DOCTORAL THESIS

Accelerating Bayesian sampling in Uncertainty Quantification

Author: Gianluca Detommaso
Supervisors: Prof. Dr. Robert Scheichl & Prof. Dr. Andreas Kyprianou

EPSRC Centre for Doctoral Training in Statistical Applied Mathematics (SAMBa)

Department of Mathematical Sciences
University of Bath, United Kingdom (UK)

Contents

■ Acknowledgments

To my supervisor, Prof. Dr. Robert Scheichl
For everything you taught me
in maths
as in life
For all the opportunities
and the guidance
I am grateful to you.

Thank you, Rob.

A special acknowledgement goes to Dr. Tiangang Cui, who has played the role of unofficial co-supervisor from the very start of my PhD. Thank you TC.

I would like to thank the co-directors of SAMBa, my PhD programme: Dr. Susie Douglas, Prof. Dr. Andreas Kyprianou and Prof. Dr. Paul Milewski. What you are doing is amazing; keep it up!

Finally, a big thank you to all my collaborators and, in general, to the multitude of people I learned from along the road, in particular Prof. Dr.Youssef Marzouk, Prof. Dr. Colin Fox, Dr. Tim Dodwell, Dr. Alessio Spantini, Dr. Tony Shardlow and Dr. Kari Heine. It's been a nomadic, vibrant, brilliant PhD experience. You all contributed to an adventure that, today as in the future, I will always be able to look back at and be proud of.

Thank you.

■ Overview

Subject & goals

This thesis is a sequel of projects that I worked on during my PhD, aimed to accelerate sample-based inference in high-dimensional and computationally expensive scenarios. This is a recurrent framework in the field of Uncertainty Quantification, where complex physical models define the relation between hidden parameters and observable outputs. Sampling algorithms constitute a fundamental methodology to perform inference in very high-dimensional problems, as their rate of convergence is, in many cases, dimension independent. Unfortunately, such rate is also very slow: a huge amount of samples is often required to reduce the variance to an acceptable level of accuracy. Because the models we consider are typically computationally expensive to evaluate, the realization of each single sample is itself expensive, making classic algorithms unfeasible in a reasonable amount of time. A first question that this thesis attempts to address is how to exploit model hierarchies in order to reduce the complexity per sample and, in turn, the overall computational cost. Papers I and II strive in this direction.

Complexity per sample is not the only problem. Parameter space exploration is in general one of the main challenges of high-dimensional inference. In fact, it is computationally unfeasible to calculate many evaluations of the model at random locations without any extra information, in particular when evaluations are expensive. Algorithms need local, possibly global, geometric information about the structure of the probability space in order to lead the exploration to sensible locations. A second question that this thesis attempts to provide answers to is how to incorporate geometric and low-rank information into algorithms that can perform high-quality inference at scale. Papers II, III and IV carry out research relevant to these issues.

A third goal of this thesis is to develop novel methodologies that can perform inference faster than classic ones. Particularly, we focus on the field of transport maps, where samples from an untreatable probability distribution are produced as transportation of samples from a treatable one. The challenge is to define transport maps that are easy to compute, feasible to optimize and whose performance scales with the dimensionality of the problem. Papers III, IV and V develop methodologies in such perspective.

Structure of the thesis

■ Introduction

The Introduction provides an overarching background to the main concepts and methodologies that stand at the foundations of the following Papers. Section 1 introduces the reader to a qualitative understanding of the motivations behind the needs for uncertainty. Sections 2 and 3 differentiate between forward and backward Uncertainty Quantification. Thereby, we motivate the frameworks in Section 2.1 and 3.1, we introduce the reader to a typical model problem in Section 2.2, to the concept of random fields in Section 2.3 and to Bayesian inverse problems in Section 3.2. From an algorithmic point of view, we treat sampling-based methods such as Monte Carlo in Section 2.5, Multilevel Monte Carlo in Section 2.6, Markov Chain Monte Carlo in Section 4.1, Variational inference and transport maps in Section 4.2.

■ Paper I

Continuous Level Monte Carlo and Sample-Adaptive Model Hierarchies [40]

Authors: Gianluca Detommaso, Tim Dodwell, Rob Scheichl

Abstract. In this paper, we present a generalization of the multilevel Monte Carlo (MLMC) method to a setting where the level parameter is a continuous variable. This continuous level Monte Carlo (CLMC) estimator provides a natural framework in PDE applications to adapt the model hierarchy to each sample. In addition, it can be made unbiased with respect to the expected value of the true quantity of interest provided the quantity of interest converges sufficiently fast. The practical implementation of the CLMC estimator is based on interpolating actual evaluations of the quantity of interest at a finite number of resolutions. As our new level parameter, we use the logarithm of a goal-oriented finite element error estimator for the accuracy of the quantity of interest. We prove the unbiasedness, as well as a complexity theorem that shows the same rate of complexity for CLMC as for MLMC. Finally, we provide some numerical evidence to support our theoretical results, by successfully testing CLMC on a standard PDE test problem. The numerical experiments demonstrate clear gains for samplewise adaptive refinement strategies over uniform refinements.

Personal contribution. In this work, I originated the main ideas, I developed the theoretical framework with related proofs, I implemented the Multilevel and Continuous Level Monte Carlo algorithms and wrote most of the paper. Description and numerical implementation of the PDE model and Finite Element

Methods were taken care by Tim Dodwell. Each author contributed to final revisions.

■ Paper II

Multilevel dimension-independent likelihood-informed MCMC for large-scale inverse problems [34]

Authors: Tiangang Cui, Gianluca Detommaso, Robert Scheichl

Abstract. We present a non-trivial integration of dimension-independent likelihood-informed (DILI) MCMC [35] and the multilevel MCMC [44] to explore the hierarchy of posterior distributions. This integration offers several advantages: First, DILI-MCMC employs an intrinsic likelihood-informed subspace (LIS) [36] – which involves a number of forward and adjoint model simulations – to design accelerated operator-weighted proposals. By exploiting the multilevel structure of the discretised parameters and discretised forward models, we design a Rayleigh-Ritz procedure to significantly reduce the computational effort in building the LIS and operating with DILI proposals. Second, the resulting DILI-MCMC can drastically improve the sampling efficiency of MCMC at each level, and hence reduce the integration error of the multilevel algorithm in [36] for fixed CPU time. To be able to fully exploit the power of multilevel MCMC and to reduce the dependencies of samples on different levels for a parallel implementation, we also suggest a new pooling strategy for allocating computational resources across different levels and constructing Markov chains at higher levels conditioned on those simulated on lower levels. Numerical results confirm the improved computational efficiency of the multilevel DILI approach.

Personal contribution. In this work, I contributed to the theoretical part of the paper, I implemented Multilevel MCMC and merged it with Tiangang Cui's toolbox for inference in Bayesian inverse problems, and I wrote part of the paper. Each author contributed to final revisions.

■ Paper III

A Stein variational Newton method [41]

Authors: Gianluca Detommaso, Tiangang Cui, Alessio Spantini, Youssef Marzouk, Robert Scheichl

Abstract. Stein variational gradient descent (SVGD) was recently proposed as a general purpose nonparametric variational inference algorithm [91]: it minimizes the Kullback-Leibler divergence between the target distribution and its approximation by implementing a form of functional gradient descent on a reproducing kernel Hilbert space. In this paper, we accelerate and generalize the

SVGD algorithm by including second-order information, thereby approximating a Newton-like iteration in function space. We also show how second-order information can lead to more effective choices of kernel. We observe significant computational gains over the original SVGD algorithm in multiple test cases.

Personal contribution. In this work, I originated the main ideas, I developed the theoretical framework together with Alessio Spantini, I implemented all numerical experiments and I wrote most of the paper. Each author contributed to final revisions.

■ Paper IV

Stein Variational Online Changepoint Detection with Applications to Hawkes Processes and Neural Networks

Authors: Gianluca Detommaso, Hanne Hoitzing, Tiangang Cui, Ardavan Alamir

Abstract. Bayesian online changepoint detection (BOCPD) [3] offers a rigorous and viable way to identify changepoints in complex systems. In this work, we introduce a Stein variational online changepoint detection (SVOCD) method to provide a computationally tractable generalization of BOCPD beyond the exponential family of probability distributions. We integrate the recently developed Stein variational Newton (SVN) method [41] and BOCPD to offer a full online Bayesian treatment for a large number of situations with significant importance in practice. We apply the resulting method to two challenging and novel applications: Hawkes processes and long short-term memory (LSTM) neural networks. In both cases, we successfully demonstrate the efficacy of our method on real data.

Personal contribution. In this work, I originated the main ideas, I developed the theoretical framework, I implemented all numerical experiments, except the neural network architecture (which was written by Ardavan Alamir) and I wrote most of the paper. Hanne Hoitzing contributed to the numerical experiments and to refine the writing of the papers. Each author contributed to final revisions.

■ Paper V

HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference

Authors: Jakob Kruse, Gianluca Detommaso, Robert Scheichl, Ullrich Köthe

Abstract. A large proportion of recent invertible neural architectures is based on a coupling block design. It operates by dividing incoming variables into two sub-spaces, one of which parameterizes an easily invertible (usually affine) transformation that is applied to the other. While the Jacobian of such a transformation is triangular, it is very sparse and thus may lack expressiveness. This work presents a simple remedy by noting that (affine) coupling can be repeated recursively within the resulting sub-spaces, leading to an efficiently invertible block with dense triangular Jacobian. By formulating our recursive coupling scheme via a hierarchical architecture, HINT allows sampling from a joint distribution $p(y, x)$ and the corresponding posterior $p(x|y)$ using a single invertible network. We demonstrate the power of our method for density estimation and Bayesian inference on a novel data set of 2D shapes in Fourier parameterization, which enables consistent visualization of samples for different dimensionalities.

Personal contribution. In this work, I originated the main ideas, I developed the theoretical framework, I implemented the numerical experiments (except for the Invertible Neural Network part, which was handled by Jakob Kruse) and I wrote most of the paper. Jakob Kruse produced most of the figures. Each author contributed to final revisions.

■ Introduction

1 A qualitative introduction to uncertainty

Wrong models.

"*All models are wrong, but some are useful*" - George Box.

During my PhD, I heard this sentence over and over, but I still find it very insightful. A model is a rigorous mathematical description of potentially complex phenomena. It typically attempts to explain the relation between inputs and outputs, depending on a possibly infinite set of model parameters. A simple example is the motion of a pendulum. Given an initial position, a second-order ordinary differential equation (ODE) can model the position of the pendulum after a fixed time with great accuracy. Although useful, at its simplest this model is wrong. In fact, it does not take into account mechanical friction or air resistance, which will inevitably create an error in the results. Another example is Black-Scholes, a financial model mathematically describing the dynamics of derivative investments via a partial differential equation (PDE). As the Black-Scholes model is just a mathematical abstraction, it is fundamentally wrong; however, it historically led to a revolution in option trading and it is still widely used (with some modifications) by option marketers (one may argue that the Black-Scholes model is still useful *because* it is still widely used).

Types of uncertainty. The discrepancy between models and reality leads to a structural form of uncertainty, but this is far from being the only type. In fact, uncertainty has many sources, like measurement errors, experimental variability, numerical errors and approximations, imprecise model parameters and lack of observations. In general, uncertainty is often classified into two major categories: *aleatoric* (statistical) uncertainty, which is due to information that cannot be sufficiently determined via current available measurement devices or is inherently random; *epistemic* (systematic) uncertainty, which is due to information that could be measured and to effects that could be modeled, but that are currently not.

Although in most real applications it is impossible to eliminate aleatoric uncertainty, an insightful quantification of epistemic uncertainty can help the scientist to refine its modelling assumption and experimental design, thereby reducing it until a satisfactory level of accuracy is reached.

Uncertainty Quantification. Whenever a decision has to be made, it is desirable to dispose of a risk measure that quantifies the credibility of our assessments. In many applications, a single number outcome might have very little meaning, as a small measurement or inference error could have led to a completely different result. Uncertainty can obviate this problem by assuming the role of risk measure, rigorously quantifying how much our decisions are sensible to random elements in the system.

Similarly to the distinction between probability and statistics, the field of Uncertainty Quantification (UQ) is usually divided into two categories: *forward* and *inverse*. Forward UQ is the quantification of uncertainties that propagate from the inputs of a model to its outputs. For example, consider a classic predator-prey (Lotka-Volterra) model (see Section 2), which describes the demographic interaction between two species. Fixing a random probability distribution for the number of preys and predators today, one may be interested in studying how the uncertainty propagates through the model and how the probability distribution looks like tomorrow. On the other hand, inverse UQ starts from actual observations and attempts to recover parameter values that may have generated such data, or probability distribution of them. Coming back to our example, given that we have observed the number of predators today - after all, preys are difficult to observe because they stay hidden - we may want to infer what was the number of predators and prey yesterday and thus make predictions about the future.

The Bayesian approach. Reasonably, any decision process is necessarily accompanied by an implicit or explicit assessment of *the extent to which* current information should be trusted. Without any quantifiable level of credibility attached to the available information, decision makers will have to compensate with their own *a priori* belief (I would not necessarily invest all my money in a single stock because data tells me to; would you?). On the other hand, as credibility level increases, subjective belief should be progressively abandoned in favor of scientific evidence.

Mathematically, it is both reasonable and desirable to include subjective belief into quantification of uncertainty. This balance between *a priori* and observable information is at the very heart of Bayesian statistics. The key point of the Bayesian formulation is that each unknown quantity is modeled as a random variable characterized by a probability distribution, which encodes our a priori knowledge of the quantity, and its structural dependence on other variables.

"*There's no such thing as no information*" - Colin Fox.

Although we might think that we know nothing about a certain quantity or process, lack of information is, itself, information. Positivity and dimensionality are examples of properties that we can use to construct an appropriate prior distribution, which could range from totally non-informative to very concentrated around some value.

Given a prior distribution and a likelihood (i.e. probabilistic) model, Bayes' theorem provides an easy and effective way to characterize (up to a normalization constant) the *posterior* probability distribution of unknown quantities given the observations (see Bayes' Theorems 3 and 4). The posterior fully describes our subjective uncertainty given our current state of information, it can be used to produce valuable statistics, make decisions and formulate effective strategies.

Especially in applications where a large amount of data information is lost, compressed or corrupted by noise, the Bayesian approach is particularly attractive. The introduction of a priori knowledge constitutes a principled way to compensate, or more rigorously *regularize*, lack of information and noise. In fact, there is a direct link between prior distribution and regularization terms in optimization, where the latter are classically adopted to fight non-identifiability and ill-posedness. The Bayesian interpretation provides a probabilistic justification to regularization, and allows to recover the optimal solution as a mode of the posterior distribution, together with its uncertainty.

2 Forward Uncertainty Quantification

2.1 Motivational example

Forward UQ studies how uncertainty propagates from inputs to outputs of a given model. Uncertainty may reside in several sources: input coefficients, for example, could be unknown [26, 96]; the model could be intrinsically random, like in the case of stochastic differential equations [109]; the shape of a domain [103] and its boundary conditions could again be unknown [24].

Let us look at an example. Consider a predator-prey (Lotka-Volterra) [92, 16] model, already mentioned in Section 1. A population of preys and predators, that we respectively denote by x_1 and x_2, interacts in a biological system. The interaction is captured by a couple of ordinary differential equations (ODEs) [32], namely

$$\begin{cases} \frac{dx_1}{dt} = \alpha x_1 - \beta x_1 x_2 \,, \\ \frac{dx_2}{dt} = \gamma x_1 x_2 - \delta x_2 \,, \end{cases} \tag{1}$$

where α, β and γ, δ are rate parameters describing growth and death rates for preys and predators, respectively. Logically, preys' growth rate only depends on the number of preys themselves, whereas their death rate also depends on the number of predators. Vice versa for predators. There are multiple ways of injecting uncertainty in system 1. For example, the initial conditions may be unknown, or the rate parameters, or both. If we want to introduce model misspecification, we could even add a random noise component to the dynamic itself. To keep it simple, let us restrict the study to unknown initial conditions. The essence of forward UQ is to assume a probability distribution over unknown quantities and study how uncertainty propagates through the model. Let us take $x_1(0), x_2(0) \overset{i.i.d}{\sim} \mathcal{U}[1,2]$, where $\mathcal{U}[1,2]$ denotes a uniform distribution with support $[1,2]$, we set $\alpha = 2/3$, $\beta = 4/3$, $\gamma = \delta = 1$, and assume that we are interested in $x_1(T), x_2(T)$, for $T = 10$. Figure 1 displays $N = 1000$ simulated dynamics, where the blue dots correspond to the random initial population configurations, whereas the red dots are the final ones. It is instructive to notice how the probability distribution of the population size at the final time is very different from the simple uniform distribution that we imposed at the initial one. In addition, although we may consider the mean of the uniform distribution as a good candidate for the initial population sizes, the black trajectory in the figure shows that its counterpart at the final time is not such anymore. This indicates how non-linear interactions in the model can lead uncertainty to unexpected propagation of uncertainty, and how small errors in the inputs can have a drastic impact in the outputs. Proper uncertainty quantification should therefore be conducted to provide not only estimators of the quantity that we are interested in, but also a risk measure on the sensibility of our outputs to random effects in the system.

Since the solution of the ODE system (1) has to be simulated numerically, the resulting output distribution is generally biased. In order to reduce such error, one should refine the discretization of the model up to a satisfactory threshold. Better accuracy, however, is computationally expensive. If the number of trajectory that has to be simulated is high, the overall inference method may result very slow. Hence, it is important to develop methodologies that can reduce the number of model evaluation while keeping a satisfactory accuracy.

2.2 A common model problem.

The example in Section 2.1 is a simple, low-dimensional model that already displays some of the main difficulties of forward UQ. In this Section, we will mathematically generalize this framework to potentially inifinite-dimensional computationally intensive model problems, and we will focus on a test case that has been commonly used in the literature.

Consider a function $\mathcal{F} : \mathcal{X} \to \mathcal{S}$ representing an input-to-output model, where $\mathcal{X}$ is a space of model parameters and $\mathcal{S}$ is a space of possible model outcomes. We will address $\mathcal{F}$ as the *forward model*. In general, the function $\mathcal{F}$ can be any parametric relation between inputs and outputs. For sake of presentation, we will focus on $\mathcal{F}$ being a particular example of a computationally expensive model depending on an infinite-dimensional set of parameters. In fact, let $\mathcal{F}$ denote the solution of an elliptic partial differential equation (PDE) with random inhomogeneous coefficients [92, 96]. The goal is to study how the uncertainty in the coefficients propagates to the model outputs, and eventually to some quantity of interest. In this scenario, $\mathcal{X}$ is the space of functions $z : D \times \Omega \to \mathbb{R}$ describing the coefficient values for every physical coordinate $\boldsymbol{x} \in D \subset \mathbb{R}^2$ and random realization $\omega \in \Omega$, where D and Ω respectively represent a bounded physical domain and a sample space. On the other hand, $\mathcal{S}$ represents the space

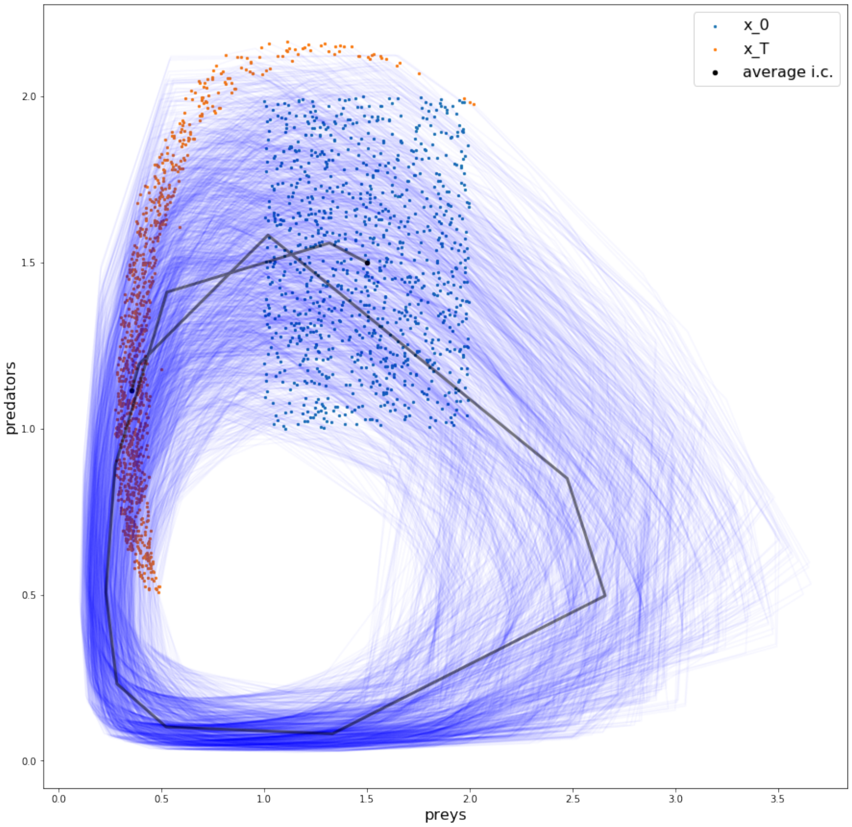


Figure 1: Predator-prey phase-space plot

of solutions $u : D \times \Omega \to \mathbb{R}$ characterized by

$$\nabla_{\boldsymbol{x}} \cdot \left(e^{z(\boldsymbol{x},\omega)} \nabla_{\boldsymbol{x}} u(\boldsymbol{x},\omega) \right) = f(\boldsymbol{x}), \quad \text{for } (\boldsymbol{x},\omega) \in D \times \Omega\,, \tag{2}$$

where $f : D \to \mathbb{R}$ is a source-term function, $\nabla_{\boldsymbol{x}}$ indicates the gradient operator with respect to $\boldsymbol{x}$ and $\nabla_{\boldsymbol{x}}\cdot$ the divergence operator. Under regularity conditions over z, there exist a unique solution to (2) [96]. Thus, the forward operator mapping $\mathcal{F}(z) = u$ is well-defined, where $u \in \mathcal{S}$ is a solution of equation (2) given the random input z.

Equation (2) has been vastly studied in UQ to model steady state, single phase, incompressible flow through porous media [126]. As in many application the composition of the porous medium is mostly unknown or at least uncertain, it is sensible to model the log-diffusion coefficients $z(\boldsymbol{x},\omega)$ as a random field [92, 122], that is a random function over the physical domain D. As a consequence, the solution $u(\boldsymbol{x},\omega)$ depends in turn on the random realization $\omega \in \Omega$, and it is inherently random.

2.3 Random field characterization.

A random field can be seen as a stochastic process $z : D \times \Omega \to \mathbb{R}$, usually defined such that $z(\boldsymbol{x}, \cdot) \in L^2(\Omega, \mathbb{P}; \mathbb{R})$ for each $\boldsymbol{x} \in D$, for some probability measure $\mathbb{P}$. Without loss of generality, let us consider a zero-mean random field, i.e. $\mathbb{E}_{\mathbb{P}}[z(\boldsymbol{x}, \cdot)] = 0$ for each $\boldsymbol{x} \in D$. Then, a covariance function $C_z(\boldsymbol{x}, \boldsymbol{z}) = \mathbb{E}_{\mathbb{P}}[z(\boldsymbol{x}, \cdot)z(\boldsymbol{z}, \cdot)]$ is a well-defined continuous function of $D \times D$. The covariance function is often modelled as correlation length: the larger its value, the more similarly nearby points of D are expected to be, and vice versa. Given the covariance function, one can define the covariance operator $C_z : L^2(D; \mathbb{R}) \to L^2(D; \mathbb{R})$ by

$$(C_z f)(\boldsymbol{x}) = \int_\Omega C_z(\boldsymbol{x}, \boldsymbol{z}) f(\boldsymbol{z}) \, d\boldsymbol{z} \,. \tag{3}$$

With the notation above, we can state the following Theorem.

Theorem 1 (Karhunen-Loève)**.** *Under the above assumptions on z, we have*

$$z(\boldsymbol{x}, \omega) = \sum_{n \in \mathbb{N}} \xi_n(\omega) \psi_n(\boldsymbol{x}) \,, \tag{4}$$

where $\{\psi_n\}_{n \in \mathbb{N}} \subseteq L^2(D; \mathbb{R})$ are orthonormal eigenfunctions of the covariance operator C_z, the corresponding eigenvalues $\{\lambda_n\}_{n \in \mathbb{N}}$ are non-negative and decreasingly ordered, the series $z(\boldsymbol{x}, \omega)$ converges in $L^2(\Omega, \mathbb{P}; \mathbb{R})$ and uniformly in D, with

$$\xi_n(\omega) = \int_D z(\boldsymbol{x}, \omega) \psi_n(\boldsymbol{x}) \, d\boldsymbol{x} \,. \tag{5}$$

Moreover, the random variables ξ_n are centered, uncorrelated and have variance λ_n:

$$\mathbb{E}_{\mathbb{P}}[\xi_n] = 0 \quad \text{and} \quad \mathbb{E}_{\mathbb{P}}[\xi_n \xi_m] = \lambda_n \delta_{nm} \quad \text{for any } n, m \in \mathbb{N} \,.$$

A proof can be found in [120]. The Karhunen-Loève (KL) expansion in (4) is optimal in the sense that the mean-square error of the truncation after any finite amount of terms K is minimal and equal to the next eigenvalue λ_{K+1} [79]. Note that there is an infinite number of random variables (5) in the series (4). Hence, the KL expansion directly shows that making inference over a random field can be though as an infinite dimensional inference task.

Given a covariance structure, the KL expansion is extremely useful to approximately sample from a Gaussian random field [2]. In fact, supposed that C_z is a self-adjoint, positive-definite, nuclear operator on a Hilbert space $\mathcal{H}$. Let $(\lambda_n, \psi_n)_{n \in \mathbb{N}}$ be an eigendecomposition of C_z ordered by decreasing eigenvalues λ_n, and take i.i.d. $\xi_n \sim \mathcal{N}(0, \lambda_n)$ for $n \in \mathbb{N}$. Then,

$$z(\boldsymbol{x}, \omega) = \sum_{n=1}^{\infty} \xi_n \psi_n \sim \mathcal{N}(0, C_z) \tag{6}$$

is a Gaussian random field. Because the eigenvalues λ_n are ordered in a decreasing manner, the importance of the eigendirections ψ_n decreases as n increases.

Hence, samples from a Gaussian random field can be approximately recovered by truncating the expansion in (6) after K terms, for K large enough. A fast eigenvalue decay will impose a low-rank structure in the probability space, hence the problem will be intrinsically low-dimensional and K can be small. On the other hand, a slow decay in the eigenvalues will necessitate a large truncation K to avoid large errors [79].

2.4 Quantity of interest

The output of the model itself may not be our main interest. Most often, we would like to calculate statistics of some quantity of interest which depends on the output of the model. Given the forward model defined via (2), for example we may want to consider $Q : \mathcal{S} \to \mathbb{R}$ to be the average pressure near some physical coordinate $\boldsymbol{x}_Q \in D$ defined by the linear functional

$$Q(u) = C \int_D \exp(-\|\boldsymbol{x} - \boldsymbol{x}_Q\|_2^2/\lambda_Q) u(\boldsymbol{x}, \omega)\, d\boldsymbol{x}\,, \tag{7}$$

where $C = \left(\int_D \exp(-\|\boldsymbol{x} - \boldsymbol{x}_Q\|_2^2/\lambda_Q)\, d\boldsymbol{x} \right)^{-1}$ is a normalizing constant and $\lambda_Q > 0$ is a length scale. Then, we may want to estimate $\mathbb{E}[Q]$, where the expectation is taken with respect to the probability measure imposed on the random field $z(\boldsymbol{x}, \omega)$.

2.5 Sample-based inference

We observe that, for a fixed a random realization $\omega \in \Omega$, equation (2) reduces to a deterministic PDE that can be solved numerically via Finite Element Methods (FEMs) [130], from which the quantity of interest can be in turn approximated. For sake of notation, let us denote by the subscript L a parametrized level of accuracy which takes into account of all the approximations involved. Thus, if we assume z to be a Gaussian random field with given covariance operator C_z, a possible way to approximate the expectation $\mathbb{E}[Q]$ consists in the following recipe: 1) produce i.i.d. approximate samples $z_L(\boldsymbol{x}, \omega^{(i)})$, for $i = 1, \dots, N$, via a suitable truncation of (6); 2) given $z_L(\boldsymbol{x}, \omega^{(i)})$, calculate an approximate solution $u_L^{(i)}$ to the elliptic PDE in (2) with a Finite Element Method; 3) given $u_L^{(i)}$, calculate an approximation $Q_L^{(i)}$ of the quantity of interest (7); 4) finally, form the Monte Carlo (or sample average) estimator [95]

$$\hat{Q}_L^{\mathrm{MC}} = \frac{1}{N} \sum_{i=1}^{N} Q_L^{(i)}\,. \tag{8}$$

Equation (8) is an unbiased estimator for $\mathbb{E}[Q_L]$. Furthermore, by the strong law of large numbers [49], $\hat{Q}_L^{\mathrm{MC}} \to \mathbb{E}[Q_L]$ a.s., i.e. the estimator is strongly consistent [76]. In addition, since $\mathrm{Var}(\hat{Q}_L^{\mathrm{MC}}) = \mathrm{Var}(Q_L)/N$, we have that $\hat{Q}_L \to \mathbb{E}[Q_L]$ in distribution, with rate of convergence $N^{-1/2}$ and constant exactly given by the standard deviation $\sqrt{\mathrm{Var}(Q_L)}$. It is important to observe that the rate of convergence does not depend on the dimensionality of the problem, which makes Monte Carlo a sensible inference methodology for infinite-dimensional problems, like the case that we are considering.

One can write a bias-variance decomposition [54] for the mean squared error of $\hat{Q}_L^{\mathrm{MC}}$:

$$\mathrm{MSE}(\hat{Q}_L^{\mathrm{MC}}) = \mathbb{E}[(\hat{Q}_L^{\mathrm{MC}} - \mathbb{E}[Q])^2] = (\mathbb{E}[Q_L - Q])^2 + \mathrm{Var}(Q_L)/N\,. \tag{9}$$

The first term on the right-hand-side of (9) is a squared-bias, which depends on the truncation threshold of the KL expansion in (6) and on the numerical approximations in (2) and (7). As the computation of each sample involves the solution of the PDE (2) plus further calculations, there is a clear trade-off between accuracy L and number of samples N to respectively reduce bias and variance terms. This observations forms the basis for Multilevel Monte Carlo (MLMC) [58], which we will introduce in Section 2.6 as a Monte Carlo method that exploits variance reduction between a sequence of estimators computed at different accuracies $\ell = 0, 1, \ldots, L$ to drastically reduce the overall computational cost.

Alongside with MLMC, Quasi-Monte Carlo (QMC) methods [101] constitute an important contribution that, under certain regularity conditions, allow to accelerate Monte Carlo from a rate of $N^{-1/2}$ to $N^{-1+\delta}$, for $\delta > 0$ arbitrarily small. The main idea is to consider a set of samples that is not random like in Monte Carlo, but that is deterministically constructed to reduce the quadrature error. Such sets can be achieved via tensor-product grids, lattice rules, cubature rules and digital nets. [61, 86] show how MLMC and QMC can be combined to achieve even better performance.

Monte Carlo-type approaches are not the only possibility. Stochastic Galerkin methods [55, 10], (generalized) polynomial chaos expansions [129] and stochastic collocation [128, 9] are examples of inference methodologies that approximate the solution of a random PDE as an expansion in two sets of basis functions, one for the physical space via standard numerical methods (e.g. FEMs) and one for the probability domain via polynomial approximations. The convergence rates of these methods can result much faster then Monte Carlo methods for small and medium size problems, as, unlike Monte Carlo, they can exploit smoothness properties of the parameter space. However, their rate of convergence typically deteriorates as the dimensionality of the problem increases. Thus, for very high-dimensional problems, Monte Carlo methods are again the most common choice.

2.6 Multilevel Monte Carlo

Multilevel Monte Carlo [58, 57] is a control variate technique [100] which exploits model hierarchies to achieve variance reduction and reduce the overall computation cost of a Monte Carlo estimator. In our model problems (2), like in many other cases, hierarchies can be exploited either in the numerical discretization of the PDE, in the truncation of the KL expansion of the random field, or in the approximation of the quantity of interest. Whereas in Monte Carlo we denoted by L a unique accuracy level capturing all of those approximations, MLMC considers a sequence of levels $\ell = 0, \dots, L$, from coarsest to finest.[1] The main idea here is that, as a coarser level is typically also cheaper, one can draw many cheap coarse samples to achieve a small variance, then progressively refine the estimator with less and less samples as the level increases. Theorem 2 shows how, under some assumptions on the decay of bias and variance and on the growth of cost over the levels, the multilevel procedure can reduce the computational cost of a Monte Carlo estimator by orders of magnitude.

Theorem 2. *Let $\varepsilon < e^{-1}$. Denote by $(Q_\ell)_{\ell=0}^L$ a sequence of approximations of a quantity of interest Q for a correspondent set of level accuracies $\ell = 0, \dots, L$, such that $Q_\ell \to Q$ for $\ell \to \infty$. Given a sequence of number of samples $(N_\ell)_{\ell=0}^L$, denote the estimators*

$$\hat{Y}_0 = \frac{1}{N_0}\sum_{k=1}^{N_0} Q_0^{(k)} \quad \text{and} \quad \hat{Y}_\ell = \frac{1}{N_\ell}\sum_{k=1}^{N_\ell}\left(Q_\ell^{(k)} - Q_{\ell-1}^{(k)}\right) \quad \text{for } \ell = 1, \dots, L\,, \tag{10}$$

where $Q_\ell^{(k)}$ and $Q_{\ell-1}^{(k)}$ are positively correlated, whereas both $Q_0^{(k)}$ and the differences $Q_\ell^{(k)} - Q_{\ell-1}^{(k)}$ are i.i.d. Denote the MLMC estimator by

$$\hat{Q}_L^{\mathrm{ML}} = \sum_{\ell=0}^{L} \hat{Y}_\ell\,. \tag{11}$$

Suppose there exist constants $\alpha, \beta, \gamma > 0$, with $\alpha \geq \min(\beta, \gamma)$, that satisfy assumptions

A1 $|\mathbb{E}[Q_\ell - Q]| = \mathcal{O}(2^{-\alpha\ell})\,,$

A2 $\mathrm{Var}(Q_\ell) = \mathcal{O}(2^{-\beta\ell})\,,$

A3 $\mathrm{Cost}(Q_\ell) = \mathcal{O}(2^{\gamma\ell})\,.$

Then, there exists L and $(N_\ell)_{\ell=0}^L$ such that, denoting by MSE the mean squared error, we have

$$MSE = \mathbb{E}[(\hat{Q}_L^{\mathrm{L}} - \mathbb{E}[Q])^2] < \varepsilon^2 \;\&\; \mathrm{Cost}(\hat{Q}_L^{\mathrm{ML}}) = \mathcal{O}\left(\varepsilon^{-2-\max(0,\frac{\gamma-\beta}{\alpha})}\right)(\log\varepsilon)^{2\delta_{\beta\gamma}} \tag{12}$$

where $\delta_{\beta\gamma}$ denotes a Kronecker delta.

For a slightly more general version of Theorem 2, together with its proof, see [121]. The complexity Theorem 2 shows that, under assumptions **A1-3**, the computational cost of the multilevel estimator in (11) is given in (12). Note

[1] If we assign to each of the three ways to generate hierarchies a different accuracy level parameter, i.e. $\ell_i = 0, \dots, L_i$, it is possible to achieve even better variance reduction in certain situations. This is the main idea behind Multi-index Monte Carlo [69].

that, for the same tolerance ε^2 on the MSE, the cost of a simple Monte Carlo estimator as in (8) is given by

$$\mathrm{Cost}(\hat{Q}_L^{\mathrm{MC}}) = \mathcal{O}\left(\varepsilon^{-2-\frac{\gamma}{\alpha}}\right) ,$$

whence the MLMC estimator is asymptotically always better, for any configuration of α, β and γ. Particularly in the case $\beta > \gamma$, that is when the variance decays faster than the cost increases, we have the best scenario $\mathrm{Cost}(\hat{Q}_L^{\mathrm{ML}}) = \mathcal{O}\left(\varepsilon^{-2}\right)$. For several practical cases where $\frac{\gamma}{\alpha}$ equals 1 or 2, this implies order of magnitude gains in the computational cost.

Multilevel techniques have been generalized to a wide range of algorithms and applications in forward Uncertainty Quantification [31, 13, 27, 1], Bayesian inverse problems [97, 44, 45, 116], filtering [72, 18, 65], finance [59, 60], rare event simulation [107, 105, 106], stochastic reaction networks [5, 6, 89] and many others. We refer to [56] for an up-to-date review of Multilevel Monte Carlo literature.

In Paper I (cf. [40]), we generalize MLMC to Continuous Level Monte Carlo (CLMC), a framework where the level parameter is not necessarily fixed to a discrete sequence, but it can range continuously from coarsest to finest. Beside the theoretical interest in the method, CLMC is practically relevant for adaptive FEM applications [11] where the adaptive scheme is taken to be sample-dependent. Here, CLMC allows to flexibly pick the level parameter according to the estimated adaptive error given by each realization of the model. This is in contrast to more standard adaptive MLMC algorithms [84], where a discrete sequence of levels has to be chosen a priori, which may incur in extra inference errors. Another important contribution of the paper consists into showing that the CLMC estimator can be made unbiased with respect to the true quantity of interest, which directly extends the work in [112]. By taking the finest level parameter L to be a random variable, we prove under which conditions unbiasedness can be achieved within a finite computational cost.

In Paper II (cf. [34]), we further develop Multilevel MCMC [44, 45] by combining it with DILI MCMC [35] and providing a hierarchical way to construct a low-rank subspace. See Section 4.1 for details.

3 Inverse Uncertainty Quantification

3.1 Motivational example

Opposite to forward UQ described in Section 2, inverse UQ attempts to recover information about inputs of a model given actual observations [19]. Again, let us formulate a toy example to motivate the problem. Borrowing notation from Section 2.1, we introduce a non-linear forward model $\mathcal{F} : \mathbb{R}^2 \to \mathbb{R}$ defined by $\mathcal{F}(\boldsymbol{\theta}) = \theta_1^3 + \theta_2$, for each $\boldsymbol{\theta} = [\theta_1, \theta_2] \in \mathbb{R}^2$. We assume there exists an unknown true parameter $\boldsymbol{\theta}_{\mathrm{true}}$ that, as an input of the forward model, produces a noisy observation $y = \mathcal{F}(\boldsymbol{\theta}_{\mathrm{true}}) + \sigma_y \xi$, where $\xi \sim \mathcal{N}(0, 1)$ can be interpreted as an additive measurement error, with a standard deviation $\sigma_y > 0$. It is immediate to see that $y|\boldsymbol{\theta}_{\text{true}} \sim \mathcal{L}(\cdot|\boldsymbol{\theta}_{\text{true}}) = \mathcal{N}(\mathcal{F}(\boldsymbol{\theta}_{\text{true}}), \sigma_y^2)$, where $\mathcal{L}$ is a *likelihood* function describing the probability of the observations given a parameter value. In this numerical example, we sample $\boldsymbol{\theta}_{\text{true}} \sim \mathcal{N}(0, I)$ and take $\sigma_y = 0.1$.

From a classic frequentist statistics perspective, one would like to recover the value of $\boldsymbol{\theta}_{\text{true}}$ generating the observation y by maximizing the likelihood function $\mathcal{L}(y|\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$. The corresponding estimator $\hat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}}\, \mathcal{L}(y|\boldsymbol{\theta})$ is called *maximum likelihood estimator* (MLE) [81]. Unfortunately, the solution to this problem is not unique: for any value y, there is an infinite amount of solutions, arbitrarily far in Euclidean distance, equally minimizing $\mathcal{L}(y|\boldsymbol{\theta})$. This issue is known as non-identifiability, which, together with sensitivity of the MLE solution to the measurement error, characterizes ill-posedness of inverse problems [50]. A classic way to remedy this problem is to introduce a regularization term [50], that is a penalization term in the objective function that encourages, more or less strongly according to a parameter $\lambda > 0$, the optimization towards particular values of the parameter space. A popular example is a Tikhonov regularization [102], a quadratic term added to the negative log-likelihood to form the objective function

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \left(-\log \mathcal{L}(y|\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2 \right) . \tag{13}$$

The regularized problem in (13) is well-posed, more stable and still easy to optimize, as the introduced regularization is quadratic. The value of the solutions, however, is shifted towards an arbitrary value, in this case the null vector $\mathbf{0}$.

Interestingly, the Bayesian formulation provides a principled justification, as well as a probabilistic interpretation, to the introduction of a regularization term. In fact, as we will see in Section 3.2, the regularization corresponds to the log-prior density, describing our a priori knowledge of the parameter values. In the common case (13), the regularization term exactly corresponds to a negative log-Gaussian prior of the form $N(\mathbf{0}, \frac{2}{\lambda}I)$, and the overall objective function can be seen as the negative log-posterior density.

This probabilistic interpretation allows to overcome the ill-posedness [118]. By introducing a joint probabilistic model over parameters and data, we can specify a solution for every possible noise realization, and attach to it a probability simultaneously taking into account the model and our belief. If two solutions are possible for a specific realization of the noise, there will not be a right and wrong one; they will simply be assigned the same probability.

Figure 2 displays contours of the resulting posterior density for $\lambda = 8$ (more details will be given in Section 3.2) and samples from it generated via a Markov Chain Monte Carlo (MCMC) algorithm (see Section 4.1 for details). We can see how the posterior uncertainty shape substantially differs from a Gaussian (ellipsoidal) shape, which is our prior. In many computationally intensive scenario, it is common to attempt to approximate the posterior distribution via a Gaussian, for example via Variational Inference [21] or Laplace approximation [117]. However, it this case this would either neglect the tails of the distribution, or assign large probability to unlikely regions of the parameters space, which may bias consequential decisions.

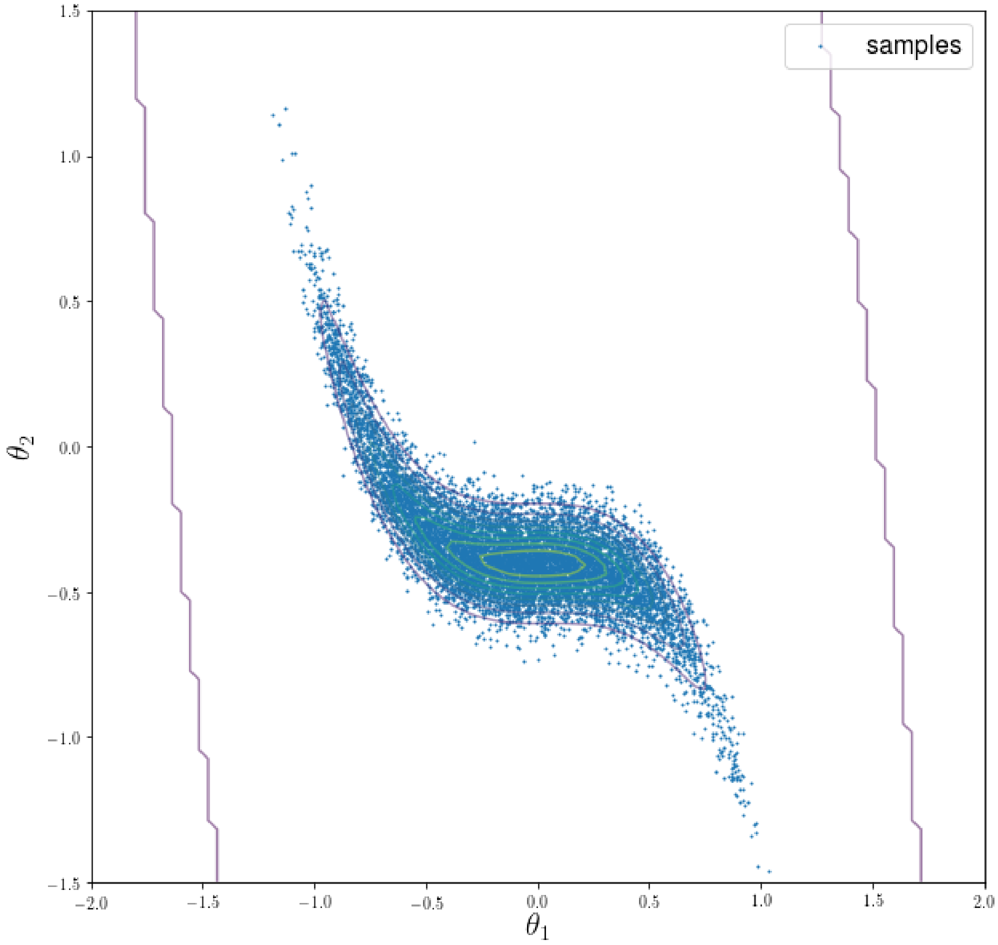


Figure 2: Contours and samples from the posterior density

3.2 Bayesian inverse problems

Inverse problems arise in natural sciences to study causal effect of observed phenomena. Mathematical models are usually complex and inspired from physics or engineering. In this Section, we will introduce a Bayesian formulation of inverse problems in infinite dimensions. We will take the elliptic PDE framework presented in Section 2.2 as a modelling example, but we will operate in a fundamentally different direction: whereas Section 2 was about fixing a random distribution for the covariance structure of the random field and analyze statistics of the quantity of interest under a pre-determined probability, here we will observe noisy local instances of the model solution and study statistics of the quantity of interest under the resulting posterior probability distribution.

Bayes' theorem. Let $(\Theta, \mathcal{A})$ and $(\mathcal{Y}, \mathcal{B})$ denote two measurable spaces and let ν_{post} and ν_0 be probability measures on the joint space $\Theta \times \mathcal{Y}$. We assume $\nu \ll \nu_0$, i.e. the measure ν is absolutely continuous [49] with respect to ν_0. Then, there exists a ν_0-measurable function $\phi : \Theta \times \mathcal{Y} \to \mathbb{R}$ such that

$$\frac{d\nu}{d\nu_0}(\theta, y) = \phi(\theta, y)\,, \quad \text{for each } (\theta, y) \in (\Theta, \mathcal{Y})\,. \tag{14}$$

In other words, ϕ is the *Radon-Nikodym derivative* of the measure ν with respect to the measure ν_0. Note that equation (14) is equivalent to

$$\mathbb{E}_\nu[f(\theta, y)] = \mathbb{E}_{\nu_0}[\phi(\theta, y) f(\theta, y)]\,,$$

for any function $f : \Theta \times \mathcal{Y} \to \mathbb{R}$ such that the expectations above are defined. The following Theorem provides existence of a conditional probability measure on Θ and an expression for its Radon-Nikodym derivative.

Theorem 3 (cf. [37])**.** *Let $y \in \mathcal{Y}$ be fixed and assume the conditional random variable $\theta|y$ exists under ν_0 with probability distribution denoted by $\nu_0^y(d\theta)$. Then the conditional random variable $\theta|y$ under ν exists, with probability distribution denoted by $\nu^y(d\theta)$. Furthermore, $\nu^y \ll \nu_0^y$ and if $C(y) = \int_\Theta \phi(\theta, y)\, d\nu^y(\theta)$ then*

$$\frac{d\nu^y}{d\nu_0^y}(\theta) = \frac{1}{C(y)}\phi(\theta, y)\,, \quad \textit{for each } \theta \in \Theta\,. \tag{15}$$

We refer to Theorem 3 as Bayes' theorem. The probability measure ν_0^y plays the role of prior distribution, whereas ν^y corresponds to the posterior distribution. The function $\phi(\theta, y)$ corresponds to the likelihood function $\mathcal{L}(y|\theta)$ that we heuristically introduced in Section 3.1.

We observe that Theorem 3 is both valid in a finite and an infinite dimensional framework. For such reason, it adds an important contribution to the literature, since it ensures that as the dimensionality of the problem grows to infinity, the prior and posterior probability measures are still well-posed. Theorem 4 provides a more classical version valid exclusively in finite dimensions, where the probability measures are expressed via their Radon-Nikodym derivative with respect to a Lebesgue measure. [73]

Theorem 4. *Let us assume that the probability measures ν^y and ν_0^y admit probability densities π and π_0, respectively. Then, equation (15) in Bayes' theorem reduces to*

$$\pi(\theta) = \frac{\mathcal{L}(y|\theta)\pi_0(\theta)}{C(y)}\,, \tag{16}$$

where $C(y) = \int_\Theta \mathcal{L}(y|\theta)\pi_0(\theta)\, d\theta$.

Unfortunately, in infinite dimensional Banach spaces it is not possible to formulate an analogue of Lebesgue measure [80], hence Theorem 4 does not hold in this framework. Yet, Theorem 4 is practically useful, because, computationally, infinite dimensional objects always have to be approximated into finite dimensional ones.

A classic example of Bayesian inverse problem. Consider $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ to be the forward model defined by (2) in Section 2. Let us assume the coefficient k is modeled via a random field characterized via the KL expansion (4). Let us denote by $\theta = (\xi_n)_{n\geq 1}$ the series of random parameters associated with such expansion.

As a possible application to the forward model $\mathcal{F}$, consider an incompressible subsurface flow, where $k(\boldsymbol{x}, \omega)$ and $u(\boldsymbol{x}, \omega)$ respectively denote the local permeability coefficient and the pressure at the physical coordinate $\boldsymbol{x} \in D$ for the random sample $\omega \in \Omega$. As we would like to make inference on the probability distribution of the coefficients θ characterizing the random field, we can drill into the ground and measure the pressure $u(\boldsymbol{x}_i, \omega)$ of the subsurface at a set of local coordinates $\boldsymbol{x}_i \in D$, for $i = 1, \dots, m$. We can mathematically describe the set of measurements $\boldsymbol{y} \in \mathbb{R}^m$ via an observation operator $\mathcal{O} : \Theta \to \mathbb{R}^m$, such that[2] $\mathcal{O}_i(\theta) = \mathcal{F}(k)(\boldsymbol{x}_i)$ and

$$\boldsymbol{y} = \mathcal{O}(\theta) + \sigma_y \boldsymbol{\xi}, \tag{17}$$

where $\boldsymbol{\xi} \sim \mathcal{N}(0, I_m)$ and $\sigma_y > 0$. The additive Gaussian noise $\boldsymbol{\xi}$ can be interpreted both as an unbiased measurement error, i.e. human or machine error committed during the observation process, as well as structural uncertainty, which embodies a lack of representation power of the model to describe the data [37]. It is immediate to check that, given the structure in (17), the likelihood has the form $\mathcal{L}(\cdot|\theta) = \mathcal{N}(\mathcal{O}(\theta), \sigma_y^2 I_m)$.

Let us assume *a priori* a Gaussian random field over k, i.e. the prior probability measure $\nu_0^y(d\theta)$ is an independent standard Gaussian distribution in each dimension. Then, given a prior measure and a likelihood model as in (17), equation (15) in Bayes' theorem 3 provides a rigorous way to integrate a quantity of interest Q with respect to the posterior measure ν^y, that is $\mathbb{E}_{\nu^y}[Q]$. In other words, we can provide statistics of the quantity of interest with respect to a probability distribution that takes into account both our subjective believe and data information.

We observe that, although $\mathbb{E}_{\nu^y}[Q]$ is theoretically well-defined for any appropriate quantity of interest Q, its calculation can be very challenging. In Section 4 describe some of the main computational issues and provides an overview of techniques that aim to surmount those problems.

4 A computational perspective

Although the Bayesian approach provides rich information about our estimates, it comes at a cost. Unlike in more classical frequentist statistics, where the task is usually to recover a point estimator that best represents the data - we have mentioned in Section 3.1 how this can be directly linked to recovering

[2] Please note that such definition of $\mathcal{O}(\theta)$ is well-posed only if point-evaluations of the solutions are defined in the Sobolev space $H_0^1(D)$. If they are not, we can observe the average pressure in a small ball around $\boldsymbol{x}_i$ and approximate the local measurement as the radius goes to zero [114].

the global maximum of the posterior distribution, also known as *Maximum-a-Posteriori* (MAP) [108] - in Bayesian statistics one would like to represents the whole posterior distribution in order to estimate the expectation $\mathbb{E}_{\nu^y}[Q]$ or other statistical properties of Q.

One of the main challenges is represented by the normalization constant $C(y)$. In high dimensions, this is given as a high-dimensional integral of the likelihood function under the prior probability measure. Since the rate of convergence of standard quadrature methods degenerates as the dimensionality increases, in very high dimensions the main available resource to estimate $C(y)$ are Monte Carlo methods, which, as we have seen in Section 2.5, have a dimension-independent rate of convergence. However, because of the slow rate of convergence, one might need a huge amount of samples to decrease the variance of the estimator to an acceptable threshold. In a scenario where either the forward model or the quantity of interest are very expensive to evaluate, like the case that we are considering, drawing many samples becomes quickly unfeasible. As shown in [116], when clear hierarchical structure is available, one way to alleviate this problematic is to combine MLMC with a self-normalized importance sampling produce [70], which allows to estimate $\mathbb{E}_{\nu^y}[Q]$, together with the normalization constant $C(y)$, with drastically reduced computational cost. However, if the normalization constant is very small, this method will still be hopeless. In general, it is a common choice to look for methodologies that allow to estimate $\mathbb{E}_{\nu^y}[Q]$ without estimating $C(y)$. In the following, we will give an overview of the most prominent methodologies, that will also be the main object of study in my Papers.

4.1 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is probably the most popular family of algorithms to produce samples in a Bayesian framework [62]. The idea is that, although we are not able to generate independent samples directly from the posterior distribution, we might be able to recover correlated samples asymptotically as states of an *ergodic* Markov Chain [62]. This way, one can estimate

$$\mathbb{E}_{\boldsymbol{\theta}\sim\nu^y}[Q(\boldsymbol{\theta})] \approx \frac{1}{N}\sum_{i=1} Q(\boldsymbol{\theta}^{(i)}) \,,$$

where $\{\boldsymbol{\theta}^{(i)}\}_{i=1}^N$ are N samples from a Markov chain in the stationary regime, whose stationary distribution is the posterior ν^y. Because samples from a Markov chain are generally correlated, the amount of information added by each sample is lower than in standard Monte Carlo, and a higher number of samples is required to reduce the variance of the estimator to the same amount. To be more precise, the Central Limit Theorem under weak dependence [23] states that the variance reduction factor that we lose via correlated samples is the *integrated autocorrelation time* (IACT) [62]

$$\tau = 1 + 2\sum_{k=1}^{\infty} \rho(k+1) \,, \tag{18}$$

where $\rho(\cdot)$ denotes the autocorrelation function of the Markov chain $\{\boldsymbol{\theta}^{(i)}\}_{i\geq 1}$. Note that $\tau \geq 1$, where the equality holds if and only if the samples are uncorrelated. Then, we define the *effective sample size* (ESS) as $N_{\text{eff}} = N/\tau$.

The Metropolis-Hastings algorithm. The question remains how to produce a Markov chain $\{\boldsymbol{\theta}^{(i)}\}_{i=\geq 1}$ that asymptotically converges in to ν^y in a distributional sense. Many different methods have been proposed in the literature over decades [64], but the most popular one is, without doubt, the Metropolis-Hastings algorithm [29]. In order to present it, we will assume that we are working in finite (but possibly very high) dimensions and that prior and posterior probability densities π_0 and π exist, respectively.

Algorithm 1 describes a generic step of the Metropolis-Hastings algorithm. Given the current state of the Markov chain $\boldsymbol{\theta}_k$, $k \in \mathbb{N}$, together with a conditional proposal probability density $q(\cdot|\cdot)$, a new sample $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}_k)$ is proposed as a candidate for the next state $\boldsymbol{\theta}_{k+1}$ and then accepted with some probability $\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}_k)$ or, otherwise, rejected. It can be shown [62] that the resulting Markov chain $\{\boldsymbol{\theta}_k\}_{k\geq 1}$ converges in distribution to π.

Algorithm 1: *k*-th iteration of Metropolis-Hastings algorithm

```
Input  : Current state $\boldsymbol{\theta}_k$; proposal density $q(\cdot|\cdot)$
Output : $\boldsymbol{\theta}_{k+1}$
1: Sample $\boldsymbol{\theta}' \sim q(\cdot|\boldsymbol{\theta}_k)$
2: Calculate acceptance probability $\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}_k) = \min\left(1, \frac{\pi(\boldsymbol{\theta}')q(\boldsymbol{\theta}_k|\boldsymbol{\theta}')}{\pi(\boldsymbol{\theta}_k)q(\boldsymbol{\theta}'|\boldsymbol{\theta}_k)}\right)$
3: Set $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}'$ with probability $\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}_k)$, otherwise set $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k$
```

It is important to notice that, in a Bayesian framework, the acceptance probability in Algorithm 1 can be rewritten via Bayes' theorem as

$$\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}_k) = \min\left(1, \frac{\mathcal{L}(\boldsymbol{y}|\boldsymbol{\theta}')\pi_0(\boldsymbol{\theta}')q(\boldsymbol{\theta}_k|\boldsymbol{\theta}')}{\mathcal{L}(\boldsymbol{y}|\boldsymbol{\theta}_k)\pi_0(\boldsymbol{\theta}_k)q(\boldsymbol{\theta}'|\boldsymbol{\theta}_k)}\right) .$$

Note that, because the normalizing constant is elided in the ratio, it never has to be estimated, which constitutes one of the main advantages of this algorithm.

The proposal density $q(\cdot|\cdot)$ can be any probability density with support containing the parameter space Θ. The proposal is largely responsible for the *mixing* time, i.e. convergence speed, of the Markov chain. A good proposal density will try to meet the following criteria. First, it wants to propose samples that are as uncorrelated as possible to the current state, in order to achieve low IACT and, consequently, high ESS. Second, it should propose samples that are likely to be accepted, since every rejection duplicates the current states therefore increasing the correlation within the chain. These two aspects often constitute a trade-off that a good proposal should balance between. For example, suppose that we propose $\boldsymbol{\theta}' = \boldsymbol{\theta}_k + \varepsilon\boldsymbol{\xi}$, where $\varepsilon > 0$ is a stepsize and $\boldsymbol{\xi} \sim \mathcal{N}(0, I)$ is a random perturbation. Equivalently, we have $q(\boldsymbol{\theta}'|\boldsymbol{\theta}_k) = \mathcal{N}(\boldsymbol{\theta}_k, \varepsilon^2 I)$, which is a classic *random walk* proposal density [62]. A small stepsize ε will propose samples $\boldsymbol{\theta}'$ that are very close to $\boldsymbol{\theta}_k$, therefore the acceptance probability $\alpha(\boldsymbol{\theta}'|\boldsymbol{\theta}_k)$ will be almost one; in other words, Algorithm 1 will accept almost always. However, the states will also be extremely correlated and the Markov chain will take a very high amount of time to explore the parameter space and reach its convergence regime. On the other hand, if ε is large, $\boldsymbol{\theta}'$ will be not very correlated to $\boldsymbol{\theta}_k$, but the rate of rejection will be much higher. One should tune the stepsize between these two regimes to achieve best performance. In [113], the authors show that the best stepsize for random walk proposal is around $\varepsilon = 0.23$, but this does not hold for other proposals.

Ideally, one would like to independently propose directly from π. In this case both the acceptance probability α and the IACT τ would equal one. Of course this is not possible, but it provides intuition that a good proposal density should mimic the target density in order to keep high acceptance probability and ESS. This is the purpose of Langevin-type proposal densities, where a proposal density is constructed as an approximation of a Langevin stochastic differential equation (SDE), with π as stationary distribution [119]. Examples of Langevin-type proposals are MALA MCMC [127] and pCN MCMC [33], where the latter does not use gradient information but targets the prior density rather than the posterior. An important feature of pCN MCMC is that it is dimension-independent, i.e. its acceptance rate does not depend on the dimensionality of the problem [33]. This is in contrast to the behaviour of more standard proposals like random walk, whose acceptance rate converges to zero as the dimensionality increases. Yet, in high-dimensional pCN can practically be very inefficient. In order to overcome this problem, one can look for data-informed low-rank structures that allows to appropriately rescale the proposal density differently along different dimensions. This is the purpose of DILI MCMC [35], where a *Likelihood-Informed Subspace* (LIS) [36] is employed together with a pCN (or MALA) procedure to obtain a proposal that is both dimension-independent and rescaled via data information.

Among other remarkable contributions to the literature of MCMC algorithms, we recall: Hamiltonian Monte Carlo (HMC) [99], where the parameter space is lifted via a *momentum* random variable, and the proposal combines random shifts of the momemtum together with the simulation of Hamitonian dynamics [88]; NUTS [74], which adaptively sets path lengths for HMC proposals; adaptive MCMC [67], where the proposal density is adaptively adjusted at an asymptotically vanishing rate; delayed-acceptance MCMC [30], where a preliminary acceptance step promotes or rejects proposed samples according to a cheap proxy of the likelihood; DRAM [68], which combines adaptive and delayed-acceptance MCMC. Among more recent promising approaches to formulate effective proposals, we refer to Randomize-Then-Optimize (RTO) [12, 125] and Tensor-Train approximations [47].

All the above-mentioned works aim to construct a proposal distribution that leads the MCMC algorithm to converge as quickly as possible. However, if the evaluation of the model at each step is very expensive, the algorithm will still be very slow. In [44, 45] (see also [71] for a different but related method), the authors integrate MCMC with multilevel methods to surmount this issue. The *Multilevel MCMC* (MLMCMC) algorithm take advantage of a telescoping sum like in standard MLMC, however every expectation here is estimated via an MCMC algorithm. Crucially, as in MLMC the samples in the differences must be positively correlated to achieve variance reduction, here the Markov chains in the differences are also constructed to be positively correlated. A similar complexity theorem to the one for MLMC is derived, achieving a drastic reduction in the overall computation cost. In Paper II (cf. [34]), we derive a coupling proposal mechanism to integrate MLMCMC [44, 45] with DILI MCMC in a non-trivial way, substantially speeding up the convergence of the Markov chain at each level. Furthermore, since the construction of the LIS can be very expensive at fine resolutions, we derive a hierarchical construction of the LIS to drastically reduce both offline and online computational costs of the method.

Other Monte Carlo algorithms & filtering applications. MCMC, and in particular the Metropolis-Hastings algorithms, are not the only possible Monte Carlo methods for sampling in a Bayesian framework. Sequential Monte Carlo is another popular choice [48], which is a population-based method based on *importance sampling* (IS) [98], where a set of particles is sequentially (i) updated according to some proposal probability density and (ii) reweighted according to the ratio of the proposal density and the (unnormalized) posterior. The reweighting procedure is important to reduce the variance of the quantity of interest estimator, however it can induce a degeneracy of particles when the importance density does not well match the posterior, which is particularly cumbersome in high-dimensions [38]. In [17], the authors show how this can be avoided by introducing an artificial sequence of target densities. Again, if hierarchical structures are available, this could be done via multilevel methods [18]. In [7], the authors show that another way to improve on this issue is by using SMC to propose efficient high-dimensional distributions for MCMC.

SMC has been particularly useful in *sequential Bayesian inference* [115] (also known as *filtering*), that is a temporal hidden Markov model (HMM) framework [22] where we would like to make inference whenever a new data point is observed. In this setting, examples of alternative methods to SMC are *Ensemble Kalman Filter* (EnKF) [52], *Ensemble Transform Particle Filter* (ETPF) [110] and their respective multilevel versions [72] and [65], both popular in data assimilation applications [51, 77]; *Thompson sampling* [25], widely used in bandit problems [8, 4]. In Paper IV (cf. [42]), we apply Stein variational Newton to changepoint detection [3], which can again be considered as a form of sequential Bayesian inference, although the problem is not necessarily a HMM. In Paper V (cf. [85]) we develop a Hierarchical Invertible Neural Transport (HINT) algorithm that we believe to be particularly suitable for sequential Bayesian inverse, since the method never requires to evaluate the prior analytically. See Section 4.2 for a discussion of Papers IV and V, relating them to the existing literature.

4.2 Variational inference and transport maps

"In theory, there is no difference between theory and practice. In practice, there is."

- Attributed to multiple people -

Although in theory MCMC is *asymptotically* guaranteed to weakly converge to the correct target distribution, in practice this might never happen in a reasonable amount of time. In particular either in high-dimensions, where the parameter space exploration is particularly challenging, or when the likelihood is computationally expensive to evaluate (because the forward model is itself expensive, or because of a huge amount of data), in practical applications the correctness of MCMC is set aside in favour of approximate but faster and more scalable methods. In the following, we will give an overview of optimization-based methods for sampling, such as Variational Inference and, more broadly, transport maps.

Suppose that, in a finite dimensional setting, we want to sample from a target probability density π, e.g. the posterior. Given a treatable reference density p, one might seek an invertible transformation $\boldsymbol{T} : \Theta \to \Theta$ such that, if $\boldsymbol{\theta} \sim p$, then $\boldsymbol{T}(\boldsymbol{\theta}) \sim \pi$. In other words, $\pi = \boldsymbol{T}_{\#}p$, where $\boldsymbol{T}_{\#}$ denotes the pushforward map via $\boldsymbol{T}$ and is defined by

$$\boldsymbol{T}_{\#}p(\boldsymbol{\theta}) = p(\boldsymbol{T}^{-1}(\boldsymbol{\theta}))|\det \nabla \boldsymbol{T}^{-1}(\boldsymbol{\theta})| \,. \tag{19}$$

We address diffeomorphic transformations $\boldsymbol{T}$ that map from a space to itself as *transport maps*. As a didactic example, consider the case where both p and π are Gaussian densities, namely $p = \mathcal{N}(\boldsymbol{\mu}_p, C_p)$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_\pi, C_\pi)$. Then, we can exactly push forward p to π via the linear transport map

$$\boldsymbol{T}(\boldsymbol{\theta}) = \boldsymbol{\mu}_\pi + C_\pi^{1/2} C_p^{-1/2}(\boldsymbol{\theta} - \boldsymbol{\mu}_p) \,.$$

In general, however, there is an infinite number of maps $\boldsymbol{T}$ that can do the job, possibly highly non-linear and complex. Hence, one would like to define a class of transport maps that is rich enough to contain, or to approximate, an exact pushforward operator and, at the same time, easy to define and optimize.

In order to assess how much reference and target density diverge from each other, it is common practical choice in variational inference to introduce a Kullback-Leibler (KL) divergence $\mathcal{D}_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ between the pushforward map $\boldsymbol{T}_{\#}p$ and the target π, that is

$$\mathcal{D}_{\mathrm{KL}}(\boldsymbol{T}_{\#}p \,\|\, \pi) = \mathbb{E}_{\boldsymbol{\theta} \sim \boldsymbol{T}_{\#}p}[\log \boldsymbol{T}_{\#}p(\boldsymbol{\theta}) - \log \pi(\boldsymbol{\theta})] \,. \tag{20}$$

Variational inference. Rather than constructing an explicit transport map $\boldsymbol{T}$, it is clear from (20) that one could directly parametrize the pushforward density $\boldsymbol{T}_{\#}p$ as a treatable *variational density* p_ϕ, and optimize over its parameters $\boldsymbol{\phi}$ [20, 21]. In a Bayesian framework, Bayes' theorem 4 shows that the normalization constant $C(y)$, also called *evidence*, appears in (20) as an additive term, since the posterior π comes into play only in logarithmic form. In other words, rather than minimizing (20), one can equivalently maximize the *Evidence Lower BOund* (ELBO) [21]

$$\underset{\phi}{\operatorname{argmax}}\, \mathrm{ELBO} = \underset{\phi}{\operatorname{argmax}} -\mathbb{E}_{\boldsymbol{\theta} \sim p_\phi}[\log p_\phi(\boldsymbol{\theta}) - \log \pi_0(\boldsymbol{\theta}) - \log \mathcal{L}(y|\boldsymbol{\theta})] \,, \tag{21}$$

which owes its name to the fact that

$$\log C(\boldsymbol{y}) = \mathrm{ELBO} - \mathcal{D}_{\mathrm{KL}}(p_\phi \,\|\, \pi) \leq \mathrm{ELBO}\,,$$

since $\mathcal{D}_{\mathrm{KL}}(p_\phi \,\|\, \pi) \geq 0$ by Jensen inequality (with equality if and only if $p_\phi = \pi$).

A typical choice for p_ϕ is to take a Gaussian density [123], or a mixure of Gaussian densities [75], with learnable mean and diagonal covariance matrix. In the Gaussian density case, this would correspond to $p_\phi = \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\phi}), D(\boldsymbol{\phi}))$, with diagonal D. The mean and covariance functions $\boldsymbol{\mu}(\boldsymbol{\phi})$ and $D(\boldsymbol{\phi})$ can be any differentiable functions of the parameter $\boldsymbol{\phi}$, which include the simplest case of an identity function as well as complex neural network [87, 63]. The restriction to diagonal matrices is often called *mean-field approximation* [21] and it corresponds to the often unrealistic assumption that the parameters are uncorrelated. Although this might lead to bad quantifications of the uncertainty of the posterior distribution, it also makes the optimization much easier and faster, which in many practical situations is considered as an acceptable trade-off.

It is worth to observe that, if several posterior densities have to be recovered for different instances of the observation vector $\boldsymbol{y}$ (think, for example, about the posterior probability of a disease for observations coming form different patiences), one might want to "amortize" the cost of re-training for each different observation by incorporating $\boldsymbol{y}$ as an input of the variational density p_ϕ, e.g. $p_\phi = \mathcal{N}(\boldsymbol{m}(\boldsymbol{\phi}, \boldsymbol{y}),$ $\boldsymbol{D}(\boldsymbol{\phi}, \boldsymbol{y}))$. By averaging over different instances of $\boldsymbol{y}$, one can then optimize over $\boldsymbol{\phi}$ to get parameters that take into account all the different observations. Most likely, the recovered variational density will be less precise in approximating the posterior for each single observation $\boldsymbol{y}$, but, if we accept the further layer of approximation, it will not need retraining. This methodology is usually referred to as amortized Variational Inference, and Variational Autoencoders (VAE) are a representative example [46].

Stein variational inference. It is well-known that

$$\mathcal{D}_{\mathrm{KL}}(\boldsymbol{T}_\# p \,\|\, \pi) = \mathcal{D}_{\mathrm{KL}}(p \,\|\, \boldsymbol{T}^\# \pi)\,,$$

where $\boldsymbol{T}^\#$ is called *pull-back operator* and is defined by $\boldsymbol{T}^\# = (\boldsymbol{T}^{-1})_\#$. In words, given an invertible transformation $\boldsymbol{T}$, in KL divergence it is equivalent to push-forward the reference density towards the target or to pull-back the target towards the reference. Then, we have

$$\mathcal{D}_{\mathrm{KL}}(p \,\|\, \boldsymbol{T}^\# \pi) = \mathbb{E}_{\boldsymbol{\theta}\sim p}[\log p(\boldsymbol{\theta}) - \log \pi(\boldsymbol{T}(\boldsymbol{\theta})) - \log|\det \nabla \boldsymbol{T}(\boldsymbol{\theta})|]\,. \tag{22}$$

Rather than equation (20), we can use (22) as an objective function to construct a map $\boldsymbol{T}$ that minimizes the divergence between p and $\boldsymbol{T}^\#\pi$. Note that, while in standard variational inference the representation power was restricted by the flexibility of the parametrized density p_θ, here the limitation is defined by the class of transport maps. However, it is much easier to design a flexible approximation function rather than a flexible (and treatable) variational density. The main difficulty of the objective function in (22) is the evaluation of the log-determinant term of the Jacobian of $\boldsymbol{T}$, which, for general transport maps, has a cubic computational cost. Thus, the challenge is to come up with transport maps $\boldsymbol{T}$ that are invertible and such that the functional gradient of (22) can be calculated at low cost.

One possibility is to take the map $\boldsymbol{T}$ to be a composition of small perturbations of the identity function, i.e. $\boldsymbol{T} = \boldsymbol{T}_L \circ \cdots \boldsymbol{T}_1$ where L is a maximum number of iterations and $\boldsymbol{T}_\ell(\boldsymbol{\theta}) = \boldsymbol{\theta} + \varepsilon \boldsymbol{Q}_\ell(\boldsymbol{\theta})$, with $\varepsilon > 0$ and $\boldsymbol{Q}_\ell : \Theta \to \Theta$ being another transport map, for $\ell = 1, \dots, L$. Note that if $\varepsilon > 0$ is small enough, $\boldsymbol{T}$ is invertible. In *Stein variational inference* (SVI) [91], $\boldsymbol{Q}_\ell$ belongs to a Reproducing Kernel Hilbert Space (RKHS) [15] $\mathcal{H}$, for some characterizing kernel $k : \mathcal{H} \times \mathcal{H} \to [0, 1]$. Stein variational gradient descent (SVGD) [91] takes the search direction $\boldsymbol{Q}_\ell$ to be the negative functional gradient direction, which can be explicitly recovered as

$$\boldsymbol{Q}_\ell(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\phi} \sim p_\ell}[\nabla \log \pi(\boldsymbol{\phi}) k(\boldsymbol{\phi}, \boldsymbol{\theta}) + \nabla_{\boldsymbol{\phi}} k(\boldsymbol{\phi}, \boldsymbol{\theta})] \,, \tag{23}$$

where p_ℓ is the current reference density at iteration ℓ. If we initially take the reference density to be the empirical approximation for a given set of particles of some treatable density (e.g. a Gaussian), we can sequentially transport these particles via a functional gradient descent optimization [102]. In [93], the authors show that the scaling limit solution of the SVGD dynamical system for the number of particles going to infinity is a non-linear PDE, whose solution exists and is unique. In [90], the authors show that if the initial empirical reference measure converge to the theoretical one when the number of particles grows to infinity, then at any step ℓ the empirical measure resulting from the SVGD iteration also converges to the theoretical one. Under several assumptions, they also show that the theoretical measure resulting from the SVGD iteration converges to the true posterior.

In Paper III (cf. [41]), we introduce *Stein Variational Newton* (SVN), where we derive second-order information that we use both to speed-up the optimization via a Newton-like procedure, and to change the distance metric in the kernel in order to improve particle spreading and scalability to higher dimensions. In Paper IV (cf. [42]), we study the application of SVN to changepoint detection, that is the detection of abrupt changes in the hidden model parameters underlying a sequence of observations. Follow-up work of Paper III has also been conducted by other research groups: in [28], the authors study the projection of SVN to a likelihood-informed subspace (LIS) [36], which is important for really high dimensions and uses similar techniques to the ones in Paper II (cf. [34]); in [124], the authors propose a matrix-valued kernel Stein variational method, where they theoretically and practically compare SVN to their algorithm.

Normalizing flow. While SVI is a non-parametric transport map approach, its parametric counterpart is often called *Normalizing Flow* [111]. Here, $\boldsymbol{T}$ is given some specific parametric structure to make the map invertible and the determinant term easy to calculate. A common choice is to pick a Knothe-

Rosenblatt rearrangement [94], that is

$$\boldsymbol{T}(\boldsymbol{\theta}) = \begin{bmatrix} T_1(\theta_1) \\ T_2(\theta_1, \theta_2) \\ \vdots \\ T_d(\theta_1, \dots, \theta_d)\,, \end{bmatrix} \tag{24}$$

such that $\frac{dT_j(\theta_1,\dots,\theta_j)}{d\theta_j} \neq 0$. Then, $\boldsymbol{T}$ is invertible and its log-determinant is simply given by the trace of its Jacobian. In [94], the components of the map $\boldsymbol{T}$ are chosen to be parametric polynomials. Another possible choice is to take regularized tensor-train approximations of the forward operator [47]. In (neural) *autoregressive flow* [83], each component is a neural network. In [78], the authors show that a specific type of neural autoregressive flows can be proven to be universal probability density approximators up to a desired tolerance.

A vast literature has been produced in the last few years about different neural network architectures that can increase the representation power of the transport map. An example is *bipartite flow* [43, 104, 82], where the set of coordinates is split into two blocks to build an autoregressive-type map whose inverse can be recovered analytically. The algorithm alternates between a sequence of orthogonal transformations (introduced in Sylvester Normalizing Flow [14]), which have the important role of reshuffling the coordinates, and triangular maps, whose goal is to capture complex non-linearities. However, in order to make sure that the overall map is analytically invertible, each triangular map is very sparse. Part of the goal of our Paper V (cf. [85]) is to explore a densification of the bipartite architecture via a hierarchical procedure, which is able to increase the representation power of each triangular map while maintaining analytical invertibility. Another work that attempts to obtain a dense architecture is Block Neural Autoregressive Flow (B-NAF) [39], but since the inverse map does not have an explicit analytical expression, the method cannot be used for sampling and is limited to density estimation.

5 Future directions

In this Section, we briefly describe some research directions that could be explored in the future.

Paper I. One key assumption of the paper is that the finest level of accuracy, which is randomly picked according to an exponential distribution, has to be statistically independent from the model random realization. We acknowledge and thank Dr. Tony Shardlow for pointing out that this assumption can be relaxed by using the Optimal Stopping Theorem from the theory of Martingales [66]. Further developments of the paper may study the application of continuous level hierarchies in Multi-index Monte Carlo [69] and Multilevel MCMC [44, 45].

Paper II. In this paper, we assume that the random field can be expressed via a KL expansion, which in practice needs to be truncated after a finite number of terms (see Section 2.3). In order to avoid this problematic, and in general to consider scenarios in which the new parameters at the new level are not statistically orthogonal to the previous ones under the prior measure, we would like to develop a Multilevel DILI MCMC algorithm that can directly work over the finite element mesh of the physical domain. Furthermore, we are interested in the development of a Multi-index DILI MCMC algorithm.

Papers III and IV. The performance of the SVN algorithm depends on the kernel in use. In particular, scalability can be problematic, because, in high-dimensions, kernel-based methods are known to struggle to distinguish between particle distances [53]. Thus, a possible future development of the SVN algorithm is to study the application of non-stationary kernels, where the distance can be locally rescaled according to the posterior geometry at the input locations themselves.

Paper V. This work is currently under further development for final submission. In particular, we are working on numerical results which study beneficial effects versus costs of the proposed hierarchical structure, providing comparisons to state-of-the-art Normalizing Flow algorithms.

6 References

[1] Assyr Abdulle, Andrea Barth, and Christoph Schwab. “Multilevel Monte Carlo methods for stochastic elliptic multiscale PDEs”. In: *Multiscale Modeling & Simulation* 11.4 (2013), pp. 1033–1070.

[2] Petter Abrahamsen. “A review of Gaussian random fields and correlation functions”. In: *Norsk Regnesentral/Norwegian Computing Center Oslo* (1997).

[3] Ryan Prescott Adams and David JC MacKay. “Bayesian online changepoint detection”. In: *Technical Report at University of Cambridge, arXiv preprint: 0710.3742* (2007).

[4] Shipra Agrawal and Navin Goyal. “Analysis of thompson sampling for the multi-armed bandit problem”. In: *Conference on Learning Theory*. 2012, pp. 39–1.

[5] David F Anderson and Desmond J Higham. “Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics”. In: *Multiscale Modeling & Simulation* 10.1 (2012), pp. 146–179.

[6] David F Anderson, Desmond J Higham, and Yu Sun. “Complexity of multilevel Monte Carlo tau-leaping”. In: *SIAM Journal on Numerical Analysis* 52.6 (2014), pp. 3106–3127.

[7] Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. “Particle Markov chain Monte Carlo methods”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3 (2010), pp. 269–342.

[8] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. “Finite-time analysis of the multiarmed bandit problem”. In: *Machine learning* 47.2-3 (2002), pp. 235–256.

[9] Ivo Babuška, Fabio Nobile, and Raul Tempone. “A stochastic collocation method for elliptic partial differential equations with random input data”. In: *SIAM Journal on Numerical Analysis* 45.3 (2007), pp. 1005–1034.

[10] Ivo Babuska, Raúl Tempone, and Georgios E Zouraris. “Galerkin finite element approximations of stochastic elliptic partial differential equations”. In: *SIAM Journal on Numerical Analysis* 42.2 (2004), pp. 800–825.

[11] I Babuvška and Werner C Rheinboldt. “Error estimates for adaptive finite element computations”. In: *SIAM Journal on Numerical Analysis* 15.4 (1978), pp. 736–754.

[12] Johnathan M Bardsley, Antti Solonen, Heikki Haario, and Marko Laine. “Randomize-then-optimize: A method for sampling from posterior distributions in nonlinear inverse problems”. In: *SIAM Journal on Scientific Computing* 36.4 (2014), A1895–A1910.

[13] Andrea Barth, Christoph Schwab, and Nathaniel Zollinger. “Multi-level Monte Carlo finite element method for elliptic PDEs with stochastic coefficients”. In: *Numerische Mathematik* 119.1 (2011), pp. 123–161.

[14] Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. “Sylvester normalizing flows for variational inference”. In: *arXiv preprint arXiv:1803.05649* (2018).

[15] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics.* Springer Science & Business Media, 2011.

[16] Alan A Berryman. “The orgins and evolution of predator-prey theory”. In: *Ecology* 73.5 (1992), pp. 1530–1535.

[17] Alexandros Beskos, Dan Crisan, and Ajay Jasra. “On the stability of sequential Monte Carlo methods in high dimensions”. In: *The Annals of Applied Probability* 24.4 (2014), pp. 1396–1445.

[18] Alexandros Beskos, Ajay Jasra, Kody Law, Raul Tempone, and Yan Zhou. “Multilevel sequential Monte Carlo samplers”. In: *Stochastic Processes and their Applications* 127.5 (2017), pp. 1417–1440.

[19] Lorenz Biegler, George Biros, Omar Ghattas, Matthias Heinkenschloss, David Keyes, Bani Mallick, Luis Tenorio, Bart Van Bloemen Waanders, Karen Willcox, and Youssef Marzouk. *Large-scale inverse problems and quantification of uncertainty.* Vol. 712. Wiley Online Library, 2011.

[20] David M Blei and Michael I Jordan. “Variational inference for Dirichlet process mixtures”. In: *Bayesian analysis* 1.1 (2006), pp. 121–143.

[21] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational inference: A review for statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), pp. 859–877.

[22] Phil Blunsom. “Hidden Markov models”. In: *Lecture notes, August* 15.18-19 (2004), p. 48.

[23] Richard C Bradley Jr. “Central limit theorems under weak dependence”. In: *Journal of Multivariate Analysis* 11.1 (1981), pp. 1–16.

[24] Somchart Chantasiriwan. “Solutions of partial differential equations with random Dirichlet boundary conditions by multiquadric collocation method”. In: *Engineering analysis with boundary elements* 29.12 (2005), pp. 1124–1129.

[25] Olivier Chapelle and Lihong Li. “An empirical evaluation of Thompson sampling”. In: *Advances in neural information processing systems.* 2011, pp. 2249–2257.

[26] Julia Charrier. “Strong and weak error estimates for elliptic partial differential equations with random coefficients”. In: *SIAM Journal on numerical analysis* 50.1 (2012), pp. 216–246.

[27] Julia Charrier, Robert Scheichl, and Aretha L Teckentrup. “Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods”. In: *SIAM Journal on Numerical Analysis* 51.1 (2013), pp. 322–352.

[28] Peng Chen, Keyi Wu, Joshua Chen, Thomas O’Leary-Roseberry, and Omar Ghattas. “Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions”. In: *Advances in neural information processing systems* (2019).

[29] Siddhartha Chib and Edward Greenberg. “Understanding the Metropolis-Hastings algorithm”. In: *The american statistician* 49.4 (1995), pp. 327–335.

[30] J Andrés Christen and Colin Fox. “Markov chain Monte Carlo using an approximation”. In: *Journal of Computational and Graphical statistics* 14.4 (2005), pp. 795–810.

[31] K Andrew Cliffe, Mike B Giles, Robert Scheichl, and Aretha L Teckentrup. “Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients”. In: *Computing and Visualization in Science* 14.1 (2011), p. 3.

[32] Earl A Coddington and Norman Levinson. *Theory of ordinary differential equations.* Tata McGraw-Hill Education, 1955.

[33] Simon L Cotter, Gareth O Roberts, Andrew M Stuart, and David White. “MCMC methods for functions: modifying old algorithms to make them faster”. In: *Statistical Science* (2013), pp. 424–446.

[34] Tiangang Cui, Gianluca Detommaso, and Robert Scheichl. “Multilevel dimension-independent likelihood-informed MCMC for large-scale inverse problems”. In: *arXiv preprint arXiv:1910.12431* (2019).

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

[36] Tiangang Cui, James Martin, Youssef M Marzouk, Antti Solonen, and Alessio Spantini. “Likelihood-informed dimension reduction for nonlinear inverse problems”. In: *Inverse Problems* 30.11 (2014), p. 114015.

[37] Masoumeh Dashti and Andrew M Stuart. “The Bayesian approach to inverse problems”. In: *Handbook of Uncertainty Quantification* (2016), pp. 1–118.

[38] Fred Daum and Jim Huang. “Particle degeneracy: root cause and solution”. In: *Signal Processing, Sensor Fusion, and Target Recognition XX*. Vol. 8050. International Society for Optics and Photonics. 2011, 80500W.

[39] Nicola De Cao, Ivan Titov, and Wilker Aziz. “Block neural autoregressive flow”. In: *arXiv preprint arXiv:1904.04676* (2019).

[40] Gianluca Detommaso, Tim Dodwell, and Rob Scheichl. “Continuous level Monte Carlo and sample-adaptive model hierarchies”. In: *SIAM/ASA Journal on Uncertainty Quantification* 7.1 (2019), pp. 93–116.

[41] Gianluca Detommaso, Tiangang Cui, Youssef Marzouk, Alessio Spantini, and Robert Scheichl. “A Stein variational Newton method”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 9169–9179.

[42] Gianluca Detommaso, Hanne Hoitzing, Tiangang Cui, and Ardavan Alamir. “Stein Variational Online Changepoint Detection with Applications to Hawkes Processes and Neural Networks”. In: *arXiv preprint arXiv:1901.07987* (2019).

[43] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. “Density estimation using Real NVP”. In: *International Conference on Learning Representations* (2017).

[44] Tim J Dodwell, Chris Ketelsen, Robert Scheichl, and Aretha L Teckentrup. “A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to Uncertainty Quantification in subsurface flow”. In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1 (2015), pp. 1075–1108.

[45] TJ Dodwell, C Ketelsen, R Scheichl, and AL Teckentrup. “Multilevel Markov chain Monte Carlo”. In: *SIAM Review* 61.3 (2019), pp. 509–545.

[46] Carl Doersch. “Tutorial on variational autoencoders”. In: *arXiv preprint arXiv:1606.05908* (2016).

[47] Sergey Dolgov, Karim Anaya-Izquierdo, Colin Fox, and Robert Scheichl. “Approximation and sampling of multivariate probability distributions in the tensor train decomposition”. In: *arXiv preprint arXiv:1810.01212* (2018).

[48] Arnaud Doucet, Nando De Freitas, and Neil Gordon. “An introduction to sequential Monte Carlo methods”. In: *Sequential Monte Carlo methods in practice*. Springer, 2001, pp. 3–14.

[49] Rick Durrett. *Probability: theory and examples*. Vol. 49. Cambridge university press, 2019.

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

[51] Geir Evensen. *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[52] Geir Evensen. “The ensemble Kalman filter: Theoretical formulation and practical implementation”. In: *Ocean dynamics* 53.4 (2003), pp. 343–367.

[53] Damien Francois, Vincent Wertz, Michel Verleysen, et al. “About the locality of kernels in high-dimensional spaces”. In: *International Symposium on Applied Stochastic Models and Data Analysis*. Citeseer. 2005, pp. 238–245.

[54] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. 10. Springer series in statistics New York, 2001.

[55] Roger G Ghanem and Pol D Spanos. *Stochastic finite elements: a spectral approach*. Courier Corporation, 2003.

[56] Michael Giles. *Multilevel Monte Carlo research*. https://people.maths.ox.ac.uk/gilesm/mlmc_community.html.

[57] Michael B Giles. “Multilevel Monte Carlo methods”. In: *Acta Numerica* 24 (2015), pp. 259–328.

[58] Michael B Giles. “Multilevel Monte Carlo path simulation”. In: *Operations Research* 56.3 (2008), pp. 607–617.

[59] Michael B Giles and Christoph Reisinger. “Stochastic finite differences and multilevel Monte Carlo for a class of SPDEs in finance”. In: *SIAM Journal on Financial Mathematics* 3.1 (2012), pp. 572–592.

[60] Michael B Giles and Lukasz Szpruch. “Multilevel Monte Carlo methods for applications in finance”. In: *High-Performance Computing in Finance*. Chapman and Hall/CRC, 2018, pp. 197–247.

[61] Michael B Giles and Benjamin J Waterhouse. “Multilevel quasi-Monte Carlo path simulation”. In: *Advanced Financial Modelling, Radon Series on Computational and Applied Mathematics* 8 (2009), pp. 165–181.

[62] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC, 1995.

[63] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[64] Peter J Green, Krzysztof Łatuszyński, Marcelo Pereyra, and Christian P Robert. “Bayesian computation: a summary of the current state, and samples backwards and forwards”. In: *Statistics and Computing* 25.4 (2015), pp. 835–862.

[65] Alastair Gregory, Colin J Cotter, and Sebastian Reich. “Multilevel ensemble transform particle filtering”. In: *SIAM Journal on Scientific Computing* 38.3 (2016), A1317–A1338.

[66] Geoffrey Grimmett and David Stirzaker. *Probability and random processes.* Oxford university press, 2001.

[67] Heikki Haario, Eero Saksman, and Johanna Tamminen. “An adaptive Metropolis algorithm”. In: *Bernoulli* 7.2 (2001), pp. 223–242.

[68] Heikki Haario, Marko Laine, Antonietta Mira, and Eero Saksman. “DRAM: efficient adaptive MCMC”. In: *Statistics and computing* 16.4 (2006), pp. 339–354.

[69] Abdul-Lateef Haji-Ali, Fabio Nobile, and Raúl Tempone. “Multi-index Monte Carlo: when sparsity meets sampling”. In: *Numerische Mathematik* 132.4 (2016), pp. 767–806.

[70] Timothy Classen Hesterberg. “Advances in importance sampling”. PhD thesis. Stanford University, 1988.

[71] Viet Ha Hoang, Christoph Schwab, and Andrew M Stuart. “Complexity analysis of accelerated MCMC methods for Bayesian inversion”. In: *Inverse Problems* 29.8 (2013), p. 085010.

[72] Håkon Hoel, Kody JH Law, and Raúl Tempone. “Multilevel ensemble Kalman filtering”. In: *SIAM Journal on Numerical Analysis* 54.3 (2016), pp. 1813–1839.

[73] Peter D Hoff. *A first course in Bayesian statistical methods.* Vol. 580. Springer, 2009.

[74] Matthew D Hoffman and Andrew Gelman. “The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo.” In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623.

[75] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. “Stochastic variational inference”. In: *The Journal of Machine Learning Research* 14.1 (2013), pp. 1303–1347.

[76] Robert V Hogg, Joseph McKean, and Allen T Craig. *Introduction to mathematical statistics.* Pearson Education, 2005.

[77] Peter L Houtekamer and Herschel L Mitchell. “Data assimilation using an ensemble Kalman filter technique”. In: *Monthly Weather Review* 126.3 (1998), pp. 796–811.

[78] Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. "Neural autoregressive flows". In: *arXiv preprint arXiv:1804.00779* (2018).

[79] SP Huang, ST Quek, and KK Phoon. "Convergence study of the truncated Karhunen–Loeve expansion for simulation of stochastic processes". In: *International journal for numerical methods in engineering* 52.9 (2001), pp. 1029–1043.

[80] Brian R Hunt, Tim Sauer, and James A Yorke. "Prevalence: a translation-invariant "almost every" on infinite-dimensional spaces". In: *Bulletin of the American mathematical society* 27.2 (1992), pp. 217–238.

[81] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

[82] Durk P Kingma and Prafulla Dhariwal. "Glow: Generative flow with invertible 1x1 convolutions". In: *Advances in Neural Information Processing Systems*. 2018, pp. 10215–10224.

[83] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems*. 2016, pp. 4743–4751.

[84] Ralf Kornhuber and Evgenia Youett. "Adaptive multilevel Monte Carlo methods for stochastic variational inequalities". In: *SIAM Journal on Numerical Analysis* 56.4 (2018), pp. 1987–2007.

[85] Jakob Kruse, Gianluca Detommaso, Robert Scheichl, and Ullrich Köthe. "HINT: Hierarchical Invertible Neural Transport for density dstimation and Bayesian inference". In: *arXiv* (2019), arXiv–1905.

[86] Frances Kuo, Robert Scheichl, Christoph Schwab, Ian Sloan, and Elisabeth Ullmann. "Multilevel quasi-Monte Carlo methods for lognormal diffusion problems". In: *Mathematics of Computation* 86.308 (2017), pp. 2827–2860.

[87] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *nature* 521.7553 (2015), pp. 436–444.

[88] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian dynamics*. Vol. 14. Cambridge university press, 2004.

[89] Christopher Lester, Christian Adam Yates, Michael B Giles, and Ruth E Baker. "An adaptive multi-level simulation algorithm for stochastic biological systems". In: *The Journal of chemical physics* 142.2 (2015), 01B612_1.

[90] Qiang Liu. "Stein variational gradient descent as gradient flow". In: *Advances in neural information processing systems*. 2017, pp. 3115–3123.

[91] Qiang Liu and Dilin Wang. "Stein variational gradient descent: A general purpose bayesian inference algorithm". In: *Advances in neural information processing systems*. 2016, pp. 2378–2386.

[92] Gabriel J Lord, Catherine E Powell, and Tony Shardlow. *An introduction to computational stochastic PDEs*. Vol. 50. Cambridge University Press, 2014.

[93] Jianfeng Lu, Yulong Lu, and James Nolen. “Scaling limit of the Stein variational gradient descent: The mean field regime”. In: *SIAM Journal on Mathematical Analysis* 51.2 (2019), pp. 648–671.

[94] Youssef Marzouk, Tarek Moselhy, Matthew Parno, and Alessio Spantini. “Sampling via measure transport: An introduction”. In: *Handbook of Uncertainty Quantification* (2016), pp. 1–41.

[95] Nicholas Metropolis and Stanislaw Ulam. “The Monte Carlo method”. In: *Journal of the American statistical association* 44.247 (1949), pp. 335–341.

[96] Antje Mugler and Hans-Jörg Starkloff. “On elliptic partial differential equations with random coefficients”. In: *Preprint* 79 (2011).

[97] Joseph B Nagel and Bruno Sudret. “A unified framework for multilevel uncertainty quantification in Bayesian inverse problems”. In: *Probabilistic Engineering Mechanics* 43 (2016), pp. 68–84.

[98] Radford M Neal. “Annealed importance sampling”. In: *Statistics and computing* 11.2 (2001), pp. 125–139.

[99] Radford M Neal. “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov chain Monte Carlo* 2.11 (2011), p. 2.

[100] Barry L Nelson. “Control variate remedies”. In: *Operations Research* 38.6 (1990), pp. 974–992.

[101] Harald Niederreiter. *Random number generation and quasi-Monte Carlo methods*. Vol. 63. Siam, 1992.

[102] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[103] Anthony Nouy, Alexandre Clement, Franck Schoefs, and N Moës. “An extended stochastic finite element method for solving stochastic partial differential equations on random domains”. In: *Computer Methods in Applied Mechanics and Engineering* 197.51-52 (2008), pp. 4663–4682.

[104] George Papamakarios, Theo Pavlakou, and Iain Murray. “Masked autoregressive flow for density estimation”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 2338–2347.

[105] Benjamin Peherstorfer, Boris Kramer, and Karen Willcox. “Multifidelity methods for rare event simulation”. In: *Book of Abstracts ENUMATH* (2017), p. 157.

[106] Benjamin Peherstorfer, Boris Kramer, and Karen Willcox. “Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation”. In: *SIAM/ASA Journal on Uncertainty Quantification* 6.2 (2018), pp. 737–761.

[107] Benjamin Peherstorfer, Tiangang Cui, Youssef Marzouk, and Karen Willcox. “Multifidelity importance sampling”. In: *Computer Methods in Applied Mechanics and Engineering* 300 (2016), pp. 490–509.

[108] Marcelo Pereyra. “Maximum-A-Posteriori estimation with Bayesian confidence regions”. In: *SIAM Journal on Imaging Sciences* 10.1 (2017), pp. 285–302.

[109] Philip E Protter. “Stochastic differential equations”. In: *Stochastic integration and differential equations*. Springer, 2005, pp. 249–361.

[110] Sebastian Reich. “A nonparametric ensemble transform method for Bayesian inference”. In: *SIAM Journal on Scientific Computing* 35.4 (2013), A2013–A2024.

[111] Danilo Jimenez Rezende and Shakir Mohamed. “Variational inference with normalizing flows”. In: *Proceedings of the 32nd International Conference on Machine Learning* (2015).

[112] Chang-han Rhee and Peter W Glynn. “A new approach to unbiased estimation for SDE’s”. In: *Proceedings of the Winter Simulation Conference*. Winter Simulation Conference. 2012, p. 17.

[113] Gareth O Roberts, Andrew Gelman, and Walter R Gilks. “Weak convergence and optimal scaling of random walk Metropolis algorithms”. In: *The annals of applied probability* 7.1 (1997), pp. 110–120.

[114] Walter Rudin. *Real and complex analysis*. Tata McGraw-hill education, 2006.

[115] Simo Särkkä. *Bayesian filtering and smoothing*. Vol. 3. Cambridge University Press, 2013.

[116] R Scheichl, AM Stuart, and AL Teckentrup. “Quasi-Monte Carlo and multilevel Monte Carlo methods for computing posterior expectations in elliptic inverse problems”. In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1 (2017), pp. 493–518.

[117] Zhenming Shun and Peter McCullagh. “Laplace approximation of high dimensional integrals”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.4 (1995), pp. 749–760.

[118] Andrew M Stuart. “Inverse problems: a Bayesian perspective”. In: *Acta numerica* 19 (2010), pp. 451–559.

[119] Andrew M Stuart, Jochen Voss, and Petter Wilberg. “Conditional path sampling of SDEs and the Langevin MCMC method”. In: *Communications in Mathematical Sciences* 2.4 (2004), pp. 685–697.

[120] Timothy John Sullivan. *Introduction to Uncertainty Quantification*. Vol. 63. Springer, 2015.

[121] Aretha Leonore Teckentrup. “Multilevel Monte Carlo methods and Uncertainty Quantification”. PhD thesis. University of Bath, 2013.

[122] Erik Vanmarcke. *Random fields: analysis and synthesis.* World Scientific, 2010.

[123] Martin J Wainwright and Michael I Jordan. “Graphical models, exponential families, and variational inference”. In: *Foundations and Trends® in Machine Learning* 1.1–2 (2008), pp. 1–305.

[124] Dilin Wang, Ziyang Tang, Chandrajit Bajaj, and Qiang Liu. “Stein Variational Gradient Descent With Matrix-Valued Kernels”. In: *arXiv preprint arXiv:1910.12794* (2019).

[125] Zheng Wang, Tiangang Cui, Johnathan Bardsley, and Youssef Marzouk. “Scalable optimization-based sampling on function space”. In: *arXiv preprint arXiv:1903.00870* (2019).

[126] Stephen Whitaker. “Flow in porous media I: A theoretical derivation of Darcy’s law”. In: *Transport in porous media* 1.1 (1986), pp. 3–25.

[127] Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. “Langevin diffusions and the Metropolis-adjusted Langevin algorithm”. In: *Statistics & Probability Letters* 91 (2014), pp. 14–19.

[128] Dongbin Xiu and Jan S Hesthaven. “High-order collocation methods for differential equations with random inputs”. In: *SIAM Journal on Scientific Computing* 27.3 (2005), pp. 1118–1139.

[129] Dongbin Xiu and George Em Karniadakis. “Modeling uncertainty in flow simulations via generalized polynomial chaos”. In: *Journal of computational physics* 187.1 (2003), pp. 137–167.

[130] Olgierd Cecil Zienkiewicz, Robert L Taylor, Perumal Nithiarasu, and JZ Zhu. *The finite element method.* Vol. 3. McGraw-hill London, 1977.

■ Paper I

Continuous Level Monte Carlo and Sample-Adaptive Model Hierarchies

Gianluca Detommaso[1,2], Tim Dodwell[3] and Rob Scheichl[2]

[1] The Alan Turing Institute, London, NW1 2DB, UK. Email: gdetommaso@turing.ac.uk

[2] Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK.

[3] College of Engineering, Mathematics and Physical Sciences, University of Exeter, Exeter, EX4 4PY, UK.

Abstract

In this paper, we present a generalisation of the Multilevel Monte Carlo (MLMC) method to a setting where the level parameter is a continuous variable. This Continuous Level Monte Carlo (CLMC) estimator provides a natural framework in PDE applications to adapt the model hierarchy to each sample. In addition, it can be made unbiased with respect to the expected value of the true quantity of interest provided the quantity of interest converges sufficiently fast. The practical implementation of the CLMC estimator is based on interpolating actual evaluations of the quantity of interest at a finite number of resolutions. As our new level parameter, we use the logarithm of a goal-oriented finite element error estimator for the accuracy of the quantity of interest. We prove the unbiasedness, as well as a complexity theorem that shows the same rate of complexity for CLMC as for MLMC. Finally, we provide some numerical evidence to support our theoretical results, by successfully testing CLMC on a standard PDE test problem. The numerical experiments demonstrate clear gains for sample-wise adaptive refinement strategies over uniform refinements.

1 Introduction

No matter whether epistemic or aleatoric, known unknown or unknown unknown, uncertainty plays a fundamental role in any real life situation. Its quantification is becoming an object of interest for ever more complex problems, where accurate solutions require huge computational costs. A lot of methods have been proposed in the last decade that aim to reduce this cost without affecting the accuracy. Among others, multilevel techniques conquered the scene arising in a multitude of algorithms, all following the pioneering work on *multilevel Monte Carlo* (MLMC) by Giles [10] and the earlier paper by Heinrich [14] (see also [9, 4] and references therein). In general, multilevel techniques aim to accelerate inference by exploiting a hierarchy of models with different levels of accuracy. By combining estimates from all the models in a telescoping sum, the computational cost is shifted towards the bottom (cheap and inaccurate) end of the hierarchy, while maintaining the accuracy of the top (expensive and high resolution) end.

Since the initial work on MLMC, several techniques have been employed to exploit model structures even further with considerable savings in computational cost. An important step forward was the introduction of *adaptive multilevel Monte Carlo* (AMLMC) [15], where error estimates and adaptive refinement strategies are exploited to increase the accuracy only where needed (see also [9, 8, 16] in the context of PDEs). In contrast to the majority of the literature on MLMC, which is based on uniform refinements, AMLMC is able to deal with problems with very localised sample-dependent noise or quantities of interest, avoiding excessive computational cost by refining the models only where necessary and, in general, differently for each sample.

A second important step forward was the introduction of an MLMC estimator that is unbiased with respect to the real quantity of interest [21] (see also [19, 24]). In most problems of consideration, the quantity of interest is a functional of the solution of an inaccessible, infinite-dimensional model. In such cases, standard MLMC is only able to provide an estimator that is unbiased with respect to an approximation of the real quantity of interest. Having an unbiased estimator for the real quantity of interest is often of great practical interest, especially if the estimator is used for further predictions. Furthermore, the bias error is typically harder to estimate than the sampling error, making it easier to avoid unnecessary computational effort with an unbiased estimator.

In this paper, we present a generalisation of MLMC to a continuous framework that we denote *continuous level Monte Carlo* (CLMC), where the underlying hierarchical structure is considered to be continuous rather than a finite sequence of discrete instances. The level parameter ℓ is assumed to be a real number rather than an integer, giving access to standard tools from Calculus, such as the integral or the derivative with respect to the level. Although this might sound just like a conceptual generalisation, we will interestingly see how the continuous framework also allows deeper understanding and different perspectives. As a first fact, it highlights a link with tools from probability theory, since the continuous sequence of approximations can now be interpreted as a continuous stochastic process over the level of resolution. In this framework, the classic telescoping sum of MLMC straightforwardly becomes a simplified version of Dynkin's formula [20], or more simply the Fundamental Theorem of Calculus. As allowed in Dynkin's formula, the finest level L of resolution can be chosen as a stopping time random variable, which stops the refining procedure differently for each sample according to some probability distribution over L. We will see that there is a simple probability distribution over L corresponding to the optimal decaying sequence of the number of samples in MLMC and the choice of this distribution is not very sensitive to an accurate estimation of the convergence rates and of the cost per sample.

The main result of the paper is a continuous version of the complexity theorem for MLMC. This provides two main contributions:

- it introduces a CLMC estimator that, under standard assumptions, satisfies the same computational cost rate as the one in MLMC;
- it proves that the CLMC estimator can potentially be unbiased, *but* the unbiased version has finite computational cost exclusively when the variance decays faster than the cost per sample grows.

Among potential applications, the continuous level framework finds his practical utility for sample-dependent hierarchical refinements: when the refinement levels depend on samples instead of being fixed, it is more natural to think of them in a continuous fashion, as the resolution of a particular model can fall anywhere on the real line. This is a typical situation in AMLMC. Indeed, the resolution level is usually interpreted as the logarithm of the error of the numerical model, therefore intrinsically continuous. Moreover, the error is sample-dependent, hence each sample will hit its own sequence of level refinements. As AMLMC involves taking sample averages of quantities of interests at some prescribed levels, approximations have to be made that may lead to slight inefficiencies especially when the improvement in the approximation error in each adaptation step varies strongly (see [16]).

Here, we develop practical CLMC algorithms that are easy to implement and do not require any such approximation. As we can arbitrarily choose the nature of the quantity of interest between the actual evaluations, to obtain a quantity of interest function that is continuous over the levels we simply interpolate the calculated values, whence we can work out a practical formula. Note that the practical formula can also be implemented for the unbiased version of the CLMC estimator. Finally, we provide some numerical experiments showing the CLMC algorithm in action for a standard two-dimensional model problem where the adaptivity and the sample-dependent hierarchies are shown to leading to significant computational savings.

The structure of the paper is as follows. In Section 2, we give a short background of Monte Carlo and MLMC; we present the main CLMC idea; we introduce the CLMC estimator and show the unbiasedness property; we state the CLMC complexity theorem; we provide a corollary showing when the estimator that provides the optimal cost is unbiased with respect to the real quantity of interest. In Section 3, we propose a practical CLMC algorithm for sample-based adaptive hierarchical refinement; we discuss the special case of uniform refinement and the similarities with MLMC and show the link between the distribution of the finest level and the sequence of number of samples; we finish the section with some proposals for other possible implementations and approaches. Finally, Section 4 introduces the PDE model problem and the adaptive finite element hierarchy for them, as well as presenting and discussing the numerical experiments. We finish the paper with some conclusions and ideas for future work in Section 5. The detailed proof of the complexity theorem, as well as some details about the goal-oriented error estimator are delegated to the appendices.

2 Continuous level Monte Carlo

2.1 Background: Monte Carlo and Multilevel Monte Carlo

Suppose one is interested in estimating the expected value $\mathbb{E}[\mathcal{Q}]$ of some (inaccessible) quantity of interest $\mathcal{Q}$, for simplicity assumed to be scalar. In uncertainty quantification (UQ), $\mathcal{Q}$ is typically a functional of the solution of some random partial differential equation (PDE), where the randomness can lie anywhere, e.g. within the coefficients, the source, the boundary conditions or the shape of the domain.

In general, the solution of a PDE can not be calculated exactly and it has to be approximated numerically, up to some desirable resolution level L. Let us call Q_L such an approximation and assume that $Q_L \to \mathcal{Q}$ almost surely (a.s.) for $L \to +\infty$. Then, for any desired tolerance $\varepsilon > 0$, there exists a fine enough resolution L, such that $|\mathbb{E}[\mathcal{Q} - Q_L]| \leq \varepsilon$, and we can focus on finding good algorithms to estimate $\mathbb{E}[Q_L]$ to the same accuracy. There are two main issues here.

1. If the underlying probability distribution is continuous and high-dimensional, which is common in UQ applications, it can be extremely expensive to approximate the expected value with standard quadrature methods.

2. If the resolution L required to compute the PDE solution with sufficient accuracy is high, then computing just one sample of Q_L will be expensive and the number of samples that can be computed on level L in a reasonable time is limited.

A standard remedy for Issue 1 is the use of *Monte Carlo* (MC) methods [22]. Indeed, the rate of converge of MC estimators is independent of the dimension of the integral and it is extremely easy to implement: given N independent samples $\big(Q_L^{(k)}\big)_{k=1}^N$ of Q_L, distributed according to the underlying probability distribution, the expected value can be estimated as

$$\mathbb{E}[Q_L] \approx \frac{1}{N}\sum_{k=1}^{N} Q_L^{(k)}. \tag{1}$$

Whilst the right-hand-side in (1) is an unbiased estimator of $\mathbb{E}[Q_L]$, unfortunately it converges very slow, especially when L is large, since $\mathcal{O}(\varepsilon^{-2})$ samples are required to reduce the sampling error to a given accuracy ε, i.e. $|\mathbb{E}[\mathcal{Q}-Q_L]| \leq \varepsilon$. As every sample requires an expensive PDE solve, the computational cost quickly becomes infeasible for small ε.

An acceleration technique suggested for (1) is the *multilevel Monte Carlo* (MLMC) method [14, 10]. It exploits a hierarchy of approximations $Q_0, Q_1, \dots, Q_L$ of $\mathcal{Q}$ at different resolutions, starting with a coarse and cheap approximation Q_0, and going up to the fine and expensive approximation Q_L. In contrast to the standard MC estimator in (1), which directly estimates $\mathbb{E}[Q_L]$ by sampling Q_L, MLMC combines samples from the sequence of approximations $(Q_\ell)_{\ell=0}^L$

to produce an overall cheaper estimator. To this purpose, the approximations are combined into the telescoping sum

$$\mathbb{E}[Q_L - Q_0] = \sum_{\ell=1}^{L} \mathbb{E}[Q_\ell - Q_{\ell-1}], \tag{2}$$

and then each term in the sum on the right-hand-side is estimated with Monte Carlo:

$$\mathbb{E}[Q_\ell - Q_{\ell-1}] \approx \frac{1}{N_\ell} \sum_{k=1}^{N_\ell} \left(Q_\ell^{(k)} - Q_{\ell-1}^{(k)} \right). \tag{3}$$

To obtain an estimator for $\mathbb{E}[Q_L]$ it suffices to add a Monte Carlo estimator for $\mathbb{E}[Q_0]$.

Crucially, the consecutive approximations $Q_{\ell-1}^{(k)}$ and $Q_\ell^{(k)}$ in the difference $Q_\ell^{(k)} - Q_{\ell-1}^{(k)}$ come from the same sample k. This means that they are strongly positively correlated, and the variance of the difference is heavily reduced:

$$\mathbb{V}[Q_\ell - Q_{\ell-1}] = \mathbb{V}[Q_{\ell-1}] + \mathbb{V}[Q_\ell] - 2\mathrm{Cov}(Q_{\ell-1}, Q_\ell) \ll \mathbb{V}[Q_{\ell-1}] + \mathbb{V}[Q_\ell]. \tag{4}$$

As $Q_\ell \to \mathcal{Q}$ a.s. for $\ell \to +\infty$, we also have $Q_\ell - Q_{\ell-1} \to 0$, so that the covariance, and in turn the variance reduction, increases as $\ell \to +\infty$. As a consequence, the required number of samples N_ℓ at level ℓ can be chosen to decrease monotonically with increasing ℓ, so that only very few expensive samples on level L are needed. The majority of samples and therefore the computational cost will be shifted to the coarser levels.

This reduction in computational complexity can be quantified rigorously, at least asymptotically as the tolerance $\varepsilon \to 0$. The complexity theorems in [10, 4] show that the overall computational cost for the MLMC algorithm can be up to a factor $\mathcal{O}(\varepsilon^2)$ smaller than the cost of the MC estimator in (1). We will return to this and give more details in Section 2.4.

2.2 Continuous Level Monte Carlo: the main idea

In this section, we introduce the *continuous level Monte Carlo* (CLMC) idea. As we have seen above, MLMC exploits a discrete sequence of approximations $(Q_\ell)_{\ell=0}^L$ of $\mathcal{Q}$. We now extend this to a continuous family of approximations $(Q(\ell))_{\ell\geq 0}$ of $\mathcal{Q}$. In other words, $(Q(\ell))_{\ell\geq 0}$ is a stochastic process of approximations over the continuous level of resolution ℓ.

Let L be assumed to be a random variable with finite expectation denoting the (random) finest level of resolution, independent from the stochastic process $(Q(\ell))_{\ell\geq 0}$. Also, let $L_{\max} \in [0, \infty]$ be a deterministic constant that we introduce for reasons that will become clearer later. We can write down the following formula:

$$\mathbb{E}[Q(L \wedge L_{\max}) - Q(0)] = \mathbb{E}\left[\int_0^{L \wedge L_{\max}} \frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell} \,\mathrm{d}\ell \right]. \tag{5}$$

For the formula in (5) to be well-posed, we need to assume that $Q \in W^{1,1}(0, L_{\max})$ as a function of ℓ, where $W^{1,1}(0, L_{\max})$ is a Sobolev space containing functions over $\ell \in (0, L_{\max})$ such that the functions and their weak first derivatives have finite L^1 norm. Note that for simplicity we are choosing 0 as coarsest level, but this can of course be generalised.

If we assume L to be a deterministic variable, the expectation in (5) can be pulled inside the integral and the derivative, so that equation (5) reduces to the Fundamental Theorem of Calculus, which guarantees the identity. However, more generally, equation (5) can be recovered as a particular case of Dynkin's Formula [20], where L is interpreted as a finite stopping time.

2.3 The CLMC estimator

Let us assume L to be a random variable independent of the whole stochastic process $(Q(\ell))_{\ell\geq 0}$. We can then define the *continuous level Monte Carlo (CLMC) estimator*

$$\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}} := \frac{1}{N}\sum_{k=1}^{N}\int_0^{L_{\max}} \frac{1}{\mathbb{P}(L\geq \ell)}\left(\frac{\mathrm{d}Q}{\mathrm{d}\ell}\right)^{(k)}(\ell)\,\mathbb{1}_{[0,L^{(k)}]}(\ell)\,\mathrm{d}\ell, \tag{6}$$

where the superscript (k) denotes the k-th realisation of the respective random variable and N is the total number of samples. For simplicity of presentation, the estimator $\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}$ is defined as an estimator for $\mathbb{E}[Q(L_{\max}) - Q(0)]$, as we see in Proposition 2.1. As in standard MLMC, it suffices to add an unbiased estimator for $\mathbb{E}[Q(0)]$ to obtain an estimator for $\mathbb{E}[Q(L_{\max})]$.

A reader familiar with the MLMC literature might be puzzled by the estimator in (6), where we use the same number of samples N for each level ℓ. However, note that, for each sample k, the integrand in (6) will only be non-zero up to the random realisation $L^{(k)}$ of L, and therefore in practice we do not need to evaluate $Q(\ell)$ beyond level $L^{(k)}$.

We are now ready to show that the CLMC estimator is unbiased.

Proposition 2.1. *The CLMC estimator* (6) *is an unbiased estimator for* $\mathbb{E}[Q(L_{\max}) - Q(0)]$, *i.e.*

$$\mathbb{E}[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}] = \mathbb{E}[Q(L_{\max}) - Q(0)]\,.$$

Proof. By exploiting the independence of L from $(Q(\ell))_{\ell\geq 0}$, we have

$$\begin{aligned}
\mathbb{E}\left[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}\right] &= \mathbb{E}\left[\frac{1}{N}\sum_{k=1}^{N}\int_0^{L_{\max}}\frac{1}{\mathbb{P}(L\geq\ell)}\left(\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right)^{(k)}\mathbb{1}_{[0,L^{(k)}]}(\ell)\,\mathrm{d}\ell\right]\\
&= \int_0^{L_{\max}}\frac{1}{\mathbb{P}(L\geq\ell)}\mathbb{E}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]\mathbb{E}\left[\mathbb{1}_{[0,L]}(\ell)\right]\,\mathrm{d}\ell\\
&= \int_0^{L_{\max}}\frac{1}{\mathbb{P}(L\geq\ell)}\mathbb{E}\left[\frac{\mathrm{d}Q}{\mathrm{d}\ell}(\ell)\right]\mathbb{P}(L\geq\ell)\,\mathrm{d}\ell\\
&= \int_0^{L_{\max}}\mathbb{E}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]\,\mathrm{d}\ell\\
&= \mathbb{E}[Q(L_{\max}) - Q(0)].
\end{aligned}$$

□

In particular, this implies the following important corollary.

Corollary 2.2. *If* $L_{\max} = +\infty$, *then*

$$\mathbb{E}[\widehat{Q}^{\mathrm{CLMC}}_{\infty}] = \mathbb{E}[\mathcal{Q} - Q(0)].$$

Corollary 2.2 shows that there is a version of the estimator (6) that is unbiased with respect to the expectation of the difference of the real quantity of interest $\mathcal{Q}$ and $Q(0)$, and one can see the connection with the unbiased MLMC estimator introduced in [21].

In the next subsection, we will prove a complexity theorem for the CLMC estimator (6). We will pick L to be distributed as an exponential random variable to facilitate calculations and mimic the exponential decay in the assumptions on the convergence of the quantity of interest. Also, we will provide sufficient and necessary conditions for the Theorem to hold in the case $L_{\max} = +\infty$, i.e. when the CLMC estimator is unbiased with respect to $\mathcal{Q} - Q(0)$. A practical algorithm will then be described in Section 3.

2.4 Complexity theorem

The fundamental theoretical result about the MLMC method is the complexity theorem, firstly proved in [10] and generalised in [4]. In this section, we state an analogous complexity theorem for the CLMC estimator (6). A full proof is given in Appendix A.

First, let us define the mean-squared-error (MSE) of the CLMC estimator $\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}$ in (6) by

$$\mathrm{MSE} := \mathbb{E}\left[\left(\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}} - \mathbb{E}[\mathcal{Q} - Q(0)]\right)^2\right] \tag{7}$$

and denote by $\mathcal{C}^{\mathrm{CLMC}}_{L_{\max}}$ its expected computational cost. Then, we have the following result.

Theorem 2.3 (Complexity Theorem)**.** *Suppose $\mathcal{Q}$ is a quantity of interest and $Q \in W^{1,1}(0,\infty)$ is a corresponding family of numerical approximations. Furthermore, suppose that there are positive constants $\alpha,\ \beta \le 2\alpha,\ \gamma,\ c_1,\ c_2,\ c_3$ such that, for any $\ell > 0$, we have:*

(i) $\left|\mathbb{E}\left[\frac{dQ(\ell)}{d\ell}\right]\right| \le c_1 e^{-\alpha\ell}$, *(ii)* $\mathbb{V}\left[\frac{dQ(\ell)}{d\ell}\right] \le c_2 e^{-\beta\ell}$,

(iii) $\mathcal{C}(\ell) \le c_3 e^{\gamma\ell}$ *, where $\mathcal{C}(\ell)$ is the cost to compute one sample of $Q(\ell)$.*

Furthermore, suppose that $L \sim \mathrm{Exponential}(r)$ with

$$r \in [\min(\beta,\gamma),\ \max(\beta,\gamma)].$$

Then, for any $\varepsilon \in (0, e^{-1})$, there exist $L_{\max} \in [0,+\infty)$, $N \in \mathbb{N}$ and $C > 0$ such that

$$MSE \ \le\ \varepsilon^2 \quad \text{and} \quad \mathcal{C}^{\mathrm{CLMC}}_{L_{\max}} \ \le\ C\,\varepsilon^{-2-\max(0,\frac{\gamma-\beta}{\alpha})}(\log\varepsilon)^{\delta_{r,\beta}+\delta_{r,\gamma}} \tag{8}$$

with δ denoting the Kronecker delta.

Note that the predicted computational cost in Theorem 2.3 is the same as in MLMC (asymptotically).

Corollary 2.4. *Suppose that the assumptions of Theorem 2.3 hold and that $L_{\max} = +\infty$, i.e. let us consider the unbiased CLMC estimator $\widehat{Q}^{CLMC}_{\infty}$.*

(a) If $\beta > \gamma$, then for any $\varepsilon \in (0, e^{-1})$ and for any $r \in (\gamma, \beta)$, there exists an $N \in \mathbb{N}$ and $C > 0$ such that

$$MSE \le \varepsilon^2 \qquad \text{and} \qquad \mathcal{C}^{\mathrm{CLMC}}_{\infty} \le C\varepsilon^{-2}\,.$$

(b) If $\beta \le \gamma$ and, in addition, there exist positive constants $\eta \in [\beta,\gamma]$, c_2' and c_3' such that

$$c_2' e^{-\eta\ell} \le \mathbb{V}\left[\frac{dQ(\ell)}{d\ell}\right] \quad \text{and} \quad c_3' e^{\eta\ell} \le \mathcal{C}(\ell)\,,$$

then $MSE \times \mathcal{C}^{\mathrm{CLMC}}_{\infty} = +\infty$, for all $r > 0$ and $N \in \mathbb{N}$, i.e. the unbiased estimator has infinite MSE or infinite cost.

Corollary 2.4 provides sufficient and necessary conditions for the CLMC estimator with $L_{\max} = +\infty$ (which is unbiased with respect to $\mathcal{Q} - Q(0)$) to have a finite expected complexity cost. Intuitively, since $L_{\max} = +\infty$, the finest level at which computations are needed is $\max_{k=1}^{N} L^{(k)}$, which tends to infinity as N grows. Therefore, the estimator (6) will have finite expected cost only if the actual variance reduction rate is bigger than the actual cost growth rate. The rates β and γ in Theorem 2.3 are only upper bounds. By analogy, we believe this constraint also applies to the unbiased estimator introduced by Rhee & Glynn [21]. However, the paper [21] is mainly concerned with timestepping methods for SDEs, where the condition $\gamma < \beta$ is usually satisfied.

Note that, if $L_{\max} = +\infty$, even in the case $\beta > \gamma$, there is a non-zero probability that the finest level $L^{(k)}$ for some sample (k) is drawn larger than the maximal refinement achievable on the particular machine that is used, but we can exactly quantify the probability for this to happen. Indeed, if $\bar{L}$ is the maximum refinement level achievable by the machine, the probability that at least one sample is greater or equal than $\bar{L}$ is given by

$$N\mathbb{P}(L \geq \bar{L}) = N \exp(-r\bar{L}).$$

We will see that for problems of interests this probability is very small. In the rare event that $L^{(k)} > \bar{L}$ for some sample k, one could simply approximate $Q^{(k)}(\ell) = Q^{(k)}(\bar{L})$ for $\ell \in [\bar{L}, L^{(k)}]$. If $\bar{L}$ is sufficiently large, this would introduce a negligible bias error to any practical values of ε.

3 Practical implementation

In the previous section, we have seen that it is possible to extend multilevel Monte Carlo to a continuous framework, where the approximations of the quantity of interest are functions over a continuous family of resolutions. This point of view comes natural when the level parameter is not associated with some fixed hierarchy of approximations, but with an adaptively chosen hierarchy for each sample, e.g. in the context of adaptive finite element approximations of a PDE with random coefficients where the level parameter ℓ is related to the accuracy of the approximation (see Section 4).

However, it still remains to show how this can be implemented in practice and how the practical implementation differs from MLMC. There are many possible ways to implement the estimator in (6). Let us first focus in some sense on the simplest one. We will comment on other approaches at the end of this section.

3.1 Sample-dependent level hierarchies and piecewise linear interpolation

Let us assume that we have estimates of the parameters α, β, γ in Theorem 2.3. In practice, these can be obtained (on the fly) from sample averages and sample variances of $Q(\ell)$ and $\mathrm{d}Q(\ell)/\mathrm{d}\ell$, as in standard MLMC. Then, given a desired tolerance $\varepsilon > 0$, Theorem 2.3 provides suitable choices for the number of samples N and for the rate r of the exponential distribution of L to achieve the optimal complexity in (8).

For any sample k, suppose that $(Q_j^{(k)})_{j\geq 1}$ denotes a countable sequence of approximations of $Q^{(k)}$ at levels $(\ell_j^{(k)})_{j\geq 1}$. Then, to define a continuous family $Q^{(k)}(\ell)$ of $Q^{(k)}$, we use linear interpolation such that

$$\left(\frac{\mathrm{d}Q}{\mathrm{d}\ell}\right)^{(k)}(\ell) := \frac{Q_j^{(k)} - Q_{j-1}^{(k)}}{\ell_j^{(k)} - \ell_{j-1}^{(k)}} \qquad \text{for } \ell \in (\ell_{j-1}^{(k)}, \ell_j^{(k)})\,.$$

Also, for each sample k, let us define the index $J^{(k)}$ corresponding to the first value of $\ell_j^{(k)}$ that is bigger than $L^{(k)} \wedge L_{\max}$, that is

$$J^{(k)} := \min\{j \geq 1 : \; \ell_j^{(k)} - (L^{(k)} \wedge L_{\max}) \geq 0\}.$$

Hence, we can write down the CLMC estimator (6) as

$$\begin{aligned}\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} &= \frac{1}{N}\sum_{k=1}^{N}\int_0^{L^{(k)}\wedge L_{\max}} \frac{1}{P(L \geq \ell)}\left(\frac{\mathrm{d}Q}{\mathrm{d}\ell}\right)^{(k)}(\ell)\,\mathrm{d}\ell \\ &= \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{J^{(k)}} w_j^{(k)}\left(Q_j^{(k)} - Q_{j-1}^{(k)}\right)\,, \qquad (9)\end{aligned}$$

where we define

$$\tilde{\ell}_j^{(k)} := \ell_j^{(k)} \wedge (L^{(k)} \wedge L_{\max}) \,, \tag{10}$$

and the integrals in the weights $w_j^{(k)}$ can be computed explicitly as

$$w_j^{(k)} := \frac{1}{\ell_j^{(k)} - \ell_{j-1}^{(k)}} \int_{\ell_{j-1}^{(k)}}^{\tilde{\ell}_j^{(k)}} \frac{1}{P(L \geq \ell)} \, \mathrm{d}\ell = \frac{\exp\big(r\tilde{\ell}_j^{(k)}\big) - \exp\big(r\ell_{j-1}^{(k)}\big)}{r\big(\ell_j^{(k)} - \ell_{j-1}^{(k)}\big)} \,, \tag{11}$$

for all $j = 1, \ldots, J^{(k)}$. Algorithm 1 provides the key instructions to implement the CLMC estimator in (9).

```
Algorithm 1: CLMC algorithm – Key steps
Input  : $\varepsilon$: tolerance;
         $r$: exponential rate;
         $N$: total number of samples;
         $L_{\max}$: maximum reachable level - potentially infinite if $\gamma < \beta$.
Output: $\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}$: CLMC estimator.
Initialise $\hat{Q} \leftarrow 0$;
for $k = 1, 2, \ldots, N$ do
    Sample $L^{(k)} \sim$ Exponential($r$);
    Evaluate and store quantity of interests $Q \leftarrow (Q_j^{(k)})_{j=1}^{J^{(k)}}$ at levels $\ell \leftarrow (\ell_j^{(k)})_{j=1}^{J^{(k)}}$;
    Calculate array $w$ of weights in (11);
    Update $\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} \leftarrow \widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} + w^T * \mathrm{diff}(Q)$, where diff($Q$) is the array of the
      differences between consecutive elements of $Q$;
end for
Set $\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} \leftarrow \widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} / N$.
```

Note that it is easy to work out an unbiased estimator for the variance of $\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}$ in (9), which is needed to estimate the total number of samples N. Let us define

$$Y^{(k)} := \sum_{j=1}^{J^{(k)}} w_j^{(k)} \left(Q_j^{(k)} - Q_{j-1}^{(k)} \right) \,.$$

Then (9) simply reduces to a standard Monte Carlo estimator with i.i.d. samples $Y^{(k)}$ and we can estimate

$$\mathbb{V}\left[\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}\right] \approx \frac{1}{N(N-1)} \sum_{k=1}^{N} \left(\left(Y^{(k)}\right)^2 - \left(\frac{1}{N} \sum_{i=1}^{N} Y^{(i)} \right)^2 \right) \,.$$

3.2 Uniform refinements as a special case

It is interesting to see what happens in the case of uniform refinements, where all samples $Q^{(k)}$, for $k = 1, \ldots, N$, are evaluated at the same deterministic points $\ell_j^{(k)} = \ell_j$, for $j \geq 1$, and then interpolated. Without loss of generality, we assume that $\ell_j = j$, as in standard MLMC.

In this case, the set of possible levels reduces to integers. Therefore, although a continuous probability distribution for L is still a valid choice, it is more natural to pick a discrete distribution over the levels, where $\mathbb{P}(L \geq j)$ is constant over the interval $(j-1, j)$. In that case, the practical CLMC estimator in (9) reduces to

$$\widehat{Q}^{\text{CLMC}}_{L_{\max}} = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{J^{(k)}} \frac{1}{\mathbb{P}(L \geq j)} \left(Q^{(k)}_j - Q^{(k)}_{j-1} \right) .$$

A natural choice would be a geometric distribution on L.

To see the relationship with the standard MLMC estimator more clearly, let us define

$$N(\ell) := N\mathbb{P}(L \geq \ell) .$$

Then, $(N(\ell))_{\ell \geq 0} \subset [0, \infty)$ corresponds to a continuous density of samples, analogous to the sequence of sample sizes at discrete levels in MLMC. Moreover, the probability that L is at least ℓ corresponds to the normalised density of samples that gets at least to level ℓ. Therefore, by plugging this relation in the equation above, we get

$$\widehat{Q}^{\text{CLMC}}_{L_{\max}} = \sum_{k=1}^{N}\sum_{j=1}^{J^{(k)}} \frac{1}{N(\ell)} \left(Q^{(k)}_j - Q^{(k)}_{j-1} \right) ,$$

which exactly corresponds to the Rhee & Glynn estimator in [21].

3.3 Other Implementations

3.3.1 Polynomial regression

Although the practical implementation discussed in Subsection 3.1 is a natural, practical implementation of the CLMC estimator, it is not the only possibility. One could think of exploiting the underlying continuous level structure in order to predict the global trend of the function $Q(\ell)$, thereby denoising the point-wise evaluations coming from the random samples. More concretely, imagine that each sample k provides evaluations $(Q^{(k)}_j)_{j=1}^{J^{(k)}}$ respectively at levels $(\ell^{(k)}_j)_{j=1}^{J^{(k)}}$. Instead of defining the function $Q^{(k)}(\ell)$ as the linear interpolant between the given points as in Subsection 3.1, one could define $Q^{(k)}(\ell)$ to be a particular polynomial interpolant or regression function. The resulting continuous function may not exactly interpolate the points but rather catch the global trend, avoiding to overfit sample-dependent noisy oscillations.

In general, for each sample k, define the polynomials

$$\left(\frac{\mathrm{d}Q}{\mathrm{d}\ell}\right)^{(k)}(\ell) := \sum_{i=0}^{n_p-1} a^{(k)}_{ij}\ell^i \qquad \text{for } \ell \in [\ell^{(k)}_{j-1}, \ell^{(k)}_j) ,$$

where the coefficient $(a^{(k)}_{ij})_{i=0}^{n_p-1}$ come from some n_p-order polynomial regression procedure, for $j = 1, \ldots, J^{(k)}$. As in standard MLMC, one needs to make sure that the consecutive increments cancel properly; therefore, the fit procedure must be such that the polynomials $Q^{(k)}(\ell)$ coincide at the interval extremes $(\ell^{(k)}_j)_{j=2}^{J^{(k)}-1}$, i.e. $Q^{(k)}(\ell)$ is a continuous function.

As in Subsection 3.1, it can be shown that the resulting CLMC estimator is given by

$$\widehat{Q}^{\text{CLMC}}_{L_{\max}} = \frac{1}{N}\sum_{k=1}^{N}\sum_{j=1}^{J^{(k)}}\sum_{i=0}^{n_p-1} a^{(k)}_{ij} \sum_{m=0}^{i}(-1)^m \frac{i^m}{r^{m+1}} \left(\left(\tilde{\ell}^{(k)}_j\right)^{i-m} e^{r\tilde{\ell}^{(k)}_j} - \left(\ell^{(k)}_{j-1}\right)^{i-m} e^{r\ell^{(k)}_{j-1}} \right) , \tag{12}$$

where $\tilde{\ell}^{(k)}_j$ is defined as in (10). Note that, when the the regression polynomial is a simple piecewise linear interpolation polynomial, the CLMC estimator (12) reduces to (9).

3.3.2 Quadrature and higher-order differences

It is also possible to derive alternative practical methods from the fundamental CLMC equation in (5), by using alternative approximations of the integral and the derivative. In order to simplify the presentation, let us assume L to be constant.

Standard MLMC can be interpreted as an estimator for the right hand side of (5) that uses a backward rectangular quadrature rule on a uniform mesh[1] with the derivative approximated by a backward finite difference. This choice of quadrature rule and finite difference approximation is special, because it is in fact exact for this simple case. However, in general one could also pick other schemes, perhaps exploiting more points and therefore catching more global information, at the price of introducing a correction term for both of the extremes of the interval $[0, L]$ that will also need to be estimated (this will be made clearer in the example below). In particular, it is possible to come up with finite difference schemes which provide better variance reduction than the standard differences in MLMC.

Here, we just give a single example to make the basic idea clearer. For sake of notation, we will denote the approximation terms with the level as subscript rather than as argument.

MLMC exploits the following approximation of the derivative:

$$\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell} \approx \frac{Q_\ell - Q_{\ell-h}}{h}\,, \tag{13}$$

for some $h > 0$. Another possible derivative approximation scheme is given by the five-point stencil formula:

$$\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell} \approx \frac{Q_{\ell-2h} - 8Q_{\ell-h} + 8Q_{\ell+h} - Q_{\ell+2h}}{12h}\,. \tag{14}$$

Let us call

$$v := \lim_{\ell\to\infty} \mathbb{V}[Q_\ell]\,, \qquad c := \lim_{\ell\to\infty} \mathrm{Cov}\big(Q_\ell, Q_{\ell+h}\big)\,.$$

Then, in the limit $\ell \to \infty$, with the derivative approximation in (13) we have

$$\mathbb{V}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right] \approx \mathbb{V}\left[\frac{Q_\ell - Q_{\ell-h}}{h}\right] \to \frac{2}{h^2}(v-c)\,,$$

whereas with the derivative approximation in (14) we have

$$\mathbb{V}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right] \approx \mathbb{V}\left[\frac{Q_{\ell-2h} - 8Q_{\ell-h} + 8Q_{\ell+h} - Q_{\ell+2h}}{12h}\right] \to \frac{130}{144h^2}(v-c)\,.$$

This shows that, for ℓ big enough, the five-point stencil formula provides more than double the variance reduction with respect to the scheme used by MLMC.

In general, it can be shown that since the coefficients of any finite difference derivative approximation have to sum up to 0, the variance of the related estimator can always be asymptotically written as some constant times $v - c$. This guarantees that, for any of these approximation schemes, the variance decreases to 0 as the covariance increases.

A practical formula for the five-point stencil CLMC method can be written as

$$\mathbb{E}[Q(L)] = \mathbb{E}[Q_{2h}] + \frac{1}{12}\sum_{i=2}^{M-1} \mathbb{E}[Q_{(i-2)h} - 8Q_{(i-1)h} + 8Q_{(i+1)h} - Q_{(i+2)h}] + \mathbb{E}[\Delta_0] + \mathbb{E}[\Delta_L]\,,$$

where $h = L/M$, for some $M \in \mathbb{N}$, and Δ_0 and Δ_L are the correction terms at Level 0 and L, respectively. They can be written as

$$\begin{aligned}\Delta_0 &= \frac{1}{12}(-Q_0 + 7Q_h - 5Q_{2h} - Q_{3h})\,,\\ \Delta_L &= \frac{1}{12}(Q_{(M-2)h} - 7Q_{(M-1)h} + 5Q_{Mh} + Q_{(M+1)h})\,.\end{aligned}$$

[1] For any integrable function on $(0, L)$, this is defined as $\int_0^L f(\ell)\,d\ell \approx h\sum_{i=0}^{M-1} f(ih)$, where $h = L/M$ and $M \in \mathbb{N}$.

Note that, by using again the asymptotic argument given before, we have $\mathbb{V}[\Delta_L] \to \frac{76}{144}(v-c)$ for $L \to \infty$, which guarantees variance reduction also for the correction term Δ_L. The correction term Δ_0 consists only of coarse approximations and is therefore cheap to compute even if many samples are needed. Note, however, that it corresponds to a finite difference approximation of a derivative at $\ell = 0$ and thus its variance is typically significantly smaller than $\mathbb{V}[Q_0]$.

4 Application to Adaptive Multilevel Monte Carlo

The development of the continuous level framework was motivated by the challenge of integrating sample-wise adaptive finite element solutions within a hierarchical framework. For a given sample, there are significant computational gains to be realised by using goal-oriented (towards the quantity of interest) schemes, particularly when the random field or quantity of interest is localised. The exciting conceptual idea here is in contrast to other adaptive multilevel MC methods [8, 16] we do not use the refinement steps or some pre-defined error tolerances as the levels, but instead use a continuous measure of error in the quantity of interest as our level. This naturally fits within our CLMC framework.

4.1 Subsurface Flow Problem & Constructing Pathwise Adaptive Solutions

We consider a toy-model describing steady state, single phase, incompressible flow in a permeable medium (e.g. rock), given by the linear, scalar elliptic partial differential equation

$$-\nabla \cdot k(\mathbf{x})\nabla u(\mathbf{x}) = f(\mathbf{x}) \quad \forall \mathbf{x} \in D \subset \mathbb{R}^d, \tag{15}$$

subject to suitable boundary conditions. Physically $u(\mathbf{x})$ is the fluid pressure, $f(\mathbf{x})$ the fluid source term and $k(\mathbf{x})$ the scalar permeability field. In practical applications (e.g. in oil reservoir simulation), the permeability field $k(\mathbf{x})$ or the source term $f(\mathbf{x})$ are not known everywhere, therefore a typical approach is to model each as a random field. Let the sample space be denoted by Ω, then the random permeability and source field $k(\mathbf{x}, \omega)$ and $f(\mathbf{x}, \omega)$ belong to $D \times \Omega$ with a certain distribution (inferred from data). Therefore the solution to (15), the unknown pressure field, is also a random field i.e. $u(\mathbf{x}, \omega) \in D \times \Omega$. For simplicity, we shall restrict ourselves to homogeneous Dirichlet conditions $u(\omega, \cdot) \equiv 0$ on the domain boundary ∂D.

For a fixed $\omega \in \Omega$ we can recast (15) as a standard variational problem, i.e. find $u(\mathbf{x}, \omega) \in V := H_0^1(D) = \{v \in H^1(D) : v = 0 \text{ on } \partial D\}$, such that

$$\underbrace{\int_D k(\mathbf{x}, \omega)\nabla u \cdot \nabla v \; d\mathbf{x}}_{=:\, a(\omega;u,v)} = \underbrace{\int_D f(\mathbf{x}, \omega) v \; d\mathbf{x}}_{=:b(\omega;v)}, \qquad \forall v \in V. \tag{16}$$

Here, D is assumed to be a bounded Lipschitz domain and $V = H_0^1(D)$ is the usual Sobolev space of weakly differentiable functions on D. Then, $a(\omega; \cdot, \cdot)$ is a symmetric, bounded and positive-definite bilinear form on $V \times V$, and as such defines an inner product and a norm on V, the so-called energy norm $\|u\|_a := \sqrt{a(u, u)}$. If f is sufficiently smooth, then the functional $b(\omega; \cdot)$ is bounded on V.

To approximate the pressure solution $u(\mathbf{x}, \omega)$, we construct a (sample-wise adapted) finite element (FE) space $V_h(\omega) \subset V$ of piecewise linear Lagrange polynomials on a grid $\mathcal{T}_h(\omega)$ that vanish on the boundary of D. The FE solution $u_h(\mathbf{x}, \omega) \in V_h(\omega)$ satisfies

$$a(\omega; u_h, v_h) = b(\omega; v_h), \qquad \forall v_h \in V_h(\omega), \tag{17}$$

resulting in a (large) linear system of equations of dimension $M_h(\omega) := \dim(V_h(\omega))$. From this, we are interested in approximating statistics (e.g. the expected value) of a *quantity of interest* $\mathcal{Q}$, defined to be (for simplicity) a linear functional of $u_h(\mathbf{x}, \omega)$.

As motivated at the beginning of this section, we are going to build our approximate solutions, sample-by-sample using adaptive finite element methods. But instead of using the number of refinement steps as the level parameter and applying MLMC, we will use a sample-wise error estimate as the level parameter and apply our new CLMC framework.

For any $\omega \in \Omega$, starting with an initial grid $\mathcal{T}^{(0)}(\omega)$, chosen to be the same for each sample, we use an h-adaptive refinement strategy to construct a sequence of grids $\mathcal{T}^{(k)}(\omega)$ for $k = 0, \ldots, K$. In our case, the adaptive procedure is driven by a local, *goal-orientated* error indicator $e_\tau^{(k)}(\omega)$, for each $\tau \in \mathcal{T}^{(k)}(\omega)$. This gives the relative contribution from each element to the error in the quantity of interest $\mathcal{Q}(u(\omega))$, so that

$$|\mathcal{Q}(u(\omega)) - \mathcal{Q}(u^{(k)}(\omega))| \leq e^{(k)}(\omega) = \left(\sum_{\tau \in \mathcal{T}^{(k)}(\omega)} e_\tau^{(k)}(\omega) \right)^{1/2}. \tag{18}$$

In addition to solving (17) (the so-called *primal problem*), goal-oriented error estimators typically also require an approximate FE solution $w_h(\omega, \mathbf{x})$ of the *dual problem*

$$a(\omega; v_h, w_h) = \mathcal{Q}(v_h) \quad \forall v_h \in V_h. \tag{19}$$

There are many different choices of goal-oriented error estimators, see for example [12]. For one particular choice, described in detail in [12], the error estimator $e_\tau^{(k)}(\omega)$ in each element $\tau \in \mathcal{T}^{(k)}$ is computed by bounding the product of the energy norms of the errors in the primal and dual FE solutions $u_h(\omega, \mathbf{x})$ and $w_h(\omega, \mathbf{x})$ of (17) and (19), respectively. Up to a sample-dependent constant, these bounds are simply the sum of the element residuals and of the jumps/discontinuities in inter-element fluxes for each of the two problems. Full details can be found in [12], but we will also provide some more details in Appendix B.

The FE grid $\mathcal{T}^{(k+1)}(\omega)$ is generated by refining the $\theta^{(k)}$ percent of elements of $\mathcal{T}^{(k)}(\omega)$ that contribute most to the error in $\mathcal{Q}$ as defined by (18). This is typically followed by some additional refinements that ensure that the FE space $V^{(k+1)}(\omega)$ is conforming, i.e. that there are no hanging nodes in $\mathcal{T}^{(k+1)}(\omega)$. In our numerical experiments below, we increase $\theta^{(k)}$ as k increases and use a so-called *red/green refinement* strategy that ensures conformity.

Finally, we now define our sample-wise *continuous level* at refinement step k to be

$$\ell_k(\omega) = -\log \left(\frac{e^{(k)}(\omega)}{e^{(0)}(\omega)} \right) \tag{20}$$

The level gives a sample-wise measure of the error in $Q_k(\omega)$, the quantity of interest computed on $\mathcal{T}^{(k+1)}(\omega)$, relative to the error on the coarsest grid. We note that with this choice, computations on $\mathcal{T}^{(0)}$ are naturally providing values $Q_0(\omega)$ at level $\ell_0(\omega) = 0$. However, the main reason for defining the error in this way is due to the explicit error estimator that are being used being only known up to an unknown constant (dependent on ω).

4.2 Numerical Experiments

All the numerical experiments are calculated using the high performance FE library DUNE [1] and its discretisation module `dune-pdelab`. Simulations are carried out on a computer consisting of four, 8-core Intel Xeon E5-4627v2 Ivybridge processors, each running at 1.2 GHz, giving a total of 32 available cores. The solutions for each sample are computed on a single processor and independent samples are equally distributed across all available cores. Individual solutions of the forward and dual problems are obtained using the sparse direct solver `UMFPACK` [6]. Each adaptive step uses the *red/green refinement* strategy, as implemented in `dune-grid` [2], refining $\theta^{(k)}$ percent of elements from $\mathcal{T}^{(k)}$ to $\mathcal{T}^{(k+1)}$.

In our numerical test, we consider $D := [0,1]^2$. The coarse grid $\mathcal{T}^{(0)}$ for all samples is taken as a uniform 32×32 triangular mesh on D. In our test we consider (15) with random permeability field k and random source term f. The permeability field $k(\mathbf{x}, \omega)$ is characterised by a log-normal random field, where $\log k(\mathbf{x}, \omega)$ has a mean of zero and a two-point exponential covariance function

$$C(\mathbf{x}, \mathbf{y}) := \exp\left(-3\, \|\mathbf{x} - \mathbf{y}\|_1\right) \quad \mathbf{x}, \mathbf{y} \in D, \tag{21}$$

with $\|\cdot\|_p$ denoting the ℓ_p-norm in $\mathbb{R}^2$. The field is parameterised with a (truncated) Karhunen-Loève (KL) expansion

$$k(\mathbf{x}, \omega) = \exp\left(\sum_{i=1}^{R} \sqrt{\mu_i}\phi_i(\mathbf{x})\xi_i\right). \tag{22}$$

where $\{\mu_i\}_{i\in\mathbb{N}}$ are the eigenvalues, $\{\phi_i(\mathbf{x})\}_{i\in\mathbb{N}}$ the corresponding L_2-normalised eigenfunctions of the covariance operator with kernel function $C(\mathbf{x}, \mathbf{y})$ and $\xi_i \sim \mathcal{N}(0,1)$. For more details on how this expansion is constructed see for example [4]. In the calculations which follow we take $R = 36$. For the random source term, we take

$$f(\mathbf{x}, \omega) = 1000\, a\ \exp\left(-20\|\mathbf{x} - \mathbf{y}_f\|_2^2\right) \tag{23}$$

where a and the components of $\mathbf{y}_f$ are all sampled from $\mathcal{U}(0,1)$.

As the quantity of interest, we consider the average pressure near $\mathbf{y}_Q := [0.25, 0.25]^T$, defined by the linear functional

$$\mathcal{Q}(u) := C_1 \int_D \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}_Q\|_2^2}{\lambda_Q}\right) u(\mathbf{x}, \omega) d\mathbf{x}, \tag{24}$$

with $\lambda_Q = 0.0005$ and $C_1 = \left(\int_D \exp(-\|\mathbf{x} - \mathbf{y}_Q\|_2^2/\lambda_Q) d\mathbf{x}\right)^{-1} \approx 0.00157$.

We now test our CLMC algorithm (Algorithm 1) by comparing uniform refinements and adaptive refinements with a variable $\theta^{(k)}$ (percentage of elements refined per step). In particular, we choose

$$\theta^{(k)} = \min(100\%, \delta^k \theta_0) \tag{25}$$

as the percentage of elements refined in $\mathcal{T}^{(k)}$, with $\theta_0 = 1\%$ and $\delta = 3$. We note that this choice is heuristic, motivated by a series of test runs. For the problem at hand, the idea of starting with small $\theta^{(0)}$ and increasing the percentage with the number of adaptive steps makes sense. Initially the error in $\mathcal{Q}$ is dominate by the fact that the grid is not well adapted to the particular random sample $\omega \in \Omega$. This includes the random field, the location of the localised source and the quantity of interest itself. Once the adaptive strategy has focused in on all those localised regions, the error in $\mathcal{Q}$ is governed by the global lack of singularity in the coefficient [3, 23] and thus distributed fairly uniformly across the whole domain. So from that point onwards, refining all elements uniformly leads to the most effective error reduction.

Before running a complete simulation we first consider a single sample $\omega \in \Omega$. Figure 1 shows the random permeability field $k(\mathbf{x}, \omega)$, pressure solution $u_h(\mathbf{x}, \omega)$, and the influence function $w_h(\mathbf{x}, \omega)$ (i.e. the solution of the dual problem (19)) for this sample after 6 adaptive steps. Snapshots of the grids, built using the goal-oriented error estimator, are shown in Figure 2 at steps 0, 2, 4 and 6. Visually, we see that the adaptive scheme is working correctly, refining near $\mathbf{y}_Q = [0.25, 0.25]^T$, the point around which the pressure is averaged in the functional $\mathcal{Q}$ in (24), whilst also adapting around the localised source. At the latter levels the refinement also starts to pick up local variations in the permeability field in regions that influence the pressure at the point of interest.

For the uniform and adaptive strategy, we first run an initial batch of 6400 samples up to $L_{\max} = 5$, in order to estimate the parameters β and γ. In a real simulation, it would not be necessary to estimate these parameters accurately and so significantly fewer samples could

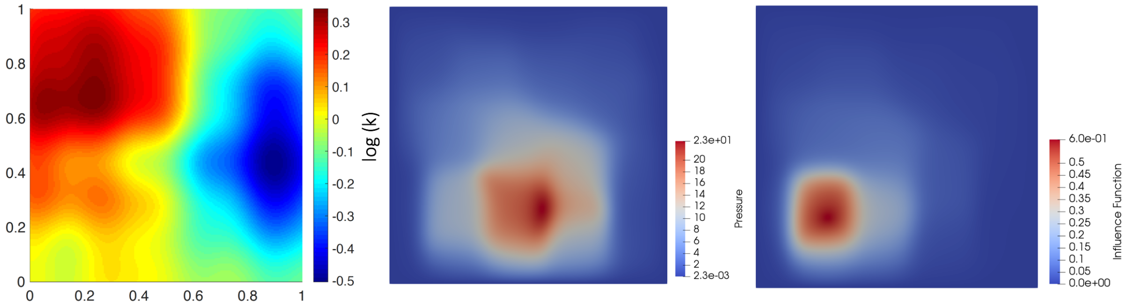
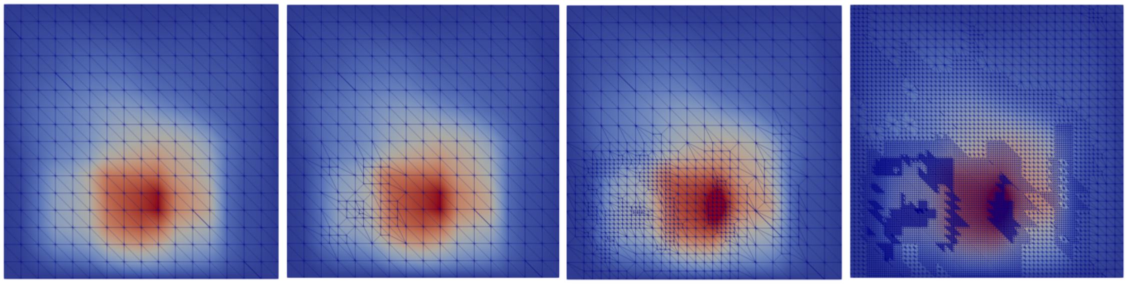


Figure 1: Permeability field k, pressure solution u_h and influence function w_h on the finest adaptive grid ($k = 6$) for a particular realisation $\omega \in \Omega$.



$\ell_0 = 0.0$ $\quad$ $\ell_2 = 0.4808$ $\quad$ $\ell_4 = 1.2582$ $\quad$ $\ell_6 = 3.2541$

Figure 2: Sequence of adaptive grids built using goal-oriented error estimator for random $\omega \in \Omega$, level is defined by ℓ_k given by (20).

be used. With uniform refinements, our estimates are $\beta_u = 2.28$ and $\gamma_u = 1.0$, whereas for adaptive refinements we get $\beta_a = 2.22$ and $\gamma_a = 0.78$. Note that, in both cases, $\beta > \gamma$, therefore by taking $L_{\max} = +\infty$ in the CLMC setting we obtain unbiased estimators with respect to $\mathbb{E}[\mathcal{Q} - Q(0)]$. In these initial runs we can already see the expected computational gains of adaptive grid refinement. We note that the rates β for $\mathbb{V}[dQ/d\ell]$ are much the same in each case, whilst γ, the rate of growth of the expected cost per sample, is clearly smaller for the adaptive strategy. Figure 3 gives a plot of the continuous level ℓ, representing the estimate of the relative finite element error, against the natural log of the cost for all samples, which shows the better rate for the adaptive scheme.

We then run the CLMC algorithm with a maximum of $N = 10^6$ samples for each case. The exponential parameter rate r is taken to be the same for each case, so that any computational gains can be attributed to the adaptive strategy, rather than a difference in r. The value is chosen so that $r = \frac{1}{2}(r_u + r_a) = \frac{1}{4}(\beta_u + \gamma_u + \beta_a + \gamma_a) = 1.57$, and we consider the unbiased estimator with $L_{\max} = +\infty$.

The numerical results show that the CLMC algorithm is working as expected. In Figure 4 (left), we observe as expected that the natural logarithm of $\mathbb{E}[dQ/d\ell]$ decreases linearly with ℓ, i.e. $\alpha \approx 1$, in both the uniform and the adaptive case, since ℓ is defined as the natural logarithm of an estimate of the relative bias error. Figure 4 (middle) shows the variance reduction for both uniform and adaptive refinement strategies. Both decay very similarly across the levels with rates of around $\beta = 2$. Finally, Figure 4 (right) shows the actual cost to compute the estimate for different choices of N. The cost (in seconds) is plotted against the root mean square error, which is equal to the sampling error, since the estimator is unbiased. As proved in Theorem 2.3, since $\beta > \gamma$ for both strategies, we observe parallel straight lines with rate of ≈ 2. Due to the reduced computational cost on the finer levels, the adaptive strategy wins over the uniform one

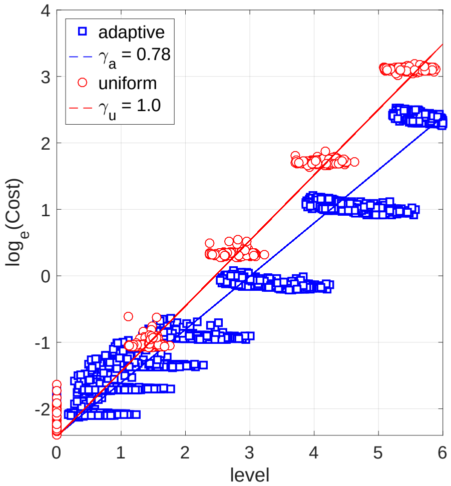


Figure 3: Level ℓ against log(Cost) for 6400 uniform (red circles) and 6400 adaptive samples (blue squares).

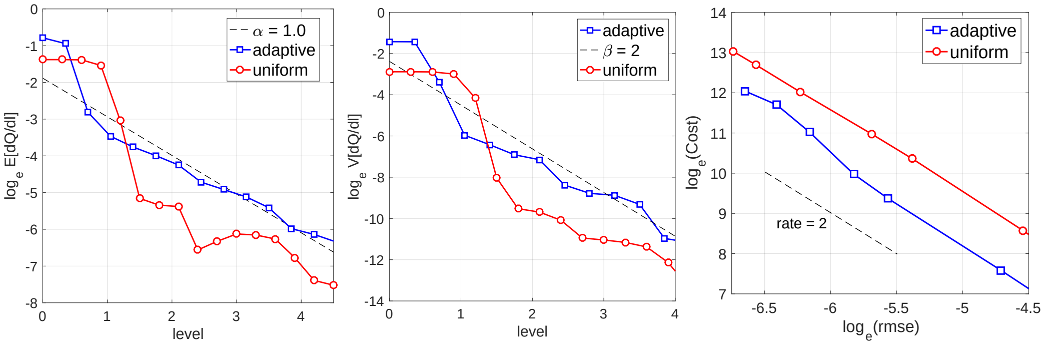


Figure 4: Results for the numerical test, in log-scale. Left and middle: Convergence plots of $\mathbb{E}[dQ/d\ell]$ and $\mathbb{V}[dQ/d\ell]$ against ℓ respectively. Right: Total cost of uniform and adaptive algorithm (in seconds) against estimated sampling error (= root mean square error due to unbiasedness).

across a range of tolerances. Especially for coarser tolerances the gains are significant and the sample-adaptive level hierarchy consistently reduces the cost by a factor of 4.

The actual gains that are possible with the new CLMC estimator and with sample-adaptive level hierarchies are very problem dependent. They also depend strongly on the error estimator and on the adaptive refinement strategy. The estimator and the strategy employed here are by no means optimal. It is known that the employed error estimator is not necessarily very effective in the context of strong coefficient variations. Finally, the gains also depend on the cost of the linear solver. For a fair comparison, we used a sparse direct solver, which outperforms iterative solvers for the problem sizes encountered in our 2D model problem. However, further experiments in three space dimensions will require iterative solvers and robust preconditioners that can cope both with the strong coefficient variations and with the locally refined finite element meshes. The cost and the memory requirements of sparse direct solvers grow too rapidly in 3D. Nevertheless, we expect the gains in 3D to be even more significant.

5 Conclusions & Further Work

In this paper, we introduce Continuous Level Monte Carlo (CLMC), a generalisation of MLMC to a continuous framework where the level is a continuous variable rather than an integer. We propose a practical estimator and prove a Complexity Theorem, showing the same order of convergence as in MLMC. Furthermore, we provide a version of the estimator that is unbiased with respect to the true quantity of interest and extend the Complexity Theorem to this case, giving sufficient and necessary conditions for the unbiased estimator to have finite cost. We apply CLMC to adaptive refinement schemes, where the continuous framework is particularly well suited in order to capture sample-based level hierarchies. We demonstrate clear computational gains when adaptive refinement strategies are adopted rather than uniform ones.

The introduction of CLMC opens the door to several new research directions. We outline a few ideas for further work:

Extension of Multi-Index Monte Carlo (MIMC) [13]. MIMC is an extension of MLMC to multi-dimensional level parameters and higher-order differences. In the same way, as CLMC generalises MLMC by replacing the sum with an integral and the difference with a derivative in the case of a scalar level parameter, one could generalise MIMC by employing multi-dimensional integrals of partial derivatives. Indeed, consider $(Q(\boldsymbol{\ell}))_{\boldsymbol{\ell}}$ to be a sequence of approximation functions of $\mathcal{Q}$, where $\boldsymbol{\ell} = (\ell_1, \dots \ell_m)$ is a m-dimensional vector of non-negative levels. To explain the idea, let us restrict our description to $m = 2$ and consider a 2-dimensional positive random variable $\boldsymbol{L} = (L_1, L_2)$. Assuming sufficient regularity, we can write

$$\mathbb{E}\big[Q(\boldsymbol{L}) - Q(\boldsymbol{0})\big] = \mathbb{E}\left[\int_0^{L_1}\!\!\int_0^{L_2} \frac{\partial^2 Q(\boldsymbol{\ell})}{\partial \ell_1 \partial \ell_2}\, d\boldsymbol{\ell}\right] \;+\; \sum_{j=1}^{2} \mathbb{E}\left[\int_0^{L_j} \frac{\partial Q(\boldsymbol{\ell})}{\partial \ell_j}\, d\ell_j\right]. \tag{26}$$

Note that (26) is a two-dimensional extension of the formula in (5). It is outside the scope of this paper, but we argue that different choices for the probability distribution of the vector of finest levels $\boldsymbol{L}$ (with potentially correlated components) correspond to different choices of the grid of levels in MIMC. A natural choice would be again to pick independent $L_i \sim \text{Exponential}(r_i)$, for $i = 1, \dots, m$, with $r_i > 0$. Classically, in MIMC, $\boldsymbol{L}$ is a fixed integer vector chosen to control the bias error, while the optimal strategy for the choice of samples avoids computation of samples for levels with $\ell_1/L_1 + \ell_2/L_2 > 1$. Here, the bias can again be completely eliminated (provided the variance decays fast enough w.r.t. the growth in cost), and the optimal strategy is a direct consequence of the choice of the exponential distributions for L_1 and L_2, making the probability that both ℓ_1 and ℓ_2 are simultaneously large practically zero.

Extension of Multilevel Monte Carlo Markov Chain (MLMCMC) [7]. Multilevel techniques have been successfully applied to sampling algorithms like MCMC, drastically reducing their complexity cost. The extension of MLMCMC to Continuous Level MCMC is object of future work, potentially leading to an estimator that is unbiased with respect to the real quantity of interest, under the real target probability distribution. Such an unbiased estimator would be of great interest: unlike forward problems, where the bias can arise only from the approximation of the quantity of interest, inverse problems have the additional issue of an approximation of the target probability distribution. Unbiasedness guarantees that the estimator is in fact estimating the correct unknown, without expensive extra computational cost to estimate the bias error. In addition, continuous level adaptive refinement strategies will significantly help to slim down MCMC's computational cost, allowing to solve even more complex problems.

A Proof of the Complexity results

A.1 Proof of Theorem 2.3

Proof. First, we want to bound the MSE by ε^2. By the bias-variance decomposition, this can be achieved by bounding both the squared bias and variance by $\varepsilon^2/2$.

By using assumption *(i)* and recalling that $L \sim \text{Exponential}(L)$, the bias term is bounded by

$$\begin{aligned}
\left|\mathbb{E}\left[\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}} - (\mathcal{Q} - Q(0))\right]\right| &= \left|\mathbb{E}\left[\int_{L\wedge L_{\max}}^{L} \frac{1}{\mathbb{P}(L \geq \ell)} \frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\, d\ell\right]\right| \\
&\leq \mathbb{E}\left[\int_{L\wedge L_{\max}}^{L} \frac{1}{\mathbb{P}(L \geq \ell)} \left|\mathbb{E}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]\right| d\ell\right] \\
&\leq c_1 \mathbb{E}\left[\int_{L\wedge L_{\max}}^{L} \frac{1}{\mathbb{P}(L \geq \ell)} e^{-\alpha\ell}\, d\ell\right] \\
&= \begin{cases} \frac{c_1}{r-\alpha}\mathbb{E}\left[e^{(r-\alpha)L} - e^{(r-\alpha)L\wedge L_{\max}}\right] & \text{if } r \neq \alpha \\ c_1\mathbb{E}[L - L\wedge L_{\max}] & \text{if } r = \alpha \end{cases} \qquad (27) \\
&= \frac{c_1}{\alpha} e^{-\alpha L_{\max}}\,, \qquad (28)
\end{aligned}$$

where we can explicitly compute the expected values in (27) using the distribution of L.

As we want to bound the squared bias by $\varepsilon^2/2$, this is equivalent to bounding the bias by $\varepsilon/\sqrt{2}$, which can be achieved by setting

$$L_{\max} \geq \left\lceil \frac{1}{\alpha} \log \frac{\sqrt{2}c_1 r \varepsilon^{-1}}{\alpha} \right\rceil . \qquad (29)$$

Then, let us provide an upper bound for the variance of the CLMC estimator (6). By the law of total variance, we have

$$\mathbb{V}[\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}] = \mathbb{E}\left[\mathbb{V}[\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}|L]\right] + \mathbb{V}\left[\mathbb{E}[\widehat{Q}_{L_{\max}}^{\mathrm{CLMC}}|L]\right] . \qquad (30)$$

Let us start by bounding the first term on the right-hand-side of (30). We will use Cauchy-

Schwarz inequality on the covariance, followed by assumption *(ii)*. We have

$$
\begin{aligned}
\mathbb{E}\left[\mathbb{V}[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}|L]\right] &= \mathbb{E}\left[\mathrm{Cov}\left(\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}, \widehat{Q}^{\mathrm{CLMC}}_{L_{\max}} \mid L\right)\right] \\
&= \frac{1}{N}\mathbb{E}\left[\int_{[0,L\wedge L_{\max}]^2} \frac{1}{\mathbb{P}(L \geq \ell)}\frac{1}{\mathbb{P}(L \geq \ell')}\mathrm{Cov}\left(\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}, \frac{\mathrm{d}Q(\ell')}{\mathrm{d}\ell'}\right)\mathrm{d}\ell\mathrm{d}\ell'\right] \\
&= \frac{1}{N}\mathbb{E}\left[\int_{[0,L\wedge L_{\max}]^2} \frac{1}{\mathbb{P}(L \geq \ell)}\frac{1}{\mathbb{P}(L \geq \ell')}\mathbb{V}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]^{\frac{1}{2}}\mathbb{V}\left[\frac{\mathrm{d}Q(\ell')}{\mathrm{d}\ell'}\right]^{\frac{1}{2}}\mathrm{d}\ell\mathrm{d}\ell'\right] \\
&= \frac{1}{N}\mathbb{E}\left[\left(\int_0^{L\wedge L_{\max}} \frac{1}{\mathbb{P}(L \geq \ell)}\mathbb{V}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]^{\frac{1}{2}}\mathrm{d}\ell\right)^2\right] \\
&\leq \frac{1}{N}c_2^2\mathbb{E}\left[\left(\int_0^{L\wedge L_{\max}} \frac{1}{\mathbb{P}(L \geq \ell)}e^{-\frac{\beta}{2}\ell}\,\mathrm{d}\ell\right)^2\right] \\
&= \begin{cases} \frac{1}{N}\frac{4c_2^2}{(2r-\beta)^2}\mathbb{E}\left[\left(e^{(r-\frac{\beta}{2})L\wedge L_{\max}} - 1\right)^2\right] & \text{if } r \neq \beta/2 \\ \frac{1}{N}c_2^2\mathbb{E}[(L \wedge L_{\max})^2] & \text{if } r = \beta/2 \end{cases} \\
&\leq \begin{cases} \frac{1}{N}\frac{4c_2^2}{(r-\beta)(2r-\beta)^2}\left((2r-\beta)e^{(r-\beta)L_{\max}} - \beta\right) & \text{if } r \neq \beta/2, \beta \\ \frac{1}{N}\frac{4c_2^2}{\beta^2}(\beta L_{\max} + 1) & \text{if } r = \beta \\ \frac{1}{N}\frac{8c_2^2}{\beta^2} & \text{if } r = \beta/2\,. \end{cases}
\end{aligned}
$$

On the other hand, the second term on the right-hand-side of (30) can be bounded as

$$
\begin{aligned}
\mathbb{V}\left[\mathbb{E}[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}|L]\right] &= \frac{1}{N}\mathbb{V}\left[\int_0^{L\wedge L_{\max}} \frac{1}{\mathbb{P}(L \geq \ell)}\mathbb{E}\left[\frac{\mathrm{d}Q(\ell)}{\mathrm{d}\ell}\right]\mathrm{d}\ell\right] \\
&\leq \frac{1}{N}c_1^2\mathbb{V}\left[\int_0^{L\wedge L_{\max}} \frac{1}{\mathbb{P}(L \geq \ell)}e^{-\alpha\ell}\,\mathrm{d}\ell\right] \\
&= \begin{cases} \frac{1}{N}\frac{c_1^2}{(r-\alpha)^2}\mathbb{V}\left[e^{(r-\alpha)L\wedge L_{\max}} - 1\right] & \text{if } r \neq \alpha \\ \frac{1}{N}c_1^2\mathbb{V}\left[L \wedge L_{\max}\right] & \text{if } r = \alpha \end{cases} \\
&\leq \begin{cases} \frac{1}{N}\frac{c_1^2}{(r-\alpha)^2}\mathbb{E}\left[e^{2(r-\alpha)L\wedge L_{\max}}\right] & \text{if } r \neq \alpha \\ \frac{1}{N}c_1^2\mathbb{V}\left[L\right] & \text{if } r = \alpha \end{cases} \\
&= \begin{cases} \frac{1}{N}\frac{c_1^2}{(r-2\alpha)(r-\alpha)^2}\left(2(r-\alpha)e^{(r-2\alpha)L_{\max}} - r\right) & \text{if } r \neq \alpha, 2\alpha \\ \frac{1}{N}\frac{2c_1^2}{\alpha}L_{\max} & \text{if } r = 2\alpha \\ \frac{1}{N}\frac{c_1^2}{\alpha^2} & \text{if } r = \alpha\,. \end{cases}
\end{aligned}
$$

In both cases in the last step, we have again used our knowledge of the distribution of L.

Note that asymptotically the bound for the first term on the right-hand-side of (30) always dominates the bound of the second, since we have assumed that $\beta \leq 2\alpha$. Hence, adding together the two bounds and using (29), as well as the fact that $\varepsilon < e^{-1}$, we obtain the following asymptotic bound on the total variance:

$$
\mathbb{V}[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}] \;\leq\; \frac{C'}{N}\begin{cases} \varepsilon^{\frac{\beta-r}{\alpha}} & \text{if } r > \beta \\ \log\varepsilon & \text{if } r = \beta \\ 1 & \text{if } r < \beta\,, \end{cases}
$$

for some constant $C' > 0$ that is independent of N and ε. Thus, to guarantee $\mathbb{V}[\widehat{Q}^{\mathrm{CLMC}}_{L_{\max}}] \leq \varepsilon^2/2$ it suffices to choose

$$N \geq 2C' \varepsilon^{-2-\max(0,\frac{r-\beta}{\alpha})} (\log \varepsilon)^{\delta_{r,\beta}} \,, \tag{31}$$

where δ denotes the Kronecker delta.

Finally, we can bound the expected overall cost:

$$\begin{aligned}
\mathcal{C}^{\mathrm{CLMC}}_{L_{\max}} &= N\mathbb{E}\left[\int_0^{L\wedge L_{\max}} \mathcal{C}(\ell)\, d\ell\right] \\
&= N\int_0^{L_{\max}} \mathcal{C}(\ell)\mathbb{P}(L \geq \ell)\, d\ell \\
&\leq Nc_3 \int_0^{L_{\max}} e^{\gamma\ell}\, \mathbb{P}(L \geq \ell)\, d\ell \\
&= \begin{cases} N\frac{c_3}{\gamma-r}\left(e^{(\gamma-r)L_{\max}} - 1\right) & \text{if } r \neq \gamma \\ Nc_3\gamma L_{\max} & \text{if } r = \gamma\,. \end{cases}
\end{aligned} \tag{32}$$

Hence, using (31) the overall cost can be bounded as

$$\mathcal{C}^{\mathrm{CLMC}}_{L_{\max}} \leq C\, \varepsilon^{-2-\max(0,\frac{r-\beta}{\alpha})-\max(0,\frac{\gamma-r}{\alpha})} (\log \varepsilon)^{\delta_{r,\beta}+\delta_{r,\gamma}} \,, \tag{33}$$

for some constant $C > 0$, which is again independent of ε. This completes the proof since we had assumed that $r \in [\min(\beta,\gamma), \max(\beta,\gamma)]$ and so $\max(0,\frac{r-\beta}{\alpha}) + \max(0,\frac{\gamma-r}{\alpha}) = \max(0,\frac{\gamma-\beta}{\alpha})$.
□

A.2 Proof of Corollary 2.4

Proof. To prove (a), suppose $L_{\max} = +\infty$. Then, the bias in (28) is zero due to Corollary 2.2, so that the MSE is equivalent to the variance of the CLMC estimator. Since $r < \beta$ it follows as in the proof of Theorem 2.3 in Section A.1, that

$$\mathbb{V}\left[\widehat{Q}^{\mathrm{CLMC}}_{\infty}\right] \leq \frac{C'}{N} \,,$$

for some constant $C' > 0$. Analogously, since $r > \gamma$, the expected overall cost can be bounded by

$$\mathcal{C}^{\mathrm{CLMC}}_{\infty} \leq C'' N \,,$$

for some constant $C'' > 0$. Therefore, we can bound the MSE with ε^2 by taking $N \geq C'\varepsilon^{-2}$ and the overall computational cost is $\mathcal{C}^{\mathrm{CLMC}}_{\infty} = \mathcal{O}\left(\varepsilon^{-2}\right)$.

To prove (b), suppose that the additional assumptions in part (*b*) of Corollary 2.4 hold. Then, by tracking back the steps in the proof of Theorem 2.3 in Section A.1, it can be seen fairly easily that for $\beta \leq \eta \leq \gamma$ we have

$$\mathbb{E}\left[\mathbb{V}[\widehat{Q}^{\mathrm{CLMC}}_{\infty}|L]\right] \geq \begin{cases} \frac{1}{N}\frac{4c_2'^2}{\eta(\eta-r)} & \text{if } r < \eta, r \neq \eta/2 \\ \frac{1}{N}\frac{8c_2'^2}{\eta^2} & \text{if } r = \eta/2 \\ +\infty & \text{if } r \geq \eta\,, \end{cases} \quad \text{and} \quad \mathcal{C}^{\mathrm{CLMC}}_{\infty} \geq \begin{cases} N\frac{c_3'}{r-\eta} & \text{if } r > \eta \\ +\infty & \text{if } r \leq \eta\,. \end{cases}$$

We see that MSE × $\mathcal{C}^{\mathrm{CLMC}}_{\infty} = +\infty$ for all choices of r.

□

B Goal-Oriented Error Estimators

We use a classical goal-oriented error estimator to drive the sample-wise adaptive scheme in our numerical experiments. The following description is taken from [12]. Let $\omega \in \Omega$ be fixed, and recall that $u \in V$ denotes the solution of (16) whilst $u_h \in V_h \subset V$ is its finite element approximation on a grid $\mathcal{T}_h$. The error in a quantity of interest (defined by a linear functional[2]) is given by

$$\mathcal{Q}(\epsilon_h) = \mathcal{Q}(u - u_h) = \mathcal{Q}(u) - Q(u_h). \tag{34}$$

This functional can be interpreted as the 'source' of the finite element discretisation error in the quantity of interest, and is a bounded linear functional on the dual space V'. The key idea of goal-oriented, a posteriori error estimators is to relate $\mathcal{Q}(\epsilon_h)$ to the solution residual r_h^u, i.e we seek a function $w \in V''$ such that $\mathcal{Q}(\epsilon_h) = w(r_h^u)$. Since V is a reflexive Hilbert Space, there exists a $w \in V$ such that $\mathcal{Q}(\epsilon_h) = r_h^u(w)$. The function w, termed the *influence function*, is the solution of the *dual problem*

$$a(v, w) = \mathcal{Q}(v) \quad \forall v \in V. \tag{35}$$

This dual solution can be approximate using the same finite element approximation as u_h, i.e. find $w_h \in V_h \subset V$ s.t

$$a(v_h, w_h) = \mathcal{Q}(v_h) \quad \forall v_h \in V_h\,.$$

Using the Galerkin orthogonality of u and u_h, we can bound $\mathcal{Q}(\epsilon_h)$ as follows:

$$\begin{aligned} |\mathcal{Q}(\epsilon_h)| &= |\mathcal{Q}(u - u_h)| = |a(u - u_h, w)| = |a(u - u_h, w)| + |a(u - u_h, w_h)| \\ &= |a(u - u_h, w - w_h)| \le \sum\nolimits_{\tau \in \mathcal{T}_h} \|u - u_h\|_{a,\tau} \|w - w_h\|_{a,\tau}\,. \end{aligned} \tag{36}$$

In the last step, we have used the Cauchy-Schwarz inequality elementwise. Hence, the product of energy norms $\|u - u_h\|_{a,\tau}\|w - w_h\|_{a,\tau}$ provides an estimate for the element-wise contribution to the error in $Q(u_h)$. It is now used to define an appropriate adaptivity scheme.

To estimate the error of the solutions of the primal and dual problem in the energy norm on each element τ, we use explicit error estimators. We only show the main ideas for estimating $\|u - u_h\|_{a,\tau}$ using one of the most basic estimators. The bound for $\|w - w_h\|_{a,\tau}$ can be derived analogously. On each element τ, using integration by parts, the FE error can be represented as

$$\begin{aligned} a(\epsilon_h, v)|_\tau &= \int_\tau f v \,\mathrm{d}\mathbf{x} - \int_\tau \nabla u_h \cdot k(\mathbf{x}) \nabla v \,\mathrm{d}\mathbf{x} \\ &= \int_\tau \mathcal{R}_u v \,\mathrm{d}\mathbf{x} + \int_{\partial\tau} \mathcal{J}_u v \,\mathrm{d}s \quad \forall v \in V\,, \end{aligned} \tag{37}$$

where the residual error on the element is define by

$$\mathcal{R}_u(\mathbf{x}) = \nabla \cdot \mathbf{k}(\mathbf{x}) \nabla u_h(\mathbf{x}) + f(\mathbf{x}) \quad \forall \mathbf{x} \in \tau, \tag{38}$$

and where $\mathcal{J}_u$ defines, for all $\mathbf{x} \in \partial\tau$ (except at the vertices), the jump of the flux in u_h across the element boundary by

$$\mathcal{J}_u(\mathbf{x}) = \begin{cases} k(\mathbf{x})\Big[\mathbf{n}_\tau(\mathbf{x}) \cdot \nabla u_h|_\tau + \mathbf{n}_{\tau'(\mathbf{x})}(\mathbf{x}) \cdot \nabla u_h|_{\tau'(\mathbf{x})}\Big], & \forall \mathbf{x} \notin \partial D\,, \\ \mathbf{n}_\tau(\mathbf{x}) \cdot k(\mathbf{x}) \nabla u_h|_\tau\,, & \forall \mathbf{x} \in \partial D\,, \end{cases} \tag{39}$$

where $\mathbf{n}_\tau$ is the outward unit normal to the element boundary $\partial\tau$ at $\mathbf{x}$ and $\tau'(\mathbf{x})$ is the neighbouring element of τ at $\mathbf{x}$. For simplicity, we assume that the boundary conditions are homogeneous Dirichlet conditions on all of ∂D.

[2] Similar error estimators can also be obtained for nonlinear functionals by first linearising about ϵ_h.

Using again Galerkin orthogonality, we can introduce the global FE interpolant $\mathcal{I}_h v$ in (37), and thus using classical interpolation theory find that

$$\begin{aligned} a(\epsilon_h, v)|_\tau &\leq \|\mathcal{R}_u\|_{L^2(\tau)} \|v - \mathcal{I}_h v\|_{L^2(\tau)} + \|\mathcal{J}_u\|_{L^2(\partial\tau)} \|v - \mathcal{I}_h v\|_{L^2(\partial\tau)} \\ &\leq c_1 \underbrace{\left(h_\tau \|\mathcal{R}_u\|_{L^2(\tau)} + \sqrt{h_\tau} \|\mathcal{J}_u\|_{L^2(\partial\tau)} \right)}_{=: \, \eta_\tau(u_h)} \|v\|_{a,\omega_\tau} \,, \end{aligned}$$

where ω_τ denotes the subdomain of elements sharing a common edge with τ, and where c_1 is problem dependent constant independent of the mesh size h_τ. Substituting $v = \epsilon_h$ and summing over all elements, we can see that (up to a constant factor c_2 depending on the geometry) this leads to the explicit global energy error estimator

$$\|\epsilon_h\|_a \leq c_1 c_2 \left(\sum\nolimits_{\tau \in \mathcal{T}_h} \eta_\tau^2(u_h) \right)^{1/2} \tag{40}$$

for the primal solution on $\mathcal{T}_h$.

The local error contribution $\eta_\tau(w_h)$ to the dual solution w_h on τ in the energy norm can be estimated analogously, and it can be shown that together with (36) this leads to the goal-oriented error estimator

$$|\mathcal{Q}(\epsilon_h)| \leq c_3 \sum\nolimits_{\tau \in \mathcal{T}^{(k)}} \eta_\tau(u_h) \eta_\tau(w_h) \,, \tag{41}$$

which is again explicit up to the unknown constant c_3. Although the exact constants in all the described estimators are not known, the relative error with respect to a coarsest reference mesh can still be used to drive a goal-oriented mesh adaptivity procedure, as described in Section 4.1.

More sophisticated error estimators exist, including estimators where the constants are known or can be computed explicitly (see e.g. [12] for more details), but in our numerical experiments we used the estimator described above.

References

[1] P. Bastian, F. Heimann, and S. Marnach, *Generic implementation of finite element methods in the distributed and unified numerics environment (DUNE)*, Kybernetika 46 (2010), pp. 294–315.

[2] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander, *A generic grid interface for parallel and adaptive scientific computing. Part I: Abstract framework*, Computing 82.2-3 (2008), pp. 103–119.

[3] J. Charrier, R. Scheichl, and A. L. Teckentrup, *Finite element error analysis of elliptic PDEs with random coefficients and its application to multilevel Monte Carlo methods*, SIAM J. Numer. Anal. 51.1 (2013), pp. 322–352.

[4] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Comput. Visual. Sci. 14.1 (2011), pp. 3–15.

[5] G. Da Prato and J. Zabczyk, *Stochastic equations in infinite dimensions*, Cambridge university press, 2014.

[6] T. A. Davis, *Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software (TOMS) 30.2 (2004), pp. 196–199.

[7] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup, *A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow*, SIAM/ASA J. Uncertain. Quant. 3.1 (2015), pp. 1075–1108.

[8] M. Eigel, C. Merdon, and J. Neumann, *An adaptive multilevel Monte Carlo method with stochastic bounds for quantities of interest with uncertain data*, SIAM/ASA J. Uncertain. Quant. 4.1 (2016), pp. 1219–1245.

[9] D. Elfverson, F. Hellman, and A. Målqvist, *A multilevel Monte Carlo method for computing failure probabilities*, SIAM/ASA J. Uncertain. Quant. 4.1 (2016), pp. 312–330.

[10] M. B. Giles, *Multilevel Monte Carlo path simulation*, Oper. Res. 56 (2008), pp. 607–617.

[11] M. B. Giles, *Multilevel Monte Carlo methods*, Acta Numerica 24 (2015), pp. 259–328.

[12] T. Grätsch and K. J. Bathe, *A posteriori error estimation techniques in practical finite element analysis*, Comput. Struct 83 (2005), pp. 235–265.

[13] A. Haji-Ali, F. Nobile, and R. Tempone, *Multi-index Monte Carlo: When sparsity meets sampling*, Numer. Math. 132.4 (2015), 767—806.

[14] S. Heinrich, *Monte Carlo complexity of global solution of integral equations*, J. Complexity 14.2 (1998), pp. 151–175.

[15] H. Hoel, E. Von Schwerin, A. Szepessy, and R. Tempone, *Adaptive multilevel Monte Carlo simulation*, in: *Numerical Analysis of Multiscale Computations*, ed. by B. Engquist, O. Runborg, and Y.-H. R. Tsai, Springer, 2012, pp. 217–234.

[16] R. Kornhuber and E. Youett, *Adaptive multilevel Monte Carlo methods for stochastic variational inequalities*, SIAM Journal on Numerical Analysis 56.4 (2018), pp. 1987–2007.

[17] I. Kossaczký, *A recursive approach to local mesh refinement in two and three dimensions*, Journal of Computational and Applied Mathematics 55.3 (1994), pp. 275–288.

[18] X. Li and J. Liu, *A multilevel approach towards unbiased sampling of random elliptic partial differential equations*, arXiv preprint arXiv:1605.06349 (2016).

[19] D. McLeish, *A general method for debiasing a Monte Carlo estimator*, Monte Carlo Methods Appl. 17.4 (2011), pp. 301–315.

[20] B. Øksendal, *Stochastic Differential Equations, An Introduction with Applicatins*, 5th ed., Springer, Berlin Heidelberg, 2000.

[21] C.-H. Rhee and P. W. Glynn, *Unbiased estimation with square root convergence for SDE models*, Oper. Res. 63.5 (2015), pp. 1026–1043.

[22] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, NY, 2004.

[23] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numer. Math. 125.3 (2013), pp. 569–600.

[24] M. Vihola, *Unbiased estimators and multilevel Monte Carlo*, Operations Research 66.2 (2018), pp. 448–462.

■ Paper II

MULTILEVEL DIMENSION-INDEPENDENT LIKELIHOOD-INFORMED MCMC FOR LARGE-SCALE INVERSE PROBLEMS [*]

TIANGANG CUI[†], GIANLUCA DETOMMASO[‡], AND ROBERT SCHEICHL[§]

Abstract. We present a non-trivial integration of dimension-independent likelihood-informed (DILI) MCMC (Cui, Law, Marzouk, 2016) and the multilevel MCMC (Dodwell et al., 2015) to explore the hierarchy of posterior distributions. This integration offers several advantages: First, DILI-MCMC employs an intrinsic *likelihood-informed subspace* (LIS) (Cui et al., 2014) – which involves a number of forward and adjoint model simulations – to design accelerated operator-weighted proposals. By exploiting the multilevel structure of the discretised parameters and discretised forward models, we design a *Rayleigh-Ritz procedure* to significantly reduce the computational effort in building the LIS and operating with DILI proposals. Second, the resulting DILI-MCMC can drastically improve the sampling efficiency of MCMC at each level, and hence reduce the integration error of the multilevel algorithm for fixed CPU time. To be able to fully exploit the power of multilevel MCMC and to reduce the dependencies of samples on different levels for a parallel implementation, we also suggest a new pooling strategy for allocating computational resources across different levels and constructing Markov chains at higher levels conditioned on those simulated on lower levels. Numerical results confirm the improved computational efficiency of the multilevel DILI approach.

Key words. multilevel Monte Carlo, likelihood-informed subspaces, dimension independent MCMC, inverse problems

AMS subject classifications. 15A29, 65C05, 65C60

1. Introduction.

Inverse problems aim to estimate unknown parameters of mathematical models from noisy and indirect observations. The unknown parameters, often represented as functions, are related to the observed data through a forward model, such as a differential equation, that maps realisations of parameters to observables. Due to smoothing properties of the forward model and incompleteness of data, such inverse problems are often ill-posed: there may exist many feasible realisations of parameters that are consistent with the observed data, and small perturbations in the data may lead to large perturbations in unregularised parameter estimates. The Bayesian approach [35, 24, 34] casts the solution of inverse problems as the posterior probability distribution of the model parameters conditioned on the data. This offers a natural way to integrate the forward model and the data together with prior knowledge and a stochastic description of measurement and/or model errors to remove the ill-posedness and to quantify uncertainties in parameters and parameter-dependent predictions. As a result, parameter estimations, model predictions, and associated uncertainty quantifications can be issued in the form of marginal distributions or expectations of some quantities of interest (QoI) over the posterior. Due to the typically high parameter dimensions and the high computational cost of the forward models, characterising the posterior and computing posterior expectations are in general computationally challenging tasks. Integrating multilevel Markov chain Monte Carlo (MCMC) [21, 13], likelihood-informed parameter reduction [11, 33, 39] and dimension-independent MCMC [4, 8, 10, 31], we present here an integrated framework to significantly accelerate the computation of posterior expectations for large-scale inverse problems.

In inverse problems, unknown parameters are often cast as functions, and hence the Bayesian inference has to be carried out over typically *high-dimensional discretisations* of the parameters that resolve the spatial and/or temporal variability of the underlying problem sufficiently. Examples are the permeability field of a porous medium [20, 9, 22, 13] or Brownian forcing of a stochastic ordinary differential equation [3]. In those settings, efficient MCMC

[*] Submitted to the editors DATE.

Funding: Funding information goes here.

[†] School of Mathematics, Monash University, Victoria 3800, Australia (tiangang.cui@monash.edu).

[‡] Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, UK (gd391@bath.ac.uk).

[§] Institute for Applied Mathematics and Interdisciplinary Center for Scientific Computing, Heidelberg University, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany (r.scheichl@uni-heidelberg.de).

methods have been developed to sample the posterior and compute posterior expectations with convergence rates that are independent of the discretised parameter dimension: for example, (preconditioned) Crank-Nicolson (pCN) methods [4, 8, 18] that establish the foundation for designing and analysing MCMC algorithms in a function space setting, stochastic Newton methods [27, 29] that utilise Hessian information to accelerate the convergence, as well as operator-weighted methods [25, 10, 31] that generalise PCN methods using (potentially location-dependent) operators to adapt to the geometry of the posterior.

Discretisation also arises in the numerical simulation of the forward model, for instance, finite-element discretisations of PDEs. As many degrees of freedom are needed to accurately resolve the forward model, simulating the posterior density (which includes a forward model evaluation) can be computationally demanding. One natural way to reduce the computational cost is to utilise a hierarchy of forward models defined by a sequence of grid discretisations, ranging from computationally cheaper and less accurate coarse models to more costly but more accurate fine models. Corresponding to this hierarchy of models, the parameters can also be represented by a sequence of discretised functions with increasing dimensions. This yields *a hierarchy of posterior distributions*. By allocating different numbers of MCMC simulations to sample posteriors across different levels and by combining all those sample-based posterior estimations using *a telescoping sum* [14], the multilevel MCMC [21, 13] provides accelerated and unbiased estimates of posterior expectations.

We present a non-trivial integration of the dimension-independent likelihood-informed (DILI) MCMC [10] and the multilevel MCMC in [13] to explore the hierarchy of posterior distributions. This integration offers several advantages: First, DILI-MCMC employs an intrinsic *likelihood-informed subspace* (LIS) [11]—which involves a number of forward and adjoint model simulations—to design accelerated operator-weighted proposals. By exploiting the multilevel structure of the discretised parameters and discretised forward models, we design a *Rayleigh-Ritz procedure* to significantly reduce the computational effort in building a *hierarchical LIS* and operating with DILI proposals. Second, the resulting DILI-MCMC can drastically improve the sampling efficiency of MCMC at each level, and hence reduce the integration error of multilevel Monte Carlo for a fixed CPU time budget. To be able to fully exploit the power of multilevel MCMC and to reduce the dependencies of samples on different levels for a parallel implementation, we also suggest a new pooling strategy for allocating computational resources across different levels and constructing Markov chains at higher levels conditioned on those simulated on lower levels. Numerical results confirm the improved computational efficiency of the proposed multilevel DILI approach.

We note that the DILI proposal has been used before in the multilevel sequential Monte Carlo (SMC) setting [2], but in a very different way. We use derivative information of the likelihood to recursively construct the LIS via matrix–free eigenvalue solves, whereas [2] uses multilevel SMC to estimate the full-rank empirical posterior covariance matrix and then builds the LIS from this posterior covariance matrix. Moreover, we construct DILI proposals by exploiting the structure of the hierarchical LIS to couple Markov chains across levels, whereas [2] employs the original DILI proposal in the mutation step of SMC to improve mixing.

The paper is structured as follows. Section 2 introduces the framework of Bayesian inverse problems and MCMC sampling while section 3 discusses the general framework of multilevel MCMC. The Rayleigh-Ritz procedure for the recursive construction of the hierarchical LIS is presented in section 4 The new coupling strategy in the implementation of multilevel MCMC, as well as the coupled DILI proposals exploiting the hierarchical LIS are introduced in section 5. Section 6 provides numerical experiments to demonstrate the efficacy of the resulting MLDILI method, while finally, in section 7, we provide some concluding remarks.

2. Background.

In this section, we review the Bayesian formulation of inverse problems, the dimension-independent likelihood-informed MCMC approach, posterior discretisation, as well as the bias-variance decomposition for MCMC algorithms.

2.1. Bayesian inference framework. Suppose the parameter of interest is some function u in a separable Hilbert space $\mathcal{H}(\Omega)$ defined over a given bounded domain $\Omega \subset \mathbb{R}^d$. We introduce a prior probability measure μ_0 satisfying $\mu_0(\mathcal{H}) = 1$ to represent the *a priori* information about the function u. The inner product on $\mathcal{H}$ is denoted by $\langle \cdot\,, \cdot \rangle_{\mathcal{H}}$, with associated norm denoted by $\| \cdot \|_{\mathcal{H}}$. For brevity, where misinterpretation is not possible, we will drop the subscript $\mathcal{H}$. We assume that the prior measure is Gaussian with mean $m_0 \in \mathcal{H}$ and a self-adjoint, positive definite covariance operator Γ_{pr} that is trace-class, so that the prior provides a full probability measure on $\mathcal{H}$.

Given observed data $\boldsymbol{y} \in \mathbb{R}^d$ and the forward model $F : \mathcal{H} \to \mathbb{R}^d$, we define the likelihood function $\mathcal{L}(\boldsymbol{y}|u)$ of $\boldsymbol{y}$ given u. Denoting the posterior probability measure by μ_y, the posterior distribution on any infinitesimal volume $du \in \mathcal{H}$ is given by

$$\mu_y(du) \propto \mathcal{L}(\boldsymbol{y}|u)\mu_0(du)\,. \tag{2.1}$$

Making the simplifying assumption that the observational noise is additive and Gaussian with zero mean and covariance matrix Γ_{obs}, the observation model has the form

$$\boldsymbol{y} = F(u) + \boldsymbol{e}, \quad \boldsymbol{e} \sim \mathcal{N}(0, \Gamma_{\mathrm{obs}})\,, \tag{2.2}$$

and it follows immediately that the likelihood function satisfies

$$\mathcal{L}(\boldsymbol{y}|u) \propto \exp(-\eta(u; \boldsymbol{y}))\,, \tag{2.3}$$

where $\eta(\boldsymbol{y}; u)$ is the data-misfit functional defined by

$$\eta(u; \boldsymbol{y}) \equiv \frac{1}{2}\big(\boldsymbol{y} - F(u)\big)^{\top} \Gamma_{\mathrm{obs}}^{-1} \big(\boldsymbol{y} - F(u)\big)\,. \tag{2.4}$$

Assumption 2.1. *We assume that the forward model $F : \mathcal{H} \to \mathbb{R}^d$ satisfies:*

1. *For all $\varepsilon > 0$, there exists a constant $K(\varepsilon) > 0$ such that, for all $u \in \mathcal{H}$,*

$$|F(u)| \le \exp\big(K(\varepsilon) + \varepsilon \|u\|_{\mathcal{H}}^2\big)\,.$$

2. *For any $u \in \mathcal{H}$, there exists a bounded linear operator $\mathrm{J}(u) : \mathcal{H} \to \mathbb{R}^d$ such that*

$$\lim_{\delta u \to 0} \frac{|F(u + \delta u) - F(u) - \mathrm{J}(u)\delta u|}{\|\delta u\|_{\mathcal{H}}} = 0, \quad \forall \delta u \in \mathcal{H}.$$

In particular, this also implies the Lipschitz continuity of F.

Given observations $\boldsymbol{y}$ such that $\|\boldsymbol{y}\| < \infty$ and a forward model that satisfies Assumption 2.1, [34] shows that the resulting data-misfit function is sufficiently bounded and locally Lipschitz, and thus the posterior measure is dominated by the prior measure. The second condition states that the forward model is first-order Fréchet differentiable, and hence the Gauss-Newton approximation of the Hessian of the data-misfit functional is bounded.

Suppose we have some quantity of interest (QoI) that is a functional of the parameter u denoted by $Q : \mathcal{H} \to \mathbb{R}^q$, e.g., flow rate. Then, posterior-based model predictions can be formulated as expectations of that QoI over the posterior. We will denote them by

$$\mathbb{E}_{\mu_y}\big[Q\big] \equiv \mathbb{E}_{U\sim\mu_y}\big[Q(U)\big].$$

MCMC methods draw (correlated) MCMC samples $U^{(1)}, \ldots, U^{(N)}$ from the posterior and then estimate expected QoI(s) using Monte Carlo integration:

$$\mathbb{E}_{\mu_y}\big[Q\big] \approx \frac{1}{N}\textstyle\sum_{i=1}^{N} Q(U^{(i)}). \tag{2.5}$$

2.2. Dimension-independent likelihood-informed MCMC on function space. The Metropolis-Hastings (MH) algorithm [28, 19] provides a general framework to design transition kernels that have the posterior as their invariant distribution to generate a Markov chain of random variables that targets the posterior.

Definition 2.2 (Metropolis-Hastings Kernel). *Given the current state $U^{(k)} = u^*$, a candidate state u' is drawn from a proposal distribution $q(u^*, \cdot)$. The transition probability from $U^{(k)} = u^*$ to u' and the reverse transition probability are defined by the pair of measures*

$$\begin{aligned} \nu(du^*, du') &= q(u^*, du')\mu_y(du^*) \\ \nu^\perp(du^*, du') &= q(u', du^*)\mu_y(du'). \end{aligned} \tag{2.6}$$

Then, the next state of the Markov chain is set to $U^{(k+1)} = u'$ with probability

$$\alpha(u^*, u') = \min\Big\{1, \frac{d\nu^\perp}{d\nu}(u^*, u')\Big\}, \tag{2.7}$$

and to $U^{(k+1)} = u^*$ otherwise.

MH algorithms require the absolute continuity condition $\nu^\perp \ll \nu$ to define a valid transition kernel with non-zero acceptance probability as the dimension goes to infinity [38]. We will refer to a MH algorithm as *well-defined* (and *dimension-independent*) if this absolute continuity condition holds. For probability measures over function spaces in the setting considered here, the sequence of papers [4, 34, 18, 8, 17] provide a viable way to construct well-defined MH algorithms using a preconditioned Crank-Nicolson (pCN) discretisation of a particular Langevin SDE. The pCN proposal has the form

$$u' = a(u^* - m_0) + m_0 - \gamma(1-a)\Gamma_{\text{pr}}\nabla_u\eta(u^*; \boldsymbol{y}) + \sqrt{1-a^2}\,\Gamma_{\text{pr}}^{\frac{1}{2}}\xi, \tag{2.8}$$

where $\xi \sim \mathcal{N}(0, \mathrm{I})$ and $\gamma \in \{0, 1\}$ is a tuning parameter to switch between Langevin ($\gamma = 1$) and Ornstein-Uhlenbeck proposal ($\gamma = 0$). It is required that $a \in (-1, 1)$. The pCN proposal (2.8) satisfies the desired absolute continuity condition and the acceptance probability does not go to zero as the discretisation of u is refined.

The pCN proposal (2.8) scales uniformly in all directions with respect to the norm induced by the prior covariance. Since the posterior necessarily contracts the prior along parameter directions that are informed by the likelihood, the Markov chain produced by the standard pCN proposal decorrelates more quickly in the likelihood-informed parameter subspace than in the orthogonal complement, which is prior-dominated [25, 10]. Thus, proposed moves of pCN can be effectively too small in prior-dominated directions, resulting in poor mixing.

The dimension-independent likelihood-informed (DILI) MCMC [10] provides a systematic way to design proposals that adapt to the anisotropic structure of the posterior while retaining dimension-independent performance. It considers operator-weighted proposals in the form of

$$u' = \Gamma_{\text{pr}}^{\frac{1}{2}}\mathrm{A}\Gamma_{\text{pr}}^{-\frac{1}{2}}(u^* - m_0) + m_0 - \Gamma_{\text{pr}}^{\frac{1}{2}}\mathrm{G}\Gamma_{\text{pr}}^{\frac{1}{2}}\nabla_u\eta(u^*; \boldsymbol{y}) + \Gamma_{\text{pr}}^{\frac{1}{2}}\mathrm{B}\xi, \tag{2.9}$$

where A, B, and G are bounded, self-adjoint operators on $\mathrm{Im}(\Gamma_{\text{pr}}^{-1/2})$ that satisfy certain properties to be discussed below. In this paper, we set G to zero throughout and thus consider only non-Langevin type proposals. By applying a whitening transform

$$v = \Gamma_{\text{pr}}^{-\frac{1}{2}}(u - m_0) \tag{2.10}$$

to the parameter u and by denoting (in a slight abuse of notation) the associated data-misfit functional again by $\eta(v; y) \leftarrow \eta(\Gamma_{\text{pr}}^{1/2}v + m_0; y)$, the proposal (2.9) simplifies to

$$v' = \mathrm{A}v^* + \mathrm{B}\xi. \tag{2.11}$$

The following theorem provides sufficient conditions for constructing the operators A and B such that the proposal (2.11) yields a well-defined MH algorithm, as well as a formula for the acceptance probability.

Theorem 2.3. *Suppose that the posterior measure μ_y is equivalent to the prior measure μ_0 and that the self-adjoint operators* A *and* B *commute, that is, they can be defined by a common set of eigenfunctions $\{\psi_i \in \mathrm{Im}(\Gamma_{\text{pr}}^{-1/2}) : i \in \mathbb{N}\}$ with corresponding eigenvalues $\{a_i\}_{i=1}^\infty$ and*

$\{b_i\}_{i=1}^\infty$, *respectively. Suppose further that*

$$\{a_i\}_{i=1}^\infty,\ \{b_i\}_{i=1}^\infty \subset \mathbb{R}\backslash\{0\} \quad \textit{and} \quad \sum_{i=1}^{\infty}\left(a_i^2+b_i^2-1\right)^2<\infty.$$

Then, the proposal (2.11) *delivers a well-defined MCMC algorithm and the acceptance probability is given by*

$$\alpha\big(v^*,v'\big)=\min\left\{1,\frac{\exp\left(-\eta(v';y)-\frac{1}{2}\langle v',\mathrm{B}^{-2}(\mathrm{A}^2+\mathrm{B}^2-\mathrm{I})v'\rangle\right)}{\exp\left(-\eta(v^*;y)-\frac{1}{2}\langle v^*,\mathrm{B}^{-2}(\mathrm{A}^2+\mathrm{B}^2-\mathrm{I})v^*\rangle\right)}\right\}.$$

Proof. The above assumptions are simplified versions of those in Theorem 3.1 of [10]. The acceptance probability directly follows from Corollary 3.5 of [10]. ∎

The DILI proposal (2.11) enables different scalings in the proposal moves along different parameter directions. By choosing appropriate eigenfunctions $\{\psi_i\}_{i=1}^\infty$ and eigenvalues $\{a_i,b_i\}_{i=1}^\infty$, it can capture the geometry of the posterior, and thus can potentially improve the mixing of the resulting Markov chain.

The likelihood-informed subspace (LIS) [11, 12] provides a viable way to construct such operators A and B. It is spanned by the leading eigenfunctions of the eigenvalue problem

$$\mathbb{E}_{V\sim\mu^*}\big[\mathrm{H}(V)\big]\psi_i=\lambda_i\psi_i, \tag{2.12}$$

where $\mathrm{H}(v)$ is some information metric of the likelihood function (with respect to the transformed parameter v), for example, the Hessian of the data-misfit functional or the Fisher information, and μ^* is some reference measure, for example, the posterior or the Laplace approximation of the posterior. In the LIS, spanned by $\{\psi_1,\ldots,\psi_r\}$, the posterior may significantly differ from the prior. Thus, we prescribe inhomogeneous eigenvalues $\{a_i\}_{i=1}^r$ and $\{b_i\}_{i=1}^r$ to ensure that the proposal follows the possibly relatively tight geometry of the posterior. In the complement of the LIS, where the posterior does not differ significantly from the prior, we can use the original pCN proposal and set $\{a_i\}_{i>r}$ and $\{b_i\}_{i>r}$ to some constant values $a_\perp$ and $b_\perp$, respectively. Further details on the computation of the LIS basis and the choice of eigenvalues will be discussed in the multilevel context in later sections.

2.3. Posterior discretisation and bias-variance decomposition.

When the forward model involves a partial/ordinary differential equation and the parameter is defined as a spatial/temporal stochastic process, it is necessary in practice to discretise the parameter and the forward model using appropriate numerical methods.

A common way to discretise the parameter is the Karhunen–Loéve expansion, which also serves the purpose of the whitening transform. Given the prior mean $m_0(x)$ and the prior covariance Γ_{pr}, we express the unknown parameter u as the linear combination of the first R eigenfunctions $\{\phi_1,\ldots,\phi_R\}$ of the eigenvalue problem $\Gamma_{\mathrm{pr}}\phi_j=\omega_j\phi_j$, such that

$$u_R(x)=m_0(x)+\textstyle\sum_{j=1}^R\sqrt{\omega_j}\,\phi_j(x)\,v_j,\quad x\in\Omega. \tag{2.13}$$

The discretised prior $p_R(\boldsymbol{v})$ associated with the random coefficients $\boldsymbol{v}=[v_1,\ldots,v_R]^\top$ is Gaussian with zero mean and covariance equal to the $R\times R$ identity matrix I_R.

We discretise the forward model using a numerical method, such as finite elements or finite differences, with M degrees of freedom, which yields a discretised forward model $F_{R,M}$ mapping from the discretised coefficients $\boldsymbol{v}$ to the observables. In this way, the posterior measure (2.1) can be discretised, leading to the discrete density

$$\pi_{R,M}(\boldsymbol{v}|\boldsymbol{y})\propto\exp(-\eta_{R,M}(\boldsymbol{v};\boldsymbol{y}))\,p_R(\boldsymbol{v}), \tag{2.14}$$

where

$$\eta_{R,M}(\boldsymbol{v};\boldsymbol{y})=\frac{1}{2}\big(\boldsymbol{y}-F_{R,M}(\boldsymbol{v})\big)^\top\Gamma_{\mathrm{obs}}^{-1}\big(\boldsymbol{y}-F_{R,M}(\boldsymbol{v})\big)$$

is the discretised data-misfit function. Correspondingly, we also define the discretised QoI $Q_{R,M}(\boldsymbol{v})$, which maps the discretise coefficient vector $\boldsymbol{v}$ to the discretised QoI.

The discretised parameters and forward models can be indexed by the discretisation level. We consider a hierarchy of $L+1$ levels of discretised parameter spaces with dimensions $R_0 \leq R_1 \leq \ldots \leq R_L$ and a hierarchy of discretised forward models with $M_0 \leq M_1 \leq \ldots \leq M_L$ degrees of freedom. Discretised parameter, forward model and QoI on level ℓ are denoted by

$$\boldsymbol{v}_\ell = [v_1, \ldots, v_{R_\ell}]^\top, \quad F_\ell(\boldsymbol{v}_\ell) \equiv F_{R_\ell, M_\ell}(\boldsymbol{v}_\ell) \;\text{ and }\; Q_\ell(\boldsymbol{v}_\ell) \equiv Q_{R_\ell, M_\ell}(\boldsymbol{v}_\ell),$$

respectively. Thus, the discretised data-misfit function, prior and posterior on level ℓ are

$$\eta_\ell(\boldsymbol{v}_\ell; \boldsymbol{y}) \equiv \eta_{R_\ell, M_\ell}(\boldsymbol{v}_\ell; \boldsymbol{y}), \;\; p_\ell(\boldsymbol{v}_\ell) \equiv p_{R_\ell}(\boldsymbol{v}_\ell), \text{ and } \;\pi_\ell(\boldsymbol{v}_\ell|\boldsymbol{y}) \equiv \pi_{R_\ell, M_\ell}(\boldsymbol{v}_\ell|\boldsymbol{y}), \tag{2.15}$$

respectively, with the associated posterior expectation $\mathbb{E}_{\pi_\ell}\big[Q_\ell\big] \equiv \mathbb{E}_{\boldsymbol{V}_\ell \sim \pi_\ell}\big[Q_\ell(\boldsymbol{V}_\ell)\big]$.

Assumption 2.4. *(i) The bias of the posterior expectation on level ℓ can be bounded in terms of the number of degrees of freedom of the forward model such that*

$$\big|\mathbb{E}_{\mu_y}\big[Q\big] - \mathbb{E}_{\pi_\ell}\big[Q_\ell\big]\big| = \mathcal{O}(M_\ell^{-\vartheta_{\rm b}}), \tag{2.16}$$

for some constant $\vartheta_{\rm b} > 0$. Implicitly, this assumes that R_ℓ is sufficiently large such that the bias due to parameter approximation is dominated by the error due to the forward model approximation on level ℓ.

(ii) For the computational cost of carrying out one step of MCMC (including a forward model simulation) it is assumed that there exists a constant $\vartheta_{\rm c} > 0$ such that

$$C_\ell = \mathcal{O}(M_\ell^{\vartheta_{\rm c}}). \tag{2.17}$$

Consider discretisation level L and let $\{\boldsymbol{V}_L^{(j)}\}_{j=1}^{N_{\rm MC}}$ be a Markov chain produced by a MCMC algorithm converging in distribution to π_L. An estimate for the expectation $\mathbb{E}_{\pi_L}\big[Q_L\big]$ is

$$Y^{\rm MC} \equiv \frac{1}{N_{\rm MC}} \textstyle\sum_{j=1}^{N_{\rm MC}} Q_L(\boldsymbol{V}_L^{(j)}) \;\approx\; \mathbb{E}_{\pi_L}\big[Q_L\big]\,. \tag{2.18}$$

The mean-squared-error (MSE) of the Monte Carlo estimator (2.18) allows a bias-variance decomposition of the form

$$\text{MSE}(Y^{\rm MC}) = \underbrace{\Big|\mathbb{E}_{\mu_y}\big[Q\big] - \mathbb{E}_{\pi_L}\big[Q_L\big]\Big|^2}_{\text{Square of Bias}} + \underbrace{\text{Var}_{\pi_L}(Q_L)\Big/ N_{\rm MC}^{\rm eff}}_{\text{Var}(Y^{\rm MC})}, \tag{2.19}$$

where $N_{\rm MC}^{\rm eff}$ is the effective sample size of the Markov chain $\{\boldsymbol{V}_L^{(j)}\}_{j=1}^{N_{\rm MC}}$. This effective sample size is proportional to the total sample size, i.e., $N_{\rm MC}^{\rm eff} = N_{\rm MC}/\tau_{\rm MC}$, where $\tau_{\rm MC} \geq 1$ is the integrated autocorrelation time (IACT) of the Markov chain.

Choosing $N_{\rm MC}$ such that the two terms in (2.19) of the MCMC estimator are balanced and using Assumption 2.4, the total computational cost to achieve $\text{MSE}(Y^{\rm MC}) < \varepsilon^2$ is

$$C^{\rm MC} = \mathcal{O}\big(\tau_{\rm MC}\, \varepsilon^{-2-\vartheta_{\rm c}/\vartheta_{\rm b}}\big)\,. \tag{2.20}$$

Thus, one of the key aims in accelerating MCMC sampling is to reduce the IACT. This will be achieved via the DILI MCMC proposal. However, the multilevel method will allow us to also improve on the asymptotic rate for the cost of the standard MCMC estimator in (2.20).

3. Multilevel MCMC.

By exploiting the hierarchy of posteriors, the rate of the computational cost in (2.20) can be reduced significantly using the multilevel idea in [13]. We expand the posterior expectation in the telescoping sum

$$\mathbb{E}_{\pi_L}[Q_L] = \mathbb{E}_{\pi_0}[Q_0] + \textstyle\sum_{\ell=1}^{L} \Big(\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell-1}}[Q_{\ell-1}]\Big)\,. \tag{3.1}$$

For level zero, the sample set $\{\boldsymbol{V}_0^{(0,\,j)}\}_{j=1}^{N_0}$ is assumed to be drawn via some MCMC method that converges to $\pi_0(\,\cdot\,|\boldsymbol{y})$ and the first term in the telescoping sum (3.1) is estimated via

$$Y_0 \equiv \frac{1}{N_0}\textstyle\sum_{j=1}^{N_0} D_0^{(j)} \;\approx\; \mathbb{E}_{\pi_0}[Q_0], \quad \text{where} \quad D_0^{(j)} = Q_0(\boldsymbol{V}_0^{(0,\,j)}).$$

Since the two expectations in the difference $\mathbb{E}_{\pi_\ell}[Q_\ell]-\mathbb{E}_{\pi_{\ell\text{-}1}}[Q_{\ell\text{-}1}]$ are with respect to different discretisations of the posterior, special treatment is required for $\ell > 0$. Let $\Delta_{\ell,\ell\text{-}1}(\boldsymbol{v}_\ell, \boldsymbol{v}_{\ell\text{-}1})$ be the joint density of $\boldsymbol{v}_\ell$ and $\boldsymbol{v}_{\ell\text{-}1}$ such that

$$\int \Delta_{\ell,\ell\text{-}1}(\boldsymbol{v}_\ell, \boldsymbol{v}_{\ell\text{-}1})\, d\boldsymbol{v}_{\ell\text{-}1} = \pi_\ell(\boldsymbol{v}_\ell|\boldsymbol{y}) \;\; \text{and} \;\; \int \Delta_{\ell,\ell\text{-}1}(\boldsymbol{v}_\ell, \boldsymbol{v}_{\ell\text{-}1})\, d\boldsymbol{v}_\ell = \pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}|\boldsymbol{y}), \tag{3.2}$$

that is, the posteriors $\pi_\ell(\boldsymbol{v}_\ell|\boldsymbol{y})$ and $\pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}|\boldsymbol{y})$ are the two marginals. Then, the difference between expectations can be expressed as

$$\mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell\text{-}1}}[Q_{\ell\text{-}1}] = \mathbb{E}_{\Delta_{\ell,\ell\text{-}1}}[D_\ell], \quad \text{where} \quad D_\ell = Q_\ell(\boldsymbol{V}_\ell) - Q_{\ell\text{-}1}(\boldsymbol{V}_{\ell\text{-}1}) \tag{3.3}$$

and $(\boldsymbol{V}_\ell, \boldsymbol{V}_{\ell\text{-}1}) \sim \Delta_{\ell,\ell\text{-}1}(\cdot,\cdot)$. The construction of the joint density and the associated sampling procedure will be critical to reduce the computational complexity.

Suppose the samples $\big\{\big(\boldsymbol{V}_\ell^{(\ell,j)}, \boldsymbol{V}_{\ell\text{-}1}^{(\ell,j)}\big)\big\}_{j=1}^{N_\ell}$ form a Markov chain that converges in distribution to $\Delta_{\ell,\ell\text{-}1}(\cdot,\cdot)$ and

$$D_\ell^{(j)} = Q_\ell\big(\boldsymbol{V}_\ell^{(\ell,j)}\big) - Q_{\ell\text{-}1}\big(\boldsymbol{V}_{\ell\text{-}1}^{(\ell,j)}\big).$$

Then, the remaining terms in (3.1), for $\ell = 1, \ldots, L$, are estimated by

$$Y_\ell \equiv \frac{1}{N_\ell}\textstyle\sum_{j=1}^{N_\ell} D_\ell^{(j)} \;\approx\; \mathbb{E}_{\pi_\ell}[Q_\ell] - \mathbb{E}_{\pi_{\ell\text{-}1}}[Q_{\ell\text{-}1}]$$

and the multilevel MCMC estimator for $\mathbb{E}_{\pi_L}[Q_L]$ is defined by

$$\mathbb{E}_{\pi_L}\big[Q_L\big] \approx Y^{\mathrm{ML}} \equiv \textstyle\sum_{\ell=0}^L Y_\ell\,, \tag{3.4}$$

The mean square error of this estimator can again be decomposed as follows:

$$\mathrm{MSE}(Y^{\mathrm{ML}}) \equiv \underbrace{\big|\mathbb{E}_{\mu_y}\big[Q\big] - \mathbb{E}_{\pi_L}[Q_L]\big|^2}_{\text{Square of Bias}} + \underbrace{\textstyle\sum_{\ell=0}^L \big(\mathrm{Var}(Y_\ell) + \sum_{k\neq\ell}^L \mathrm{Cov}(Y_\ell, Y_k)\big)}_{\mathrm{Var}(Y^{\mathrm{ML}})}. \tag{3.5}$$

3.1. Variance management. For optimal efficiency, we now choose the numbers of samples N_ℓ, $\ell = 0, \ldots, N$, such as to minimise $\mathrm{Var}(Y^{\mathrm{ML}})$ for fixed computational effort. This includes the *within-level variance* $\mathrm{Var}(Y_\ell)$ and the *cross-level variance* $\mathrm{Cov}(Y_\ell, Y_k)$ for $k \neq \ell$. We will provide justifications on managing these variances using the following assumptions.

ASSUMPTION 3.1. *The Markov chains on all levels are assumed to be ergodic. This implies that the effective sample sizes are proportional to the total sample sizes, i.e.,* $N_\ell^{\mathrm{eff}} = N_\ell/\tau_\ell$, *for all* ℓ, *where* $\tau_\ell \geq 1$ *is the IACT of the Markov chain* $D_\ell^{(j)}$.

Remark 3.2. The ergodicity in Assumption 3.1 can be satisfied by removing burn-in samples using shorter MCMC simulations. The within-level variance has the form

$$\mathrm{Var}(Y_\ell) = \frac{1}{N_\ell^{\mathrm{eff}}}\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell) = \frac{\tau_\ell}{N_\ell}\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell), \tag{3.6}$$

where we set $\mathrm{Var}_{\Delta_{0,-1}}(D_0) = \mathrm{Var}_{\pi_0}(Q_0)$ and have

$$\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell) = \mathrm{Var}_{\pi_\ell}(Q_\ell) + \mathrm{Var}_{\pi_{\ell\text{-}1}}(Q_{\ell\text{-}1}) - 2\mathrm{Cov}_{\Delta_{\ell,\ell\text{-}1}}(Q_\ell, Q_{\ell\text{-}1}) \geq 0, \;\; \forall \ell > 0,$$

by Cauchy–Schwarz inequality. Thus, to reduce $\mathrm{Var}(Y_\ell)$, the joint density should be constructed in such a way that $\mathrm{Cov}_{\Delta_{\ell,\ell\text{-}1}}(Q_\ell, Q_{\ell\text{-}1})$ is *positive* and (if possible) maximised. In addition, the MCMC simulation should be made *statistically efficient* in the sense that τ_ℓ is as close to one as possible.

Assumption 3.3. *The variance* $\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell)$ *converges to zero as* $M_\ell \to \infty$ *and*

$$\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell) = \mathcal{O}(M_\ell^{-\vartheta_\mathrm{v}})\,, \tag{3.7}$$

for some constant $\vartheta_\mathrm{v} > 0$.

Proposition 3.4. *Let us assume* $\{\mathrm{Var}(Y_\ell)\}_{\ell=0}^L$ *are ordered as* $\mathrm{Var}(Y_\ell) > \mathrm{Var}(Y_k)$ *for* $\ell < k$. *Suppose that there exists an* $r < 1$ *such that*

$$\frac{\mathrm{Cov}(Y_\ell, Y_k)}{\mathrm{Var}(Y_\ell)} < r^{k-l}, \quad \textit{for all} \;\; \ell < k, \tag{3.8}$$

i.e., the cross-level covariance is insignificant compared to the within-level variance. Then

$$\mathrm{Var}(Y^{\mathrm{ML}}) = \sum_{\ell=0}^{L}\Big(\mathrm{Var}(Y_\ell) + \sum_{k\neq\ell}^{L}\mathrm{Cov}(Y_\ell,Y_k)\Big) \le \frac{1+r}{1-r}\sum_{\ell=0}^{L}\mathrm{Var}(Y_\ell)\,. \tag{3.9}$$

Proof. We have the bound

$$\begin{aligned}
\mathrm{Var}(Y^{\mathrm{ML}}) &= \sum_{\ell=0}^{L}\Big(\mathrm{Var}(Y_\ell) + 2\sum_{k>\ell}^{L}\mathrm{Cov}(Y_\ell,Y_k)\Big)\\
&\le \sum_{\ell=0}^{L}\mathrm{Var}(Y_\ell)\Big(1 + 2\sum_{k>\ell}^{\infty}\frac{\mathrm{Cov}(Y_\ell,Y_k)}{\mathrm{Var}(Y_\ell)}\Big)\\
&\le \sum_{\ell=0}^{L}\mathrm{Var}(Y_\ell)\Big(1 + 2\sum_{k>\ell}^{\infty}r^{(k-\ell)}\Big) \;=\; \frac{1+r}{1-r}\sum_{\ell=0}^{L}\mathrm{Var}(Y_\ell).
\end{aligned}$$

□

Using Proposition 3.4 and (3.6), the variance of the multilevel estimator satisfies

$$\mathrm{Var}(Y^{\mathrm{ML}}) = \mathcal{O}\Big(\textstyle\sum_{\ell=0}^{L}\dfrac{\tau_\ell}{N_\ell}\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell)\Big).$$

The total computational cost is $C^{\mathrm{ML}} = \sum_{\ell=0}^{L} N_\ell\, C_\ell$. This way, for a fixed variance, the computational cost is minimised by choosing the sample size

$$N_\ell \propto \sqrt{\tau_\ell\,\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell)/C_\ell}, \tag{3.10}$$

which leads to a total computational cost that satisfies

$$C^{\mathrm{ML}} \propto \sum_{\ell=0}^{L}\sqrt{\tau_\ell\, C_\ell\,\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(D_\ell)}. \tag{3.11}$$

Theorem 3.5. *For the multilevel MCMC estimator to satisfy* $\mathrm{MSE}(Y^{\mathrm{ML}}) < \varepsilon^2$, *the multilevel MCMC with* N_ℓ *chosen as in* (3.10) *requires an overall computational cost*

$$C^{\mathrm{ML}} = \begin{cases} \mathcal{O}(\varepsilon^{-2}) & \textit{if } \vartheta_\mathrm{v} > \vartheta_\mathrm{c} \\ \mathcal{O}(\varepsilon^{-2}|\log\varepsilon|^2) & \textit{if } \vartheta_\mathrm{v} = \vartheta_\mathrm{c} \\ \mathcal{O}(\varepsilon^{-2-(\vartheta_\mathrm{c}-\vartheta_\mathrm{v})/\vartheta_\mathrm{b}}) & \textit{if } \vartheta_\mathrm{v} < \vartheta_\mathrm{c} \end{cases}. \tag{3.12}$$

Proof. Given Assumptions 2.4, 3.1, 3.3, and Proposition 3.4, this result directly follows from the complexity theorems for multilevel Monte Carlo in [14, 7]. □

It is difficult to rigorously verify Assumption (3.8) in Proposition 3.4. However, it is often observed that the cross-level variance $\mathrm{Cov}(Y_\ell, Y_k)$ rapidly decays to zero in practice, as the Markov chains used for computing Y_ℓ and Y_k with $\ell \neq k$ are statistically independent. For example, independent Markov chains are constructed in [23] and a subsampling strategy of coarse chains are employed in [13] to ensure independence. Under this assumption, we are able to reduce the bound on the computational complexity of multilevel MCMC compared to that presented in [13], which has an extra $|\log\varepsilon|$ factor. For any positive values of $\vartheta_{\mathrm{b}}, \vartheta_{\mathrm{v}}$ and ϑ_{c}, the multilevel MCMC approach asymptotically requires less computational effort than single-level MCMC. To choose optimal numbers of samples on the various levels, estimates of the IACTs τ_ℓ, of the variances $\mathrm{Var}_{\Delta_{\ell,\ell-1}}(D_\ell)$, and of the computational costs C_ℓ are needed. Such quantities may not be known *a priori*, but they can all be obtained and adaptively improved (on the fly) as the simulation progresses.

3.2. Notations. To map vectors and matrices across adjacent levels of discretisation we define the following notation. Given the canonical basis $(\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \ldots, \hat{\boldsymbol{e}}_{R_\ell})$ of the parameter space at level ℓ, where $\hat{\boldsymbol{e}}_j \in \mathbb{R}^{R_\ell}$, we define the basis matrices $\Theta_{\ell,c} \equiv (\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \ldots, \hat{\boldsymbol{e}}_{R_{\ell-1}})$ and $\Theta_{\ell,f} \equiv (\hat{\boldsymbol{e}}_{R_{\ell-1}+1}, \ldots, \hat{\boldsymbol{e}}_{R_\ell})$, which correspond to the parameter coefficients 'active' at level ℓ-1 and the additional coefficients. We can split the parameter $\boldsymbol{v}_\ell$ into two components

$$\boldsymbol{v}_\ell = \begin{bmatrix} \boldsymbol{v}_{\ell,c} \\ \boldsymbol{v}_{\ell,f} \end{bmatrix}, \text{ where } \boldsymbol{v}_{\ell,c} = \Theta_{\ell,c}^\top \boldsymbol{v}_\ell \text{ and } \boldsymbol{v}_{\ell,f} = \Theta_{\ell,f}^\top \boldsymbol{v}_\ell, \tag{3.13}$$

which correspond to the coefficients on the previous level ℓ-1 and the additional coefficients. Given a matrix $\mathrm{A}_\ell \in \mathbb{R}^{R_\ell \times R_\ell}$, we partition the matrix as

$$\mathrm{A}_\ell = \begin{bmatrix} \mathrm{A}_{\ell,cc} & \mathrm{A}_{\ell,cf} \\ \mathrm{A}_{\ell,fc} & \mathrm{A}_{\ell,ff} \end{bmatrix}, \tag{3.14}$$

where $\mathrm{A}_{\ell,cc} \equiv \Theta_{\ell,c}^\top \mathrm{A}_\ell \Theta_{\ell,c}$ and $\mathrm{A}_{\ell,ff}$, $\mathrm{A}_{\ell,fc}$ and $\mathrm{A}_{\ell,cf}$ are defined analogously. The matrices $\Theta_{\ell,c}$ and $\Theta_{\ell,f}$ are never constructed explicitly. Operations with those matrices only involve the selection of the corresponding rows or columns of the matrix or vector.

4. Multilevel LIS.

In this section, we develop a Rayleigh-Ritz procedure to recursively compute multilevel likelihood-informed subspaces. The resulting hierarchical LIS basis can be used to generalise DILI proposals to the multilevel setting and to improve the efficiency of multilevel MCMC sampling.

For each level $\ell \in \{0, 1, \ldots, L\}$, we denote the linearisation of the forward model F_ℓ at a given parameter $\boldsymbol{v}_\ell$ by

$$\mathrm{J}_\ell(\boldsymbol{v}_\ell) = \nabla_{\boldsymbol{v}_\ell} F_\ell(\boldsymbol{v}_\ell).$$

This yields the Gauss-Newton approximation of the Hessian of the data-misfit functional at $\boldsymbol{v}_\ell$ (hereafter referred to as the GNH) in the form of

$$\mathrm{H}_\ell(\boldsymbol{v}_\ell) = \mathrm{J}_\ell(\boldsymbol{v}_\ell)^\top \Gamma_{\mathrm{obs}}^{-1} \mathrm{J}_\ell(\boldsymbol{v}_\ell). \tag{4.1}$$

Commonly used in optimisation and regression, the GNH measures local sensitivity of the parameter-to-likelihood map. The leading eigenvectors of $\mathrm{H}_\ell(\boldsymbol{v}_\ell)$ (corresponding to the largest eigenvalues) indicate parameter directions along which the likelihood function varies rapidly.

To measure the global sensitivity of the parameter-to-likelihood map, we compute the expectation of the local GNH matrix $\mathrm{H}_\ell(\boldsymbol{v}_\ell)$ over some reference distribution $p_\ell^*(\boldsymbol{v}_\ell)$:

$$\mathbb{E}_{\boldsymbol{V}_\ell \sim p_\ell^*}\big[\mathrm{H}_\ell(\boldsymbol{V}_\ell)\big]. \tag{4.2}$$

We approximate the above expectation using the sample average with K_ℓ random samples drawn from the reference distribution, which yields

$$\mathbb{E}_{\boldsymbol{V}_\ell \sim p_\ell^*}\big[\mathrm{H}_\ell(\boldsymbol{V}_\ell)\big] \approx \widehat{\mathrm{H}}_\ell \equiv \frac{1}{K_\ell}\sum_{k=1}^{K_\ell} \mathrm{H}_\ell(\boldsymbol{v}_\ell^{(k)}), \text{ where } \boldsymbol{v}_\ell^{(k)} \sim p_\ell^*(\cdot). \tag{4.3}$$

Note that the matrix $\widehat{\mathrm{H}}_\ell$ is symmetric and positive semidefinite. Different choices of the reference distribution, such as the prior or the posterior, lead to different ways to construct the LIS and different performance characteristics.

Remark 4.1. Following the discussion in [12, 39], using the posterior as the reference leads to sharp approximation properties [39] compared to other choices. However, the posterior exploration relies on MCMC sampling, and thus this choice requires adaptively estimating LIS during the MCMC sampling. The Laplace approximation to the posterior provides a reasonable alternative in a wide range of problems where the posterior is unimodal. We use the Laplace approximation as the reference distribution in this work.

The choice of the reference distribution can have an impact on the quality of the LIS basis and on the IACT of the Markov chains produced by DILI MCMC, but it does not affect the convergence of MCMC, as DILI samples the full parameter space and only uses the LIS to reduce the IACT and thus to accelerate posterior sampling.

It is often computationally infeasible to explicitly form the GNH matrix (4.1). However, all we need are matrix-vector-products (MVPs) with the GNH matrix. This requires only applications of the linearised forward model $\mathrm{J}_\ell(\boldsymbol{v}_\ell)$ and its adjoint $\mathrm{J}_\ell(\boldsymbol{v}_\ell)^\top$, which are well-established operations in the PDE-constraint optimisation literature. We refer the readers to recent applications in Bayesian inverse problems for further details, e.g., [5, 27, 29].

4.1. Base level LIS. At the base level, we use the samples $\{\boldsymbol{v}_0^{(k)}\}_{k=1}^{K_0}$ drawn from the reference $p_0^*(\cdot)$ to construct the sample-averaged GNH, $\widehat{\mathrm{H}}_0$. Then, we use the Rayleigh quotient $\langle \boldsymbol{\phi}, \widehat{\mathrm{H}}_\ell\, \boldsymbol{\phi}\rangle \,/\, \langle \boldsymbol{\phi}, \boldsymbol{\phi}\rangle$ to measure the (quadratic) change in the parameter-to-likelihood map along a parameter direction $\boldsymbol{\phi}$. Hence, the LIS can be identified via a sequence of optimisation problems of the form

$$\boldsymbol{\psi}_{0,k+1} = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max} \langle \boldsymbol{\phi}, \widehat{\mathrm{H}}_0\, \boldsymbol{\phi}\rangle, \quad \text{subject to} \quad \langle \boldsymbol{\phi}, \boldsymbol{\psi}_{0,i}\rangle = 0, \qquad \text{for } i = 1, \dots, k, \tag{4.4}$$

where $\boldsymbol{\psi}_{0,1}$ is the solution to the unconstrained optimisation problem. The sequence of optimisation problems in (4.4) is equivalent to finding the leading eigenvectors of $\widehat{\mathrm{H}}_0$.

Definition 4.2 (Base level LIS). *Given the sample–averaged GNH* $\widehat{\mathrm{H}}_0$ *on level* 0 *and a threshold* $\varrho > 0$*, we solve the eigenproblem*

$$\widehat{\mathrm{H}}_0\, \boldsymbol{\psi}_{0,i} = \lambda_{0,i} \boldsymbol{\psi}_{0,i}, \tag{4.5}$$

and then use the r_0 *leading eigenvectors with eigenvalues* $\lambda_{0,i} > \varrho$*, for* $i = 1, \dots, r_0$*, to define the LIS basis* $\Psi_{0,r_0} = [\boldsymbol{\psi}_{0,1}, \boldsymbol{\psi}_{0,2} \dots, \boldsymbol{\psi}_{0,r_0}]$*, which spans an* r_0*-dimensional subspace in* $\mathbb{R}_0^R$*.*

The eigenvalues in (4.5) provide empirical sensitivity measures of the likelihood function relative to the prior (which here is i.i.d. Gaussian) along corresponding eigenvectors [11, 39]. Eigenvectors corresponding to eigenvalues less than 1 can be interpreted as parameter directions where the likelihood is dominated by the prior. Thus, we typically choose a value less than one for the truncation threshold, i.e., $\varrho < 1$.

4.2. LIS enrichment. Because the computational cost of a MVP with the GNH scales at least linearly with the degrees of freedom M_ℓ of the forward model on level ℓ, constructing the LIS can be computationally costly. We present a new approach to accelerate the LIS construction by employing a *recursive LIS enrichment* using the hierarchy of forward models and parameter discretisations. The resulting hierarchy of LISs will be used to reduce the computational complexity of constructing and operating with the resulting DILI proposals.

We reuse the LIS bases computed on the coarser levels by 'lifting' them and then recursively enrich them at each new level using a *Rayleigh-Ritz procedure*, rather than recomputing the entire basis from scratch on each level. Ideally, the subspace added on each level will have decreasing dimension, as the model and parameter approximations were assumed to converge with $\ell \to \infty$ and thus no longer provide additional information for the parameter inference.

Definition 4.3 (Lifted LIS basis). *Suppose we have an orthogonal LIS basis* $\Psi_{\ell\text{-}1,r} \in \mathbb{R}^{R_{\ell\text{-}1} \times r_{\ell\text{-}1}}$ *on level* ℓ-1*. We lift* $\Psi_{\ell\text{-}1,r}$ *from the coarse parameter space* $\mathbb{R}^{R_{\ell\text{-}1}}$ *to the fine* *parameter space $\mathbb{R}^{R_\ell}$ using the basis matrix $\Theta_{\ell,c}$ defined in Section 3.2. The lifted LIS basis vectors are collect in the matrix*

$$\Psi_{\ell,c} = \Theta_{\ell,c}\,\Psi_{\ell\text{-}1,r}. \tag{4.6}$$

Proposition 4.4. *The lifted LIS basis matrix $\Psi_{\ell,c}$ has orthonormal columns that span an $r_{\ell\text{-}1}$-dimensional subspace in $\mathbb{R}^{R_\ell}$, i.e., $\Psi_{\ell,c}^\top \Psi_{\ell,c} = \mathrm{I}_{r_{\ell\text{-}1}}$.*

Proof. The proof directly follows as the matrix $\Theta_{\ell,c}$ has orthonormal columns. □

Given K_ℓ samples $\{\boldsymbol{v}_\ell^{(k)}\}_{k=1}^{K_\ell}$ from the reference distribution $p_\ell^*(\cdot)$, let $\widehat{\mathrm{H}}_\ell$ be the resulting sample-averaged GNH. To enrich the lifted LIS basis $\Psi_{\ell,c}$ we now identify likelihood-sensitive parameter directions in the null space null$(\Psi_{\ell,c})$ by recursively optimising the Rayleigh quotient in the orthogonal complement of range$(\Psi_{\ell,c})$, i.e.,

$$\boldsymbol{\psi}_{\ell,k+1} = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max}\langle \boldsymbol{\phi}, \widehat{\mathrm{H}}_\ell\,\boldsymbol{\phi}\rangle, \tag{4.7}$$

$$\text{subject to }\ \Pi_{\ell,c}\phi = 0\ \text{ and }\ \langle \boldsymbol{\phi}, \boldsymbol{\psi}_{\ell,i}\rangle = 0,\ \text{for } i = 1, \ldots, k,$$

where $\Pi_{\ell,c} = \Psi_{\ell,c}\Psi_{\ell,c}^\top$ is an orthogonal projector. This optimisation problem can be solved as an eigenvalue problem using the *Rayleigh-Ritz procedure* [32].

Theorem 4.5. *The optimisation problem (4.7) is equivalent to finding the leading eigenvectors of the projected eigenproblem*

$$\big(\mathrm{I}_{R_\ell} - \Pi_{\ell,c}\big)\,\widehat{\mathrm{H}}_\ell\,\boldsymbol{\psi}_{\ell,i} = \gamma_{\ell,i}\boldsymbol{\psi}_{\ell,i}, \quad \|\boldsymbol{\psi}_{\ell,i}\| = 1. \tag{4.8}$$

Proof. This result follows from the properties of orthogonal projectors and of the stationary points of the Rayleigh quotient. Here, we sketch the proof as follows. The constraint $\Pi_{\ell,c}\boldsymbol{\phi} = 0$ implies $\boldsymbol{\phi} = (\mathrm{I}_{R_\ell} - \Pi_{\ell,c})\boldsymbol{\phi}$, since $(\mathrm{I}_{R_\ell} - \Pi_{\ell,c})$ is also an orthogonal projector. Hence, the optimisation problem becomes

$$\boldsymbol{\psi}_{\ell,k+1} = \underset{\|\boldsymbol{\phi}\|=1}{\arg\max}\langle \boldsymbol{\phi}, (\mathrm{I}_{R_\ell} - \Pi_{\ell,c})\widehat{\mathrm{H}}_\ell(\mathrm{I}_{R_\ell} - \Pi_{\ell,c})\,\boldsymbol{\phi}\rangle,\ \ \text{subject to }\ \langle \boldsymbol{\phi}, \boldsymbol{\psi}_{\ell,i}\rangle = 0,\ i = 1, \ldots, k.$$

The solutions (for $k = 1, 2, \ldots$) to these optimisation problems are given by the leading eigenvectors of the eigenproblem

$$\big(\mathrm{I}_{R_\ell} - \Pi_{\ell,c}\big)\,\widehat{\mathrm{H}}_\ell\,\big(\mathrm{I}_{R_\ell} - \Pi_{\ell,c}\big)\,\boldsymbol{\psi}_{\ell,i} = \gamma_{\ell,i}\boldsymbol{\psi}_{\ell,i}.$$

However, since $\boldsymbol{\psi}_{\ell,i} \in \text{range}\big(\mathrm{I}_{R_\ell} - \Pi_{\ell,c}\big)$ this is equivalent to

$$\big(\mathrm{I}_{R_\ell} - \Pi_{\ell,c}\big)\,\widehat{\mathrm{H}}_\ell\,\boldsymbol{\psi}_{\ell,i} = \gamma_{\ell,i}\boldsymbol{\psi}_{\ell,i}.$$ □

Definition 4.6 (LIS enrichment on level ℓ). *The leading s_ℓ (normalised) eigenvectors of the eigenproblem (4.8) with eigenvalues $\gamma_{\ell,i} > \varrho$ are denoted by*

$$\Psi_{\ell,f} = [\boldsymbol{\psi}_{\ell,1}, \ldots, \boldsymbol{\psi}_{\ell,s_\ell}]. \tag{4.9}$$

They are added to the lifted LIS basis from level ℓ - 1 to form the enriched LIS basis

$$\Psi_{\ell,r} = [\Psi_{\ell,c}, \Psi_{\ell,f}] \tag{4.10}$$

on level ℓ, where the basis vectors in (4.9) denote the auxiliary "fine scale" directions added on level ℓ. By construction, all the LIS basis vectors at level ℓ are mutually orthogonal. That is, $\Psi_{\ell,r}^\top\Psi_{\ell,r} = \mathrm{I}_{r_\ell}$. We also have $r_\ell = r_{\ell\text{-}1} + s_\ell$.

4.3. Computational complexity. By construction, the LIS basis $\Psi_{\ell,r}$ is block upper triangular and can be recursively defined as

$$\Psi_{\ell,r} = [\Psi_{\ell,c}, \Psi_{\ell,f}] = \begin{bmatrix} \Psi_{\ell\text{-}1,r} & \mathrm{Z}_{\ell,c} \\ 0 & \mathrm{Z}_{\ell,f} \end{bmatrix}, \tag{4.11}$$

where $\mathrm{Z}_{\ell,c} = \Theta_{\ell,c}^{\top}\Psi_{\ell,f} \in \mathbb{R}^{R_{\ell-1}\times s_\ell}$, $\mathrm{Z}_{\ell,f} = \Theta_{\ell,f}^{\top}\Psi_{\ell,f} \in \mathbb{R}^{(R_\ell - R_{\ell-1})\times s_\ell}$, and $\Psi_{\ell-1,r} \in \mathbb{R}^{R_{\ell-1}\times r_{\ell-1}}$. We have $s_\ell = r_\ell - r_{\ell-1}$ and define $s_0 = r_0$ for consistency. The hierarchical LIS reduces the computational cost of operating with the LIS basis and the associated storage cost. This is critical for building efficient multilevel DILI proposals that will be discussed later. In addition, the recursive LIS enrichment is computationally more efficient, since the amount of costly PDE solves on the finer levels will be significantly reduced. Here we analyse and compare the complexities of the construction of the hierarchical LIS and of the single-level LIS, constructed directly on level L.

We analyse the construction of the hierarchical LIS under the following assumptions.

Assumption 4.7. 1. *The parameter dimensions satisfy $R_\ell = R_0 e^{\beta_{\mathrm{p}}\ell}$ for some $\beta_{\mathrm{p}} > 0$.*
2. *The number of auxiliary LIS basis vectors satisfies $s_\ell \le s_0\, e^{-\beta_{\mathrm{r}}\ell}$ for some $\beta_{\mathrm{s}} > 0$.*
3. *The degrees of freedom in the forward model satisfy $M_\ell = M_0\, e^{\beta_{\mathrm{m}}\ell}$ for some $\beta_{\mathrm{m}} > 0$.*
4. *The computational cost of a MVP with one sample of the GNH $\mathrm{H}_\ell(\boldsymbol{v}_\ell^{(k)})$ is proportional to one evaluation of the forward model and thus $\mathcal{O}(M_\ell^{\vartheta_{\mathrm{c}}})$ (cf. Assumption 2.4).*
5. *The number of samples to compute the sample-averaged GNH is the same on all levels, i.e., $K_\ell = K$ independent of ℓ.*
6. *For the single-level LIS constructed on level L, we assume that the LIS dimension satisfies $r_L^{\mathrm{single}} \ge c\, r_0$ for some constant $c > 0$.*

The storage cost of the hierarchical LIS basis and the storage cost of the single-level LIS basis on level L are, respectively,

$$\zeta_{\mathrm{multi}} = \textstyle\sum_{l=0}^{L} R_\ell\, s_\ell, \quad \text{and} \quad \zeta_{\mathrm{single}} = R_L\, r_L^{\mathrm{single}}.$$

The floating point operations for one MVP with the hierarchical LIS basis and with the single-level LIS basis are $O\big(\zeta_{\mathrm{multi}}\big)$ and $O\big(\zeta_{\mathrm{single}}\big)$, respectively, with the same hidden constant.

Corollary 4.8. *The reduction factor of storing and operating with the hierarchical LIS basis (as opposed to the standard single-level LIS on level L) satisfies the upper bound*

$$\frac{\zeta_{\mathrm{multi}}}{\zeta_{\mathrm{single}}} \le \frac{1}{c}\, \min\left(L+1, \frac{1}{1-e^{-|\beta_{\mathrm{p}}-\beta_{\mathrm{r}}|}}\right) e^{-\min(\beta_{\mathrm{p}},\beta_{\mathrm{r}})L}\,. \tag{4.12}$$

Proof. See Appendix A. □

Using a similar derivation, we can also obtain the reduction factor for constructing the hierarchical LIS basis. The number of MVPs (with the sample-averaged GNH $\widehat{\mathrm{H}}_0$) in the construction of the base level LIS via the eigenproblems (4.5) is linear in the number of leading eigenvectors obtained, i.e., $\mathcal{O}(s_0)$. The same holds for the number of MVPs with $\widehat{\mathrm{H}}_\ell$ in the construction of the auxiliary LIS vectors in the recursive enrichment solving the eigenproblems in (4.8). Thus, the overall computational complexities for constructing the hierarchical LIS basis is

$$\chi_{\mathrm{multi}} = \mathcal{O}\big(K \textstyle\sum_{l=0}^{L} s_\ell\, M_\ell^{\vartheta_{\mathrm{c}}}\big).$$

Similarly, the construction of the single level LIS on level L is

$$\chi_{\mathrm{single}} = \mathcal{O}\big(K r_L^{\mathrm{single}}\, M_L^{\vartheta_{\mathrm{c}}}\big),$$

where the prefactors are the same. The following corollary can be proved in the same way as Corollary 4.8, since we have assumed that $M_\ell^{\vartheta_{\mathrm{c}}} = M_0^{\vartheta_{\mathrm{c}}}\, e^{\beta_{\mathrm{m}}\vartheta_{\mathrm{c}}\ell}$.

Corollary 4.9. *The reduction factor of building the hierarchical LIS basis (as opposed to the standard single-level LIS basis on level L) satisfies the upper bound*

$$\frac{\chi_{\mathrm{multi}}}{\chi_{\mathrm{single}}} \le \frac{1}{c}\, \min\left(L+1, \frac{1}{1-e^{-|\beta_{\mathrm{m}}\vartheta_{\mathrm{c}}-\beta_{\mathrm{r}}|}}\right) e^{-\min(\beta_{\mathrm{m}}\vartheta_{\mathrm{c}}\,,\,\beta_{\mathrm{r}})L}\,. \tag{4.13}$$

5. Multilevel DILI. In this section, we first present a coupling strategy to construct positively correlated Markov chains for adjacent levels for computing multilevel MCMC estimators. The coupling strategy introduced in the original MLMCMC [13] can be viewed as a special case of our new approach. Utilising this coupling framework, we design a computationally efficient way to couple DILI proposals within MLMCMC by exploiting the structure of the hierarchical LIS.

5.1. Coupling strategy. We want to construct positively correlated Markov chains $\{\boldsymbol{V}_{\ell-1}^{(\ell,j)}\}$ and $\{\boldsymbol{V}_{\ell}^{(\ell,j)}\}$ for adjacent levels $\ell-1$ and ℓ with the invariant densities $\pi_{\ell-1}(\boldsymbol{v}_{\ell-1}|\boldsymbol{y})$ and $\pi_{\ell}(\boldsymbol{v}_{\ell}|\boldsymbol{y})$, respectively. As described in Section 3, this can be seen as sampling from the joint density $\Delta_{\ell,\ell-1}(\boldsymbol{v}_{\ell},\boldsymbol{v}_{\ell-1})$ such that condition (3.2) holds and such that the within-level variance $\mathrm{Var}_{\Delta_{\ell,\ell-1}}(D_{\ell})$ is reduced (cf. Remark 3.2).

Suppose the j-th states of the Markov chains at levels $\ell-1$ and ℓ are $\boldsymbol{V}_{\ell-1}^{(\ell,j)} = \boldsymbol{v}_{\ell-1}^{*}$ and $\boldsymbol{V}_{\ell}^{(\ell,j)} = \boldsymbol{v}_{\ell}^{*}$, respectively. The state at level ℓ has the form $\boldsymbol{v}_{\ell}^{*} = (\boldsymbol{v}_{\ell,c}^{*}, \boldsymbol{v}_{\ell,f}^{*})$, corresponding to the coarse part of the parameters (shared with level $\ell-1$) and the refined part, respectively. We call the two Markov chains *coupled* at the j-th state if $\boldsymbol{v}_{\ell,c}^{*} = \boldsymbol{v}_{\ell-1}^{*}$. Assuming the two chains to be coupled at the jth state, the next states are proposed as follows.

Proposition 5.1. *Given a set of posterior samples* $\mathcal{V}_{\ell-1} = \{\boldsymbol{v}_{\ell-1}^{(i)}\}_{i=1}^{N_{\ell-1}}$ *drawn from level* $\ell-1$*], we can define an empirical probability density in the form*

$$\widetilde{\pi}_{\ell-1}(\boldsymbol{v}_{\ell-1}) = \tfrac{1}{N_{\ell-1}} \textstyle\sum_{i=1}^{N_{\ell-1}} \delta\big(\boldsymbol{v}_{\ell-1} - \boldsymbol{v}_{\ell-1}^{(i)}\big). \tag{5.1}$$

Drawing a sample $\boldsymbol{v}_{\ell-1}'$ *from the set* $\mathcal{V}_{\ell-1}$ *according to a uniform discrete distribution, the sample* $\boldsymbol{v}_{\ell-1}'$ *has the probability density*

$$\widetilde{\pi}_{\ell-1}(\boldsymbol{v}_{\ell-1}') = \pi_{\ell-1}(\boldsymbol{v}_{\ell-1}'|\boldsymbol{y}).$$

It can be used as the proposal density for the coarse components in MH to sample from the fine chain $\{\boldsymbol{V}_{\ell}^{(\ell,j)}\}$*, thus, positively correlating the two Markov chains.*

The idea in Proposition 5.1 is discussed in [19, 37]. In the coupling strategy, we use the independent proposal defined by (5.1) to sample $\pi_{\ell-1}(\boldsymbol{v}_{\ell-1}|\boldsymbol{y})$. This way, the proposed candidate $\boldsymbol{v}_{\ell-1}' \sim \widetilde{\pi}_{\ell-1}(\cdot)$ is independent of the current state $\boldsymbol{v}_{\ell-1}^{*}$. Given Proposition 5.1, the proposed state has acceptance probability one for sampling $\pi_{\ell-1}(\boldsymbol{v}_{\ell-1}|\boldsymbol{y})$. To sample $\pi_{\ell}(\boldsymbol{v}_{\ell}|\boldsymbol{y})$, we consider a factorised proposal in conditional form,

$$q\big(\boldsymbol{v}_{\ell}' \,\big|\, \boldsymbol{v}_{\ell}^{*}\big) = q\big(\boldsymbol{v}_{\ell,c}'\,,\, \boldsymbol{v}_{\ell,f}' \,\big|\, \boldsymbol{v}_{\ell}^{*}\big) = \widetilde{\pi}_{\ell-1}\big(\boldsymbol{v}_{\ell,c}'\big)\, q\big(\boldsymbol{v}_{\ell,f}' \,\big|\, \boldsymbol{v}_{\ell}^{*}, \boldsymbol{v}_{\ell,c}'\big), \tag{5.2}$$

where the proposal candidate $\boldsymbol{v}_{\ell,c}'$ is set to be identical to the candidate $\boldsymbol{v}_{\ell-1}'$ from the previous level and where the proposal candidate $\boldsymbol{v}_{\ell}'$ conditioned on $\boldsymbol{v}_{\ell}^{*}$ can then be expressed as

$$\boldsymbol{v}_{\ell,c}' = \boldsymbol{v}_{\ell-1}' \qquad \text{(copy from level } \ell-1 \text{ proposal)}, \tag{5.3}$$

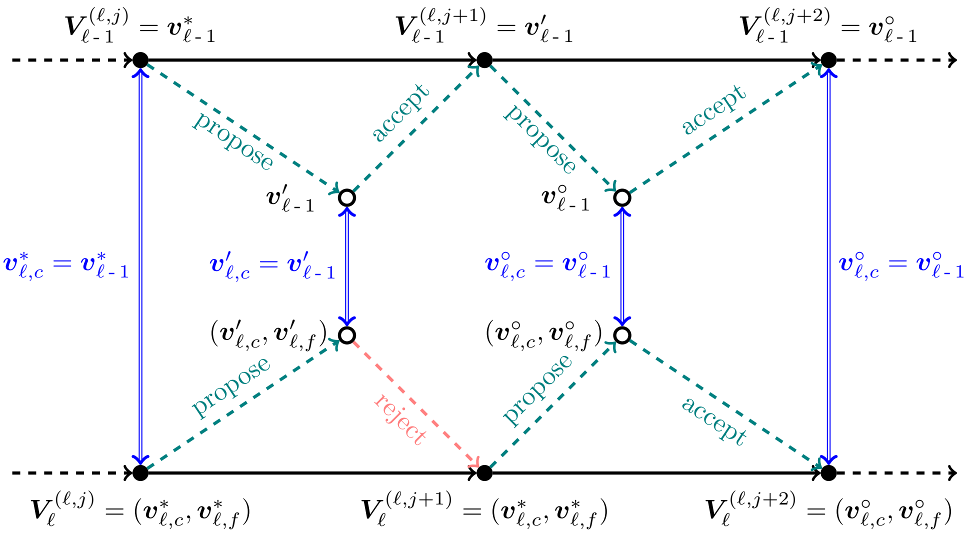
$$\boldsymbol{v}_{\ell,f}' \sim q\big(\,\cdot\,|\, \boldsymbol{v}_{\ell}^{*}, \boldsymbol{v}_{\ell,c}'\big) \qquad \text{(conditional proposal)}. \tag{5.4}$$

Corollary 5.2. *Using the factorised proposal* (5.2) *to sample from the level-*ℓ *posterior* $\pi_{\ell}\big(\boldsymbol{v}_{\ell}\,|\,\boldsymbol{y}\big)$*, the acceptance probability takes the form*

$$\alpha_{\ell}^{\mathrm{ML}}(\boldsymbol{v}_{\ell}^{*}, \boldsymbol{v}_{\ell}') = \min\left\{1, \frac{\pi_{\ell}\big(\boldsymbol{v}_{\ell}'|\boldsymbol{y}\big)\,\pi_{\ell-1}\big(\boldsymbol{v}_{\ell-1}^{*}|\boldsymbol{y}\big)}{\pi_{\ell}\big(\boldsymbol{v}_{\ell}^{*}|\boldsymbol{y}\big)\,\pi_{\ell-1}\big(\boldsymbol{v}_{\ell-1}'|\boldsymbol{y}\big)}\,\frac{q\big(\boldsymbol{v}_{\ell,f}^{*}|\boldsymbol{v}_{\ell}', \boldsymbol{v}_{\ell-1}^{*}\big)}{q\big(\boldsymbol{v}_{\ell,f}'|\boldsymbol{v}_{\ell}^{*}, \boldsymbol{v}_{\ell-1}'\big)}\right\}. \tag{5.5}$$

Proof. The result follows directly from Proposition 5.1. □

Figure 1 shows a schematic of the coupling strategy. The double dashed arrows represent the coupling of two MCMC states across the levels or the coupling of two proposal candidates across the levels. The dashed arrows represent the proposal and acceptance/rejection steps. The top half represents the Markov chain on level $\ell-1$, where all the proposed states are



$\boldsymbol{V}_{\ell-1}^{(\ell,j)} = \boldsymbol{v}_{\ell-1}^{*}$
$\boldsymbol{V}_{\ell-1}^{(\ell,j+1)} = \boldsymbol{v}_{\ell-1}'$
$\boldsymbol{V}_{\ell-1}^{(\ell,j+2)} = \boldsymbol{v}_{\ell-1}^{\circ}$
propose
accept
propose
accept
$\boldsymbol{v}_{\ell-1}'$
$\boldsymbol{v}_{\ell-1}^{\circ}$
$\boldsymbol{v}_{\ell,c}^{*} = \boldsymbol{v}_{\ell-1}^{*}$
$\boldsymbol{v}_{\ell,c}' = \boldsymbol{v}_{\ell-1}'$
$\boldsymbol{v}_{\ell,c}^{\circ} = \boldsymbol{v}_{\ell-1}^{\circ}$
$\boldsymbol{v}_{\ell,c}^{\circ} = \boldsymbol{v}_{\ell-1}^{\circ}$
$(\boldsymbol{v}_{\ell,c}', \boldsymbol{v}_{\ell,f}')$
$(\boldsymbol{v}_{\ell,c}^{\circ}, \boldsymbol{v}_{\ell,f}^{\circ})$
propose
reject
propose
accept
$\boldsymbol{V}_{\ell}^{(\ell,j)} = (\boldsymbol{v}_{\ell,c}^{*}, \boldsymbol{v}_{\ell,f}^{*})$
$\boldsymbol{V}_{\ell}^{(\ell,j+1)} = (\boldsymbol{v}_{\ell,c}^{*}, \boldsymbol{v}_{\ell,f}^{*})$
$\boldsymbol{V}_{\ell}^{(\ell,j+2)} = (\boldsymbol{v}_{\ell,c}^{\circ}, \boldsymbol{v}_{\ell,f}^{\circ})$

FIG. 1. *A Diagram showing the coupling strategy.*

accepted. The bottom half represents the Markov chain on level ℓ, where all the proposal candidates are coupled. All states on level ℓ that follow the acceptance of a proposal candidate are coupled with the corresponding states on level ℓ - 1.

5.2. Coupled DILI proposal. The discretised DILI proposal (2.11) is

$$\boldsymbol{v}_\ell' = \mathrm{A}_\ell \boldsymbol{v}_\ell^* + \mathrm{B}_\ell \boldsymbol{\xi}_\ell, \quad \text{where} \quad \boldsymbol{\xi}_\ell \sim \mathcal{N}\big(0, \mathrm{I}_{R_\ell}\big), \tag{5.6}$$

as it was introduced in [10]. Suppose we have a LIS basis $\Psi_{\ell,r} \in \mathbb{R}^{R_\ell \times r_\ell}$. By treating the likelihood-informed parameter directions and the prior-dominated directions separately, we can construct the matrices A_ℓ and B_ℓ as

$$\mathrm{A}_\ell = \Psi_{\ell,r}\, \mathrm{A}_{\ell,r}\, \Psi_{\ell,r}^\top + a_\perp (\mathrm{I}_{R_\ell} - \Pi_\ell) \in \mathbb{R}^{R_\ell \times R_\ell}, \tag{5.7}$$

$$\mathrm{B}_\ell^2 = \Psi_{\ell,r}\, \mathrm{B}_{\ell,r}^2\, \Psi_{\ell,r}^\top + b_\perp^2 (\mathrm{I}_{R_\ell} - \Pi_\ell) \in \mathbb{R}^{R_\ell \times R_\ell}, \tag{5.8}$$

where $A_{\ell,r}, B_{\ell,r} \in \mathbb{R}^{r_\ell \times r_\ell}$, $a_\perp$ and $b_\perp \in \mathbb{R}$ and $\Pi_\ell = \Psi_{\ell,r}\Psi_{\ell,r}^\top$ are rank-r_ℓ orthogonal projectors.

COROLLARY 5.3. *In the proposal* (5.6)*, suppose that* $\mathrm{A}_{\ell,r}, \mathrm{B}_{\ell,r} \in \mathbb{R}^{r_\ell \times r_\ell}$ *are non-singular matrices satisfying* $\mathrm{A}_{\ell,r}^2 + \mathrm{B}_{\ell,r}^2 = \mathrm{I}_{\ell,r}$*, and* $a_\perp$ *and* $b_\perp$ *are scalars satisfying* $a_\perp^2 + b_\perp^2 = 1$*. Then, the corresponding proposal distribution* $q(\boldsymbol{v}_\ell'|\boldsymbol{v}_\ell^*)$ *satisfies the conditions of Theorem* 2.3 *and has the prior as its invariant measure, i.e., this proposal has acceptance probability one if we use it to sample the prior. The acceptance probability as samples from* $\pi_\ell\big(\boldsymbol{v}_\ell|\boldsymbol{y}\big)$ *is*

$$\alpha\big(\boldsymbol{v}_\ell^*, \boldsymbol{v}_\ell'\big) = \min\Big\{1, \exp\Big[\eta_\ell\big(\boldsymbol{v}_\ell^*; \boldsymbol{y}\big) - \eta_\ell\big(\boldsymbol{v}_\ell'; \boldsymbol{y}\big)\Big]\Big\}. \tag{5.9}$$

Proof. Given $\mathrm{A}_{\ell,r}^2 + \mathrm{B}_{\ell,r}^2 = \mathrm{I}_{\ell,r}$, the symmetric matrices $\mathrm{A}_{\ell,r}$ and $\mathrm{B}_{\ell,r}$ can be simultaneously diagonalised under some orthogonal transformation. Thus, the operators A_ℓ and B_ℓ can be simultaneously diagonalised, where the eigenspectrum of A_ℓ consists of the eigenvalues of $\mathrm{A}_{\ell,r}$ and $a_\perp$, and the same applies to B_ℓ. This way, it is easy to check that the proposal distribution $q(\boldsymbol{v}_\ell'|\boldsymbol{v}_\ell^*)$ has the prior as invariant measure and that the conditions of Theorem 2.3 are satisfied. The form of the acceptance probability to sample from $\pi_\ell\big(\boldsymbol{v}_\ell|\boldsymbol{y}\big)$ directly follows from the acceptance probability defined in Theorem 2.3. □

We use the empirical posterior covariance, commonly used in adaptive MCMC [30, 16, 15] to construct matrices $\mathrm{A}_{\ell,r}$ and $\mathrm{B}_{\ell,r}$ for our DILI proposal (5.6). On each level, the empirical covariance matrix $\Sigma_{\ell,r} \in \mathbb{R}^{r_\ell \times r_\ell}$ is estimated from past posterior samples projected onto the LIS. Given a jump size Δt, we can then define the matrices $\mathrm{A}_{\ell,r}$ and $\mathrm{B}^2_{\ell,r}$ by

$$\begin{aligned}\mathrm{A}_{\ell,r} &= (2\mathrm{I}_{r_\ell}+\Delta t\Sigma_{\ell,r})^{-1}(2\mathrm{I}_{r_\ell}-\Delta t\Sigma_{\ell,r}) = \mathrm{I}_{r_\ell} - 2\big(\mathrm{I}_{r_\ell}+\tfrac{\Delta t}{2}\Sigma_{\ell,r}\big)^{-1}\big(\tfrac{\Delta t}{2}\Sigma_{\ell,r}\big),\\ \mathrm{B}^2_{\ell,r} &= \mathrm{I}_{r_\ell} - \mathrm{A}^2_{\ell,r} = 4\left(2\,\mathrm{I}_{r_\ell} + \big(\tfrac{\Delta t}{2}\Sigma_{\ell,r}\big)^{-1} + \tfrac{\Delta t}{2}\Sigma_{\ell,r}\right)^{-1},\end{aligned}$$

respectively. The operators $\mathrm{A}_{\ell,r}$ and $\mathrm{B}_{\ell,r}$ satisfy $\mathrm{A}^2_{\ell,r} + \mathrm{B}^2_{\ell,r} = \mathrm{I}_{\ell,r}$ by construction.

5.2.1. Conditional DILI proposal. On level 0, the vanilla DILI proposal (cf. [10]) can be used to sample the Markov chain with invariant distribution $\pi_0(\boldsymbol{v}_0|\boldsymbol{y})$. On level ℓ, to simulate coupled Markov chains, the independent proposal $\widetilde{\pi}_{\ell-1}(\boldsymbol{v}_{\ell-1})$, as defined in (5.1), is used to sample the posterior $\pi_{\ell-1}(\boldsymbol{v}_{\ell-1}|\boldsymbol{y})$, while the factorised proposal

$$q(\boldsymbol{v}'_\ell \mid \boldsymbol{v}^*_\ell) = \widetilde{\pi}_{\ell-1}(\boldsymbol{v}'_{\ell,c})\, q(\boldsymbol{v}'_{\ell,f} \mid \boldsymbol{v}^*_\ell, \boldsymbol{v}'_{\ell,c}),$$

is used to sample $\pi_\ell(\boldsymbol{v}_\ell|\boldsymbol{y})$. We now use DILI to generate the fine components $\boldsymbol{v}'_{\ell,f}$ of the proposal candidate and thus to fix the conditional probability $q(\boldsymbol{v}'_{\ell,f}|\boldsymbol{v}^*_\ell, \boldsymbol{v}'_{\ell,c})$. Defining the precision matrix

$$\mathrm{P}_\ell = \mathrm{B}_\ell^{-2} = \Psi_{\ell,r}\,\mathrm{B}^{-2}_{\ell,r}\,\Psi^\top_{\ell,r} + b_\perp^{-2}(\mathrm{I}_{R_\ell} - \Pi_\ell), \tag{5.10}$$

the DILI proposal (5.6) can be split as follows:

$$\begin{bmatrix}\boldsymbol{v}'_{\ell,c}\\ \boldsymbol{v}'_{\ell,f}\end{bmatrix} = \mathrm{A}_\ell \boldsymbol{v}^*_\ell + \begin{bmatrix}\boldsymbol{r}_{\ell,c}\\ \boldsymbol{r}_{\ell,f}\end{bmatrix}, \quad \begin{bmatrix}\boldsymbol{r}_{\ell,c}\\ \boldsymbol{r}_{\ell,f}\end{bmatrix} \sim \mathcal{N}\Big(0, \begin{bmatrix}\mathrm{P}_{\ell,cc} & \mathrm{P}_{\ell,cf}\\ \mathrm{P}_{\ell,fc} & \mathrm{P}_{\ell,ff}\end{bmatrix}^{-1}\Big), \tag{5.11}$$

where the partitions of the vectors and of the matrix P_ℓ correspond to the parameter coordinates shared with level ℓ-1 and the refined parameter coordinates on level ℓ.

Definition 5.4. *The following procedure is used to draw candidates from the factorised proposal distribution $\widetilde{\pi}_{\ell-1}(\boldsymbol{v}'_{\ell,c})\, q(\boldsymbol{v}'_{\ell,f}|\boldsymbol{v}^*_\ell, \boldsymbol{v}'_{\ell,c})$ such that the DILI proposal in (5.11) is used as the conditional distribution $q(\boldsymbol{v}'_{\ell,f}|\boldsymbol{v}^*_\ell, \boldsymbol{v}'_{\ell,c})$:*

1. *The proposed candidate $\boldsymbol{v}'_{\ell,c}$ is randomly selected from the sample set $\mathcal{V}_{\ell-1}$, as defined in Proposition 5.1. Note that the sample $\boldsymbol{v}'_{\ell,c}$ is identical to the proposal candidate $\boldsymbol{v}'_{\ell-1}$ used for sampling $\pi_{\ell-1}(\boldsymbol{v}_{\ell-1}|\boldsymbol{y})$, c.f. (5.3).*
2. *Since $\boldsymbol{v}'_{\ell,c}$ and $\boldsymbol{v}^*_\ell$ are known, the variable $\boldsymbol{r}_{\ell,c}$ can be determined from (5.11) as*
$$\boldsymbol{r}_{\ell,c} = \boldsymbol{v}'_{\ell,c} - \Theta^\top_{\ell,c}\,\mathrm{A}_\ell\,\boldsymbol{v}^*_\ell.$$
3. *Then, we can draw a random variable $\boldsymbol{r}_{\ell,f}$ conditioned on $\boldsymbol{r}_{\ell,c}$ such that the joint distribution of $(\boldsymbol{r}_{\ell,c}, \boldsymbol{r}_{\ell,f})$ follows the Gaussian $\mathcal{N}(0, \mathrm{P}_\ell^{-1})$. Due to (5.11), the refined part of the proposed candidate $\boldsymbol{v}'_{\ell,f}$ satisfies*
$$\boldsymbol{v}'_{\ell,f} = \Theta^\top_{\ell,f}\,\mathrm{A}_\ell\,\boldsymbol{v}^*_\ell + \boldsymbol{r}_{\ell,f}, \quad \boldsymbol{r}_{\ell,f} \sim \mathcal{N}\big(-\mathrm{P}^{-1}_{\ell,ff}\mathrm{P}_{\ell,fc}\boldsymbol{r}_{\ell,c}, \mathrm{P}^{-1}_{\ell,ff}\big), \tag{5.12}$$
where $\boldsymbol{r}_{\ell,f}$ is a conditional Gaussian random vector.

Corollary 5.5. *Using the above procedure to draw candidates from the factorised proposal distribution $\widetilde{\pi}_{\ell-1}\big(\boldsymbol{v}'_{\ell,c}\big)q\big(\boldsymbol{v}'_{\ell,f}|\boldsymbol{v}^*_\ell, \boldsymbol{v}'_{\ell,c}\big)$, the acceptance probability to sample from the posterior distribution $\pi_\ell(\boldsymbol{v}_\ell|\boldsymbol{y})$ is*

$$\alpha^{\mathrm{ML}}_\ell(\boldsymbol{v}^*_\ell, \boldsymbol{v}'_\ell) = \min\Big\{1, \exp\Big[\Big(\eta_\ell(\boldsymbol{v}^*_\ell;\boldsymbol{y}) - \eta_{\ell-1}(\boldsymbol{v}^*_{\ell-1};\boldsymbol{y})\Big) - \Big(\eta_\ell(\boldsymbol{v}'_\ell;\boldsymbol{y}) - \eta_{\ell-1}(\boldsymbol{v}'_{\ell-1};\boldsymbol{y})\Big)\Big]\Big\}.$$

Proof. See Appendix B. ∎

5.2.2. Generating conditional samples. The computational cost of the coupling procedure is dictated by the multiplication with A_ℓ in Step 2 and the generation of conditional proposal samples in Step 3. The multiplication with A_ℓ has a computational complexity of $\mathcal{O}(\sum_{j=0}^{\ell} R_j s_j)$ using the low-rank representation (5.7) and the upper-triangular hierarchical

LIS basis in (4.11), which has the form

$$\Psi_{\ell,r} = [\Psi_{\ell,c}, \Psi_{\ell,f}] = \begin{bmatrix} \Psi_{\ell-1,r} & \mathrm{Z}_{\ell,c} \\ 0 & \mathrm{Z}_{\ell,f} \end{bmatrix}.$$

We can also exploit the hierarchical LIS to reduce the computational cost of generating conditional proposal samples. As shown in Equation (5.10), given the LIS basis $\Psi_{\ell,r}$, the precision matrix P_ℓ is dictated by the matrix $\mathrm{B}_{\ell,r}^{-2}$, which has the block form

$$\mathrm{B}_{\ell,r}^{-2} = \begin{bmatrix} \Xi_{\ell,cc} & \Xi_{\ell,cf} \\ \Xi_{\ell,fc} & \Xi_{\ell,ff} \end{bmatrix}, \tag{5.13}$$

corresponding to the splitting of the enriched LIS basis into $\Psi_{\ell,c}$ and $\Psi_{\ell,f}$. Generating conditional proposal samples only involves the blocks $\mathrm{P}_{\ell,ff}$ and $\mathrm{P}_{\ell,fc}$ in the matrix P_ℓ, i.e.,

$$\mathrm{P}_{\ell,ff} = \mathrm{Z}_{\ell,f} \left(\Xi_{\ell,ff} - b_\perp^{-2}\, \mathrm{I} \right) \mathrm{Z}_{\ell,f}^\top + b_\perp^{-2}\, \mathrm{I}_{\ell,f}\,, \tag{5.14}$$

$$\mathrm{P}_{\ell,fc} = \mathrm{Z}_{\ell,f} \Xi_{\ell,fc} \Psi_{\ell-1,r}^\top + \mathrm{Z}_{\ell,f} \Xi_{\ell,ff} \mathrm{Z}_{\ell,c}^\top - b_\perp^{-2} \mathrm{Z}_{\ell,f} \mathrm{Z}_{\ell,c}^\top\,, \tag{5.15}$$

which in turn only require the blocks $\Xi_{\ell,fc} \in \mathbb{R}^{s_\ell \times r_{\ell-1}}$ and $\Xi_{\ell,ff} \in \mathbb{R}^{s_\ell \times s_\ell}$ in the matrix $\mathrm{B}_{\ell,r}^{-2}$.

We derive low-rank operations to avoid the direct inversion or factorisation of the matrices $\mathrm{P}_{\ell,ff}$ and $\mathrm{P}_{\ell,fc}$ in the generation of conditional samples and to reduce the computational cost. Suppose the block $\mathrm{Z}_{\ell,f} \in \mathbb{R}^{(R_\ell - R_{\ell-1}) \times s_\ell}$ has the thin QR factorisation

$$\mathrm{Z}_{\ell,f} = \mathrm{U}_\ell \mathrm{T}_\ell, \tag{5.16}$$

where U_ℓ has orthonormal columns and T_ℓ is upper triangular. Then the matrix $\mathrm{P}_{\ell,ff}$ can be expressed as

$$\mathrm{P}_{\ell,ff} = b_\perp^{-2} \Big(\mathrm{U}_\ell \big(\mathrm{T}_\ell (b_\perp^2\, \Xi_{\ell,ff} - \mathrm{I}) \mathrm{T}_\ell^\top \big) \mathrm{U}_\ell^\top + \mathrm{I}_{\ell,f} \Big).$$

Computing the $s_\ell \times s_\ell$ eigendecomposition

$$\mathrm{T}_\ell (b_\perp^2\, \Xi_{\ell,ff} - \mathrm{I}) \mathrm{T}_\ell^\top = \mathrm{W}_\ell \mathrm{D}_\ell \mathrm{W}_\ell^\top, \tag{5.17}$$

where W_ℓ and D_ℓ are respectively orthogonal and diagonal matrices, we have

$$\mathrm{P}_{\ell,ff} = b_\perp^{-2} \Big(\Phi_\ell\, \mathrm{D}_\ell\, \Phi_\ell^\top + \mathrm{I}_{\ell,f} \Big), \quad \text{with} \quad \Phi_\ell := \mathrm{U}_\ell \mathrm{W}_\ell.$$

Note that $\Phi_\ell \in \mathbb{R}^{(R_\ell - R_{\ell-1}) \times s_\ell}$ has orthonormal columns, so that

$$\mathrm{P}_{\ell,ff}^{-1} \mathrm{P}_{\ell,fc} = b_\perp^2\, \Phi_\ell \underbrace{\big((\mathrm{D}_\ell + \mathrm{I})^{-1} \mathrm{W}_\ell^\top \mathrm{T}_\ell \big)}_{s_\ell \times s_\ell} \underbrace{\big(\Xi_{\ell,fc} \Psi_{\ell-1,r}^\top + \Xi_{\ell,ff} \mathrm{Z}_{\ell,c}^\top - b_\perp^{-2} \mathrm{Z}_{\ell,c}^\top \big)}_{s_\ell \times R_{\ell-1}}, \tag{5.18}$$

$$\mathrm{P}_{\ell,ff}^{-\frac{1}{2}} = b_\perp \left(\Phi_\ell \underbrace{\big((\mathrm{D}_\ell + \mathrm{I})^{-\frac{1}{2}} - \mathrm{I} \big)}_{s_\ell \times s_\ell} \Phi_\ell^\top + \mathrm{I}_{\ell,f} \right). \tag{5.19}$$

Using these representations of the matrices $\mathrm{P}_{\ell,ff}^{-1} \mathrm{P}_{\ell,fc}$ and $\mathrm{P}_{\ell,ff}^{-1/2}$, the conditional Gaussian in (5.12) can be simulated efficiently using

$$\boldsymbol{r}_{\ell,f} | \boldsymbol{r}_{\ell,c} = -\mathrm{P}_{\ell,ff}^{-1}\, \mathrm{P}_{\ell,fc}\, \boldsymbol{r}_{\ell,c} + \mathrm{P}_{\ell,ff}^{-\frac{1}{2}} \xi, \quad \text{where} \quad \xi \sim \mathcal{N}\big(0, \mathrm{I}_{(R_\ell - R_{\ell-1})}\big). \tag{5.20}$$

The associated computational cost is $\mathcal{O}(R_\ell s_\ell)$.

5.3. Final MLDILI algorithm. Here, we assemble all the elements of the multilevel DILI method defined in previous sections in algorithmic form. For the base level ($\ell = 0$), the LIS construction and the DILI–MCMC sampling are presented in Algorithm 5.1. The recursive LIS construction and the coupled DILI–MCMC are presented in Algorithm 5.2.

In both algorithms, we need to use both the LIS basis $\Psi_{\ell,r}$ and an empirical covariance matrix $\Sigma_{\ell,r}$ projected onto the LIS to define operators A_ℓ and B_ℓ in the DILI proposal. Computing the LIS basis needs some reference distribution $p_\ell^*(\cdot)$. We employ the Laplace approximation

Algorithm 5.1 Base level algorithm.

Input: A set of samples $\mathcal{W}_0 = \{\boldsymbol{v}_0^{(k)}\}_{k=1}^{K_0}$ drawn from the base level reference $p_0^*(\cdot)$, the number of MCMC iterations N_0, and an initial MCMC state $\boldsymbol{V}_0^{(0)}$.

Output: A LIS basis $\Psi_{0,r}$ and a Markov chain of posterior samples $\mathcal{V}_0 = \{\boldsymbol{V}_0^{(j)}\}_{j=1}^{N_0}$.

```
procedure Base level LIS and MCMC
    Use $\mathcal{W}_0$ to solve the eigenproblem in (4.5) to obtain the base level LIS basis $\Psi_{0,r}$.
    Estimate the empirical covariance matrix $\Sigma_{0,r}$ from the samples in $\mathcal{W}_0$ and define the
      operators $\mathrm{A}_0$ and $\mathrm{B}_0$ as in (5.7)–(5.8).
    for $j = 1, \ldots, N_0$ do
        Propose a candidate $\boldsymbol{v}_0'$ using the base level proposal in (5.6).
        Compute the acceptance probability $\alpha(\boldsymbol{V}_0^{(j-1)}, \boldsymbol{v}_0')$ defined in (5.9).
        With probability $\alpha(\boldsymbol{V}_0^{(j-1)}, \boldsymbol{v}_0')$, set $\boldsymbol{V}_0^{(j)} = \boldsymbol{v}_0'$, otherwise set $\boldsymbol{V}_0^{(j)} = \boldsymbol{V}_0^{(j-1)}$.
    end for
end procedure
```

Note: Optionally, $\Sigma_{0,r}$, A_0 and B_0 can be adaptively updated within the MCMC after a pre-fixed number of iterations, cf. [1, 16].

Algorithm 5.2 Level–ℓ algorithm.

Input: A set of samples $\mathcal{W}_\ell = \{\boldsymbol{v}_\ell^{(k)}\}_{k=1}^{K_\ell}$ from the level–ℓ reference $p_\ell^*(\cdot)$, the number of MCMC iterations N_ℓ, a set of MCMC samples $\mathcal{V}_{\ell-1} = \{\boldsymbol{v}_{\ell-1}^{(j)}\}_{j=1}^{N_{\ell-1}}$ on level $\ell-1$ and an initial MCMC state $\boldsymbol{V}_\ell^{(0)}$.

Output: A LIS basis $\Psi_{\ell,r}$ and a Markov chain of posterior samples $\mathcal{V}_\ell = \{\boldsymbol{V}_\ell^{(j)}\}_{j=1}^{N_\ell}$.

```
procedure Level–$\ell$ LIS and MCMC
    Lift previous LIS basis, $\Psi_{\ell,c} = \Theta_{\ell,c}\,\Psi_{\ell-1,r}$.
    Use $\mathcal{W}_\ell$ to solve the eigenproblem in (4.8) to obtain the auxiliary LIS vectors $\Psi_{\ell,f}$.
    Estimate the empirical covariance matrix $\Sigma_{\ell,r}$ from the samples in $\mathcal{W}_\ell$ and define the
      operators $\mathrm{A}_\ell$ and $\mathrm{B}_\ell$ as in (5.7)–(5.8).
    Compute the matrices $\mathrm{P}_{\ell,ff}^{-1}\,\mathrm{P}_{\ell,fc}$ and $\mathrm{P}_{\ell,ff}^{-\frac{1}{2}}$ as in (5.18)-(5.19).
    for $j = 1, \ldots, N_\ell$ do
        Propose a candidate $\boldsymbol{v}_\ell' = (\boldsymbol{v}_{\ell,c}', \boldsymbol{v}_{\ell,f}')$ using Definition 5.4, which needs $\mathcal{V}_{\ell-1}$.
        Compute the acceptance probability $\alpha_\ell^{\mathrm{ML}}(\boldsymbol{V}_\ell^{(j-1)}, \boldsymbol{v}_\ell')$ defined in Corollary 5.5.
        With probability $\alpha_\ell^{\mathrm{ML}}(\boldsymbol{V}_\ell^{(j-1)}, \boldsymbol{v}_\ell')$, set $\boldsymbol{V}_\ell^{(j)} = \boldsymbol{v}_\ell'$, otherwise set $\boldsymbol{V}_\ell^{(j)} = \boldsymbol{V}_\ell^{(j-1)}$.
    end for
end procedure
```

Note: Optionally, $\Sigma_{\ell,r}$, A_ℓ, B_ℓ, and the matrices $\mathrm{P}_{\ell,ff}^{-1}\,\mathrm{P}_{\ell,fc}$ and $\mathrm{P}_{\ell,ff}^{-\frac{1}{2}}$ can be adaptively updated within MCMC after a pre-fixed number of iterations.

to the posterior (e.g., [27, 29]). This way, all the samples from $p_\ell^*(\cdot)$ can be generated in parallel and prior to the DILI–MCMC simulation. The empirical covariance $\Sigma_{\ell,r}$ can be estimated using either samples drawn from the reference distribution (before the start of MCMC) or adaptively using posterior samples generated in MCMC. The latter option is the classical adaptive MCMC method [16]. The adaptation of $\Sigma_{\ell,r}$ is optional in Algorithms 5.1 and 5.2. Similar to the adaptation of the covariance, the LIS basis can also be adaptively updated using newly generated posterior samples during MCMC simulations. The implementation details for the adaptation of the LIS can be found in Algorithm 1 of [10].

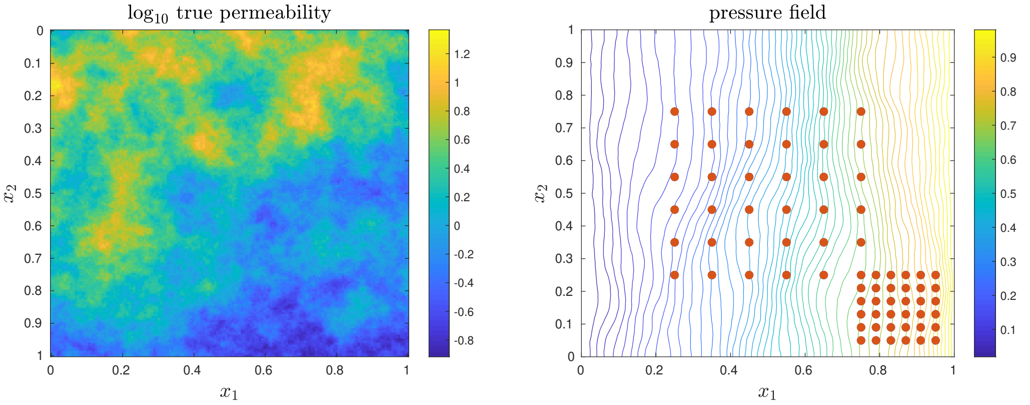


FIG. 2. *Setup of elliptic inverse problem. Left: "true" permeability field used for generating the synthetic data set. Right: observation sensors (red dots) and pressure field corresponding to "true" permeability field.*

6. Numerical experiments.

In this section, we test our algorithms on test problem involving an elliptic PDE with random coefficients. The setup is described in section 6.1, while numerical comparisons are given in section 6.2.

6.1. Setup.

We consider an elliptic PDE in a domain $\Omega = [0,1]^2$ with boundary $\partial\Omega$, which models, e.g., the pressure distribution $p(\boldsymbol{x})$ of a stationary fluid in a porous medium described by a spatially heterogeneous permeability field $k(\boldsymbol{x})$. Here, $\boldsymbol{x} \in \Omega$ denotes the spatial coordinate and $\boldsymbol{n}(\boldsymbol{x})$ denotes the outward normal vector along the boundary.

The goal is to recover the permeability field from pressure observations. We assume that the permeability field follows a log–normal prior, and thus we denote the permeability field by $k(\boldsymbol{x}) = \exp(u(\boldsymbol{x}))$, where $u(\boldsymbol{x})$ is a random function equipped with a Gaussian process prior. In this setting, the pressure $p(\boldsymbol{x})$ depends implicitly on the (random) realisation of $u(\boldsymbol{x})$.

For a given realisation $u(\boldsymbol{x})$, the pressure satisfies the elliptic PDE

$$-\nabla \cdot \left(e^{u(\boldsymbol{x})} \nabla p(\boldsymbol{x})\right) = 0, \quad \boldsymbol{x} \in \Omega. \tag{6.1}$$

On the left and right boundaries, we specify Dirichlet boundary conditions, while on the top and bottom we assume homogeneous Neumann boundary conditions:

$$\begin{cases} p(\boldsymbol{x}) = 0, & \text{for } \ \boldsymbol{x} \in \partial\Omega_{\text{left}}\,, \\ p(\boldsymbol{x}) = 1, & \text{for } \ \boldsymbol{x} \in \partial\Omega_{\text{right}} \ \text{ and} \\ e^{u(\boldsymbol{x})}\nabla p(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0, & \text{for } \ \boldsymbol{x} \in \{\partial\Omega_{\text{top}}, \partial\Omega_{\text{bottom}}\}. \end{cases} \tag{6.2}$$

As the quantity of interest, we define the outflow through the left vertical boundary, i.e.

$$Q(u) = -\int_0^1 e^{u(\boldsymbol{x})} \frac{\partial p(\boldsymbol{x})}{\partial x_1}\Big|_{x_1=0} dx_2 = -\int_\Omega e^{u(\boldsymbol{x})} \nabla p(\boldsymbol{x}) \cdot \nabla \varphi(\boldsymbol{x})\, d\boldsymbol{x}\,, \tag{6.3}$$

where $\varphi(\boldsymbol{x})$ is a linear function taking value one on $\partial\Omega_{\text{left}}$ and value zero on $\partial\Omega_{\text{right}}$, as suggested in [36].

The Gaussian process prior for $u(\boldsymbol{x})$ is defined by the exponential kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-5|\boldsymbol{x} - \boldsymbol{x}'|)$. Figure 2 (left) displays the true (synthetic) permeability field in $\log_{10}$ scale. Noisy observations of the pressure field are collected from 71 sensors located as in Figure 2 (right), with a signal-to-noise ratio 50. A likelihood function can then be defined as in (2.3), which, together with the prior, characterizes the posterior distribution in (2.1).

In practice, (6.1)–(6.3) has to be solved numerically. We use standard, piecewise bilinear finite elements on a hierarchy of nested Cartesian grids with mesh size $h_\ell = \frac{1}{20} \times 2^{-\ell}$, for $\ell =$

Level	0	1	2	3
Non-recursive	80	91	97	100
Recursive (added on level ℓ)	80	21	19	12
Recursive (total)	80	101	120	132
Storage reduction factor	1	0.74	0.60	0.43

Table 1

LIS dimensions: Results of non-recursive construction (single-level LIS for each ℓ) reported in first row; for the recursive construction, the number of vectors added on the current level and the total LIS dimension are given in the second and third row, respectively; the fourth row displays the storage reduction factor for the recursive procedure at each level.

	Refined parameters		D_ℓ	
Level	MLDILI	MLpCN	MLDILI	MLpCN
0	34	4300	9.0	4100
1	11	45	4.6	4.9
2	3.6	48	2.4	2.8
3	2.0	24	1.8	1.9

Table 2

Comparison of IACTs of Markov chains generated by MLDILI and MLpCN. This table reports the IACTs of the refined parameters and the level-ℓ correction of the quantity of interest $D_\ell = Q_\ell(\boldsymbol{V}_\ell) - Q_{\ell\text{-}1}(\boldsymbol{V}_{\ell\text{-}1})$.

$0, 1, 2, 3$. Furthermore, we approximate the unknown function $u(\boldsymbol{x})$ by truncated Karhunen-Loève expansions with $R_\ell = 50 + 100 \times 2^\ell$ random modes, respectively.

6.2. Comparisons. In this section, we test and compare our algorithms on the model problem described above. First, we proceed as in section 4 to build a LIS at every level, both with the non-recursive and recursive constructions. Table 1 summarizes the number of basis functions obtained in each case using the truncation threshold $\rho = 10^{-2}$ and the storage reduction factor given by the recursive procedure at each level.

Because the recursive LIS construction recycles LIS bases from previous levels and enriches them with a number of auxiliary LIS vectors on each level, it is expected that the total number of basis functions obtained by the enriching procedure at each level is slightly higher than the direct (spectral) LIS on the same level. However, in the recursive construction, the dimension of the auxiliary set of vectors is expected to decrease as the level increases, requiring less storage and less computational effort on the finer levels, since the posterior distributions were assumed to converge with $\ell \to \infty$. For problems with parametrisations where the parameter dimension increases more rapidly with the discretisation level—e.g., using the same finite element grid to discretise the prior covariance, the setting used in the original DILI paper [10]—we expect the reduction factor to be even smaller.

In the comparison of sampling performances, we denote by **MLpCN** the MLMCMC algorithm using the pCN proposal for the additional parameters on each level (as in [13]), but using the coupling procedure in Proposition 5.1. The MLMCMC algorithm using the recursive LIS and the coupled DILI proposals, as summarised in Algorithms 5.1 and 5.2, is denoted by **MLDILI**. The integrated autocorrelation times of Markov chains constructed by MLpCN and MLDILI are reported in Table 2. The IACTs for two functionals are reported for each algorithm. In the "refined parameters" case, at every level ℓ we report the average IACTs of the refined parameters $\boldsymbol{v}_{\ell,f}$. This quantifies how well the algorithm performs in exploring the posterior distribution. In the second case, we consider the IACT of the level-ℓ corrections of the quantity of interest $D_\ell = Q_\ell(\boldsymbol{V}_\ell) - Q_{\ell\text{-}1}(\boldsymbol{V}_{\ell\text{-}1})$.

In the "refined parameters" case, we observe a significant improvement for MLDILI over

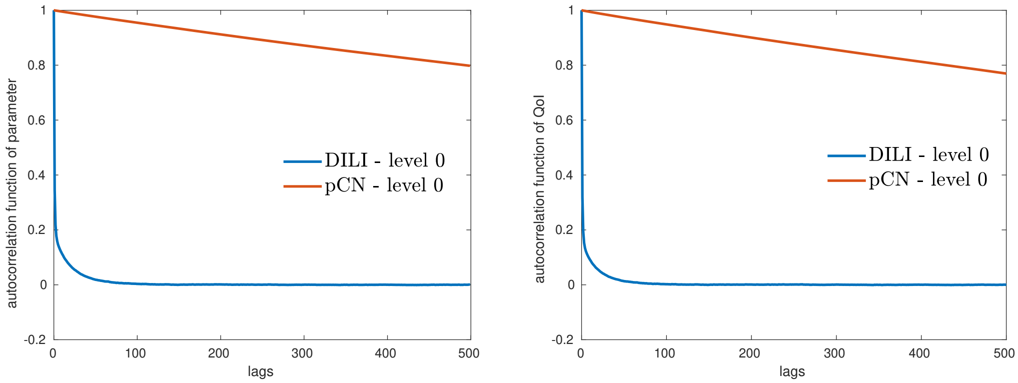


Fig. 3. *IACTs of the chains* $\{(\boldsymbol{V}_0^{(j)})_1\}$ *and* $\{Q_0(\boldsymbol{V}_0^{(j)})\}$ *on the coarsest level (Blue: DILI. Red: pCN).*

MLpCN: the coupled DILI proposal is able to reduce the IACT at every level compared to that obtained by MLpCN. At the base level, DILI is able to reduce the IACT by two orders of magnitude compared to that of pCN. This suggests that coarse parameter modes are very informed by the data, and thus utilising the DILI proposal is highly beneficial. In the case of the quantity of interest, we observe an even more impressive improvement at the base level (a factor of 456!), while the IACTs of MLDILI and MLpCN on the finer levels are comparable. This suggests that the posterior distribution of the chosen quantity of interest (the integrated flux over the boundary) is not affected strongly by the high frequency parameter modes on the finer levels. Nevertheless, in both cases, using DILI provides a huge acceleration compared to pCN. Figure 3 compares the integrated autocorrelation times of DILI and pCN on level 0, for both the first parameter component and the quantity of interest.

The IACTs for the level-ℓ corrections of the quantity of interest in Table 2 suggest that using a mixed strategy—in which one employs the LIS and DILI only at the coarsest level and uses pCN in refined levels—is also a reasonable approach in cases where the important likelihood-informed directions that have any influence on the quantity of interest are already well enough identified in the base-level LIS. We refer to this as the **MLmixed** strategy.

We compare the computational performance of the three multilevel algorithms (MLDILI, MLpCN, MLmixed) with the two single level algorithms using DILI and pCN proposals. The finite element model and all MCMC algorithms are implemented in MATLAB; we use sparse Cholesky factorisation [6] to solve the finite element systems and ARPACK [26] to solve the eigenproblems. All simulations are carried out on a workstation equipped with 28 cores (two Intel Xeon E5-2680 CPUs). The performance of MLmixed is only estimated using the IACTs and the actual computing times measured in the MLDILI and MLpCN runs.

The computational complexities of the five algorithms for approximating $\mathbb{E}_\pi[Q]$ on (discretisation) levels $L = 1, 2$ and 3 with Q defined in (6.3) are compared in Figure 4 (right). In the multilevel estimators, the coarsest level is always $\ell = 0$, so that the number of levels is 2, 3 and 4, respectively. The sampling error tolerance on each level is adapted to the corresponding bias error due to finite element discretisation and parameter truncation, such that the squared bias is equal to the variance of the estimator. The bias errors were estimated beforehand to be 9×10^{-3}, 4×10^{-3}, and 2×10^{-3} on levels $L = 1, 2$ and 3, leading to a total error of 1.27×10^{-2}, 5.7×10^{-3}, 2.8×10^{-3}, respectively. Those bias estimates are plotted in Figure 4 (left) together with estimates of $\mathrm{Var}_{\pi_\ell}(Q_\ell)$ and $\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(Q_\ell - Q_{\ell\text{-}1})$, which suggest that $\theta_b \approx 0.5$ and $\theta_v \approx 0.5$ in Assumptions 2.4(i) and 3.3. This agrees with the theoretical results in [13]. The cost per sample is dominated by the sparse Cholesky factorisation on each level and scales roughly like $\mathcal{O}(M_\ell^{1.2})$, so that $\theta_c \approx 1.2$ in Assumption 2.4(ii). Optimally scaling

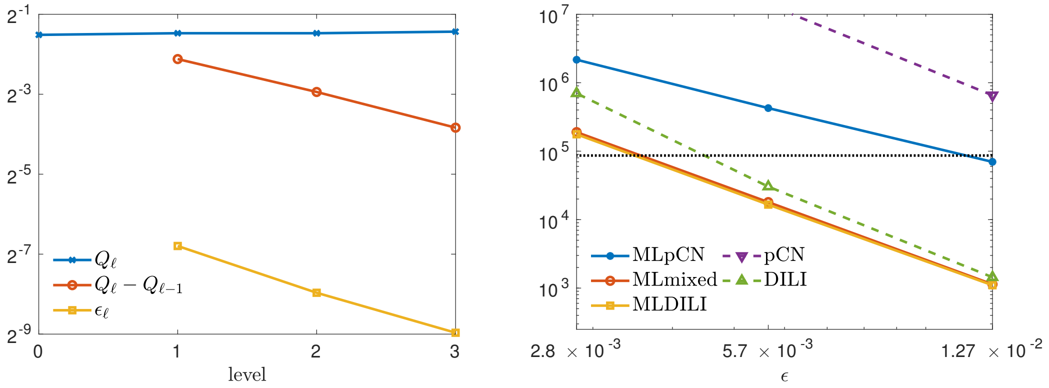


FIG. 4. *Left: the variance* $\mathrm{Var}_{\pi_\ell}(Q_\ell)$ *(blue) and the bias* $\left|\mathbb{E}_{\mu_y}[Q] - \mathbb{E}_{\pi_\ell}[Q_\ell]\right|$ *(yellow) at each level, and the cross-level variances* $\mathrm{Var}_{\Delta_{\ell,\ell\text{-}1}}(Q_\ell - Q_{\ell\text{-}1})$ *(red) used for estimating the CPU time for various MCMC methods. Right: Total CPU time (in seconds) for various methods to achieve different total error tolerances. The LISs are constructed by recycling Cholesky factors. The dotted line represents a CPU day.*

multigrid solvers exist for this model problem, but for the FE problem sizes considered here they are more costly in absolute terms. Moreover, we can also exploit the fact that the adjoint problem is identical to the forward problem here, so that the Cholesky factors can be reused for the adjoint solves required in the LIS construction.

Let us now discuss the results. The single level pCN becomes impractical in this example, since the data is very informative and leads to an extremely low effective sample size. Some of this bad statistical efficiency is inherited by MLpCN, at least in absolute terms, due to the poor effective sample size on level 0. Asymptotically this effect disappears and the rate of growth of the cost is smallest for MLpCN with an observed assymptotic cost of about $\mathcal{O}(\epsilon^{-2.3})$. As observed in [13], this is better than the theoretically predicted asymptotic rate and is likely a pre-asymptotic effect due to the high cost on level 0. Unsurprisingly, given the low IACTs reported in Table 2, the methods based on DILI proposals all perform significantly better. MLDILI and MLmixed perform almost identically, since the corresponding IACTs on all levels are very similar. They are consistently better than single-level DILI and the asymptotic rate of growth of the cost is also better, $\mathcal{O}(\epsilon^{-3.4})$ compared to $\mathcal{O}(\epsilon^{-4.1})$. Both rates are consistent with the theoretically predicted rates in Theorem 3.5, given the estimates for θ_b, θ_v, θ_c above. For the most accurate estimates, MLDILI is almost 4 times faster than DILI, and due to the better asymptotic behaviour this reduction factor will grow as $\varepsilon \to 0$. For grid level $L = 4$, we even expect MLpCN to outperform single-level DILI, but the computational costs of the estimators for higher accuracies are starting to become impractical even using the multilevel acceleration, as the dashed line representing one CPU day in Figure 4 (right) indicates.

The dominating cost in solving the eigenproblems (4.5) and (4.8) is the Cholesky factorisation. As mentioned above, sparse direct solvers are used to solve the stationary forward model and we are able to recycle the Cholesky factors from the forward solve to compute the actions of the adjoint model in (4.5) and (4.8) for each sample. As a result, the computational cost of building the LIS is negligible compared to that of the MCMC simulation here (for both the single level and the recursive construction). This also explains why MLmixed performs almost identically to MLDILI.

However, in many other applications this is not possible due to the high storage cost or when the adjoint is different. Each action of the adjoint problem typically has a comparable cost to solving the forward model in the stationary case. It can even be more expensive than solving the forward model in time-dependent problems. To provide a thorough comparison in that case, we also report the total CPU time of all the estimators in Figure 5 when the

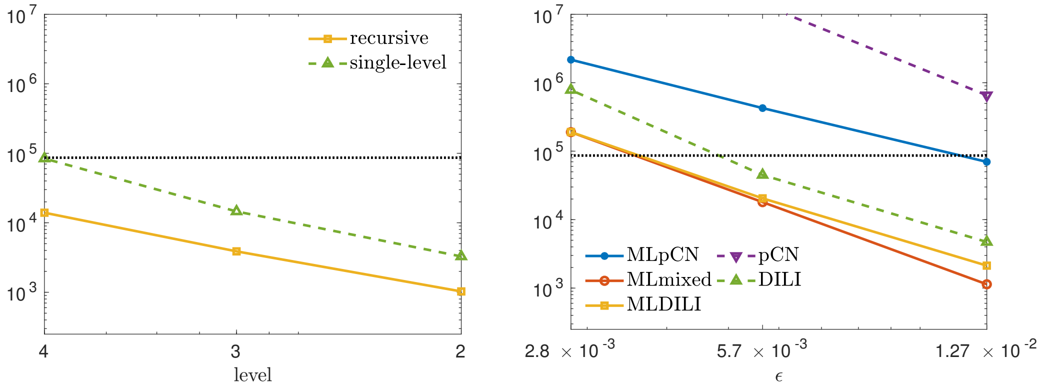


FIG. 5. *Left: Total CPU time (in seconds) for the single level and recursive constructions of the LISs at level* $2, 3$ *and* 4. *Right: Total CPU time (in seconds) for various methods to achieve different error tolerances. The LISs are constructed without recycling Cholesky factors. The dotted line represents a CPU day.*

LIS setup cost is included. Here, we compute both the single level LIS and the recursive LIS without storing the Cholesky factors, to mimic the behaviour in the general, large-scale case. In this setup, we observe that a significant amount of computing effort is spent on building the LIS, and thus MLmixed and MLDILI significantly outperform the single level DILI for all error thresholds. MLmixed is more than 4 times faster than DILI even for the largest error threshold of 1.27×10^{-2}. The construction of the single-level LIS requires two times more CPU time than performing the actual MCMC simulation in that case. In comparison, a significant number of adjoint model solves can be saved by the recursive LIS construction. Furthermore, we do expect that the computational cost for constructing the recursive LIS will stop increasing, since the dimension of the auxiliary LIS will eventually be zero at higher levels. Overall, for large–scale problems where the adjoint cannot be cheaply computed by recycling the forward model simulation, the recursive LIS construction, and hence the MLDILI, is clearly more computationally efficient than the single level DILI.

7. Conclusion.

We integrate the dimension-independent likelihood-informed MCMC from [10] into the multilevel MCMC framework in [13] to improve the computational efficiency of estimating the expectation of functionals of interests over posterior measures. Several novel elements are introduced in this integration. We first design a Rayleigh-Ritz procedure to recursively construct likelihood informed subspaces that exploit the hierarchy of model discretisations. The resulting hierarchical LIS needs lower computational effort to construct and has lower operation cost compared to the original LIS proposed in [11]. Then, we present a new pooling strategy to couple Markov chains on consecutive levels. This enables more flexible parallelisation and management of computing resources. Finally, we design new coupled DILI proposals by exploiting the hierarchical LIS, so that the DILI proposal can be applied in the multilevel MCMC setting. We also demonstrate the efficacy of our integrated approach on a model inverse problem governed by an elliptic PDE.

Appendix A. Proof of Corollary 4.8.

Using Assumption 4.7, the required storage for the hierarchical and for the single-level LIS bases can be bounded by

$$\zeta_{\text{multi}} = \textstyle\sum_{l=0}^{L} R_\ell \, s_\ell \le R_0 \, s_0 \textstyle\sum_{l=0}^{L} e^{(\beta_{\text{p}} - \beta_{\text{r}})\ell} \quad \text{and} \quad \zeta_{\text{single}} = R_L \, r_L \ge c \, R_0 \, s_0 \, e^{\beta_{\text{p}} L}.$$

Thus, the reduction factor satisfies

$$\frac{\zeta_{\text{multi}}}{\zeta_{\text{single}}} \le \frac{1}{c} \, e^{-\beta_{\text{p}} L} \Big(\textstyle\sum_{l=0}^{L} e^{(\beta_{\text{p}} - \beta_{\text{r}})\ell} \Big). \tag{A.1}$$

We first consider the case $\beta_{\mathrm{p}} \neq \beta_{\mathrm{r}}$. Using the property of geometric series, we have

$$\textstyle\sum_{l=0}^{L} e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})\ell} = \frac{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})(L+1)}}{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})}}.$$

For the case $\beta_{\mathrm{p}} < \beta_{\mathrm{r}}$, the reduction factor satisfies

$$\frac{\zeta_{\text{multi}}}{\zeta_{\text{single}}} \leq \frac{1}{c}\, e^{-\beta_{\mathrm{p}} L}\, \frac{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})(L+1)}}{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})}}, \tag{A.2}$$

whereas for $\beta_{\mathrm{p}} > \beta_{\mathrm{r}}$, the reduction factor satisfies

$$\frac{\zeta_{\text{multi}}}{\zeta_{\text{single}}} \leq \frac{1}{c}\, e^{-\beta_{\mathrm{p}} L}\, \frac{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})(L+1)}}{1-e^{(\beta_{\mathrm{p}}-\beta_{\mathrm{r}})}} = \frac{1}{c}\, e^{-\beta_{\mathrm{r}} L}\, \frac{1-e^{(\beta_{\mathrm{r}}-\beta_{\mathrm{p}})(L+1)}}{1-e^{(\beta_{\mathrm{r}}-\beta_{\mathrm{p}})}}. \tag{A.3}$$

In both cases, the reduction factor can be expressed as

$$\frac{\zeta_{\text{multi}}}{\zeta_{\text{single}}} \leq \frac{1}{c}\, e^{-\min(\beta_{\mathrm{p}},\beta_{\mathrm{r}})L}\, \frac{1-a^{L+1}}{1-a}, \tag{A.4}$$

where $a = e^{-|\beta_{\mathrm{p}}-\beta_{\mathrm{r}}|} \in (0,1)$. Using induction, one can easily show that

$$\frac{1-a^{L+1}}{1-a} \leq \min\Big(L+1, \frac{1}{1-a}\Big), \quad \forall L \geq 0, \forall a \in (0,1), \tag{A.5}$$

which completes the proof for $\beta_{\mathrm{p}} \neq \beta_{\mathrm{r}}$.

For $\beta_{\mathrm{p}} = \beta_{\mathrm{r}} = \min(\beta_{\mathrm{p}}, \beta_{\mathrm{r}})$, the result of Corollary 4.8 follows trivially from (A.1), since each of the terms in the sum on the right hand side is 1 and $L + 1 < \infty = \frac{1}{1-a}$.

Appendix B. Proof of Corollary 5.5.

Due to Corollary 5.2, we have

$$\beta_\ell(\boldsymbol{v}_\ell^*, \boldsymbol{v}_\ell') = \min\left\{1, \frac{\pi_\ell(\boldsymbol{v}_\ell' \,|\, \boldsymbol{y})\, \pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^* \,|\, \boldsymbol{y})}{\pi_\ell(\boldsymbol{v}_\ell^* \,|\, \boldsymbol{y})\, \pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}' \,|\, \boldsymbol{y})}\, \frac{q(\boldsymbol{v}_{\ell,f}^* \,|\, \boldsymbol{v}_\ell', \boldsymbol{v}_{\ell\text{-}1}^*)}{q(\boldsymbol{v}_{\ell,f}' \,|\, \boldsymbol{v}_\ell^*, \boldsymbol{v}_{\ell\text{-}1}')}\right\},$$

where, by definition,

$$\frac{\pi_\ell(\boldsymbol{v}_\ell' \,|\, \boldsymbol{y})\, \pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^* \,|\, \boldsymbol{y})}{\pi_\ell(\boldsymbol{v}_\ell^* \,|\, \boldsymbol{y})\, \pi_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}' \,|\, \boldsymbol{y})} = \frac{p_\ell(\boldsymbol{v}_\ell')\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^*)}{p_\ell(\boldsymbol{v}_\ell^*)\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}')}\, \frac{\exp\left(-\eta_\ell(\boldsymbol{v}_\ell'; \boldsymbol{y}) + \eta_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}'; \boldsymbol{y})\right)}{\exp\left(-\eta_\ell(\boldsymbol{v}_\ell^*; \boldsymbol{y}) + \eta_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^*; \boldsymbol{y})\right)},$$

such that we can write

$$\beta_\ell(\boldsymbol{v}_\ell^*, \boldsymbol{v}_\ell') = \min\Bigg\{1, \underbrace{\frac{p_\ell(\boldsymbol{v}_\ell')\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^*)\, q(\boldsymbol{v}_{\ell,f}^* | \boldsymbol{v}_\ell', \boldsymbol{v}_{\ell\text{-}1}^*)}{p_\ell(\boldsymbol{v}_\ell^*)\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}')\, q(\boldsymbol{v}_{\ell,f}' | \boldsymbol{v}_\ell^*, \boldsymbol{v}_{\ell\text{-}1}')}}_{\textcircled{1}} \underbrace{\frac{\exp\left(-\eta_\ell(\boldsymbol{v}_\ell'; \boldsymbol{y}) + \eta_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}'; \boldsymbol{y})\right)}{\exp\left(-\eta_\ell(\boldsymbol{v}_\ell^*; \boldsymbol{y}) + \eta_{\ell\text{-}1}(\boldsymbol{v}_{\ell\text{-}1}^*; \boldsymbol{y})\right)}}_{\textcircled{2}}\Bigg\}. \tag{B.1}$$

The level ℓ parameter vectors can be split as $\boldsymbol{v}_\ell' = (\boldsymbol{v}_{\ell,f}', \boldsymbol{v}_{\ell,c}')$ and $\boldsymbol{v}_\ell^* = (\boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*)$. and we have $\boldsymbol{v}_{\ell,c}' = \boldsymbol{v}_{\ell\text{-}1}'$ and $\boldsymbol{v}_{\ell,c}^* = \boldsymbol{v}_{\ell\text{-}1}^*$ by construction in the coupling procedure. Thus,

$$\textcircled{1} = \frac{p_\ell(\boldsymbol{v}_{\ell,f}', \boldsymbol{v}_{\ell,c}')\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell,c}^*)\, q(\boldsymbol{v}_{\ell,f}^* | \boldsymbol{v}_{\ell,f}', \boldsymbol{v}_{\ell,c}', \boldsymbol{v}_{\ell,c}^*)}{p_\ell(\boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*)\, p_{\ell\text{-}1}(\boldsymbol{v}_{\ell,c}')\, q(\boldsymbol{v}_{\ell,f}' | \boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*, \boldsymbol{v}_{\ell,c}')}. \tag{B.2}$$

The density of the conditional DILI proposal $q(\boldsymbol{v}_{\ell,f}' | \boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*, \boldsymbol{v}_{\ell,c}')$ is defined as

$$q(\boldsymbol{v}_{\ell,f}' | \boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*, \boldsymbol{v}_{\ell,c}') = \frac{q(\boldsymbol{v}_{\ell,f}', \boldsymbol{v}_{\ell,c}' | \boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*)}{q(\boldsymbol{v}_{\ell,c}' | \boldsymbol{v}_{\ell,f}^*, \boldsymbol{v}_{\ell,c}^*)}, \tag{B.3}$$

that is the ratio between the DILI proposal density and the marginal DILI proposal density, which takes the form

$$q(\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c}) \equiv \int q(\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})d\boldsymbol{v}'_{\ell,f}. \tag{B.4}$$

Due to Corollary 5.3, the DILI proposal $q(\boldsymbol{v}'_\ell|\boldsymbol{v}^*_\ell)$ has the prior distribution $p_\ell(\boldsymbol{v}_\ell)$ as invariant measure, i.e.,

$$p_\ell(\boldsymbol{v}^*_\ell)q(\boldsymbol{v}'_\ell|\boldsymbol{v}^*_\ell) = p_\ell(\boldsymbol{v}'_\ell). \tag{B.5}$$

Hence, if $\boldsymbol{v}^*_\ell = (\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})$ is drawn from the prior $p_\ell(\boldsymbol{v}_\ell)$, then the proposal candidate $\boldsymbol{v}'_\ell = (\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})$ also follows the prior $p_\ell(\boldsymbol{v}_\ell)$. Furthermore, if $\boldsymbol{v}^*_\ell$ is drawn from $p_\ell(\boldsymbol{v}_\ell)$, then the marginal DILI proposal $q(\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})$ generates candidates with coarse components that follow the marginal prior

$$\int p_\ell(\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})d\boldsymbol{v}'_{\ell,f},$$

which for our particular choice of parametrisation is the same as the prior $p_{\ell-1}(\boldsymbol{v}'_{\ell,c})$ on level $\ell-1$, that is, $p_\ell(\boldsymbol{v}^*_\ell)q(\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_\ell) = p_{\ell-1}(\boldsymbol{v}'_{\ell,c})$. Using this identity and substituting (B.3) into (B.2), the ratio $\textcircled{1}$ can be simplified to

$$\begin{aligned}
\textcircled{1} &= \frac{p_\ell(\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})\,q(\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c}|\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})\,q(\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}^*_{\ell,c})}{p_\ell(\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})\,q(\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})\,q(\boldsymbol{v}^*_{\ell,c}|\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}'_{\ell,c})}\\
&= \frac{p_\ell(\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})\,q(\boldsymbol{v}'_{\ell,c}|\boldsymbol{v}^*_{\ell,f},\boldsymbol{v}^*_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}^*_{\ell,c})}{p_\ell(\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})\,q(\boldsymbol{v}^*_{\ell,c}|\boldsymbol{v}'_{\ell,f},\boldsymbol{v}'_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}'_{\ell,c})} = \frac{p_{\ell-1}(\boldsymbol{v}'_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}^*_{\ell,c})}{p_{\ell-1}(\boldsymbol{v}^*_{\ell,c})\,p_{\ell-1}(\boldsymbol{v}'_{\ell,c})} = 1.
\end{aligned}$$

The result then follows immediately from (B.1).

REFERENCES

[1] C. Andrieu and E. Moulines, *On the ergodicity properties of some adaptive MCMC algorithms*, The Annals of Applied Probability, 16 (2006), pp. 1462–1505.

[2] A. Beskos, A. Jasra, K. Law, Y. Marzouk, and Y. Zhou, *Multilevel sequential Monte Carlo with dimension-independent likelihood-informed proposals*, SIAM/ASA Journal on Uncertainty Quantification, 6 (2018), pp. 762–786.

[3] A. Beskos, O. Papaspiliopoulos, G. O. Roberts, and P. Fearnhead, *Exact and computationally efficient likelihood based estimation for discretely observed diffusion processes (with discussion)*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68 (2006), pp. 333–382.

[4] A. Beskos, G. O. Roberts, A. M. Stuart, and J. Voss, *MCMC methods for diffusion bridges*, Stochastic Dynamics, 8 (2008), pp. 319–350.

[5] T. Bui-Thanh, O. Ghattas, J. Martin, and G. Stadler, *A computational framework for infinite-dimensional Bayesian inverse problems. Part I: The linearized case, with application to global seismic inversion*, SIAM Journal on Scientific Computing, 35 (2013), pp. A2494–A2523.

[6] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, *Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate*, PACM Transactions on Mathematical Software, 35 (2008), pp. 22:1–22:14.

[7] K. Cliffe, M. Giles, R. Scheichl, and A. Teckentrup, *Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients*, Computing and Visualization in Science, 14 (2011), pp. 3–15.

[8] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, *MCMC methods for functions: modifying old algorithms to make them faster*, Statistical Science, 28 (2013), pp. 424–446.

[9] T. Cui, C. Fox, and M. J. O'Sullivan, *Bayesian calibration of a large-scale geothermal reservoir model by a new adaptive delayed acceptance Metropolis Hastings algorithm*, Water Resource Research, 47 (2011), p. W10521.

[10] T. Cui, K. J. H. Law, and Y. M. Marzouk, *Dimension-independent likelihood-informed MCMC*, Journal of Computational Physics, 304 (2016), pp. 109–137.

[11] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini, *Likelihood-informed dimension reduction for nonlinear inverse problems*, Inverse Problems, 30 (2014), p. 114015.

[12] T. Cui, Y. M. Marzouk, and K. E. Willcox, *Scalable posterior approximations for large-scale Bayesian*

inverse problems via likelihood-informed parameter and state reduction, Journal of Computational Physics, 315 (2016), pp. 363–387.

[13] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup, *A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow*, SIAM/ASA Journal on Uncertainty Quantification, 3 (2015), pp. 1075–1108.

[14] M. B. Giles, *Multi-level Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.

[15] H. Haario, M. Laine, M. Lehtinen, E. Saksman, and J. Tamminen, *Markov chain Monte Carlo methods for high dimensional inversion in remote sensing*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66 (2004), pp. 591–608.

[16] H. Haario, E. Saksman, and J. Tamminen, *An adaptive Metropolis algorithm*, Bernoulli, 7 (2001), pp. 223–242.

[17] M. Hairer, A. M. Stuart, and S. Vollmer, *Spectral gaps for a Metropolis-Hastings algorithm in infinite dimensions*, Annals of Applied Probability, 24 (2014), pp. 2455–2490.

[18] M. Hairer, A. M. Stuart, and J. Voss, *Signal processing problems on function space: Bayesian formulation, stochastic PDEs and effective MCMC methods*, in The Oxford Handbook of Nonlinear Filtering, D. Crisan and B. Rozovsky, eds., Oxford University Press, 2011.

[19] W. Hastings, *Monte Carlo sampling using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.

[20] D. Higdon, H. Lee, and C. Holloman, *Markov chain Monte Carlo-based approaches for inference in computationally intensive inverse problems*, in Bayesian Statistics 7, J. M. Bernardo, M. J. Bayarri, J. O. Berger, A. P. Dawid, D. Heckerman, A. F. M. Smith, and M. West, eds., Oxford University Press, 2003, pp. 181–197.

[21] V. H. Hoang, C. Schwab, and A. M. Stuart, *Complexity analysis of accelerated MCMC methods for Bayesian inversion*, Inverse Problems, 29 (2013), p. 085010.

[22] M. A. Iglesias, K. J. H. Law, and A. M. Stuart, *Evaluation of Gaussian approximations for data assimilation in reservoir models*, Computational Geosciences, 17 (2013), pp. 851–885, https://doi.org/10.1007/s10596-013-9359-x.

[23] A. Jasra, K. Kamatani, K. J. H. Law, and Y. Zhou, *A multi-index Markov chain Monte Carlo method*, International Journal for Uncertainty Quantification, 8 (2018), pp. 61–73.

[24] J. P. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, vol. 160, Springer, New York, 2004.

[25] K. J. H. Law, *Proposals which speed up function-space MCMC*, Journal of Computational and Applied Mathematics, 262 (2014), pp. 127–138.

[26] R. B. Lehoucq, D. C. Sorenson, and C. Yang, *ARPACK Users' Guide.*, Philadelphia, PA: SIAM, 1998.

[27] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM Journal on Scientific Computing, 34 (2012), pp. A1460–A1487.

[28] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, *Equation of state calculations by fast computing machines*, Journal of Chemical Physics, 21 (1953), pp. 1087–1092.

[29] N. Petra, J. Martin, G. Stadler, and O. Ghattas, *A computational framework for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet flow inverse problems*, SIAM Journal on Scientific Computing, 34 (2014), pp. A1525–A1555.

[30] G. O. Roberts and J. S. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 60 (1998), pp. 255–268.

[31] D. Rudolf and B. Sprungk, *On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm*, Foundations of Computational Mathematics, 18 (2018), pp. 309–343.

[32] Y. Saad, *Numerical methods for large eigenvalue problems: revised edition*, SIAM, 2011.

[33] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. M. Marzouk, *Optimal low-rank approximation of linear Bayesian inverse problems*, SIAM Journal on Scientific Computing, 37 (2015), pp. A2451–A2487.

[34] A. M. Stuart, *Inverse problems: a Bayesian perspective*, Acta Numerica, 19 (2010), pp. 451–559.

[35] A. Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, Society for Industrial Mathematics, Philadelphia, 2005.

[36] A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann, *Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients*, Numerische Mathematik, 125 (2013), pp. 569–600.

[37] L. Tierney, *Markov chains for exploring posterior distributions*, Annals of Statistics, 22 (1994), pp. 1701–1728.

[38] L. Tierney, *A note on Metropolis-Hastings kernels for general state spaces*, Annals of Applied Probability, 8 (1998), pp. 1–9.

[39] O. Zahm, T. Cui, K. Kody Law, A. Spantini, and Y. Youssef Marzouk, *Certified dimension reduction in nonlinear Bayesian inverse problems*, Preprint arXiv:1807.03712, 2018.

■ Paper III

A Stein variational Newton method

Gianluca Detommaso
University of Bath & The Alan Turing Institute
gd391@bath.ac.uk

Tiangang Cui
Monash University
Tiangang.Cui@monash.edu

Alessio Spantini
Massachusetts Institute of Technology
spantini@mit.edu

Youssef Marzouk
Massachusetts Institute of Technology
ymarz@mit.edu

Robert Scheichl
Heidelberg University
r.scheichl@uni-heidelberg.de

Abstract

Stein variational gradient descent (SVGD) was recently proposed as a general purpose nonparametric variational inference algorithm [Liu & Wang, NIPS 2016]: it minimizes the Kullback–Leibler divergence between the target distribution and its approximation by implementing a form of functional gradient descent on a reproducing kernel Hilbert space. In this paper, we accelerate and generalize the SVGD algorithm by including second-order information, thereby approximating a Newton-like iteration in function space. We also show how second-order information can lead to more effective choices of kernel. We observe significant computational gains over the original SVGD algorithm in multiple test cases.

1 Introduction

Approximating an intractable probability distribution via a collection of samples—in order to evaluate arbitrary expectations over the distribution, or to otherwise characterize uncertainty that the distribution encodes—is a core computational challenge in statistics and machine learning. Common features of the target distribution can make sampling a daunting task. For instance, in a typical Bayesian inference problem, the posterior distribution might be strongly non-Gaussian (perhaps multimodal) and high dimensional, and evaluations of its density might be computationally intensive.

There exist a wide range of algorithms for such problems, ranging from parametric variational inference [4] to Markov chain Monte Carlo (MCMC) techniques [10]. Each algorithm offers a different computational trade-off. At one end of the spectrum, we find the parametric mean-field approximation—a cheap but potentially inaccurate variational approximation of the target density. At the other end, we find MCMC—a nonparametric sampling technique yielding estimators that are consistent, but potentially slow to converge. In this paper, we focus on Stein variational gradient descent (SVGD) [17], which lies somewhere in the middle of the spectrum and can be described as a particular *nonparametric* variational inference method [4], with close links to the density estimation approach in [2].

The SVGD algorithm seeks a deterministic coupling between a tractable reference distribution of choice (e.g., a standard normal) and the intractable target. This coupling is induced by a transport map T that can *transform* a collection of reference samples into samples from the desired target distribution. For a given pair of distributions, there may exist infinitely many such maps [28]; several existing algorithms (e.g., [27, 24, 21]) aim to approximate feasible transport maps of various forms.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

The distinguishing feature of the SVGD algorithm lies in its definition of a suitable map T. Its central idea is to approximate T as a growing composition of simple maps, computed *sequentially*:

$$T = T_1 \circ \cdots \circ T_k \circ \cdots , \tag{1}$$

where each map T_k is a perturbation of the identity map along the steepest descent direction of a functional J that describes the Kullback–Leibler (KL) divergence between the pushforward of the reference distribution through the composition $T_1 \circ \cdots \circ T_k$ and the target distribution. The choice of the steepest descent direction is further restricted to a reproducing kernel Hilbert space (RKHS) in order to give T_k a nonparametric closed form [3]. Even though the resulting map T_k is available explicitly without any need for numerical optimization, the SVGD algorithm implicitly approximates a steepest descent iteration on a space of maps of given regularity.

A primary goal of this paper is to explore the use of second-order information (e.g., Hessians) within the SVGD algorithm. Our idea is to develop the analogue of a Newton iteration—rather than gradient descent—for the purpose of sampling distributions more efficiently. Specifically, we design an algorithm where each map T_k is now computed as the perturbation of the identity function along the direction that minimizes a certain local quadratic approximation of J. Accounting for second-order information can dramatically accelerate convergence to the target distribution, at the price of additional work per iteration. The tradeoff between speed of convergence and cost per iteration is resolved in favor of the Newton-like algorithm—which we call a Stein variational Newton method (SVN)—in several numerical examples.

The efficiency of the SVGD and SVN algorithms depends further on the choice of reproducing kernel. A second contribution of this paper is to design geometry-aware Gaussian kernels that also exploit second-order information, yielding substantially faster convergence towards the target distribution than SVGD or SVN with an isotropic kernel.

In the context of *parametric* variational inference, second-order information has been used to accelerate the convergence of certain variational approximations, e.g., [14, 13, 21]. In this paper, however, we focus on *nonparametric* variational approximations, where the corresponding optimisation problem is defined over an infinite-dimensional RKHS of transport maps. More closely related to our work is the Riemannian SVGD algorithm [18], which generalizes a gradient flow interpretation of SVGD [15] to Riemannian manifolds, and thus also exploits geometric information within the inference task.

The rest of the paper is organized as follows. Section 2 briefly reviews the SVGD algorithm, and Section 3 introduces the new SVN method. In Section 4 we introduce geometry-aware kernels for the SVN method. Numerical experiments are described in Section 5. Proofs of our main results and further numerical examples addressing scaling to high dimensions are given in the Appendix. Code and all numerical examples are collected in our GitHub repository [1].

2 Background

Suppose we wish to approximate an intractable target distribution with density π on $\mathbb{R}^d$ via an empirical measure, i.e., a collection of samples. Given samples $\{x_i\}$ from a tractable reference density p on $\mathbb{R}^d$, one can seek a transport map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that the pushforward density of p under T, denoted by $T_* p$, is a close approximation to the target π.[1] There exist infinitely many such maps [28]. The image of the reference samples under the map, $\{T(x_i)\}$, can then serve as an empirical measure approximation of π (e.g., in the weak sense [17]).

Variational approximation. Using the KL divergence to measure the discrepancy between the target π and the pushforward $T_* p$, one can look for a transport map T that minimises the functional

$$T \mapsto \mathcal{D}_{\text{KL}}(T_* \, p \,||\, \pi) \tag{2}$$

over a broad class of functions. The Stein variational method breaks the minimization of (2) into several simple steps: it builds a sequence of transport maps $\{T_1, T_2, \ldots, T_l, \ldots\}$ to iteratively push an initial reference density p_0 towards π. Given a scalar-valued RKHS $\mathcal{H}$ with a positive definite kernel $k(x, x')$, each transport map $T_l : \mathbb{R}^d \to \mathbb{R}^d$ is chosen to be a perturbation of the identity map $I(x) = x$ along the vector-valued RKHS $\mathcal{H}^d \simeq \mathcal{H} \times \cdots \times \mathcal{H}$, i.e.,

$$T_l(x) := I(x) + Q(x) \quad \text{for} \quad Q \in \mathcal{H}^d. \tag{3}$$

[1] If T is invertible, then $T_* p(x) = p(T^{-1}(x)) \, | \det(\nabla_x T^{-1}(x))|$.

The transport maps are computed iteratively. At each iteration l, our best approximation of π is given by the pushforward density $p_l = (T_l \circ \cdots \circ T_1)_* \, p_0$. The SVGD algorithm then seeks a transport map $T_{l+1} = I + Q$ that further decreases the KL divergence between $(T_{l+1})_* p_l$ and π,

$$Q \mapsto J_{p_l}[Q] := \mathcal{D}_{\mathrm{KL}}((I + Q)_* \, p_l \,||\, \pi), \tag{4}$$

for an appropriate choice of $Q \in \mathcal{H}^d$. In other words, the SVGD algorithm seeks a map $Q \in \mathcal{H}^d$ such that

$$J_{p_l}[Q] < J_{p_l}[\mathbf{0}], \tag{5}$$

where $\mathbf{0}(x) = 0$ denotes the zero map. By construction, the sequence of pushforward densities $\{p_0, p_1, p_2, \ldots, p_l, \ldots\}$ becomes increasingly closer (in KL divergence) to the target π. Recent results on the convergence of the SVGD algorithm are presented in [15].

Functional gradient descent. The first variation of J_{p_l} at $S \in \mathcal{H}^d$ along $V \in \mathcal{H}^d$ can be defined as

$$DJ_{p_l}[S](V) := \lim_{\tau \to 0} \frac{1}{\tau} \big(J_{p_l}[S + \tau V] - J_{p_l}[S] \big). \tag{6}$$

Assuming that the objective function $J_{p_l} : \mathcal{H}^d \to \mathbb{R}$ is Fréchet differentiable, the *functional gradient* of J_{p_l} at $S \in \mathcal{H}^d$ is the element $\nabla J_{p_l}[S]$ of $\mathcal{H}^d$ such that

$$DJ_{p_l}[S](V) = \langle \nabla J_{p_l}[S], V \rangle_{\mathcal{H}^d} \quad \forall \, V \in \mathcal{H}^d, \tag{7}$$

where $\langle \cdot, \cdot \rangle_{\mathcal{H}^d}$ denotes an inner product on $\mathcal{H}^d$.

In order to satisfy (5), the SVGD algorithm defines T_{l+1} as a perturbation of the identity map along the steepest descent direction of the functional J_{p_l} evaluated at the zero map, i.e.,

$$T_{l+1} = I - \varepsilon \nabla J_{p_l}[\mathbf{0}], \tag{8}$$

for a small enough $\varepsilon > 0$. It was shown in [17] that the functional gradient at $\mathbf{0}$ has a closed form expression given by

$$-\nabla J_{p_l}[\mathbf{0}](z) = \mathbb{E}_{x \sim p_l}[k(x, z) \nabla_x \log \pi(x) + \nabla_x k(x, z)]. \tag{9}$$

Empirical approximation. There are several ways to approximate the expectation in (9). For instance, a set of particles $\{x_i^0\}_{i=1}^n$ can be generated from the initial reference density p_0 and pushed forward by the transport maps $\{T_1, T_2, \ldots\}$. The pushforward density p_l can then be approximated by the empirical measure given by the particles $\{x_i^l\}_{i=1}^n$, where $x_i^l = T_l(x_i^{l-1})$ for $i = 1, \ldots, n$, so that

$$-\nabla J_{p_l}[\mathbf{0}](z) \approx G(z) := \frac{1}{n} \textstyle\sum_{j=1}^n \left[k(x_j^l, z) \nabla_{x_j^l} \log \pi(x_j^l) + \nabla_{x_j^l} k(x_j^l, z) \right]. \tag{10}$$

The first term in (10) corresponds to a weighted average steepest descent direction of the log-target density π with respect to p_l. This term is responsible for transporting particles towards high-probability regions of π. In contrast, the second term can be viewed as a "repulsion force" that spreads the particles along the support of π, preventing them from collapsing around the mode of π. The SVGD algorithm is summarised in Algorithm 1.

Algorithm 1: One iteration of the Stein variational gradient algorithm

```
Input  : Particles $\{x_i^l\}_{i=1}^n$ at previous iteration $l$; step size $\varepsilon_{l+1}$
Output : Particles $\{x_i^{l+1}\}_{i=1}^n$ at new iteration $l+1$
for $i = 1, 2, \ldots, n$ do
    Set $x_i^{l+1} \leftarrow x_i^l + \varepsilon_{l+1} \, G(x_i^l)$, where $G$ is defined in (10).
end for
```

3 Stein variational Newton method

Here we propose a new method that incorporates second-order information to accelerate the convergence of the SVGD algorithm. We replace the steepest descent direction in (8) with an approximation of the Newton direction.

Functional Newton direction. Given a differentiable objective function J_{p_l}, we can define the second variation of J_{p_l} at $\mathbf{0}$ along the pair of directions $V, W \in \mathcal{H}^d$ as

$$D^2 J_{p_l}[\mathbf{0}](V, W) := \lim_{\tau \to 0} \frac{1}{\tau} \big(DJ_{p_l}[\tau W](V) - DJ_{p_l}[\mathbf{0}](V) \big).$$

At each iteration, the Newton method seeks to minimize a local quadratic approximation of J_{p_l}. The minimizer $W \in \mathcal{H}^d$ of this quadratic form defines the Newton direction and is characterized by the first order stationarity conditions

$$D^2 J_{p_l}[\mathbf{0}](V, W) = -DJ_{p_l}[\mathbf{0}](V), \quad \forall\, V \in \mathcal{H}^d. \tag{11}$$

We can then look for a transport map T_{l+1} that is a local perturbation of the identity map along the Newton direction, i.e.,

$$T_{l+1} = I + \varepsilon W, \tag{12}$$

for some $\varepsilon > 0$ that satisfies (5). The function W is guaranteed to be a descent direction if the bilinear form $D^2 J_{p_l}[\mathbf{0}]$ in (11) is positive definite. The following theorem gives an explicit form for $D^2 J_{p_l}[\mathbf{0}]$ and is proven in Appendix.

Theorem 1. *The variational characterization of the Newton direction* $W = (w_1, \dots, w_d)^\top \in \mathcal{H}^d$ *in* (11) *is equivalent to*

$$\sum_{i=1}^d \left\langle \sum_{j=1}^d \langle h_{ij}(y, z), w_j(z) \rangle_{\mathcal{H}} + \partial_i J_{p_l}[\mathbf{0}](y), v_i(y) \right\rangle_{\mathcal{H}} = 0, \tag{13}$$

for all $V = (v_1, \dots, v_d)^\top \in \mathcal{H}^d$, *where*

$$h_{ij}(y, z) = \mathbb{E}_{x \sim p_l} \left[-\partial^2_{ij} \log \pi(x) k(x, y) k(x, z) + \partial_i k(x, y) \partial_j k(x, z) \right]. \tag{14}$$

We propose a Galerkin approximation of (13). Let $(x_k)_{k=1}^n$ be an ensemble of particles distributed according to $p_l(\,\cdot\,)$, and define the finite dimensional linear space $\mathcal{H}^d_n = \operatorname{span}\{k(x_1, \cdot), \dots, k(x_n, \cdot)\}$. We look for an approximate solution $W = (w_1, \dots, w_d)^\top$ in $\mathcal{H}^d_n$—i.e.,

$$w_j(z) = \sum_{k=1}^n \alpha_j^k \, k(x_k, z) \tag{15}$$

for some unknown coefficients (α_j^k)—such that the residual of (13) is orthogonal to $\mathcal{H}^d_n$. The following corollary gives an explicit characterization of the Galerkin solution and is proven in the Appendix.

Corollary 1. *The coefficients* (α_j^k) *are given by the solution of the linear system*

$$\sum_{k=1}^n H^{s,k} \, \alpha^k = \nabla J^s, \quad \textit{for all} \quad s = 1, \dots, n, \tag{16}$$

where $\alpha^k := \left(\alpha_1^k, \dots, \alpha_d^k \right)^\top$ *is a vector of unknown coefficients,* $(H^{s,k})_{ij} := h_{ij}(x_s, x_k)$ *is the evaluation of the symmetric form* (14) *at pairs of particles, and where* $\nabla J^s := -\nabla J_{p_l}[\mathbf{0}](x_s)$ *represents the evaluation of the first variation at the* s*-th particle.*

In practice, we can only evaluate a Monte Carlo approximation of $H^{s,k}$ and ∇J^s in (16) using the ensemble $(x_k)_{k=1}^n$.

Inexact Newton. The solution of (16) by means of direct solvers might be impractical for problems with a large number of particles n or high parameter dimension d, since it is a linear system with nd unknowns. Moreover, the solution of (16) might not lead to a descent direction (e.g., when π is not log-concave). We address these issues by deploying two well-established techniques in nonlinear optimisation [31]. In the first approach, we solve (16) using the inexact Newton–conjugate gradient (NCG) method [31, Chapters 5 and 7], wherein a descent direction can be guaranteed by appropriately terminating the conjugate gradient iteration. The NCG method only needs to evaluate the matrix-vector product with each $H^{s,k}$ and does not construct the matrix explicitly, and thus can be scaled to high dimensions. In the second approach, we simplify the problem further by taking a block-diagonal approximation of the second variation, breaking (16) into n decoupled $d \times d$ linear systems

$$H^{s,s}\alpha^s = \nabla J^s, \qquad s = 1, \dots n\,. \tag{17}$$

Here, we can either employ a Gauss-Newton approximation of the Hessian $\nabla^2 \log \pi$ in $H^{s,s}$ or again use inexact Newton–CG, to guarantee that the approximation of the Newton direction is a descent direction.

Both the block-diagonal approximation and inexact NCG are more efficient than solving for the full Newton direction (16). In addition, the block-diagonal form (17) can be solved in parallel for each of the blocks, and hence it may best suit high-dimensional applications and/or large numbers of particles. In the Appendix, we provide a comparison of these approaches on various examples. Both approaches provide similar progress per SVN iteration compared to the full Newton direction.

Leveraging second-order information provides a natural scaling for the step size, i.e., $\varepsilon = O(1)$. Here, the choice $\varepsilon = 1$ performs reasonably well in our numerical experiments (Section 5 and the Appendix). In future work, we will refine our strategy by considering either a line search or a trust region step. The resulting Stein variational Newton method is summarised in Algorithm 2.

Algorithm 2: One iteration of the Stein variational Newton algorithm

```
Input  : Particles $\{x_i^l\}_{i=1}^n$ at stage $l$; step size $\varepsilon$
Output : Particles $\{x_i^{l+1}\}_{i=1}^n$ at stage $l+1$
for $i = 1, 2, \dots, n$ do
    Solve the linear system (16) for $\alpha^1, \dots, \alpha^n$
    Set $x_i^{l+1} \leftarrow x_i^l + \varepsilon W(x_i^l)$ given $\alpha^1, \dots, \alpha^n$
end for
```

4 Scaled Hessian kernel

In the Stein variational method, the kernel weighs the contribution of each particle to a locally *averaged* steepest descent direction of the target distribution, and it also spreads the particles along the support of the target distribution. Thus it is essential to choose a kernel that can capture the underlying geometry of the target distribution, so the particles can traverse the support of the target distribution efficiently. To this end, we can use the curvature information characterised by the Hessian of the logarithm of the target density to design anisotropic kernels.

Consider a positive definite matrix $A(x)$ that approximates the local Hessian of the negative logarithm of the target density, i.e., $A(x) \approx -\nabla_x^2 \log \pi(x)$. We introduce the metric

$$M_\pi := \mathbb{E}_{x\sim\pi}[A(x)]\,, \tag{18}$$

to characterise the average curvature of the target density, stretching and compressing the parameter space in different directions. There are a number of computationally efficient ways to evaluate such an $A(x)$—for example, the generalised eigenvalue approach in [20] and the Fisher information-based approach in [11]. The expectation in (18) is taken against the target density π, and thus cannot be directly computed. Utilising the ensemble $\{x_i^l\}_{i=1}^n$ in each iteration, we introduce an alternative metric

$$M_{p_l} := \frac{1}{n}\sum_{i=1}^n A(x_i^l), \tag{19}$$

to approximate M_π. Similar approximations have also been introduced in the context of dimension reduction for statistical inverse problems; see [7]. Note that the computation of the metric (19) does not incur extra computational cost, as we already calculated (approximations to) $\nabla_x^2 \log \pi(x)$ at each particle in the Newton update.

Given a kernel of the generic form $k(x, x') = f(\|x - x'\|^2)$, we can then use the metric M_{p_l} to define an anisotropic kernel

$$k_l(x, x') = f\left(\frac{1}{g(d)}\|x - x'\|^2_{M_{p_l}}\right),$$

where the norm $\|\cdot\|_{M_{p_l}}$ is defined as $\|x\|^2_{M_{p_l}} = x^\top M_{p_l} x$ and $g(d)$ is a positive and real-valued function of the dimension d. For example, with $g(d) = d$, the Gaussian kernel used in the SVGD of

[17] can be modified as

$$k_l(x, x') := \exp\left(-\frac{1}{2d} \|x - x'\|^2_{M_{p_l}} \right). \tag{20}$$

The metric M_{p_l} induces a deformed geometry in the parameter space: distance is greater along directions where the (average) curvature is large. This geometry directly affects how particles in SVGD or SVN flow—by shaping the locally-averaged gradients and the "repulsion force" among the particles—and tends to spread them more effectively over the high-probability regions of π.

The dimension-dependent scaling factor $g(d)$ plays an important role in high dimensional problems. Consider a sequence of target densities that converges to a limit as the dimension of the parameter space increases. For example, in the context of Bayesian inference on function spaces, e.g., [26], the posterior density is often defined on a discretisation of a function space, whose dimensionality increases as the discretisation is refined. In this case, the $g(d)$-weighed norm $\|\cdot\|^2/d$ is the square of the discretised L^2 norm under certain technical conditions (e.g., the examples in Section 5.2 and the Appendix) and converges to the functional L^2 norm as $d \to \infty$. With an appropriate scaling $g(d)$, the kernel may thus exhibit robust behaviour with respect to discretisation if the target distribution has appropriate infinite-dimensional limits. For high-dimensional target distributions that do not have a well-defined limit with increasing dimension, an appropriately chosen scaling function $g(d)$ can still improve the ability of the kernel to discriminate inter-particle distances. Further numerical investigation of this effect is presented in the Appendix.

5 Test cases

We evaluate our new SVN method with the scaled Hessian kernel on a set of test cases drawn from various Bayesian inference tasks. For these test cases, the target density π is the (unnormalised) posterior density. We assume the prior distributions are Gaussian, that is, $\pi_0(x) = \mathcal{N}(m_{\text{pr}}, C_{\text{pr}})$, where $m_{\text{pr}} \in \mathbb{R}^d$ and $C_{\text{pr}} \in \mathbb{R}^{d\times d}$ are the prior mean and prior covariance, respectively. Also, we assume there exists a forward operator $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^m$ mapping from the parameter space to the data space. The relationship between the observed data and unknown parameters can be expressed as $y = \mathcal{F}(x) + \xi$, where $\xi \sim \mathcal{N}(0, \sigma^2 I)$ is the measurement error and I is the identity matrix. This relationship defines the likelihood function $\mathcal{L}(y|x) = \mathcal{N}(\mathcal{F}(x), \sigma^2 I)$ and the (unnormalised) posterior density $\pi(x) \propto \pi_0(x)\mathcal{L}(y|x)$.

We will compare the performance of SVN and SVGD, both with the scaled Hessian kernel (20) and the heuristically-scaled isotropic kernel used in [17]. We refer to these algorithms as SVN-H, SVN-I, SVGD-H, and SVGD-I, where 'H' or 'I' designate the Hessian or isotropic kernel, respectively. Recall that the heuristic used in the '-I' algorithms involves a scaling factor based on the number of particles n and the median pairwise distance between particles [17]. Here we present two test cases, one multi-modal and the other high-dimensional. In the Appendix, we report on additional tests. First, we evaluate the performance of SVN-H with different Hessian approximations: the exact Hessian (full Newton), the block diagonal Hessian, and a Newton–CG version of the algorithm with exact Hessian. Second, we provide a performance comparison between SVGD and SVN on a high-dimensional Bayesian neural network. Finally, we provide further numerical investigations of the dimension-scalability of our scaled kernel.

5.1 Two-dimensional double banana

The first test case is a two-dimensional bimodal and "banana" shaped posterior density. The prior is a standard multivariate Gaussian, i.e., $m_{\text{pr}} = 0$ and $C_{\text{pr}} = I$, and the observational error has standard deviation $\sigma = 0.3$. The forward operator is taken to be a scalar logarithmic Rosenbrock function, i.e.,

$$\mathcal{F}(x) = \log\left((1 - x_1)^2 + 100(x_2 - x_1^2)^2\right),$$

where $x = (x_1, x_2)$. We take a single observation $y = \mathcal{F}(x_{\text{true}}) + \xi$, with x_{true} being a random variable drawn from the prior and $\xi \sim \mathcal{N}(0, \sigma^2 I)$.

Figure 1 summarises the outputs of four algorithms at selected iteration numbers, each with $n = 1000$ particles initially sampled from the prior π_0. The rows of Figure 1 correspond to the choice of algorithms and the columns of Figure 1 correspond to the outputs at different iteration numbers. We run 10, 50, and 100 iterations of SVN-H. To make a fair comparison, we rescale the number

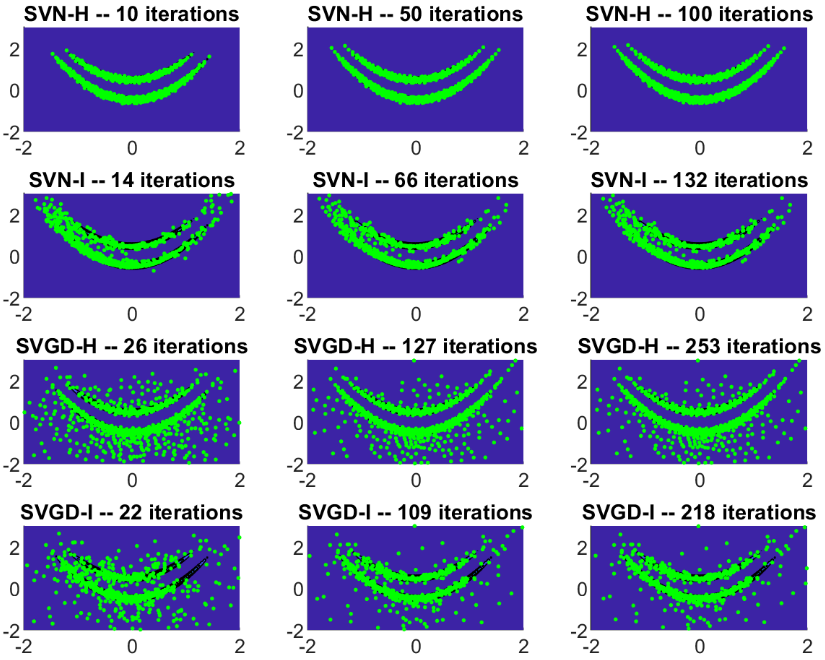


Figure 1: Particle configurations superimposed on contour plots of the double-banana density.

of iterations for each of the other algorithms so that the total cost (CPU time) is approximately the same. It is interesting to note that the Hessian kernel takes considerably less computational time than the Isotropic kernel. This is because, whereas the Hessian kernel is automatically scaled, the Isotropic kernel calculates the distance between the particles at each iterations to heuristically rescale the kernel.

The first row of Figure 1 displays the performance of SVN-H, where second-order information is exploited both in the optimisation and in the kernel. After only 10 iterations, the algorithm has already converged, and the configuration of particles does not visibly change afterwards. Here, all the particles quickly reach the high probability regions of the posterior distribution, due to the Newton acceleration in the optimisation. Additionally, the scaled Hessian kernel seems to spread the particles into a structured and precise configuration.

The second row shows the performance of SVN-I, where the second-order information is used exclusively in the optimisation. We can see the particles quickly moving towards the high-probability regions, but the configuration is much less structured. After 47 iterations, the algorithm has essentially converged, but the configuration of the particles is noticeably rougher than that of SVN-H.

SVGD-H in the third row exploits second-order information exclusively in the kernel. Compared to SVN-I, the particles spread more quickly over the support of the posterior, but not all the particles reach the high probability regions, due to slower convergence of the optimisation. The fourth row shows the original algorithm, SVGD-I. The algorithm lacks both of the benefits of second-order information: with slower convergence and a more haphazard particle distribution, it appears less efficient for reconstructing the posterior distribution.

5.2 100-dimensional conditioned diffusion

The second test case is a high-dimensional model arising from a Langevin SDE, with state $u : [0, T] \to \mathbb{R}$ and dynamics given by

$$du_t = \frac{\beta u\,(1-u^2)}{(1+u^2)}\,dt + dx_t, \quad u_0 = 0\,. \tag{21}$$

Here $x = (x_t)_{t \geq 0}$ is a Brownian motion, so that $x \sim \pi_0 = \mathcal{N}(0, C)$, where $C(t, t') = \min(t, t')$. This system represents the motion of a particle with negligible mass trapped in an energy potential, with thermal fluctuations represented by the Brownian forcing; it is often used as a test case for MCMC algorithms in high dimensions [6]. Here we use $\beta = 10$ and $T = 1$. Our goal is to infer the driving process x and hence its pushforward to the state u.

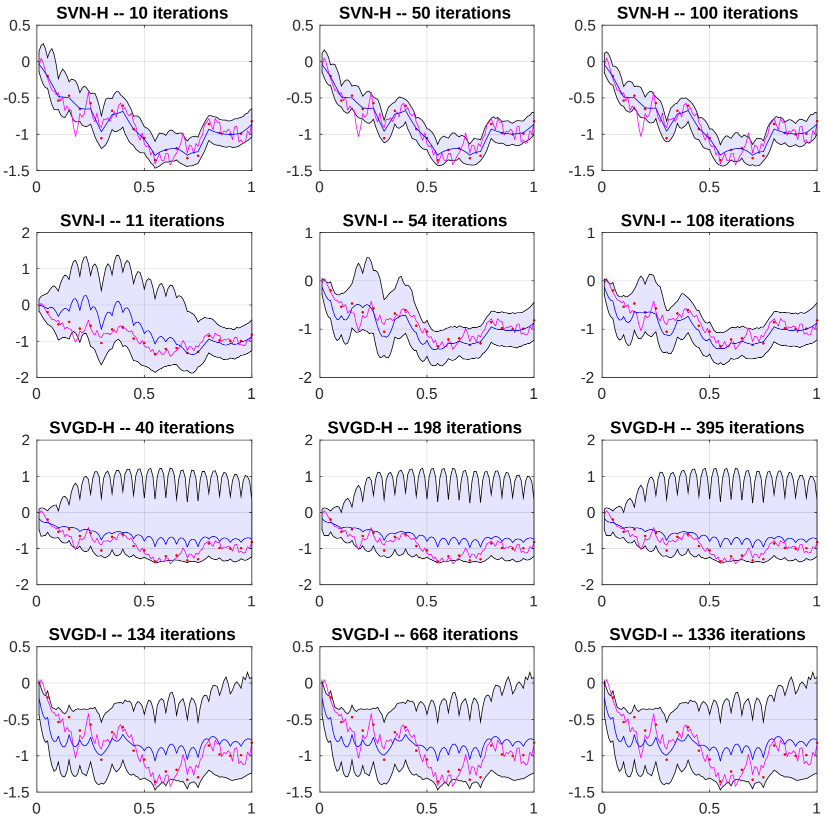


Figure 2: In each plot, the magenta path is the true solution of the discretised Langevin SDE; the blue line is the reconstructed posterior mean; the shaded area is the 90% marginal posterior credible interval at each time step.

The forward operator is defined by $\mathcal{F}(x) = [u_{t_1}, u_{t_2}, \ldots, u_{t_{20}}]^\top \in \mathbb{R}^{20}$, where t_i are equispaced observation times in the interval $(0, 1]$, i.e., $t_i = 0.05\,i$. By taking $\sigma = 0.1$, we define an observation $y = \mathcal{F}(x_{\text{true}}) + \xi \in \mathbb{R}^{20}$, where x_{true} is a Brownian motion path and $\xi \sim \mathcal{N}(0, \sigma^2\, I)$. For discretization, we use an Euler-Maruyama scheme with step size $\Delta t = 10^{-2}$; therefore the dimensionality of the problem is $d = 100$. The prior is given by the Brownian motion $x = (x_t)_{t \geq 0}$, described above.

Figure 2 summarises the outputs of four algorithms, each with $n = 1000$ particles initially sampled from π_0. Figure 2 is presented in the same way as Figure 1 from the first test case. The iteration numbers are scaled, so that we can compare outputs generated by various algorithms using approximately the same amount of CPU time. In Figure 2, the path in magenta corresponds to the solution of the Langevin SDE in (21) driven by the true Brownian path x_{true}. The red points correspond to the 20 noisy observations. The blue path is the reconstruction of the magenta path, i.e., it corresponds to the solution of the Langevin SDE driven by the *posterior mean* of $(x_t)_{t \geq 0}$. Finally, the shaded area represents the marginal 90% credible interval of each dimension (i.e., at each time step) of the posterior distribution of u.

We observe excellent performance of SVN-H. After 50 iterations, the algorithm has already converged, accurately reconstructing the posterior mean (which in turn captures the trends of the true path) and the posterior credible intervals. (See Figure 3 and below for a validation of these results against a reference MCMC simulation.) SVN-I manages to provide a reasonable reconstruction of the target distribution: the posterior mean shows fair agreement with the true solution, but the credible intervals are slightly overestimated, compared to SVN-H and the reference MCMC. The overestimated credible interval may be due to the poor dimension scaling of the isotropic kernel used by SVN-I. With the same amount of computational effort, SVGD-H and SVGD-I cannot reconstruct the posterior distribution: both the posterior mean and the posterior credible intervals depart significantly from their true values.

In Figure 3, we compare the posterior distribution approximated with SVN-H (using $n = 1000$ particles and 100 iterations) to that obtained with a reference MCMC run (using the DILI algorithm of [6] with an effective sample size of 10^5), showing an overall good agreement. The thick blue and green paths correspond to the posterior means estimated by SVN-H and MCMC, respectively. The blue and green shaded areas represent the marginal 90% credible intervals (at each time step) produced by SVN-H and MCMC. In this example, the posterior mean of SVN-H matches that of MCMC quite closely, and both are comparable to the data-generating path (thick magenta line). (The posterior means are much smoother than the true path, which is to be expected.) The estimated credible intervals of SVN-H and MCMC also match fairly well along the entire path of the SDE.

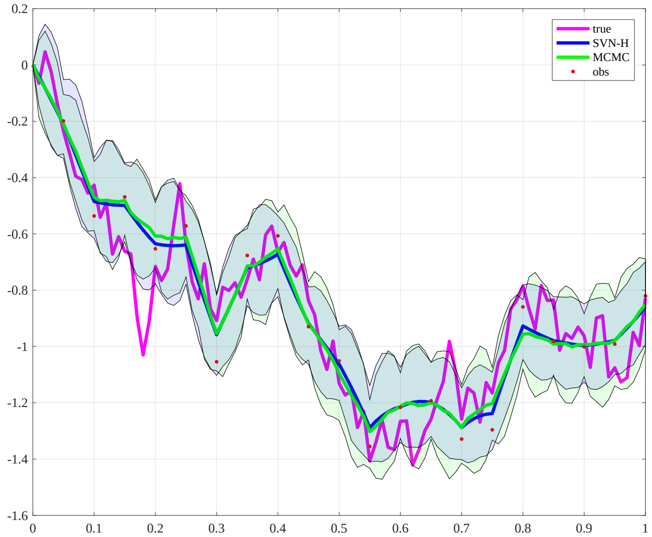


Figure 3: Comparison of reconstructed distributions from SVN-H and MCMC

6 Discussion

In general, the use of Gaussian reproducing kernels may be problematic in high dimensions, due to the locality of the kernel [8]. While we observe in Section 4 that using a properly rescaled Gaussian kernel can improve the performance of the SVN method in high dimensions, we also believe that a truly general purpose nonparametric algorithm using local kernels will inevitably face further challenges in high-dimensional settings. A sensible approach to coping with high dimensionality is also to design algorithms that can detect and exploit essential *structure* in the target distribution, whether it be decaying correlation, conditional independence, low rank, multiple scales, and so on. See [25, 29] for recent efforts in this direction.

7 Acknowledgements

G. Detommaso is supported by the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (EP/L015684/1) and by a scholarship from the Alan Turing Institute. T. Cui, G. Detommaso, A. Spantini, and Y. Marzouk acknowledge support from the MATRIX Program on “Computational Inverse Problems” held at the MATRIX Institute, Australia, where this joint collaboration was initiated. A. Spantini and Y. Marzouk also acknowledge support from the AFOSR Computational Mathematics Program.

A Proof of Theorem 1

The following proposition is used to prove Theorem 1.

Proposition 1. *Define the directional derivative of J_p as the first variation of J_p at $S \in \mathcal{H}^d$ along a direction $V \in \mathcal{H}^d$,*

$$DJ_p[S](V) := \lim_{\tau \to 0} \frac{1}{\tau}\big(J_p[S + \tau V] - J_p[S]\big) .$$

The first variation takes the form

$$DJ_p[S](V) = -\mathbb{E}_{x \sim p}\left[\big(\nabla_x \log \pi(x + S(x))\big)^\top V(x) + \mathrm{trace}\big((I + \nabla_x S(x))^{-1} \nabla_x V(x)\big)\right] . \tag{22}$$

Proof. Given the identity map I and a transport map in the form of $T = I + S + \tau V$, the pullback density of π is defined as

$$T^* \pi = \pi(T(x))\,|\det \nabla_x T(x)| = \pi\big(x + S(x) + \tau V(x)\big)\,\big|\det\big(I + \nabla_x S(x) + \tau \nabla_x T(x)\big)\big| .$$

The perturbed objective function $J_p[S + \tau V]$ takes the form

$$\begin{aligned} J_p[S + \tau V] &= \mathcal{D}_{\mathrm{KL}}((I + S + \tau V)_* p \,\|\, \pi) \\ &= \mathcal{D}_{\mathrm{KL}}(p \,\|\, (I + S + \tau V)^* \pi) \\ &= \int p(x) \log p(x) dx - \int p(x) \Big(\log \pi\big(x + S(x) + \tau V(x)\big) \\ &\qquad + \log \big|\det\big(I + \nabla_x S(x) + \tau \nabla_x V(x)\big)\big| \Big)\, dx . \end{aligned}$$

Thus we have

$$\begin{aligned} J_p[S + \tau V] - J_p[S] = &- \int p(x) \Big(\underbrace{\log \pi\big(x + S(x) + \tau V(x)\big) - \log \pi(x + S(x))}_{(i)} \Big)\, dx \\ &- \int p(x) \big(\underbrace{\log \big|\det\big(I + \nabla_x S(x) + \tau \nabla_x V(x)\big)\big| - \log\big|\det\big(I + \nabla_x S(x)\big)\big|}_{(ii)} \big)\, dx . \end{aligned} \tag{23}$$

Performing a Taylor expansion of the terms (i) and (ii) in (23), we have

$$\begin{aligned} (i) &= \tau\big(\nabla_x \log \pi(x + S(x))\big)^\top V(x) + O(\tau^2) , \\ (ii) &= \tau\, \mathrm{trace}\big((I + \nabla_x S(x))^{-1} \nabla_x V(x)\big) + O(\tau^2) , \end{aligned}$$

where $\nabla_x \log \pi(x + S(x))$ is the partial derivative of $\log \pi$ evaluated at $x + S(x)$. Plugging the above expression into (23) and the definition of the directional derivative, we obtain

$$DJ_p[S](V) = -\mathbb{E}_{x \sim p}\left[\big(\nabla_x \log \pi(x + S(x))\big)^\top V(x) + \mathrm{trace}\big(\nabla_x (x + \nabla_x S(x))^{-1} \nabla_x V(x)\big)\right] . \tag{24}$$

□

The Fréchet derivative of J_p evaluated at $S \in \mathcal{H}^d$, $\nabla J_p[S] : \mathcal{H}^d \to \mathcal{L}(\mathcal{H}^d, \mathbb{R})$ satisfies

$$DJ_p[S](V) = \langle \nabla J_p[S] \,, V \rangle_{\mathcal{H}^d}, \quad \forall\, V \in \mathcal{H}^d ,$$

and thus we can use Proposition 1 to prove Theorem 1.

Proof of Theorem 1. The second variation of J_p at $\mathbf{0}$ along directions $V, W \in \mathcal{H}^d$ takes the form

$$D^2 J_p[\mathbf{0}](V, W) := \lim_{\tau \to 0} \frac{1}{\tau}\big(DJ_p[\tau W](V) - DJ_p[\mathbf{0}](V)\big) .$$

Following Proposition 3, we have

$$\begin{aligned} D^2 J_p[\mathbf{0}](V, W) &= \lim_{\tau \to 0} \frac{1}{\tau}\big(DJ_p[\tau W](V) - DJ_p[\mathbf{0}](V)\big) \\ &= -\,\mathbb{E}_{x\sim p}\big[\underbrace{\lim_{\tau\to 0}\frac{1}{\tau}\big(\nabla_x \log \pi(x+\tau W(x)) - \nabla_x \log \pi(x)\big)}_{(i)}{}^{\top} V(x)\big] \\ &\quad - \mathbb{E}_{x\sim p}\big[\text{trace}\big(\underbrace{\lim_{\tau\to 0}\frac{1}{\tau}[(I+\tau\nabla_x W(x))^{-1} - I]}_{(ii)}\,\nabla_x V(x)\big)\big]\,. \end{aligned} \tag{25}$$

By Taylor expansion, the limits (i) and (ii) of the above equation can be written as

$$\begin{aligned} (i) &= \nabla_x^2 \log \pi(x) W(x)\,, \\ (ii) &= -\nabla_x W(x)\,. \end{aligned}$$

Thus, the second variation of J_p at $\mathbf{0}$ along directions $V, W \in \mathcal{H}^d$ becomes

$$D^2 J_p[\mathbf{0}](V, W) = -\mathbb{E}_{x\sim p}\big[W(x)^\top \nabla_x^2 \log \pi(x) V(x) - \text{trace}\big(\nabla_x W(x) \nabla_x V(x)\big)\big]\,. \tag{26}$$

Using the reproducing property of $V, W \in \mathcal{H}^d$, i.e.

$$\begin{aligned} v_i(x) &= \langle k(x,\cdot), v_i(\cdot)\rangle_{\mathcal{H}}\,, & w_j(x) &= \langle k(x,\cdot), w_j(\cdot)\rangle_{\mathcal{H}} \\ \nabla_x v_i(x) &= \langle \nabla_x k(x,\cdot), v_i(\cdot)\rangle_{\mathcal{H}^d}\,, & \nabla_x w_i(x) &= \langle \nabla_x k(x,\cdot), w_i(\cdot)\rangle_{\mathcal{H}^d} \end{aligned}$$

we then have

$$\mathbb{E}_{x\sim p}\big[W(x)^\top \nabla_x^2 \log \pi(x) V(x)\big] = \sum_{i=1}^d \sum_{j=1}^d \Big\langle \langle \mathbb{E}_{x\sim p}\big[\partial_{ij}^2 \log \pi(x) k(x,y) k(x,z)\big], w_j(z)\rangle_{\mathcal{H}}, v_i(y)\Big\rangle_{\mathcal{H}}$$

and

$$\mathbb{E}_{x\sim p}\big[\text{trace}\big(\nabla_x W(x) \nabla_x V(x)\big)\big] = \sum_{i=1}^d \sum_{j=1}^d \Big\langle \langle \mathbb{E}_{x\sim p}\big[\partial_i k(x,y) \partial_j k(x,z)\big], w_j(z)\rangle_{\mathcal{H}}, v_i(y)\Big\rangle_{\mathcal{H}}\,.$$

Plugging the above identities into (26), the second variation can be expressed as

$$D^2 J_p[\mathbf{0}](V, W) = \sum_{i=1}^d \sum_{j=1}^d \Big\langle \langle h_{ij}(y,z), w_j(z)\rangle_{\mathcal{H}}, v_i(y)\Big\rangle_{\mathcal{H}}\,,$$

where

$$h_{ij}(y,z) := \mathbb{E}_{x\sim p}\big[-\partial_{ij}^2 \log \pi(x) k(x,y) k(x,z) + \partial_i k(x,y) \partial_j k(x,z)\big]\,.$$

Hence the result. □

B Proof of Corollary 1

Proof. Here we drop the subscript p_l. The ensemble of particles $(x_k)_{k=1}^n$ defines a linear function space $\mathcal{H}_n = \text{span}\{k(x_1,\cdot), \dots, k(x_n,\cdot)\}$. In the Galerkin approach, we seek a solution $W = (w_1, \dots, w_d)^\top \in \mathcal{H}_n^d$ such that the residual of the Newton direction

$$\sum_{i=1}^d \left\langle \sum_{j=1}^d \langle h_{ij}(y,z), w_j(z)\rangle_{\mathcal{H}} + \partial_i J[\mathbf{0}](y), v_i(y) \right\rangle_{\mathcal{H}} = 0, \tag{27}$$

is zero for all possible $V \in \mathcal{H}_n^d$. This way, we can approximate each component w_j of the function W as

$$w_j(z) = \sum_{k=1}^n \alpha_j^k \, k(x_k, z), \tag{28}$$

for a collection of unknown coefficients (α_j^k). We define $V^s = (v_1^s, \dots, v_d^s)^\top$ to be the test function where $v_i^s(y) = k(x_s, y)$ for all $s = 1, \dots, n$.

We first project the Newton direction (27) onto V^s for all $s = 1, \dots, n$. Applying the reproducing property of the kernel, this leads to

$$\sum_{j=1}^{d} \langle h_{ij}(x_s, z), w_j(z)\rangle_{\mathcal{H}^d} + \partial_i J_{p_l}[\mathbf{0}](x_s) = 0, \qquad i = 1, \dots, d, \quad s = 1, \dots, n. \tag{29}$$

Plugging (28) into (29), we obtain the fully discrete set of equations

$$\sum_{j=1}^{d} \sum_{\ell=1}^{n} h_{ij}(x_s, x_k)\, \alpha_j^k + \partial_i J_{p_l}[\mathbf{0}](x_s) = 0, \quad i = 1, \dots, d,\ s = 1, \dots, n,\ k = 1, \dots, n. \tag{30}$$

We denote the coefficient vector $\alpha^k := \left(\alpha_1^k, \dots, \alpha_d^k\right)^\top$ for each x_k, the block Hessian matrix $(H^{s,k})_{ij} := h_{ij}(x_s, x_k)$ for each pair of x_s and x_k, and $\nabla J^s := \nabla J[\mathbf{0}](x_s)$ for each x_s. Then equation (30) can be expressed as

$$\sum_{k=1}^{n} H^{s,k}\, \alpha^k = \nabla J^s, \qquad s = 1, \dots, n. \tag{31}$$

□

C Additional test cases

C.1 Comparison between the full and inexact Newton methods

Here we compare three different Stein variational Newton methods: `SVNfull` denotes the method that solves the fully coupled Newton system in equation (16) of the main paper, with no approximations; `SVNCG` denotes the method that applies inexact Newton–CG to the fully coupled system (16); and `SVNbd` employs the block-diagonal approximation given in equation (17) of the main paper.

We first make comparisons using the two-dimensional double banana distribution presented in Section 5.1. We run our test case for $N = 100$ particles and 20 iterations. Figure 4 shows the contours of the target density and the samples produced by each of the three algorithms. Compared to the full Newton method, both the block-diagonal approximation and the inexact Newton–CG generate results of similar quality.

We use an additional nonlinear regression test case for further comparisons. In this case, the forward operator is given by

$$\mathcal{F}(x) = c_1 x_1^3 + c_2 x_2\,,$$

where $x = [x_1, x_2]^\top$ and c_1, c_2 are some fixed coefficients sampled independently from a standard normal distribution. A data point is then given by $y = \mathcal{F}(x) + \varepsilon$, where $\varepsilon \sim N(0, \sigma^2)$ and $\sigma = 0.3$. We use a standard normal prior distribution on x.

We run our test case for $N = 100$ particles and 20 iterations. Figure 5 shows contours of the posterior density and the samples produced by each of the three algorithms. Again, both the block-diagonal approximation and the inexact Newton–CG generate results of similar quality to those of the full Newton method.

These numerical results suggest that the block-diagonal approximation and the inexact Newton–CG can be effective methods for iteratively constructing the transport maps in SVN. We will adopt these approximate SVN strategies on large-scale problems, where computing the full Newton direction is not feasible.

C.2 Bayesian neural network

In this test case, we set up a Bayesian neural network as described in [17]. We use the open-source "yacht hydrodynamics" data set[2] and denote the data by $\mathcal{D} = (x_i, y_i)_{i=1}^{M}$, where x_i is an input, y_i is

[2] `http://archive.ics.uci.edu/ml/datasets/yacht+hydrodynamics`

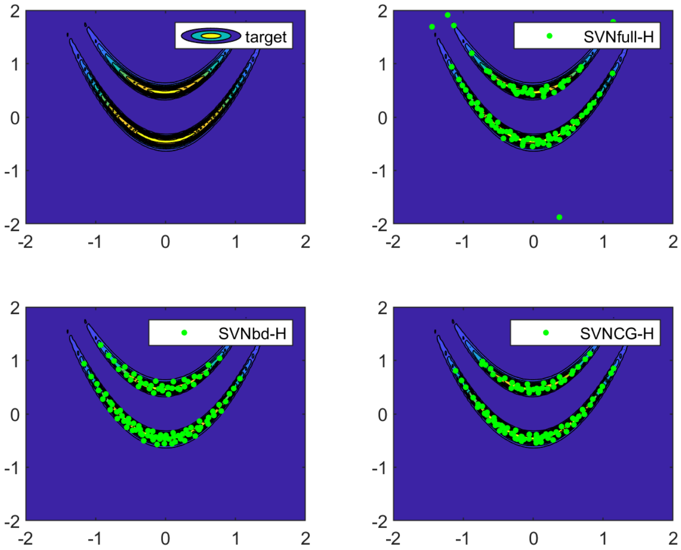


Figure 4: Double-banana example: performance comparison between `SVNfull`, `SVNCG`, and `SVNbd` after 20 iterations

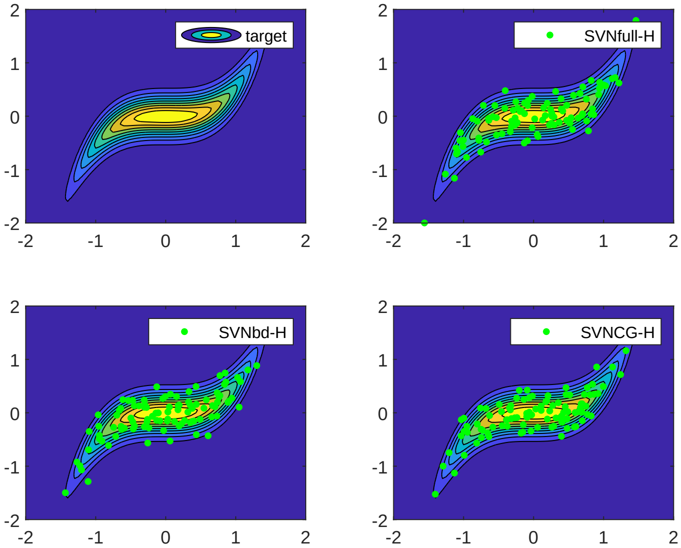


Figure 5: Nonlinear regression example: performance comparison between `SVNfull`, `SVNCG`, and `SVNbd` after 20 iterations

the corresponding scalar prediction, and $M = 308$. We divide the data into a training set of $m = 247$ input–prediction pairs and a validation set of $M - m = 61$ additional pairs. For each input, we model the corresponding prediction as

$$y_i = f(x_i, w) + \varepsilon_i \,,$$

where f denotes the neural network with weight vector $w \in \mathbb{R}^d$ and $\varepsilon_i \sim N(0, \gamma^{-1})$ is an additive Gaussian error. The dimension of the weight vector is $d = 2951$. We endow the weights w with independent Gaussian priors, $w \sim N(0, \lambda^{-1} I)$. The inference problem then follows from the likelihood function,

$$\mathcal{L}(\mathcal{D}|w,\gamma) = \left(\frac{\gamma}{2\pi}\right)^{\frac{m}{2}} \exp\left(-\frac{\gamma}{2}\sum_{i=1}^{m}(f(x,w)-y_i)^2\right),$$

and the prior density,

$$\pi_0(w|\lambda) = \left(\frac{\lambda}{2\pi}\right)^{\frac{m}{2}} \exp\left(-\frac{\gamma}{2}\sum_{i=1}^{m} w_j^2\right),$$

where γ and λ play the role of hyperparameters.

Performance comparison of SVN-H with SVGD-I. We compare SVN-H with the original SVGD-I algorithm on this Bayesian neural network example, with hyperparameters fixed to $\log\lambda = -10$ (which provides a very uninformative prior distribution) and $\log\gamma = 0$. First, we run a line-search with Newton–CG to find the posterior mode w^*. Figure 6 shows that neural network predictions at the posterior mode almost perfectly match the validation data. Then, we randomly initialise $n = 30$ particles $(x_i)_{i=1}^n$ around the mode, i.e., by independently drawing $x_i \sim \mathcal{N}(w^*, I)$. As in the previous test cases, we make a fair comparison of SVN-H and SVGD-I by taking 10, 20, and 30 iterations of SVN-H and rescaling the number of iterations of SVGD-I to match the computational costs of the two algorithms. Because this test case is very high-dimensional, rather than storing the entire Hessian matrix and solving the Newton system we use the inexact Newton–CG approach within SVN, which requires only matrix-vector products and yields enormous memory savings. Implementation details can be found in our GitHub repository.

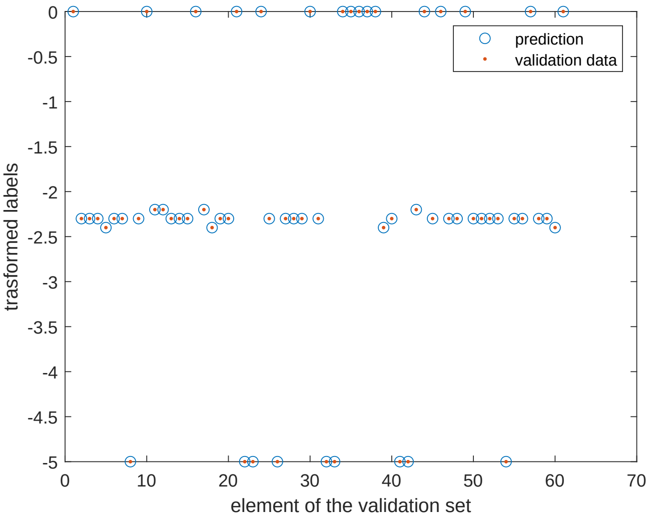


Figure 6: Neural network prediction at the posterior mode very closely matches the validation data.

Figure 7 shows distributions of the error on the validation set, as resulting from posterior predictions. To obtain these errors, we use the particle representation of the posterior on the weights w to evaluate posterior predictions on the validation inputs $(x_i)_{i=m+1}^M$. Then we evaluate the error of each of

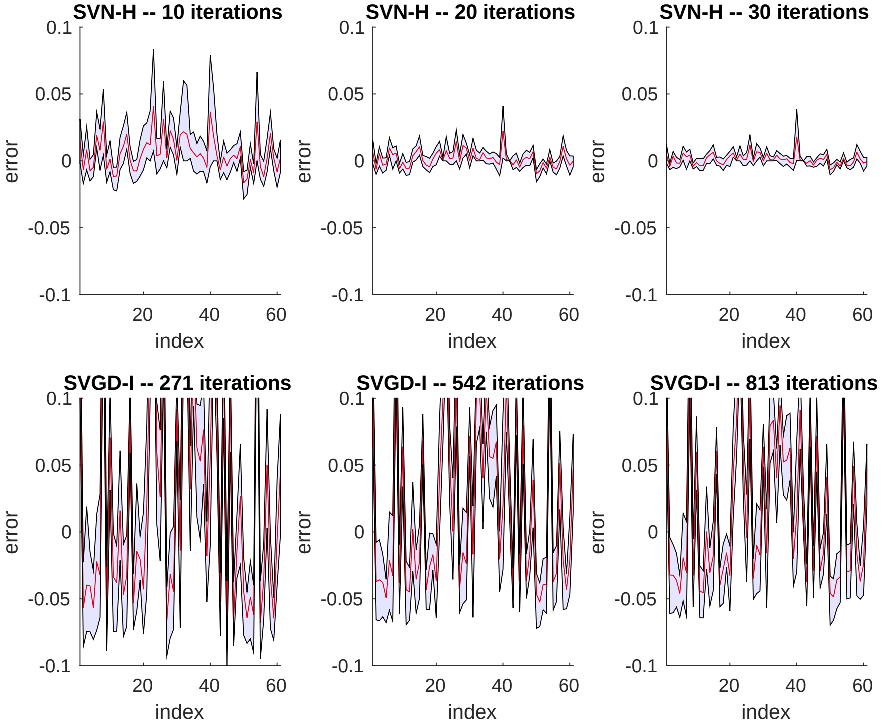


Figure 7: Bayesian neural network example: Comparison between SVN-H and SVGD-I, showing the distribution of errors between the validation data and samples from the posterior predictive.

these predictions. The red line represents the mean of these errors at each validation input x_i, and the shaded region represents the 90% credible interval of these error distribution. Although both algorithms "work" in the sense of producing errors of small range overall, SVN-H yields distributions of prediction error with smaller means and considerably reduced variances, compared to SVGD-I.

C.3 Scalability of kernels in high dimensions

Discretization-invariant posterior distribution. Here we illustrate the dimension scalability of the scaled Hessian kernel, compared to the isotropic kernel used in [17]. We consider a linear Bayesian inverse problem in a function space setting [26]: the forward operator is a linear functional $\mathcal{F}(x) = \langle \sin(\pi s), x(s) \rangle$, where the function x is defined for $s \in [0, 1]$. The scalar observation $y = \mathcal{F}(x) + \xi$, where ξ is Gaussian with zero mean and standard deviation $\sigma = 0.3$. The prior is a Gaussian measure $\mathcal{N}(0, \mathcal{K}^{-1})$ where $\mathcal{K}$ is the Laplace operator $-x''(s),\ s \in [0, 1]$, with zero essential boundary conditions.

Discretising this problem with finite differences on a uniform grid with d degrees of freedom, we obtain a Gaussian prior density $\pi_0(x)$ with zero mean and covariance matrix K^{-1}, where K is the finite difference approximation of the Laplacian. Let the vector a denote the discretised function $\sin(\pi s), s \in [0, 1]$, and let the corresponding discretised parameter be denoted by x (overloading notation for convenience). Then the finite-dimensional forward operator can be written as $\mathcal{F}(x) = a^\top x$. After discretization, the posterior has a Gaussian density of the form $\pi = \mathcal{N}(m_{\text{pos}}, C_{\text{pos}})$, where

$$m_{\text{pos}} = \frac{y}{\sigma^2} C_{\text{pos}}\, a\,, \qquad C_{\text{pos}} = \left(K^{-1} + \frac{1}{\sigma^2} a a^\top\right)^{-1}.$$

To benchmark the performance of various kernels, we construct certain summaries of the posterior distribution. In particular, we use our SVN methods with the scaled Hessian kernel (SVN-H) and the isotropic kernel (SVN-I) to estimate the component-wise average of the posterior mean, $\frac{1}{d}\sum_{i=1}^{d} m_{\text{pos},i}$, and the trace of the posterior covariance, $\text{trace}(C_{\text{pos}})$, for problems discretised at different resolutions $d \in \{40, 60, 80, 100\}$. We run each experiment with $n = 1000$ particles and 50 iterations of SVN. We compare the numerical estimates of these quantities to the analytically known results. These comparisons are summarised in Tables 1 and 2.

From Table 1, we can observe that all algorithms almost perfectly recover the average of the posterior mean up to the first three significant figures. However, Table 2 shows that SVN-H does a good job in estimating the trace of the posterior covariance consistently for all dimensions, whereas SVN-I under-estimates the trace—suggesting that particles are under-dispersed and not correctly capturing the uncertainty in the parameter x. This example suggests that the scaled Hessian kernel can lead to a more accurate posterior reconstruction for high-dimensional distributions than the isotropic kernel.

Table 1: Comparison of theoretical and estimated averages of the posterior mean

Averages of the posterior mean $\frac{1}{d}\sum_{i=1}^{d} m_{\text{pos},i}$				
d	40	60	80	100
Theoretical	0.4658	0.4634	0.4622	0.4615
SVN-H	0.4658	0.4634	0.4623	0.4614
SVN-I	0.4657	0.4633	0.4622	0.4615

Table 2: Comparison of theoretical and estimated traces of the posterior covariance

Traces of the posterior covariance trace(C_{pos})				
d	40	60	80	100
Theoretical	0.1295	0.1297	0.1299	0.1299
SVN-H	0.1271	0.1281	0.1304	0.1293
SVN-I	0.0925	0.0925	0.0925	0.0923

A posterior distribution that is not discretization invariant. Now we examine the dimension-scalability of various kernels in a problem that does not have a well-defined limit with increasing parameter dimension. We modify the linear Bayesian inverse problem introduced above: now the prior covariance is the identity matrix, i.e., $K^{-1} = I$ and the vector a used to define the forward operator is drawn from a uniform distribution, $a_i \sim \mathcal{U}(2, 10),\ i = 1, \ldots, d$. This way, the posterior is not discretization invariant. We perform the same set of numerical experiments as above and summarise the results in Tables 3 and 4. Although the target distribution used in this case is not discretization invariant, the scaled Hessian kernel is still reasonably effective in reconstructing the target distributions of increasing dimension (according to the summary statistics below), whereas the isotropic kernel under-estimates the target variances for all values of dimension d that we have tested.

Table 3: Comparison of theoretical and estimated averages of the posterior mean

Averages of the posterior mean $\frac{1}{d}\sum_{i=1}^{d} m_{\text{pos},i}$				
d	40	60	80	100
Theoretical	0.0037	0.0025	0.0019	0.0015
SVN-H	0.0037	0.0025	0.0019	0.0015
SVN-I	0.0037	0.0025	0.0019	0.0015

Table 4: Comparison of theoretical and estimated traces of the posterior covariance

Traces of the posterior covariance trace(C_{pos})				
d	40	60	80	100
Theoretical	39.0001	59.0000	79.0000	99.0000
SVN-H	37.7331	55.8354	73.6383	90.7689
SVN-I	8.7133	8.2588	7.9862	7.6876

References

[1] `http://github.com/gianlucadetommaso/Stein-variational-samplers`

[2] E. Anderes, M. Coram. A general spline representation for nonparametric and semiparametric density estimates using diffeomorphisms. *arXiv preprint arXiv:1205.5314*, 2012.

[3] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American mathematical society*, p. 337–404, 1950.

[4] D. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, p. 859–877, 2017.

[5] W. Y. Chen, L. Mackey, J. Gorham, F. X. Briol, C. J. Oates. Stein points. In *International Conference on Machine Learning. arXiv:1803.10161*, 2018.

[6] T. Cui, K. J. H. Law, Y. M. Marzouk. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304: 109–137, 2016.

[7] T. Cui, J. Martin, Y. M. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

[8] D. Francois, V. Wertz, and M. Verleysen. About the locality of kernels in high-dimensional spaces. *International Symposium on Applied Stochastic Models and Data Analysis*, p. 238–245, 2005.

[9] S. Gershman, M. Hoffman, D. Blei. Nonparametric variational inference. *International Conference on Machine Learning (ICML)*, 2012.

[10] W. R. Gilks, S. Richardson, and D. Spiegelhalter. Markov chain Monte Carlo in practice. *CRC press*, 1995.

[11] M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

[12] J. Han and Q. Liu. Stein variational adaptive importance sampling. *arXiv preprint arXiv:1704.05201*, 2017.

[13] M. E. Khan, Z. Liu, V. Tangkaratt, Y. Gal. Vprop: Variational inference using RMSprop. *arXiv preprint arXiv:1712.01038*, 2017.

[14] M. E. Khan, W. Lin, V. Tangkaratt, Z. Liu, D. Nielsen. Adaptive-Newton method for explorative learning. *arXiv preprint arXiv:1711.05560*, 2017.

[15] Q. Liu. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing systems* (I. Guyon et al., Eds.), Vol. 30, p. 3118–3126, 2017.

[16] Y. Liu, P. Ramachandran, Q. Liu, and J. Peng. Stein variational policy gradient. *arXiv preprint arXiv:1704.02399*, 2017.

[17] Q. Liu and D. Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems* (D. D. Lee et al., Eds.), Vol. 29, p. 2378–2386, 2016.

[18] C. Liu and J. Zhu. Riemannian Stein variational gradient descent for Bayesian inference. *Thirty-second aaai conference on artificial intelligence*, 2018.

[19] D. G. Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[20] J. Martin, L. C. Wilcox, C. Burstedde, and O. Ghattas. A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. *SIAM Journal on Scientific Computing*, 34(3), A1460–A1487, Chapman & Hall/CRC, 2012

[21] Y. M. Marzouk, T. Moselhy, M. Parno, and A. Spantini. Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, Springer, p. 1–41, 2016.

[22] R. M. Neal. MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo* (S. Brooks et al., Eds.), Chapman & Hall/CRC, 2011.

[23] Y. Pu, Z. Gan, R. Henao, C. Li, S. Han, and L. Carin. VAE Learning via Stein Variational Gradient Descent. *Advances in Neural Information Processing Systems (NIPS)*, 2017.

[24] D. Rezende and S. Mohamed. Variational inference with normalizing flows. *International Conference on Machine Learning (ICML)*, 2015.

[25] A. Spantini, D. Bigoni, and Y. Marzouk. Inference via low-dimensional couplings. *Journal of Machine Learning Research*, to appear. *arXiv:1703.06131*, 2018.

[26] A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19, p. 451–559, 2010.

[27] E. G. Tabak and T. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, p. 145–164, 2013.

[28] C. Villani. *Optimal Transport: Old and New*. Springer-Verlag Berlin Heidelberg, 2009.

[29] D. Wang, Z. Zeng, and Q. Liu. Stein variational message passing for continuous graphical models. *arXiv:1711.07168*, 2017.

[30] S. Wright, J. Nocedal. *Numerical Optimization*. Springer Science, 1999.

■ Paper IV

Stein Variational Online Changepoint Detection with Applications to Hawkes Processes and Neural Networks

Gianluca Detommaso [*]
InfoSec division
G-Research
London (UK)
detommaso.gianluca@gmail.com

Hanne Hoitzing
InfoSec division
G-Research
London (UK)
hanne.hoitzing@gresearch.co.uk

Tiangang Cui
School of Mathematical Sciences
Monash University
Melbourne (Australia)
tiangang.cui@monash.edu

Ardavan Alamir
InfoSec division
G-Research
London (UK)
ardavan.alamir@gresearch.co.uk

Abstract

Bayesian online changepoint detection (BOCPD) [1] offers a rigorous and viable way to identify changepoints in complex systems. In this work, we introduce a Stein variational online changepoint detection (SVOCD) method to provide a computationally tractable generalization of BOCPD beyond the exponential family of probability distributions. We integrate the recently developed Stein variational Newton (SVN) method [5] and BOCPD to offer a full online Bayesian treatment for a large number of situations with significant importance in practice. We apply the resulting method to two challenging and novel applications: Hawkes processes and long short-term memory (LSTM) neural networks. In both cases, we successfully demonstrate the efficacy of our method on real data.

1 Introduction

In most applied sciences and real-life scenarios, the ability to promptly detect and react to sudden changes is extremely desirable. Examples of current applications include hedge coverage in financial trading, attack detection in cybersecurity, prediction of natural disasters, and many others. In statistical analysis, the attempt to identify these changes is called *changepoint detection*.

Methods that fall under this category try to simultaneously minimize the following three important metrics: i) false negative rate, ii) false positive rate and iii) detection delay. False negatives must be avoided: missing the occurrence of an earthquakes could be fatal for thousands of people. Similarly, avoiding false positives has significant importance: too many alerts will hide ‘true’ changes, leading the analyst to underestimate important information and to lose confidence in the statistical methodology. Finally, most applications require a realtime reaction once new data is observed. Online algorithms should minimize detection delay, without undermining the first two metrics.

Among the literature in changepoint detection, probabilistic approaches have gained popularity for their ability to predict both the next observation and its uncertainty in an online fashion. A probabilistic approach that has significantly characterized the field is *Bayesian online changepoint*

[*] Corresponding author: Gianluca Detommaso

Preprint. Under review.

detection (BOCPD) [1]. In its original formulation, this method exploits conjugate priors to construct predictive models in closed form. Although conjugate priors constitute an ingenious tool to decrease detection delay, their simplicity also represents their major limitation: if the data substantially differ from the simple model in use, false negative and false positive rates are very large.

Several Bayesian inference methods have been proposed to extend BOCPD to non-conjugate scenarios [22, 21, 28, 29]. Although these methods form major contributions and have their own strengths, they also come with natural weaknesses. In this work, we propose a *Stein variational online changepoint detection* (SVOCD) method, a combination of BOCPD and Stein variational inference [18]. Stein variational inference is a cutting-edge Bayesian inference methodology which transports, sequentially and deterministically, a set of particles towards a posterior probability density. The advantage of SVOCD compared to the extensions mentioned above is twofold: i) rather than merely approximating the posterior density, the empirical density represented by the particles asymptotically converges to the posterior density as the number of particles increases [17], ii) rather than re-computing the posterior density from scratch when new data become available, it can be updated quickly which is crucial for online applications. Quick updates are possible because the posterior density can be used as the initial particle density to infer the next posterior density as new data points arrive. Assuming the true posterior does not change significantly, the particle locations can be adjusted with a few iterations. As a Stein variational algorithm we adopt Stein variational Newton (SVN), which was shown to drastically improve convergence speed and scalability to high-dimension compared to Stein variational gradient descent (SVGD) [5].

Additionally, we successfully apply our methodology to two complex models which represent a large number of real-life scenarios and currently lack a rigorous Bayesian changepoint analysis:[2] i) Hawkes processes, and ii) the combination of BOCPD with a Bayesian *long short-term memory* (LSTM) neural network model. Hawkes processes are an example of point processes for which the intensity increases with the occurrence of an event and exponentially decays over time; they have been used in a wide range of applications [7, 2, 23, 34]. To the best of our knowledge, this is the first attempt to perform online changepoint detection on Hawkes process in a fully Bayesian fashion. In many situations, no model exists to describe the evolution of data. LSTMs form a flexible modelling tool which can be trained to describe a sequence of data points and predict future points. However, the absence of an explicitly defined model structure can lead to large computational costs: LSTM's descriptive power comes from over-parametrization, making training computationally intensive and likely to end up in flat regions in parameter space. In this paper, we try to overcome these issues by combining a Bayesian formulation of LSTM with BOCPD and train the model using SVN.

The paper is structured as follows: Section 2 describes the background of BOCPD and SVN, and then presents SVOCD. In Sections 3 and 4, we apply SVOCD to Hawkes processes and LSTM, respectively. A conclusion is stated in section 5.

2 Stein variational online changepoint detection

In this section, we introduce Stein variational online changepoint detection (SVOCD). SVOCD generalizes BOCPD to probability distributions beyond the exponential family by using the Stein variational Newton method to perform online inference.

2.1 Background on Bayesian online changepoint detection

Suppose we sequentially observe data points $y_{1:m}$, where the subscript denotes the observation time. Assuming that each observation y_i depends on a model driven by a hidden parameter $\theta \in \mathbb{R}^d$, changepoint detection aims to identify abrupt changes in the parameter θ. We denote a *changepoint* by a time index $\tau > 1$ at which the abrupt change in θ occurs. We will focus on online changepoint detection: given past observations $y_{1:m}$, we want to detect whether θ at time $m+1$ is the same as θ at time m. We want to perform this task recursively as new data becomes available.

Bayesian online changepoint detection (BOCPD) has been introduced as a probabilistic approach for online detection of changepoints in a time series [1]. The algorithm has pioneered a considerable

[2] Open source code is available at gianlucadetommaso/Stein-variational-samplers..

amount of interesting follow-up work. Here we provide a description of the general formulation of BOCPD.[3] BOCPD adopts the following reasonable assumption.

A1. Observed data before and after changepoints are independent. That is, y_i is independent of y_j if there exists a changepoint τ such that $i < \tau \leq j$. This way, the dynamics of the underlying system after a changepoint is not affected by what happened before the changepoint.

Let us define $\tau_{m+1} \in \{1, \ldots, m+1\}$ to be the changepoint indicator at time $m+1$ which records the time of the occurrence of the last changepoint. The case $\tau_{m+1} = 1$ indicates there has been no changepoint up until time $m+1$. Although a priori τ_{m+1} can assume any value between 1 and $m+1$, in practice one should consider pruning the possible set of changepoints according to their posterior probability for significant computational speed-ups [30].

Predictive posterior. Suppose we have observed $y_{1:m}$ and we want to detect whether y_{m+1} is a changepoint. For this purpose, we introduce the predictive posterior density $p(y_{m+1}|y_{1:m})$, which measures the probability that y_{m+1} is observed given $y_{1:m}$. However, because of assumption **A1**, y_{m+1} is only dependent on observations since the last changepoint τ_{m+1}. Then, if we define $Y_{\tau_{m+1}} := \{\tau_{m+1}, y_{\tau_{m+1}:m}\}$ to be the information set given by both the changepoint τ_{m+1} and the sequence of observed data points $y_{\tau_{m+1}:m}$ (we define $y_{m+1:m} = \emptyset$), we can marginalize the predictive posterior density as follows:

$$p(y_{m+1}|y_{1:m}) = \sum_{\tau_{m+1}=1}^{m+1} p(y_{m+1}|Y_{\tau_{m+1}})\, p(\tau_{m+1}|y_{1:m})\,. \tag{1}$$

We will now analyse the two factors on the right-hand-side of equation (1).

Predictive model. $p(y_{m+1}|Y_{\tau_{m+1}})$ denotes the predictive probability given the last changepoint τ_{m+1}. By marginalising y_{m+1} over the hidden parameter θ, we can write

$$p(y_{m+1}|Y_{\tau_{m+1}}) = \int p(y_{m+1}|Y_{\tau_{m+1}}, \theta)\, p(\theta|Y_{\tau_{m+1}})\, d\theta, \tag{2}$$

where $p(\theta|Y_{\tau_{m+1}})$ denotes the posterior distribution of θ and we refer to $p(y_{m+1}|Y_{\tau_{m+1}}, \theta)$ as the *predictive likelihood.*

In original BOCPD, the authors exploit conjugate priors for exponential families of probability distributions to express the predictive density (2) in closed form. In section 2, we will generalize BOCPD to non-exponential families of probability distributions by introducing Stein variational Newton. This enables us to accurately approximate $p(y_{m+1}|Y_{\tau_{m+1}})$ for more complex model choices that can better represent the data and their changepoints, while keeping the detection delay small.

Changepoint posterior. $p(\tau_{m+1}|y_{1:m})$ denotes the posterior probability of the changepoint indicator τ_{m+1}. Using an approach analogous to [1], it is easy to show that the joint probability of τ_{m+1} and $y_{1:m}$ can be recursively expressed as

$$p(\tau_{m+1}, y_{1:m}) = \sum_{\tau_m=1}^{m} p(y_m|Y_{\tau_m})\, p(\tau_{m+1}|\tau_m)\, p(\tau_m, y_{1:m-1}). \tag{3}$$

Hence, the joint density on the left-hand-side of (3) can be evaluated by a forward message-passing algorithm which stores the joint density evaluations at the previous iteration and updates them accordingly. The posterior density can then be recovered by normalizing the joint density via $p(y_{1:m}) = \sum_{\tau_{m+1}=1}^{m+1} p(\tau_{m+1}, y_{1:m})$. Note that, given τ_m, we can only have either $\tau_{m+1} = \tau_m$ if y_{m+1} follows the same dynamics as y_m, or $\tau_{m+1} = m+1$ if $m+1$ is a changepoint. Then, we define the *changepoint prior* density $p(\tau_{m+1}|\tau_m)$ as being equal to either H_m (if $\tau_{m+1} = m+1$) or $1 - H_m$ (if $\tau_{m+1} = \tau_m$), where H_m can be interpreted as a *hazard rate*. Hence, whenever $\tau_{m+1} \neq m+1$, the sum over τ_m in (3) reduces to a single term with $\tau_m = \tau_{m+1}$.

2.2 Background on Stein variational Newton

Consider an intractable target density π on $\mathbb{R}^d$ that we wish to approximate via an empirical measure or, equivalently, a collection of particles. Given a set of particles $(\theta^{(k)})_{k=1}^{N_\theta}$ characterizing an initial

[3] We adopt a formulation without the concept of *run length*, however the method is equivalent.

reference density q_0, we seek a transport map $T : \mathbb{R}^d \to \mathbb{R}^d$ such that $T_* q_0$, the push-forward map of q_0 through T, is a close approximation of π.[4] Such a map T is not unique: there exist an infinite number of such maps that can serve the purpose [31]. In the following, we construct T as a composition of simple maps T_l which are iteratively applied on reference densities q_l such that $q_{l+1} = T_{l*} q_l$. We define each T_l as a perturbation Q_l of the identity map:

$$T_l(\theta) = \theta + Q_l(\theta)\,. \tag{4}$$

When applied to the current reference density q_l, equation (4) defines the push-forward measure $T_{l*} q_l$ as an update of q_l itself along the direction Q_l. The latter will be taken along a vector-valued *Reproducing Kernel Hilbert Space* (RKHS) $\mathcal{H}^d \simeq \mathcal{H} \times \cdots \times \mathcal{H}$ characterized by a kernel $k(\cdot,\cdot)$.

A variational approach. We define the functional

$$Q \mapsto J_{q_l}[Q] \coloneqq \mathcal{D}_{\text{KL}}((I+Q)_* \, q_l \,||\, \pi)\,, \tag{5}$$

with $Q \in \mathcal{H}^d$. $J_{q_l}[Q]$ measures the Kullback-Leibler (KL) divergence $\mathcal{D}_{\text{KL}}$ between the push-forward map of q_l, along the direction Q, and π. Thus, we want to find a map Q_l such that $J_{q_l}[Q_l] < J_{q_l}[\mathbf{0}]$, where $\mathbf{0}(\theta) = 0$ denotes the zero map. In other words, we are constructing a sequence of densities $q_0, q_1, q_2, \ldots$ that weakly converges to π (see [17] for convergence results).

It was shown in [18, 5] how to define a functional gradient $\nabla J_{q_l}[\mathbf{0}]$ and functional Hessian $\nabla J_{q_l}[\mathbf{0}]$ of the map in (5), where $\mathbf{0}$ is the null map, which symbolize the evaluation of the variational information at the current density q_l. For details about the methodology we refer to [18, 5]. Here we report the following theorem.

Theorem 1 *With the notation above, we have*

$$\nabla J_{q_l}[\mathbf{0}](\phi) = -\mathbb{E}_{\theta\sim q_l}[\nabla_\theta \log \pi(\theta) k(\theta,\phi) + \nabla_\theta k(\theta,\phi)]\,, \tag{6}$$

$$\nabla^2 J_{q_l}[\mathbf{0}](\phi,\psi) = \mathbb{E}_{\theta\sim q_l}[-\nabla^2_\theta \log \pi(\theta) k(\theta,\phi) k(\theta,\psi) + \nabla_\theta k(\theta,\phi) \nabla_\theta k(\theta,\psi)^\top]\,. \tag{7}$$

Given the variational informations in (6) and (7), Q_l is constructed via a Newton-type iteration (see the supplementary material for details). The overall method is addressed as *Stein variational Newton* (SVN) and a possible implementation is described in Algorithm 1.

2.3 A new method: BOCPD via SVN

Here we introduce a novel method: Stein variational online changepoint detection (SVOCD). This algorithm generalizes BOCPD to non-exponential families of probability distributions. The $(m+1)$-th iteration of SVOCD is described in Algorithm 2.3, which we break down into the following steps.

Changepoint posterior update (line 3). Given samples $(\theta^{(k)}_{\tau_m})_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau_m})$ and the changepoint posterior $p(\tau_m|y_{1:m-1})$ from the previous iteration, this step aims to update the changepoint posterior by the recurrent relation in (3) given the new observation y_m. We observe that this involves the evaluation of the posterior probability $p(y_m|Y_{\tau_m})$, which is a not available explicitly for a non-exponential family of probability distributions. However, because the samples $(\theta^{(k)}_{\tau_m})_{k=1}^{N_\theta}$ are available to us, we can simply estimate it by the Monte Carlo approach

$$p(y_m|Y_{\tau_m}) \approx \frac{1}{N_\theta} \sum\nolimits_{k=1}^{N_\theta} p(y_m|Y_{\tau_m}, \theta^{(k)}_{\tau_m})\,. \tag{8}$$

Samples update (line 4). Next, we use SVN to generate samples $(\theta^{(k)}_{\tau_{m+1}})_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau_{m+1}})$. Note that, given the changepoint τ_{m+1}, we can only have either $\tau_{m+1} = m+1$ in the case of a changepoint or $\tau_{m+1} = \tau_m$ otherwise. In the case $\tau_{m+1} = m+1$, the information set $Y_{\tau_{m+1}}$ contains no data points and, as a consequence, the posterior distribution $p(\theta|Y_{\tau_{m+1}})$ corresponds to the prior $p(\theta)$. A collection of samples $(\theta^{(k)}_{\tau_{m+1}})_{k=1}^{N_\theta}$ can now simply be taken from this prior. In the case $\tau_{m+1} = \tau_m$, we have $Y_{\tau_{m+1}} = Y_{\tau_m} \cup \{y_m\}$ which means the following decomposition holds:

$$p(\theta|Y_{\tau_{m+1}}) \propto p(y_m|Y_{\tau_m}, \theta)\, p(\theta|Y_{\tau_m})\,. \tag{9}$$

[4] If T is an invertible map, the push-forward map is defined by $T_* q(\theta) = q(T^{-1}(\theta))\,|\det(\nabla_\theta T(\theta))|$.

The relation in (9) shows that we can recast $p(\theta|Y_{\tau_{m+1}})$ as a sequential update of the previous posterior $p(\theta|Y_{\tau_m})$ through the information given by y_m. The transport method of SVN very well fits this framework: SVN can be initialized using the current particles $(\theta^{(k)}_{\tau_m})_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau_m})$, which are then adjusted to get $(\theta^{(k)}_{\tau_{m+1}})_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau_{m+1}})$. Since the initialization of the particles is optimal up to the available information Y_{τ_m}, the algorithm most likely needs only a few iterations to converge, particularly if the amount of information that y_{m+1} adds to Y_{τ_m} is small.

Data prediction (line 6). Given the updated changepoint posterior and particles, Algorithm 2 is a standard valid mechanism [20] to produce samples $(y^{(i)})_{i=1}^{N_y}$ from the predictive posterior density $p(y|y_{1:m})$. We can use these samples to work out a prediction for the next observation y_{m+1} and statistics summarizing the distribution, for example left and right quantiles y_ℓ and y_r in the case of one-dimensional data.

Data classification (line 7). Finally, the data y_{m+1} is observed and immediately alerted as a changepoint if it does not belong to the credible interval $[y_\ell, y_r]$.

Algorithm 1 l-th iteration of (block-diagonal) SVN

```
1: Input: Particles $(\theta^{(k)}_l)_{k=1}^{N_\theta}$ at stage $l$; step size $\varepsilon$
2: Output: Particles $(\theta^{(k)}_{l+1})_{k=1}^{N_\theta}$ at stage $l+1$
3: for $k = 1, 2, \dots, N_\theta$ do
4:     Evaluate the gradient $\nabla J_{q_l}[\mathbf{0}](\theta^{(k)}_l)$ in (11)
5:     Evaluate the Hessian $H(\theta^{(k)}_l, \theta^{(k)}_l)$ in (14)
6:     Solve the linear system
           $H(\theta^{(k)}_l, \theta^{(k)}_l)\, Q_l(\theta^{(k)}_l) = -\nabla J_l[\mathbf{0}](\theta^{(k)}_l)$
7:     Update the particle
           $\theta^{(k)}_{l+1} \leftarrow \theta^{(k)}_l + \varepsilon Q_l(\theta^{(k)}_l)$
8: end for
```

Algorithm 2 Sample $y^{(i)} \sim p(y^{(i)}|y_{1:m})$

```
1: Sample $\tau^{(i)} \sim p(\tau^{(i)}|y_{1:m})$
2: Randomly select $\theta^{(i)}$ from $\big(\theta^{(k)}_{\tau^{(i)}}\big)_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau^{(i)}})$
3: Sample $y^{(i)} \sim p(\cdot|Y_{\tau^{(i)}}, \theta^{(i)})$
```

Algorithm 3 $(m+1)$-th iteration of SVOCD

```
1: Input: $(\theta^{(k)}_{\tau_m})_{k=1}^{N_\theta}$ and $p(\tau_m|y_{1:m-1})$ for each $\tau_m$
2: for $\tau_{m+1} = 1, \dots, m+1$ do
3:     Evaluate $p(\tau_{m+1}|y_{1:m})$ via (16) and (3)
4:     Sample $(\theta^{(k)}_{\tau_{m+1}})_{k=1}^{N_\theta} \sim p(\theta|Y_{\tau_{m+1}})$ via Alg. 1
5: end for
6: Sample $(y^{(i)})_{i=1}^{N_y}$ by Alg. 2 and calculate $\bar{y}, y_\ell, y_r$
7: Observe $y_{m+1}$ and alert changepoint if $y_{m+1} \notin [y_\ell, y_r]$
```

Remarks. We stress that the loops over τ_{m+1} can be executed in parallel. In particular, parallelizing the samples update step is fundamental to massively speed up the algorithm. Additional steps for pruning the set of possible values of τ_{m+1} or for optimizing over the hyper-parameters could be added [30, 33] to Algorithm 2.3, but this goes beyond the scope of this paper.

3 Application to Hawkes processes

In this section, we apply SVOCD to Hawkes processes: common self-exciting point processes that play a central role in analysing time series in a range of applications such as telecommunications, epidemiology, and neuroscience. Though frequentist's methods have previously been developed [16, 24], this is, to the best of our knowledge, the first fully-Bayesian online treatment of detecting changepoints in Hawkes processes.

Hawkes processes. Unlike standard inhomogeneous Poisson processes, the intensity function of a self-exciting process directly depends on the occurrence of past events, which can "excite" the arrival of future events. In a Hawkes process, the rate of arrivals bursts whenever an event occurs, and decays over time. We denote the sequence $(y_k)_{k\geq 1}$ to be the arrival times of the process. Given $Y_{\tau_{m+1}}$, the rate of arrival of the next event y_{m+1} can be described by the following *conditional intensity*:

$$\lambda_{\tau_{m+1}}(t) := \mu + \gamma \sum\nolimits_{\substack{y_k \in Y_{\tau_{m+1}} \\ y_k < t}} e^{-\delta(t-y_k)}\,, \tag{10}$$

where $t > 0$, $\mu > 0$ is the baseline intensity rate, $\gamma > 0$ represents how much the intensity bursts whenever an event occurs, and $\delta > 0$ represents the decay rate of the intensity function. When no event has arrived yet, a Hawkes process behaves like a homogeneous Poisson process with parameter μ. We define $\theta := [\mu, \gamma, \delta]^\top$ as the 3-dimensional vector collecting all parameters. More general definitions of Hawkes processes could be considered (e.g. marked Hawkes processes, power decay functions, ...) [27], but we restrict ourselves to the most common one.

Given that we observed the events in $Y_{\tau_{m+1}}$ within the time interval $(y_{\tau_{m+1}-1}, y_m]$, it can be shown that the predictive likelihood function for the next event y_{m+1} is given by

$$p(y_{m+1}|Y_{\tau_{m+1}},\theta) = \lambda_{\tau_{m+1}}(y_{m+1})\, e^{-\Lambda_{\tau_{m+1}}\left((y_m, y_{m+1}]\right)}, \tag{11}$$

where $\Lambda_{Y_{\tau_{m+1}}}(\mathcal{I}) := \int_{\mathcal{I}} \lambda_{Y_{\tau_{m+1}}}(t)\, dt$ is known as a *compensator*, for some time interval $\mathcal{I}$ [25]. The likelihood function can then be explicitly defined as

$$p(y_{\tau_{m+1}:m}|\tau_{m+1},\theta) \;=\; \prod\nolimits_{i=\tau_{m+1}}^{m} \lambda_{\tau_{m+1}}(y_i)\, e^{-\Lambda_{\tau_{m+1}}(\mathcal{I}_{\tau_{m+1}})}\,, \tag{12}$$

where $\mathcal{I}_{\tau_{m+1}} := (y_{\tau_{m+1}-1}, y_m]$. In order to enforce positivity for each component of θ, we impose a log-normal prior distribution, i.e. $\ln\theta \sim \mathcal{N}(\mu_0, \sigma_0^2 I)$, where μ_0 and σ_0 are hyper-parameters. Using Bayes' Theorem, we have

$$p(\theta|Y_{\tau_{m+1}}) \propto p(y_{\tau_{m+1}:m}|\theta,\tau_{m+1})\, p(\theta)\,. \tag{13}$$

A choice for the Hessian. In order to apply SVOCD to Hawkes processes, a positive definite approximation of the Hessian of the log-likelihood density, $\nabla_\theta^2 \log p(y_{\tau_{m+1}:m}|\tau_{m+1},\theta)$, is required. We represent this approximation by the asymptotic Fisher information, shown [26] to be given by

$$H_{\mathcal{L},\tau_{m+1}}(\theta) \;:= \sum\nolimits_{y_i \in Y_{\tau_{m+1}}} \nabla_\theta \log \lambda_{\tau_{m+1}}(y_i) \nabla_\theta \log \lambda_{\tau_{m+1}}(y_i)^\top\,. \tag{14}$$

An approximation of the Hessian of the log-posterior density $\nabla_\theta^2 \log p(\theta|Y_{\tau_{m+1}})$ can then be given by

$$H_{\pi,\tau_{m+1}}(\theta) = \sigma_0^2 I + H_{\mathcal{L},\tau_{m+1}}(\theta)\,. \tag{15}$$

We note that when calculating the gradient $\nabla_\theta \log p(\theta|Y_{\tau_{m+1}})$, each $\nabla \log \lambda_{Y_{\tau_{m+1}}}(y_i)$ in (14) needs to be evaluated and hence the calculation of $H_{\pi,\tau_{m+1}}(\theta)$ does not require additional operations.

Validation via SMC. To benchmark the performance of SVOCD, we will also employ BOCPD using Sequential Monte Carlo (SMC) with adaptive systematic resampling [6] to update θ samples (line 4, Algorithm 3). The importance density is taken as the Laplace approximation of the posterior, i.e. a Gaussian centered at the MAP with covariance matrix the inverse of the Hessian evaluated at the MAP. We do not aim to face the difficult task of a rigorous performance comparison, but rather we introduce an alternative not requiring structural choices of proposal or approximating densities.

Application: WannaCry cyber attack. WannaCry caught world headlines in May 2017 by infecting over 200,000 computers and causing damages worth at least in the hundreds of millions of dollars. In this section, we consider the packet capture traffic logs of the WannaCry spread through three computers in a test environment.[5] The spread of the malware triggers a snowball effect of logs as each computer gets infected. In order to capture this self-exciting phenomenon, we employ a Hawkes process to model the log arrivals in time and perform online changepoint detection to efficiently detect when the three computers become infected.

The data contains 207 time observations. The prior distribution for $\ln\theta$ was deliberately chosen as an uninformative Gaussian with parameters $\mu_0 = 0$ and $\sigma_0^2 = 10$. As we look for sudden bursts of activity, we construct a one-sided credible interval by taking y_r as the 95th percentile and we signal a changepoint m whenever $y_m > y_r$. The hazard rate in the changepoint prior and the number of predictive samples were fixed at $H_m = 100$ and $N_y = 100$, respectively.

Figure 1 displays the results of both SVOCD as well as BOCPD in conjuction with SMC applied to the WannaCry data. In the top figures, the blue line represents the observations; vertical jumps indicate that no log events occur in that time interval, whereas horizontal regions indicate event arrivals close together. The red line is the average of the predictive distribution, attempting to reconstruct the data; the green shadowed area represents the credibility region up to the right 95th percentile of the distribution; the vertical red lines are the detected changepoints. For the bottom figures we reverted the axes so that the blue line represents a counting process which increases by 1 every time an observation occurs.

[5] Data can be found here: https://www.malware-traffic-analysis.net/2017/05/18/index2.html.

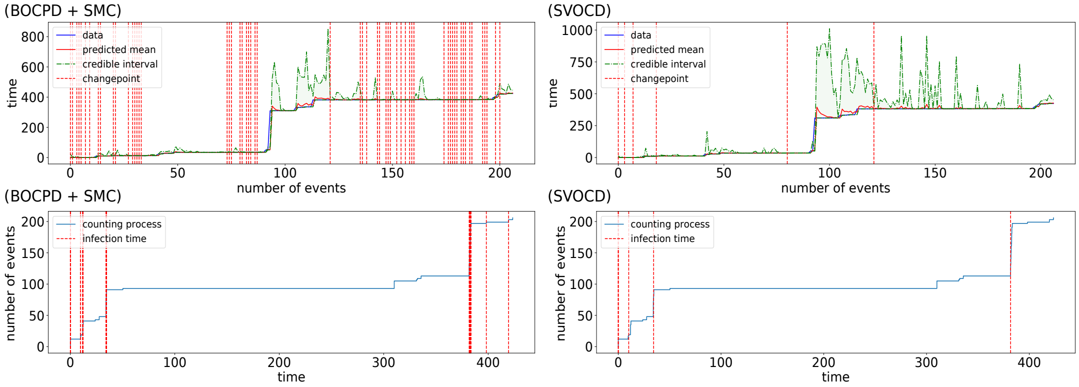


Figure 1: Hawkes model on WannaCry data. (left) BOCPD + SMC: several false positives are identified along the three infections. (right) SVOCD: after burn-in, only three changepoints are detected corresponding to the three infections. In the SVN algorithm, we used $N_\theta = 100$ particles and only 30 iterations. For BOCPD + SMC, 1000 particles were used.

We find that SVOCD takes some time to adapt at the beginning of the time series, which is to be expected as the prior distribution is not yet properly tuned to the data. However, the algorithm quickly adapts and detects three meaningful changepoints. Thus, apart from the initial burn-in phase, all the detected changepoints correspond to drastic bursts in activity, i.e. to the infections of the three computers, and no false positives were detected. For BOCPD + SMC, the algorithm keeps detecting changepoints without adapting to changes in data trends. Although these changepoints correspond to actual bursts in activity, several false positives are detected along with the machine infections.

	SVOCD	BOCPD+SMC
False pos. rate	1.03 (0.91)	1.43 (0.99)
False neg. rate	0.37 (0.48)	0.60 (0.49)
	SVN	SMC
MSE at $\tau = 15$	$\sim 3{\times}10^{-2}$	$\sim 2{\times}10^{1}$

Table 1: Quantitative comparisons between SVOCD and BOCPD + SMC show lower false positive and false negative rates (mean (std)) for SVOCD (30 iterations). The MSE (as defined in the text) of SVN at an arbitrarily chosen timepoint is about 3 orders of magnitude lower than that of SMC when using 500 particles.

Synthetic data. In order to provide a more direct quantitative comparison, a synthetic Hawkes trajectory of 60 events was constructed with a changepoint every 10th event (see appendix C for details). We measure the trace of the covariance matrix of the posterior via MCMC, and then calculate its mean squared error (MSE) via SVN and SMC (details in appendix C). Figure 2 shows how the MSE changes as a function of the number of particles N_θ, at three different values of τ. As N_θ increases, the error of SVN decreases much faster than that of SMC. Even when using 10^4 particles for SMC, its error at $\tau = 15, 35$ remains three orders of magnitude larger compared to SVN with 500 particles. A video[6] visualizing the changing posterior contours over time confirms SVN's superiority in tracking changes in complex non-Gaussian distributions: whereas SVN's particles accurately define the posterior, the locations of SMC's particles show instabilities and jump between peaks in the distribution, failing to capturate it as a whole. Table 1 summarizes quantitative comparisons between the two methods, including false positive and negative rates of changepoint detection, showing a better performance for SVOCD over BOCPD + SMC.

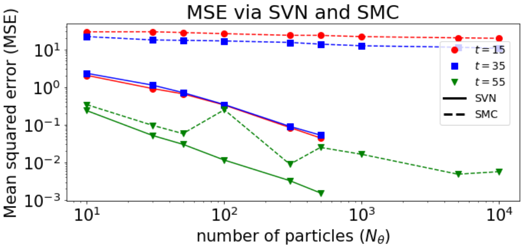


Figure 2: SVN shows a lower mean squared error compared to SMC for lower N_θ.

4 Application to long short-term memory neural networks

In this section, we adopt *Bayesian long short-term memory* (BLSTM, [14, 9, 32]) neural networks as a predictive model. We note that although frequentist's methods have been proposed to detect

[6] https://gfycat.com/blaringforthrightbullfrog

changepoints in LSTM [12], this is, to the best of our knowledge, the first fully-Bayesian online changepoint analysis for LSTM. To demonstrate the effectiveness of SVOCD, we also apply BOCPD using a similar SMC method as in Section 3 to perform the parameter sampling step.

BLSTM. Here we describe our Bayesian approach to LSTM. For simplicity, we use a time series of scalar data points $y_m \in \mathbb{R}$; our analysis can readily be extended to more general cases. Consider $\mathcal{F}_{y_{\tau:m}}(\theta)$ to be the output of the forward pass of a many-to-one LSTM, trained on data $y_{\tau:m}$ and evaluated at $\theta \in \mathbb{R}^d$. The latter contains all the unknown weights and biases in the architecture of the network. If the training set of LSTM is empty, the network will output the bias of the last layer. It will prove to be useful to also construct the corresponding many-to-many LSTM defined by $\boldsymbol{F}_{\tau:m} = [\mathcal{F}_\emptyset, \mathcal{F}_{y_\tau}, \mathcal{F}_{y_{\tau:\tau+1}} \ldots, \mathcal{F}_{y_{\tau:m}}]$.

When $\tau = \tau_{m+1}$, the output of $\mathcal{F}_{y_{\tau_{m+1}:m}}(\theta)$ is considered a noisy prediction of the data point y_{m+1}:

$$y_{m+1} = \mathcal{F}_{\tau_{m+1}:m}(\theta) + \sigma\xi\,, \tag{16}$$

where $\xi \sim \mathcal{N}(0, 1)$ and $\sigma > 0$. Equation (16) is equivalent to defining the predictive likelihood

$$p(y_{m+1}|Y_{\tau_{m+1}}, \theta) = \mathcal{N}(\mathcal{F}_{\tau_{m+1}:m}(\theta), \sigma^2)(y_{m+1})\,, \tag{17}$$

where the right-hand-side of (17) denotes a Gaussian density evaluated at y_{m+1} with mean $\mathcal{F}_{\tau_{m+1}:m}(\theta)$ and variance σ^2. From equation (17) and the relation between $\mathcal{F}_{\tau_{m+1}:m}$ and $\boldsymbol{F}_{\tau_{m+1}:m}$, we find that the likelihood is given by

$$p(y_{\tau_{m+1}:m}|\theta, \tau_{m+1}) {=} \mathcal{N}(\boldsymbol{F}_{\tau_{m+1}:m}(\theta), \sigma^2 I)(y_{\tau_{m+1}:m})\,. \tag{18}$$

Finally, we define a Gaussian prior $p(\theta) = \mathcal{N}(\mu_0, \sigma_0^2 I)$ and use Bayes' theorem as in (13) to obtain the posterior distribution.

Backprop and Fisher Information. Here we describe how to calculate the Fisher Information of the log-likelihood, which will be used in the SVN algorithm. In deterministic LSTM, backpropagation consists of a gradient descent step which runs backwards, from the last data point to the first. In a Bayesian framework, this corresponds to calculating the gradient of the log-likelihood density:

$$\nabla \log p(y_{\tau_{m+1}:m}|\theta, \tau_{m+1}) = \frac{1}{\sigma^2} \sum\nolimits_{i=\tau_{m+1}-1}^{m-1} \nabla_\theta \mathcal{F}_{\tau_{m+1}:i}(\theta)^\top (\mathcal{F}_{\tau_{m+1}:i}(\theta) - y_{i+1})\,. \tag{19}$$

Given the Gaussian error assumption in (18), the Fisher Information of the likelihood is given by

$$H_{\mathcal{L},\tau_{m+1}}(\theta) := \frac{1}{\sigma^2} \sum\nolimits_{i=\tau_{m+1}-1}^{m-1} \nabla_\theta \mathcal{F}_{\tau_{m+1}:i}(\theta)^\top \nabla_\theta \mathcal{F}_{\tau_{m+1}:i}(\theta)\,. \tag{20}$$

The Hessian $H_{\pi,\tau_{m+1}}$ of the log-posterior density can now be approximated as in (15).

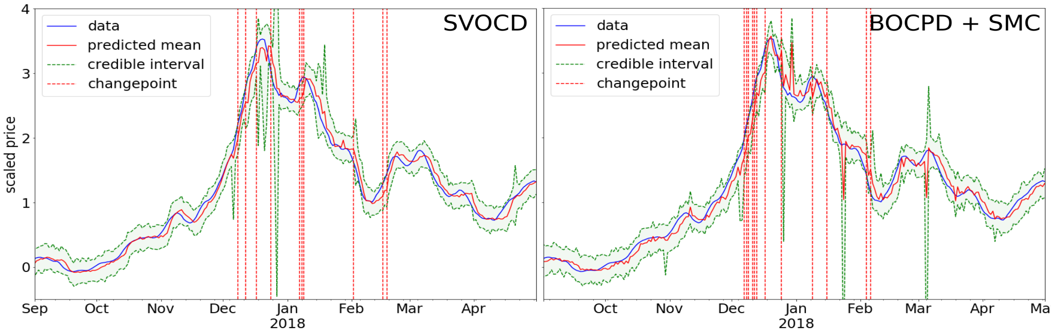


Figure 3: Changepoint detection on bitcoin data: (left) SVOCD with BLSTM model; (right) BOCPD + SMC with BLSTM model. BODPC + SMC has more difficulty adapting to changes in trend. SVOCD and BOCPD + SMC were simulated using $N_\theta = 30$ and $N_\theta = 100$ particles, respectively. 100 iterations were used.

Application: bitcoin price. Bitcoin is a cryptocurrency created in 2009 whose value has fluctuated wildly in the last few years. We apply SVOCD and BOCPD + SMC with a BLSTM model to data on bitcoin price evolution (Figure 2 in the supplement), with $\theta \in \mathbb{R}^{64}$. We use a standard Gaussian $\mathcal{N}(0, I)$ as a prior and a noise level $\sigma = 0.1$ in the likelihood. The hazard rate and the number of predictive samples were fixed at $H_m = 1000$ and $N_y = 100$, respectively.

Figure 3 shows that SVOCD starts to detect changepoints from December 2017 due to a large increase in stock price. After the all-time peak, as the price starts to decrease steeply, another changepoint is detected. Various others are found corresponding to large fluctuations in price. BOCPD + SMC detects changepoints in similar locations, though the increased number of changepoints in the rising phase indicates a difficulty in adapting to changes in trend. In addition, the predicted mean is rougher and less accurate than the one produced by SVOCD, despite using more particles in the simulation.

5 Conclusion

In this work we introduced SVOCD, a fully-Bayesian method that combines BOCPD and SVN to detect changepoints both online and accurately. We successfully applied SVOCD to novel and challenging applications, namely Hawkes processes and LSTM neural networks on WannaCry and Bitcoin real data sets, respectively. A quantitative comparison between SVN and SMC shows that SVN, given its transport nature, is able to carry forward the current estimation of the posterior density which leads to more accurate estimations compared to SMC, even when the number of particles used for SMC is an order of magnitude higher. Further comparisons between SVOCD and BOCPD + SMC, using synthetic data with known changepoints, showed that the former has lower false positive and false negative rates. Because SVOCD samples from the correct posterior, it is able to quickly adapt to changes in trends, to return informative changepoints and to avoid false positives.

6 Acknowledgements

This work was fully sponsored by G-Research. We thank the InfoSecurity division of G-Research for their support. In particular, we would like to acknowledge David Thomas for facilitating the project and Antoine Vianey-Liaud for setting up the WannaCry data set.

A Appendix: SVN for Bayesian LSTM

In this section, we validate the use of SVN on a Bayesian LSTM model in order to sample correctly from a posterior distribution. In addition, we provide evidence that the prediction given by a Bayesian LSTM is substantially better than the one using a corresponding regularized LSTM.

We use the following simple test case: the data is generated by a noisy sinusoidal signal:

$$y_j = \sin(j) + \xi \, ,$$

where $j = 0, \ldots, 50$ are time indices and $\xi \sim \mathcal{N}(0, 0.15^2)$ is Gaussian noise. We attempt to reconstruct the data $y_{1:50}$ and its uncertainty by using the Bayesian LSTM model described in the paper. We set a Gaussian prior $p(\theta) = \mathcal{N}(0, 1)$. We further assume a likelihood of the form $\mathcal{N}(\mathcal{F}_{1:49}(\theta), 0.3^2)$, where $\mathcal{F}_{1:49}(\theta)$ is the output of the forward pass of a many-to-many LSTM trained on $y_{1:49}$ and evaluated at parameters $\theta \in \mathbb{R}^{64}$. In order to train the BLSTM model, we use SVN with 30 particles initialized around the MAP of the distribution and run it for 100 iterations.

Figure 4 displays the results of our simulation. The blue line is the real data. The red line is the average of the particles representing our prediction. The green shaded area represents a 95% credible interval around the mean. We can see how the red prediction captures the sinusoidal motion of the signal, while the uncertainty of the signal is well represented by the green area. In contrast, the dashed magenta line is the mode of the distribution, i.e. the estimator that would be returned by a deterministic regularized LSTM. We can see very clearly that the mode of the distribution overweights the importance of the last observations: at every stage the magenta line almost exactly replicates the previous observation.

In conclusion, we find that: i) SVN is able to correctly represent the posterior distribution, and ii) a Bayesian framework is superior to a deterministic one: it allows us to calculate the average of the posterior distribution, leading to much better predictions compared to using the mode found by deterministic models.

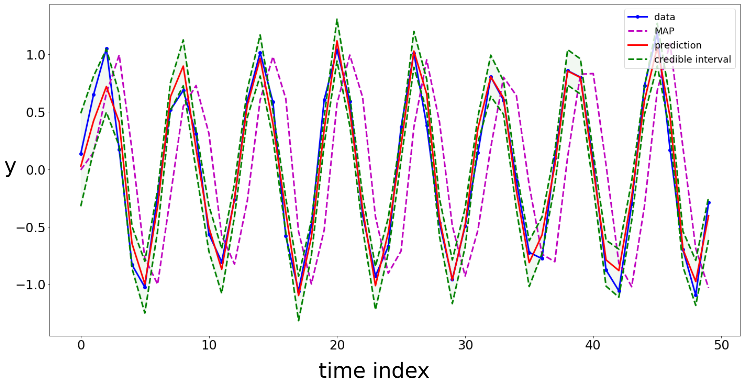


Figure 4: BLSTM trained with SVN on sinusoidal signal

B Appendix: Bitcoin price changepoint detection

Figure 5 shows the weekly rolling-averaged data of the evolution of bitcoin price from the beginning of 2016 to the 13th of December 2018. Price started at $998\$$ in 2017 and rose to $13,412.44\$$ on the 1st of January 2018, with an all-time peak of $19,666\$$ on the 17th of December 2017. From that point on, the price fluctuated downwards up to $3,690\$$ at the end of 2018, about 81% down from the all-time peak.

Figure 3 shows the results of SVOCD in the region where the price dynamics reaches its all-time peak and then suddenly drops (outside this region, predictions are very stable and no changepoints are detected).

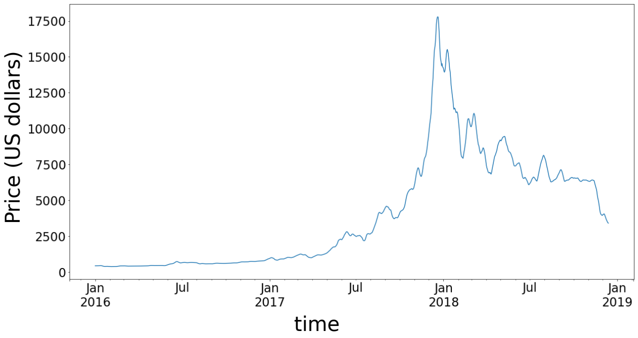


Figure 5: Bitcoin price evolution in US dollars.

C Appendix: Synthetic data comparison

We constructed 60 data points $y_{1:60}$ where each of these observations is generated by a Hawkes process driven by a parameter $\theta = [\mu, \gamma, \delta] \in \mathbb{R}^3$. A changepoint is inserted after every 10th observation, i.e. changepoints occur at $\tau = 10, 20, \ldots, 60$. Between these changepoints, the generating parameter θ alternates between two states, namely $\theta_1 = [-1, -2, 1]$ and $\theta_2 = [2, 4, 0]$.

We set a standard Gaussian prior and a Hawkes process likelihood (12). The goal is to sequentially retrieve samples from the posteriors $p(\theta | Y_{\tau_{m+1}})$, for $m = 0, 1, \ldots, 59$. We do this via both SVN and SMC ($N = 500$ particles) and compare their performance. Figure 6 shows snapshots of the video[7] we made to visualize the performance of SVN and SMC on sequentially tracking the changes of the posterior over time. From these snapshots, we clearly see that SMC fails to accurately describe the posterior density when the geometry of the density is highly non-Gaussian.

[7] https://gfycat.com/blaringforthrightbullfrog

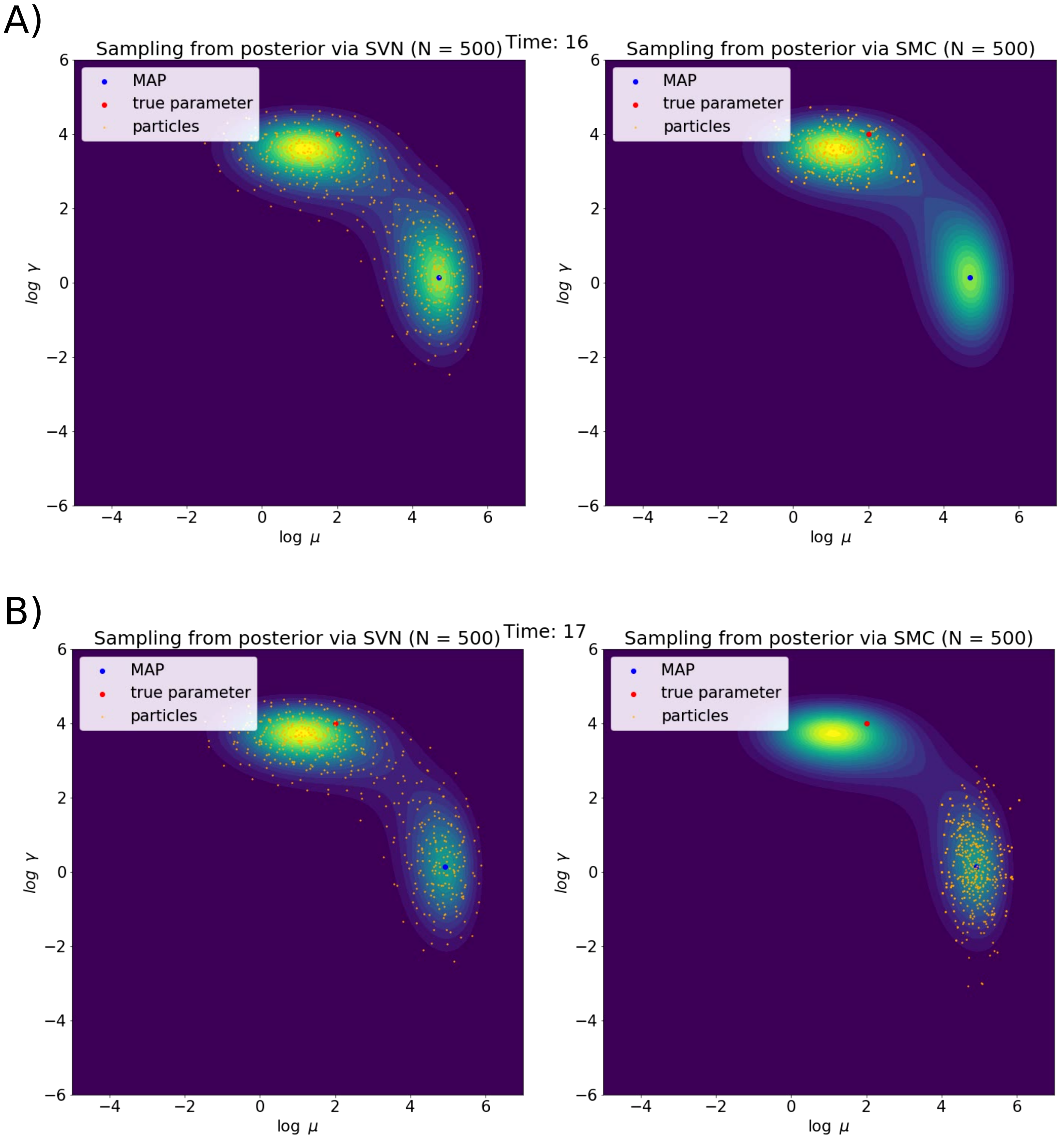


Figure 6: **SVN more accurately estimates the posterior density.** We visualize the changing posterior contours, as well as the particle positions of SVN and SMC, at time points A) $\tau = 16$ and B) $\tau = 17$. While SVN's particles accurately describe the posterior, SMC's particles tend to jump from one peak to another without properly describing the entire distribution. These plots were generated using the synthetic data as described above.

In order to produce Figure 2, the same synthetic data was used. First, at each time step, we measure the trace of the covariance matrix of the posterior distribution calculated via a random walk MCMC with 5×10^5 steps. We then also retrieve posterior samples via SVN and SMC for a range of values of N ($N = 10, 30, 50, 100, 300, 500, 1000$ for SVN and $N = 10, 30, 50, 100, 300, 500, 1000, 5000, 10000$ for SMC). For each value of N, 30 runs are performed to measure the mean and variance of the mean squared error (MSE), calculated as the mean squared difference between the traces of the covariance matrix via the random walk and via SVN or SMC. In Figure 2, we can clearly see a better convergence for SVN compared to SMC, confirming the better performance of SVN as observed in the video.

Furthermore, on the same synthetic data, we perform 30 runs of both SVOCD and BOCPD + SMC to measure mean and standard deviation of false positive and false negative changepoint rates for both methods. Table 1 shows that, as expected from the better convergence results discussed previously, SVOCD achieves smaller means for both false positive and false negative rates.

References

[1] Ryan Prescott Adams and David JC MacKay. “Bayesian online changepoint detection”. In: *arXiv preprint arXiv:0710.3742* (2007).

[2] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. “Hawkes processes in finance”. In: *Market Microstructure and Liquidity* 1.01 (2015), p. 1550005.

[3] François Caron, Arnaud Doucet, and Raphael Gottardo. “On-line changepoint detection and parameter estimation with application to genomic data”. In: *Statistics and Computing* 22.2 (2012), pp. 579–595.

[4] Angelos Dassios, Hongbiao Zhao, et al. “Exact simulation of Hawkes process with exponentially decaying intensity”. In: *Electronic Communications in Probability* 18.62 (2013), pp. 1–13.

[5] Gianluca Detommaso et al. “A Stein variational Newton method”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 9187–9197.

[6] Arnaud Doucet and Adam M Johansen. “A tutorial on particle filtering and smoothing: Fifteen years later”. In: *Handbook of nonlinear filtering* 12.656-704 (2009), p. 3.

[7] Paul Embrechts, Thomas Liniger, and Lu Lin. “Multivariate Hawkes processes: an application to financial data”. In: *Journal of Applied Probability* 48.A (2011), pp. 367–378.

[8] Paul Fearnhead and Zhen Liu. “On-line inference for multiple changepoint problems”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.4 (2007), pp. 589–605.

[9] Meire Fortunato, Charles Blundell, and Oriol Vinyals. “Bayesian recurrent neural networks”. In: *arXiv preprint arXiv:1704.02798* (2017).

[10] Damien Francois, Vincent Wertz, Michel Verleysen, et al. “About the locality of kernels in high-dimensional spaces”. In: *International Symposium on Applied Stochastic Models and Data Analysis*. Citeseer. 2005, pp. 238–245.

[11] Roman Garnett, Michael A Osborne, and Stephen J Roberts. “Sequential Bayesian prediction in the presence of changepoints”. In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 345–352.

[12] Tian Guo et al. “Robust online time series prediction with recurrent neural networks”. In: *Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference on*. Ieee. 2016, pp. 816–825.

[13] Nicholas A Heard and Melissa JM Turcotte. “Adaptive sequential Monte Carlo for multiple changepoint analysis”. In: *Journal of Computational and Graphical Statistics* 26.2 (2017), pp. 414–423.

[14] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[15] Erik Lewis et al. “Self-exciting point process models of civilian deaths in Iraq”. In: *Security Journal* 25.3 (2012), pp. 244–264.

[16] Shuang Li et al. “Detecting changes in dynamic events over networks”. In: *IEEE Transactions on Signal and Information Processing over Networks* 3.2 (2017), pp. 346–359.

[17] Qiang Liu. “Stein variational gradient descent as gradient flow”. In: *Advances in neural information processing systems*. 2017, pp. 3115–3123.

[18] Qiang Liu and Dilin Wang. “Stein variational gradient descent: A general purpose Bayesian inference algorithm”. In: *Advances In Neural Information Processing Systems*. 2016, pp. 2378–2386.

[19] Song Liu et al. “Change-point detection in time-series data by relative density-ratio estimation”. In: *Neural Networks* 43 (2013), pp. 72–83.

[20] Scott M Lynch. *Introduction to applied Bayesian statistics and estimation for social scientists*. Springer Science & Business Media, 2007.

[21] Lida Mavrogonatou and Vladislav Vyshemirsky. “Sequential importance sampling for online Bayesian changepoint detection”. In: *22nd International Conference on Computational Statistics*. 2016, pp. 73–84.

[22] Scott Niekum et al. “Online Bayesian changepoint detection for articulated motion models”. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE. 2015, pp. 1468–1475.

[23] Yosihiko Ogata. “Space-time point-process models for earthquake occurrences”. In: *Annals of the Institute of Statistical Mathematics* 50.2 (1998), pp. 379–402.

[24] Matthew Price-Williams and Nick Heard. “Statistical Modelling of Computer Network Traffic Event Times”. In: *arXiv preprint arXiv:1711.10416* (2017).

[25] Jakob Gulddahl Rasmussen. “Lecture Notes: Temporal Point Processes and the Conditional Intensity Function”. In: *arXiv preprint arXiv:1806.00221* (2018).

[26] Alex Reinhart et al. “A review of self-exciting spatio-temporal point processes and their applications”. In: *Statistical Science* 33.3 (2018), pp. 299–318.

[27] Marian-Andrei Rizoiu et al. “A Tutorial on Hawkes Processes for Events in Social Media”. In: *arXiv preprint arXiv:1708.06401* (2017).

[28] Yunus Saatçi, Ryan D Turner, and Carl E Rasmussen. “Gaussian process change point models”. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. Citeseer. 2010, pp. 927–934.

[29] Ryan D Turner, Steven Bottone, and Clay J Stanek. “Online variational approximations to non-exponential family change point models: with application to radar tracking”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 306–314.

[30] Ryan Turner, Yunus Saatci, and Carl Edward Rasmussen. “Adaptive sequential Bayesian change point detection”. In: *Advances in Neural Information Processing Systems: Temporal Segmentation Workshop*. 2009.

[31] Cédric Villani. *Optimal transport: old and new*. Vol. 338. Springer Science & Business Media, 2008.

[32] Jos van der Westhuizen and Joan Lasenby. “Bayesian LSTMs in medicine”. In: *arXiv preprint arXiv:1706.01242* (2017).

[33] Robert C Wilson, Matthew R Nassar, and Joshua I Gold. “Bayesian online learning of the hazard rate in change-point problems”. In: *Neural computation* 22.9 (2010), pp. 2452–2476.

[34] Ke Zhou, Hongyuan Zha, and Le Song. “Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes”. In: *Artificial Intelligence and Statistics*. 2013, pp. 641–649.

■ Paper V

HINT: Hierarchical Invertible Neural Transport for Density Estimation and Bayesian Inference

Jakob Kruse [*1] **Gianluca Detommaso** [*2] **Robert Scheichl** [1] **Ullrich Köthe** [1]

Abstract

A large proportion of recent invertible neural architectures is based on a coupling block design. It operates by dividing incoming variables into two sub-spaces, one of which parameterizes an easily invertible (usually affine) transformation that is applied to the other. While the Jacobian of such a transformation is triangular, it is very sparse and thus may lack expressiveness. This work presents a simple remedy by noting that (affine) coupling can be repeated recursively within the resulting sub-spaces, leading to an efficiently invertible block with dense triangular Jacobian. By formulating our recursive coupling scheme via a hierarchical architecture, HINT allows sampling from a joint distribution $p(\mathbf{y}, \mathbf{x})$ and the corresponding posterior $p(\mathbf{x} \,|\, \mathbf{y})$ using a single invertible network. We demonstrate the power of our method for density estimation and Bayesian inference on a novel data set of 2D shapes in Fourier parameterization, which enables consistent visualization of samples for different dimensionalities.

1. Introduction

Invertible neural networks based on the normalizing flow principle have recently gained increasing attention for generative modeling tasks. Arguably the most popular group of such networks are those built on the coupling block design. Their success is due to a number of useful properties setting them apart from other generative approaches:

(a) they offer tractable likelihood computation for any sample or data point, *(b)* training via the maximum likelihood criterion is generally very stable, *(c)* their interpretable and easily accessible latent space opens up possibilities for e.g. style transfer and *(d)* the same trained model can be used for efficient data generation as well as efficient density estimation.

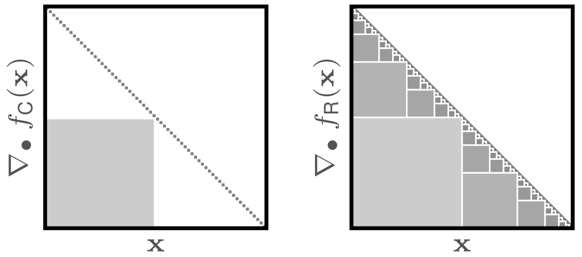

$\nabla \bullet f_C(\mathbf{x})$
$\mathbf{x}$
$\nabla \bullet f_R(\mathbf{x})$
$\mathbf{x}$

Figure 1. Sparse triangular Jacobian of a single coupling *(left)* and dense triangular Jacobian of hierarchical coupling block *(right)*. Gray marks populated areas of the matrix, the remainder is empty.

While autoregressive models can also be trained as normalizing flows and share the former two properties, they sacrifice efficient invertibility for expressive power and thus lose the latter two properties. In contrast, the expressive power of a single invertible block is a core limitation of invertible networks, leading to impractically deep models with dozen or hundreds of blocks, as is exemplified by the GLOW architecture of Kingma & Dhariwal (2018). While invertibility actually allows to backpropagate through very deep networks with minimal memory footprint (Gomez et al., 2017), the search for more expressive invertible building blocks remains an important question.

The superior performance of autoregressive approaches such as (Van den Oord et al., 2016a) stems from the fact that they model a larger range of interactions between variables, as reflected in the dense triangular Jacobian matrix of the mapping they represent. From the theory of transport maps we know that certain guarantees of universality exist for triangular maps. These do not hold for the standard coupling block design, which has a comparatively sparse Jacobian as shown in figure 1 *(left)*.

In this work we propose an extension to the coupling block design that recursively fills out the previously unused portions of the Jacobian until we obtain a dense triangular map (figure 1, *right*), or any intermediate design if we choose to stop early. The advantages of the original coupling block architecture are retained in the process.

[*] Equal contribution [1] Heidelberg University, Germany [2] University of Bath, United Kingdom. Correspondence to: Jakob Kruse <jakob.kruse@iwr.uni-heidelberg.de>.

Preprint, under review. Code for data preparation and experiments: https://github.com/VLL-HD/HINT

We furthermore show that the recursive structure of this mapping can be used to split the variables of interest into two subsets $\mathbf{x}$ and $\mathbf{y}$, such that the transformation of $\mathbf{x}$ is modeled conditionally on $\mathbf{y}$. This property can be exploited for a new sampling scheme, where a single normalizing flow model efficiently generates samples from both the joint distribution $p(\mathbf{x}, \mathbf{y})$ and the conditional $p(\mathbf{x} \mid \mathbf{y})$ of the factorized variables.

Finally, we introduce a new family of data sets based on 2d Fourier curve parameterizations. In the normalizing flow literature, there is an abundance of two-dimensional toy densities that provide an easy visual check for correctness of the model's output. Beyond two dimensions however, it is challenging to visualize the distribution or even individual samples produced by a model. Image data sets solve this problem, but are quickly of high enough dimensionality that it becomes infeasible to evaluate statistical baselines like Approximate Bayesian Computation (ABC) to compare against.

A step towards visualizable data sets of intermediate size has been made in (Kruse et al., 2019), but their four-dimensional problems are still two simple to demonstrate the advantages of the recursive coupling approach described above. To fill the gap, we describe a way to generate data sets of arbitrary dimensionality such that each data point parameterizes a closed curve in 2d space which is easy to visualize. With more data dimensions, i.e. more degrees of freedom, distributions of more detailed curves can be represented.

To summarize, the contributions of this paper to the field of normalizing flow research are as follows:

- A simple, efficiently invertible flow module with dense lower triangular Jacobian
- A hierarchical coupling architecture that models joint and conditional distributions simultaneously
- A novel family of data sets allowing 2d visualization for a flexible number of dimensions

The remainder of this work is consists of a literature review (section 2), some mathematical background (section 3), a description of our method (section 4) and numerical experiments (section 5), followed by some closing remarks (section 6).

2. Related Work

Normalizing flows were popularized in the context of deep learning chiefly by the work of Rezende & Mohamed (2015) and Dinh et al. (2015). At present, a large variety of different architectures exist to realize normalizing flows in practice. A comprehensive overview of many of these, as well as general background on invertible neural networks and normalizing flows, can be found in Kobyzev et al. (2019) and in the simultaneous work of Papamakarios et al. (2019).

Many existing networks fall into one of two groups, namely coupling block architectures and autoregressive models. First additive and then affine coupling blocks were introduced by Dinh et al. (2015; 2017). Kingma & Dhariwal (2018), besides demonstrating the power of flow networks as generators, went on to generalize the permutation of variables between blocks by learning the corresponding matrices. There have been a number of works that focus on improving on the limiting nature of the affine transformation at the core of most coupling block networks. E.g. Durkan et al. (2019) replace affine couplings with more expressive monotonous splines, albeit at the cost of evaluation speed.

On the other hand, there is a rich body of research into autoregressive (flow) networks for various purposes, ranging from Van den Oord et al. (2016b;a) over Kingma et al. (2016) and Papamakarios et al. (2017) to Huang et al. (2018). More recently, Jaini et al. (2019) applied second-order polynomials to improve expressive power compared to typical autoregressive models, and proved that their model is a universal density approximator. While such models are known for excellent density estimation results compared to coupling architectures (Ma et al., 2019; Liao et al., 2019), generating samples is often not a priority and can be prohibitively slow.

Outside of these two subfields, there are other approaches towards a favorable trade-off between expressive power and efficient invertibility.

Residual Flows (Behrmann et al., 2019; Chen et al., 2019) impose Lipschitz constraints on a standard residual block, which guarantees invertibility with a full Jacobian and enables approximate maximum likelihood training but requires an iterative procedure for sampling. Similarly, Song et al. (2019) use lower triangular weight matrices which can be inverted via fixed-point iteration. The normalizing flow principle is formulated continuously as a differential equation by Grathwohl et al. (2019), which allows free-form Jacobians but requires integrating an ODE for each network pass. Karami et al. (2019) introduce another method with dense Jacobian based on invertible convolutions performed in the Fourier domain.

In terms of modeling conditional densities with invertible neural networks, Ardizzone et al. (2019a) proposed an approach that divides the network output into conditioning variables and latent vector, training the flow part with an MMD (Gretton et al., 2012) objective instead of maximum likelihood. Later Ardizzone et al. (2019b) introduced a simple conditional coupling block from which a conditional normalizing flow can be constructed.

3. Background

For an input vector $\mathbf{x} \in \mathbb{R}^N$, the function defined by a standard invertible coupling block is

$$f_{\mathrm{C}}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 \\ C\big(\mathbf{x}_2 \,|\, \mathbf{x}_1\big) \end{bmatrix} \tag{1}$$

with $\mathbf{x}_1 = \mathbf{x}_{0:\lfloor N/2 \rfloor}$ and $\mathbf{x}_2 = \mathbf{x}_{\lfloor N/2 \rfloor:N}$ being the first and second half of the input vector and $C(\mathbf{x}_2 \,|\, \mathbf{x}_1)$ an easily invertible transform of $\mathbf{x}_2$ conditioned on $\mathbf{x}_1$. Its inverse is then simply given as

$$f_{\mathrm{C}}^{-1}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_1 \\ C^{-1}\big(\mathbf{x}_2 \,|\, \mathbf{x}_1\big) \end{bmatrix}. \tag{2}$$

In the case of an affine coupling block (Dinh et al., 2017), C takes the form $C(\mathbf{u} \,|\, \mathbf{v}) = \mathbf{u} \odot \exp\big(s(\mathbf{v})\big) + t(\mathbf{v})$ with s and t unconstrained feed-forward networks. The logarithm of this block's Jacobian determinant can be computed very efficiently as

$$\log \big| \det \mathbf{J}_{f_{\mathrm{C}}}(\mathbf{x}) \big| = \log \left| \det \frac{\partial f_{\mathrm{C}}(\mathbf{x})}{\partial \mathbf{x}} \right| = \sum s(\mathbf{x}_1)\,. \tag{3}$$

To ensure that all entries of $\mathbf{x}$ are transformed, and interact with each other in different combinations, they construct a pipeline that alternates between coupling blocks and random orthogonal matrices $\mathbf{Q}$, where $f_{\mathbf{Q}}(\mathbf{x}) = \mathbf{Q}\mathbf{x}$ can trivially be inverted as $f_{\mathbf{Q}}^{-1}(\mathbf{x}) = \mathbf{Q}^{\top}\mathbf{x}$ and has Jacobian log-determinant $\log \big|\det \mathbf{J}_{f_{\mathbf{Q}}}(\mathbf{x})\big| = 0$.

3.1. Normalizing flows and transport maps

To create a normalizing flow, this pipeline $T = f_{\mathrm{C}1} \circ f_{\mathbf{Q}1} \circ f_{\mathrm{C}2} \circ f_{\mathbf{Q}2} \circ \ldots$ is trained via maximum likelihood loss

$$\mathcal{L}(\mathbf{x}) = \tfrac{1}{2}\|T(\mathbf{x})\|_2^2 - \log |\mathbf{J}_T(\mathbf{x})| \tag{4}$$

to transport the data distribution $p_X(\mathbf{x})$ to a standard normal latent distribution $p_Z(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$.

We can draw samples from $p_X(\mathbf{x})$ via T by drawing latent samples $\mathbf{z}$ from $p_Z(\mathbf{z})$ and passing them through the inverse model: $\mathbf{x} = T^{-1}(\mathbf{z})$. Using the change-of-variables formula, we can also evaluate the density $p_X(\mathbf{x})$ of a given data point $\mathbf{x}$ as $p_X(\mathbf{x}) = p_Z(T(\mathbf{x})) \cdot |\det \mathbf{J}_T(\mathbf{x})|$.

This procedure is founded on the theory of transport maps (Villani, 2008), which are employed in exactly the same way to push a reference density (e.g. Gaussian) to a target density (e.g. data distribution, Marzouk et al. (2016)).

The objective in equation (4) is in fact the KL divergence between the data distribution $p_X(\mathbf{x})$ and the pull-back $T_{\#}^{-1}$ of the latent density $p_Z(\mathbf{z})$, with $H(p_X)$ the unknown fixed entropy of the data distribution:

$$\begin{aligned} D_{\mathrm{KL}}(p_X \,\|\, T_{\#}^{-1} p_Z) &= \int p_X(\mathbf{x}) \log \frac{p_X(\mathbf{x})}{T_{\#}^{-1} p_Z(\mathbf{x})} \,\mathrm{d}\mathbf{x} \\ &= -\int p_X(\mathbf{x}) \log\big(p_Z(T(\mathbf{x}))\mathbf{J}_T(\mathbf{x})\big) \,\mathrm{d}\mathbf{x} + H(p_X) \\ &= \mathbb{E}_{\mathbf{x} \sim p_X}[\mathcal{L}(\mathbf{x})] + \text{const.}. \end{aligned}$$

Note also that each pair $f_{\mathrm{C}i} \circ f_{\mathbf{Q}i}$ in T is a composition of an orthogonal transformation and a triangular map, where the latter is better known in the field of transport maps as a *Knothe-Rosenblatt rearrangement* (Marzouk et al., 2016). This factorization can be seen as a non-linear generalization of the classic QR decomposition (Stoer & Bulirsch, 2013). Whereas the triangular part encodes the possibility to represent non-linear transformations, the orthogonal part reshuffles variables to foster dependence of each part of the input to the final output, thereby drastically increasing the representational power of the map T.

3.2. Bayesian inference with conditional flows

When a data set does not just represent a distribution of data points $\mathbf{x}$, but consists of tuples $(\mathbf{x}, \mathbf{y})$, the labels/features/observations $\mathbf{y}$ don't have a clear place in the normalizing flow framework described above. It is of course possible to just concatenate $\mathbf{x}$ and $\mathbf{y}$ into a single vector and let the transport $T([\mathbf{x}, \mathbf{y}]^{\top})$ model their joint distribution. But in many cases the paired data stems from a known (probabilistic) forward process $\mathbf{x} \to \mathbf{y}$ and what we are interested in is the inverse process, in particular evaluating and sampling from the conditional density $p(\mathbf{x} \,|\, \mathbf{y})$. This task is called Bayesian inference.

Ardizzone et al. (2019b) and Winkler et al. (2019) independently introduced *conditional* coupling blocks that allow an entire normalizing flow to be conditioned on external variables. By conditioning the transport T between $p_X(\mathbf{x})$ and $p_Z(\mathbf{z})$ on the corresponding values of $\mathbf{y}$ as $\mathbf{z} = T(\mathbf{x} \,|\, \mathbf{y})$, its inverse $T^{-1}(\mathbf{z} \,|\, \mathbf{y})$ can be used to turn the latent distribution $p_Z(\mathbf{z})$ into an approximation of the posterior $p(\mathbf{x} \,|\, \mathbf{y})$. A similar idea can be found in Marzouk et al. (2016).

The maximum likelihood objective from equation (4) is readily adjusted to the conditional setting as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \tfrac{1}{2}\|T(\mathbf{x} \,|\, \mathbf{y})\|_2^2 - \log |\mathbf{J}_T(\mathbf{x} \,|\, \mathbf{y})|\,, \tag{5}$$

where the Jacobian is w.r.t. $\mathbf{x}$, i.e. $\mathbf{J}_T(\mathbf{x} \,|\, \mathbf{y}) = \nabla_{\mathbf{x}} T(\mathbf{x} \,|\, \mathbf{y})$. This is equivalent to minimizing the KL divergence between $p_{X,Y}(\mathbf{x}, \mathbf{y})$ and the pull-back of a standard normal distribution through the inverse of T:

$$\begin{aligned} &D_{\mathrm{KL}}(p_{X,Y} \,\|\, T(\,\cdot\,|Y)_{\#}^{-1} p_Z) \\ &\quad = \mathbb{E}_{\mathbf{x},\mathbf{y} \sim p_{X,Y}}[\mathcal{L}(\mathbf{x}, \mathbf{y})] + H(p_{X,Y}). \end{aligned}$$

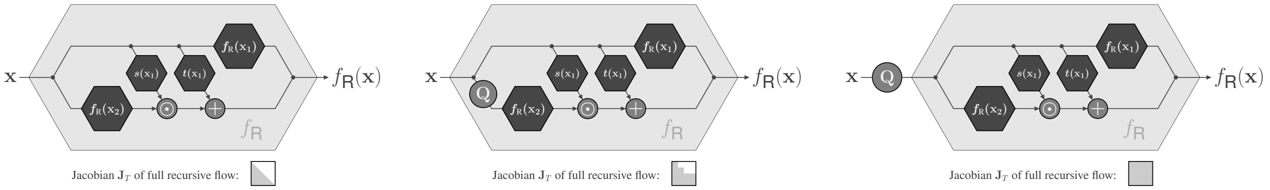


Figure 2. Recursive affine coupling blocks. In each case, the inner functions $f_R(\mathbf{x}_i)$ again take the form of the outer gray block, repeated until the vectors $\mathbf{x}_i$ can not be split any further. Each block in itself has triangular Jacobian. Placement of the orthogonal permutation matrix $\mathbf{Q}$ determines how the Jacobian $\mathbf{J}_T$ of a composition T of many blocks is populated. *Left:* No permutation means T is a strict Knothe-Rosenblatt rearrangement. *Middle:* Permutation in the lower branch only induces a recursive conditional structure where variables in the lower branches can never influence variables in the upper branches. *Right:* Permutation between coupling blocks yields a map with full Jacobian. Expressive power of the map increases from left to right while interpretability decreases.

4. Method

We will now extend the basic coupling block architecture described above in two ways.

4.1. The recursive coupling block

As visualized in figure 1 *(left)*, the Jacobian $\mathbf{J}_f$ of a simple coupling block is very sparse, i.e. many possible interactions between variables are not modelled. However the efficient determinant computation in equation (3) works for any lower triangular $\mathbf{J}_f$, and indeed Theorem 1 of Hyvärinen & Pajunen (1999) states that a single triangular transformation can, in theory, already represent arbitrary distributions.

To make use of this potential and fill the empty areas below the diagonal in $\mathbf{J}_{f_C}$, we propose a recursive coupling scheme

$$f_R(\mathbf{x}) = \begin{cases} f_C(\mathbf{x}), & \text{if } \mathbf{x} \in \mathbb{R}^{N \le 3} \\ \begin{bmatrix} f_R(\mathbf{x}_1) \\ C\left(f_R(\mathbf{x}_2) \,\middle|\, \mathbf{x}_1\right) \end{bmatrix}, & \text{otherwise} \end{cases} \tag{6}$$

where each sub-space $\mathbf{x}_1, \mathbf{x}_2$ is again split and transformed until we end up with single dimensions (or stop early). Note that each sub-coupling has its own – e.g. affine – coupling function C with independent parameters. The inverse transform is

$$f_R^{-1}(\mathbf{x}) = \begin{cases} f_C^{-1}(\mathbf{x}), & \text{if } \mathbf{x} \in \mathbb{R}^{N \le 3} \\ \begin{bmatrix} f_R^{-1}(\mathbf{x}_1) \\ f_R^{-1}\left(C^{-1}\left(\mathbf{x}_2 \,\middle|\, f_R^{-1}(\mathbf{x}_1)\right)\right) \end{bmatrix}, & \text{otherwise} \end{cases} \tag{7}$$

This procedure leads to the dense lower triangular Jacobian visualized in figure 1 *(right)*, the log-determinant of which is simply the sum of the log-determinants of all sub-couplings C. If N is a power of two and hence the dimensions of all splits work out, f_R becomes a full Knothe-Rosenblatt map.

Having defined the recursive coupling block, we find that the way we integrate it with the permutation matrix $\mathbf{Q}$ has a profound effect on the type of transport T we get when compositing many blocks.

If we omit the permutations entirely, the composition of lower triangular maps gives us another dense lower triangular map (figure 2, *left*) and thus, in fact, an autoregressive model.

Placing a permutation matrix at the beginning of the lower branch in each recursive split (figure 2, *middle*) interestingly produces a conditional structure: The sub-transport performed by the lower branch at each level is informed by the upper branch, but has no influence on the latter itself. Because variables never get permuted between the outermost branches, we end up with a lower "lane" that performs a transport conditioned on intermediate representations from the upper lane. The Jacobian of T in this case exhibits a block triangular structure.

Finally, if we simply apply a permutation over all variables outside of each recursive block (figure 2, *right*), we end up with a transport T with full Jacobian and, presumably, the greatest expressive power among the options considered here.

4.2. Hierarchical Invertible Neural Transport

While the recursive coupling block defined above is motivated by the search for a more expressive architecture, we have seen that its structure also lends itself well to a hierarchical treatment where splits of the variables at different levels carry different meaning.

Specifically, in a setting with paired data $(\mathbf{x}_i, \mathbf{y}_i)$, we can provide both variables as input to the flow and use the top level split to transform them separately at the next recursion level. To this end we take inspiration from figure 2 *(middle)* to design a hierarchical affine coupling block. Instead of relying on the recursive permutation structure implied there, however, we only apply a single permutation $\mathbf{Q}_\mathbf{y}$ and $\mathbf{Q}_\mathbf{x}$ to each respective lane at the beginning of the block. This preserves the hierarchical setup while letting variables within each lane interact more freely and saving on expensive tensor operations.

Figure 3 shows a hierarchical affine coupling block of this

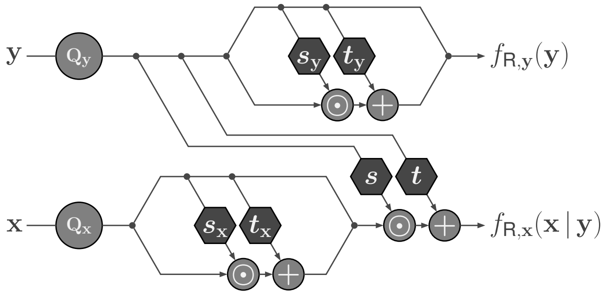


Figure 3. Hierarchical affine coupling block with one level of recursion. Note how the transformation in the $\mathbf{x}$-lane is influenced by $\mathbf{y}$, but not vice-versa, imposing a hierarchy on variables.

type, with one level of recursion for simplicity. A normalizing flow model constructed in this way performs *hierarchical invertible neural transport*, or HINT for short.

The output of a HINT model is a latent code with two components, $\mathbf{z} = [\mathbf{z_y}, \mathbf{z_x}]^\top = T(\mathbf{y}, \mathbf{x})$, but the training objective stays the same as in equation (4):

$$\mathcal{L}(\mathbf{y}, \mathbf{x}) = \tfrac{1}{2}\|T(\mathbf{y}, \mathbf{x})\|_2^2 - \log|\mathbf{J}_T(\mathbf{y}, \mathbf{x})| \tag{8}$$

As with a standard normalizing flow, the joint density of input variables is the pull-back of the latent density via T:

$$p_T(\mathbf{y}, \mathbf{x}) = T_\#^{-1} p_Z(\mathbf{z}) = T_\#^{-1}\mathcal{N}(\mathbf{0}, \mathbf{I}_{|\mathbf{y}|+|\mathbf{x}|}). \tag{9}$$

But because the $\mathbf{y}$-lane in HINT can be evaluated independently of the $\mathbf{x}$-lane, we can determine the partial latent code $\mathbf{z_y}$ for a given $\mathbf{y}$ and hold it fixed, while drawing $\mathbf{z_x}$ from the $\mathbf{x}$-part of the latent distribution. Doing so yields samples from the conditional density

$$p_T(\mathbf{x} \,|\, \mathbf{y}) = T_\#^{-1} \begin{bmatrix} T_\mathbf{y}(\mathbf{y}) \\ \mathcal{N}(\mathbf{0}, \mathbf{I}_{|\mathbf{x}|}) \end{bmatrix}. \tag{10}$$

This means HINT gives access to both the joint and the conditional density of the two variables $\mathbf{x}$ and $\mathbf{y}$.

5. Experiments

All experiments were carried out on an artificial data set of 2d shapes that allows visualizing samples regardless of the chosen data dimensionality. All flow models were trained on a single NVIDIA GeForce RTX 2080 Ti.

5.1. Fourier shapes data set

A curve $\mathbf{f}(t) \in \mathbb{R}^2$ parameterized by $2M+1$ complex 2d Fourier coefficients $\mathbf{a}_m \in \mathbb{C}^2$ can be traced as

$$\mathbf{f}(t) = \sum_{m=-M}^{M} \mathbf{a}_m \cdot e^{2\pi \cdot i \cdot m \cdot t} \tag{11}$$

with parameter t running from 0 to 1. This parameterization will always yield a closed, possibly self-intersecting curve (McGarva & Mullineux, 1993).

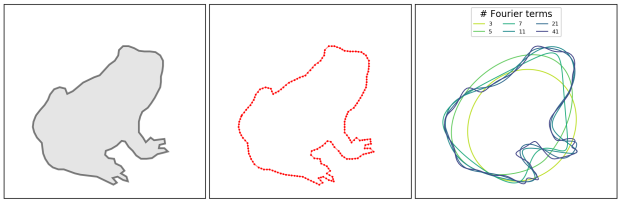


Figure 4. Left: A 2d polygon obtained from the segmentation of a natural image. *Middle:* The vertices $\mathbf{p}_n$ of the polygon, which are the basis for computing the Fourier coefficients in equation (12). *Right:* Tracing the parameterized 2d curve $\mathbf{f}(t)$ according to equation (11) for different numbers M of Fourier terms $\mathbf{a}_m$.

Vice-versa, given a sequence of N 2d points $\mathbf{p}_n \in \mathbb{R}^2$ we can calculate the Fourier coefficients of a curve approximating this sequence as

$$\mathbf{a}_m = \frac{1}{N} \sum_{n=0}^{N-1} \mathbf{p}_n \cdot e^{-2\pi \cdot i \cdot m \cdot n / N} \text{ for } m \in [-M, M] \tag{12}$$

When we increase the limit M, higher order terms are added to the parameterization in equation (11) and the shape can be approximated in greater detail. An example of this effect for a natural shape is shown in figure 4 *(right)*. Note that the actual dimensionality of the parameterization in our data set is $|\mathbf{x}| = 4 \cdot (2M+1)$, as each complex 2d coefficient $\mathbf{a}_m$ needs four real numbers to represent it.

In this paper we perform experiments on two specific Fourier shapes data sets. The first one uses $M = 2$, i.e. $|\mathbf{x}| = 20$, and we choose the prior to be an arbitrary Gaussian mixture model with five components. Figure 5 *(left)* shows a large sample from this shape prior and four single representatives. Note that these shapes carry little geometric meaning, often self-intersect and are best understood as a way to visualize samples $\mathbf{x} \in \mathbb{R}^{20}$ in two dimensions.

The second one uses $M = 12$, i.e. $|\mathbf{x}| = 100$, to represent a distribution of simple shapes that are generated geometrically by crossing two bars of randomized length and width at a right angle. This results in a variation of X, L and T shapes, some of which are shown in figure 5 *(right)*.

In both instances, our training set has 10^6 samples and the test set has 10^5 samples.

5.2. Forward process

Our forward operator in the Bayesian inference task returns three simple features of the 2d shape parameterized by $\mathbf{x}$. We determine the largest diameter as d_{max} and the total width of the shape orthogonal to this direction as d_{min}. The

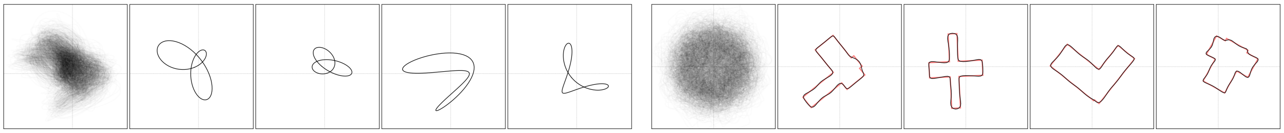


Figure 5. Left: Large batch and individual samples drawn from the prior of the Gaussian mixture data set. *Right:* The same for the geometrical shapes data set. Red lines behind the individual Fourier curves show the original shapes they approximate.

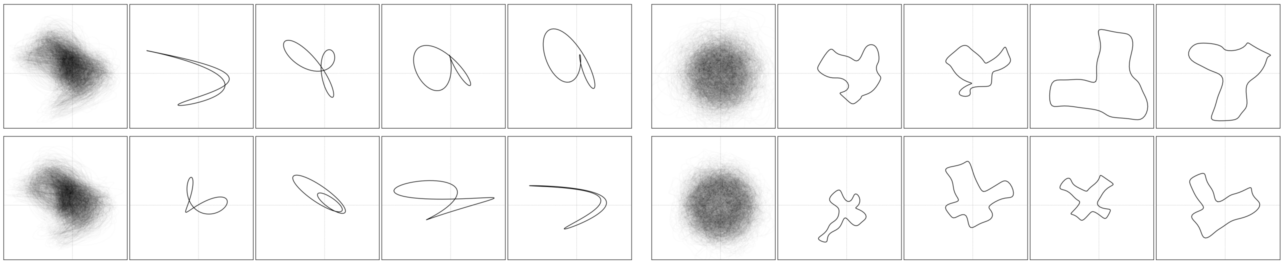


Figure 6. Samples from standard coupling block networks *(top row)* and recursive coupling block networks *(bottom row)*, trained on the two data sets presented in figure 5. Each network has four blocks and either 10^5 *(left)* or $2 \cdot 10^6$ *(right)* parameters. For the geometrical shapes it is clearly visible that recursive blocks better capture the details of the data distribution.

Table 1. Comparison of normal and recursive coupling for sampling and density estimation on Gaussian mixture prior. MMD (Gretton et al., 2012) compares the model's sample distribution to test data, bits/dim is the log-likelihood of test data under the model. Mean and standard deviation from five independent training runs.

Block type	#	MMD ↓	Bits/Dim ↑
Normal	1	0.0936 ± 0.0038	0.8380 ± 0.0582
Normal	2	0.0034 ± 0.0010	1.5106 ± 0.0062
Recursive	1	0.0345 ± 0.0004	1.4195 ± 0.0007
Recursive	2	0.0016 ± 0.0001	1.5288 ± 0.0012

angle α of the former is one feature, the *aspect ratio* $a = \frac{d_{\text{min}}}{d_{\text{max}}}$ another. The third feature is *circularity*, defined as $c = \frac{4\pi \cdot \text{area}}{\text{perimeter}^2}$.

For ambiguity we add some noise $\sigma \sim \mathcal{N}(\mathbf{0}, \frac{1}{20}\mathbf{I})$ to these three values to obtain our vector $\mathbf{y}$.

5.3. Density estimation

On the Gaussian mixture data set, we trained a single-block and a two-block network for density estimation, once with standard coupling blocks and once with our recursive design. Each internal sub-network s and t consist of three fully connected layers. Sub-networks further down in the recursion use progressively fewer parameters, and all four normalizing flows are scaled to have a total of 10^5 trainable parameters.

For these and all following experiments, we trained each network for 50 epochs with Adam and a learning rate decaying exponentially from 10^{-2} to 10^{-4}. With a small amount of weight regularization and clamping gradients to $[-5, 5]$, we observed very stable training for all networks.

Qualitative samples from these models are shown in figure 6 *(left)*, with the left-most panel being an empirical estimate of the prior $p_X(\mathbf{x})$. Quantitative results can be found in table 1. We compare across several training runs per model in terms of maximum mean discrepancy (Gretton et al., 2012) and bits per dimension.

The former measures the dissimilarity of two distributions using only samples from both. Following Ardizzone et al. (2019a), we use MMD with an inverse multi-quadratic kernel and average the results from 100 batches from the data prior and from each trained model.

Bits per dimension are a representation of the log-likelihood of the test data under a model, calculated as

$$\text{bits/dim}_T(\mathbf{x}) = -\frac{\frac{1}{2}\|T(\mathbf{x})\|_2^2 - \log|\mathbf{J}_T(\mathbf{x})|}{|\mathbf{x}| \cdot \ln 2}.$$

For both metrics, we see that a single standard coupling block performs very badly, which is unsurprising since it leaves half the variables untouched. A single recursive coupling block on the other hand is reasonably successful and achieves bits/dimension almost on par with a two standard coupling blocks. Out of the tested arrangements, a recursive two-block network performs the best.

We also trained networks with standard and recursive coupling blocks on the larger geometrical shape data set, where we afforded each network a total of $2 \cdot 10^6$ parameters spread over four coupling blocks. Qualitative samples from these models are shown in figure 6 *(right)*, with those from the normal coupling network in the top row and those from the

Table 2. Comparison of normal (conditional) and hierarchical coupling for Bayesian inference on Gaussian mixture prior. Columns as in table 1. Mean and standard deviation over three training runs.

BLOCK TYPE	#	MMD ↓	BITS/DIM ↑
NORMAL	1	0.3718 ± 0.0011	0.9663 ± 0.0073
NORMAL	2	0.1263 ± 0.0332	1.7072 ± 0.0276
NORMAL	4	0.0273 ± 0.0038	1.7868 ± 0.0094
NORMAL	8	0.0158 ± 0.0018	1.7939 ± 0.0070
HINT	1	0.0358 ± 0.0009	1.6465 ± 0.0004
HINT	2	0.2028 ± 0.0460	1.6695 ± 0.0012
HINT	4	0.0588 ± 0.0542	1.7553 ± 0.0162
HINT	8	0.0241 ± 0.0003	1.7379 ± 0.0129

recursive network in the bottom row.

While it is difficult to judge how well the entire distribution of shapes matches the prior (see figure 5, *(right)*), it is very clear that the network based on recursive coupling blocks produces individual samples that are more faithful to the data set.

5.4. Bayesian inference

For the Bayesian inference task on the Gaussian mixture data set, we trained a range of conditional flow models and HINT models with the following properties:

- with 1 block and $2 \cdot 10^5$ trainable parameters
- with 2 blocks and $2 \cdot 10^5$ trainable parameters
- with 4 blocks and $4 \cdot 10^5$ trainable parameters
- with 8 blocks and $4 \cdot 10^5$ trainable parameters

We train multiple instances of each model as described above and again evaluate in terms of MMD and bits per dimension.

In this case however the MMD comparison is not with samples from the prior $p_X(\mathbf{x})$, but with samples from an estimate of the posterior $p(\mathbf{x} \,|\, \mathbf{y}))$ for 10^3 values of $\mathbf{y}$ obtained via the prior and forward process. We create these estimated ground truth posterior samples via quantitative approximate Bayesian computation as explained in Ardizzone et al. (2019a).

The bits per dimension measure also requires extra attention in this setting, since we have to compute the log-likelihood of only the $\mathbf{x}$-lane of our HINT models. Fortunately, this is easily done by excluding the contributions from the $\mathbf{y}$-lane when adding up the log-determinants of the Jacobians in each sub-coupling.

The quantitative results of these experiments are summarized in table 2, where we can see that the standard coupling model with eight blocks achieves the best score. As in the unconditional case, however, the recursive design vastly outperforms the standard one when only a single block is compared.

Qualitative results are presented in figure 7. Each individual sample has a red bounding box that displays its true aspect ratio and angle of greatest diameter, and a green box displaying the conditioning target $\mathbf{y}$. The jagged rings around the samples are chosen to have the exact circularity measured from the sample *(red)* or required by $\mathbf{y}$ *(green)*, respectively. Visually, we find both approaches to be on equal footing.

In figure 8, we also show qualitative results from models with four and eight blocks and $4 \cdot 10^6$ parameters each trained on the geometrical shape data set. Adherence to the condition is visualized just like in figure 7.

As in the unconditional case, it is clear that the Fourier coefficients produced by the HINT models parameterize shapes that are much closer to those in the data set. Both conditional coupling block models have trouble recovering the rectangular nature of the shapes. While all four models stay true to the circularity they are conditioned on, they all struggle to place the greatest diameter of the shapes at the required angle. The HINT models appear slightly more faithful to the aspect ratio even when the rotation is off.

5.5. Proof of Concept: Automatic Posterior Transformation using HINT

In the following, we use HINT to implement the Automatic Posterior Transformation (APT) algorithm proposed by Greenberg et al. (2019), and compare the implementation to a standard coupling block INN. In short, given some observations, APT estimates the posterior of unobserved variables. The posterior is then used as a prior for the next step, and subsequently updated by newly arriving observations. This can be repeated many times. To prevent errors from accumulating over many timesteps, the density models have to be very accurate, making this an interesting application for HINT.

As a task, we use the general predator-prey model, also termed *competitive Lotka-Volterra equations*, which typically describes the interaction of d species in a biological system over time:

$$\frac{\partial}{\partial t} x_i = \beta_i x_i \left(1 - \sum_{j=1}^{d} \alpha_{ij} x_j \right) \tag{13}$$

The undisturbed growth rate of species i is given by β_i (can be $<$ or $>$ 0, growing or shrinking naturally), and is further affected by the other species in a positive way ($\alpha_{ij} > 0$, predator), or in a negative way ($\alpha_{ij} < 0$, prey). The solutions to this system of equations can not be expressed

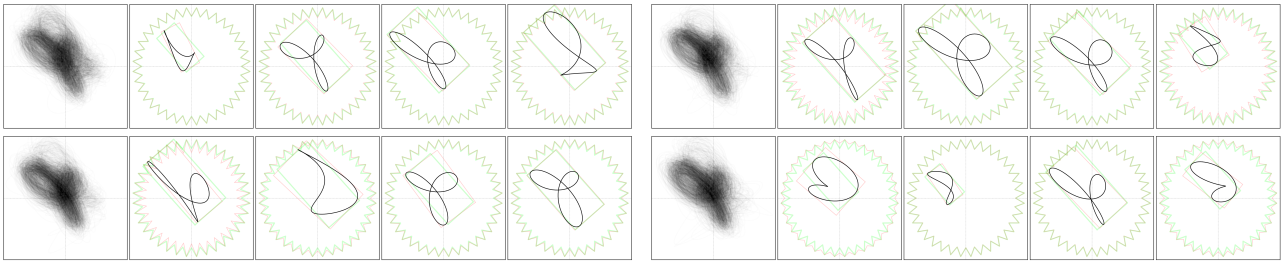


Figure 7. Samples from conditional flow *(left)* and HINT *(right)* with 4 *(top)* and 8 blocks *(bottom)* trained on Gaussian mixture data.

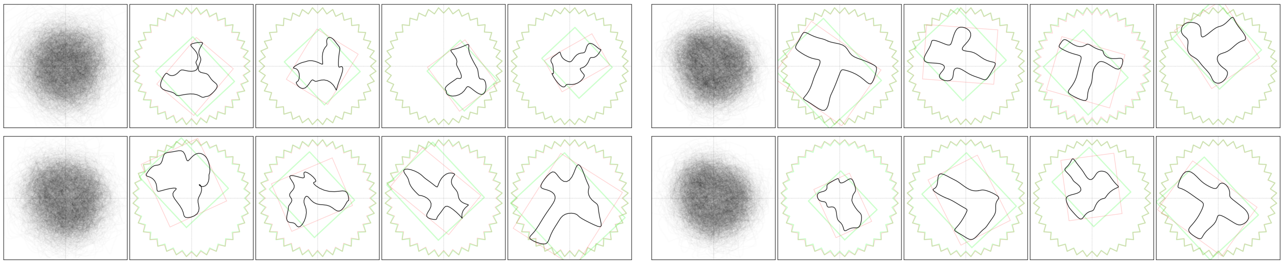


Figure 8. Samples from conditional flow *(left)* and HINT *(right)* with 4 *(top)* and 8 blocks *(bottom)* trained on geometrical shape data.

analytically, which makes their prediction a challenging task. Additionally, we make the process stochastic, by adding random noise at each time step. Therefore, at each time step, noisy measurements of x_1, x_2, x_3 are observed, with the task to predict the remaining population x_4, given the current and past observations. The exact parameters of the data model are found in appendix section C.

We compare this to an implementation using a standard coupling block INN with the training procedure proposed by Ardizzone et al. (2019a). The results for a single example time-series are shown in figure 9. We find that the standard INN does not make meaningful predictions in the short- or mid-term, only correctly predicting the final dying off of population x_4. HINT on the other hand is able to correctly model the entire sequence. Importantly, the modeled posterior distribution at each step correctly contracts over time, as more measurements are accumulated. Experimental details are found in appendix section C.

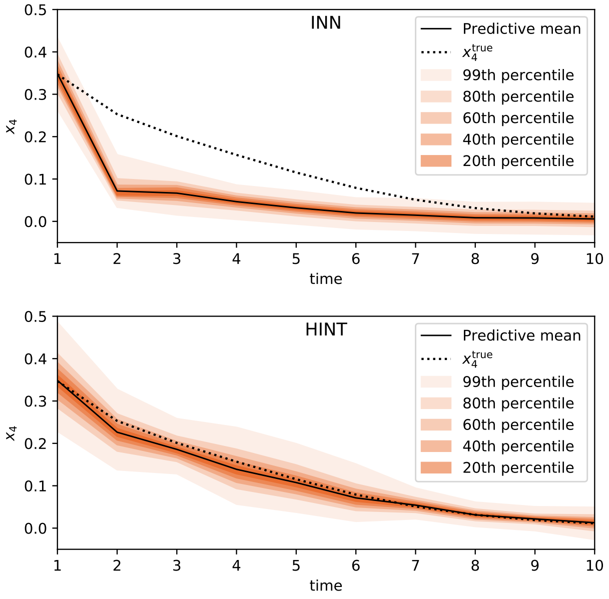


Figure 9. Contracting posterior for sequential Lotka-Volterra.

6. Conclusion

In this work, we presented HINT, a new architecture with the primary use as a normalizing flow. It improves on the traditional coupling block design in terms of expressive power by making the Jacobian densly triangular. Hereby, HINT keeps the advantages of interpretable latent space and equally efficient sampling and density estimation, which are typically not present in other models with densely triangular Jacobians, specifically autoregressive flows. To evaluate the model, we present a new data set based on Fourier decompositions of 2D shapes, which can be easily visualized while still posing a reasonable level of challenge to compare different methods. In terms of future improvements, we expect that our formulation can be made even more computationally efficient through the use of masking operations enabling more advanced parallelization.

Acknowledgments

J. Kruse was supported by Informatics for Life funded by the Klaus Tschira Foundation. G. Detommaso was supported by the EPSRC Centre for Doctoral Training in Statistical Applied Mathematics at Bath (EP/L015684/1).

References

Ardizzone, L., Kruse, J., Rother, C., and Köthe, U. Analyzing inverse problems with invertible neural networks. In *Intl. Conf. on Learning Representations*, 2019a.

Ardizzone, L., Lüth, C., Kruse, J., Rother, C., and Köthe, U. Guided image generation with conditional invertible neural networks. *arXiv:1907.02392*, 2019b. URL http://arxiv.org/abs/1907.02392.

Behrmann, J., Grathwohl, W., Chen, R. T. Q., Duvenaud, D., and Jacobsen, J.-H. Invertible residual networks. In *International Conference on Machine Learning*, 2019. URL http://proceedings.mlr.press/v97/behrmann19a.html.

Chen, T. Q., Behrmann, J., Duvenaud, D. K., and Jacobsen, J.-H. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems 32*, pp. 9913–9923, 2019.

Dinh, L., Krueger, D., and Bengio, Y. NICE: non-linear independent components estimation. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1410.8516.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=HkpbnH9lx.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *Advances in Neural Information Processing Systems*, pp. 7509–7520, 2019.

Gomez, A. N., Ren, M., Urtasun, R., and Grosse, R. B. The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems*, pp. 2214–2224, 2017.

Grathwohl, W., Chen, R. T. Q., Bettencourt, J., and Duvenaud, D. Scalable reversible generative models with free-form continuous dynamics. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJxgknCcK7.

Greenberg, D. S., Nonnenmacher, M., and Macke, J. H. Automatic posterior transformation for likelihood-free inference. *arXiv:1905.07488*, 2019. URL http://arxiv.org/abs/1905.07488.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087, 2018.

Hyvärinen, A. and Pajunen, P. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, 12(3):429–439, 1999.

Jaini, P., Selby, K. A., and Yu, Y. Sum-of-squares polynomial flow. In *International Conference on Machine Learning*, pp. 3009–3018, 2019.

Karami, M., Schuurmans, D., Sohl-Dickstein, J., Dinh, L., and Duckworth, D. Invertible convolutional flow. In *Advances in Neural Information Processing Systems*, pp. 5636–5646, 2019.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *arXiv:1807.03039*, 2018.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Kobyzev, I., Prince, S., and Brubaker, M. A. Normalizing flows: An introduction and review of current methods. *arXiv:1908.09257*, 2019.

Kruse, J., Ardizzone, L., Rother, C., and Köthe, U. Benchmarking invertible architectures on inverse problems. *Workshop on Invertible Neural Networks and Normalizing Flows, International Conference on Machine Learning*, 2019.

Liao, H., He, J., and Shu, K. Generative model with dynamic linear flow. *IEEE Access*, 7:150175–150183, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2947567.

Ma, X., Kong, X., Zhang, S., and Hovy, E. Macow: Masked convolutional generative flow. *Advances in Neural Information Processing Systems 32*, pp. 5891–5900, 2019. URL http://papers.nips.cc/paper/8824-macow-masked-convolutional-generative-flow.pdf.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, pp. 1–41, 2016.

McGarva, J. and Mullineux, G. Harmonic representation of closed curves. *Applied Mathematical Modelling*, 17(4):213 – 218, 1993. ISSN 0307-904X. URL http://www.sciencedirect.com/science/article/pii/0307904X9390109T.

Papamakarios, G., Pavlakou, T., and Murray, I. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pp. 2338–2347, 2017.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *arXiv:1912.02762*, 2019. URL http://arxiv.org/abs/1912.02762.

Rezende, D. and Mohamed, S. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pp. 1530–1538, 2015.

Song, Y., Meng, C., and Ermon, S. Mintnet: Building invertible neural networks with masked convolutions. In *Advances in Neural Information Processing Systems*, pp. 11002–11012, 2019.

Stoer, J. and Bulirsch, R. *Introduction to numerical analysis*, volume 12. Springer Science & Business Media, 2013.

Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al. Conditional image generation with PixelCNN decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016a.

Van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pp. 1747–1756, 2016b.

Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. Learning likelihoods with conditional normalizing flows. *arXiv preprint arXiv:1912.00042*, 2019.

– Appendix –

A. Details on Gaussian mixture data set

The prior for our smaller data set instance is a Gaussian mixture model with five components in 20 dimensions. The parameters μ and σ for this distribution can be generated with the following Python snippet:

```python
rng = np.random.RandomState(seed=123)
coeffs_shape = (2,5,2)
n_components = 5
component_weights = (.5 + rng.rand(n_components))
component_weights /= np.sum(component_weights)
mus = [.5 * rng.randn(*coeffs_shape)
    for i in range(n_components)]
sigmas = [.1 + .2 * rng.rand(*coeffs_shape)
    for i in range(n_components)]
```

B. Details on geometrical shape data set

The shapes for the larger data set instance are generated by taking the union of two oblong rectangles crossing each other at a right angle. For both rectangles, the longer side length is drawn uniformly from $[\,3,5\,]$ and the other from $[\,\frac{1}{2},2\,]$. We shift both rectangles along their longer side by a uniformly random amount from $[\,-\frac{3}{2},\frac{3}{2}\,]$.

Then we form the union and insert equally spaced points along the resulting polygon's sides such that no line segment is longer than $\frac{1}{5}$. This is necessary to obtain point sequences which are approximated more faithfully by Fourier curves.

Finally, we center the shape at the origin, rotate it by a random angle and shift it in either direction by a distance drawn from $\mathcal{N}(0,\frac{1}{2})$.

Python code for creating both data set variants can be found under https://github.com/VLL-HD/HINT.

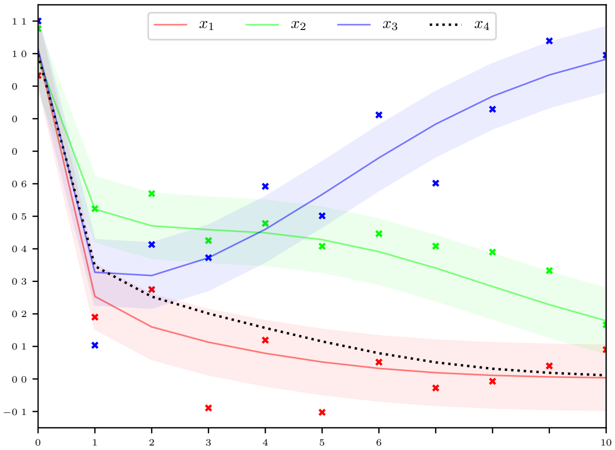


Figure 10. Population development in competitive Lotka-Volterra model. x_1, x_2 and x_3 are only acessible via noisy measurements *(small crosses)*, the colored bands show the standard deviation of this noise. The task is to predict x_4 as the sequence developes.

C. Details on extra experiment in 5. 5.

Parameters $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ for the 4d competitive Lotka-Volterra model in equation (13) were picked randomly as

$$\boldsymbol{\beta} = \begin{bmatrix} 1.14143055 \\ 0.64270729 \\ 1.42981209 \\ 0.90620443 \end{bmatrix} \text{ and}$$

$$\boldsymbol{\alpha} = \begin{bmatrix} 0.78382338 & 1.26614888 & 1.25787652 & 0.80904295 \\ 1.00470891 & 0.32719451 & 1.34501072 & 1.29758381 \\ 1.28599724 & 0.39362355 & 0.89977679 & 1.00063551 \\ 1.12163602 & 1.08672758 & 1.39634746 & 0.53592833 \end{bmatrix},$$

and the true initial population values $\mathbf{x}_0^*$ set to

$$\mathbf{x}_0^* = \begin{bmatrix} 1.00471435 \\ 0.98809024 \\ 1.01432707 \\ 0.99687348 \end{bmatrix}.$$

Figure 10 shows the resulting development of the four populations over ten unit time steps, including noise on the observed values for x_1, x_2 and x_3.

Both the INN and the HINT network we trained consist of ten (non-recursive) coupling blocks with a total of 10^6 trainable weights. We used Adam to train both for 50 epochs per time step with 64000 training samples and a batch size of 500. The learning rate started at 10^{-2} and exponentially decayed to 10^{-3} (HINT) and 10^{-4} (INN), respectively. Inference on x_4 begins with an initial guess at $t = 0$ drawn from $\mathcal{N}(1,\frac{1}{10})$.