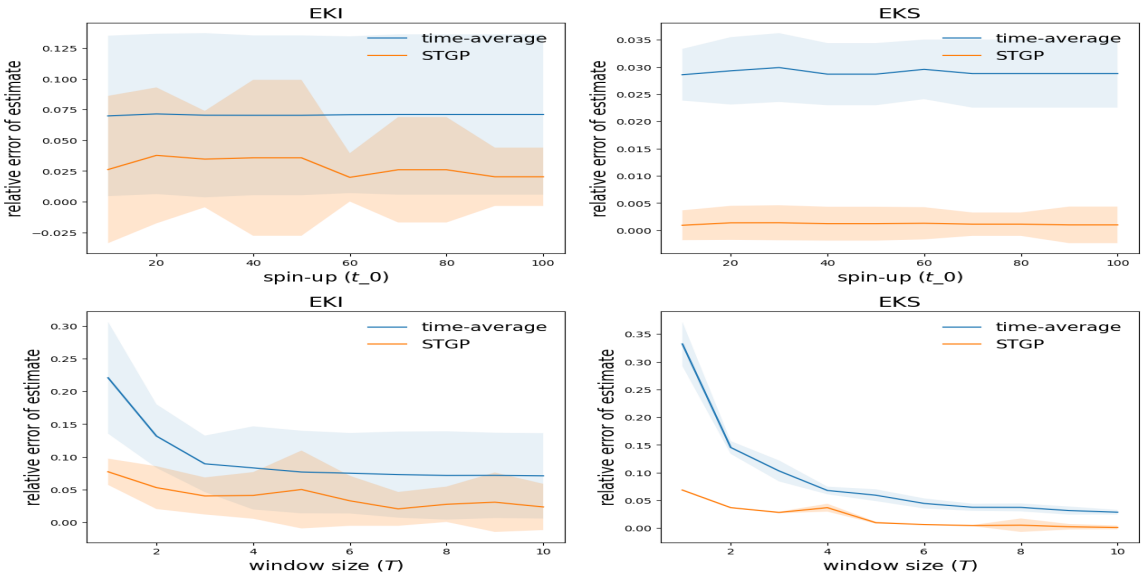


Figure 20. Lorenz inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of mean REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. The upper row shows the results by varying the spin-up t_0 and fixing $T = 10$. The lower row shows the results by varying the observation window size T and fixing $t_0 = 100$. Each experiment is repeated for 10 runs of EnK (EKI and EKS) with $J = 500$ ensembles, and the shaded regions indicate standard deviations of such repeated experiments.

spatiotemporal data from the chaotic dynamics (3.25) with $u^\dagger = (\sigma^\dagger, \beta^\dagger, \rho^\dagger)$ by observing its trajectory on $J = 100$ equally spaced time points $t_j \in [t_0, t_0 + T]$: $\mathbf{X}(u^\dagger)_{3\times 100} := \{\mathcal{G}(u^\dagger)(t_j) = (x(t_j; u^\dagger), y(t_j; u^\dagger), z(t_j; u^\dagger))\}_{j=1}^J$. These observations can be viewed as a 3-dimensional time series that estimates the empirical covariance Γ_{obs} as in [29]. The inverse problem involves learning the parameter u given these observations, also known as parameter identification [116].

Following [29], a log-Normal prior is endow on u: $\log u \sim \mathcal{N}(\mu_0, \sigma_0^2)$ with $\mu_0 = (2.0, 1.2, 3.3)$ and $\sigma_0 = (0.2, 0.5, 0.15)$. I compare the two likelihood models (3.9) and (3.13) for this dynamical inverse problem. For the time-averaged model (3.9), instead of the 3-dimensional time series from the trajectory (3.26), I substitute $\overline{\mathbf{X}}(u)_{3\times 1}$ with

$\overline{\mathbf{X}^\star}(u)_{9\times 1} = \mathcal{O}\mathcal{G}^\star(u)(t)$ by averaging the following augmented trajectory $\mathcal{G}^\star(u)(t)$ in time [29]:

$$\mathcal{G}^\star(u)(t) = (x(t), y(t), z(t), x^2(t), y^2(t), z^2(t), x(t)y(t), x(z)z(t), y(t)z(t))$$

For the spatiotemporal likelihood model STGP (3.13), set the correlation length $\ell_{\mathbf{x}} = 0.4$ and $\ell_t = 0.1$ for the spatial kernel $\mathcal{C}_{\mathbf{x}}$ and the temporal kernel $\mathcal{C}_t$ respectively. They are chosen to reflect the spatial and temporal resolutions.

It is important to note that spatiotemporal modeling can help in learning the true parameter, denoted as $u^\dagger$. As illustrated in Figure 19, despite of the rough landscape, the marginal (e.g. $p(\sigma, \beta^\dagger, \rho^\dagger)$) and pairwise (e.g. $p(\sigma, \beta, \rho^\dagger)$) sections of the joint density $p(u)$ by the STGP model (3.13) are more convex in the neighbourhood of $u^\dagger$ compared with the time-averaged model (3.9). This verifies the implication of Theorem 3.3.2 on their difference in convexity. Therefore, particle-based algorithms such as EnK methods have a higher chance of concentrating their ensemble particles around the true parameter value $u^\dagger$, leading to better estimates. Here, the roughness of the posterior creates a barrier to the direct application of MCMC algorithms. Therefore, I apply more robust EnK methods for parameter estimation.

I run each EnK algorithm for $N = 50$ iterations and choose the ensembles (of size J) when its ensemble mean attains the minimal error in estimating the parameter u with reference to its true value $u^\dagger$. In practice, EnK algorithms usually converge quickly within a few iterations, so $N = 50$ suffices the need for most applications.

To investigate the roles of spin-up length t_0 and observation window size T, I run EnK multiple times while varying each of the two quantities one at a time. Seen from Figure 20, observations indicate a noticeable reduction in errors when utilizing the STGP model (3.13) as opposed to the time-averaged model (3.9). More specifically, the upper row indicates that the estimation errors, measured by REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$, are

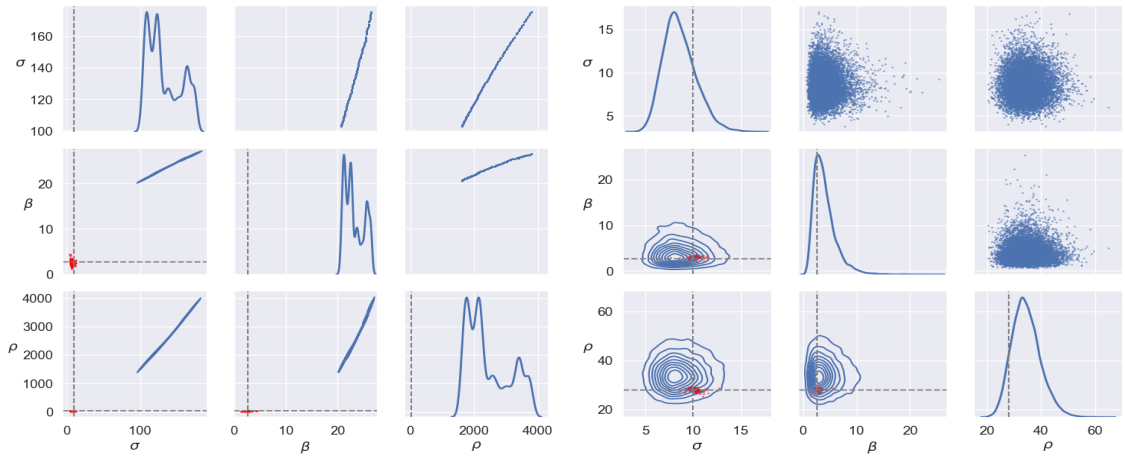


Figure 21. Lorenz inverse problem: marginal (diagonal) and pairwise (lower triangle) distributions estimated with 10000 samples (upper triangle) by the pCN algorithm based on NN emulators for the time-averaged model (left) and the STGP model (right) respectively. Red dots (lower triangle) are selective 10000 ensemble particles from running the EKS algorithm.

not very sensitive to the spin-up t_0 given sufficient window size $T = 10$. On the other hand, for fixed spin-up $t_0 = 100$, both models decrease errors with increasing window size T as they aggregate more information. However, the STGP model requires only about $\frac{1}{4}$ time length as the time-averaged model to attain accuracy at the same level ($T = 1$ vs $T = 4$). This supports that the STGP is preferable to the time-average approach as the former may add a small overhead for the statistical inference but could save much more in resolving the physics (solving ODE/PDE), which is usually more expensive.

Set spin-up $t_0 = 100$ long enough to ignore the effect of the initial condition in the dynamics and choose the observation window size $T = 10$. Compare the two models (3.9) (3.13) using EnK algorithms with different ensemble sizes (J) to obtain an estimate $\hat{u}$ of the parameter u. Figure B.5 shows that the STGP model performs better than the time-averaged model in generating more minor errors (REM)

Model-Algorithms	J=50	J=100	J=200	J=500	J=1000
Tavg-EKI	0.06 (0.03)	0.09 (0.03)	0.09 (0.01)	0.06 (0.04)	0.07 (0.02)
Tavg-EKS	0.10 (0.02)	0.07 (4.62e-03)	0.05 (2.60e-03)	0.03 (3.04e-03)	0.03 (8.56e-04)
STGP-EKI	0.07 (0.03)	0.04 (0.03)	0.03 (0.02)	0.02 (0.03)	0.02 (0.01)
STGP-EKS	0.09 (0.03)	0.05 (0.03)	0.03 (0.02)	3.97e-04 (1.06e-03)	5.52e-04 (6.37e-04)

Table 4. Lorenz inverse problem: comparing posterior estimates of parameter u for two models, time-average (Tavg) and STGP, in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS), and the numbers in the bracket are standard deviations of such repeated experiments.

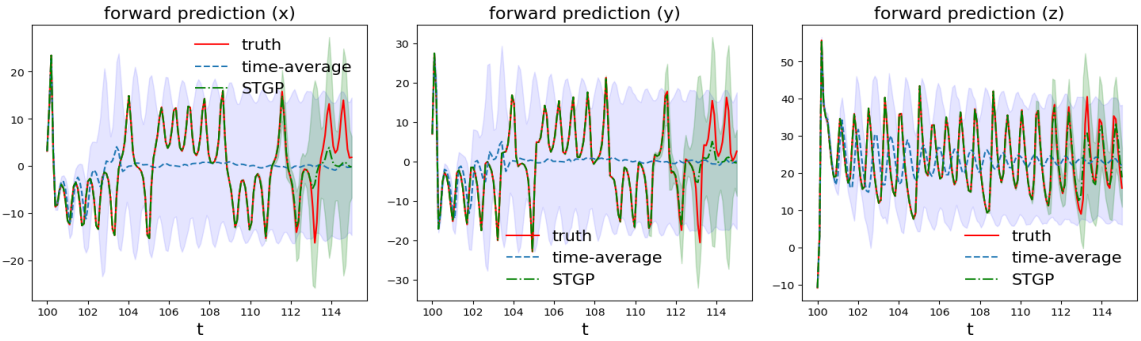


Figure 22. Lorenz Inverse Problem: Comparing Forward Predictions $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ Based on the Time-averaged Model and the STGP Model

for almost all cases. In general, more ensembles help reduce errors, except for the time-averaged model using the EKI algorithm. Note the STGP model with the EKS algorithm yields parameter estimates with the lowest errors. Table 4 summarizes the REM's by different combinations of the two likelihood models (time-averaged and STGP) and two EnK algorithms (EKI and EKS). Once again, it is apparent that the spatiotemporal likelihood model STGP (as shown in equation (3.13)) outperforms the basic time-averaged model (as shown in equation (3.9)) when it comes to achieving precise parameter estimation.

Next, I apply CES (Section 2.2) [29, 94] to quantify the uncertainty of the estimate $\hat{u}$. Direct application of MCMC suffers from an extremely low acceptance rate because of the rough density landscape (Figure 19). Ensemble particles from the EnK algorithm

cannot provide rigorous systematic UQ due to the ensemble collapse [135, 136, 159, 24] (See red dots in Figure 21). Therefore, I run approximate MCMC based on NN emulators built from EnK outputs $\{u_n^{(j)}, \mathcal{G}(u_n^{(j)})\}_{j=1,n=0}^{J,N}$. Note, structures are different for the observed data in the two models (3.9) (3.13): 9-dimensional summary of time series for the time-averaged model (3.9) and 3×100 time series for the STGP model (3.13). Therefore I build densely connected NN (DNN) $\mathcal{G}^e : \mathbb{R}^3 \to \mathbb{R}^9$ for the former and DNN-RNN (recurrent NN) type of network $\mathcal{G}^e : \mathbb{R}^3 \to \mathbb{R}^{3\times 100}$ for the latter to account for their different data structures in the forward output. Figure 21 compares the marginal (diagonal) and pairwise (lower triangle) posterior densities of u estimated by 10000 samples (upper triangle) of the pCN algorithm based on the corresponding NN emulators for the two models. The spatiotemporal model STGP (3.13) yields more reasonable UQ results than the time-averaged model (3.9).

Finally, I consider the forward prediction $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ (3.23) for $t_* \in [t_0, t_0 + 1.5T]$ with $J = 500$ EKS ensembles corresponding to the lowest error. Figure 22 compares the prediction results of these two models. The result by the STGP model is very close to the truth till $t = 113$, while the prediction by the time-averaged model quickly departs from the truth only after $t = 102$. The STGP model predicts the future of the challenging chaotic dynamics significantly better than the time-averaged model.

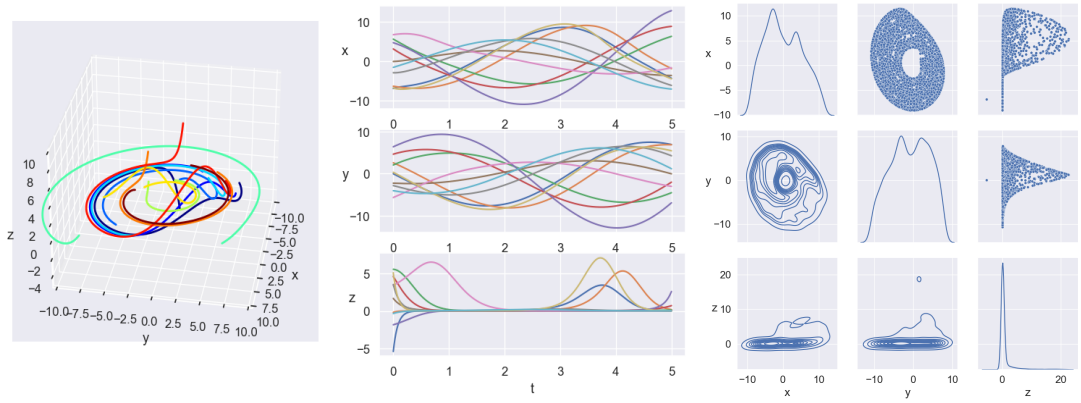


Figure 23. Rössler dynamics: single-lobe orbits (left), chaotic solutions (middle) and coordinates' distributions (right).

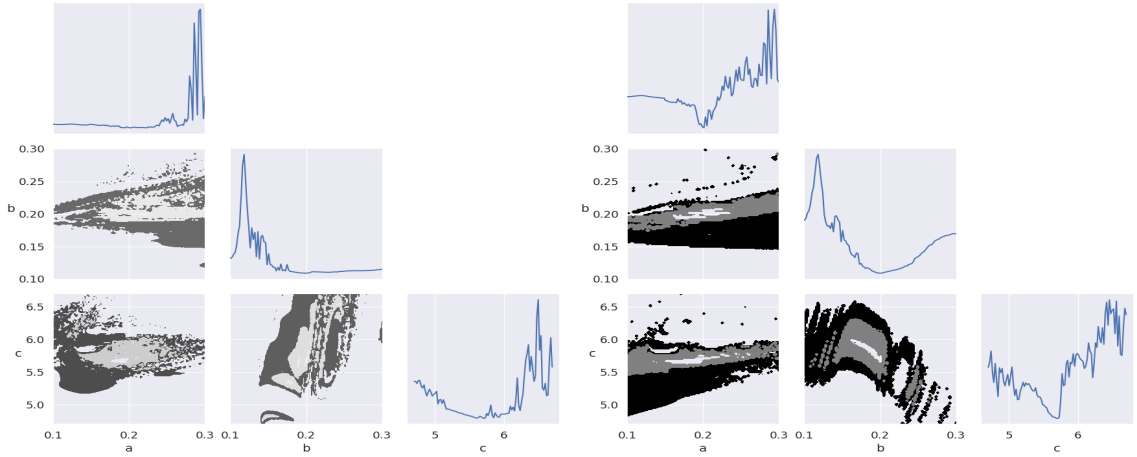


Figure 24. Rössler inverse problem: marginal (diagonal) and pairwise (lower triangle) sections of the joint density $p(u)$ by the time-averaged model (left) and the STGP model (right) respectively.

3.5.2.2 Rössler system

Next, I consider the following Rössler system [64] governed by the system of autonomous differential equations:

$$\begin{cases} \dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c). \end{cases} \tag{3.27}$$

where $a, b, c > 0$ are parameters determining the system's behavior. The Rössler attractor was initially discovered by German biochemist Otto Eberhard Rössler [133, 132]. When $c^2 > 4ab$, the system (3.27) exhibits continuous-time chaos and has two unstable equilibrium points $(a\gamma_-, -\gamma_-, \gamma_-)$ and $(a\gamma_+, -\gamma_+, \gamma_+)$ with $\gamma_+ = \frac{c+\sqrt{c^2-4ab}}{2a}$, $\gamma_- = \frac{c-\sqrt{c^2-4ab}}{2a}$. Note that the Rössler attractor has similarities to the Lorenz attractor. Nevertheless, it has a single lobe and offers more flexibility in qualitative analysis. The true parameter trying to infer is $u^\dagger = (a^\dagger, b^\dagger, c^\dagger) = (0.2, 0.2, 5.7)$. Figure 23 illustrates the single-lobe orbits (left), the chaotic solutions (middle), and their marginal and pairwise distributions (right) of their coordinates viewed as random variables.

Note, the Rössler dynamics evolve at a lower rate compared with the Lorenz63 dynamics (compare the middle panels of Figure 23 and Figure 18). Therefore, I adopt a longer spin-up length ($t_0 = 1000$) and a larger window size ($T = 100$). For the STGP model (3.13), spatiotemporal data are generated by observing the trajectory (3.26) of the chaotic dynamics (3.27) with $u^\dagger = (a^\dagger, b^\dagger, c^\dagger)$ for $J = 100$ time points in $[t_0, t_0 + T]$. I also augment the time-averaged data with second-order moments for the time-averaged model (3.9). In this Bayesian inverse problem, a

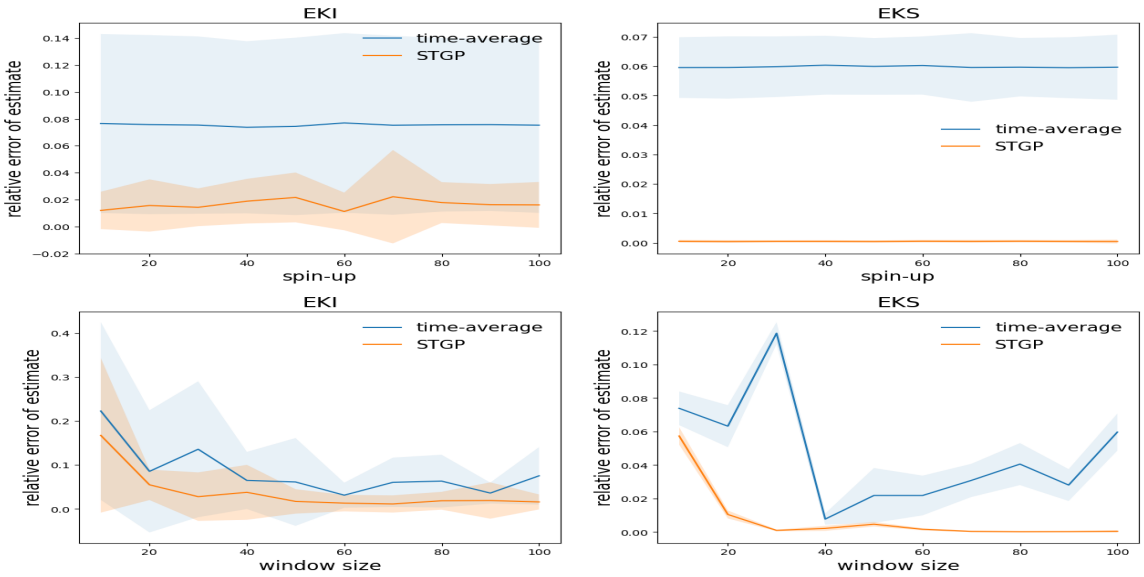


Figure 25. Rössler inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of mean REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. The upper row shows the results by varying the spin-up t_0 and fixing $T = 100$. The lower row shows the results by varying the window size T and fixing $t_0 = 100$. Each experiment is repeated for 10 runs of EnK (EKI and EKS) with $J = 500$ ensembles, and the shaded regions indicate standard deviations of such repeated experiments.

log-Normal prior is adopted on u: $\log u \sim \mathcal{N}(\mu_0, \sigma_0^2)$ with $\mu_0 = (-1.5, -1.5, 2.0)$ and $\sigma_0 = (0.15, 0.15, 0.2)$. Once again, with spatiotemporal likelihood model STGP (3.13), learning the true parameter value $u^\dagger$ becomes more accessible because the posterior density $p(u)$ concentrates more on $u^\dagger$ compared with the time-averaged model (3.9), as indicated by Theorem 3.3.2. See Figure 24 for comparing their marginal and pairwise sections of the joint density $p(u)$.

I also compare the two models (3.9) (3.13) when investigating the roles of spin-up length t_0 and observation window size T in Figure 25. Despite the consistently smaller errors (expressed in terms of REM) by the STGP model, REM is not very sensitive to the spin-up t_0 given sufficient window size $T = 100$. However, for fixed spin-up $t_0 = 100$, the STGP model (3.13) is superior to the time-averaged approach (3.9) in

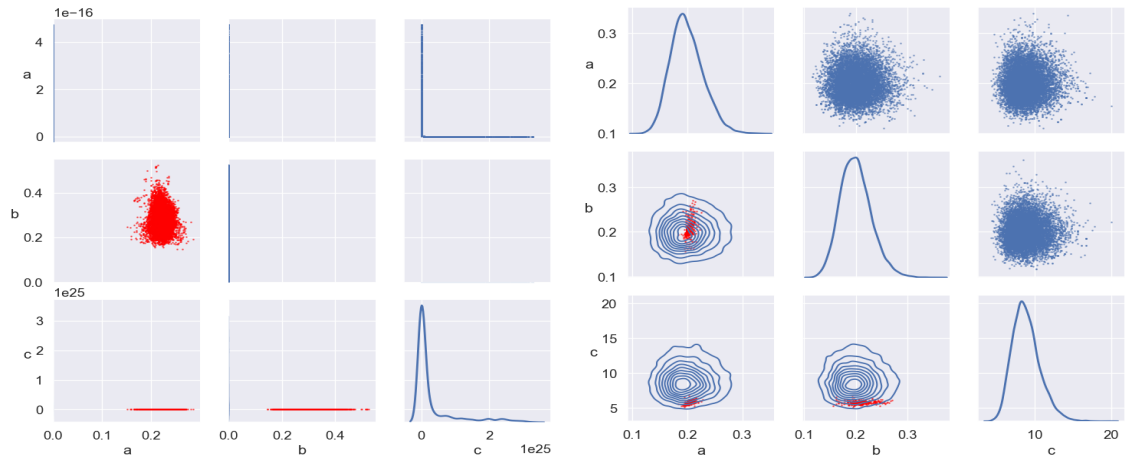


Figure 26. Rössler inverse problem: marginal (diagonal) and pairwise (lower triangle) distributions estimated with 10000 samples (upper triangle) by the pCN algorithm based on NN emulators for the time-averaged model (left) and the STGP model (right) respectively. Red dots (lower triangle) are selective 10000 ensemble particles from running the EKS algorithm.

reducing the estimation error using smaller observation time window T: the former requires only half time length as the latter to attain the same level of accuracy ($T = 30$ vs. $T = 60$ with EKI and $T = 20$ vs $T = 40$ with EKS).

Model-Algo	J=50	J=100	J=200	J=500	J=1000
Tavg-EKI	0.16 (0.09)	0.11 (0.06)	0.10 (0.07)	0.07 (0.04)	0.11 (0.07)
Tavg-EKS	0.06 (0.02)	0.06 (7.61e-03)	0.06 (6.20e-03)	0.06 (5.37e-03)	0.06 (2.53e-03)
STGP-EKI	0.02 (0.02)	0.01 (0.01)	0.02 (0.02)	0.01 (9.09e-03)	0.01 (0.02)
STGP-EKS	0.02 (0.01)	2.47e-03 (0.02)	7.63e-04 (2.86e-03)	4.23e-04 (2.45e-04)	3.62e-04 (1.19e-04)

Table 5. Rössler inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS), and the numbers in the bracket are standard deviations of such repeated experiments.

Now fix $t_0 = 1000$ and $T = 100$. Figure B.6 compares these two models (3.9) (3.13) in terms of REM's of the parameter estimation by EnK algorithms with different ensemble sizes (J). The STGP model (3.13) shows a universal advantage over the

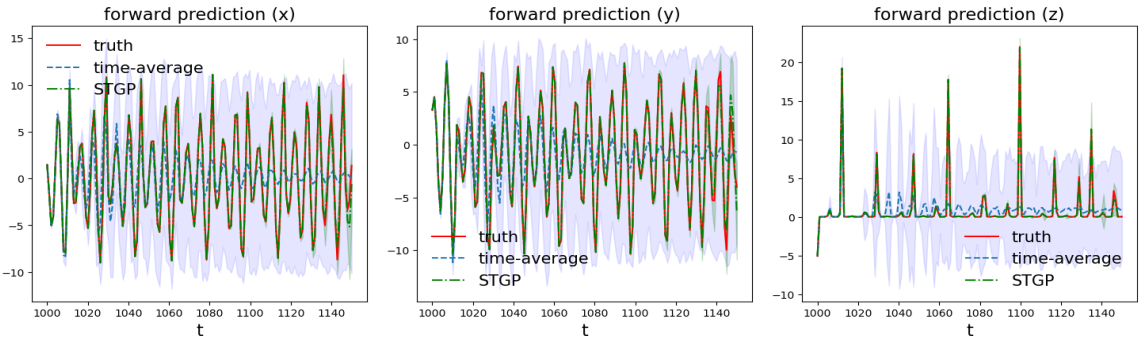


Figure 27. Rössler Inverse Problem: Comparing Forward Predictions $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ Based on the Time-averaged Model and the STGP Model

time-averaged model (3.9) in generating smaller REM's. Note, the time-averaged model becomes over-fitting if running EKS more than 10 iterations, a phenomenon also reported in [74, 73]. Table 5 summarizes the REM's by different combinations of likelihood models and EnK algorithms and confirms the consistent advantage of the STGP model over the time-averaged model in rendering more accurate parameter estimation.

CES (Section 2.2) is applied for the UQ. Based on the EKS ($J = 500$) outputs, I build DNN $\mathcal{G}^e : \mathbb{R}^3 \to \mathbb{R}^9$ for the time-averaged model (3.9) and DNN-RNN $\mathcal{G}^e : \mathbb{R}^3 \to \mathbb{R}^{3\times 100}$ for the STGP model (3.13) to account for their different data structures. Figure 26 compares the marginal and pairwise posterior densities of u estimated by 10000 samples of the pCN algorithm based on the corresponding NN emulators for the two models. The STGP model (3.13) generates more appropriate UQ results than the time-averaged model (3.9) does. Finally, consider the forward prediction $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ (3.23) for $t_* \in [t_0, t_0 + 1.5T]$ with $J = 500$ EKS ensembles corresponding to the lowest error. Figure 27 shows that the STGP model provides better prediction consistent with the truth throughout the whole time window while the result by the time-averaged model deviates from the truth quickly after $t = 1020$.

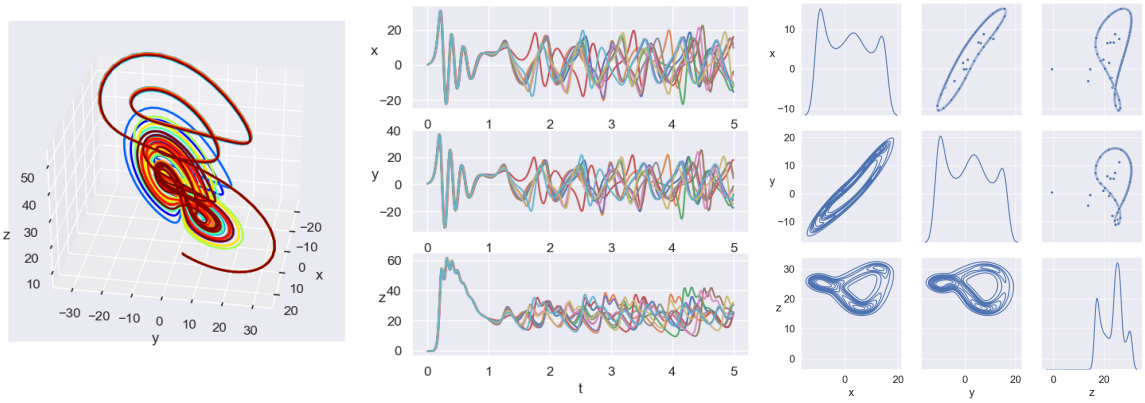


Figure 28. Chen dynamics: double-scroll attractor (left), chaotic solutions (middle), and coordinates' distributions (right).

3.5.2.3 Chen system

Yet another chaotic dynamical system I consider is the Chen system [25] described by the following ODE:

$$\begin{cases} \dot{x} &= a(y-x), \\ \dot{y} &= (c-a)x - xz + cy, \\ \dot{z} &= xy - bz. \end{cases} \tag{3.28}$$

where $a, b, c > 0$ are parameters. When $a = 35, b = 3, c = 28$, the system (3.28) has a double-scroll chaotic attractor often observed from a physical, electronic chaotic circuit. The true parameter that I will infer is $u^\dagger = (a^\dagger, b^\dagger, c^\dagger) = (35, 3, 28)$. With $u^\dagger$, the system has three unstable equilibrium states given by $(0, 0, 0)$, $(\gamma, \gamma, 2c - a)$, and $(-\gamma, -\gamma, 2c - a)$ where $\gamma = \sqrt{b(2c - a)}$ [165]. Figure 28 illustrates the two-scroll attractor (left), the chaotic trajectories (middle), and their marginal and pairwise distributions (right) of their coordinates viewed as random variables.

The Chen dynamics has trajectories changing rapidly as the Lorenz63 dynamics (compare the middle panels of Figure 28 and Figure 18). Therefore I adopt the

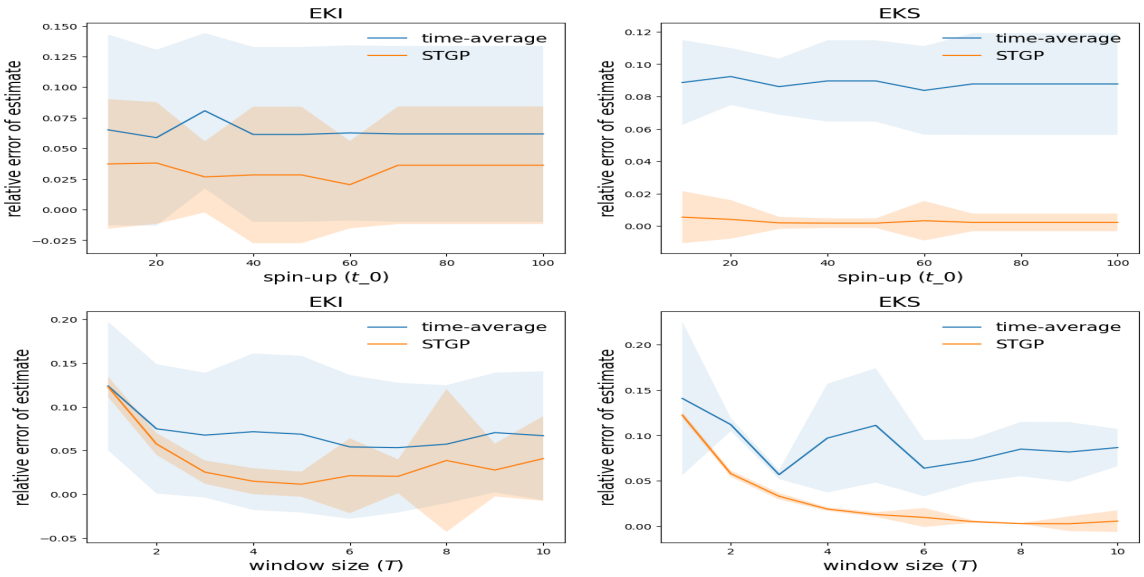


Figure 29. Chen inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of mean REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. The upper row shows the results by varying the spin-up t_0 and fixing $T = 10$. The lower row shows the results by varying the observation window size T and fixing $t_0 = 100$. Each experiment is repeated for 10 runs of EnK (EKI and EKS) with $J = 500$ ensembles, and the shaded regions indicate standard deviations of such repeated experiments.

same spin-up length ($t_0 = 100$) and observation window size ($T = 10$) as in the Lorenz inverse problem (Section 3.5.2.1). I generate the spatiotemporal data and the augmented time-averaged summary data by observing the trajectory of (3.28) over $[t_0, t_0 + T]$ solved with $u^\dagger$ similarly as in the previous sections. A log-Nomral prior is adopted for u: $\log u \sim \mathcal{N}(\mu_0, \sigma_0^2)$ with $\mu_0 = (3.5, 1.2, 3.3)$ and $\sigma_0 = (0.35, 0.5, 0.15)$. The STGP model (3.13) still posses more convex posterior density $p(u)$ than the time-averaged model (3.9) as illustrated by its marginal and pairwise sections plotted in Figure B.7.

Varying the spin-up length t_0 and the observation window size T one at a time in Figure 29, despite the insensitivity of errors concerning t_0, it is evident that the

STGP model offers comparable benefits to the time-averaged model. Similarly, the STGP model demands a smaller observation window than the time-averaged model ($T = 2$ vs. $T = 6$ with EKI and $T = 2$ vs $T = 3$ with EKS) to reach the same level of accuracy.

Model-Algo	J=50	J=100	J=200	J=500	J=1000
Tavg-EKI	0.07 (0.03)	0.04 (0.04)	0.04 (0.04)	0.05 (0.04)	0.04 (0.04)
Tavg-EKS	0.12 (0.03)	0.10 (0.02)	0.09 (0.02)	0.09 (0.01)	0.09 (0.01)
STGP-EKI	0.14 (0.09)	0.09 (0.08)	0.09 (0.08)	0.03 (0.03)	0.01 (9.87e-03)
STGP-EKS	0.07 (0.04)	0.05 (0.04)	0.01 (0.01)	2.89e-03 (6.07e-03)	3.32e-04 (4.66e-04)

Table 6. Chen inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS), and the numbers in the bracket are standard deviations of such repeated experiments.

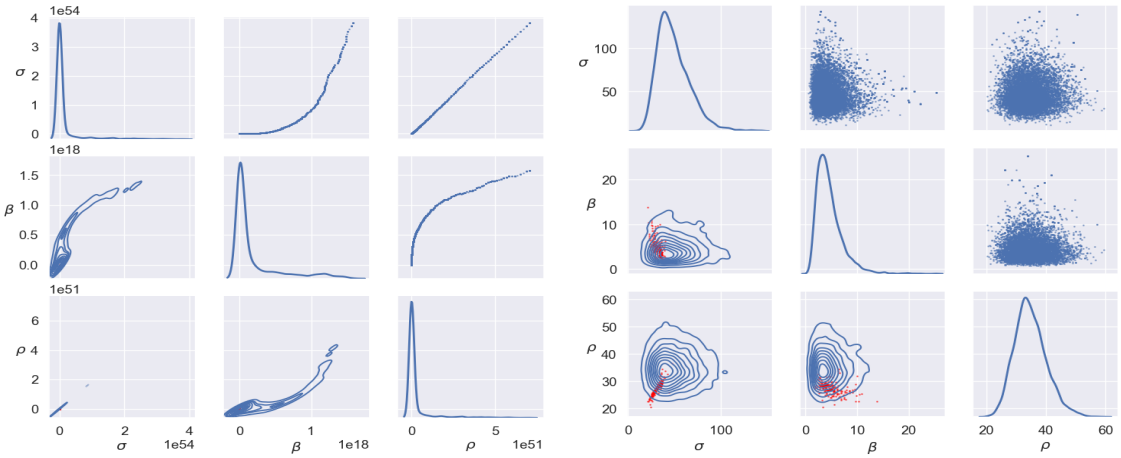


Figure 30. Chen inverse problem: marginal (diagonal) and pairwise (lower triangle) distributions estimated with 10000 samples (upper triangle) by the pCN algorithm based on NN emulators for the time-averaged model (left) and the STGP model (right) respectively. Red dots (lower triangle) are selective 10000 ensemble particles from running the EKS algorithm.

Again it's clear to see the merit of the STGP model (3.13) in reducing the error (REM) of parameter estimation compared with the time-averaged model (3.9) in

various combinations of EnK algorithms with different ensemble sizes (J) in Figure B.8 and Table 6. As in the previous problem (Section 3.5.2.2), similar over-fitting (bottom left of Figure B.8) by the time-averaged model occurs if running EKS algorithms more than 5 iterations (or earlier).

UQ results (Figure 30) by CES show the STGP model estimates the uncertainty of parameter u more appropriately than the time-averaged model. Finally, though the prediction is challenging to the Chen dynamics (3.28), the STGP model still performs much better than the time-averaged model by predicting a more accurate trajectory for a longer time ($t = 111$ vs $t = 101$) as shown in Figure 31.

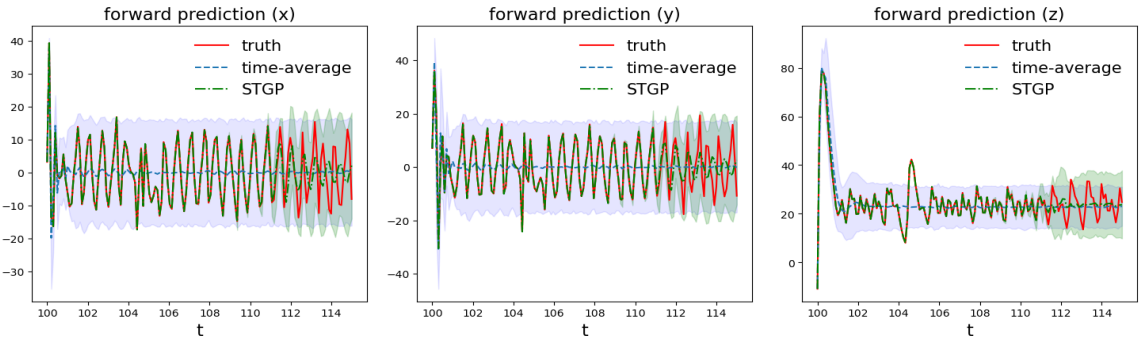


Figure 31. Chen Inverse Problem: Comparing Forward Predictions $\overline{\mathcal{G}}(\mathbf{x}, t_*)$ Based on the Time-averaged Model and the STGP Model

3.6 Conclusion

Traditional models often fail to incorporate spatiotemporal information. In contrast, the STGP model fits the observed data trajectories more accurately, resulting in improved parameter estimation and more appropriate UQ. The STGP model's superiority is supported by theorems that demonstrate its ability to provide a more convex likelihood, making parameter learning easier. Ultimately, I showcase the

benefits of spatiotemporal modeling through an inverse problem that is limited by an advection-diffusion PDE and three inverse problems that involve chaotic dynamics.

Furthermore, theorems 3.3.1 and 3.3.2 compare the STGP model with the static and time-averaged models regarding their statistical convexity. These novel qualitative results imply that the parameter learning (based on EnK methods) with the STGP model converges faster than the other two traditional methods. In future work, I will explore a quantitative characterization of their convergence rates, particularly in terms of covariance properties.

Chapter 4

SPATIOTEMPORAL PRIOR MODELING

4.1 Introduction

Recovering model parameters from observed data is the main goal of inverse problems. There are three emerging challenges in solving large-scale and data-intensive inverse problems: i) ill-posedness of the problem, ii) high dimensionality of the parameter space, and iii) complexity of the model constraints. To overcome the ill-posedness and the model constraints, regularization methods are developed to find meaningful solutions to inverse problems [63].

In the literature of optimization, deterministic regularization methods for inverse problems date back to 1943 with the seminal work by Tikhonov [148], followed by Ivanov [77] and [6]. A regularization term reflecting the properties of the target solution is typically added to a pre-determined energy function that depends on the forward model and the statistical assumptions of the observational noise. More recent methods and algorithms in this class have been developed for solving large-scale ill-posed problems [45, 56, 82] in signal and image processing [82, 56, 18], geophysics and seismic monitoring [144], satellite imaging [145], etc.

Compared with the optimization-based methods, the Bayesian approach to these inverse problems has the added benefit of quantifying the uncertainty of model parameters and evaluating the adequacy of model itself [38]. For more information on the background of the Bayesian inverse problem, please see Section 1.2.2.

In statistics, there is also a long history of developing models with penalty based

on the prior knowledge of unknown parameters. If we consider the Bayesian regression model as an inverse problem, we can draw a parallel between the likelihood (prior) and the energy (regularization) function. Within this context, it is imperative to offer a comprehensive outline of widely-used priors that researchers commonly incorporate as a form of penalty.

4.1.1 Gaussian Prior

GP [127, 9] has been introduced in chapter 3.2.1 and widely used as an L_2 penalty or a prior on the function space. Despite the flexibility, sometimes random candidate functions drawn from GP are over-smooth for modeling certain objects, such as images with sharp edges.

To address this issue and promote sparsity, an extensive list of classic shrinkage priors are adopted in Bayesian statistics, including Bayesian Lasso [121] and bridge [126], elastic net [103], group Lasso [23], and horseshoe priors [22, 150]. For non-parametric modeling, there are many heavy-tailed priors based on Markov random fields, including Laplace [69], Cauchy [110], and total variation (TV) [101]. There has also been a class of data-informed priors based on level set functions [21, 41, 75, 129] recently proposed for solving Bayesian inverse problems while retaining important spatial/graphical features (e.g., shape, edges) of the solution. All these sparsity-promoting and edge-preserving priors find remarkable applications, especially in imaging analysis, such as image deblurring, X-ray CT reconstruction, and image classification.

Among those edge-preserving priors, I particularly focus on the Besov prior, which corresponds to the L_q (usually set $q = 1$) type regularization [35, 16, 102]. [99]

discovered that the TV prior degenerates to Gaussian prior as the discretization mesh becomes denser, thus losing the edge-preserving properties in high dimensional applications. Therefore, [98] proposed the Besov prior defined particularly in terms of wavelet basis and random coefficients and proved its discretization-invariant property.

4.1.2 Besov Prior

Let the spatial domain be d-dimensional torus, i.e. $\mathcal{X} = \mathbb{T}^d = (0,1]^d$ for $d \leq 3$. Consider the Hilbert space $\dot{L}^2(\mathcal{X}) := \{u : \mathbb{T}^d \to \mathbb{R} | \int_{\mathbb{T}^d} |u(\mathbf{x})|^2 d\mathbf{x} < \infty, \int_{\mathcal{X}} u(\mathbf{x}) d\mathbf{x} = 0\}$ of real valued periodic functions $\mathcal{X}$ with inner product $\langle \cdot | \cdot \rangle$ and norm $\| \cdot \|$. Given an orthonormal basis $\{\phi_\ell\}_{\ell=1}^\infty$ for $\dot{L}^2(\mathcal{X})$, any function $u \in \dot{L}^2(\mathcal{X})$ can be written as

$$u(\mathbf{x}) = \sum_{\ell=1}^{\infty} u_\ell \phi_\ell(\mathbf{x}), \quad u_\ell = \langle u, \phi_\ell \rangle \tag{4.1}$$

Based on the above series (4.1), for $s > 0$ and $q \geq 1$, define the Bananch space $\mathbb{X}^{s,q} := \{u = \sum_{\ell=1}^\infty u_\ell \phi_\ell : \mathcal{X} \to \mathbb{R} | \|u\|_{s,q} < \infty, \int_{\mathcal{X}} u(\mathbf{x}) d\mathbf{x} = 0\}$ with the norm $\| \cdot \|_{s,q}$ specified as

$$\|u\|_{s,q} = \left(\sum_{\ell=1}^{\infty} \ell^{(\frac{sq}{d} + \frac{q}{2} - 1)} |u_\ell|^q \right)^{\frac{1}{q}} \tag{4.2}$$

Note, if $q = 2$ and $\{\phi_\ell\}_{\ell=1}^\infty$ form the Fourier basis, then $\mathbb{X}^{s,2}$ reduces to the Sobolev space $\dot{H}(\mathbb{T}^d)$ of mean-zero periodic functions with s-regularity; in particular, $\mathbb{X}^{0,2} = \dot{L}^2(\mathbb{T}^d)$. If $\{\phi_\ell\}_{\ell=1}^\infty$ is an r-regular wavelet basis for $r > s$, then $\mathbb{X}^{s,q}$ becomes the Besov space B^s_{qq} [149, 170].

Now construct a probability measure on functions by randomizing the coefficients $\{u_\ell\}_{\ell=1}^\infty$ of the series expansion (4.1) in the basis $\{\phi_\ell\}_{\ell=1}^\infty$. More specifically, from (4.1)

$$u_\ell := \gamma_\ell \xi_\ell, \quad \gamma_\ell = \kappa^{-\frac{1}{q}} \ell^{-(\frac{s}{d} + \frac{1}{2} - \frac{1}{q})}, \quad \xi_\ell \overset{iid}{\sim} \pi_\xi(\cdot) \propto \exp\left(-\frac{1}{2}|\xi|^q\right). \tag{4.3}$$

where $s > 0$, $1 \leq q < \infty$, $\kappa > 0$ are fixed, and π_ξ denotes the probability density function of the *q-exponential distribution*.

Denote infinite sequences $\gamma = \{\gamma_\ell\}_{\ell=1}^\infty$ and $\xi = \{\xi_\ell\}_{\ell=1}^\infty$. Then ξ is a random element of the probability space $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$ with $\Omega = \mathbb{R}^\infty$, product σ-algebra $\mathcal{B}(\Omega)$ and probability measure $\mathbb{P}$ defined by extending the finite product of π_ξ to infinite product by the Kolmogorov extension theorem [c.f. Theorem 29 in section A.2.1 of 38]. Then define the Besov measure as the pushforward of $\mathbb{P}$ as follows.

Definition 2 (Besov Measure)**.** *Let $\mathbb{P}$ be the measure of random sequences ξ. Suppose we have the following map*

$$f : \Omega \to \mathbb{X}^{s,q}, \quad \xi \mapsto u = \sum_{\ell=1}^\infty u_\ell \phi_\ell = \sum_{\ell=1}^\infty \gamma_\ell \xi_\ell \phi_\ell, \text{ where } \xi_\ell \overset{iid}{\sim} \pi_\xi \tag{4.4}$$

Then the pushforward $f^\sharp \mathbb{P}$ is Besov measure on $\mathbb{X}^{s,q}$, denoted as $\mathcal{B}(\kappa, \mathbb{X}^{s,q})$.

To make sense of (4.1) and (4.3) for $u \sim \mathcal{B}(\kappa, \mathbb{X}^{s,q})$, we need the following function space

$$L^q_{\mathbb{P}}(\Omega; \mathbb{X}^{t,q}) = \{u : D \times \Omega \to \mathbb{R} | \mathbb{E}(\|u\|^q_{t,q}) < \infty\} \tag{4.5}$$

This is a Banach space equipped with the norm $\mathbb{E}(\|u\|^q_{t,q})^{\frac{1}{q}}$. Then one can show that the random series (4.4) exists as an $L^q_{\mathbb{P}}$-limit in $\mathbb{X}^{t,q}$ for $t < s - \frac{d}{q}$ [Thorem 4 of 38].

Remark 4. *If $q = 2$ and $\{\phi_\ell\}_{\ell=1}^\infty$ is either a wavelet or Fourier basis, we obtain a Gaussian measure with the Cameron-Martin space B^s_{22} [20], which is the Hilbert space $H^s = H^s(\mathbb{T}^d)$. Indeed, we have (4.4) reduced to*

$$u(\boldsymbol{x}) = \kappa^{-\frac{1}{2}} \sum_{\ell=1}^\infty \ell^{-\frac{s}{d}} \xi_\ell \phi_\ell(\boldsymbol{x}), \quad \xi_\ell \overset{iid}{\sim} \mathcal{N}(0, 1) \tag{4.6}$$

After providing a review on BP (Section 4.1.2) with a flexible edge-preserving prior, the rest of the chapter is organized as follows. Section 4.2 introduces Q-exponential

process as random coefficient functions on the time domain. I then formally define STBP and study its theoretic properties in Section 4.3. In Section 4.4 I describe a white-noise representation of STBP that facilitates the inference for models with STBP prior. Section 4.5 covers experiments on Q-EP (Subsection 4.5.1) and STBP (4.5.2). In Subsection 4.5.1, I have shown that Q-EP outperforms GP and Besov in both time series modeling and image reconstruction. In Subsection 4.5.2, I demonstrate the advantage of the proposed STBP in retaining spatial features and capturing temporal correlations for the spatiotemporal inverse problems, including two dynamic CT reconstruction Finally, I discuss future research in Section 4.6.

4.2 Q-Exponential Process(Q-EP)

This section is adapted from: "Li, S., O'Connor, M., &Lan, S. (2022). Bayesian Learning via Q-Exponential Process (Version 2). arXiv. https://doi.org/10.48550/ARXIV.2210.07987".

Before generalizing the series representation of Besov random function (4.4) to a representation for spatiotemporal Besov process by replacing the random variable ξ_ℓ with a stochastic process $\xi_\ell(t)$ on the temporal domain $\mathcal{T} \subset \mathbb{R}_+$, I will first propose a properly defined q-exponential process which generalizes the q-exponential distribution to capture the temporal dependence in the data.

4.2.1 The Q-Exponential Distribution and its Multivariate Generalizations

Start with the q-exponential distribution for a scalar random variable $u \in \mathbb{R}$. It is named in [36] and defined with the following density, not in an exact form (as a

probability density normalized to 1):

$$\pi_q(u) \propto \exp\left(-\frac{1}{2}|u|^q\right). \tag{4.7}$$

This q-exponential distribution (4.7) is actually a special case of the following *exponential power (EP)* distribution $\mathrm{EP}(\mu, \sigma, q)$ with $\mu = 0$, $\sigma = 1$:

$$p(u|\mu, \sigma, q) = \frac{q}{2^{1+1/q}\sigma\Gamma(1/q)} \exp\left\{-\frac{1}{2}\left|\frac{u-\mu}{\sigma}\right|^q\right\} \tag{4.8}$$

where Γ denotes the gamma function. Note the parameter $q > 0$ in (4.8) controls the tail behavior of the distribution: the smaller q, the heavier tail, and vice versa. This distribution also includes many commonly used ones, such as the normal distribution $\mathcal{N}(\mu, \sigma^2)$ for $q = 2$ and the Laplace distribution $L(\mu, b)$ with $\sigma = 2^{-1/q}b$ when $q = 1$.

The question now arises: How can I generalize it to a multivariate distribution and, further, to a stochastic process? Gomez [57] provided one possibility of a multivariate EP distribution, denoted as $\mathrm{EP}_d(\boldsymbol{\mu}, \mathbf{C}, q)$, with the following density:

$$p(\mathbf{u}|\boldsymbol{\mu}, \mathbf{C}, q) = \frac{q\Gamma(\frac{d}{2})}{2\Gamma(\frac{d}{q})} 2^{-\frac{d}{q}} \pi^{-\frac{d}{2}} |\mathbf{C}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\left[(\mathbf{u}-\boldsymbol{\mu})^\mathsf{T}\mathbf{C}^{-1}(\mathbf{u}-\boldsymbol{\mu})\right]^{\frac{q}{2}}\right\} \tag{4.9}$$

When $q = 2$, it reduces to the familiar multivariate normal (MVN) distribution $\mathcal{N}_d(\boldsymbol{\mu}, \mathbf{C})$.

Unfortunately, unlike MVN being the foundation of GP, the Gomez's EP distribution $\mathrm{EP}_d(\boldsymbol{\mu}, \mathbf{C}, q)$ fails to generalize to a valid stochastic process because it does not satisfy the marginalization consistency as MVN does (See Subsection 4.2.2 for more details). It turns out we need to seek candidates in an even larger family of *elliptic* (contour) distributions $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$:

Definition 3 (Elliptic distribution)**.** *A multivariate elliptic distribution* $\mathrm{EC}_d(\boldsymbol{\mu}, \boldsymbol{C}, g)$ *has the following density [78]*

$$p(\boldsymbol{u}) = k_d|\boldsymbol{C}|^{-\frac{1}{2}} g(r), \quad r(\boldsymbol{u}) = (\boldsymbol{u}-\boldsymbol{\mu})^\mathsf{T}\boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) \tag{4.10}$$

where $k_d > 0$ is the normalizing constant and $g(\cdot)$, a one-dimensional real-valued function independent of d and k_d, is named density generating function *[51].*

Every elliptic (contour) distributed random vector $\mathbf{u} \sim \mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ has a stochastic representation mainly due to Schoenberg [138, 19, 78], as stated in the following theorem.

Theorem 4.2.1. *$\boldsymbol{u} \sim \mathrm{EC}_d(\boldsymbol{\mu}, \boldsymbol{C}, g)$ if and only if*

$$\boldsymbol{u} \overset{d}{=} \boldsymbol{\mu} + R\boldsymbol{L}S \tag{4.11}$$

where $S \sim \mathrm{Unif}(\mathcal{S}^{d+1})$ uniformly distributed on the unit-sphere $\mathcal{S}^{d+1}$, $\boldsymbol{L}$ is the Cholesky factor of $\boldsymbol{C}$ such that $\boldsymbol{C} = \boldsymbol{L}\boldsymbol{L}^\top$, $R \perp S$ and $R^2 \overset{d}{=} r(\boldsymbol{u}) \sim f(r) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2})} k_d r^{\frac{d}{2}-1} g(r)$.

The Gomez's EP distribution $\mathrm{EP}_d(\boldsymbol{\mu}, \mathbf{C}, q)$ is a special elliptic distribution $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ with $g(r) = \exp\{-\frac{1}{2} r^{\frac{q}{2}}\}$ and $R^q \sim \Gamma(\alpha = \frac{d}{q}, \beta = \frac{1}{2})$ [57]. Not all elliptical distributions can be used to create a valid process [7].

In the following, I will carefully choose the density generator g in $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ to define a consistent multivariate q-exponential distribution generalizable to a process appropriately.

4.2.2 The Q-Exponential Process

To generalize $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ to a valid stochastic process, I need to choose proper g such that the resulting distribution satisfies two conditions of Kolmogorov extension theorem [118]:

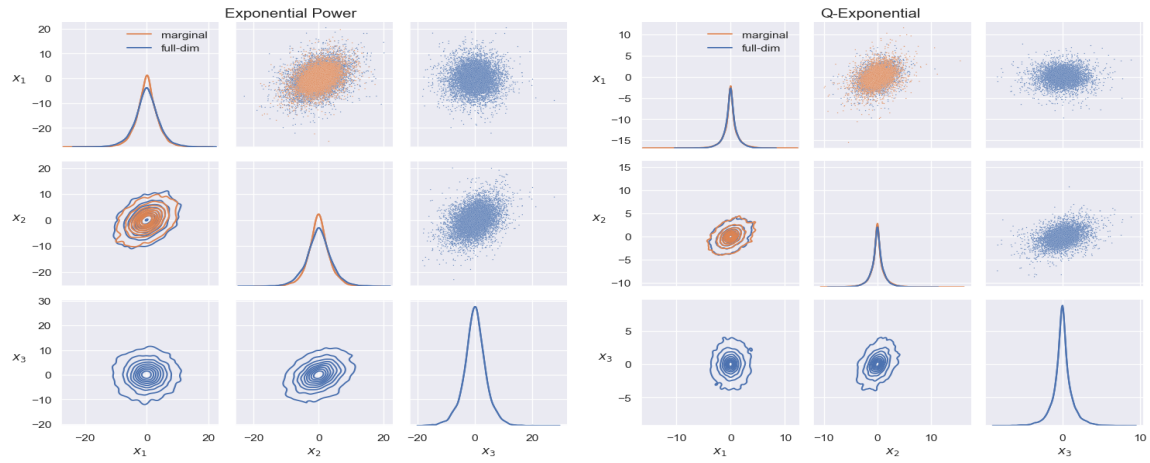


Figure 32. Inconsistent (Gomez's) EP distribution $\mathrm{EP}_d(\boldsymbol{\mu}, \mathbf{C}, q)$ (left) vs. consistent Q-exponential distribution $\mathrm{q-ED}_d(\boldsymbol{\mu}, \mathbf{C})$ (right). Both can be sampled using (4.11) with $R^q \sim \Gamma(\alpha = \frac{d}{q}, \beta = \frac{1}{2})$ and $R^q \sim \Gamma\left(\alpha = \frac{d}{2}, \beta = \frac{1}{2}\right)$ respectively. Note there is significant discrepancy between the marginalization of $\mathrm{EP}_3(\boldsymbol{\mu}, \mathbf{C}, q)$ and $\mathrm{EP}_2(\boldsymbol{\mu}, \mathbf{C}, q)$. However, the marginalization of $\mathrm{q-ED}_3(\boldsymbol{\mu}, \mathbf{C})$ coincides with $\mathrm{q-ED}_2(\boldsymbol{\mu}, \mathbf{C})$. Empirical densities are estimated based on 10000 samples (shown as dots).

Theorem 4.2.2 (Kolmogorov's Extension)**.** *For all $t_1, \cdots, t_k \in T$, $k \in \mathbb{N}$ let $\nu_{t_1,\cdots,t_k}$ be probability measures on $\mathbb{R}^{nk}$ satisfying*

$$\begin{aligned}
&(K1): \nu_{t_{\sigma(1)},\cdots,t_{\sigma(k)}}(F_1 \times \cdots \times F_k) \\
&\qquad = \nu_{t_1,\cdots,t_k}(F_{\sigma^{-1}(1)} \times \cdots \times F_{\sigma^{-1}(k)}) \; \textit{for all permutations } \sigma \in S(k) \\
&(K2): \nu_{t_1,\cdots,t_k}(F_1 \times \cdots \times F_k) \\
&\qquad = \nu_{t_1,\cdots,t_k,t_{k+1},\cdots,t_{k+m}}(F_1 \times \cdots \times F_k \times \mathbb{R}^k \times \cdots \times \mathbb{R}^n) \; \textit{for all } m \in \mathbb{N}
\end{aligned} \tag{4.12}$$

Then there exists a probability space $(\Omega, \mathcal{F}, P)$ and a stochastic process $\{X_t\}$ on Ω, $X_t : \Omega \to \mathbb{R}^n$ such that

$$\nu_{t_1,\cdots,t_k}(F_1 \times \cdots \times F_k) = P[X_{t_1} \in F_1, \cdots, X_{t_k} \in F_k] \tag{4.13}$$

for all $t_i \in T$, $k \in \mathbb{N}$ and all Borel sets $F_i \in \mathcal{F}$. (K1) and (K2) are referred to as ***exchangeability*** *and* ***consistency*** *conditions respectively.*

As pointed out by Kano [81], the elliptic distribution $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ in the format of Gomez's EP distribution (4.9) with $g(r) = \exp\{-\frac{1}{2}r^{\frac{q}{2}}\}$ does not satisfy the consistency condition [also c.f. Proposition 5.1 of 57]. Figure 32 (left panel) also illustrates such inconsistency numerically. However, Kano's consistency theorem [81] suggests a different viable choice of g to make a valid generalization of $\mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ to a stochastic process [7]:

Theorem 4.2.3 (Kano's Consistency)**.** *An elliptic distribution is* consistent *if and only if its density generator function, $g(\cdot)$, has the following form*

$$g(r) = \int_0^\infty \left(\frac{s}{2\pi}\right)^{\frac{d}{2}} \exp\left\{-\frac{rs}{2}\right\} p(s)ds \tag{4.14}$$

where $p(s)$ is a strictly positive mixing distribution independent of d and $p(s = 0) = 0$.

4.2.2.1 Consistent Multivariate Q-exponential Distribution

In the above theorem 4.2.3, if choosing $p(s) = \delta_{r^{\frac{q}{2}-1}}(s)$, then $g(r) = r^{\left(\frac{q}{2}-1\right)\frac{d}{2}} \exp\left\{-\frac{r^{\frac{q}{2}}}{2}\right\}$, which leads to the following consistent *multivariate q-exponential distribution* $\mathrm{q-ED}_d(\boldsymbol{\mu}, \mathbf{C})$.

Definition 4. *A multivariate q-exponential distribution, denoted as* $\mathrm{q-ED}_d(\boldsymbol{\mu}, \boldsymbol{C})$*, has the following density*

$$p(\boldsymbol{u}|\boldsymbol{\mu}, \boldsymbol{C}, q) = \frac{q}{2}(2\pi)^{-\frac{d}{2}}|\boldsymbol{C}|^{-\frac{1}{2}}\boxed{r^{(\frac{q}{2}-1)\frac{d}{2}}}\exp\left\{-\frac{r^{\frac{q}{2}}}{2}\right\}, \quad r(\boldsymbol{u}) = (\boldsymbol{u}-\boldsymbol{\mu})^\top \boldsymbol{C}^{-1}(\boldsymbol{u}-\boldsymbol{\mu}) \tag{4.15}$$

Remark 5. *When $q = 2$, $\mathrm{q–ED}_d(\boldsymbol{\mu}, \boldsymbol{C})$ reduces to MVN $\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{C})$. When $d = 1$, if we let $C = 1$, then we have the density for u as $p(u) \propto |u|^{\frac{q}{2}-1} \exp\left\{-\frac{1}{2}|u|^q\right\}$, differing from the original un-normalized density π_q in (4.7) by a term $|u|^{\frac{q}{2}-1}$. This is needed for the*

consistency of process generalization. Numerically, it has a similar "edge-preserving property" as the Besov prior.

Regardless of the normalizing constant, the proposed multivariate q-exponential distribution $\mathrm{q}-\mathrm{ED}_d(\boldsymbol{\mu}, \mathbf{C})$ differs from the Gomez's EP distribution $\mathrm{EP}_d(\boldsymbol{\mu}, \mathbf{C}, q)$ by a boxed term $r^{(\frac{q}{2}-1)\frac{d}{2}}$. As stated in the following theorem, $\mathrm{q}-\mathrm{ED}_d$ satisfies the two conditions of the Kolmogorov extension theorem and thus is ready to generalize to a stochastic process (See the right panel of Figure 32 for the consistency).

Theorem 4.2.4. *The multivariate q-exponential distribution is both* ***exchangeable*** *and* ***consistent****.*

Proof. See Appendix A.3.1. □

Like student-t distribution [139] and other elliptic distributions [7], it can be demonstrated (refer to Appendix A.3.2) that the representation of $\mathrm{q}-\mathrm{ED}_d$ is a scale mixture of Gaussian distributions [81, 5].

Numerically, thanks to the choice of density generator $g(r) = r^{\left(\frac{q}{2}-1\right)\frac{d}{2}} \exp\left\{-\frac{r^{\frac{q}{2}}}{2}\right\}$, one can show that $R^q \sim \chi^2_d$ (as in Appendix A.3.3) thus R in Theorem 4.2.1 can be sampled as q-root of a χ^2 random variable, which completes the recipe for generating random vector $\mathbf{u} \sim \mathrm{q}-\mathrm{ED}_d(0, \mathbf{C})$ based on the stochastic representation (4.11). This is important for the Bayesian inference as detailed in Section 4.4. Note the matrix $\mathbf{C}$ in the definition (4.15) characterizes the covariance between the components, as shown in the following proposition.

Proposition 4.2.1. *If $\boldsymbol{u} \sim \mathrm{q}-\mathrm{ED}_d(\boldsymbol{\mu}, \boldsymbol{C})$, then we have*

$$\mathrm{E}[\boldsymbol{u}] = \boldsymbol{\mu},\ \mathrm{Cov}(\boldsymbol{u}) = \frac{2^{\frac{2}{q}}\Gamma(\frac{d}{2}+\frac{2}{q})}{d\Gamma(\frac{d}{2})}\boldsymbol{C} \dot{\sim} d^{\frac{2}{q}-1}\boldsymbol{C},\ \textit{as } d \to \infty \tag{4.16}$$

Proof. See Appendix A.3.4. □

4.2.2.2 Q-exponential Process as Probabilistic Definition of Besov Process

To generalize $\mathbf{u} \sim \mathrm{q}-\mathrm{ED}_d(0, \mathbf{C})$ to a stochastic process, I want to scale it to $\mathbf{u}^* = d^{\frac{1}{2}-\frac{1}{q}}\mathbf{u}$ so that its covariance is asymptotically finite. If $\mathbf{u} \sim \mathrm{q}-\mathrm{ED}_d(0, \mathbf{C})$, then denote $\mathbf{u}^* \sim \mathrm{q}-\mathrm{ED}_d^*(0, \mathbf{C})$ following a *scaled* q-exponential distribution. At this juncture, the task at hand is to establish the *q-exponential process (Q-EP)* with the scaled q-exponential distribution.

Definition 5 (Q-EP)**.** *A (centered) q-exponential process $u(x)$ with kernel $\mathcal{C}$, $\mathrm{q}-\mathcal{EP}(0, \mathcal{C})$, is a collection of random variables such that any finite set, $\boldsymbol{u} = (u(x_1), \cdots u(x_d))$, follows a scaled multivariate q-exponential distribution, i.e. $\boldsymbol{u} \sim \mathrm{q}-\mathrm{ED}_d^*(0, \boldsymbol{C})$.*

Both Besov and Q-EP are valid stochastic processes stemming from the q-exponential distribution π_q. They are both designed to generalize GP to have sharper regularization (through q), but Q-EP has advantages in 1) the capability of specifying correlation structure directly and 2) the tractable prediction formula.

It follows from (4.1) immediately that the covariance of the Besov process $u(\cdot)$ at two points $x, x' \in \mathbb{R}^{d^\star}$:

$$\mathrm{Cov}(u(x), u(x')) = \sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(x) \otimes \phi_\ell(x') \tag{4.17}$$

Compared with (4.16), we have less control over the correlation strength once the orthonormal basis $\{\phi_\ell\}$ is chosen. On the other hand, Q-EP has more freedom on the correlation structure through (4.16) with flexible choices from a large class of kernels, such as powered exponential and Matérn, that allow us to specify the correlation length directly.

While Q-EP can be viewed as a probabilistic definition of Besov, the following theorem further establishes their connection in sharing equivalent series representations.

Theorem 4.2.5 (Karhunen-Loéve)**.** *If* $u(x) \sim \text{q}-\mathcal{EP}(0, \mathcal{C})$ *with* $\mathcal{C}$ *having eigen-pairs* $\{\lambda_\ell, \phi_\ell(x)\}_{\ell=1}^{\infty}$ *such that* $\mathcal{C}\phi_\ell(x) = \phi_\ell(x)\lambda_\ell$, $\|\phi_\ell\|_2 = 1$ *for all* $\ell \in \mathbb{N}$ *and* $\sum_{\ell=1}^{\infty} \lambda_\ell < \infty$, *then we have the following series representation for* $u(x)$*:*

$$u(x) = \sum_{\ell=1}^{\infty} u_\ell \phi_\ell(x), \quad u_\ell := \int_D u(x)\phi_\ell(x) \overset{ind}{\sim} \text{q}-\text{ED}^*(0, \lambda_\ell) \tag{4.18}$$

where $\text{E}[u_\ell] = 0$ *and* $\text{Cov}(u_\ell, u_{\ell'}) = \lambda_\ell \delta_{\ell\ell'}$ *with Dirac function* $\delta_{\ell\ell'} = 1$ *if* $\ell = \ell'$ *and* 0 *otherwise.*

Proof. See Appendix A.3.5. □

Remark 6. *If we factor* $\sqrt{\lambda_\ell}$ *out of* u_ℓ*, we have the following expansion for Q-EP more comparable to* (4.1) *for Besov:*

$$u(x) = \sum_{\ell=1}^{\infty} \sqrt{\lambda_\ell} u_\ell \phi_\ell(x), \quad u_\ell \overset{iid}{\sim} \text{q}-\text{ED}(0, 1) \propto \pi_q(\cdot) \tag{4.19}$$

4.3 Spatiotemporal Besov Process

While BP outperforms GP in producing high-quality image reconstructions or solutions to general inverse problems that preserve spatial features(see subsection 4.5.1), it does not account for the temporal correlations existing in a series of dynamically changing images. Thus I generalize BP to the spatiotemporal domain by taking advantage of Q-EP in capturing temporal changes. More specifically, I replace the random coefficients (following univariate q-exponential distribution) in the series representation of BP with stochastic time functions following Q-EP.

Generalize the Banach space $\mathbb{X}^{s,q}$ to include the temporal domain $\mathcal{T} \subset \mathbb{R}_+$. Let the coefficients in (4.1) be $L^p(\mathcal{T})$ functions over some finite temporal domain $\mathcal{T}$. Denote

$\mathcal{Z} = \mathcal{X} \times \mathcal{T}$. Then a spatiotemporal function $u(\mathbf{x}, t)$ on $\mathcal{Z}$ is obtained by the following series expansion with an infinite sequence of $L^p(\mathcal{T})$ functions:

$$u(\mathbf{x}, t) = \sum_{\ell=1}^{\infty} u_\ell(t)\phi_\ell(\mathbf{x}), \quad u_\ell(\cdot) \in L^p(\mathcal{T}),\ \forall \ell \in \mathbb{N} \tag{4.20}$$

Denote $u_\mathcal{T} := \{u_\ell(\cdot)\}_{\ell=1}^{\infty}$. Define the following (r, q, p) norm for such sequence $u_\mathcal{T}$ with spatial (Besov) index q and temporal (Q-EP) index p:

$$\|u\|_{r,q,p} = \left(\sum_{\ell=1}^{\infty} \ell^{rq} \|u_\ell(\cdot)\|_p^q \right)^{\frac{1}{q}} \tag{4.21}$$

where we can choose $r = r_0 := \frac{s}{d} + \frac{1}{2} - \frac{1}{q}$. Denote the space of such infinite sequences $\ell^{r,q}(L^p(\mathcal{T})) := \{u_\mathcal{T} = \{u_\ell(\cdot)\}_{\ell=1}^{\infty} | \|u\|_{r,q,p} < \infty\}$. For a fixed orthonormal spatial basis $\{\phi_\ell(\mathbf{x})\}_{\ell=1}^{\infty}$, the Banach space of spatiotemporal functions can be defined based on the series representation (4.20), i.e., $\mathbb{X}^{r,q,p} := \{u(\mathbf{x}, t) = \sum_{\ell=1}^{\infty} u_\ell(t)\phi_\ell(\mathbf{x}) : \mathcal{X} \times \mathcal{T} \to \mathbb{R} | u_\mathcal{T} \in \ell^{r,q}(L^p(\mathcal{T}))\}$, with (r, q, p) norm as specified in (4.21) for the associated sequence $u_\mathcal{T}$.

Next I generalize Besov process $u(\mathbf{x}) \sim \mathcal{B}(\kappa, \mathbb{X}^{s,q})$ as in (4.4) to be spatiotemporal by letting random coefficients $\{\xi_\ell\}$ vary in time according to a q$-\mathcal{EP}$ process. In (4.20) set

$$u_\ell(t) = \gamma_\ell \xi_\ell(t), \quad \gamma_\ell = \kappa^{-\frac{1}{q}} \ell^{-r_0}, \quad \xi_\ell(\cdot) \overset{iid}{\sim} \text{q}-\mathcal{EP}(0, \mathcal{C}) \tag{4.22}$$

Denote the resulting stochastic process as *spatiotemporal Besov process* $\mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r,q,p})$. Similarly as above, the infinite random sequence $\xi_\mathcal{T} := \{\xi_\ell(\cdot)\}_{\ell=1}^{\infty}$ is a random element of the probability space $(\Omega, \mathcal{B}(\Omega), \mathbb{P})$ with $\Omega = \ell^{r,q}(L^p(\mathcal{T}))$, product σ-algebra $\mathcal{B}(\Omega)$ and probability measure $\mathbb{P}$ defined by extending the finite product of q$-\mathcal{EP}(0, \mathcal{C})$ to infinite product by the Kolmogorov extension theorem [c.f. Theorem 29 in section A.2.1 of 38]. Then $\mathcal{STBP}$ also defines a spatiotemporal Besov measure on $\mathbb{X}^{r,q,p}$ as follows.

Definition 6 (Spatiotemporal Besov Measure)**.** *Let $\mathbb{P}$ be the measure of random sequences $\xi_{\mathcal{T}}$. Suppose we have the following map*

$$\begin{aligned} &f : \Omega \to \mathbb{X}^{r,q,p} \\ &\xi_{\mathcal{T}} \mapsto u(\boldsymbol{x}, t) = \sum_{\ell=1}^{\infty} u_\ell(t)\phi_\ell(\boldsymbol{x}) = \sum_{\ell=1}^{\infty} \gamma_\ell \xi_\ell(t)\phi_\ell(\boldsymbol{x}),\ \textit{where } \xi_\ell(\cdot) \overset{iid}{\sim} \mathrm{q}{-}\mathcal{EP}(0, \mathcal{C}) \end{aligned} \tag{4.23}$$

Then the pushforward $f^\sharp\mathbb{P}$ is spatiotemporal Besov measure Π on $\mathbb{X}^{r,q,p}$.

For a given random draw $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r,q,p})$, its norm can be computed as

$$\|u\|_{r,q,p} = \left(\sum_{\ell=1}^{\infty} \ell^{rq} \|u_\ell\|_p^q \right)^{\frac{1}{q}} = \kappa^{-\frac{1}{q}} \left(\sum_{\ell=1}^{\infty} \ell^{(r-r_0)q} \|\xi_\ell\|_p^q \right)^{\frac{1}{q}} \tag{4.24}$$

4.3.1 STBP as A Prior

The following theorem states the conditions such that a random function $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r,q,p})$ in (4.23) is well-defined in the context of almost sure convergence.

Theorem 4.3.1. *Let $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r_0,q,p})$ be a random draw as in* (4.23)*. We have the following equivalent:*

(i) $u \in \mathbb{X}^{r,q,p}$ $\mathbb{P}$*-a.s.*

(ii) $\mathbb{E}[\exp(\alpha \|u\|_{r,q,p}^q)] < \infty$ *for any* $\alpha \in (0, \kappa/2)$.

(iii) $r < r_0 - \frac{1}{q}$.

Similarly as in [93], one can prove that an STBP $u(\mathbf{x}, t)$ defined in (4.23) can be represented completely in a series of spatial $(\{\phi_\ell\}_{\ell=1}^{\infty})$ and temporal $\{\psi_{\ell'}\}_{\ell'=1}^{\infty}$ bases.

Theorem 4.3.2. *If $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r_0,q,p})$ with $\mathcal{C}$ having eigen-pairs $\{\lambda_\ell, \psi_\ell(t)\}_{\ell=1}^{\infty}$ such that $\mathcal{C}\psi_\ell(t) = \psi_\ell(t)\lambda_\ell$, $\|\psi_\ell\|_2 = 1$ for all $\ell \in \mathbb{N}$ and $\sum_{\ell=1}^{\infty} \lambda_\ell < \infty$, then we have*

the following series representation for $u(\boldsymbol{x}, t)$*:*

$$u(\boldsymbol{x}, t) = \sum_{\ell=1}^{\infty} \sum_{\ell'=1}^{\infty} u_{\ell\ell'} \phi_\ell(\boldsymbol{x}) \psi_{\ell'}(t), \quad u_{\ell\ell'} := \int_{\mathcal{T}} u_\ell(t) \psi_{\ell'}(t) dt \overset{ind}{\sim} \text{q}-\text{ED}^*(0, \gamma_\ell^2 \lambda_{\ell'}) \tag{4.25}$$

Let $\{\phi_\ell\}_{\ell=1}^{\infty}$ be an r-regular wavelet basis of the Besov space B_{qq}^s. Then there exists the following Fernique-type theorem regarding the regularity of a random function from STBP.

Theorem 4.3.3. *Let* u *be a random function defined as in* (4.23) *with* $q \geq 1$ *and* $r_0 > \frac{1}{q}$*. Then for any* $r < r_0 - \frac{1}{q}$*,*

$$\text{E}[\exp(\alpha \|u\|_{C^t(\mathcal{Z})})] < \infty \tag{4.26}$$

for all $\alpha \in (0, \kappa/(2r^*))$*, with* r^* *a constant depend on* q*,* d*,* s *and* $t := d\left(r + \frac{1}{q} - \frac{1}{2}\right)$*.*

Based on the construction (4.23), one can show that the spatiotemporal covariance of STBP bears a separable structure, i.e., $\sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(\cdot)\phi_\ell(\cdot) \otimes \mathcal{C}$, as stated in the following proposition.

Proposition 4.3.1. *If* $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r_0,q,p})$*, then we have*

$$\text{Cov}(u(\boldsymbol{x}, t), u(\boldsymbol{x}', t')) = \sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(\boldsymbol{x}) \phi_\ell(\boldsymbol{x}') \mathcal{C}(t, t') \tag{4.27}$$

Proof. See Appendix A.4. □

The resulted *spatiotemporla Besov process (STBP)* can be flexible in modeling functional data with spatial features while explicitly controlling the temporal correlations through a covariance kernel. To the best of my knowledge, this is by far the first spatiotemporal generalization of BP. The proposed work on STBP has multiple contributions to the literature:

1. It generalizes BP to the spatiotemporal domain to capture spatial features and model the temporal correlations.

2. It provides a theoretical characterization of the posterior contraction in the data limit, justifying its validity as a nonparametric learning tool.
3. It demonstrates utility in CT reconstruction and indicates the potential impact on medical imaging analysis.

4.4 Bayesian Inference

In this section, I describe the inference of the Bayesian inverse problem with spatiotemporal data using a spatiotemporal Besov prior. Assume the unknown function u is evaluated at I locations $\mathbf{X} := \{\mathbf{x}_i\}_{i=1}^I$ and J time points $\mathbf{t} := \{t_j\}_{j=1}^J$, which is $u(\mathbf{X}, \mathbf{t}) := \{u(\mathbf{x}_i, t_j)\}_{i,j=1}^{I,J}$. In the dynamic tomography imaging problems, $u(\mathbf{x}_i, t_j)$ refers to the image pixel value of point $\mathbf{x}_i$ at time t_j with resolution $I = n_x \times n_y$. The data $\mathbf{Y} = \{\mathbf{y}_j\}_{j=1}^J$ with $\mathbf{y}_j \in \mathbb{R}^m$ is observed through the forward operator $\mathcal{G}$, which could be a linear (Radon) transform or governed by a PDE. In this work, I consider Gaussian noise and recap the model as follows.

$$\begin{aligned} &\mathbf{y}_j = \mathcal{G}(u)(\mathbf{X}, t_j) + \boldsymbol{\varepsilon}_j, \quad \boldsymbol{\varepsilon}_j \overset{iid}{\sim} \mathcal{N}_I(0, \Gamma_{\text{noise}}), \quad j = 1, 2, \ldots, J, \\ &u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r,q,p}) \end{aligned} \tag{4.28}$$

In applications of inverse problems, the spatial dimension I is usually tremendously higher than the temporal dimension ($I \gg J$). Therefore I truncate u in (4.23) for the first $L > 0$ terms: $u(\mathbf{x}, t) \approx u^L(\mathbf{x}, t) = \sum_{\ell=1}^L \gamma_\ell \xi_\ell(t) \phi_\ell(\mathbf{x})$, Denote $\mathbf{u}_j = u(\mathbf{X}, t_j) \in \mathbb{R}^I$, and $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_J]_{I \times J} = u^L(\mathbf{X}, \mathbf{t}) = \boldsymbol{\Phi}\, \text{diag}(\boldsymbol{\gamma}) \boldsymbol{\Xi}^\mathsf{T}$ where $\boldsymbol{\Phi} = [\phi_1(\mathbf{X}), \cdots, \phi_L(\mathbf{X})]_{I \times L}$, $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_L)$, and $\boldsymbol{\Xi} = [\xi_1(\mathbf{t}), \cdots, \xi_L(\mathbf{t})]_{J \times L}$. Instead of the large dimensional matrix $\mathbf{U}$, I can work with $\boldsymbol{\Xi}$ of much smaller size. Let

$r_\ell = \xi_\ell(\mathbf{t})^\mathsf{T}\mathbf{C}_J^{-1}\xi_\ell(\mathbf{t})$. Then (log) posterior for $\boldsymbol{\Xi}$ can be computed directly as

$$\begin{aligned}\log p(\boldsymbol{\Xi},\theta|\mathbf{Y}) = &-\frac{J}{2}\log|\Gamma_{\text{noise}}| - \frac{1}{2}\sum_{j=1}^{J}\|\mathbf{y}_j - \mathcal{G}(\mathbf{u}_j)\|^2_{\Gamma_{\text{noise}}} \\ &-\frac{L}{2}\log|\mathbf{C}_J| + \frac{J}{2}(\frac{q}{2}-1)\sum_{\ell=1}^{L}\log r_\ell - \frac{1}{2}\sum_{\ell=1}^{L} r_\ell^{\frac{q}{2}}\end{aligned} \tag{4.29}$$

To obtain the MAP estimate, one needs to optimize $\log p(\boldsymbol{\Xi},\theta|\mathbf{Y})$. In order to accurately measure uncertainty, I require sampling techniques that are independent of dimensions for models that are non-Gaussian. Refer to the work of dimension-robust MCMC by [26] based on the pushforward of Gaussian white noise, I would introduce a particular white noise representation for STBP different from the one by [26] in the following.

4.4.1 White Noise Representation

Recall from theorem 4.2.1 we have the stochastic representation which states $\boldsymbol{\xi} \sim \text{q}-\text{ED}_J(\mathbf{0},\mathbf{C})$: $\boldsymbol{\xi} = R\mathbf{L}S$ with $R^q \sim \chi^2(J)$ and $S \sim \text{Unif}(\mathcal{S}^{J+1})$. Write

$$S = \frac{\mathbf{z}}{\|\mathbf{z}\|_2}, \quad R^q = \|\mathbf{z}\|_2^2, \quad \text{for } \mathbf{z} \sim \mathcal{N}_J(\mathbf{0},\mathbf{I}_J) \tag{4.30}$$

Therefore, $\boldsymbol{\xi}$ can be represented in terms of white noise $\mathbf{z}$ by a mapping Λ:

$$\boldsymbol{\xi} = \Lambda(\mathbf{z}) = \mathbf{L}\mathbf{z}\|\mathbf{z}\|^{\frac{2}{q}-1} \tag{4.31}$$

and its inverse can be solved as follows

$$\mathbf{z} = \Lambda^{-1}(\boldsymbol{\xi}) = \mathbf{L}^{-1}\boldsymbol{\xi}\|\mathbf{L}^{-1}\boldsymbol{\xi}\|^{\frac{q}{2}-1} \tag{4.32}$$

Therefore, I propose the following representation of $u(\mathbf{x},t)$ in terms of infinite sequence of white noises, i.e. $z := \{z_\ell(\cdot)\}_{\ell=1}^{\infty}$:

$$u(\mathbf{x},t) = T(z) = \sum_{\ell=1}^{\infty}\gamma_\ell\Lambda(z_\ell(t))\phi_\ell(\mathbf{x}), \quad z_\ell(\cdot) \overset{iid}{\sim} \mathcal{GP}(0,\mathcal{I}) \tag{4.33}$$

Denote $\mathbf{Z} = [z_1(\mathbf{t}), \cdots, z_L(\mathbf{t})]_{J\times L}$. Currently, the solution for $\mathbf{Z}_{\text{MAP}} = T^{-1}(\mathbf{U}_{\text{MAP}})$ needs to be determined. Let's identify the appropriate approach to achieve this objective. From (4.33), $\mathbf{U} = T(\mathbf{Z}) = \mathbf{\Phi}\,\text{diag}(\boldsymbol{\gamma})\Lambda(\mathbf{Z})^{\mathsf{T}}$. When $L \leq I$, solve $\Lambda(\mathbf{Z}) = \mathbf{U}^{\mathsf{T}}\mathbf{\Phi}\,\text{diag}(\boldsymbol{\gamma}^{-1})$ which can be further solved column by column with (4.32):

$$\mathbf{Z}_{\text{MAP}} = T^{-1}(\mathbf{U}_{\text{MAP}}) = [\Lambda^{-1}(\mathbf{U}_{\text{MAP}}^{\mathsf{T}}\boldsymbol{\phi}_1\gamma_1^{-1}), \cdots, \Lambda^{-1}(\mathbf{U}_{\text{MAP}}^{\mathsf{T}}\boldsymbol{\phi}_L\gamma_L^{-1})] \tag{4.34}$$

where $\boldsymbol{\phi}_\ell = \phi_\ell(\mathbf{X})$.

4.4.2 White Noise MCMC

Denote the measure formed by the infinite product of $\mathcal{GP}(0, \mathcal{I})$ as ν. Then the STBP prior measure Π can be regained by the pushforward using T, i.e. $\Pi = T^\sharp\nu$. A class of dimension-independent MCMC algorithms for Gaussian prior based models including preconditioned Crank-Nicolson (pCN) [31], infinite-dimensional Metropolis adjusted Langevin algorithm (∞-MALA) [13], infinite-dimensional Hamiltonian Monte Carlo (∞-HMC) [10], and infinite-dimensional manifold MALA (∞-mMALA) [11] and HMC (∞-mHMC) [12] can be reintroduced to posterior sampling with STBP prior.

Let $u = T(z)$ with $z \sim \nu$. Recall we have continuous-time Hamiltonian dynamics from (2.5). More generally, set $\mathcal{K}(z)^{-1} = \mathcal{I} + \beta\mathcal{H}(z)$ where $\mathcal{H}(z)$ can be chosen as Hessian, Gauss-Newton Hessian, or Fisher information operator [90]. For example, choose the Gauss-Newton Hessian computed as $\mathcal{H}(z) = dT^*\mathcal{H}(u)dT$ with dT being the Jacobian. Let $g(z) := -\mathcal{K}(z)\{\alpha\nabla\Phi(z) - \beta\mathcal{H}(z)z\}$ where $\nabla_z\Phi(z) = dT^*\nabla_u\Phi(u) - \nabla_z \log|dT(z)|$. Equation (2.6) gives rise to the leapfrog map $\Psi_\varepsilon : (z_0, \zeta_0) \mapsto (z_\varepsilon, \zeta_\varepsilon)$. Given a time horizon τ and current position z, the MCMC mechanism proceeds by

concatenating $I = \lfloor \tau/\varepsilon \rfloor$ steps of leapfrog map consecutively,

$$z' = \mathcal{P}_z\big\{\Psi_\varepsilon^I(z,\zeta)\big\}\ , \quad \zeta \sim \mathcal{N}(0, \mathcal{K}(z))\ .$$

where $\mathcal{P}_z$ denotes the projection onto the z-argument. Then, the proposal z' is accepted with probability $a(z,z') = 1 \wedge \exp(-\Delta E(z,\zeta))$, where

$$\begin{aligned}\Delta E(z,\zeta) =& E(\Psi_\varepsilon^I(z,\zeta)) - E(z,\zeta)\\ =& \Phi(z_I) - \Phi(z_0) + \frac{\beta}{2}\langle \zeta_I, \mathcal{H}(z_I)\zeta_I\rangle - \frac{\beta}{2}\langle \zeta_0, \mathcal{H}(z_0)\zeta_0\rangle\\ &- \log|\mathcal{K}^{-\frac{1}{2}}(z_I)| + \log|\mathcal{K}^{-\frac{1}{2}}(z_0)| - \frac{\varepsilon^2}{8}\left\{\|g(z_I)\|^2 - \|g(z_0)\|^2\right\}\\ &- \frac{\varepsilon}{2}\sum_{i=0}^{I-1}(\langle g(u_i), \zeta_i\rangle + \langle g(u_{i+1}), \zeta_{i+1}\rangle)\end{aligned} \tag{4.35}$$

At last, I convert the sample z back to $u = T(z)$. This yields ∞-mHMC [12] which reduces to ∞-HMC [10] when $\beta = 0$. Different step-sizes could be used in (2.6): ε_1 for the first and third equations, and ε_2 for the second equation and let $I = 1$, $\varepsilon_1^2 = h, \cos\varepsilon_2 = \frac{1-h/4}{1+h/4}, \sin\varepsilon_2 = \frac{\sqrt{h}}{1+h/4}$. Then, ∞-HMC reduces to ∞-MALA, which can also be derived from Langevin dynamics [13, 12]. When $\alpha = 0$, ∞-MALA further reduces to pCN [12]. Summarize all the above methods in Algorithm 2 and name them as *white-noise dimension-independent MCMC (wn-∞-MCMC)*.

4.5 Numerical Experiments

This section intends to showcase the advantages of the suggested Q-EP(Subsection 4.5.1) and STBP(Subsection 4.5.2) algorithms through their implementation in various problem-solving scenarios.

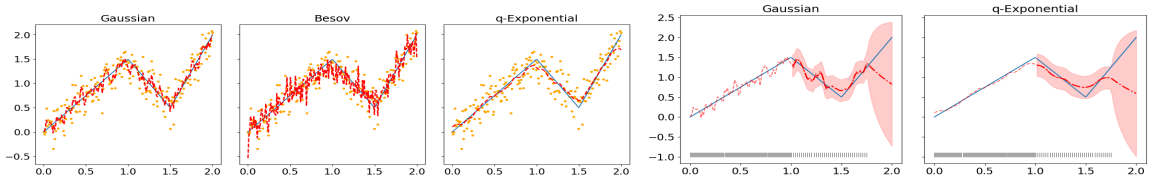
Algorithm 2 White-noise dimension-independent MCMC (wn-∞-MCMC)

Initialize current state $u^{(0)}$ and transform it into the whitened space $z^{(0)} = T^{-1}(u^{(0)})$

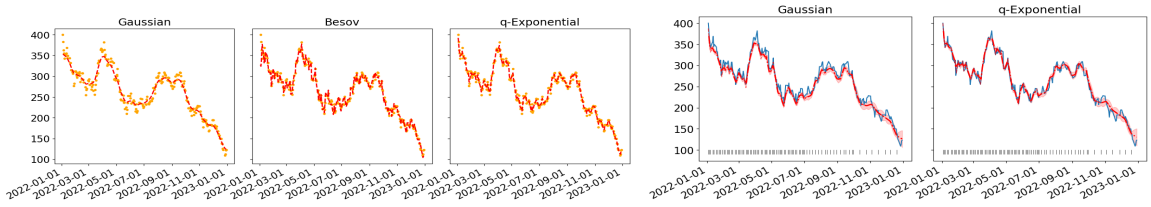
Sample velocity $\zeta^{(0)} \sim \mathcal{N}(0, I)$
Calculate current energy $E_0 = \Phi(z^{(0)}) - \frac{\varepsilon^2}{8}\|g(z^{(0)})\|^2 + \frac{1}{2}\log|\mathcal{K}(z^{(0)})|$
for $i = 0$ to $I - 1$ **do**
    Run $\Psi_\varepsilon : (z^{(i)}, \zeta^{(i)}) \mapsto (z^{(i+1)}, \zeta^{(i+1)})$ according to (2.6).
    Update the energy $E_0 \leftarrow E_0 + \frac{\varepsilon}{2}(\langle g(u^{(i)}), \zeta^{(i)}\rangle + \langle g(u^{(i+1)}), \zeta^{(i+1)}\rangle)$
end for
Calculate new energy $E_1 = \Phi(z^{(I)}) - \frac{\varepsilon^2}{8}\|g(z^{(I)})\|^2 + \frac{1}{2}\log|\mathcal{K}(z^{(I)})|$
Calculate acceptance probability $a = \exp(-E_1 + E_0)$
Accept $z^{(I)}$ with probability a for the next state z' or set $z' = z^{(0)}$.
Record the next state $u' = T(z')$ in the original space.

4.5.1 Experiments with Q-EP

In this subsection, I compare GP, Besov and Q-EP by modeling time series (temporal) and reconstructing images (spatial) from computed tomography. These numerical experiments demonstrate that the proposed Q-EP enables faster convergence in obtaining a better MAP estimate. Moreover, white-noise MCMC-based inference provides appropriate uncertainty quantification (UQ) (by the posterior standard deviation). All the computer codes will be publicly available at https://github.com/lanzithinking/Q-EXP.



(a) Time series with sharp turnings(model fitting). (b) Time series with turnings(prediction).



(c) 2022 Tesla stock prices(model fitting). (d) 2022 Tesla stock prices(prediction).

Figure 33. (a)(c) MAP estimates by GP (left), Besov (middle) and Q-EP (right) models. (b)(d) Predictions by GP (left) and Q-EP (right) models. Orange dots are actual realizations (data points). Blue solid lines are true trajectories. Black ticks indicate the training data points. Red dashed lines are MAP estimates. Red dot-dashed lines are predictions with shaded region being credible bands.

4.5.1.1 Time Series Modeling

I first consider two simulated time series, one with step jumps and the other with sharp turnings, whose true trajectories are as follows:

$$u_{\mathrm{J}}(t) = 1, \quad t \in [0,1]; \quad 0.5, \quad t \in (1,1.5]; \quad 2, \quad t \in (1.5,2]; \quad 0, \quad otherwise$$

$$u_{\mathrm{T}}(t) = 1.5t, \quad t \in [0,1]; \quad 3.5-2t, \quad t \in (1,1.5]; \quad 3t-4, \quad t \in (1.5,2]; \quad 0, \quad otherwise$$

Generate the time series $\{y_i\}$ by adding Gaussian noises to the true trajectories evaluated at $N = 200$ evenly spaced points $\{t_i\}$ in $[0,2]$, that is, $y_i^* = u_*(t_i) + \varepsilon_i$, $\varepsilon_i \overset{ind}{\sim} N(0, \sigma_*^2(t_i))$, $i = 1, \cdots, N$, $* = \mathrm{J}, \mathrm{T}$. Let $\sigma_{\mathrm{J}}/\|u_{\mathrm{J}}\| = 0.015$ for $t_i \in [0,2]$ and $\sigma_{\mathrm{T}}/\|u_{\mathrm{T}}\| = 0.01$ if $t_i \in [0,1]$; 0.07 if $t_i \in (1,2]$. In addition, I also consider two real data sets of Tesla and Google stock prices in 2022. See Figures 33 (and Figures B.11) for the true trajectories (blue lines) and realizations (orange dots) respectively.

Use the above likelihood and test three priors: GP, Besov and Q-EP. For Besov,

choose the Fourier basis $\phi_0(t) = \sqrt{2},\ \phi_\ell(t) = \cos(\pi\ell t),\ \ell \in \mathbb{N}$. For both GP and Q-EP, adopt the Matérn kernel with $\nu = \frac{1}{2}$, $\sigma^2 = 1$, $\rho = 0.5$ and $s = 1$: $C(t,t') = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} w^\nu K_\nu(w), \quad w = \sqrt{2\nu}(\|t-t'\|/\rho)^s$. In both Besov and Q-EP, set $q = 1$. Figures 33a and 33c (and Figures B.11a and B.11c) compare the MAP estimates (red dashed lines). It's clear that Q-EP yields the best estimates closest to the true trajectories in the simulation and the best fit to the Tesla/Google stock prices. I also investigate the negative posterior densities and relative errors, $\|\hat{u}_* - u_*\|/\|u_*\|$, as functions of iterations in Figure B.10. Though incomparable in the absolute values, the negative posterior densities indicate faster convergence in both GP and Q-EP models than in Besov model. The error reducing plots on the right panels of subplots in Figure B.10 indicate that the Q-EP prior model can achieve the smallest errors. Table 7 compares them in terms of root mean of squared error (RMSE) and log-likelihood (LL).

Table 7. Time series modeling: root mean of squared errors (RMSE) and log-likelihood (LL) values at MAP estimates by GP, Besov and Q-EP prior models.

	RMSE			log-likelihood (LL)		
Data Sets	GP	Besov	Q-EP	GP	Besov	Q-EP
simulation(jumps)	1.2702	2.1603	**1.1083**	-31.4582	-89.8549	-74.0590
simulation(turnings)	1.4270	2.4556	**0.9987**	-39.8234	-56.7874	-87.3124
Tesla stocks	180.3769	136.8769	**51.2236**	-488.6458	-281.3796	-39.4070
Google stocks	44.4236	39.4809	**36.8686**	-386.1546	-305.0058	-265.9790

Next, let's examine the prediction problem. In the simulations, the last 1/8 portion and every other of the last but 3/8 part of the data points are selected for testing. The models with GP and Q-EP priors are trained on the rest of the data, as indicated by short "ticks" in Figures 33b and 33d (and Figures B.11b and B.11d). For the Tesla/google stocks, select every other day in the first half year, every 4 days in the 3rd quarter and every 8 days in the last quarter for training and test on the rest.

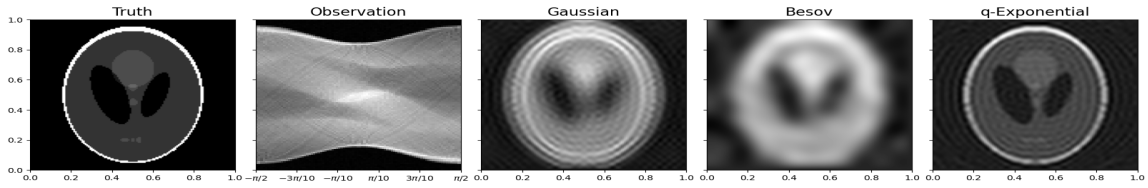


Figure 34. Shepp-Logan phantom: true image, observation (sinogram), and MAP estimates by GP, Besov and Q-EP models with relative errors 68.10%, 70.27% and 40.87% respectively.

They pose challenges on both interpolation (among observations) and extrapolation (at no-observation region) tasks. As we can see in those figures, uncertainty grows as the data becomes scarce. Nevertheless, the Q-EP yields smaller errors than GP. Note such prediction is not immediately available for models with Besov prior.

4.5.1.2 Computed Tomography Imaging

CT is a medical imaging technique used to obtain detailed internal images of the human body. CT scanners use a rotating X-ray tube and a row of detectors to measure X-ray attenuations by different tissues inside the body from different angles. Denote the true imaging as a function $u(x)$ on the square unit $D = [0, 1]^2$ taking values as the pixels. The observed data, $\mathbf{y}$, (a.k.a. sinogram) are results of Radon transformation ($\mathbf{A}$) of the discretized $n \times n$ field $\mathbf{u}$ with n_θ angles and n_s sensors, contaminated by noise $\boldsymbol{\varepsilon}$ [8]:

$$\mathbf{y} = \mathbf{A}\mathbf{u} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_\varepsilon^2 \mathbf{I}), \quad \mathbf{y} \in \mathbb{R}^{n_\theta n_s}, \quad \mathbf{A} \in \mathbb{R}^{n^2 \times n_\theta n_s}, \quad \mathbf{u} \in \mathbb{R}^{n^2}$$

In general, $n_\theta n_s \ll d = n^2$, so the linear inverse problem is under-determined. The Bayesian approach could fill useful prior information (e.g., edges) in the sparse data.

Consider the Shepp–Logan phantom, a standard test image created by Shepp and

Logan in [142] to model a human head and to test image reconstruction algorithms. In this simulation, I create the true image $u^\dagger$ for a resolution of $n^2 = 128 \times 128$ and project it at $n_\theta = 90$ angles with $n_s = 100$ equally spaced sensors. The generated sinogram is then added by noise with signal noise ratio SNR $= \|\mathbf{A}u^\dagger\|/\|\boldsymbol{\varepsilon}\| = 100$. The first two panels of Figure 34 show the truth and the observation.

Table 8. Posterior estimates of Shepp–Logan phantom by GP, Besov and Q-EP prior models: relative error, RLE $:= \|\hat{u} - u^\dagger\|/\|u^\dagger\|$, of MAP ($\hat{u} = u^*$) and posterior mean ($\hat{u} = \overline{u}$) respectively, log-likelihood (LL), PSNR, SSIM and HarrPSI. Numbers in the bracket are standard deviations obtained by repeating the experiments 10 times with different random seeds.

	MAP			Posterior Mean		
	GP	Besov	Q-EP	GP	Besov	Q-EP
RLE	0.6810	0.7027	**0.4087**	0.4917(6.16e-7)	0.4894(3.53e-5)	**0.4890**(4.79e-5)
LL	-1.55e+6	-1.54e+6	-1.57e+5	-5.21e+5(8.47)	-4.80e+5(196.34)	-4.56e+5(307.97)
PSNR	15.5531	15.2806	**19.9887**	18.3826(1.09e-5)	18.4226(6.27e-4)	**18.4303**(8.51e-4)
SSIM	0.4028	0.3703	**0.5967**	**0.5561**(3.92e-7)	0.5535(2.38e-4)	0.5403(5.26e-4)
HaarPSI	0.0961	0.0870	**0.3105**	0.3126(1.52e-8)	**0.3126**(3.36e-4)	0.3122(3.06e-4)

Note the computation involving a full-sized $(d \times d)$ kernel matrix $\mathbf{C}$ for GP and Q-EP is prohibitive. Therefore, I consider Mercer's expansion (4.17) for a truncation with the first $L = 2000$ items. Figure 34 shows that while GP and Besov models reconstruct very blurry phantom images, the Q-EP prior model produces the highest quality MAP estimate. For each of the three models, I also apply wn-pCN to generate 10000 posterior samples (after discarding 5000) and use them to reconstruct u (posterior mean or median) and quantify uncertainty (posterior standard deviation).

Table 8 summarizes the errors relative to MAP (u^*) and posterior mean ($\overline{u}$) respectively, $\|\hat{u} - u^\dagger\|/\|u^\dagger\|$ (with $\hat{u}$ being u^* or $\overline{u}$), log-likelihood (LL), and several quality metrics in imaging analysis, including the peak signal-to-noise ratio (PSNR) [52], the structured similarity index (SSIM) [157], and the Haar wavelet-based perceptual

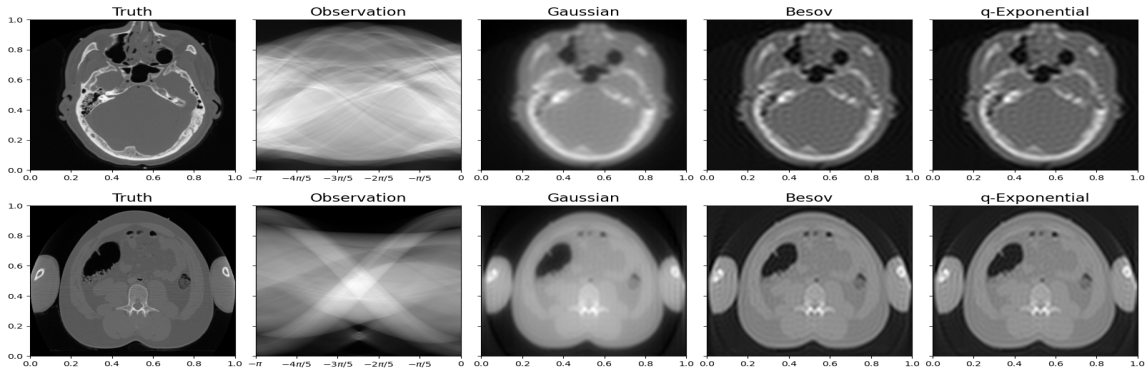


Figure 35. CT of human head (upper) and torso (lower): true image, observation (sinogram), and MAP estimates by GP, Besov and Q-EP models with relative errors 29.99%, 22.41% and 22.24% (for head) and 26.11%, 21.77% and 21.53% (for torso) respectively.

similarity index (HaarPSI) [130]. Q-EP attains the lowest error and highest quality scores in most cases. In Figure B.12, I compare the uncertainty of these models. It seems that GP has uncertainty field with a more recognizable shape than the other two. However, the posterior standard deviation by GP is much smaller (about 1% of that with Q-EP) compared with the other two. Therefore, this raises a red flag that GP could be over-confident about a less accurate estimate.

Finally, I apply these methods to CT scans of a human cadaver and torso from the Visible Human Project [3]. These images contain $n^2 = 512 \times 512$ pixels, and the sinograms are obtained with $n_\theta = 200$ angles and $n_s = 512$ sensors. The first two panels of each row in Figure 35 show a highly calibrated CT reconstruction (treated as "truth") and the observed sinogram. The rest three panels illustrate that both Besov and Q-EP models outperform GP in reconstructions, as verified in the quantitative summaries in Table B.2. Figure B.13 indicates that GP underestimates the uncertainty.

4.5.2 Spatiotemporal Experiments with STBP

In this subsection, I compare the proposed STBP with STGP and a time-uncorrelated approach (for which we set $\mathcal{C} = \mathcal{I}$ in STBP) using two dynamic tomography imaging examples and one inverse problem of recovering a spatiotemporal function. The numerical results demonstrate the advantage of Besov-type priors over Gaussian-type priors in reconstructing images with edges. Moreover, these examples highlight the importance of temporal correlations in dynamic imaging analysis and spatiotemporal inverse problems.

To assess the quality of image reconstruction (view the inverse solutions defined on 2d space as images), I refer to several quantitative measures such as the relative error (RLE), $\text{RLE} = \frac{\|u^* - u^\dagger\|_2}{\|u^\dagger\|_2}$, where $u^\dagger$ denotes the reference/true image and u^* its reconstruction. Additionally, I adopt the peak signal-to-noise ratio (PSNR), $\text{PSNR} = 10 * \log_{10}(\frac{\|u^\dagger\|_\infty^2}{\|u^* - u^\dagger\|_2^2})$, by using the maximum possible pixel value(MAX_I^2) as a reference point to normalise the MSE. Another option to consider is the structured similarity index (SSIM) [157], $\text{SSIM}(u^*, u^\dagger) = \frac{(2\overline{u^*}\,\overline{u^\dagger} + c_1)(2s_{u^* u^\dagger} + c_2)}{(\overline{u^*}^2 + \overline{u^\dagger}^2 + c_1)(s_{u^*}^2 + s_{u^\dagger}^2 + c_2)}$, where $\overline{u}$, s_u^2 and $s_{u_1 u_2}$ denote the sample mean, sample variance, and sample covariance respectively, $c_i = (k_i L)^2$ for $i = 1, 2$, $k_1 = 0.01$, $k_2 = 0.03$ and L is the dynamic range of the pixel values of the reference images. Lastly, I report the Haar wavelet-based perceptual similarity index (HaarPSI) proposed in [130]. It is an innovative and computationally affordable image quality assessment method that uses Haar wavelet-based decomposition to measure local similarities and the relative importance of image areas. The validation on four extensive benchmark databases confirms its alignment with human perception, ensuring greater consistency.

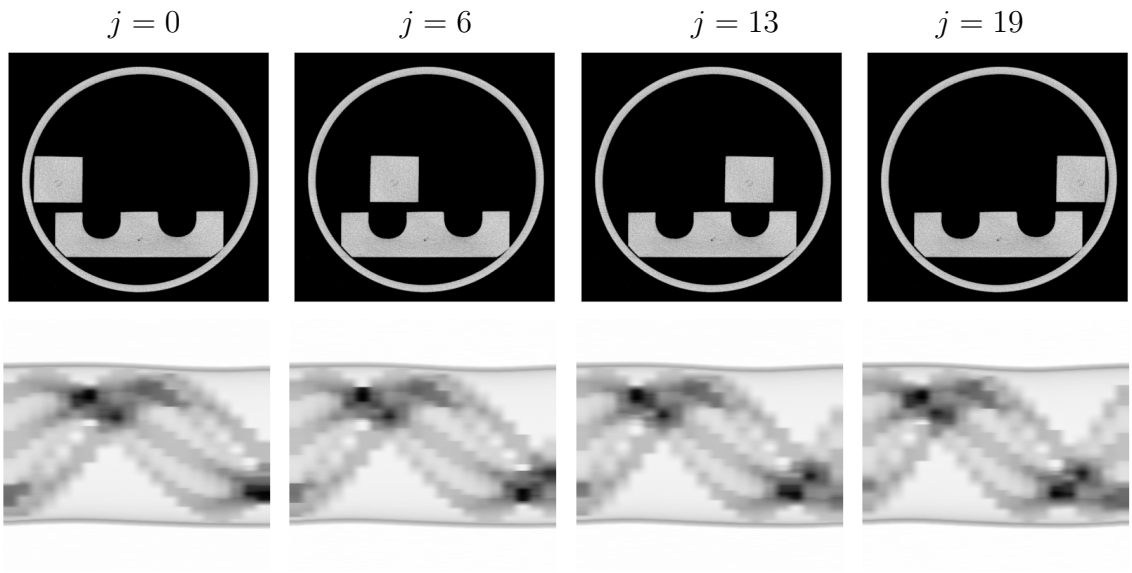


Figure 36. Test 1: STEMPO test problem. True images (first row) and sinograms (second row) from left to right at time instances $j = 0, 6, 13, 19$, respectively.

4.5.2.1 STEMPO Tomography Reconstruction

I investigate STBP, STGP and time-uncorrelated prior models on a simulated dynamic tomography reconstruction problem in this example. In particular, consider the Spatio-TEmporal Motor-POwered (STEMPO) ground truth phantom from [65], `stempo_ground_truth_2d_b4.mat` that contains 360 images. From the dataset, obtain $J = 20$ images of size 560×560 chosen uniformly from 1 to 360 with a factor of 8, i.e., I choose the 1st, the 8th, the 16th up to the 360th image that represents the truth at 20 time instances. Using the ASTRA toolbox [151] I generate the forward operators $\mathcal{G}_j$, $j = 1, 2, \ldots, J$ by considering J vectors of length 11 containing projection angles. Each angles vector is generated by choosing 11 equispaced degree angles from $(5 * (j-1), 5 * (j-1) + 140)$, for $j = 1, 2, \ldots, J$ that are then converted to radian.

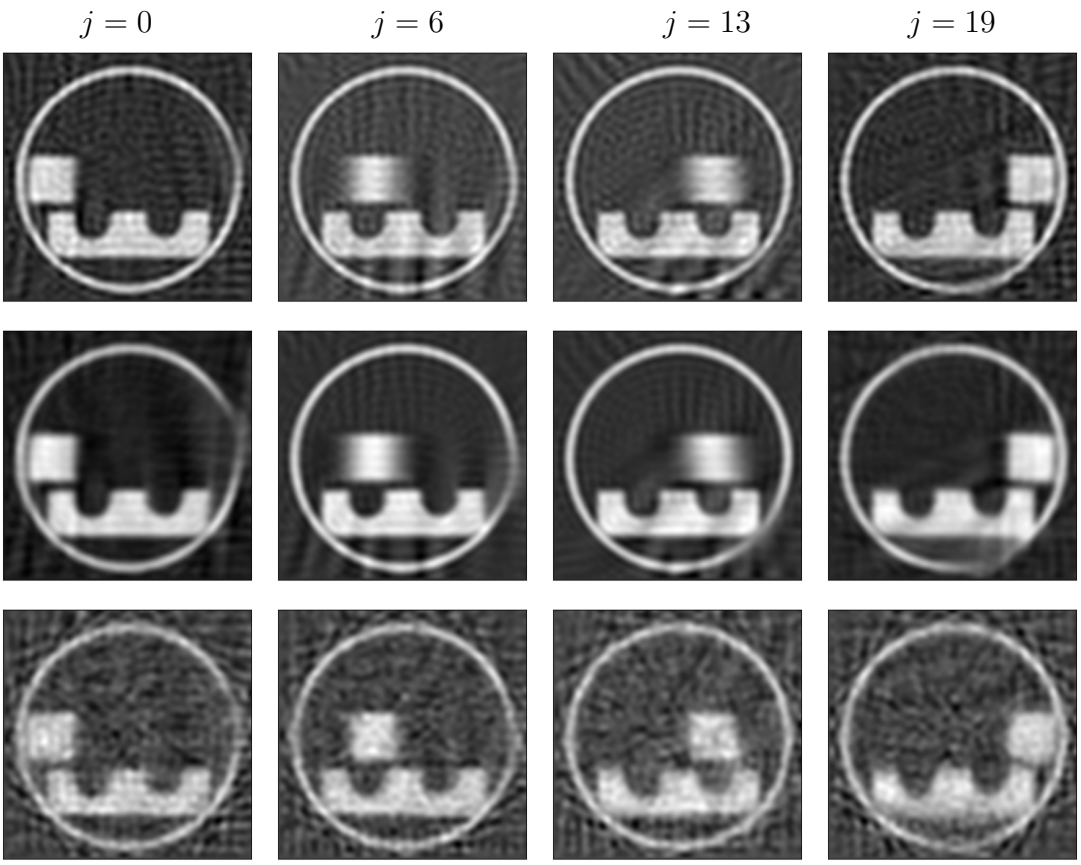


Figure 37. Reconstruction results of dynamic STEMPO test problem in the whitened space. Row from top to bottom: MAP for STBP ($q = 1, p = 1$), STGP ($q = 2, p = 2$) and time-uncorrelated model. Left to right: time step $j = 0, 6, 13, 19$.

Throughout the thesis, I denote the number of angles used to generate the forward problem with n_a. For this example, set $n_a = 20$.

To generate the forward operators using ASTRA, we provide the additional parameters listed below. Choose origin to detector distance (detector_origin) to be $3 * n_x$, the source to origin distance (source_origin) to be n_x, and the detector pixel size to be computed as $\frac{(\text{source\_origin}+\text{detector\_origin})}{\text{source\_origin}}$. Each forward operator $\mathcal{G}_j \in \mathbb{R}^{8701\times 313600}$ and the large operator can be represented by a block-diagonal matrix $\overline{\mathcal{G}} = \text{diag}(\mathcal{G}_1, \cdots \mathcal{G}_j) \in \mathbb{R}^{174020\times 6272000}$. Apply the forward operators $\mathcal{G}_j$, to the $j - th$

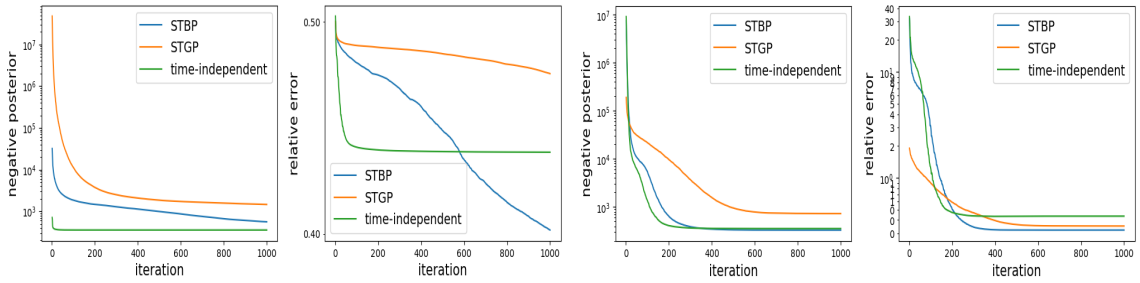


Figure 38. Dynamic STEMPO tomography: negative posterior densities and relative errors for the optimization in the original space (left) and in the whitened space (right) as functions of iterations in the BFGS algorithm used to obtain MAP estimates. Early termination is implemented if the error falls below some threshold or the maximal iteration (1000) is reached.

true images $u^{\text{true}}(\mathbf{X}, t_j)$), to obtain J sinograms $y_j \in \mathbb{R}^{8701}$, with $\mathbf{Y}_j \in \mathbb{R}^{791\times 11}$, for $j = 1, 2, \ldots, J$. Assume that the noise vector follows a multivariate normal Gaussian distribution with mean zero and covariance $\mathbf{\Gamma}_{\text{noise}}$, i.e., $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_{\text{noise}})$. I perturb each measured vectorized sinogram y_j^{true} with white Gaussian noise, i.e., the noise vector ε_j has mean zero and a rescaled identity covariance matrix (i.e., $\Gamma_{\text{noise}} = \mathbf{I}$, where $\mathbf{I} \in \mathbb{R}^{313600\times 313600}$). Refer to the ratio $\sigma_j = \|\varepsilon_j\|_2 / \|\mathcal{G}_j\left(u^{\text{true}}(\mathbf{X}, t_j)\right)\|_2$ as the noise level. The true images $u^{\text{true}}(\mathbf{X}, t_j)$) at time steps $j = 0, 6, 13, 19$ are shown in the first rows of Figure 36. Observed noisy sinograms are shown in the second row of Figure 36.

To obtain the MAP estimates, I minimize negative log-posterior densities for the three models with STBP, STGP, and time-uncorrelated priors. Figure 37 compares these MAP estimates obtained in the whitened space and mapped to the original space. With STBP, it is evident that the first-row reconstruction is the most precisely aligned with the truth. Nevertheless, the results of the other two models are either blurry (by STGP on the second row) or noisy (by the time-uncorrelated model on the

Table 9. MAP estimates of STEMPO by STBP, STGP and time-uncorrelated prior models. Relative error of MAP (u^*), RLE, log-likelihood, PSNR, SSIM, and HaarPSi measures. Standard deviations are obtained by repeating the experiments 10 times with different random seeds.

	time-uncorrelated	STGP	STBP
RLE	0.4354 (2.91e-5)	0.3512(1.42e-4)	**0.3217** (2.72e-5)
log-likelihood	-39190.72 (0.65)	-39085.37 (5.49)	**-39697.93** (0.71)
PSNR	15.8250	18.7639	**19.5753**
SSIM	0.9916	0.9968	**0.9977**
HaarPSI	0.2751	0.4088	**0.4983**

last row). Table 9 confirms that the STBP model yields higher reconstruction quality with the lowest relative error. Though their log-likelihood values are not comparable in the regularized optimization, STBP achieves the lowest RLE = 32.17% on average in 10 experiments repeated with different random seeds. Other reconstruction quality measures such as PSNR, SSIM, and HaarPSI are shown in Table 9, rows 3-5. The other measures also reflect higher reconstruction quality observed in RLE.

On the other hand, the MAP estimates generated by these three models in the original space are illustrated in Figure B.14. They are more than 40% RLE's and are generally more blurry than those obtained in the whitened space. Such difference can also be seen in Figure 38 where the objective functions and RLE's are compared between the whiten space optimization (left two panels) and the original space optimization (right two panels) for these three models: optimization in the whitened space outputs better results with lower errors and fewer iterations. STBP generally converges faster to the lowest error state among the three models.

Lastly, I apply the white-noise manifold infinite-dimensional MALA (wn-minfMALA) algorithm to sample the posterior samples for the two models with STBP and STGP priors (the result for time-uncorrelated prior is far worse and hence

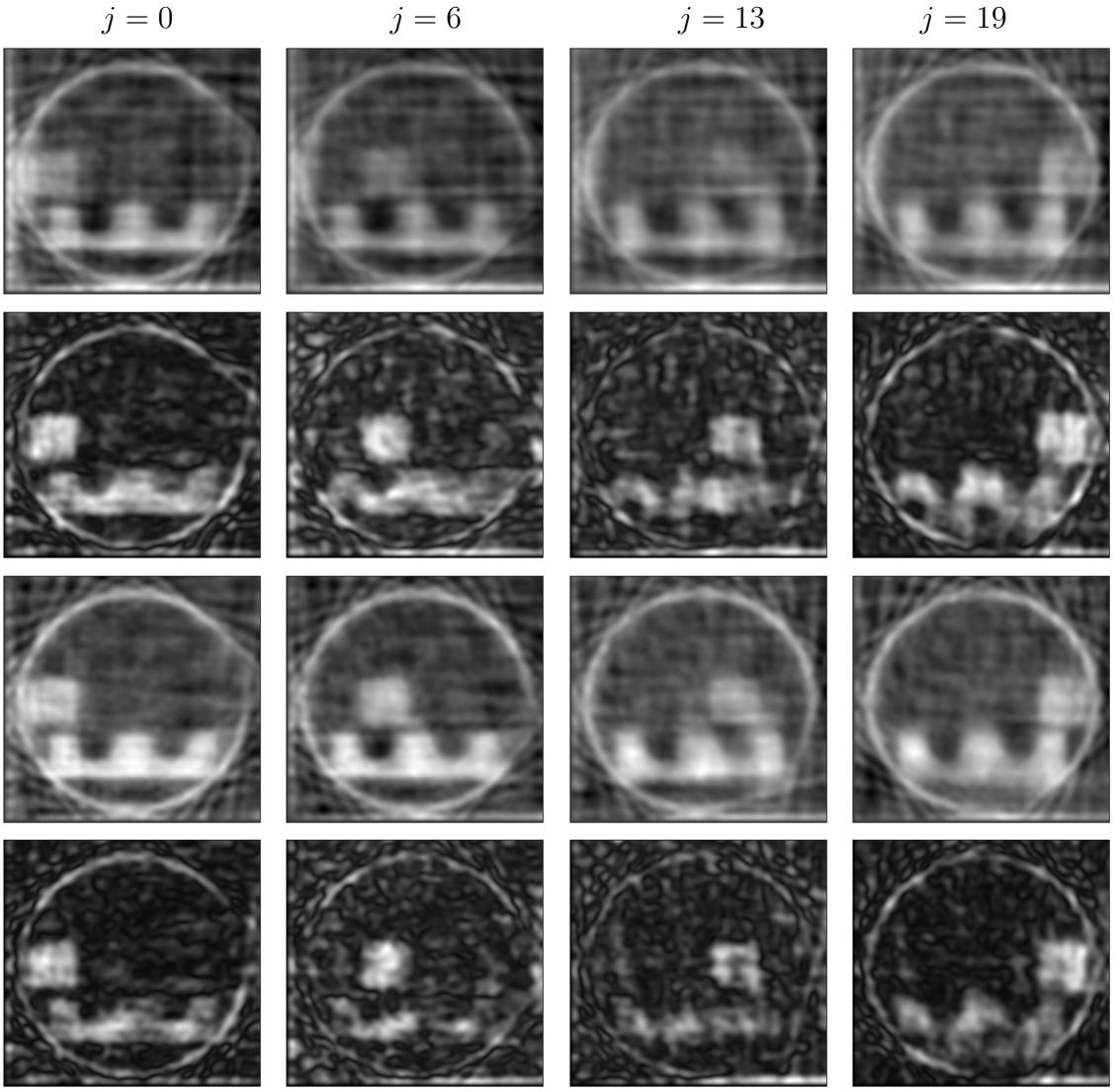


Figure 39. MCMC results of dynamic STEMPO test problem in the whitened space. Row from top to bottom: posterior mean for STBP ($q = 1, p = 1$), posterior standard deviation for STBP ($q = 1, p = 1$), posterior mean for STGP ($q = 2, p = 2$), and posterior standard deviation for STGP ($q = 2, p = 2$). Left to right: time step $t = 0, 6, 13, 19$.

omitted) and compare their posterior estimates in Figure 39. Generate 3000 samples and discard the first 1000 samples. The remaining 2000 samples are used to estimate the posterior means (the first and third rows) and posterior standard deviations (the second and the last rows). Due to the large dimensionality ($560 \times 560 \times 20$) and the limited number of samples, these posterior estimates tend to be noisy. The posterior mean estimates are not good reconstructions as their MAP estimates. Yet the posterior standard deviations by STBP (the second row) provide more clear uncertainty information compared with those by the STGP model (the last row).

4.5.2.2 Emoji Tomography Reconstruction

In this example, test our methods on real data of an "emoji" phantom measured at the University of Helsinki [112]. The forward operator and the data can be obtained from the file `DataDynamic_128x30.mat`. The available data represents $J = 33$ time steps of a series of the X-ray sinogram of emojis made of small ceramic stones obtained by shining 217 projections from $n_a = 10$ angles.

The goal is to reconstruct a sequence of images $u(\mathbf{X}, t_j)$, $j = 0, 1, \ldots, 32$, of size $n_x \times n_y$, where $n_x = n_x = 128$, from low-dose observations measured from a limited number of angles n_a. Hence, the unknown images are collected in $u \in \mathbb{R}^{540,672}$, with $u = \left[(u(\mathbf{X}, t_0)^T, u(\mathbf{X}, t_1)^T, \ldots, u(\mathbf{X}, t_{32})^T\right]^T$ representing the dynamic sequence of the emoji changing from an expressionless face with closed eyes and a straight mouth to a face with smiling eyes and mouth, where the outmost circular shape does not change. Refer to Figure 40 for a sample of 4 setup images (first row) and sinograms (second row) at time steps $j = 6, 14, 22, 30$. The low-dose available observations can be modeled by the measurement matrix $\mathcal{G}$, which describes the forward model of the

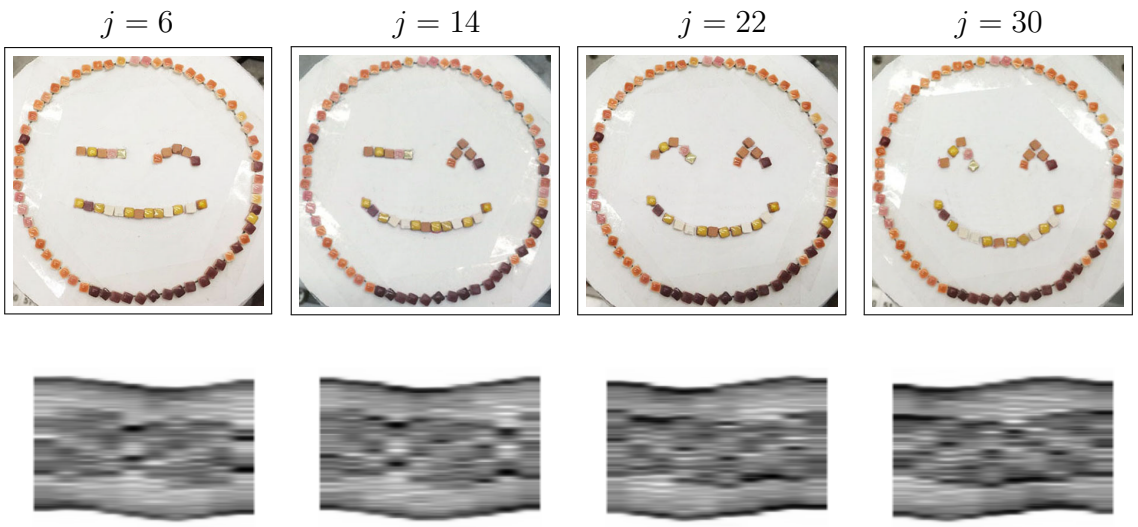


Figure 40. Test 2: Emoji test problem. Setup images (first row) and sinograms (second row) from left to right at time instances $j = 6, 14, 22, 30$, respectively.

Radon transform that represents line integrals. In this case, I have a block-diagonal matrix with 33 blocks. Although the ground truth is unavailable, I can qualitatively compare the visual results.

Figure 41 compares the MAP estimates by STBP (the first row), STBP (the second row), and the time-uncorrelated (the last row) prior models in the whitened space. Again I observe a similar advantage in reconstructing a sequence of sharper tomography images by STBP compared with those more blurry results by STGP. By ignoring the temporal correlation, the time-uncorrelated prior model yields reconstruction images that are difficult to recognize.

I also compare the UQ results generated by white-noise manifold MALA for STBP and STGP two models in Figure B.16. Again the noisy posterior mean estimates are observed for both models. However, the posterior standard deviation estimates by

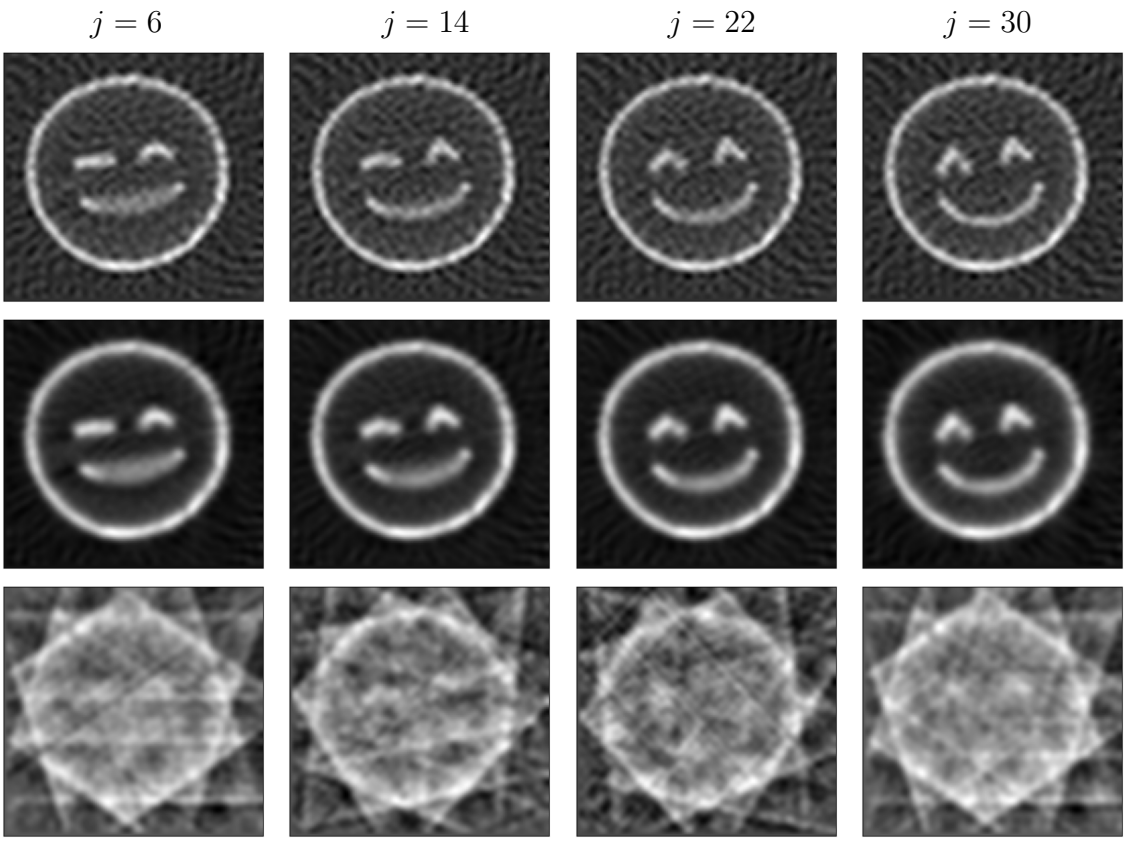


Figure 41. Reconstruction results for the emoji problem with $n_a = 10$ in the whitened space. Row from top to bottom: MAP for STBP ($q = 1, p = 1$), MAP for STGP ($q = 2, p = 2$), and MAP for time-uncorrelated model. Left to right: time step $t = 6, 14, 22, 30$.

STBP are slightly more precise than STGP's in characterizing the uncertainty field representing the changing smiling faces.

4.6 Conclusion

To summarize, I propose a computational framework for Bayesian inverse problems to construct spatiotemporal solutions. I generalize series-based Besov measure for spatial functions to a nonparametric prior for spatiotemporal functions, hence named spatiotemporal Besov process (STBP). I firstly propose the Q-EP as a prior on L^q

functions. This approach allows for a customizable parameter $q > 0$, which provides a higher degree of control over the regularization level. To capture the temporal correlations of multiple images, the STBP can be created by using Q-EP to replace the univariate q-exponential random variables in the series-defining Besov process. The proposed STBP simultaneously models the inhomogeneity in space and correlation in time in the spatiotemporal target of Bayesian inverse problems.

To address the challenges in high-dimensional posterior sampling with non-Gaussian priors, I take advantage of the existing literature on dimension-independent MCMC algorithms for Gaussian priors and propose a white-noise representation of the STBP. The derived white-noise MCMC provides robust and efficient inference for Bayesian models with Q-EP and STBP priors.

Extensive experiments have been investigated to show the advantage of Q-EP and STBP. Regarding Q-EP, it possesses the ability to enforce more stringent regulation through q in contrast to GP. Furthermore, Q-EP offers an explicit formula that affords greater control over the correlation structure, similar to GP. The numerical experiments in time series modeling and image reconstruction demonstrate the proposed Q-EP is superior in Bayesian functional data modeling.

I show that spatiotemporal Besov priors with a properly chosen covariance kernel outperform the priors with uncorrelated time and yield higher quality reconstructions with edges well-preserved compared to the spatiotemporal Gaussian priors while reducing the posterior uncertainty by conducting numerous numerical experiments using computed tomography with both simulated and real data. Such promising results from real computerized tomography and limited angle data suggest potential applications in medical imaging analysis.

Chapter 5

CONCLUSION AND FUTURE DIRECTIONS

In the contemporary age of abundant data availability, a thorough understanding, accurate modeling, and reliable forecasting of data are essential for conducting comprehensive research. Among various aspects, UQ holds paramount importance as it offers a systematic approach to assess decision-making, predictions, and simulations from various perspectives. Within the realm of scientific studies and engineering applications, the presence of spatiotemporal observations in inverse problems is a common occurrence. However, the process of solving inverse problems linked to these observations can be challenging due to the high dimensionality and nonlinearity of the system. Therefore, two crucial aspects of my thesis involve accurately modeling spatiotemporal data and obtaining UQ for inverse problems in high-dimensional space.

In Chapter 1.2, I begin by giving an overview of the inverse problem and then discuss how to integrate Bayesian methods into it. After learning the architecture of the Bayesian inverse problem, UQ is covered in Chapter 2. The calibration-emulation-sampling (CES) scheme has been proven to be successful in large dimensional UQ problems to solve the issue of traditional Bayesian inference methods based on Markov Chain Monte Carlo (MCMC), which is computationally intensive and inefficient. Therefore I proposed a new framework to scale up Bayesian UQ for physics-constrained inverse problems based on CES. More specifically, I utilize deep neural networks in the emulation and sampling step. CNN is adopted in emulation space to be capable of learning spatial features. In addition, the resulting algorithm has low computational complexity and is robust across different training sizes. Furthermore, I implement AE

to reduce the dimension of the parameter space and speed up the sampling process. Overall, the resulting DREAM algorithm helps to scale Bayesian UQ up to thousands of dimensions. Currently, I use a regular grid mesh to aid in the training of CNN. This involves converting a discretized function over a 2D mesh into a matrix of image pixels. An effective method for addressing the limitations of irregular mesh, like the triangular mesh commonly used for solving PDEs, is through the utilization of mesh CNN [62]. This approach enables direct training of CNN on the irregular mesh, thereby providing a comprehensive solution. An avenue for advancing the methodology entails investigating the potential direction mentioned. Another promising approach involves replacing the use of AE with CAE, which is capable of producing a more discernible latent representation. Figure B.3 serves as evidence of this. In this case, the latent parameter can be interpreted as a representation of the original function on a coarser mesh. Lastly, there are spatiotemporal data in some inverse problems (e.g., advection-diffusion equation). In such cases, I could model the temporal pattern of observations in the emulation using some recurrent neural networks (RNN) [134], e.g., long short-term memory (LSTM) [67]. I can then build a 'CNN-RNN' emulator with the convolutional layer for function (image) inputs and the recurrent layer for multivariate time series outputs. Although I have achieved encouraging preliminary outcomes, I plan to continue exploring this concept in my future endeavors.

Spatiotemporal modeling is discussed in Chapter 3(spatiotemporal likelihood modeling) and in Chapter 4(spatiotemporal prior modeling). In Chapter 3, I implement STGP on the spatiotemporal inverse problem to fully utilize data information that contains the spatiotemporal interactions and show that the spatial and temporal information provides more effective parameter estimation and UQ. In conducting my research, I undertook a comparative analysis of Bayesian spatiotemporal likelihood

modeling for inverse problems, as contrasted with static and time-averaged methods. Specifically, I utilized a time-dependent advection-diffusion PDE in conjunction with three chaotic ODEs to demonstrate the effectiveness of this approach. Moreover, I furnished theoretical support to establish the superiority of STGP in the context of fitting trajectories, even when confronted with chaotic dynamics. Nevertheless, the STGP model (3.13) discussed in this chapter features a classical separation structure in its joint kernel, which may not be adequate to capture complex spatiotemporal relationships, such as the temporal evolution of spatial dependence (TESD) [92]. To address this, I intend to explore non-stationary non-separable STGP models, [32, 166, 156] which have been recommended by previous research as a means of addressing more intricate space-time interactions in spatiotemporal inverse problems.

Chapter 4 generalizes series-based Besov measure for spatial functions to a nonparametric prior for spatiotemporal functions — STBP. I have developed a probabilistic formulation called Q-EP, which is an extension of the q-exponential distribution. This formulation allows for control over the correlation length. The Q-EP method has been shown to be superior in obtaining faster and better reconstruction in numerical experiments for time series modeling and image reconstruction. The Q-EP is employed to replace the univariate q-exponential random variables present in the series which defines the Besov process. This technique enables the capture of temporal correlations of multiple images, and consequently, leads to the formulation of the STBP. The proposed STBP simultaneously models the inhomogeneity in space and correlation in time. To address the challenges in high-dimensional posterior sampling with non-Gaussian priors, I take advantage of the existing literature on dimension-independent MCMC algorithms for Gaussian priors and propose a white-noise representation of the STBP. The derived white-noise MCMC provides robust and efficient inference for

Bayesian models with STBP priors. Through a series of rigorous numerical experiments involving computerized tomography utilizing both simulated and real data, I have made a noteworthy discovery. Specifically, I have found that the implementation of STBP featuring a carefully selected covariance kernel produces superior results compared to priors which lack temporal correlation. This approach yields high-quality reconstructions that effectively preserve edges while simultaneously reducing the level of posterior uncertainty. In contrast, spatiotemporal Gaussian priors deliver comparatively inferior results. The outcomes obtained from authentic CT scans and a restricted angle data set exhibit the viability of this methodology in the analysis of medical imagery. After conducting an in-depth analysis of various numerical examples, it has become evident that implementing the STBP approach offers significant advantages. In order to strengthen our argument, it would be necessary to explore the theory of posterior contraction more deeply. For instance, I could investigate whether STBP provides a superior posterior contraction rate in non-parametric problems and function spaces when compared to a Gaussian prior.

REFERENCES

[1] Sigurd I. Aanonsen et al. "The Ensemble Kalman Filter in Reservoir Engineering—a Review". In: *SPE Journal* 14.03 (Sept. 2009), pp. 393–412. DOI: 10.2118/117274-PA. eprint: https://onepetro.org/SJ/article-pdf/14/03/393/2554039/spe-117274-pa.pdf. URL: https://doi.org/10.2118/117274-PA.

[2] Henry Abarbanel. *Predicting the Future: completing models of observed complex systems*. Vol. 1. 16. Springer New York, 2013. DOI: 10.1007/978-1-4614-7218-6. URL: https://doi.org/10.1007%2F978-1-4614-7218-6.

[3] M.J. Ackerman. "The Visible Human Project". In: *Proceedings of the IEEE* 86.3 (1998), pp. 504–511. DOI: 10.1109/5.662875.

[4] HN Agiza and MT Yassen. "Synchronization of Rossler and Chen chaotic dynamical systems using active control". In: *Physics Letters A* 278.4 (2001), pp. 191–197.

[5] D. F. Andrews and C. L. Mallows. "Scale Mixtures of Normal Distributions". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 36.1 (1974), pp. 99–102. URL: http://www.jstor.org/stable/2984774 (visited on 05/02/2023).

[6] Anatolii Borisovich Bakushinskii. "A general method of constructing regularizing algorithms for a linear incorrect equation in Hilbert space". In: *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7.3 (1967), pp. 672–677.

[7] Maria Bånkestad et al. "The Elliptical Processes: a Family of Fat-tailed Stochastic Processes". In: (Mar. 2020). eprint: 2003.07201. URL: https://arxiv.org/pdf/2003.07201.pdf.

[8] Johnathan M. Bardsley. "Applications of a nonnegatively constrained iterative method with statistically based stopping rules to CT, PET, and SPECT imaging". In: *Electron. Trans. Numer. Anal.* 38 (2011), pp. 34–43.

[9] J. M. Bernardo et al. "Regression and Classification Using Gaussian Process Priors". In: *Bayesian Statistics* 6 (1998), pp. 475–501. DOI: 130.203.136.95/viewdoc/summary?doi=10.1.1.156.1910. URL: http://130.203.136.95/viewdoc/summary?doi=10.1.1.156.1910.

[10] A. Beskos et al. "Hybrid Monte-Carlo on Hilbert spaces". In: *Stochastic Processes and their Applications* 121 (2011), pp. 2201–2230.

[11] Alexandros Beskos. "A stable manifold MCMC method for high dimensions". In: *Statistics & Probability Letters* 90 (2014), pp. 46–52.

[12] Alexandros Beskos et al. "Geometric MCMC for infinite-dimensional inverse problems". In: *Journal of Computational Physics* 335 (2017). URL: http://www.sciencedirect.com/science/article/pii/S0021999116307033.

[13] Alexandros Beskos et al. "MCMC methods for diffusion bridges". In: *Stochastics and Dynamics* 8.03 (2008), pp. 319–350.

[14] Robert Bishop. "Chaos". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Spring 2017. Metaphysics Research Lab, Stanford University, 2017.

[15] Chris Brooks. "Chaos in foreign exchange markets: a sceptical view". In: *Computational Economics* 11.3 (1998), pp. 265–281. DOI: 10.1023/a:1008650024944. URL: https://doi.org/10.1023%2Fa%3A1008650024944.

[16] Tan Bui-Thanh and Omar Ghattas. "A scalable algorithm for MAP estimators in Bayesian inverse problems with Besov priors". In: *Inverse Problems & Imaging* 9.1 (2015), p. 27.

[17] Nicholas W. Bussberg. "Spatio-Temporal Statistics With R." In: *The American Statistician* 75.1 (2021), pp. 114–114. DOI: 10.1080/00031305.2020.1865066. eprint: https://doi.org/10.1080/00031305.2020.1865066. URL: https://doi.org/10.1080/00031305.2020.1865066.

[18] D. Calvetti and E. Somersalo. *An Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*. Vol. 2. Springer Science & Business Media, 2007.

[19] Stamatis Cambanis, Steel Huang, and Gordon Simons. "On the theory of elliptically contoured distributions". In: *Journal of Multivariate Analysis* 11.3 (1981), pp. 368–385. DOI: https://doi.org/10.1016/0047-259X(81)90082-8. URL: https://www.sciencedirect.com/science/article/pii/0047259X81900828.

[20] Robert H Cameron and William T Martin. "Transformations of weiner integrals under translations". In: *Annals of Mathematics* (1944), pp. 386–396.

[21] M Cardiff and PK Kitanidis. "Bayesian inversion for facies detection: An extensible level set framework". In: *Water Resources Research* 45.10 (2009).

[22] Carlos M Carvalho, Nicholas G Polson, and James G Scott. "The horseshoe estimator for sparse signals". In: *Biometrika* 97.2 (2010), pp. 465–480.

[23] George Casella et al. "Penalized regression, standard errors, and Bayesian Bassos". In: *Bayesian analysis* 5.2 (2010), pp. 369–411.

[24] Neil K. Chada, Andrew M. Stuart, and Xin T. Tong. *Tikhonov Regularization Within Ensemble Kalman Inversion*. 2019. arXiv: 1901.10382 [math.NA].

[25] Guanrong Chen and Tetsushi Ueta. "Yet another chaotic attractor". In: *International Journal of Bifurcation and chaos* 9.07 (1999), pp. 1465–1466.

[26] Victor Chen et al. "Dimension-robust MCMC in Bayesian inverse problems". In: *arXiv preprint arXiv:1803.03344* (2018).

[27] Yan Chen and Dean S. Oliver. "Ensemble Randomized Maximum Likelihood Method as an Iterative Ensemble Smoother". In: *Mathematical Geosciences* 44.1 (Dec. 2011), pp. 1–26. DOI: 10.1007/s11004-011-9376-z. URL: http://dx.doi.org/10.1007/s11004-011-9376-z.

[28] Emmet Cleary et al. *Calibrate, Emulate, Sample*. 2020. arXiv: 2001.03689 [stat.CO].

[29] Emmet Cleary et al. *Calibrate, Emulate, Sample*. 2020. arXiv: 2001.03689 [stat.CO].

[30] Maxime Conjard and Henning Omre. "Spatio-Temporal Inversion Using the Selection Kalman Model". In: *Frontiers in Applied Mathematics and Statistics* 7 (Apr. 2021). DOI: 10.3389/fams.2021.636524. URL: https://doi.org/10.3389%2Ffams.2021.636524.

[31] Simon L Cotter et al. "MCMC methods for functions: modifying old algorithms to make them faster". In: *Statistical Science* 28.3 (2013), pp. 424–446.

[32] N. Cressie and C.K. Wikle. *Statistics for Spatio-Temporal Data*. CourseSmart Series. Wiley, 2011. URL: https://books.google.com/books?id=-kOC6D0DiNYC.

[33] Tiangang Cui, Kody J.H. Law, and Youssef M. Marzouk. "Dimension-independent likelihood-informed MCMC". In: *Journal of Computational Physics* 304 (2016), pp. 109–137.

[34] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals, and Systems* 2.4 (Dec. 1989), pp. 303–314. DOI: 10.1007/bf02551274. URL: https://doi.org/10.1007%2Fbf02551274.

[35] Masoumeh Dashti, Stephen Harris, and Andrew Stuart. "Besov priors for Bayesian inverse problems". In: *arXiv preprint arXiv:1105.0889* (2011).

[36] Masoumeh Dashti, Stephen Harris, and Andrew Stuart. "Besov priors for Bayesian inverse problems". In: *Inverse Problems and Imaging* 6.2 (May 2012), pp. 183–200. DOI: 10.3934/ipi.2012.6.183. URL: https://doi.org/10.3934%2Fipi.2012.6.183.

[37] Masoumeh Dashti and Andrew M. Stuart. *The Bayesian Approach To Inverse Problems*. 2015. arXiv: 1302.6989 [math.PR].

[38] Masoumeh Dashti and Andrew M. Stuart. "The Bayesian Approach to Inverse Problems". In: *Handbook of Uncertainty Quantification*. Ed. by Roger Ghanem, David Higdon, and Houman Owhadi. Cham: Springer International Publishing, 2017, pp. 311–428. DOI: 10.1007/978-3-319-12385-1_7. URL: https://doi.org/10.1007/978-3-319-12385-1_7.

[39] Cees Diks and Jasper Vrugt. "Comparison of point forecast accuracy of model averaging methods in hydrologic applications". In: *Stochastic Environmental Research and Risk Assessment* 24 (Aug. 2010), pp. 809–820. DOI: 10.1007/s00477-010-0378-z.

[40] Qingyun Duan et al. "Multi-model ensemble hydrologic prediction using Bayesian model averaging". In: *Advances in Water Resources* 30.5 (2007), pp. 1371–1386. DOI: https://doi.org/10.1016/j.advwatres.2006.11.014. URL: https://www.sciencedirect.com/science/article/pii/S030917080600220X.

[41] Matthew M Dunlop and Andrew M Stuart. "MAP estimators for piecewise continuous inversion". In: *Inverse Problems* 32.10 (2016), p. 105003.

[42] David Echeverría Ciaurri and T. Mukerji. "A Robust Scheme for Spatio-Temporal Inverse Modeling of Oil Reservoirs". In: (Jan. 2009).

[43] S Effah-Poku, William Obeng-Denteh, and IK Dontwi. "A study of chaos in dynamical systems". In: *Journal of Mathematics* 2018 (2018).

[44] Alexandre A. Emerick and Albert C. Reynolds. "Investigation of the sampling performance of ensemble-based methods with a simple reservoir model". In:

Computational Geosciences 17.2 (Jan. 2013), pp. 325–350. DOI: 10.1007/s10596-012-9333-z. URL: http://dx.doi.org/10.1007/s10596-012-9333-z.

[45] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*. Vol. 375. Springer Science & Business Media, 1996.

[46] G. Evensen. "The Ensemble Kalman Filter: theoretical formulation and practical implementation". In: *Ocean Dyn.* 53 (2003), pp. 343–367. DOI: 10.1007/s10236-003-0036-9.

[47] Geir Evensen. "Analysis of iterative ensemble smoothers for solving inverse problems". In: *Computational Geosciences* 22.3 (Mar. 2018), pp. 885–908. DOI: 10.1007/s10596-018-9731-y. URL: http://dx.doi.org/10.1007/s10596-018-9731-y.

[48] Geir Evensen. *Data Assimilation: The Ensemble Kalman Filter*. 2nd ed. Springer-Verlag Berlin Heidelberg, 2009.

[49] Geir Evensen. "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics". In: *Journal of Geophysical Research* 99.C5 (1994), p. 10143. DOI: 10.1029/94jc00572. URL: http://dx.doi.org/10.1029/94JC00572.

[50] Geir Evensen and Peter Jan van Leeuwen. "Assimilation of Geosat Altimeter Data for the Agulhas Current using the Ensemble Kalman Filter with a Quasi-Geostrophic Model". In: 1996.

[51] K. Fang and Y.T. Zhang. *Generalized Multivariate Analysis*. Science Press, 1990. URL: https://books.google.com/books?id=WibvAAAAMAAJ.

[52] Osama S. Faragallah et al. "A Comprehensive Survey Analysis for Present Solutions of Medical Image Fusion and Future Directions". In: *IEEE Access* 9 (2021), pp. 11358–11371. DOI: 10.1109/ACCESS.2020.3048315.

[53] R. A. Fisher and Edward John Russell. "On the mathematical foundations of theoretical statistics". In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 222.594-604 (1922), pp. 309–368. DOI: 10.1098/rsta.1922.0009. eprint: https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.1922.0009. URL: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1922.0009.

[54] Kunihiko Fukushima. "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In:

Biological Cybernetics 36.4 (1980), pp. 193–202. DOI: 10.1007/BF00344251. URL: https://doi.org/10.1007/BF00344251.

[55] Alfredo Garbuno-Inigo et al. "Interacting Langevin Diffusions: Gradient Structure and Ensemble Kalman Sampler". In: *SIAM Journal on Applied Dynamical Systems* 19.1 (2020), pp. 412–441. DOI: 10.1137/19M1251655. eprint: https://doi.org/10.1137/19M1251655. URL: https://doi.org/10.1137/19M1251655.

[56] Gene H Golub, Per Christian Hansen, and Dianne P O'Leary. "Tikhonov regularization and total least squares". In: *SIAM journal on matrix analysis and applications* 21.1 (1999), pp. 185–194.

[57] E. Gómez, M.A. Gomez-Viilegas, and J.M. Marın. "A multivariate generalization of the power exponential family of distributions". In: *Communications in Statistics - Theory and Methods* 27.3 (Jan. 1998), pp. 589–600. DOI: 10.1080/03610929808832115. URL: https://doi.org/10.1080%2F03610929808832115.

[58] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[59] Xifeng Guo et al. "Deep Clustering with Convolutional Autoencoders". In: *Neural Information Processing*. Springer International Publishing, 2017, pp. 373–382. DOI: 10.1007/978-3-319-70096-0_39. URL: https://doi.org/10.1007%2F978-3-319-70096-0_39.

[60] A.K. Gupta and D.K. Nagar. "Matrix Variate Distributions". In: Chapman and Hall/CRC, May 2018. Chap. Chapter 2: MATRIX VARIATE NORMAL DISTRIBUTION. DOI: 10.1201/9780203749289. URL: https://doi.org/10.1201%2F9780203749289.

[61] R. Haining et al. "Bayesian modelling of environmental risk: A small area ecological study of coronary heart disease mortality in relation to modelled outdoor nitrogen oxide levels". In: *Stochastic Environ. Res. Risk Assess.* 21 (2007), pp. 501–509.

[62] Rana Hanocka et al. "MeshCNN: A Network with an Edge". In: *ACM Trans. Graph.* 38.4 (July 2019). DOI: 10.1145/3306346.3322959. URL: https://doi.org/10.1145/3306346.3322959.

[63] Per Christian Hansen. *Discrete inverse problems: insight and algorithms*. SIAM, 2010.

[64] A Hegazi, HN Agiza, and MM El-Dessoky. "Controlling chaotic behaviour for spin generator and Rossler dynamical systems with feedback control". In: *Chaos, Solitons & Fractals* 12.4 (2001), pp. 631–658.

[65] Tommi Heikkilä. "STEMPO–dynamic X-ray tomography phantom". In: *arXiv preprint arXiv:2209.12471* (2022).

[66] G.E. Hinton and R.R. Salakhutdinov. "Reducing the Dimensionality of Data with Neural Networks". In: *Science* 313.5786 (2006), pp. 504–507.

[67] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[68] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Apr. 1991. DOI: 10.1017/cbo9780511840371. URL: https://doi.org/10.1017%2Fcbo9780511840371.

[69] Bamdad Hosseini and Nilima Nigam. "Well-posed Bayesian inverse problems: priors with exponential tails". In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 436–465.

[70] P. L. Houtekamer and Herschel L. Mitchell. "A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation". In: *Monthly Weather Review* 129.1 (Jan. 2001), pp. 123–137. DOI: 10.1175/1520-0493(2001)129<0123:asekff>2.0.co;2. URL: http://dx.doi.org/10.1175/1520-0493(2001)129%3C0123:ASEKFF%3E2.0.CO;2.

[71] P. L. Houtekamer and Fuqing Zhang. "Review of the Ensemble Kalman Filter for Atmospheric Data Assimilation". In: *Monthly Weather Review* 144.12 (Dec. 2016), pp. 4489–4532. DOI: 10.1175/mwr-d-15-0440.1. URL: http://dx.doi.org/10.1175/MWR-D-15-0440.1.

[72] Daniel Zhengyu Huang et al. *Efficient Derivative-free Bayesian Inference for Large-Scale Inverse Problems*. 2022. DOI: 10.48550/ARXIV.2204.04386. URL: https://arxiv.org/abs/2204.04386.

[73] Marco A Iglesias. "A regularizing iterative ensemble Kalman method for PDE-constrained inverse problems". In: *Inverse Problems* 32.2 (Jan. 2016), p. 025002. DOI: 10.1088/0266-5611/32/2/025002. URL: https://doi.org/10.1088%2F0266-5611%2F32%2F2%2F025002.

[74] Marco A Iglesias, Kody J H Law, and Andrew M Stuart. "Ensemble Kalman methods for inverse problems". In: *Inverse Problems* 29.4 (Mar. 2013), p. 045001. DOI: 10.1088/0266-5611/29/4/045001. URL: http://dx.doi.org/10.1088/0266-5611/29/4/045001.

[75] Marco A Iglesias, Yulong Lu, and Andrew M Stuart. "A Bayesian level set method for geometric inverse problems". In: *Interfaces and free boundaries* 18.2 (2016), pp. 181–217.

[76] Vladimir G. Ivancevic and Tijana T. Ivancevic. *Complex Nonlinearity*. Springer Berlin Heidelberg, 2008. DOI: 10.1007/978-3-540-79357-1. URL: https://doi.org/10.1007%2F978-3-540-79357-1.

[77] Valentin Konstantinovich Ivanov. "On linear problems which are not well-posed". In: *Doklady akademii nauk*. Vol. 145. 2. Russian Academy of Sciences. 1962, pp. 270–272.

[78] Mark E. Johnson. "Multivariate Statistical Simulation". In: *Multivariate Statistical Simulation*. Probability and Statistics. John Wiley & Sons, Ltd, 1987. Chap. 6 Elliptically Contoured Distributions, pp. 106–124. DOI: https://doi.org/10.1002/9781118150740.ch6. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118150740.ch6. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118150740.ch6.

[79] Charles E. Baukal Jr., Vladimir Gershtein, and Xianming Jimmy Li, eds. *Computational Fluid Dynamics in Industrial Combustion*. CRC Press, Oct. 2000. DOI: 10.1201/9781482274363. URL: https://doi.org/10.1201%2F9781482274363.

[80] Eugenia Kalnay. "Atmospheric Modeling, Data Assimilation and Predictability". In: (Nov. 2002). DOI: 10.1017/cbo9780511802270. URL: http://dx.doi.org/10.1017/CBO9780511802270.

[81] Y. Kano. "Consistency Property of Elliptic Probability Density Functions". In: *Journal of Multivariate Analysis* 51.1 (1994), pp. 139–147. DOI: https://doi.org/10.1006/jmva.1994.1054. URL: https://www.sciencedirect.com/science/article/pii/S0047259X84710542.

[82] W Clem Karl. "Regularization in image restoration and reconstruction". In: *Handbook of Image and Video Processing*. Elsevier, 2005, pp. 183–V.

[83] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and TrendsÂ® in Machine Learning* 12.4 (2019),

pp. 307–392. DOI: 10.1561/2200000056. URL: http://dx.doi.org/10.1561/2200000056.

[84] Diederik P. Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. 2014. arXiv: http://arxiv.org/abs/1312.6114v10 [stat.ML].

[85] Jason M. Klusowski and Andrew R. Barron. *Uniform Approximation by Neural Networks Activated by First and Second Order Ridge Splines*. 2018. arXiv: 1607.07819 [stat.ML].

[86] Konstantinos Zygalakis Kody Law Andrew Stuart. *Data Assimilation: A Mathematical Introduction*. 1st ed. Vol. 62. Texts in Applied Mathematics. Springer International Publishing, 2015.

[87] Nikola B Kovachki and Andrew M Stuart. "Ensemble Kalman inversion: a derivative-free technique for machine learning tasks". In: *Inverse Problems* 35.9 (Aug. 2019), p. 095005. DOI: 10.1088/1361-6420/ab1c3a. URL: http://dx.doi.org/10.1088/1361-6420/ab1c3a.

[88] Tomasz J. Kozubowski, Krzysztof Podgórski, and Igor Rychlik. "Multivariate generalized Laplace distribution and related random fields". In: *Journal of Multivariate Analysis* 113 (2013). Special Issue on Multivariate Distribution Theory in Memory of Samuel Kotz, pp. 59–72. DOI: https://doi.org/10.1016/j.jmva.2012.02.010. URL: https://www.sciencedirect.com/science/article/pii/S0047259X12000516.

[89] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012, pp. 1097–1105. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[90] Shiwei Lan. "Adaptive dimension reduction to accelerate infinite-dimensional geometric Markov Chain Monte Carlo". In: *Journal of Computational Physics* 392 (Sept. 2019), pp. 71–95. DOI: https://doi.org/10.1016/j.jcp.2019.04.043. URL: http://www.sciencedirect.com/science/article/pii/S002199911930289X.

[91] Shiwei Lan. *Learning Temporal Evolution of Spatial Dependence with Generalized Spatiotemporal Gaussian Process Models*. 2019. arXiv: 1901.04030 [stat.ME].

[92] Shiwei Lan. "Learning Temporal Evolution of Spatial Dependence with Generalized Spatiotemporal Gaussian Process Models". arXiv:1901.04030. Aug. 2021. eprint: 1901.04030. URL: https://arxiv.org/pdf/1901.04030.

[93] Shiwei Lan, Shuyi Li, and Michael O'Connor. "Bayesian Regularization on Function Spaces via Q-Exponential Process". In: (Oct. 2022). eprint: 2210.07987. URL: https://arxiv.org/pdf/2210.07987.pdf.

[94] Shiwei Lan, Shuyi Li, and Babak Shahbaba. "Scaling Up Bayesian Uncertainty Quantification for Inverse Problems using Deep Neural Networks". In: *SIAM/ASA Journal on Uncertainty Quantification* to appear (2022). URL: https://arxiv.org/abs/2101.03906.

[95] Shiwei Lan, Bo Zhou, and Babak Shahbaba. "Spherical Hamiltonian Monte Carlo for Constrained Target Distributions". In: *Proceedings of The 31st International Conference on Machine Learning*. Beijing, China, 2014, pp. 629–637.

[96] Shiwei Lan et al. "Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems". In: *Journal of Computational Physics* 308 (2016), pp. 81–101.

[97] Shiwei Lan et al. "Markov Chain Monte Carlo from Lagrangian Dynamics". In: *Journal of Computational and Graphical Statistics* 24.2 (2015), pp. 357–378.

[98] Matti Lassas, Eero Saksman, and Samuli Siltanen. "Discretization-invariant Bayesian inversion and Besov space priors". In: *Inverse Problems and Imaging* 3.1 (2009), pp. 87–122.

[99] Matti Lassas and Samuli Siltanen. "Can one use total variation prior for edge-preserving Bayesian inversion?" In: *Inverse Problems* 20.5 (2004), p. 1537.

[100] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[101] J Lee and PK Kitanidis. "Bayesian inversion with total variation prior for discrete geologic structure identification". In: *Water Resources Research* 49.11 (2013), pp. 7658–7669.

[102] David Leporini and J-C Pesquet. "Bayesian wavelet denoising: Besov priors and non-Gaussian noises". In: *Signal processing* 81.1 (2001), pp. 55–67.

[103] Qing Li and Nan Lin. "The Bayesian elastic net". In: *Bayesian analysis* 5.1 (2010), pp. 151–170.

[104] Eduardo Liz and Alfonso Ruiz-Herrera. "Chaos in Discrete Structured Population Models". In: *SIAM Journal on Applied Dynamical Systems* 11.4 (Jan. 2012), pp. 1200–1214. DOI: 10.1137/120868980. URL: https://doi.org/10.1137%2F120868980.

[105] Christopher J. Long et al. "State-space solutions to the dynamic magnetoencephalography inverse problem using high performance computing". In: *The Annals of Applied Statistics* 5.2B (June 2011). DOI: 10.1214/11-aoas483. URL: https://doi.org/10.1214%2F11-aoas483.

[106] Edward N. Lorenz. "Deterministic Nonperiodic Flow". In: *Journal of the Atmospheric Sciences* 20.2 (Mar. 1963), pp. 130–141. DOI: 10.1175/1520-0469(1963)020<0130:dnf>2.0.co;2. URL: https://doi.org/10.1175%2F1520-0469%281963%29020%3C0130%3Adnf%3E2.0.co%3B2.

[107] Zhou Lu et al. "The Expressive Power of Neural Networks: A View from the Width". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017, pp. 6231–6239. URL: https://proceedings.neurips.cc/paper/2017/file/32cbf687880eb1674a07bf717761dd3a-Paper.pdf.

[108] Felix Lucka. "Fast Markov chain Monte Carlo sampling for sparse Bayesian inference in high-dimensional inverse problems using L1-type priors". In: *Inverse Problems* 28.12 (Nov. 2012), p. 125012. DOI: 10.1088/0266-5611/28/12/125012. URL: https://doi.org/10.1088/0266-5611/28/12/125012.

[109] Y. Makovoz. "Random Approximants and Neural Networks". In: *Journal of Approximation Theory* 85.1 (1996), pp. 98–109. DOI: https://doi.org/10.1006/jath.1996.0031. URL: http://www.sciencedirect.com/science/article/pii/S0021904596900313.

[110] Markku Markkanen et al. "Cauchy difference priors for edge-preserving Bayesian inversion". In: *Journal of Inverse and Ill-posed Problems* 27.2 (2019), pp. 225–240.

[111] Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and Its Applications*. 2nd. Springer New York, 2011. DOI: 10.1007/978-0-387-68276-1. URL: https://doi.org/10.1007%2F978-0-387-68276-1.

[112] Alexander Meaney, Zenith Purisha, and Samuli Siltanen. "Tomographic X-ray data of 3D emoji". In: *arXiv preprint arXiv:1802.09397* (2018).

[113] L. Mirsky. "A Trace Inequality of John von Neumann." eng. In: *Monatshefte für Mathematik* 79 (1975), pp. 303–306. URL: http://eudml.org/doc/177697.

[114] M. Morzfeld et al. "Feature-based data assimilation in geophysics". In: *Nonlinear Processes in Geophysics* 25.2 (2018), pp. 355–374. DOI: 10.5194/npg-25-355-2018. URL: https://npg.copernicus.org/articles/25/355/2018/.

[115] R. M. Neal. "MCMC using Hamiltonian dynamics". In: *Handbook of Markov Chain Monte Carlo*. Ed. by S. Brooks et al. Chapman and Hall/CRC, 2010.

[116] Elisa Negrini, Giovanna Citti, and Luca Capogna. "System identification through Lipschitz regularized deep neural networks". In: *Journal of Computational Physics* 444 (Nov. 2021), p. 110549. DOI: 10.1016/j.jcp.2021.110549. URL: https://doi.org/10.1016%2Fj.jcp.2021.110549.

[117] Alejandro Ojeda et al. "A Bayesian framework for unifying data cleaning, source separation and imaging of electroencephalographic signals". In: *bioRxiv* (2019). DOI: 10.1101/559450. eprint: https://www.biorxiv.org/content/early/2019/11/20/559450.full.pdf. URL: https://www.biorxiv.org/content/early/2019/11/20/559450.

[118] Bernt Øksendal. *Stochastic Differential Equations*. Springer Berlin Heidelberg, 2003. DOI: 10.1007/978-3-642-14394-6. URL: https://doi.org/10.1007%2F978-3-642-14394-6.

[119] Dean S. Oliver, Albert C. Reynolds, and Ning Liu. "Inverse Theory for Petroleum Reservoir Characterization and History Matching". In: (2008). DOI: 10.1017/cbo9780511535642. URL: http://dx.doi.org/10.1017/CBO9780511535642.

[120] Edward Ott. "Strange attractors and chaotic motions of dynamical systems". In: *Reviews of Modern Physics* 53.4 (1981), p. 655.

[121] Trevor Park and George Casella. "The Bayesian Lasso". In: *Journal of the American Statistical Association* 103.482 (2008), pp. 681–686.

[122] Mirjeta Pasha et al. *Efficient edge-preserving methods for dynamic inverse problems*. 2021. DOI: 10.48550/ARXIV.2107.05727. URL: https://arxiv.org/abs/2107.05727.

[123] N. Petra and G. Stadler. *Model Variational Inverse Problems Governed by Partial Differential Equations*. Tech. rep. The Institute for Computational Engineering and Sciences, The University of Texas at Austin., 2011.

[124] Allan Pinkus. "Approximation theory of the MLP model in neural networks". In: *Acta Numerica* 8 (Jan. 1999), pp. 143–195. DOI: 10.1017/s0962492900002919. URL: https://doi.org/10.1017%2Fs0962492900002919.

[125] Krzysztof Podgórski and Jörg Wegener. "Estimation for Stochastic Models Driven by Laplace Motion". In: *Communications in Statistics - Theory and Methods* 40.18 (Sept. 2011), pp. 3281–3302. DOI: 10.1080/03610926.2010.499051. URL: https://doi.org/10.1080%2F03610926.2010.499051.

[126] Nicholas G. Polson, James G. Scott, and Jesse Windle. "The Bayesian Bridge". In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 76.4 (Nov. 2013), pp. 713–733. DOI: 10.1111/rssb.12042. eprint: https://academic.oup.com/jrsssb/article-pdf/76/4/713/49514175/jrsssb_76_4_713.pdf. URL: https://doi.org/10.1111/rssb.12042.

[127] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2005. DOI: 10.7551/mitpress/3206.001.0001. URL: https://doi.org/10.7551%2Fmitpress%2F3206.001.0001.

[128] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. the MIT Press, 2006.

[129] William Reese, Arvind K Saibaba, and Jonghyun Lee. "Bayesian Level Set Approach for Inverse Problems with Piecewise Constant Reconstructions". In: *arXiv preprint arXiv:2111.15620* (2021).

[130] Rafael Reisenhofer et al. "A Haar wavelet-based perceptual similarity index for image quality assessment". In: *Signal Processing: Image Communication* 61 (2018), pp. 33–43.

[131] Manassés Ribeiro, André Eugênio Lazzaretti, and Heitor Silvério Lopes. "A study of deep convolutional auto-encoders for anomaly detection in videos". In: *Pattern Recognition Letters* 105 (2018). Machine Learning and Applications in Artificial Intelligence, pp. 13–22. DOI: https://doi.org/10.1016/j.patrec.2017.07.016. URL: http://www.sciencedirect.com/science/article/pii/S0167865517302489.

[132] O.E. Rossler. "An equation for hyperchaos". In: *Physics Letters A* 71.2 (1979), pp. 155–157. DOI: https://doi.org/10.1016/0375-9601(79)90150-6. URL: https://www.sciencedirect.com/science/article/pii/0375960179901506.

[133] O.E. Rössler. "An equation for continuous chaos". In: *Physics Letters A* 57.5 (1976), pp. 397–398. DOI: https://doi.org/10.1016/0375-9601(76)90101-8. URL: https://www.sciencedirect.com/science/article/pii/0375960176901018.

[134] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. DOI: 10.1038/323533a0. URL: https://doi.org/10.1038%2F323533a0.

[135] C. Schillings and A. Stuart. "Analysis of the Ensemble Kalman Filter for Inverse Problems". In: *SIAM Journal on Numerical Analysis* 55.3 (2017), pp. 1264–1290. DOI: 10.1137/16M105959X. eprint: https://doi.org/10.1137/16M105959X. URL: https://doi.org/10.1137/16M105959X.

[136] C. Schillings and A. M. Stuart. "Convergence analysis of ensemble Kalman inversion: the linear, noisy case". In: *Applicable Analysis* 97.1 (Oct. 2017), pp. 107–123. DOI: 10.1080/00036811.2017.1386784. URL: http://dx.doi.org/10.1080/00036811.2017.1386784.

[137] Tapio Schneider et al. "Earth System Modeling 2.0: A Blueprint for Models That Learn From Observations and Targeted High-Resolution Simulations". In: *Geophysical Research Letters* 44.24 (2017), pp. 12, 396–12, 417. DOI: 10.1002/2017GL076101. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017GL076101.

[138] I. J. Schoenberg. "Metric spaces and completely monotone functions". In: *Annals of Mathematics* 39 (1938), pp. 811–841.

[139] Amar Shah, Andrew Wilson, and Zoubin Ghahramani. "Student-t Processes as Alternatives to Gaussian Processes". In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. Ed. by Samuel Kaski and Jukka Corander. Vol. 33. Proceedings of Machine Learning Research. Reykjavik, Iceland: PMLR, Apr. 2014, pp. 877–885. URL: https://proceedings.mlr.press/v33/shah14.html.

[140] Babak Shahbaba et al. "Deep Markov Chain Monte Carlo". arXiv:1910.05692. 2019. arXiv: 1910.05692 [stat.CO].

[141] A I Shcherbakova, Y A Kupriyanova, and G V Zhikhareva. "Spatio-temporal analysis the results of solving the inverse problem of electrocardiography". In: *Journal of Physics: Conference Series* 2091.1 (Nov. 2021), p. 012028. DOI: 10.1088/1742-6596/2091/1/012028. URL: https://doi.org/10.1088%2F1742-6596%2F2091%2F1%2F012028.

[142] L. A. Shepp and B. F. Logan. "The Fourier reconstruction of a head section". In: *IEEE Transactions on Nuclear Science* 21.3 (1974), pp. 21–43. DOI: 10.1109/TNS.1974.6499235.

[143] Pridi Siregar and Jean-Paul Sinteff. "Introducing spatio-temporal reasoning into the inverse problem in electroencephalography". In: *Artificial Intelligence in Medicine* 8.2 (1996), pp. 97–122. DOI: https://doi.org/10.1016/0933-3657(95)00028-3. URL: https://www.sciencedirect.com/science/article/pii/0933365795000283.

[144] Roel Snieder and Jeannot Trampert. "Inverse problems in geophysics". In: *Wavefield inversion*. Springer, 1999, pp. 119–190.

[145] Scott A Starks and Vladik Kreinovich. "Multispectral inverse problems in satellite image processing". In: *Bayesian Inference for Inverse Problems*. Vol. 3459. SPIE. 1998, pp. 138–146.

[146] Gemma Stephenson. "Using derivative information in the statistical analysis of computer models". PhD thesis. University of Southampton, 2010.

[147] A. M. Stuart. "Inverse problems: A Bayesian perspective". In: *Acta Numerica* 19 (2010), pp. 451–559. DOI: 10.1017/S0962492910000061.

[148] Andrey Nikolayevich Tikhonov. "On the stability of inverse problems". In: *Dokl. Akad. Nauk SSSR*. Vol. 39. 1943, pp. 195–198.

[149] Hans Triebel. *Function spaces and wavelets on domains*. 7. European Mathematical Society, 2008.

[150] Felipe Uribe, Yiqiu Dong, and Per Christian Hansen. "Horseshoe priors for edge-preserving linear Bayesian inversion". In: *arXiv preprint arXiv:2207.09147* (2022).

[151] Wim Van Aarle et al. "The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography". In: *Ultramicroscopy* 157 (2015), pp. 35–47.

[152] Simopekka Vänskä et al. “Statistical X-ray tomography using empirical Besov priors”. In: *International Journal of Tomography and Statistics* 11 (June 2009).

[153] L. Verlet. “Computer “Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules”. In: *Phys. Rev.* 159.1 (1967), pp. 98–103.

[154] Umberto Villa, Noemi Petra, and Omar Ghattas. *hIPPYlib: An Extensible Software Framework for Large-Scale Inverse Problems Governed by PDEs; Part I: Deterministic Inversion and Linearized Bayesian Inference*. 2020. arXiv: 1909.03948 [math.NA].

[155] Jianfeng Wang et al. “Weather Simulation Uncertainty Estimation Using Bayesian Hierarchical Models”. In: *Journal of Applied Meteorology and Climatology* 58.3 (2019), pp. 585–603. DOI: https://doi.org/10.1175/JAMC-D-18-0018.1. URL: https://journals.ametsoc.org/view/journals/apme/58/3/jamc-d-18-0018.1.xml.

[156] Kangrui Wang et al. “Non-separable Non-stationary random fields”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 9887–9897. URL: http://proceedings.mlr.press/v119/wang20g.html.

[157] Zhou Wang et al. “Image quality assessment: From error visibility to structural similarity”. In: *IEEE transactions on image processing* 13.4 (2004), pp. 600–612.

[158] M. Welling and Y. W. Teh. “Bayesian Learning via Stochastic Gradient Langevin Dynamics”. In: *Proceedings of the International Conference on Machine Learning*. 2011.

[159] J. de Wiljes, S. Reich, and W. Stannat. “Long-Time Stability and Accuracy of the Ensemble Kalman–Bucy Filter for Fully Observed Processes and Small Measurement Noise”. In: *SIAM Journal on Applied Dynamical Systems* 17.2 (2018), pp. 1152–1181. DOI: 10.1137/17M1119056. eprint: https://doi.org/10.1137/17M1119056. URL: https://doi.org/10.1137/17M1119056.

[160] M.W. Woolrich et al. “Fully Bayesian Spatio-Temporal Modeling of FMRI Data”. In: *IEEE Transactions on Medical Imaging* 23.2 (Feb. 2004), pp. 213–231. DOI: 10.1109/tmi.2003.823065. URL: https://doi.org/10.1109%2Ftmi.2003.823065.

[161] Guang Yang et al. “A Bayesian adaptive reservoir operation framework incorporating streamflow non-stationarity”. In: *Journal of Hydrology* 594.C ().

Ed. by null. DOI: 10.1016/j.jhydrol.2021.125959. URL: https://par.nsf.gov/biblio/10280506.

[162] Shyi-Kae Yang, Chieh-Li Chen, and Her-Terng Yau. "Control of chaos in Lorenz system". In: *Chaos, Solitons & Fractals* 13.4 (2002), pp. 767–780.

[163] Ying Yang. "Source-Space Analyses in MEG/EEG and Applications to Explore Spatio-temporal Neural Dynamics in Human Vision". In: (2017). DOI: 10.1184/R1/6723065.V1. URL: https://kilthub.cmu.edu/articles/Source-Space_Analyses_in_MEG_EEG_and_Applications_to_Explore_Spatio-temporal_Neural_Dynamics_in_Human_Vision/6723065/1.

[164] Bing Yao and Hui Yang. "Physics-driven Spatiotemporal Regularization for High-dimensional Predictive Modeling: A Novel Approach to Solve the Inverse ECG Problem". In: *Scientific Reports* 6.1 (Dec. 2016). DOI: 10.1038/srep39012. URL: https://doi.org/10.1038%2Fsrep39012.

[165] MT Yassen. "Chaos control of Chen chaotic dynamical system". In: *Chaos, Solitons & Fractals* 15.2 (2003), pp. 271–283.

[166] Bohai Zhang and Noel Cressie. "Bayesian Inference of Spatio-Temporal Changes of Arctic Sea Ice". In: *Bayesian Analysis* 15.2 (June 2020), pp. 605–631. DOI: 10.1214/20-ba1209.

[167] Jingwen Zhang et al. "A Bayesian model averaging method for the derivation of reservoir operating rules". In: *Journal of Hydrology* 528 (2015), pp. 276–285. DOI: https://doi.org/10.1016/j.jhydrol.2015.06.041. URL: https://www.sciencedirect.com/science/article/pii/S0022169415004540.

[168] Yiheng Zhang, Alireza Ghodrati, and Dana H Brooks. "An analytical comparison of three spatio-temporal regularization methods for dynamic linear inverse problems in a common statistical framework". In: *Inverse Problems* 21.1 (Jan. 2005), pp. 357–382. DOI: 10.1088/0266-5611/21/1/022. URL: https://doi.org/10.1088%2F0266-5611%2F21%2F1%2F022.

[169] Ding-Xuan Zhou. "Universality of deep convolutional neural networks". In: *Applied and Computational Harmonic Analysis* 48.2 (2020), pp. 787–794. DOI: https://doi.org/10.1016/j.acha.2019.06.004. URL: http://www.sciencedirect.com/science/article/pii/S1063520318302045.

[170] Kehe Zhu. *Operator theory in function spaces*. 138. American Mathematical Soc., 2007.

APPENDIX A

PROOFS

A.1 UQ

Proof of Theorem 2.3.1

Theorem (2.3.1)**.** *Let $2 \leq s \leq d$ and $\Omega \subset [-1,1]^d$. Assume $\mathcal{G}_j \in H^r(\mathbb{R}^d)$ for $r > 2 + d/2$, $j = 1, \cdots, m$. If $K \geq 2d/(s-1)$, then there exist $\mathcal{G}^e$ by CNN with ReLU activation function such that*

$$\|\Phi - \Phi^e\|_{H^1(\Omega)} \leq c\|\mathcal{G}\|\sqrt{\log K}K^{-\frac{1}{2}-\frac{1}{2d}} \tag{A.1}$$

where we have $\|\Phi\|_{H^1(\Omega)} = \left(\|\Phi\|^2_{L^2(\Omega)} + \|D\Phi\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}$, c is an absolute constant and $\|\mathcal{G}\| = \max_{1\leq j\leq m}\|\mathcal{G}_j\|_{H^r(\mathbb{R}^d)}$ with $\|\mathcal{G}_j\|_{H^r(\mathbb{R}^d)} := \|(1+|\omega|^2)^{r/2}\widehat{\mathcal{G}}_j(\omega)\|_{L^2(\mathbb{R}^d)}$.

Proof. Note because we have

$$|\Phi(u) - \Phi^e(u)| \leq |\langle \mathcal{G}(u) - \mathcal{G}^e(u), y - \mathcal{G}(u)\rangle_\Gamma| + |\langle y - \mathcal{G}^e(u), \mathcal{G}(u) - \mathcal{G}^e(u)\rangle_\Gamma|$$
$$|D\Phi(u) - D\Phi^e(u)| \leq |\langle \mathcal{G}(u) - \mathcal{G}^e(u), D\mathcal{G}(u)\rangle_\Gamma| + |\langle y - \mathcal{G}^e(u), D\mathcal{G}(u) - D\mathcal{G}^e(u)\rangle_\Gamma|$$

it suffices to prove

$$\|\mathcal{G}_j - \mathcal{G}^e_j\|_{H^1(\Omega)} \leq c_j\|\mathcal{G}_j\|\sqrt{\log K}K^{-\frac{1}{2}-\frac{1}{2d}}, \quad j = 1, \cdots, m \tag{A.2}$$

Let K^* be the integer part of $\frac{(s-1)K}{d} - 1$, i.e. $K^* = \left[\frac{(s-1)K}{d}\right] - 1 \geq 1$. For each $j = 1, \cdots, m$, there exists a linear combination of ramp ridge functions of the following form by Theorem 2 of [85]:

$$\mathcal{G}^e_j(u) = \mathcal{G}_j(0) + D\mathcal{G}_j(0)\cdot u + \frac{v}{K^*}\sum_{k=1}^{K^*}\beta_k(\alpha_k\cdot u - t_k)_+ \tag{A.3}$$

with $\beta_k \in [-1,1]$, $\|\alpha_k\|_1 = 1$, $t_k \in [0,1]$ and $|v| \leq 2v_{\mathcal{G}_j,2} := \int_{\mathbb{R}^d}\|\omega\|_1^2|\widehat{\mathcal{G}}_j(\omega)|d\omega \leq c_{d,r}\|\mathcal{G}_j\|$ such that

$$\|\mathcal{G}_j - \mathcal{G}^e_j\|_{L^\infty([-1,1]^d)} \leq c_0 v_{\mathcal{G}_j,2}(\sqrt{\log K^*} \vee \sqrt{d}\,)(K^*)^{-\frac{1}{2}-\frac{1}{d}} \tag{A.4}$$

[169, Theorem B] constructs weights W and biases b of a CNN that has output of the form in Equation (A.3). Therefore,

$$\|\mathcal{G}_j - \mathcal{G}^e_j\|_{L^2(\Omega)} \leq C\|\mathcal{G}_j - \mathcal{G}^e_j\|_{L^\infty(\Omega)} \leq c_j\|\mathcal{G}_j\|\sqrt{\log K}K^{-\frac{1}{2}-\frac{1}{d}}, \quad j = 1, \cdots, m \tag{A.5}$$

Now we take derivative on both sides of (A.3) to get

$$D\mathcal{G}^e_j(u) = D\mathcal{G}_j(0) + \frac{v}{K}\sum_{k=1}^{K}\alpha_k\beta_k H(\alpha_k\cdot u - t_k) \tag{A.6}$$

where $H(x) = I(x \geq 0)$ is the Heaviside function. For any $i = 1, \cdots, d$, we have $v_{D_i\mathcal{G}_j,1} := \int_{\mathbb{R}^d} \|\omega\|_1 |\widehat{D_i\mathcal{G}_j}(\omega)| d\omega \leq C \int_{\mathbb{R}^d} \|\omega\|_1 |\omega_i| |\widehat{\mathcal{G}_j}(\omega)| d\omega \leq C v_{\mathcal{G}_j,2}$. Therefore, by Theorem 3 of [109] we have

$$\|D_i\mathcal{G}_j - D_i\mathcal{G}_j^e\|_{L^2([-1,1]^d)} \leq c_0' v_{\mathcal{G}_j,2} K^{-\frac{1}{2}-\frac{1}{2d}} \tag{A.7}$$

Inequality (A.5) and inequality (A.7) yield error bound (A.2) thus complete the proof. □

A.2 STIP

A.2.1 Proof of Theorem 3.3.1

Theorem (3.3.1)**.** *If we set the maximal eigenvalues of $\boldsymbol{C_x}$ and $\boldsymbol{C}_t$ such that $\lambda_{\max}(\boldsymbol{C_x})\lambda_{\max}(\boldsymbol{C}_t) \leq \sigma_\varepsilon^2$, then the following inequality holds regarding the Fisher information matrices, $\mathcal{I}_S$ and $\mathcal{I}_{ST}$, of the static model and the STGP model respectively:*

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_S(u) \tag{A.8}$$

If we control the maximal eigenvalues of $\boldsymbol{C_x}$ and $\boldsymbol{C}_t$ such that $\lambda_{\max}(\boldsymbol{C_x})\lambda_{\max}(\boldsymbol{C}_t) \leq J\lambda_{\min}(\Gamma_{obs})$, then the following inequality holds regarding the Fisher information matrices, $\mathcal{I}_T$ and $\mathcal{I}_{ST}$, of the time-averaged model and the STGP model respectively:

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_T(u) \tag{A.9}$$

Proof. Denote $\mathbf{Y}_0 = \mathbf{Y} - \mathbf{M}$. We have $\Phi_*(u) = \frac{1}{2}\mathrm{tr}\left[\mathbf{V}_*^{-1}\mathbf{Y}_0^\mathsf{T}\mathbf{U}_*^{-1}\mathbf{Y}_0\right]$ with $*$ being S or ST. $\mathbf{U}_\mathrm{S}$, $\mathbf{V}_\mathrm{S}$, $\mathbf{U}_\mathrm{ST}$ and $\mathbf{V}_\mathrm{ST}$ are specified in (3.14). We notice that both $\mathbf{U}_*$ and $\mathbf{V}_*$ are symmetric, then we have

$$\begin{aligned}
\frac{\partial\Phi_*}{\partial u_i} &= \frac{1}{2}\left\{\mathrm{tr}\left[\mathbf{V}_*^{-1}\frac{\partial\mathbf{Y}_0^\mathsf{T}}{\partial u_i}\mathbf{U}_*^{-1}\mathbf{Y}_0\right] + \mathrm{tr}\left[\mathbf{V}_*^{-1}\mathbf{Y}_0^\mathsf{T}\mathbf{U}_*^{-1}\frac{\partial\mathbf{Y}_0}{\partial u_i}\right]\right\} \\
&= \mathrm{tr}\left[\mathbf{V}_*^{-1}\mathbf{Y}_0^\mathsf{T}\mathbf{U}_*^{-1}\frac{\partial\mathbf{Y}_0}{\partial u_i}\right] \\
\frac{\partial^2\Phi_*}{\partial u_i\partial u_j} &= \mathrm{tr}\left[\mathbf{V}_*^{-1}\mathbf{Y}_0^\mathsf{T}\mathbf{U}_*^{-1}\frac{\partial^2\mathbf{Y}_0}{\partial u_i\partial u_j}\right] + \mathrm{tr}\left[\mathbf{V}_*^{-1}\frac{\partial\mathbf{Y}_0^\mathsf{T}}{\partial u_i}\mathbf{U}_*^{-1}\frac{\partial\mathbf{Y}_0}{\partial u_j}\right]
\end{aligned}$$

Due to the i.i.d. assumption in both models, $\mathbf{Y}_0$ is independent of either $\frac{\partial\mathbf{Y}_0}{\partial u_i}$ or $\frac{\partial^2\mathbf{Y}_0}{\partial u_i\partial u_j}$. Therefore

$$\begin{aligned}
(\mathcal{I}_*)_{ij} &= \mathrm{E}\left[\frac{\partial^2\Phi_*}{\partial u_i\partial u_j}\right] = \mathrm{E}\left[\mathrm{tr}\left(\mathbf{V}_*^{-1}\frac{\partial\mathbf{Y}_0^\mathsf{T}}{\partial u_i}\mathbf{U}_*^{-1}\frac{\partial\mathbf{Y}_0}{\partial u_j}\right)\right] \\
&= \mathrm{E}\left[\mathrm{vec}\left(\frac{\partial\mathbf{Y}_0}{\partial u_i}\right)^\mathsf{T}(\mathbf{V}_*^{-1}\otimes\mathbf{U}_*^{-1})\mathrm{vec}\left(\frac{\partial\mathbf{Y}_0}{\partial u_j}\right)\right]
\end{aligned} \tag{A.10}$$

For any $\mathbf{w} = (w_1, \cdots, w_p) \in \mathbb{R}^p$ and $\mathbf{w} \neq \mathbf{0}$, denote $\tilde{\mathbf{w}} := \sum_{i=1}^{p} w_i \text{vec}\left(\frac{\partial \mathbf{Y}_0}{\partial u_i}\right)$. To prove $\mathcal{I}_{\text{ST}}(u) \geq \mathcal{I}_{\text{S}}(u)$, it suffices to show $\tilde{\mathbf{w}}^{\mathsf{T}}(\mathbf{V}_{\text{ST}} \otimes \mathbf{U}_{\text{ST}})^{-1}\tilde{\mathbf{w}} \geq \tilde{\mathbf{w}}^{\mathsf{T}}(\mathbf{V}_{\text{S}} \otimes \mathbf{U}_{\text{S}})^{-1}\tilde{\mathbf{w}}$.

By [Theorem 4.2.12 in 68], we know that any eigenvalue of $\mathbf{V}_* \otimes \mathbf{U}_*$ has the format as a product of eigenvalues of $\mathbf{V}_*$ and $\mathbf{U}_*$ respectively, i.e. $\lambda_k(\mathbf{V}_* \otimes \mathbf{U}_*) = \lambda_i(\mathbf{V}_*)\lambda_j(\mathbf{U}_*)$, where where $\{\lambda_j(M)\}$ are the ordered eigenvalues of M, i.e. $\lambda_1(M) \geq \cdots \geq \lambda_d(M)$. By the given condition we have

$$\begin{aligned} &\lambda_{IJ}((\mathbf{V}_{\text{ST}} \otimes \mathbf{U}_{\text{ST}})^{-1}) = \lambda_1^{-1}(\mathbf{V}_{\text{ST}} \otimes \mathbf{U}_{\text{ST}}) \\ &= \lambda_1^{-1}(\mathbf{C}_t)\lambda_1^{-1}(\mathbf{C}_{\mathbf{x}}) \geq \sigma_\varepsilon^{-2} = \lambda_1((\mathbf{V}_{\text{S}} \otimes \mathbf{U}_{\text{S}})^{-1}) \end{aligned} \tag{A.11}$$

Thus it completes the proof of the first inequality.

Similarly by the second condition, we have

$$\begin{aligned} &\lambda_{IJ}((\mathbf{V}_{\text{ST}} \otimes \mathbf{U}_{\text{ST}})^{-1}) = \lambda_1^{-1}(\mathbf{C}_t)\lambda_1^{-1}(\mathbf{C}_{\mathbf{x}}) \\ &\geq J^{-1}\lambda_{\min}^{-1}(\Gamma_{\text{obs}}) = \lambda_1(\mathbf{V}_{\text{S}}^{-} \otimes \mathbf{U}_{\text{S}}^{-1}) \end{aligned} \tag{A.12}$$

and complete the proof of the second inequality. □

A.2.2 Proof of Theorem 3.3.2

Theorem (3.3.2)**.** *If we choose $\boldsymbol{C_x} = \Gamma_{obs}$ and require the maximal eigenvalue of $\boldsymbol{C}_t$, $\lambda_{\max}(\boldsymbol{C}_t) \leq J$, then the following inequality holds regarding the Fisher information matrices, $\mathcal{I}_T$ and $\mathcal{I}_{ST}$, of the time-averaged model and the STGP model respectively:*

$$\mathcal{I}_{ST}(u) \geq \mathcal{I}_T(u) \tag{A.13}$$

Proof. Denote $\mathbf{Y}_0 = \mathbf{Y} - \mathbf{M}$. We have $\Phi_*(u) = \frac{1}{2}\text{tr}\left[\mathbf{V}_*^{-1}\mathbf{Y}_0^{\mathsf{T}}\mathbf{U}_*^{-1}\mathbf{Y}_0\right]$ with $*$ being T or ST. $\mathbf{U}_{\text{T}}$, $\mathbf{V}_{\text{T}}$, $\mathbf{U}_{\text{ST}}$ and $\mathbf{V}_{\text{ST}}$ are specified in (3.14).

By the similar argument of the proof in Theorem 3.3.1, we have

$$(\mathcal{I}_*)_{ij} = \text{E}\left[\frac{\partial^2 \Phi_*}{\partial u_i \partial u_j}\right] = \text{tr}\left[\mathbf{V}_*^{-1}\text{E}\left(\frac{\partial \mathbf{Y}_0^{\mathsf{T}}}{\partial u_i}\mathbf{U}_*^{-1}\frac{\partial \mathbf{Y}_0}{\partial u_j}\right)\right] \tag{A.14}$$

For any $\mathbf{w} = (w_1, \cdots, w_p) \in \mathbb{R}^p$ and $\mathbf{w} \neq \mathbf{0}$, denote $\mathbf{W} := \sum_{i,j=1}^{p} w_i \text{E}\left(\frac{\partial \mathbf{Y}_0^{\mathsf{T}}}{\partial u_i}\mathbf{U}_*^{-1}\frac{\partial \mathbf{Y}_0}{\partial u_j}\right) w_j$. We know $\mathbf{W} \geq \mathbf{0}_{J\times J}$. It suffices to show $\text{tr}[\mathbf{V}_{\text{ST}}^{-1}\mathbf{W}] \geq \text{tr}[\mathbf{V}_{\text{T}}^{-1}\mathbf{W}]$.

By the corollary [111] of Von Neumann's trace inequality [113], we have

$$\begin{aligned} &\sum_{j=1}^{J} \lambda_j(\mathbf{V}_*^{-1})\lambda_{J-j+1}(\mathbf{W}) \leq \text{tr}(\mathbf{V}_*^{-1}\mathbf{W}) \\ &\leq \sum_{j=1}^{J} \lambda_j(\mathbf{V}_*^{-1})\lambda_j(\mathbf{W}) \end{aligned} \tag{A.15}$$

where $\{\lambda_j(M)\}$ are the ordered eigenvalues of M, i.e. $\lambda_1(M) \geq \cdots \geq \lambda_d(M)$. The only non-zero eigenvalue of $\mathbf{V}_T^- = J^{-2}(\mathbf{1}_J\mathbf{1}_J^\mathsf{T})$ is $\lambda_1(\mathbf{V}_T^-) = J^{-1}$. Therefore, we have

$$\begin{aligned} &\mathrm{tr}[\mathbf{V}_T^-\mathbf{W}] \leq J^{-1}\lambda_1(\mathbf{W}) \leq \lambda_J(\mathbf{V}_{\mathrm{ST}}^{-1})\lambda_1(\mathbf{W}) + \\ &\sum_{j=1}^{J-1}\lambda_j(\mathbf{V}_{\mathrm{ST}}^{-1})\lambda_{J-j+1}(\mathbf{W}) \leq \mathrm{tr}[\mathbf{V}_{\mathrm{ST}}^{-1}\mathbf{W}] \end{aligned} \tag{A.16}$$

where $\lambda_J(\mathbf{V}_{\mathrm{ST}}^{-1}) = \lambda_1^{-1}(\mathbf{C}_t) \geq J^{-1}$ and $\lambda_j(\mathbf{V}_{\mathrm{ST}}^{-1}), \lambda_j(\mathbf{W}) \geq 0$. □

A.3 Q-EP

A.3.1 Proof of Theorem 4.2.4

Proof. First we prove the exchangeability of $\mathrm{q{-}ED}_d(\boldsymbol{\mu}, \mathbf{C})$ with general (non-identity) covariance matrix $\mathbf{C} = [\mathcal{C}(t_i, t_j)]_{d\times d}$ for some kernel function $\mathcal{C}$. It actually holds for all elliptic distributions including MVN. Their densities contain the essential quadratic form $r(\mathbf{u}) = \mathbf{u}^\mathsf{T}\mathbf{C}^{-1}\mathbf{u}$ which is invariant under any permutation of coordinates.

Denote $\mathbf{u} = [u_{t_1}, \cdots, u_{t_i}, \cdots, u_{t_j}, \cdots, u_{t_d}]^T$. Without loss of generality, we only need to show $r(\mathbf{u})$ is invariant by switching two coordinates, say, $t_i \leftrightarrow t_j$. Denote $\mathbf{u}' = [u_{t_1}, \cdots, u_{t_j}, \cdots, u_{t_i}, \cdots, u_{t_d}]^T$. Switching t_i and t_j leads to a different covariance matrix $\mathbf{C}'$ obtained by switching both i-th and j-th rows and columns simultaneously in $\mathbf{C}$. If we denote the elementary matrix E_{ij} as derived from switching i-th and j-th rows of the identity matrix $\mathbf{I}$. Then we have

$$\mathbf{u}' = E_{ij}\mathbf{u}, \quad \mathbf{C}' = E_{ij}\mathbf{C}E_{ij}$$

Note E_{ij} is idempotent, i.e. $E_{ij} = E_{ij}^{-1}$. Therefore

$$(\mathbf{u}')^\mathsf{T}(\mathbf{C}')^{-1}\mathbf{u}' = \mathbf{u}^\mathsf{T}E_{ij}E_{ij}\mathbf{C}^{-1}E_{ij}E_{ij}\mathbf{u} = \mathbf{u}^\mathsf{T}\mathbf{C}^{-1}\mathbf{u}$$

Next, the consistency directly follows from Kano's consistency Theorem 3.2 with our choice of $g(r)$. The proof is hence completed. □

A.3.2 Theorem of Q-EP as a mixture of Gaussians

Theorem A.3.1. *Suppose* $\boldsymbol{u} \sim \mathrm{q{-}ED}_d(0, \boldsymbol{C})$ *for* $0 < q < 2$*, then there exist an random variable* $V > 0$ *and a standard normal random vector* $\boldsymbol{Z} \sim \mathcal{N}_d(\boldsymbol{0}, \boldsymbol{I})$ *independent of each other such that* $\boldsymbol{u} \overset{d}{=} \boldsymbol{Z}/V$*.*

Proof. Based on [5], it suffices to show $(-\frac{d}{dr})^k g(r) \geq 0$ for all $k \in \mathbb{N}$. Observe that $g'(r) = \left[(\frac{q}{2}-1)\frac{d}{2}r^{(\frac{q}{2}-1)\frac{d}{2}-1} - \frac{q}{4}r^{(\frac{q}{2}-1)(\frac{d}{2}+1)}\right]\exp\{-\frac{r^{\frac{q}{2}}}{2}\} \leq 0$ when $q \leq 2$. Denote $(-\frac{d}{dr})^k g(r) := p_k(r^{(\frac{q}{2}-1)/2}, r^{-1})\exp\{-\frac{r^{\frac{q}{2}}}{2}\}$ where the coefficients of polynomial p_k are all non-negative. Then we have

$$\left(-\frac{d}{dr}\right)^{k+1} g(r) = \left[-\frac{d}{dr}p_k(r^{(\frac{q}{2}-1)/2}, r^{-1}) + \frac{q}{4}r^{(\frac{q}{2}-1)}p_k(r^{(\frac{q}{2}-1)/2}, r^{-1})\right]\exp\{-\frac{r^{\frac{q}{2}}}{2}\}$$

where $p_{k+1}(r^{(\frac{q}{2}-1)/2}, r^{-1})$ being the term in the square bracket has all positive coefficients because the powers $(\frac{q}{2}-1)/2$ and -1 appear as coefficients in $\frac{d}{dr}p_k(r^{(\frac{q}{2}-1)/2}, r^{-1})$ and are both negative. The proof is completed by induction. □

A.3.3 Proposition of distribution of $r(\mathbf{u})$

The following proposition determines the distribution of $R = \sqrt{r(\mathbf{u})}$ as q-root of a gamma (also *chi*-squared) distribution thus gives a complete recipe for generating random vector $\mathbf{u} \sim \mathrm{q{-}ED}_d(0, \mathbf{C})$ based on the stochastic representation (4.11).

Proposition A.3.1. *If* $\boldsymbol{u} \sim \mathrm{q{-}ED}_d(0, \boldsymbol{C})$*, then we have*

$$\begin{aligned} R^q = r^{\frac{q}{2}} &\sim \Gamma\left(\alpha = \frac{d}{2}, \beta = \frac{1}{2}\right) = \chi^2_d, \quad \textit{and} \\ \mathrm{E}[R^k] &= 2^{\frac{k}{q}}\frac{\Gamma(\frac{d}{2}+\frac{k}{q})}{\Gamma(\frac{d}{2})} \dot{\sim} d^{\frac{k}{q}}, \ \textit{as } d \to \infty, \ \forall k \in \mathbb{N} \end{aligned} \tag{A.17}$$

Proof. With out chosen $g(r)$, the density of r becomes

$$f(r) \propto r^{\frac{d}{2}-1}r^{(\frac{q}{2}-1)\frac{d}{2}}\exp\left\{-\frac{r^{\frac{q}{2}}}{2}\right\} = r^{\frac{q}{2}\cdot\frac{d}{2}-1}\exp\left\{-\frac{r^{\frac{q}{2}}}{2}\right\}$$

A change of variable $r \to r^{\frac{q}{2}}$ yields the density of $R^q = r^{\frac{q}{2}}$ that can be recognized as the density of χ^2_d.

On the other hand, since $v := R^q \sim \Gamma\left(\alpha = \frac{d}{2}, \beta = \frac{1}{2}\right)$, we have:

$$\begin{aligned} \mathrm{E}[R^k] &= \int_0^\infty v^{\frac{k}{q}} f(v)dv = \frac{1}{\Gamma(\frac{d}{2})}\left(\frac{1}{2}\right)^{\frac{d}{2}}\int_0^\infty v^{\frac{k}{q}+\frac{d}{2}-1}\exp\left\{-\frac{1}{2}v\right\}dv \\ &= 2^{\frac{k}{q}}\frac{\Gamma(\frac{d}{2}+\frac{k}{q})}{\Gamma(\frac{d}{2})} \dot{\sim} 2^{\frac{k}{q}}\left(\frac{d}{2}\right)^{\frac{k}{q}} = d^{\frac{k}{q}} \end{aligned}$$

where we use $\Gamma(x+\alpha) \dot{\sim} \Gamma(x)x^\alpha$ as $x \to \infty$ with $x = \frac{d}{2}$ and $\alpha = \frac{k}{q}$ when $d \to \infty$. □

A.3.4 Proof of Proposition 4.2.1

Proof. By Theorem 2.6.4 in [51] for $\mathrm{q}-\mathrm{ED}_d(\boldsymbol{\mu}, \mathbf{C}) = \mathrm{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ with our chosen g, we know $\mathrm{E}[\mathbf{u}] = \boldsymbol{\mu}$ and $\mathrm{Cov}(\mathbf{u}) = (\mathrm{E}[R^2]/\mathrm{rank}(\mathbf{C}))\mathbf{C}$. It follows by letting $k = 2$ in Proposition 3.1 and using a similar asymptotic analysis. □

A.3.5 Proof of Theorem 4.2.5

Proof. Note we can approximate $\phi_\ell(x) \in L^2(D)$ with simple functions $\tilde{\phi}_\ell(x) = \sum_{i=1}^d k_i \chi_{D_i}(x)$ where D_i's are measurable subsets of D and $\chi_{D_i}(x) = 1$ if $x \in D_i$ and 0 otherwise. By the linear combination property of elliptic distributions [c.f. Theorem 2.6.3 in 51], $\tilde{u}_\ell = \int_D u(x)\tilde{\phi}_\ell(x)dx \sim \mathrm{q}-\mathrm{ED}(0, c)$ with $c = \alpha_d^{-1}\mathrm{E}[\tilde{u}_\ell^2]$ to be determined. Note $\alpha_d = \frac{2^{\frac{2}{q}}\Gamma(\frac{d}{2}+\frac{2}{q})}{d\Gamma(\frac{d}{2})}d^{1-\frac{2}{q}}$ comes from Proposition 3.2 and the scaling $\mathbf{u}^* = d^{\frac{1}{2}-\frac{1}{q}}\mathbf{u}$ in Definition 3.2. We have $\alpha_d = \frac{\Gamma(\frac{d}{2}+\frac{2}{q})}{\Gamma(\frac{d}{2})}\left(\frac{2}{d}\right)^{\frac{2}{q}} \to 1$ as $d \to \infty$. Taking the limit $d \to \infty$, we have $u_\ell = \int_D u(x)\phi_\ell(x)dx \sim \mathrm{q}-\mathrm{ED}(0, c)$. In general, by the similar argument we have

$$\begin{aligned}\mathrm{Cov}(u_\ell, u_{\ell'}) = \mathrm{E}[u_\ell u_{\ell'}] &= \int_D\int_D \mathrm{E}[u(x)u(x')]\phi_\ell(x)\phi_{\ell'}(x')dxdx' \\ &= \int_D\int_D \mathcal{C}(x, x')\phi_\ell(x)\phi_{\ell'}(x')dxdx' = \int_D \lambda_\ell\phi_\ell(x')\phi_{\ell'}(x')dx' = \lambda_\ell\delta_{\ell\ell'}\end{aligned}$$

Thus it completes the proof. □

A.3.6 Proof of Posterior Prediction Theorem

Consider the generic Bayesian regression model:

$$\begin{aligned} y &= u(x) + \varepsilon, \quad \varepsilon \sim L(\cdot; 0, \Sigma) \\ u &\sim \mu_0(du) \end{aligned} \tag{A.18}$$

where $L(\cdot; 0, \Sigma)$ denotes some likelihood model with zero mean and covariance Σ, and the mean function u can be given a prior either Besov or Q-EP.

Theorem A.3.2 (Posterior Prediction)**.** *Given covariates $\boldsymbol{x} = \{x_i\}_{i=1}^N$ and observations $\boldsymbol{y} = \{y_i\}_{i=1}^N$ following* $\mathrm{q}-\mathrm{ED}$ *in the model* (A.18) *with* $\mathrm{q}-\mathcal{EP}$ *prior for the same $q > 0$, we have the following posterior predictive distribution for $u(x_*)$ at (a) new point(s) x_*:*

$$u(x_*)|\boldsymbol{y}, \boldsymbol{x}, x_* \sim \mathrm{q}-\mathrm{ED}(\boldsymbol{\mu}^*, \boldsymbol{C}^*), \quad \boldsymbol{\mu}^* = \boldsymbol{C}_*^\mathsf{T}(\boldsymbol{C}+\Sigma)^{-1}\boldsymbol{y}, \quad \boldsymbol{C}^* = \boldsymbol{C}_{**} - \boldsymbol{C}_*^\mathsf{T}(\boldsymbol{C}+\Sigma)^{-1}\boldsymbol{C}_* \tag{A.19}$$

where $\boldsymbol{C} = \mathcal{C}(\boldsymbol{x}, \boldsymbol{x})$, $\boldsymbol{C}_* = \mathcal{C}(\boldsymbol{x}, x_*)$, *and* $\boldsymbol{C}_{**} = \mathcal{C}(x_*, x_*)$.

Before proving Theorem A.3.2, we first prove the following lemma based on the conditional of elliptic distribution [19, 51].

Lemma A.3.1. *If* $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2) \sim \text{q}-\text{ED}_d(\boldsymbol{\mu}, \boldsymbol{C})$ *with* $\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$ *and* $\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} \end{bmatrix}$, $\boldsymbol{u} \in \mathbb{R}^d$, $\boldsymbol{u}_i \in \mathbb{R}^{d_i}$ *for* $i = 1, 2$ *and* $d_1 + d_2 = d$, *then we have the following conditional distribution*

$$\begin{aligned} &\boldsymbol{u}_1 | \boldsymbol{u}_2 \sim \text{q}-\text{ED}_{d_1}(\boldsymbol{\mu}_{1\cdot 2}, \boldsymbol{C}_{11\cdot 2}), \\ &\boldsymbol{\mu}_{1\cdot 2} = \boldsymbol{\mu}_1 + \boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1}(\boldsymbol{u}_2 - \boldsymbol{\mu}_2), \quad \boldsymbol{C}_{11\cdot 2} = \boldsymbol{C}_{11} - \boldsymbol{C}_{12}\boldsymbol{C}_{22}^{-1}\boldsymbol{C}_{21} \end{aligned}$$

Proof. This directly follows from [Corollary 5 of Theorem 5 in 19] or [Corollary 3 of Theorem 2.6.6 in 51] for $\text{q}-\text{ED}_d(\boldsymbol{\mu}, \mathbf{C}) = \text{EC}_d(\boldsymbol{\mu}, \mathbf{C}, g)$ with our chosen g. □

Now we prove the Theorem 3.5.

Proof. By the linear combination property of the elliptic distributions [78, 51], we have $\mathbf{y} \sim \text{q}-\text{ED}(\mathbf{0}, \mathbf{C} + \Sigma)$. Then based on the consistency, we have the joint distribution

$$\begin{bmatrix} \mathbf{y} \\ u(x_*) \end{bmatrix} \sim \text{q}-\text{ED}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C} + \Sigma & \mathbf{C}_* \\ \mathbf{C}_*^\mathsf{T} & \mathbf{C}_{**} \end{bmatrix}\right)$$

Therefore, the conclusion follows from Lemma A.3.1. □

A.3.7 Proposition of Conditional Conjugacy for Variance Magnitude (σ^2)

Proposition A.3.2. *If we assume a proper inverse-gamma prior for the variance magnitude such that* $\boldsymbol{u}|\sigma^2 \sim \text{q}-\text{ED}_d(\boldsymbol{\mu}, \boldsymbol{C} = \sigma^2 \boldsymbol{C}_0)$, *and* $\sigma^q \sim \Gamma^{-1}(\alpha, \beta)$, *then we have*

$$\sigma^q | \boldsymbol{u} \sim \Gamma^{-1}(\alpha', \beta'), \quad \alpha' = \alpha + \frac{d}{2}, \quad \beta' = \beta + \frac{(\boldsymbol{u} - \boldsymbol{\mu})^\mathsf{T} \boldsymbol{C}_0^{-1}(\boldsymbol{u} - \boldsymbol{\mu})}{2} \tag{A.20}$$

Proof. Denote $r_0 = (\mathbf{u} - \boldsymbol{\mu})^\mathsf{T} \mathbf{C}_0^{-1} (\mathbf{u} - \boldsymbol{\mu})$. We can compute the joint density of $\mathbf{u}$ and σ^2

$$\begin{aligned} p(\mathbf{u}, \sigma^2) =& p(\mathbf{u}|\sigma^2) p(\sigma^q) \\ =& \frac{q}{2} (2\pi)^{-\frac{d}{2}} |\mathbf{C}_0|^{-\frac{1}{2}} r_0^{(\frac{q}{2}-1)\frac{d}{2}} \sigma^{-\frac{qd}{2}} \exp\left\{ -\sigma^{-q} \frac{r_0^{\frac{q}{2}}}{2} \right\} \frac{\beta^\alpha}{\Gamma(\alpha)} (\sigma^q)^{-(\alpha+1)} \exp(-\beta\sigma^{-q}) \\ \propto& (\sigma^q)^{-(\alpha + \frac{d}{2} + 1)} \exp\left\{ -\sigma^{-q} \left(\beta + \frac{r_0^q}{2} \right) \right\} \end{aligned}$$

By identifying the parameters for σ^q we recognize that $\sigma^q|\mathbf{u}$ is another inverse-gamma with parameters α' and β' as given. □

A.4 STBP

Proof of Proposition 4.3.1

Proposition (4.3.1)**.** *If* $u \sim \mathcal{STBP}(\kappa, \mathcal{C}, \mathbb{X}^{r_0,q,p})$*, then we have*

$$\mathrm{Cov}(u(\boldsymbol{x},t), u(\boldsymbol{x}',t')) = \sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(\boldsymbol{x})\phi_\ell(\boldsymbol{x}')\mathcal{C}(t,t') \tag{A.21}$$

Proof. We can directly compute

$$\begin{aligned}
&\mathrm{Cov}(u(\mathbf{x},t), u(\mathbf{x}',t')) \\
&= \mathrm{E}(u(\mathbf{x},t)u(\mathbf{x}',t')) = \mathrm{E}\left[\sum_{\ell=1}^{\infty} \gamma_\ell \xi_\ell(t)\phi_\ell(\mathbf{x}) \sum_{\ell'=1}^{\infty} \gamma_{\ell'} \xi_{\ell'}(t')\phi_{\ell'}(\mathbf{x}')\right] \\
&= \sum_{\ell,\ell'=1}^{\infty} \gamma_\ell \gamma_{\ell'} \phi_\ell(\mathbf{x})\phi_{\ell'}(\mathbf{x}')\mathrm{E}[\xi_\ell(t)\xi_{\ell'}(t')] = \sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(\mathbf{x})\phi_\ell(\mathbf{x}')\mathrm{E}[\xi_\ell(t)\xi_\ell(t')] \\
&= \sum_{\ell=1}^{\infty} \gamma_\ell^2 \phi_\ell(\mathbf{x})\phi_\ell(\mathbf{x}')\mathcal{C}(t,t')
\end{aligned} \tag{A.22}$$

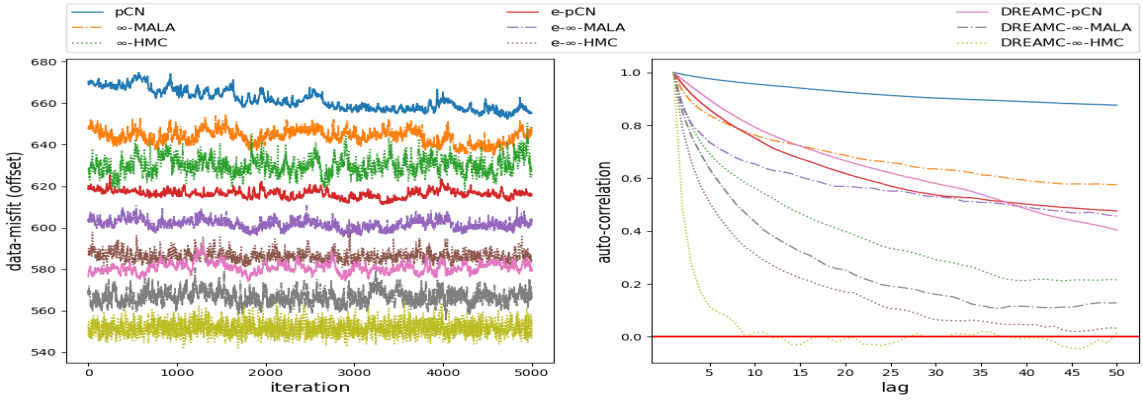
where we use the assumption that $\mathrm{E}[\xi_\ell(t)\xi_{\ell'}(t')] = \mathrm{E}[\xi_\ell(t)\xi_\ell(t')]\delta_{\ell\ell'}$ □

APPENDIX B

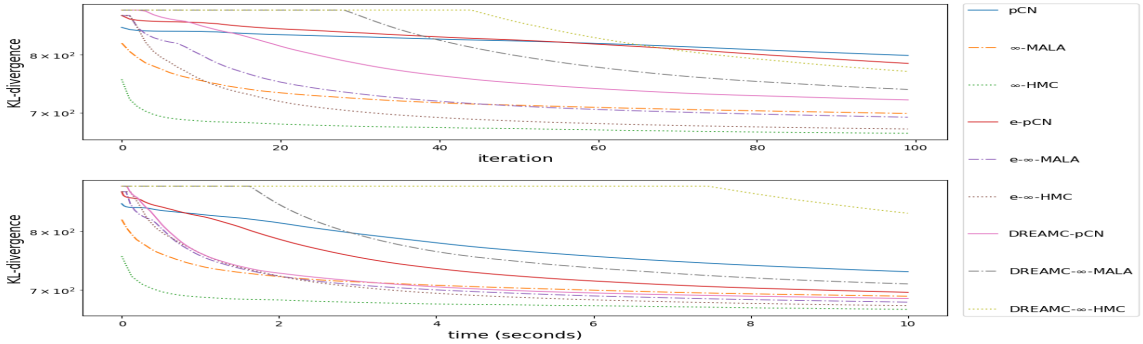
MORE NUMERICAL RESULTS

B.1 UQ

Here I provide more analysis results(misfit, KL divergence, auto-correlation, etc.) for the Elliptic inverse problem and Advection-diffusion inverse problem.



(a) The trace plots of data-misfit function evaluated with each sample (left, values have been offset to be better compared with) and the auto-correlation of data-misfits as a function of lag (right).



(b) The KL divergence between the posterior and the prior as a function of iteration (upper) and time (lower) respectively.

Figure B.1. Advection-diffusion Inverse Problem: Analysis of Posterior Samples

For the advection-diffusion inverse problem (Section 2.4.2), figure B.1a verifies DREAMC ∞-HMC is the most efficient MCMC algorithm that has the smallest autocorrelation as shown in the right panel. It is followed by other HMC algorithms and DREAMC ∞-MALA, which is even better than ∞-HMC. Figure B.1b plots the KL divergence between the posterior and the prior in terms of iteration (top panel) and time (panel) respectively. As we can see, ∞-HMC converges the fastest.

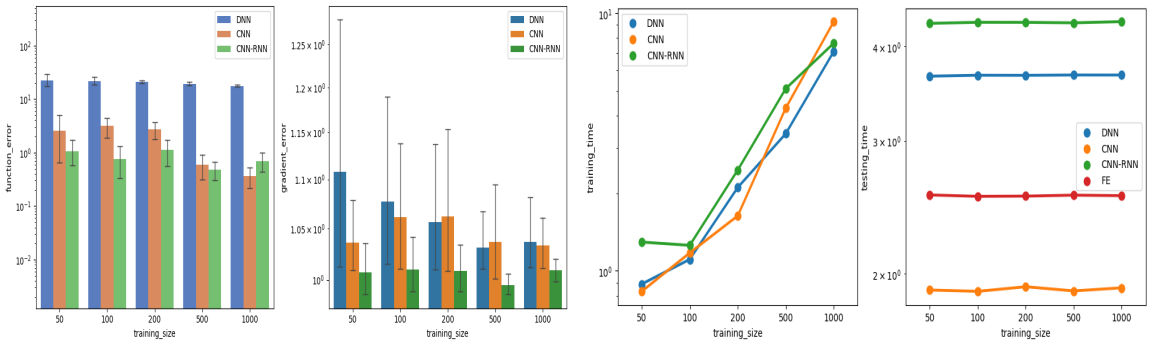


Figure B.2. Comparing the emulation $\mathcal{G}^e : \mathbb{R}^{3413} \to \mathbb{R}^{1280}$ in an advection-diffusion inverse problem (Section 2.4.2) by DNN, CNN and CNN-RNN in terms of error (left) and time (right). Time is also compared with exact calculation of gradients (labeled 'FE') using adjoint codes in testing.

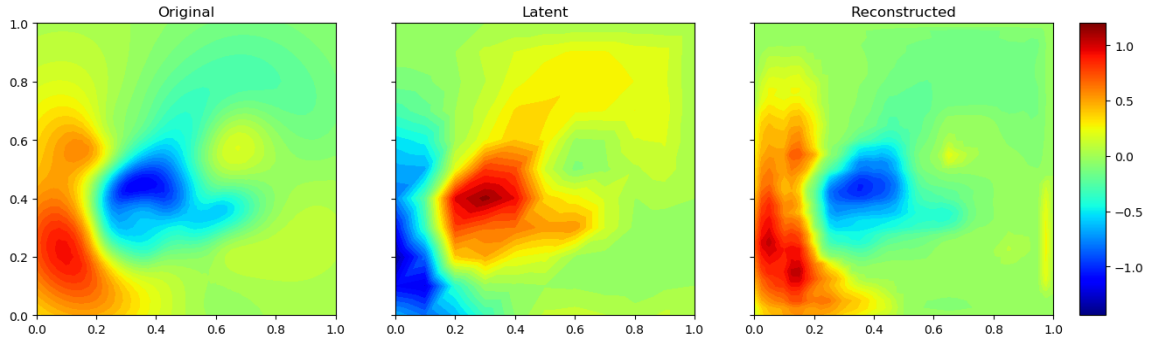


Figure B.3. Elliptic inverse problem: CAE compressing the original function (left) into latent space (middle) and reconstructing it in the original space (right).

Since advection-diffusion (Section 2.4.2) is a spatiotemporal problem, Figure B.2 suggests a combination of CNN (for spatial inputs) and RNN (temporal outputs) might perform better than CNN alone because RNN fits the time series better. The left two panels show CNN-RNN yields the smallest function and gradient errors across different training sets. The right two panels show the computational cost is about the same as other NNs in training. Though it is less efficient in prediction in this example, CNN-RNN seems promising for efficient emulation in inverse problems with spatiotemporal data.

The framework I propose posits that the use of a standard AE and its corresponding latent projection created by dense layers may not be the most effective approach. Instead, implementing a CAE may be a better choice for images. I tested this in the

Elliptic inverse problem as depicted in Figure B.3. This way, the latent parameter can be construed as a portrayal of the original function on a less intricate mesh.

B.2 STIP

Here I provide more results on chaotic dynamic systems Lorenz, Rössler and Chen inverse problems.

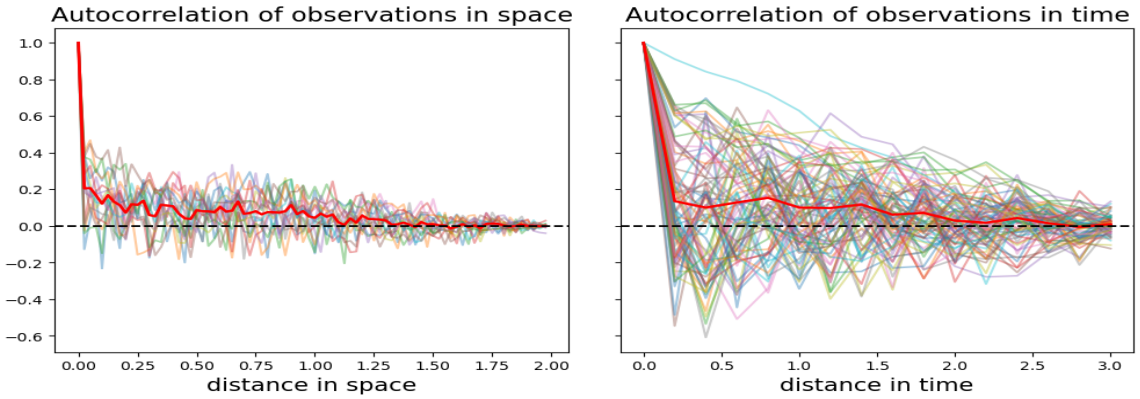


Figure B.4. Advection-diffusion inverse problem: auto-correlations of observations in space (left) and time (right) respectively.

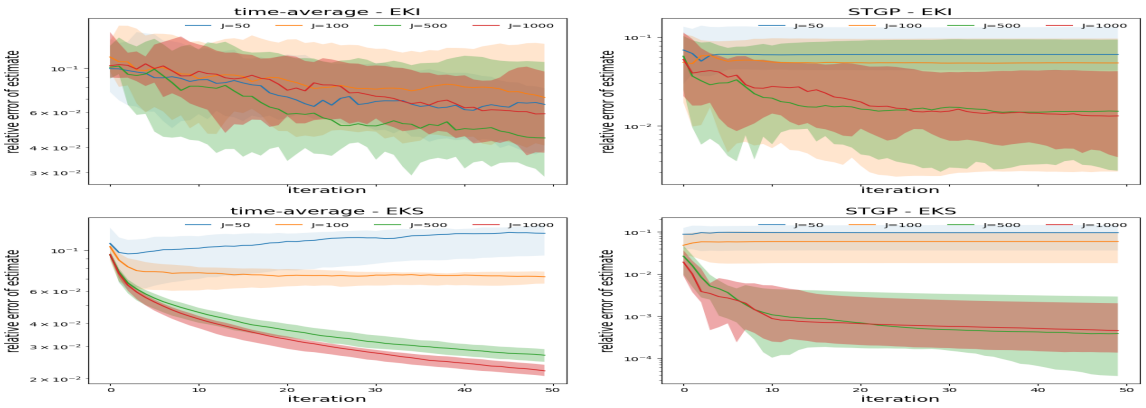


Figure B.5. Lorenz inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS respectively), and shaded regions indicate $5 \sim 95\%$ quantiles of such repeated results.

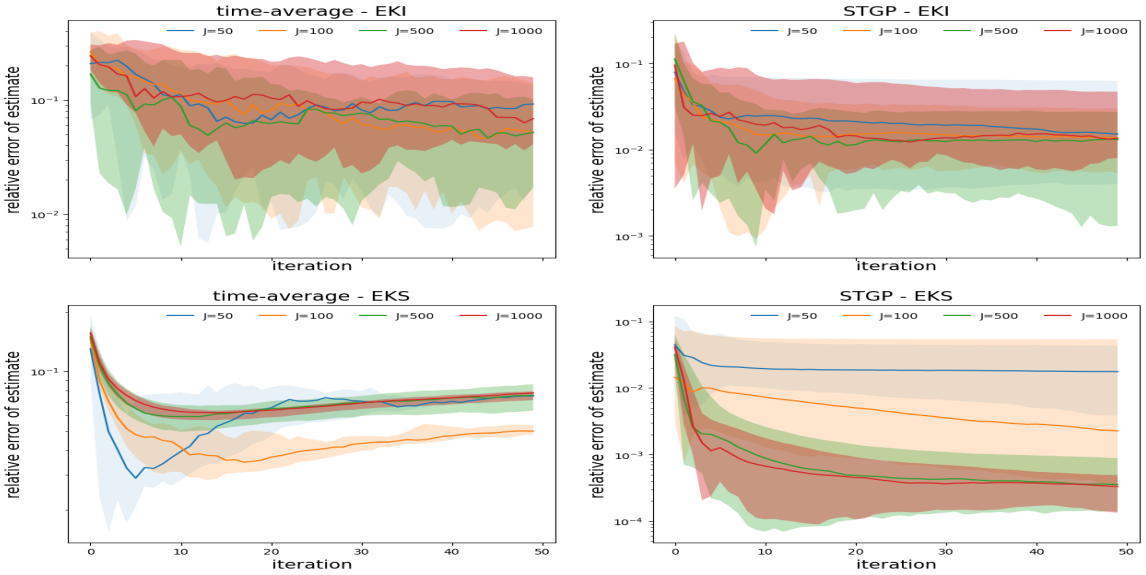


Figure B.6. Rössler inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS respectively), and shaded regions indicate $5 \sim 95\%$ quantiles of such repeated results.

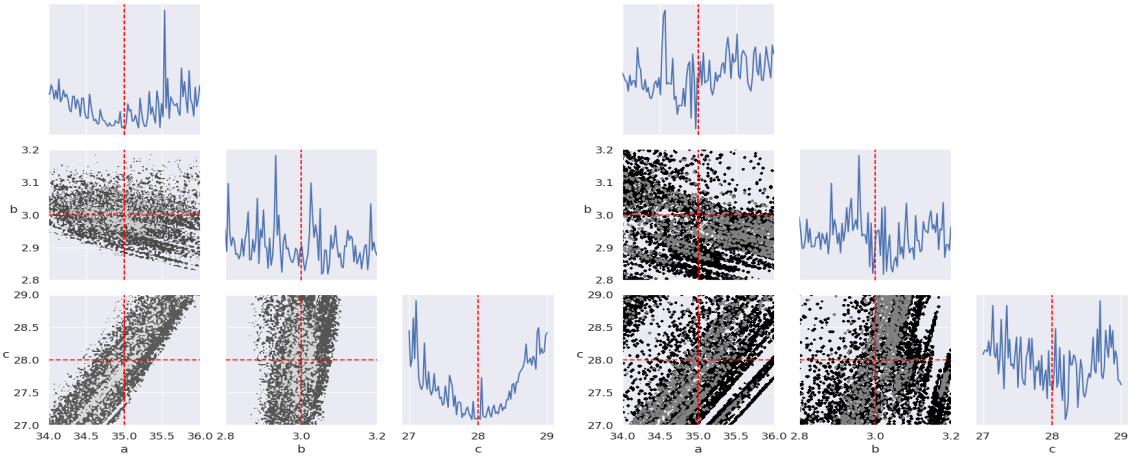


Figure B.7. Chen inverse problem: marginal (diagonal) and pairwise (lower triangle) sections of the joint density $p(u)$ by the time-averaged model (left) and the STGP model (right) respectively. Red dashed lines indicate the true parameter values.

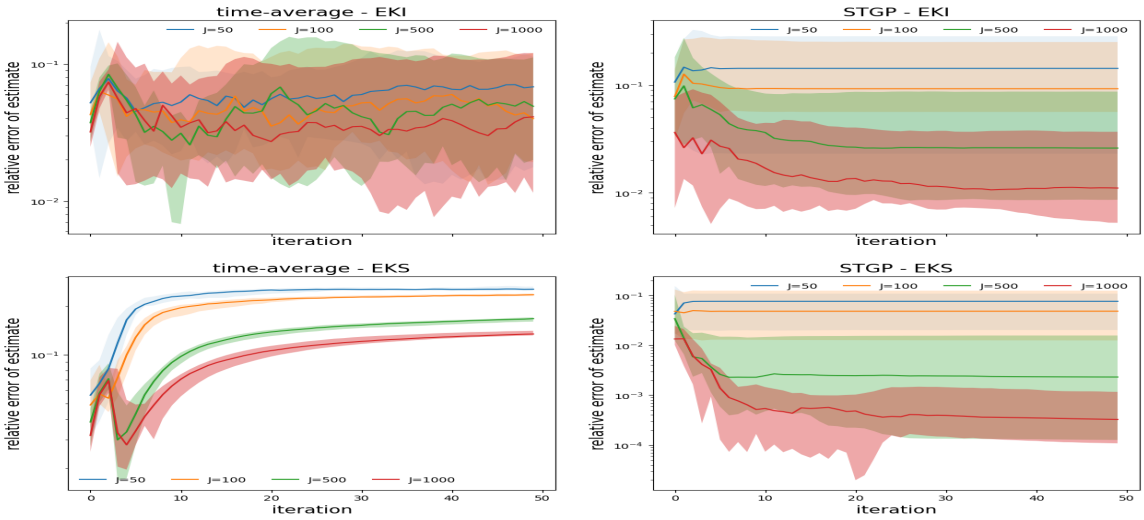


Figure B.8. Chen inverse problem: comparing posterior estimates of parameter u for two models (time-average and STGP) in terms of relative error of median REM $= \frac{\|\hat{u}-u^\dagger\|}{\|u^\dagger\|}$. Each experiment is repeated for 10 runs of EnK (EKI and EKS respectively) and shaded regions indicate $5 \sim 95\%$ quantiles of such repeated results.

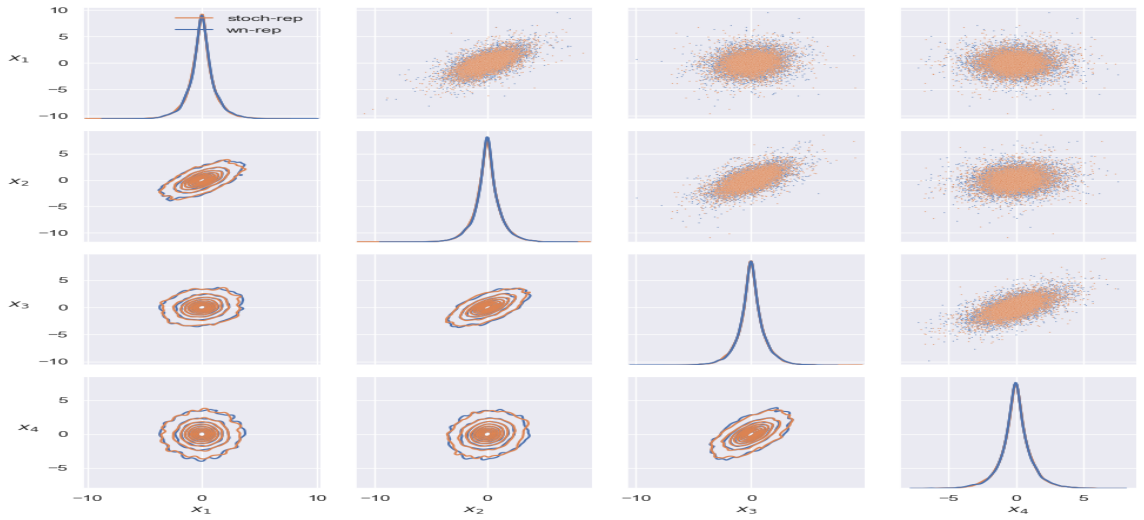


Figure B.9. Comparison in sampling q$-$ED$_d$ using the stochastic representation (4.11) (organge) and the white-noise representation (4.31) (blue). Numerical results show their sampling distributions are indistinguishable. Empirical densities are estimated based on 10000 samples (shown as dots).

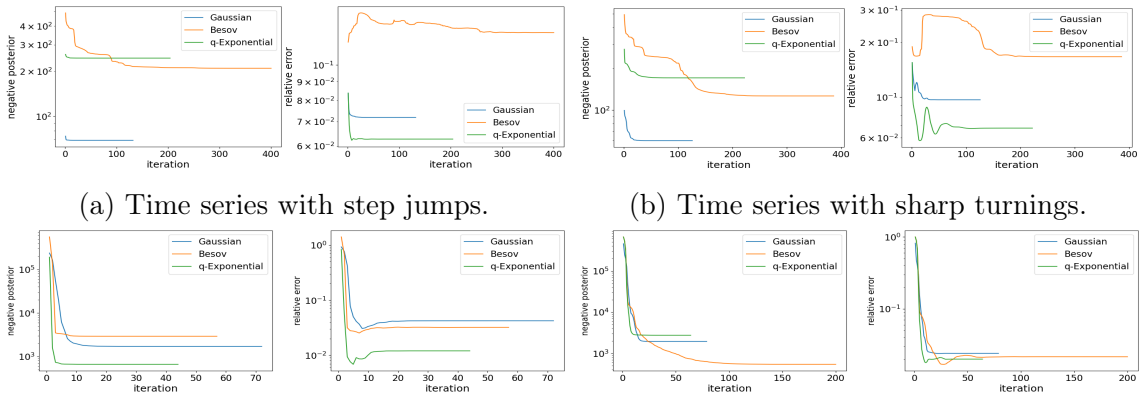
B.3 QEP

In this section, I present some additional numerical experimental results that cannot be included in the main text due to the page limit.

First, I numerically verify the equivalence between the stochastic representation (4.11) and the white-noise representation (4.31) of $\mathrm{q}-\mathrm{ED}_d$ random variable in Figure B.9. More specifically, I generate 10000 samples using each of these two representations and illustrate in Figure B.9 that the two samples yield empirical marginal distributions (1d and 2d) close enough to each other.

B.3.1 Time Series Modeling

For modeling the simulated time series and stock prices, I include the optimization trace of negative (log)-posterior densities and relative errors for the two simulations and two stock prices in Figure B.10. As commented in the main text, these plots show that Q-EP model can converge faster to lower errors compared with GP and Besov models.



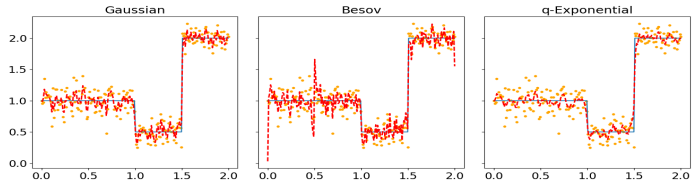
(a) Time series with step jumps.

(b) Time series with sharp turnings.

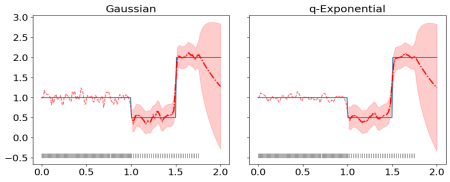
(c) Tesla stock prices in 2022.

(d) Google stock prices in 2022.

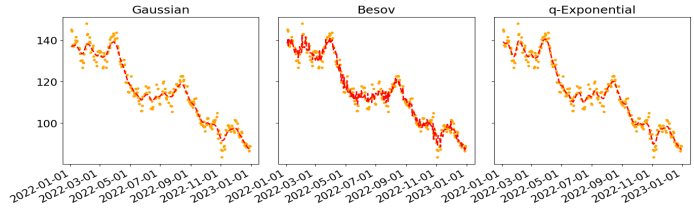
Figure B.10. Negative posterior densities (left) and errors (right) as functions of iterations in the BFGS algorithm used to obtain MAP estimates. Early termination is implemented if the error falls below some threshold or the maximal iteration (1000) is reached. Relative errors are compared against truth in the simulation and the actual data in the Tesla stock.



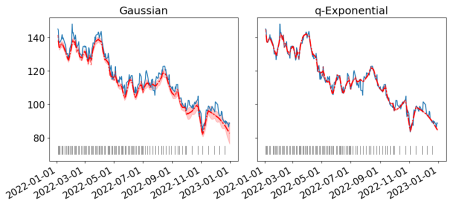
(a) Time series(TS) with step jumps(model fitting).



(b) TS with jumps(prediction).



(c) 2022 Google stock prices(model fitting).



(d) Google stock prices(prediction).

Figure B.11. (a)(c) MAP estimates by GP (left), Besov (middle) and Q-EP (right) models. (b)(d) Predictions by GP (left) and Q-EP (right) models. Orange dots are actual realizations (data points). Blue solid lines are true trajectories. Black ticks indicate the training data points. Red dashed lines are MAP estimates. Red dot-dashed lines are predictions with shaded region being credible bands.

Table B.1. Posterior estimates of Tesla and Google stock prices by GP, Besov and Q-EP prior models: RMSE $:= \|\overline{u} - u\|_2$. Results are repeated 10 times with different random seeds.

	Tesla			Google		
	GP	Besov	Q-EP	GP	Besov	Q-EP
RMSE	171.8515	90.3086	**83.8130**	20.4095	25.2012	**18.3597**
std(RMSE)	1.8018	1.1478	2.6949	0.7115	0.1698	0.9617

Next, I compare MAP estimates by GP, Besov and Q-EP models in Figure B.11a for simulated time series with step jumps and in Figure B.11c for the Google stock prices in 2022. I also investigate the prediction results by GP and Q-EP in these two examples in Figures B.11b and B.11d. Table B.1 summarizes the RMSE of estimated stock prices by the three models and its standard deviation for repeating the experiments 10 times independently.

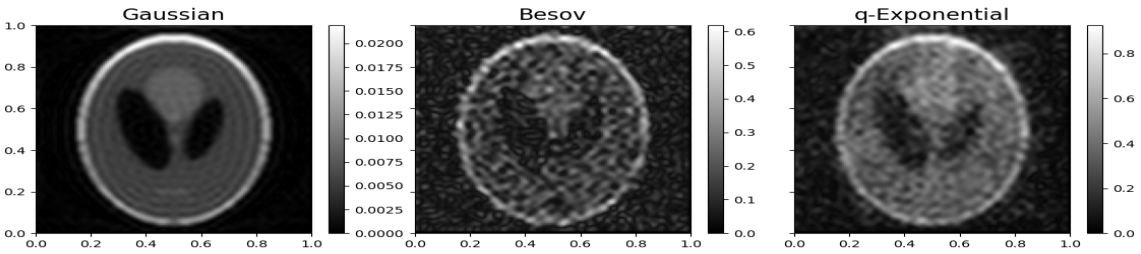


Figure B.12. Shepp–Logan phantom: uncertainty field (posterior standard deviation) given by GP, Besov and Q-EP models. GP tends to underestimate the uncertainty values (about 1% of that with Q-EP).

B.3.2 Computed Tomography Imaging

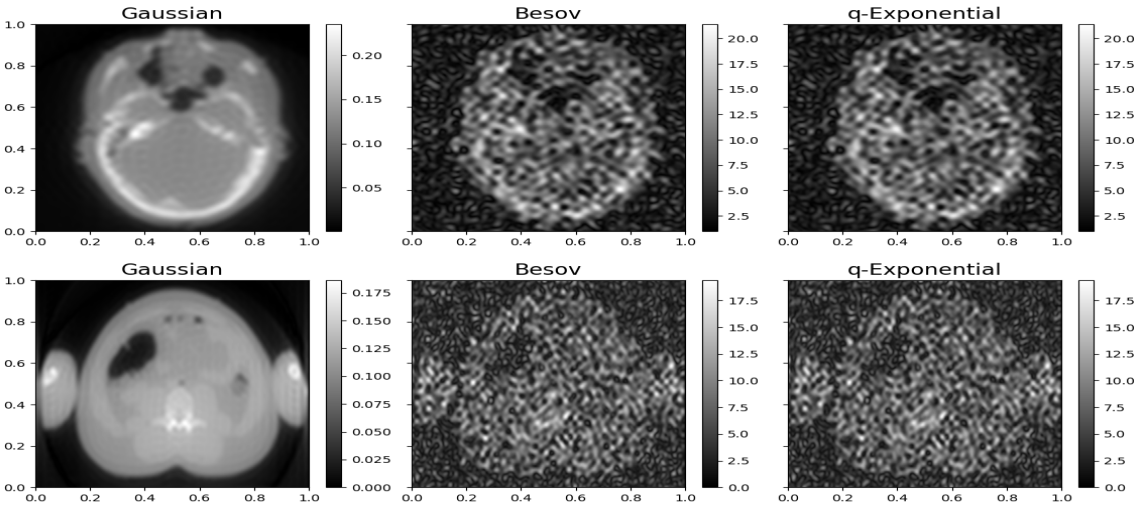


Figure B.13. CT of human head (upper) and torso (lower): uncertainty field (posterior standard deviation) given by GP, Besov and Q-EP models. Note GP tends to underestimate the uncertainty values (about 1% of that with Q-EP).

In the problem of reconstructing human head and torso CT images, Table B.2 compares GP, Besov and Q-EP models in terms of relative error (RLE), log-likelihood (LL), and imaging quality metrics including PSNR, SSIM and HarrPSI. In most cases, Q-EP outperforms or achieves comparable scores with the other two methods.

Lastly, Figures B.12 and B.13 show that the posterior standard deviations esti-

Table B.2. MAP estimates for CT of human head and torso by GP, Besov and Q-EP prior models: relative error, RLE $:= \|\hat{u} - u^\dagger\|/\|u^\dagger\|$ of MAP ($\hat{u} = u^*$), log-likelihood (LL), PSNR, SSIM and HarrPSI.

	Head			Torso		
	GP	Besov	Q-EP	GP	Besov	Q-EP
RLE	0.2999	0.2241	**0.2224**	0.2611	0.2177	**0.2153**
LL	-4.05e+5	-1.12e+4	-1.17e+4	-3.30e+5	-3.86e+3	-4.37e+3
PSNR	24.2321	26.7633	**26.8281**	23.6450	25.2231	**25.3190**
SSIM	0.7010	0.7914	**0.8096**	0.5852	**0.6983**	0.6982
HaarPSI	0.0525	**0.0593**	0.0587	0.0666	**0.0732**	0.07190

mated by wn-pCN using GP model could be misleading because the seemingly more recognizable shape deludes the fact that they are about two orders of magnitude smaller in value compared with the other two models. This implies that GP might underestimate the uncertainty present in the observed sinograms in the CT imaging analysis.

B.4 STBP

In this section, I provide more numerical results on STEMPO and Emoji problem, including MAP estimates in the original space and posterior sample.

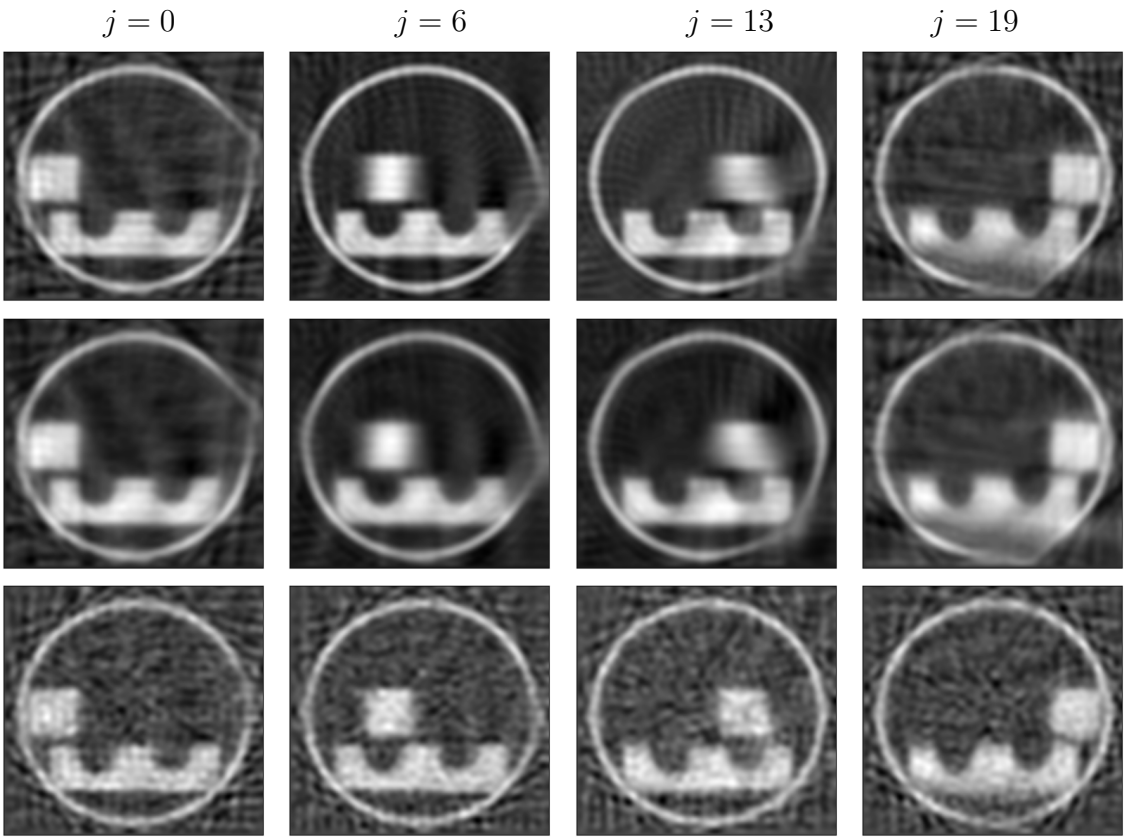


Figure B.14. Reconstruction results of dynamic STEMPO test problem in original space. Row from top to bottom: MAP for $q = 1, p = 1$, MAP for $q = 2, p = 2$ and MAP $q = 1, p = 1$ without time correlation. Left to right: time step $t = 0, 6, 13, 19$.

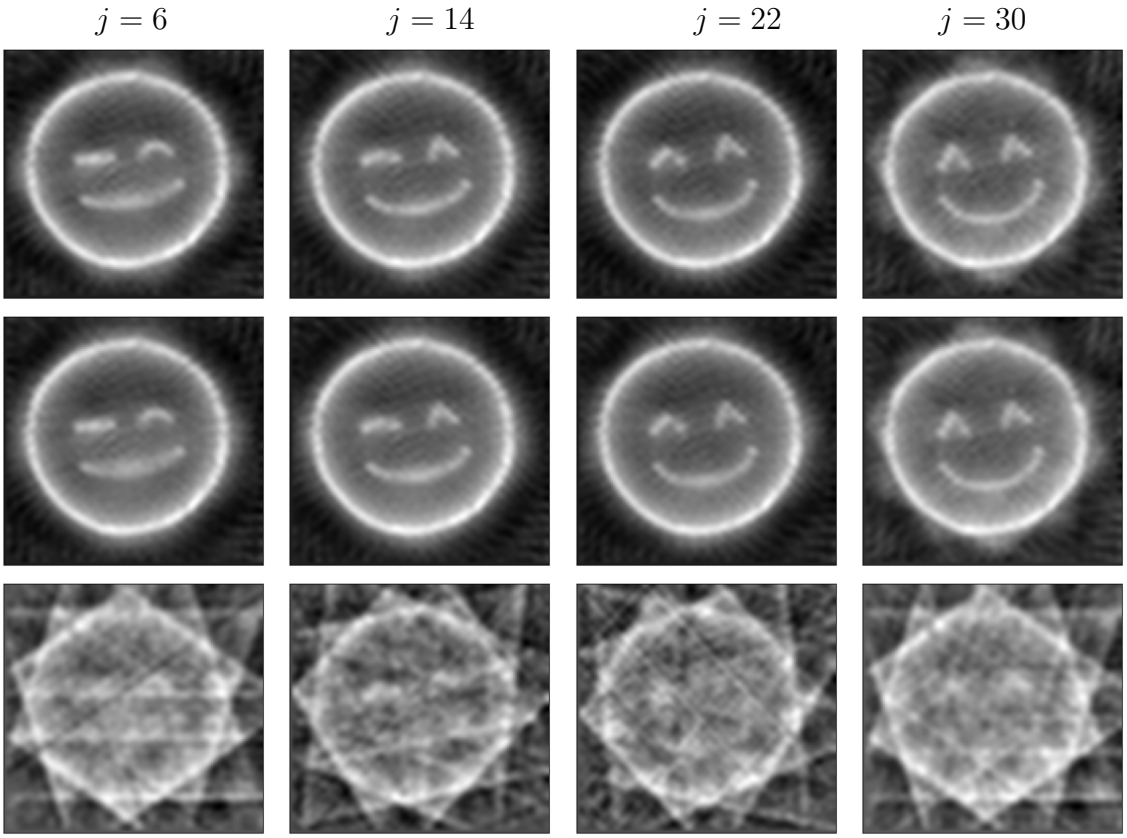


Figure B.15. Reconstruction results for the emoji problem with $n_a = 10$ in the original space. Row from top to bottom: MAP for STBP ($q = 1, p = 1$), MAP for STGP ($q = 2, p = 2$) and MAP for time-uncorrelated model. Left to right: time step $t = 6, 14, 22, 30$.

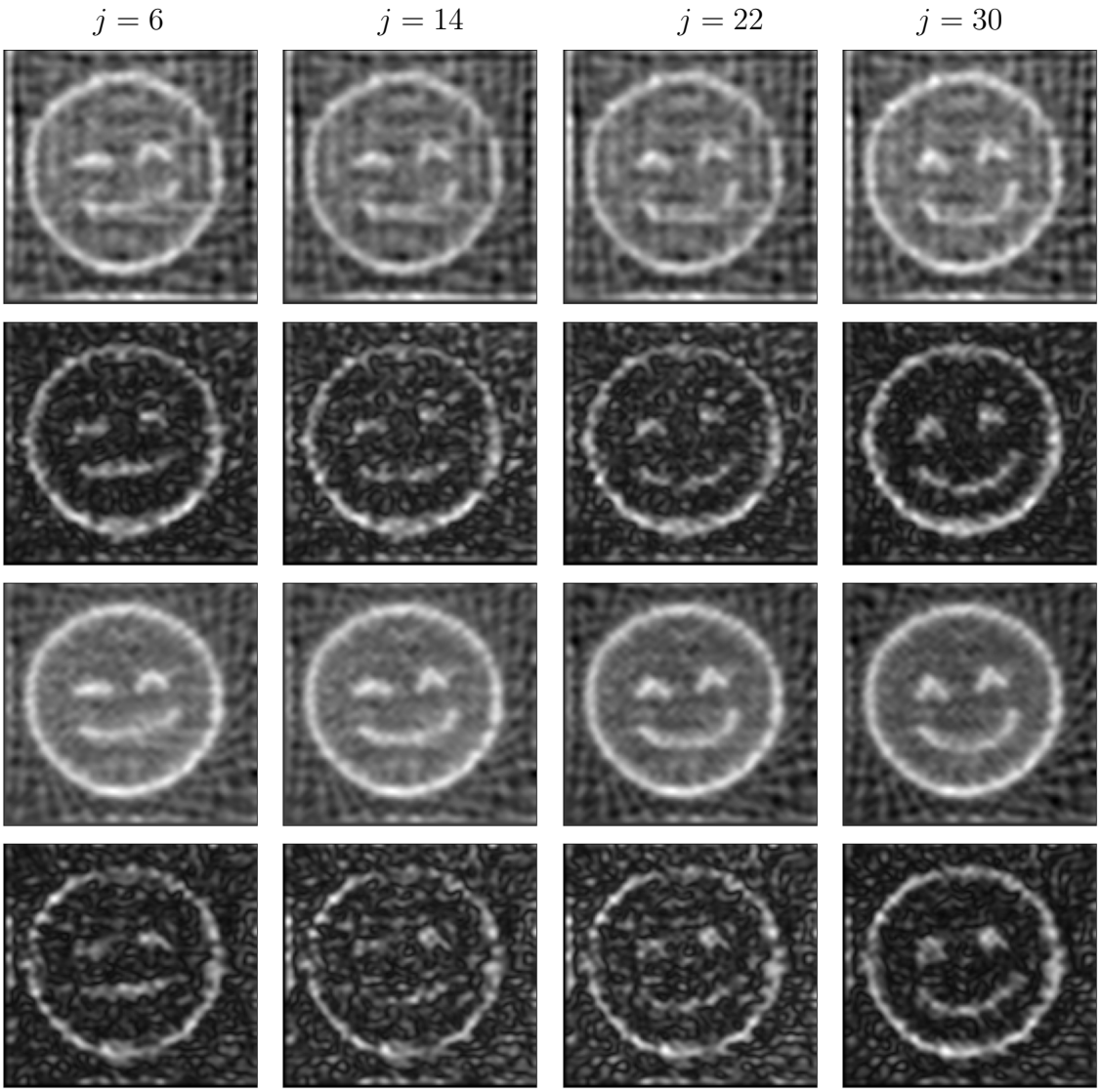


Figure B.16. MCMC results of the emoji problem with $n_a = 10$ in the whitened space. Row from top to bottom: posterior mean for STBP ($q = 1, p = 1$), posterior standard deviation for STBP ($q = 1, p = 1$), posterior mean for STGP ($q = 2, p = 2$), and posterior standard deviation for STGP ($q = 2, p = 2$). Left to right: time step $t = 6, 14, 22, 30$.