Statistics and Computing (2023) 33:90
https://doi.org/10.1007/s11222-023-10262-y

ORIGINAL PAPER

A data-driven and model-based accelerated Hamiltonian Monte Carlo method for Bayesian elliptic inverse problems

Sijing Li[1] · Cheng Zhang[2] · Zhiwen Zhang[1] · Hongkai Zhao[3]

Received: 1 April 2022 / Accepted: 27 May 2023

Abstract
In this paper, we consider a Bayesian inverse problem modeled by elliptic partial differential equations (PDEs). Specifically, we propose a data-driven and model-based approach to accelerate the Hamiltonian Monte Carlo (HMC) method in solving large-scale Bayesian inverse problems. The key idea is to exploit (model-based) and construct (data-based) intrinsic approximate low-dimensional structure of the underlying problem which consists of two components—a training component that computes a set of data-driven basis to achieve significant dimension reduction in the solution space, and a fast solving component that computes the solution and its derivatives for a newly sampled elliptic PDE with the constructed data-driven basis. Hence we develop an effective data and model-based approach for the Bayesian inverse problem and overcome the typical computational bottleneck of HMC—repeated evaluation of the Hamiltonian involving the solution (and its derivatives) modeled by a complex system, a multiscale elliptic PDE in our case. Finally, we present numerical examples to demonstrate the accuracy and efficiency of the proposed method.

Keywords Elliptic inverse problems · Bayesian inversion · Hamiltonian Monte Carlo (HMC) method · Proper orthogonal decomposition (POD) · Model reduction

Mathematics subject classification 35R60 · 60J22 · 65N21 · 65N30 · 78M34

1 Introduction

Inverse problems are ubiquitous in models used in science and engineering where problem-specific parameters or inputs need to be estimated from indirect and noisy observations. However, inverse problems are often nonlinear (even if the forward problems are linear) and ill-posed (or unstable) in that either the existence and uniqueness of the solutions may not be guaranteed or the dependence of the parameters on the data (and noise) may be sensitive. As a result, pointwise estimates may be erroneous and misleading and additional regularization is often required. On the other hand, the Bayesian approach to inverse problems (Kaipio and Somersalo 2005; Mondal et al. 2010; Dashti and Stuart 2011; Martin et al. 2012; Beskos et al. 2015; Lan 2019) can provide another alternative. In the Bayesian paradigm, the solution to the inverse problem is posited as the posterior distribution of the unknowns conditioned on observations, where regularization is naturally imposed in the form of an appropriate prior distribution. Bayesian inversion, therefore, provides a principled way of uncertainty quantification in the presence of data and noise. We refer interested readers to Stuart (2010) for a comprehensive review of the Bayesian approach to inverse problems.

As the posterior is generally intractable due to the complexity of the system, people often resort to computational approximation approaches such as Markov chain Monte

✉ Cheng Zhang
chengzhang@math.pku.edu.cn

✉ Zhiwen Zhang
zhangzw@hku.hk

Sijing Li
lsj17@hku.hk

Hongkai Zhao
zhao@math.duke.edu

[1] Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China

[2] School of Mathematical Sciences and Center for Statistical Science, Peking University, Beijing, China

[3] Department of Mathematics, Duke University, Durham, NC 27708, USA

Published online: 16 June 2023

Carlo (MCMC) methods. In a typical MCMC method, samples from the posterior distribution are generated by updating the current states according to a proposing mechanism and a correction criterion designed to keep the posterior invariant. The efficiency of MCMC methods heavily depends on the design of proposing mechanism, i.e. its computation cost, acceptance probability, and mixing property. For complicated and large systems in practice, it is important to strike an appropriate balance among these factors. For example, simple MCMC algorithms (e.g., generating proposals based on random walk Metropolis), although easy and cheap to implement, usually have a low acceptance rate and mix poorly for complex and high dimensional problems since no information or structure of the underlying problem is utilized.

In recent years, many advanced MCMC methods have been proposed to improve the sampling efficiency for high-dimensional problems (Duane et al. 1987; Neal 2011). Based on an intelligent design of Hamiltonian dynamics, the Hamiltonian Monte Carlo (HMC) method uses gradient information of the underlying posterior distribution to make distant and less correlated proposals with high acceptance probabilities, greatly improving the mixing rate of the Markov chains. On the other hand, the associated computation cost of those advanced MCMC methods can be a bottleneck that makes it difficult to scale up to complicated models and large data. Note that the sampling procedure requires repetitive evaluations of the likelihood function and its derivatives and maybe other geometric and statistical quantities, e.g., Fisher information for Riemannian Hamiltonian Monte Carlo (RHMC) method (Girolami and Calderhead 2011).

To alleviate this issue, one popular attempt is to find a computationally cheap surrogate approximation to replace the original Hamiltonian (Zhang et al. 2017a, b, 2018; Strathmann et al. 2015; Lan et al. 2016) in the sampling process. The key in designing an effective surrogate function is to capture the collective property of large datasets while removing redundancy. The overall computation efficiency is improved due to significant cost reduction in the Hamiltonian proposing process and insignificant loss in acceptance rate. Although these surrogate approximations can provide significant empirical performance improvement, they are usually obtained as blackbox approximations by fitting the training data where the mechanism and structure of the model that generates the data have largely remained unexplored and unexploited.

For our Bayesian inverse problem, not only the unknown quantity is a random field that lives in a high-dimensional space after discretization or can be a parametric model with many parameters, the solution to an elliptic PDE and its derivatives are also involved in generating the data and evaluating the posterior and Hamiltonian in the HMC method. These computational challenges make traditional MCMC methods extremely costly to use for Bayesian inverse problems.

In this work, we propose a data-driven and model-based approach that can significantly reduce the computation cost of the HMC method for Bayesian elliptic inverse problems. The key idea is to exploit the intrinsic approximate low-dimensional structure of elliptic differential operators and construct a data-driven basis as proposed in Li et al. (2020). First, a set of data-driven basis functions are constructed from training data, e.g., from real measurements or the initial burn-in stage of MCMC methods, to achieve significant dimension reduction in the solution space. With the constructed basis, a newly sampled elliptic PDE can be solved efficiently. Note that the derivatives (with respect to some parameters) of a solution to a linear PDE satisfies the same PDE (with different righthand sides) that can be computed efficiently as well. Hence, this model-based and data-driven strategy can reduce the computation cost of the HMC sampling for our Bayesian inverse problem significantly.

The rest of the paper is organized as follows. We first describe the forward model and the Bayesian inversion problem in Sect. 2 and the HMC method for Bayesian inversion in Sect. 3. Intrinsic low-dimensional structure of the forward problem, model-based and data-driven dimension reduction, and approximation of the parameter-to-solution map are discussed in Sect. 4. The accelerated HMC method for Bayesian inverse problems is presented with implementation details in Sect. 5. We also discuss alternative approaches for the gradient computation in Sect. 6. We present numerical experiments and results of the accelerated HMC method and compare its performance to other state-of-the-art HMC methods in Sect. 7. Concluding remarks are made in Sect. 8.

2 Model problem

2.1 Forward problem

In this paper, we consider a classical inverse problem that involves inference of the diffusion coefficient in an elliptic PDE that is commonly used to model isothermal steady flow in porous media, hydrology and reservoir simulation, and many other applications. To be specific, we consider the following elliptic PDEs with random coefficients $a(\mathbf{x}, \omega)$, where one would like to infer, as the forward model,

$$\begin{aligned} &\mathcal{L}(\mathbf{x}, \omega)u(\mathbf{x}, \omega) \equiv -\nabla \cdot \big(a(\mathbf{x}, \omega)\nabla u(\mathbf{x}, \omega)\big) \\ &= f(\mathbf{x}), \quad \mathbf{x} \in D, \omega \in \Omega, \end{aligned} \tag{1}$$

$$u(\mathbf{x}, \omega) = 0, \qquad \mathbf{x} \in \partial D, \tag{2}$$

where $D \in \mathbb{R}^d$ is a bounded spatial domain, Ω is a sample space, and the source function $f(\mathbf{x}) \in L^2(D)$. We assume

$a(\mathbf{x}, \omega)$ in (1) is almost surely uniformly elliptic, namely, there exist $a_{\min}, a_{\max} > 0$, such that

$$P\big(\omega \in \Omega : a(\mathbf{x}, \omega) \in [a_{\min}, a_{\max}], \forall \mathbf{x} \in D\big) = 1. \tag{3}$$

In general, we can assume the random coefficient $a(\mathbf{x}, \omega)$ is of some parametric form. For example, a commonly used affine form is the following,

$$a(\mathbf{x}, \omega) = \bar{a}(\mathbf{x}) + \sum_{m=1}^{r} a_m(\mathbf{x})\xi_m(\omega), \tag{4}$$

where $\xi_m(\omega)$, $m = 1, ..., r$ are random variables and $a_m(\mathbf{x})$ are some spatial basis functions, e.g., finite element basis, polynomial basis, Fourier basis, radial basis, etc. Usually, one assumes $a(\mathbf{x}, \omega)$ is a random field and obtain the affine form (4) by computing the Karhunen Loève approximation of the random field $a(\mathbf{x}, \omega)$. In this setting, the basis $a_m(x), m = 1, ..., r$ are the eigenfunctions of the covariance kernel of the random field $a(\mathbf{x}, \omega)$ and the number of bases r is determined such that the error $||a(\mathbf{x}, \omega) - \bar{a}(\mathbf{x}) - \sum_{m=1}^{r} a_m(\mathbf{x})\xi_m(\omega)||_2$ is less than some threshold in the mean square error. The dependence of r on the threshold reveals the intrinsic complexity of the random field (see Bryson et al. (2019)).

Once a parametric form of the random coefficient $a(\mathbf{x}, \omega) = a(\mathbf{x}, \boldsymbol{\xi}(\omega))$ is given, computing the solution $u(\mathbf{x}, \omega)$ to the problem (1)-(2) defines a map from the parameter domain $\boldsymbol{\xi}(\omega) = \big(\xi_1(\omega), \cdots, \xi_r(\omega)\big)^T \in \mathcal{W} \subset \mathbb{R}^r$ to the solution space

$$\boldsymbol{\xi}(\omega) \mapsto u(\mathbf{x}, \omega) = u(\mathbf{x}, \boldsymbol{\xi}(\omega)) \in H_0^1(D), \tag{5}$$

which is a Banach-space-valued function of the random input vector $\boldsymbol{\xi}(\omega)$.

Many efficient numerical methods have been developed for solving elliptic PDEs with random coefficients; see e.g. Ghanem and Spanos (1991); Xiu and Karniadakis (2003); Asokan and Zabaras (2006); Babuska et al. (2004, 2007); Nobile et al. (2008); Graham et al. (2011); Abdulle et al. (2013); Graham et al. (2015) and references therein. By solving the forward problem, one can quantify the uncertainty in the elliptic PDEs with randomness. However, when the elliptic PDEs involve multiscale features and/or high-dimensional random inputs, these problems become challenging due to high computational costs. In recent years, we have developed data-driven methods to solve multiscale elliptic PDEs with random coefficients (1) based on intrinsic dimension reduction (Zhang et al. 2015; Chung et al. 2018; Li et al. 2020). We also refer the intertested reader to Wan and Zabaras (2013); Abdulle et al. (2013); Hou et al. (2016); Efendiev et al. (2015); Hou et al. (2019) and references therein for other methods to solve the forward problem (1).

2.2 Bayesian inverse problems

Let $\mathcal{W}$ be the space of admissible unknowns and $\mathcal{F} : \mathcal{W} \to \mathcal{U}$ be a forward map representing a mathematical model that assigns an output $u \in \mathcal{U}$ to an input $\boldsymbol{\xi} \in \mathcal{W}$. In this paper, we focus on the elliptic PDE (1), where $\boldsymbol{\xi}$ is the parameter in the random coefficient $a(\mathbf{x}, \boldsymbol{\xi})$ and u is the solution to the PDE with the corresponding coefficient. The inverse problem is to recover the unknown parameter $\boldsymbol{\xi} \in \mathcal{W}$ (and hence the coefficient $a(\mathbf{x}, \boldsymbol{\xi})$) from some measurements of the solution u in the domain and at the boundary. Often in practice, u can only be recorded at finite discrete locations with noise, which is the data denoted by $\mathbf{y} \in \mathbb{R}^m$ related by

$$\mathbf{y} = \mathcal{G}(\boldsymbol{\xi}) + \boldsymbol{\eta}. \tag{6}$$

Here the forward model $\mathcal{G} : \mathbb{R}^r \to \mathbb{R}^m$ is a composition of the forward map $\mathcal{F}$ and a discretized observation operator through which observable quantities (e.g., point-wise evaluation of the solution) are collected, and $\boldsymbol{\eta} \in \mathbb{R}^m$ is the measurement error (or the noise).

In the Bayesian formulation of the inverse problem (6), one treats the parameter $\boldsymbol{\xi}$ as a random variable (vector) with a prior distribution $p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$. The noisy model, i.e., distribution of $\boldsymbol{\eta}$, gives the likelihood $p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi})$. For simplicity and concreteness, in this paper we assume that $\boldsymbol{\eta}$ is a zero-mean Gaussian with diagonal covariance $\sigma^2 \boldsymbol{I}_m$, so that

$$p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) \propto \exp\left(-\Phi(\boldsymbol{\xi}; \mathbf{y})\right), \qquad \Phi(\boldsymbol{\xi}; \mathbf{y}) := \frac{\|\mathbf{y} - \mathcal{G}(\boldsymbol{\xi})\|^2}{2\sigma^2}. \tag{7}$$

The posterior distribution of $\boldsymbol{\xi}$ conditioned on the data $\mathbf{y}$ then follows the Bayes' rule:

$$p_{\boldsymbol{\xi}|\mathbf{y}}(\boldsymbol{\xi}|\mathbf{y}) \propto p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) \cdot p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \tag{8}$$

and Bayesian inversion can be performed by estimating the posterior via, e.g., the HMC method and other MCMC methods.

In addition to the usual computational issues for MCMC type of methods, there is another challenge for the Bayesian elliptic inverse problem due to the complicated forward model (1). Instead of a simple explicit probabilistic model that prescribes the likelihood of data given the parameter of interest, one needs to solve the elliptic PDE (1) for each random coefficient corresponding to a new sample of the parameter $\boldsymbol{\xi}$ to compute the likelihood function (7), which is the computation bottleneck for the Bayesian inversion. To address these challenges, we propose a data-driven and model-based accelerated HMC method that improves the convergence rate of the MCMC method and exploit the underlying forward model (1) using a data-driven approach

proposed in Li et al. (2020), which enables us to reduce the computational cost in solving the forward model problem and hence the overall sampling cost.

3 The HMC method for Bayesian inversion

The HMC method is one of the state-of-the-art MCMC methods suitable for complex high dimensional target distributions with strong dependencies between parameters, which is the case for Bayesian inverse problems. Leveraging geometric information from the target distribution, the HMC method (Duane et al. 1987; Neal 2011) extends the parameter space with auxiliary momentum variables $\boldsymbol{\zeta}$, and introduces a Hamiltonian dynamics system to propose samples of model parameters within the Metropolis framework, greatly enhancing the exploration efficiency in the parameter space compared to simple random walk proposals. More specifically, the HMC method generates proposals jointly for $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ using the following system of differential equations

$$\frac{d\boldsymbol{\xi}}{dt} = \frac{\partial H}{\partial \boldsymbol{\zeta}}, \qquad \frac{d\boldsymbol{\zeta}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\xi}}. \tag{9}$$

where the Hamiltonian function is defined as $H(\boldsymbol{\xi}, \boldsymbol{\zeta}) = U(\boldsymbol{\xi}) + K(\boldsymbol{\zeta})$.

Here in the Bayesian elliptic inverse problem, the potential energy U is defined as $U(\boldsymbol{\xi}) = -\log p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) - \log p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$, and the kinetic energy $K(\boldsymbol{\zeta}) = \frac{1}{2}\boldsymbol{\zeta}^T M^{-1}\boldsymbol{\zeta}$ corresponds to the negative log-density of a zero-mean multivariate Gaussian distribution with covariance M (also known as the mass matrix and is often set to be the identity). As the analytical solution of the Hamiltonian dynamics (9) is usually unavailable, proposals in the HMC method are often made by numerical simulation via the leap-frog scheme. Specifically, given the sample $(\boldsymbol{\xi}^{(t)}, \boldsymbol{\zeta}^{(t)})$ at time t, we generate the sample at time $t+1$ by the following scheme

$$\begin{aligned}
\boldsymbol{\zeta}^{(t+\frac{1}{2})} &= \boldsymbol{\zeta}^{(t)} - \frac{\Delta t}{2}\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi}^{(t)}),\\
\boldsymbol{\xi}^{(t+1)} &= \boldsymbol{\xi}^{(t)} + \Delta t \nabla_{\boldsymbol{\zeta}} K(\boldsymbol{\zeta}^{(t+\frac{1}{2})}),\\
\boldsymbol{\zeta}^{(t+1)} &= \boldsymbol{\zeta}^{(t+\frac{1}{2})} - \frac{\Delta t}{2}\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi}^{(t+1)}),
\end{aligned} \tag{10}$$

where Δt is the step size. Starting from the current state $(\boldsymbol{\xi}, \boldsymbol{\zeta})$, where $\boldsymbol{\xi}$ is the current parameter and $\boldsymbol{\zeta}$ is resampled from the multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, M)$, the proposed state $(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*)$ at the end of a simulated trajectory of length L is accepted with probability

$$\alpha_{\text{HMC}} = \min\Big(1, \exp[-H(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*) + H(\boldsymbol{\xi}, \boldsymbol{\zeta})]\Big). \tag{11}$$

From this point of view, the HMC method can be viewed as a Metropolis algorithm that samples from the joint distribution

$$p(\boldsymbol{\xi}, \boldsymbol{\zeta}) \propto \exp\left(-U(\boldsymbol{\xi}) - \frac{1}{2}\boldsymbol{\zeta}^T M^{-1}\boldsymbol{\zeta}\right). \tag{12}$$

The marginal distribution of $\boldsymbol{\xi}$ then follows the target posterior distribution since $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are separated (i.e., independent). Note that the Hamiltonian is preserved for analytical solutions of (9), and the discretization error in (10) can be controlled by appropriate choice of the step size Δt. Therefore, the HMC method is often able to generate distant, uncorrelated proposals with a high acceptance probability, allowing for efficient exploration of the parameter space.

For our Bayesian inverse problem, however, there is still a computational bottleneck we have to resolve, that is repetitive computation of solution $u(\mathbf{x}, \boldsymbol{\xi})$ to the elliptic PDE (1) in order to evaluate the potential energy $U(\boldsymbol{\xi}) = -\log p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) - \log p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$ in the Hamiltonian, and even more, the gradient with respect to the parameter $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ needs to be repetitively evaluated to simulate a trajectory for HMC proposals as in (10). Note that the key to evaluation of $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ is the evaluation of derivatives $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_j} = u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})$ of the solution to (1) for $j = 1, ..., r$, which satisfy

$$-\nabla\cdot\big(a(\mathbf{x}, \boldsymbol{\xi})\nabla u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})\big) = \nabla\cdot\big(a_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})\nabla u(\mathbf{x}, \boldsymbol{\xi})\big), \quad \mathbf{x}\in D, \tag{13}$$

$$u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi}) = 0, \qquad \mathbf{x}\in \partial D, \tag{14}$$

which is the same elliptic PDE as (1) with a right-hand side that depends on the solution to (1) corresponding to the current sample of $\boldsymbol{\xi}$. This could easily become prohibitively expensive in practice since so many PDEs have to be solved for each sampling step.

In what follows, we describe how to approximate the low-dimensional structure of the solution space to the elliptic PDE (1) with varying coefficients and right-hand sides and the data-driven approach proposed in Li et al. (2020) that can take advantage of the approximate low-dimensional structure of the forward model to accelerate the HMC method.

4 Low-dimensional structure of the forward problem and approximation of the parameter-to-solution map

For the Bayesian elliptic inverse problem, we are facing the challenge to solve the forward model problem, i.e., the elliptic PDE (1) with different coefficients and different right-hand sides (13) repetitively in the sampling process. This motivates us to exploit the low-dimensional structures in the solution space and develop data-driven and model-based dimension

reduction methods to solve the forward model problem efficiently.

4.1 Low-dimensional structure of the forward problem

We assume the coefficient $a(\mathbf{x}, \boldsymbol{\xi}(\omega))$ in the elliptic PDE (1) is almost surely uniform elliptic and smoothly depends on the parameter $\boldsymbol{\xi}$. Therefore, the solution $u(\mathbf{x}, \boldsymbol{\xi})$ also smoothly depends on the parameters and can be approximated via a polynomial expansion in $\boldsymbol{\xi}$ of the following form

$$\sum_{\boldsymbol{\alpha}\in\mathcal{J}_r} u_{\boldsymbol{\alpha}}(\mathbf{x})\boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega), \tag{15}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_r)$ is a multi-index, $\mathcal{J}_r = \{\boldsymbol{\alpha} \,|\, \alpha_i \geq 0, \alpha_i \in \mathbb{N}, 1 \leq i \leq r\}$ is a multi-index set of countable cardinality, and $\boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega) = \prod_{1\leq i\leq r} \xi_i^{\alpha_i}(\omega)$ is a multivariate polynomial.

Studying the approximation error of the expansion form (15) to the solution $u(\mathbf{x}, \boldsymbol{\xi})$ is an important problem. If $a(\mathbf{x}, \boldsymbol{\xi})$ satisfies the uniform ellipticity assumption and has a holomorphic extension to an open set in a complex domain that contains the real domain for $\boldsymbol{\xi}$, we can obtain explicit estimates for the coefficients $u_{\boldsymbol{\alpha}}$ similar to those estimates for the polynomial approximation for an analytic function. The following result for the best n-term approximation can be proved, where the details of the proof can be found in Cohen and DeVore (2015).

Proposition 4.1 *Consider a parametric elliptic PDE of the form (1)-(2) with a random coefficient (4). Both the Taylor series and Legendre series of the form (15) converge to $u(x, \boldsymbol{\xi}(\omega))$ in $H_0^1(D)$ for all $\boldsymbol{\xi}(\omega) \in \mathcal{W}$. Moreover, for any set $\mathfrak{J}_r^n$ of indices corresponding to the n largest of $||u_{\boldsymbol{\alpha}}(\cdot)||_{H_0^1(D)}$, we have*

$$\sup_{\boldsymbol{\xi}(\omega)\in\mathcal{W}} \Big|\Big|u(\cdot, \boldsymbol{\xi}(\omega)) - \sum_{\boldsymbol{\alpha}\in\mathfrak{J}_r^n} u_{\boldsymbol{\alpha}}(\cdot)\boldsymbol{\xi}^{\boldsymbol{\alpha}}(\omega)\Big|\Big|_{H_0^1(D)} \leq C\exp(-cn^{1/r}), \tag{16}$$

where $\mathcal{J}_r^n$ is a subset of $\mathcal{J}_r$ with cardinality $\#\mathcal{J}_r^n = n$, C and c are positive and depend on r.

Proposition 4.1 reveals the existence of low-dimensional structure in the solution space of the elliptic PDE (1). Specifically, given a threshold ϵ for the approximation error, there exist a linear subspace with the dimension at most $O(n \sim (\frac{\log C}{c} + \frac{|\log \epsilon|}{c})^r)$ (e.g., spanned by $u_{\boldsymbol{\alpha}}(x)$, $\boldsymbol{\alpha} \in \mathcal{J}_r^n$), which can approximate the solution of the elliptic PDE (1) within ϵ error.

The result in Proposition 4.1 provides a theoretical framework to study the approximation property of the parametric elliptic PDE. However, this approximation is obtained by mathematical analysis, which cannot be directly implemented via a computational algorithm. Moreover, using polynomial basis functions (15) (e.g., Taylor series or Legendre polynomials) may not be optimal to approximate the solution $u_{\boldsymbol{\alpha}}(x)$ in general, especially when the dimension of the random input is high. In Li et al. (2020), a data-driven approach was proposed to construct problem-dependent basis functions that can approximate the solution space of (1)-(2) effectively. We will adopt this approach in this paper to solve the Bayesian elliptic inverse problems.

When the coefficient $a(\mathbf{x}, \omega)$ is a nonlinear function of a finite number of random variables, one can apply the empirical interpolation method (EIM) (Barrault et al. 2004) to approximately convert $a(\mathbf{x}, \omega)$ into an affine form. Thus, low-dimensional structures still exist in the solution space. In addition, we refer the reader to Hoang and Schwab (2014); Bachmayr et al. (2017) for the results of the best n-term polynomial approximation of elliptic PDEs with lognormal coefficients.

4.2 Data-driven basis functions for dimension reduction

Since there exist low-dimensional structures in the solution space of elliptic PDEs with random coefficients, we use problem-specific and data-driven basis functions to achieve a significant dimension reduction in solving the elliptic PDEs (1) with random coefficients. Our method consists of a training process and a solving process. In the training process, we extract the low-dimensional structure of the solution space and construct a set of data-driven basis functions from training data or real measurements, e.g., a set of solution samples $\{u(\mathbf{x}, \omega_i)\}_{i=1}^N$, which can be obtained from measurements or generated by solving the elliptic PDE (1)-(2) with coefficient samples $\{a(\mathbf{x}, \omega_i)\}_{i=1}^N$ during the burn-in stage of the HMC method. Let $V_{snap} = \{u(\mathbf{x}, \omega_1), ..., u(\mathbf{x}, \omega_N)\}$ denote the solution samples that will be used for the construction of the data-driven basis functions.

To make the paper self-contained, we introduce how to use the proper orthogonal decomposition (POD) method (Berkooz et al. 1993; Sirovich 1987; Benner et al. 2015) to find the optimal subspace and its orthonormal basis functions to approximate the solution samples V_{snap} to a certain accuracy. Specifically, we define the correlation matrix $\Sigma = (\sigma_{ij}) \in \mathbb{R}^{N\times N}$ with $\sigma_{ij} =< u(\cdot, \omega_i), u(\cdot, \omega_j) >_D$, $i, j = 1, \ldots, N$, where $< \cdot, \cdot >_D$ denotes the standard inner product on $L^2(D)$. Let the eigenvalues of the correlation matrix be $\lambda_1 \geq \lambda_2 \geq \ldots \geq \ldots \geq \lambda_N \geq 0$ and the corresponding eigenvectors be $\phi_1(x), \phi_2(x), \ldots, \phi_N(x)$, which will be referred to as data-driven basis functions in this paper. In addition, we have the following estimate for the approximation error, where the proof can be found in Sect. 3.3.2 of Holmes et al. (1998) or page 502 of Benner et al. (2015).

Proposition 4.2 *The space spanned by the leading K data-driven basis functions has the following approximation property to V_{snap},*

$$\frac{\sum_{i=1}^{N}\left\|u(x,\omega_i)-\sum_{j=1}^{K}<u(\cdot,\omega_i),\phi_j(\cdot)>_D\phi_j(x)\right\|^2_{L^2(D)}}{\sum_{i=1}^{N}\left\|u(x,\omega_i)\right\|^2_{L^2(D)}} = \frac{\sum_{s=K+1}^{N}\lambda_s}{\sum_{s=1}^{N}\lambda_s}. \tag{17}$$

where the number K of the basis function will be determined according to the decay speed of the ratio $\rho=\frac{\sum_{s=K+1}^{N}\lambda_s}{\sum_{s=1}^{N}\lambda_s}$.

In practice, we choose the first K dominant eigenvalues such that the ratio ρ is small enough (e.g., $\rho = 0.1\%$) to achieve an expected accuracy. We expect a fast decay in the eigenvalues λ_s's so that a small set of data-driven basis functions ($K \ll N$) will be enough to approximate the solution samples well in the root mean square sense. We refer the interested reader to Schwab and Todor (2006) for some estimates on the rate of decay of the eigenvalues in the Karhunen Loève approximation of random fields, which are essentially the eigenvalues in POD method. In addition, since data-driven basis functions $\phi_j(\mathbf{x})$, $j = 1, ..., K$, have captured the low-dimensional structure of the solution space, we can approximate the solution $u(\mathbf{x}, \omega)$ well by $u(\mathbf{x}, \omega) \approx \sum_{j=1}^{K} c_j(\omega)\phi_j(\mathbf{x})$ almost surely for $\omega \in \Omega$. The number of data-driven basis functions K is determined according to the decay of the eigenvalues λ_s's.

Determining a set of good solution samples is important for the construction of the data-driven basis functions. The solution samples should span a linear space that approximates the solution space of the original PDEs well. However, the POD method itself gives no guidance on how to select the snapshots (page 503 of Benner et al. (2015)). Under certain assumptions on the random coefficient, we obtained some criteria on how to choose the coefficient samples in order to obtain a set of data-driven basis functions; see Sect. 3.4 of Li et al. (2020). In general, this issue is very challenging especially when the dimension of the random coefficient is high, which will be studied in our future work.

The computational costs of constructing the data-driven basis functions consist of two parts: (1) compute solution samples $\{u(\mathbf{x}, \omega_i)\}_{i=1}^{N}$; and (2) compute the data-driven basis by the POD method. This is common nature for many model reduction methods. Effective samples of solutions (see Sect. 3.4 of Li et al. (2020)) and the use of randomized algorithms (Halko et al. 2011) for the singular value decomposition (SVD) (utilizing the low-rank structure) help reduce the offline computation cost.

4.3 POD-based Galerkin method

Equipped with the data-driven basis function $\phi_j(\mathbf{x})$, $j = 1, ..., K$, we can solve the problem (1)-(2) on the domain D by the standard Galerkin formulation for new realizations of $a(\mathbf{x}, \omega)$. Specifically, given a new realization of the coefficient $a(\mathbf{x}, \omega)$, we approximate the corresponding solution $u(\mathbf{x}, \omega)$ as

$$u(\mathbf{x}, \omega) \approx \sum_{j=1}^{K} c_j(\omega)\phi_j(\mathbf{x}), \quad \text{a.s. } \omega \in \Omega, \tag{18}$$

and use the Galerkin projection to determine the coefficients $c_j(\omega)$, $j = 1, ..., K$. We substitute the approximation (18) into Eq.(1), multiply both side by $\phi_l(\mathbf{x})$, $l = 1, ..., K$, take integration over the domain D, and obtain a coupled linear system as follows:

$$\sum_{j=1}^{K}\int_D a(\mathbf{x}, \omega)c_j(\omega)\nabla\phi_j(\mathbf{x})\cdot\nabla\phi_l(\mathbf{x})d\mathbf{x} = \int_D f(\mathbf{x})\phi_l(\mathbf{x})d\mathbf{x}, \quad l = 1, ..., K. \tag{19}$$

The computational cost of solving the linear system (19) is small compared to using a Galerkin method, such as the finite element method, directly for $u(\mathbf{x}, \omega)$ because K is much smaller than the degree of freedom needed to discretize $u(\mathbf{x}, \omega)$ in the whole domain.

Note that if $a(\mathbf{x}, \omega)$ has the affine form (4), we first compute the terms that do not depend on randomness, including $\int_D \bar{a}(\mathbf{x})\nabla\phi_j(\mathbf{x})\cdot\nabla\phi_l(\mathbf{x})d\mathbf{x}$, $\int_D a_m(\mathbf{x})\nabla\phi_j(\mathbf{x})\cdot\nabla\phi_l(\mathbf{x})d\mathbf{x}$, and $\int_D f(\mathbf{x})\phi_j(\mathbf{x})d\mathbf{x}$, $j, l = 1, ..., K$. Then, we save them in the offline stage. This leads to considerable savings in assembling the stiffness matrix for each new realization of the coefficient $a(\mathbf{x}, \omega)$ in the online stage.

4.4 The parameter-to-solution map

To solve the Bayesian inverse problem modeled by the elliptic PDE (1), we need to compute $c_j(\omega)$ by solving the linear equation system (19) for many realizations of $a(\mathbf{x}, \omega)$. Although the data-driven basis functions provide considerable saving over standard finite element basis functions in solving (1), it still requires a certain amount of computational cost in solving the linear equation system (19) in the HMC methods. To further reduce the computational cost in the HMC method, we construct parameter-to-solution maps based on the training solution data and the data-driven basis functions.

According to our assumption, $a(\mathbf{x}, \omega)$ is parameterized by r independent random variables, i.e., $a(\mathbf{x}, \omega) = a(\mathbf{x}, \xi_1(\omega), ..., \xi_r(\omega))$.

Thus, the solution can be represented as a functional of these random variables as well, i.e., $u(\mathbf{x}, \omega) = u(\mathbf{x}, \xi_1(\omega), ..., \xi_r(\omega))$. Let $\boldsymbol{\xi}(\omega) = [\xi_1(\omega), \cdots, \xi_r(\omega)]^T$ denote the random input vector and $\mathbf{c}(\omega) = [c_1(\omega), \cdots, c_K(\omega)]^T$ denote the vector of solution coefficients in (18), respectively. Now, the problem can be viewed as constructing a parameter-to-solution map from $\boldsymbol{\xi}(\omega)$ to $\mathbf{c}(\omega)$, denoted by $\mathbf{F} : \boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$, which is nonlinear. We approximate this nonlinear map through the given solution or measurement data. Given a set of solution samples $\{u(\mathbf{x}, \omega_i)\}_{i=1}^N$ corresponding to $\{\boldsymbol{\xi}(\omega_i)\}_{i=1}^N$, e.g., by solving (1)-(2) with $a(\mathbf{x}, \xi_1(\omega_i), ..., \xi_r(\omega_i))$, from which the set of data driven basis $\phi_j(\mathbf{x})$, $j = 1, ..., K$ is obtained by using POD method as described above, we can easily compute the projection coefficients $\{\mathbf{c}(\omega_i)\}_{i=1}^N$ of $u(\mathbf{x}, \omega_i)$ on $\phi_j(\mathbf{x})$, $j = 1, ..., K$, i.e., $c_j(\omega_i) =< u(\mathbf{x}, \omega_i), \phi_j(\mathbf{x}) >_D$. From the data set, $F(\boldsymbol{\xi}(\omega_i)) = \mathbf{c}(\omega_i)$, $i = 1, ..., N$, we construct the parameter-to-solution map $\mathbf{F}$. Note the significant dimension reduction by reducing the map $\boldsymbol{\xi}(\omega) \mapsto u(\mathbf{x}, \omega)$ to the map $\boldsymbol{\xi}(\omega) \mapsto \mathbf{c}(\omega)$. We provide several ways to construct the map $\mathbf{F}$, depending on the dimension of the random input vector. More implementation details can be found in Li et al. (2020).

When the dimension of the random input r is small or moderate, one can use interpolation. In particular, if the solution samples correspond to $\boldsymbol{\xi}$ located on a uniform or sparse grid, standard polynomial interpolation can be used to approximate the coefficient c_j at a new point of $\boldsymbol{\xi}$. If the solution samples correspond to $\boldsymbol{\xi}$ at scattered points or the dimension of the random input r is moderate or high, one can first find a few nearest neighbors to the new point efficiently using the k-d tree algorithm Wald and Havran (2006) and then use the moving least square approximation centered at the new point to approximate the mapped value.

When the dimension of the random input r is high, the interpolation approach becomes expensive and less accurate. Due to the dimension reduction by the data-driven basis functions, one can train a neural network with a small output dimension to approximate the parameter-to-solution map $\mathbf{F}$. Numerical results in Li et al. (2020) show that this approach works well. We will adopt the neural network approach to approximate the parameter-to-solution maps for both the solution and its derivatives in this work.

In the HMC method, one can compute the solution $u(\mathbf{x}, \omega)$ using the constructed map $\mathbf{F}$. For example, given a new sample of $a(\mathbf{x}, \xi_1(\omega), ..., \xi_r(\omega))$, we plug $\boldsymbol{\xi}(\omega)$ into the constructed map $\mathbf{F}$ to approximate $\mathbf{c}(\omega) = \mathbf{F}(\boldsymbol{\xi}(\omega))$, which are the projection coefficients of the solution on the data-driven basis. So we can quickly obtain the new solution $u(\mathbf{x}, \omega)$ using Eq.(18), where the computational time is negligible. Similarly, we can construct data-driven basis functions and approximate the parameter-to-solution map for computing the partial derivatives of the solution. Once we obtain the numerical solutions and their derivatives, we can use them as a proposal in the HMC method. Numerical experiments show that our new method achieves significant savings in computing a new proposed sample over the standard HMC method.

5 The accelerated HMC method and its implementation

In this section, we present the data-driven and model-based accelerated HMC method for solving Bayesian elliptic inverse problems with implementation details.

In the burn-in stage, we run the standard HMC method, i.e., solving the forward elliptic problem (1) for u and solving (13) for u_{ξ_i} for the numerical evaluation of Hamiltonian dynamics in (10) using the standard finite element method. The samples of solution and its derivatives computed during the burn-in stage are collected and used to construct the data-driven basis for dimension reductions using the POD method as described in Sect. 4.2. In particular, a set of basis functions are computed for u and each u_{ξ_i}, separately. Then we use the collected samples of solution and its derivatives to train two neural networks using the Adam optimization method (see Kingma and Ba (2014)) to approximate the parameter-to-solution map described in Sect. 4.4.

Although u and u_{ξ_j} satisfy the same elliptic PDE, u has a fixed right-hand source and u_{ξ_i} has a varying right-hand source. We find that it is more efficient and accurate to construct two separate neural networks to approximate the parameter-to-solution maps, one for u and one for all u_{ξ_j}. The neural network that approximates the parameter-to-solution map

for u has a first layer that is a fully connected affine transform $\mathbf{h}_1 = \mathbf{W}_1\boldsymbol{\xi} + \mathbf{b}_1$. The following hidden layers are residual connections $\mathbf{h}_l = \tanh(\mathbf{W}_l\mathbf{h}_{l-1} + \mathbf{b}_l) + \mathbf{h}_{l-1}$ (see He et al. (2016)). The output layer is another affine transform with output $\mathbf{c}(\boldsymbol{\xi}) = \big(c_1(\boldsymbol{\xi}), c_2(\boldsymbol{\xi}), \ldots, c_K(\boldsymbol{\xi})\big)^T$ and the error to minimize is $\sum_{j=1}^N \sum_{k=1}^K |c_k(\boldsymbol{\xi}_j) - \bar{c}_k(\boldsymbol{\xi}_j)|^2$, where $\bar{c}_k(\boldsymbol{\xi}_j)$, $k = 1, 2, \ldots, K$, are the projected coefficients from j-th data $u(\mathbf{x}, \boldsymbol{\xi}_j)$, $j = 1, 2, \ldots, N$, collected during the burn-in stage. The neural network that approximates the parameter-to-solution map for all u_{ξ_i}, $i = 1, 2, \ldots, r$, where r is the dimension of the parameter space, has a similar network structure as above with an output of $\big(\mathbf{c}^1(\boldsymbol{\xi}), \mathbf{c}^2(\boldsymbol{\xi}), \ldots, \mathbf{c}^r(\boldsymbol{\xi})\big)$ and the error to minimize is $\sum_{i=1}^r \sum_{j=1}^N \sum_{k=1}^{K_i} |c_k^i(\boldsymbol{\xi}_j) - \bar{c}_k^i(\boldsymbol{\xi}_j)|^2$, where $\bar{c}_k^i(\boldsymbol{\xi}_j)$, $k = 1, 2, \ldots, K_i$, are the projected coefficients computed from j-th data $u_{\xi_i}(\mathbf{x}, \boldsymbol{\xi}_j)$, $j = 1, 2, \ldots, N$ collected during the burn-in stage.

Once the parameter-to-solution maps are trained, we can approximate the gradient of potential energy $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ effi-

ciently without solving any PDEs and hence significantly accelerate the HMC method to get the posterior samples by evolving the Markov chain as described in Sect. 3 for Bayesian inverse problems. More specifically, let $\hat{u}_{\text{net}}$ and $\hat{u}^{\text{grad}}_{\text{net}}$ denote the resulting approximations for u and $\nabla_{\boldsymbol{\xi}} u$ respectively. The gradient $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ then can be approximated as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi}) &= \frac{1}{\sigma^2}\sum_{i=1}^{m}(u(\mathbf{x}_i,\boldsymbol{\xi}) - y_i)\nabla_{\boldsymbol{\xi}} u(\mathbf{x}_i,\boldsymbol{\xi}) - \nabla_{\boldsymbol{\xi}} \log p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \\ &\approx \frac{1}{\sigma^2}\sum_{i=1}^{m}(\hat{u}_{\text{net}}(\mathbf{x}_i,\boldsymbol{\xi}) - y_i)\hat{u}^{\text{grad}}_{\text{net}}(\mathbf{x}_i,\boldsymbol{\xi}) - \nabla_{\boldsymbol{\xi}} \log p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \\ &:= \hat{U}^{\text{grad}}_{\text{net}}(\boldsymbol{\xi})\end{aligned}$$

where $\mathbf{y} = (y_1, \ldots, y_m)$ is the observed data at locations $\mathbf{x}_1, \ldots, \mathbf{x}_m$. We then substitute $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ with $\hat{U}^{\text{grad}}_{\text{net}}(\boldsymbol{\xi})$ when simulating the Hamiltonian dynamics using the following leap-frog scheme

$$\begin{aligned}\boldsymbol{\zeta}^{(t+\frac{1}{2})} &= \boldsymbol{\zeta}^{(t)} - \frac{\Delta t}{2}\hat{U}^{\text{grad}}_{\text{net}}(\boldsymbol{\xi}^{(t)}), \\ \boldsymbol{\xi}^{(t+1)} &= \boldsymbol{\xi}^{(t)} + \Delta t \nabla_{\boldsymbol{\zeta}} K(\boldsymbol{\zeta}^{(t+\frac{1}{2})}), \\ \boldsymbol{\zeta}^{(t+1)} &= \boldsymbol{\zeta}^{(t+\frac{1}{2})} - \frac{\Delta t}{2}\hat{U}^{\text{grad}}_{\text{net}}(\boldsymbol{\xi}^{(t+1)}).\end{aligned}$$

The acceptance probability of the proposed state $(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*)$ at the end of a simulated trajectory is computed the same way as in (11), i.e.,

$$\alpha_{\text{HMC}} = \min\Big(1, \exp[-H(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*) + H(\boldsymbol{\xi}, \boldsymbol{\zeta})]\Big).$$

where $(\boldsymbol{\xi}, \boldsymbol{\zeta})$ is the starting state and H is the exact Hamiltonian computed using the standard FEM solution. We list the implementation steps of the accelerated HMC algorithm in Algorithm 1.

The extra computation cost for our proposed accelerated HMC method is the construction of the data-driven basis and training of the two neural networks to approximate the parameter-to-solution maps. Due to the intrinsic approximate low-dimensional structure of the solution (and its derivatives) of the forward elliptic model, the singular values of the data covariance matrix decay very fast. So one only needs to approximate the space spanned by a few leading singular vectors which can be computed efficiently using randomized SVD algorithms as described in Sect. 4.2 (and more details in Li et al. (2020)). After significant model-based dimension reduction, a rather shallow and small neural network with a simple structure and low-dimensional input and output is enough to approximate the parameter-to-solution map well in practice. Hence, evaluation of the constructed parameter-to-solution map vs a full computation of the forward elliptic PDE significantly reduces the computation cost in each leap-frog step of the HMC dynamic after the burn-in stage.

Algorithm 1 The accelerated HMC algorithm.

1: Input: the prior distribution for $a(\mathbf{x}, \boldsymbol{\xi}(\omega))$.
2: Collect samples of solution and its partial derivatives, i.e. $\{\boldsymbol{\xi}_j, u(\boldsymbol{\xi}_j), \frac{\partial u(\boldsymbol{\xi}_j)}{\partial \xi_1}, \cdots, \frac{\partial u(\boldsymbol{\xi}_j)}{\partial \xi_r}\}_{j=1}^{N}$ during the burn-in stage.
3: Extract basis functions $\{\phi_j(\mathbf{x})\}_{j=1}^{K}$ for the solution and basis functions $\{\phi_j^i(\mathbf{x})\}_{j=1}^{K_i}$ for the partial derivatives of the solution, $i = 1, \cdots, r$, from the collected data using the POD method.
4: Get the training data $\{\boldsymbol{\xi}_j, \mathbf{c}(\boldsymbol{\xi}_j), \mathbf{c}^1(\boldsymbol{\xi}_j), \cdots, \mathbf{c}^r(\boldsymbol{\xi}_j)\}_{j=1}^{N}$ by projecting the samples of solution and its partial derivatives onto the corresponding basis.
5: Train a neural network fitting the data pair $\{\boldsymbol{\xi}, \mathbf{c}(\boldsymbol{\xi})\}$ and train another neural network fitting the data pair $\{\boldsymbol{\xi}, (\mathbf{c}^1(\boldsymbol{\xi}), \cdots, \mathbf{c}^r(\boldsymbol{\xi}))\}$ to approximate the parameter-to-solution maps.
6: Generate samples from posterior distribution via the accelerated HMC algorithm.
   (1) at the current position $\boldsymbol{\xi}$, sample a new momentum $\boldsymbol{\zeta} \sim \mathcal{N}(0, M)$ to get a starting point $(\boldsymbol{\xi}, \boldsymbol{\zeta})$;
   (2) apply the leap-frog scheme (10) to compute the Hamiltonian dynamic, with data-driven gradient for the potential by using the learned parameter-to-solution maps in step 5;
   (3) accept the proposed sample $(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*)$ at the end of the trajectory with probability (11), where H is computed using the FEM reference solution.
7: Output: samples of $\{\boldsymbol{\xi}\}$ that converge to the posterior distribution.

Moreover, our data and model-based dimension reduction method captures the intrinsic low-dimensional structure of the underlying problem with a data-driven basis and accuracy control (through the POD method) that strikes a good balance between computation efficiency (by dimension reduction and parameter-to-solution approximation) and exploration efficiency (by proposing well-decorrelated samples with high acceptance rate), as demonstrated by numerical experiments in the next section.

6 Alternative approaches for gradient computation

In this section, we discuss two alternative approaches for gradient computation in the HMC method. The first one is based on the adjoint method (Givoli 2021; Natterer 2015). Let's consider the following general mixed boundary condition for the forward problem

$$-\nabla\cdot\big(a(\mathbf{x},\boldsymbol{\xi}(\omega))\nabla u(\mathbf{x},\boldsymbol{\xi}(\omega))\big) = f(\mathbf{x}), \quad \mathbf{x}\in D, \quad \omega\in\Omega, \tag{20}$$

$$u(\mathbf{x},\boldsymbol{\xi}(\omega)) = g(\mathbf{x}), \qquad \mathbf{x}\in\Gamma_d, \tag{21}$$

$$a(\mathbf{x},\boldsymbol{\xi}(\omega))\frac{\partial u(\mathbf{x},\boldsymbol{\xi}(\omega))}{\partial \mathbf{n}} + b(\mathbf{x},\boldsymbol{\xi}(\omega))u(\mathbf{x},\boldsymbol{\xi}(\omega)) = h(\mathbf{x}), \qquad \mathbf{x}\in\Gamma_r. \tag{22}$$

where $\Gamma_d \cup \Gamma_r = \partial D$. Let $\mathcal{C} = \{w : w \in H^1(D), w(\mathbf{x}) = 0, \forall\, \mathbf{x} \in \Gamma_d\}$. The corresponding weak formulation

$$\int_D a\nabla u \nabla w d\mathbf{x} + \int_{\Gamma_r} buw dS = \int_{\Gamma_r} hw dS + \int_D fw d\mathbf{x}, \quad \forall w \in \mathcal{C}. \tag{23}$$

Taking derivatives with respective to $\boldsymbol{\xi}$ (i.e. ξ_i, $i = 1, ..., r$), we have

$$\int_D \frac{\partial a}{\partial \xi_i}\nabla u \nabla w d\mathbf{x} + \int_D a\nabla \frac{\partial u}{\partial \xi_i}\nabla w d\mathbf{x} = -\int_{\Gamma_r} \frac{\partial b}{\partial \xi_i} uw dS - \int_{\Gamma_r} b\frac{\partial u}{\partial \xi_i} w dS. \tag{24}$$

Let $v(\mathbf{x}, \boldsymbol{\xi})$ solves the corresponding adjoint equation

$$-\nabla \cdot \big(a(\mathbf{x}, \boldsymbol{\xi}(\omega))\nabla v(\mathbf{x}, \boldsymbol{\xi}(\omega))\big) = y(\mathbf{x}) - u(\mathbf{x}, \boldsymbol{\xi}), \quad \mathbf{x} \in D, \tag{25}$$

$$v(\mathbf{x}, \boldsymbol{\xi}(\omega)) = 0, \qquad \mathbf{x} \in \Gamma_d, \tag{26}$$

$$a(\mathbf{x}, \boldsymbol{\xi}(\omega))\frac{\partial v(\mathbf{x}, \boldsymbol{\xi}(\omega))}{\partial \mathbf{n}} + b(\mathbf{x}, \boldsymbol{\xi}(\omega))v(\mathbf{x}, \boldsymbol{\xi}(\omega)) = 0, \qquad \mathbf{x} \in \Gamma_r. \tag{27}$$

where $y(\mathbf{x})$ is the measurement at $\mathbf{x}$. Then, the the corresponding weak formulation for v is

$$\int_D a\nabla v \nabla w d\mathbf{x} + \int_{\Gamma_r} bvw dS = \int_D (y - u)w d\mathbf{x}, \quad \forall w \in \mathcal{C}. \tag{28}$$

Define the energy function as $J(\boldsymbol{\xi}) = \frac{1}{2}\int_D (y - u)^2 d\mathbf{x}$, which is proportional to negative log-likelihood (see Eq.(7)). Then, we can easily find its derivative with respect to $\boldsymbol{\xi}$ as follows:

$$\begin{aligned}\frac{\partial J}{\partial \xi_i} &= -\int_D (y - u)\frac{\partial u}{\partial \xi_i} d\mathbf{x} \\ &= -\int_D a\nabla v \nabla \frac{\partial u}{\partial \xi_i} d\mathbf{x} - \int_{\Gamma_r} bv\frac{\partial u}{\partial \xi_i} dS \\ &= \int_D \frac{\partial a}{\partial \xi_i}\nabla u \nabla v d\mathbf{x} + \int_{\Gamma_r} \frac{\partial b}{\partial \xi_i} uv dS.\end{aligned} \tag{29}$$

When the random field $a(\mathbf{x}, \boldsymbol{\xi})$ and the boundary coefficient $b(\mathbf{x}, \boldsymbol{\xi})$ are known, $\nabla_{\boldsymbol{\xi}} J$ can be computed from (29) once u is solved for the forward problem and v is solved for the adjoint problem, e.g., by FEM. In practice, $\int_D (y - u)w d\mathbf{x}$ in (28) is replaced by the discrete approximation $\frac{1}{m}\sum_{j=1}^m \big(y_j - u(x_j, \boldsymbol{\xi})\big)w(x_j)$. We point out that although the adjoint method is an efficient alternative for gradient computation to the standard approach in (13) and (14) that uses multiple PDEs, the adjoint method does not take advantage of the low dimensional structure of the underlying PDE and still requires to solve one forward PDE and one adjoint PDE for each gradient evaluation during the simulation of Hamiltonian dynamics. By exploiting the intrinsic low-dimensional structures of the underlying model and constructing a set of data-driven basis functions, our proposed accelerated HMC method can efficiently approximate the nonlinear map $\boldsymbol{\xi} \to U(\boldsymbol{\xi})$ and its gradient without solving any PDE for gradient evaluation after training/burning stage. Numerical results in the next section show that our proposed method outperforms the adjoint method approach.

We can also consider a direct approach for the gradient computation in the HMC method using the deep learning method. Specifically, we use a neural network to directly approximate the complicated nonlinear map $\boldsymbol{\xi} \to U(\boldsymbol{\xi})$ and thus we can compute $\Phi(\cdot; \mathbf{y})$ in the likelihood function (7). This idea can be viewed as an end-to-end learning technique, which is easy to implement and relies on a single neural network to explore the full model and approximate the map. However, it is hard to incorporate the intrinsic low-dimensional structure of the PDE solution space directly into the network. Therefore, using this approach, one needs to choose a large size of hidden layers and neurons per layer, in order to obtain an accurate approximation. One can see the low acceptance rate associated with the end-to-end learning approach in Fig. 1.

7 Numerical experiments and results

In this section, we use numerical experiments to demonstrate the accuracy and efficiency of the accelerated HMC method for Bayesian inverse problems, with comparison to other state-of-the-art methods, including the standard HMC method and random network surrogate method (Zhang et al. 2017a). The Python codes are published on GitHub.[1]

We consider the elliptic inverse problem

$$-\nabla \cdot \big(a(\mathbf{x}, \omega)\nabla u(\mathbf{x}, \omega)\big) = 0, \quad \mathbf{x} = (x_1, x_2) \in [0, 1] \times [0, 1] \tag{30}$$

with mixed boundary conditions,

$$\frac{\partial u(\mathbf{x}, \omega)}{\partial \mathbf{n}}|_{x_1=0, x_1=1} = 0, \quad u(\mathbf{x}, \omega)|_{x_2=0} = x_1, \quad u(\mathbf{x}, \omega)|_{x_2=1} = 1 - x_1. \tag{31}$$

[1] https://github.com/LSijing/Bayesian-pde-inverse-problem.

7.1 A log-normal coefficient with isotropic heterogeneity

In the first example, a Gaussian prior with zero mean and covariance function

$$c(\mathbf{x}, \mathbf{x}') = \sigma_a^2 \exp\Big(-\frac{||\mathbf{x}-\mathbf{x}'||_2^2}{2l^2}\Big) \tag{32}$$

is assumed on $\log(a(\mathbf{x}, \omega))$, where $\mathbf{x}, \mathbf{x}'$ are any two points on $[0, 1] \times [0, 1]$, and the parameters σ_a^2 and l denote the variance and the correlation length, respectively. The diffusion coefficient is approximated via a truncated Karhunen-Loève (KL) expansion

$$\log(a(\mathbf{x}, \boldsymbol{\xi})) = \sum_{i=1}^{r} \xi_i(\omega)\sqrt{\lambda_i}\, v_i(\mathbf{x}), \tag{33}$$

by r i.i.d. Gaussian random variables $\xi_i(\omega)$, where $\boldsymbol{\xi} = (\xi_1(\omega), ..., \xi_r(\omega))$, λ_i and $v_i(\mathbf{x}), i = 1, 2, \cdots, r$ are eigenvalues and eigenfunctions of the prior covariance function (32). In this experiment, we test the performance of the accelerated HMC Algorithm 1 for different input random dimensions, $r = 25, 30, 35$.

Suppose the observation $\mathbf{y} = \big(y_1, y_2, \cdots, y_m\big)$ is obtained by adding independent Gaussian noise to the exact solutions at some measurable locations

$$y_j = u(x_j, \boldsymbol{\xi}) + \eta_j, \quad \eta_j \sim \mathcal{N}(0, \sigma^2), \quad j = 1, 2, \cdots, m. \tag{34}$$

Our goal is to infer $\boldsymbol{\xi}$ and hence $a(\mathbf{x}, \boldsymbol{\xi})$ based on the observation data $\mathbf{y}$. In the Bayesian framework, the posterior on $\boldsymbol{\xi}$ is

$$\begin{aligned} p_{\boldsymbol{\xi}|\mathbf{y}}(\boldsymbol{\xi}|\mathbf{y}) &\propto p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) \cdot p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \\ &\propto \exp(-\frac{1}{2\sigma^2}\|\mathbf{y} - u(\mathbf{x}, \boldsymbol{\xi})\|^2) \exp(-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\xi}), \end{aligned} \tag{35}$$

which is the target distribution. Thanks to an efficient approximation to the parameter-to-solution maps, the computation cost of $u(\mathbf{x}, \boldsymbol{\xi})$ and $\nabla_{\boldsymbol{\xi}} u(\mathbf{x}, \boldsymbol{\xi})$ is significantly reduced and hence the computation of likelihood (35) and Hamiltonian dynamics (10) in the accelerated HMC method is very fast. Moreover, as demonstrated later on, the acceptance rate and exploration efficiency do not compromise much as a consequence of model and data-based dimension reduction. So the overall performance of the HMC method is enhanced significantly.

To generate the training data, the discretization is done on a uniform grid with 31×31 points through triangle finite element basis functions. Suppose the measurements are placed on 11×11 grids of the numerical solution $u(\mathbf{x}, \cdot)$, i.e. $m = 121$ in (34). We choose $\sigma = 0.1$ be the noise in the observation data (34), and $\sigma_a = 0.5$, $l = 0.2$ be parameters in the prior covariance function (32).

The burn-in stage consists of 10000 steps of standard HMC, of which 9000 accepted samples of solutions u and its derivatives $\frac{\partial u}{\partial \xi_i}, i = 1, 2, \ldots, r$ are collected during the burn-in stage. These collected data are first used to construct a set of data-driven basis using POD method for dimension reduction. In our previous study (Li et al. 2020), we found that more basis functions are needed to approximate the derivatives of the solution than those needed to approximate the solution. Specifically, we construct $K = 20$ basis $\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \ldots, \phi_K(\mathbf{x})$ for the approximation of u, and $K_i = 40$ basis $\phi_1^i(\mathbf{x}), \phi_2^i(\mathbf{x}), \ldots, \phi_{K_i}^i(\mathbf{x})$ for the approximation of each $\frac{\partial u}{\partial \xi_i}, i = 1, 2, \ldots, r$.

Once the data-driven basis are constructed, we then use the collected data to train two neural networks to approximate the parameter-to-solution maps, one for $\boldsymbol{\xi} \to \mathbf{c}(\boldsymbol{\xi})$ that gives $u(\mathbf{x}; \boldsymbol{\xi}) = \sum_{k=1}^{K} c_k(\boldsymbol{\xi})\phi_k(\mathbf{x})$, and another one for $\boldsymbol{\xi} \to (\mathbf{c}^1(\boldsymbol{\xi}), \mathbf{c}^2(\boldsymbol{\xi}) \ldots, \mathbf{c}^r(\boldsymbol{\xi}))$ that gives $\frac{\partial u(\mathbf{x};\boldsymbol{\xi})}{\partial \xi_i} = \sum_{k=1}^{K^i} c_k^i(\boldsymbol{\xi})\phi_k^i(\mathbf{x})$, $i = 1, 2, \ldots, r$, as described in Sect. 5. In our experiments, the first network has 4 hidden layers and 20 units within each hidden layer. The second network has the same structure except there are 40 hidden units in each hidden layer.

We specify the number of leap-frog steps in (10) to be 10, and the step size $\Delta t = 0.16$ for all methods. Typically, we start sampling from the posterior after observing mixing. For the standard HMC method and random network surrogate method (Zhang et al. 2017a), with which we compare, they share the same burn-in stage and starting point. We compute the relative error of the posterior mean up to a computation time t by

$$\frac{\big|\big|\frac{1}{\#\{i:t_i \le t\}}\sum_{i:t_i \le t} \boldsymbol{\xi}_i - E_{\boldsymbol{\xi}|\mathbf{y}}(\boldsymbol{\xi})\big|\big|_2}{\big|\big|E_{\boldsymbol{\xi}|\mathbf{y}}(\boldsymbol{\xi})\big|\big|_2}. \tag{36}$$

The left column of Fig. 1 plots the relative error of the posterior mean vs computation time in log scale for the standard HMC, the random network surrogate method, the proposed accelerated HMC method, the adjoint method, and the end-to-end learning method for $r = 25, 30, 35$, respectively. We see significantly improved performance of the proposed method. The right column of Fig. 1 plots the corresponding acceptance rate for the proposal by Hamiltonian dynamics for each method. As we can see, the significant dimension reduction and the efficient neural network approximation of the parameter-to-solution map does not compromise the acceptance rate much. Moreover, the acceptance rate maintains high and stable as the input dimension increases for our model-based and data-driven approach.

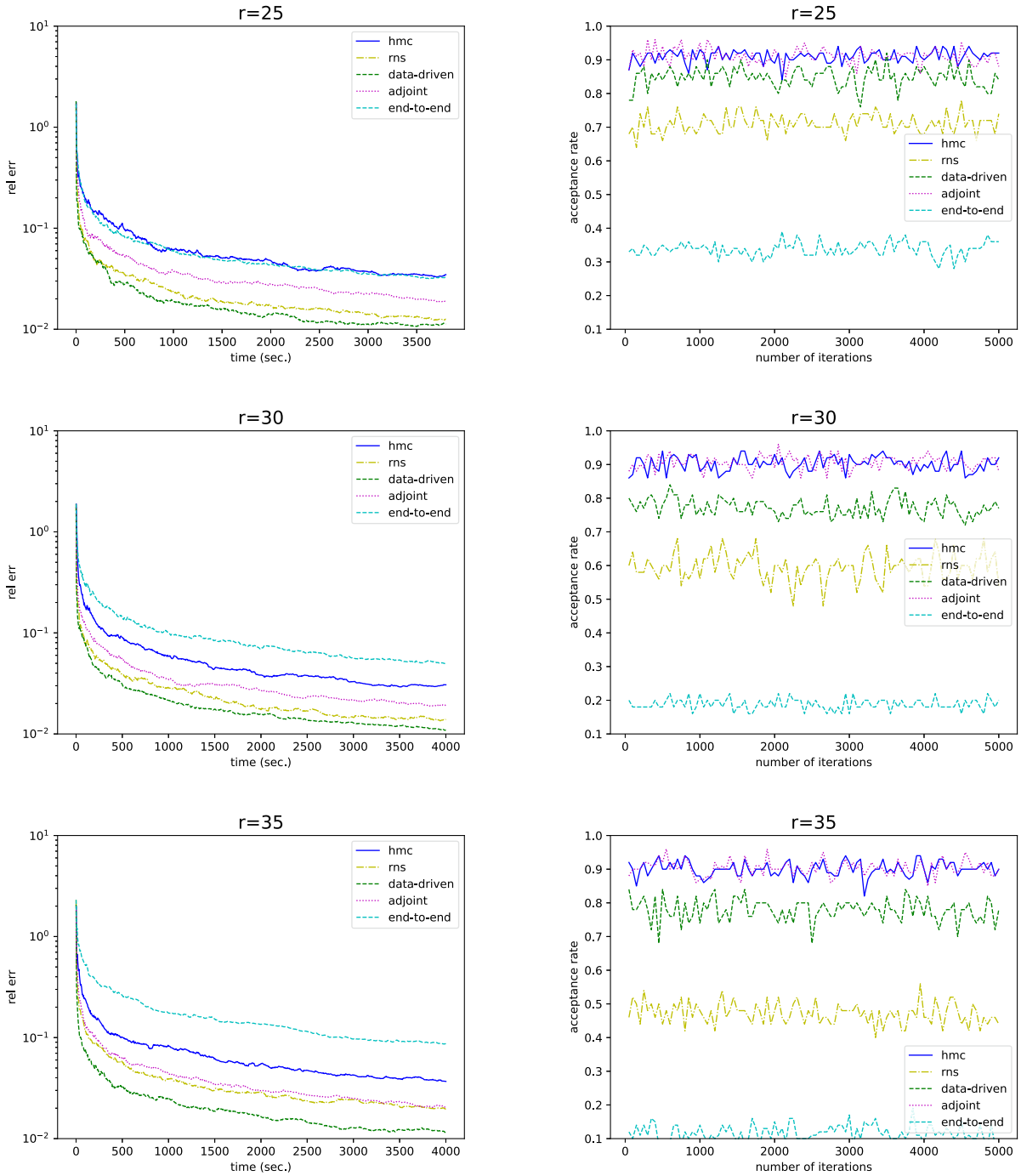


Fig. 1 Numerical results for the random coefficient with 25-, 30- and 35-dimensional inputs, where "hmc", "rns", "data-driven", "adjoint", and "end-to-end" refer to the standard HMC method, the random network surrogate method, the accelerated HMC method, the adjoint method, and the end-to-end learning method, respectively

For the random network surrogate method, the surrogate of the Hamiltonian in the parameter space is based on the least square approximation of sampled data, e.g, from the burn-in stage, using a set of random basis. Since this approach is purely data-driven without model knowledge, to maintain the approximation accuracy, the number of random basis has to increase with the dimension of the parameter space although the intrinsic dimension of the underlying model remains the same. In this experiment, we fix the number of random basis at 1000. As the input dimension increases, the approximation power of the surrogate using a fixed number of basis decreases, and hence the approximation error becomes larger and the acceptance rate drops quite sharply.

Table 1 shows the averaged time per each HMC iteration, averaged acceptance rate, effective sample size (min, median, max), and time normalized effective sample size for each method, respectively. The effective sample size is defined as

$$ESS = B\big(1 + 2\sum_{k=1}^{K}\gamma(k)\big)^{-1}, \tag{37}$$

where B is the number of the MCMC samples and $\sum_{k=1}^{K}\gamma(k)$ is the sum of K monotone sample autocorrelations. It shows that our model-based and data-driven approach has a good balance between the computation efficiency and exploration efficiency and hence achieves the best overall performance.

7.2 A log-normal coefficient with anisotropic heterogeneity

In the second example, a Gaussian prior with zero mean and covariance function

$$c(\mathbf{x}, \mathbf{x}') = \sigma_a^2 \exp\Big(-\frac{|x_1 - x_1'|^2}{2l_1^2} - \frac{|x_2 - x_2'|^2}{2l_2^2}\Big) \tag{38}$$

is assumed on $\log(a(\mathbf{x}, \omega))$, where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{x}' = (x_1', x_2')$ are any two points on $[0, 1] \times [0, 1]$, and l_1 and l_2 are the correlation lengths in x_1 and x_2. The diffusion coefficient is approximated via a truncated Karhunen-Loève (KL) expansion as in (33), only with a different prior covariance function (38).

To generate the training data, we solve the elliptic problem (30) with the same boundary condition (31). The discretization is done on a uniform grid with 65×65 points through triangle finite element basis functions. We choose the number of truncated KL modes $r = 30$, $\sigma = 0.1$ be the noise in the observation data (34), and $\sigma_a = 0.5$, $l_1 = 0.08$ and $l_2 = 0.4$ in the prior covariance function (38). All other settings are the same as in Sect. 7.1. Suppose the measurements are placed on 17×17 grids of the numerical solution $u(\mathbf{x}, \cdot)$, i.e. $m = 289$ in (34). We now infer the log-normal coefficient $\log(a(\mathbf{x}, \boldsymbol{\xi}))$ based on the observation data.

To illustrate that our method indeed converges to the right target distribution, Fig. 2 provides the one- and two-dimensional posterior marginals of some selected parameters obtained by the standard HMC method and the accelerated HMC method. Figure 3 shows the posterior mean and posterior standard deviation obtained by the standard HMC method and the accelerated HMC method, respectively. The relative errors of the posterior mean and posterior standard deviation are 0.047 and 0.024. Therefore, with the accelerated HMC method, we can significantly reduce the computation cost (by almost an order of magnitude in this case) while maintaining the approximation accuracy of the standard HMC method.

Finally, we compare the partial derivatives $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_1}$ and $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_2}$ at the approximate the MAP state obtained via standard HMC method and the accelerated HMC method in Fig. 4. The relative errors of $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_1}$ and $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_2}$ are 0.013 and 0.019, respectively. We also examine the relative errors at ten posterior sample of $\boldsymbol{\xi}$ and the result is presented in Table 2. These results demonstrate the effectiveness of our intrinsic low-dimensional data-driven basis on providing fast and accurate gradient approximations for accelerating the Hamiltonian dynamics in the HMC method.

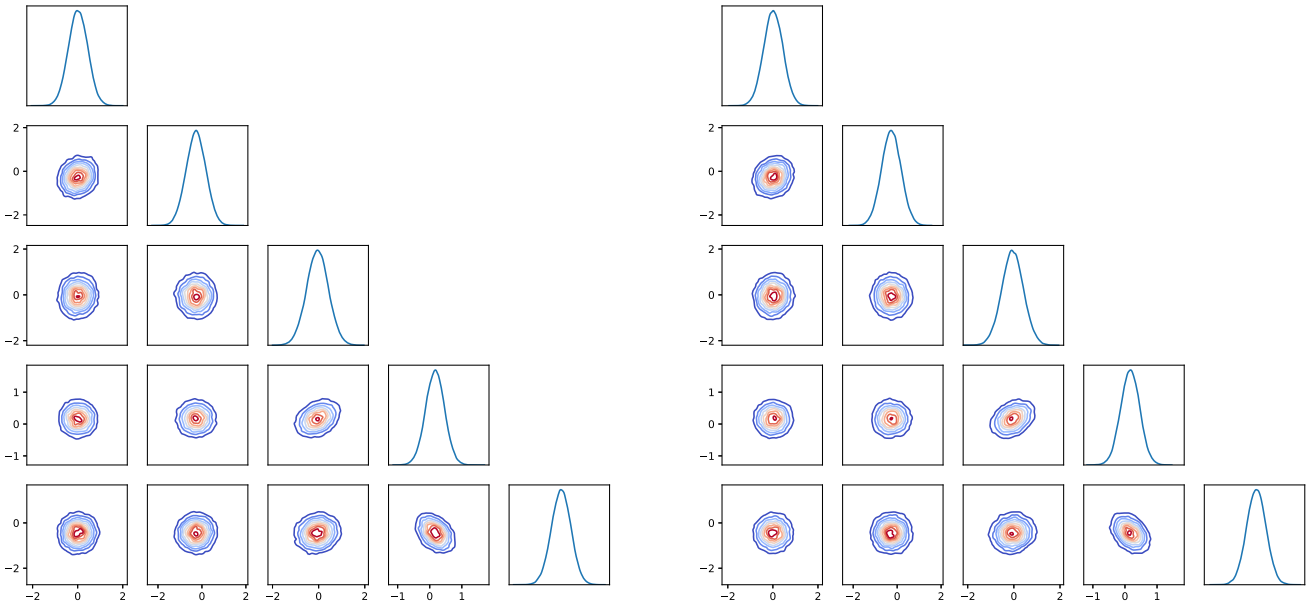
8 Conclusion

The HMC method can generate less correlated proposals with high acceptance probabilities, which greatly improves the performance of the MCMC methods in solving Bayesian inverse problems. However, when applying the HMC method to solve a Bayesian inverse problem modeled by elliptic partial differential equations, one needs to compute the solution to the elliptic PDEs and their derivatives repeatedly in order to generate data and evaluate the Hamiltonian, which makes the HMC method extremely expensive.

By exploiting the intrinsic low-dimensional structures of the underlying model and constructing a set of data-driven basis, our proposed method achieves significant dimension reduction in the solution space. Then, equipped with the data-driven basis, neural networks are trained as efficient approximations of the parameter-to-solution maps, which significantly reduce the computation cost in obtaining the

Table 1 Comparisons of different methods. Here "hmc", "rns", "data-driven", "end-to-end", and "adjoint" refer to the standard HMC method, the random network surrogate method, the accelerated HMC method, the end-to-end learning method, and the adjoint method, respectively. The acceptance rate (AR), computational time for each iteration, effective sample size (ESS), and time-normalized ESS are provided

Dimension	Method	AR	s/Iter	ESS	min(ESS)/s	med(ESS)/s
r = 25	hmc	0.91	1.27	(4569, 5000, 5000)	0.72	0.79
	rns	0.71	0.076	(1890, 2518, 3094)	4.98	6.63
	data-driven	0.85	0.094	(3395, 4306, 4932)	7.22	9.16
	end-to-end	0.34	0.108	(507, 626, 867)	0.94	1.16
	adjoint	0.91	0.466	(4475, 5000, 5000)	1.92	2.15
r = 30	hmc	0.90	1.49	(4592, 5000, 5000)	0.62	0.67
	rns	0.60	0.082	(1312, 1661, 2068)	3.20	4.05
	data-driven	0.77	0.108	(2907, 3414, 4116)	5.38	6.32
	end-to-end	0.19	0.127	(238, 275, 389)	0.38	0.43
	adjoint	0.90	0.5	(4482, 5000, 5000)	1.79	2.00
r = 35	hmc	0.90	1.72	(4388, 5000, 5000)	0.51	0.58
	rns	0.48	0.095	(693, 1008, 1308)	1.46	2.12
	data-driven	0.78	0.125	(2831, 3445, 4029)	4.53	5.51
	end-to-end	0.12	0.148	(160, 178, 222)	0.22	0.24
	adjoint	0.90	0.61	(4341, 4995, 5000)	1.42	1.64



(a) The standard HMC method.

(b) The accelerated HMC method.

Fig. 2 Comparing one- and two-dimensional posterior marginals of $\xi_2, \xi_4, \xi_7, \xi_9, \xi_{13}$

Table 2 relative errors between the FEM reference solution and the data-driven approximation at ten random samples from the posterior of $\boldsymbol{\xi}$

$\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_1}$	0.032	0.014	0.036	0.054	0.016	0.021	0.032	0.031	0.028	0.018
$\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_2}$	0.054	0.045	0.022	0.045	0.024	0.037	0.043	0.057	0.060	0.032

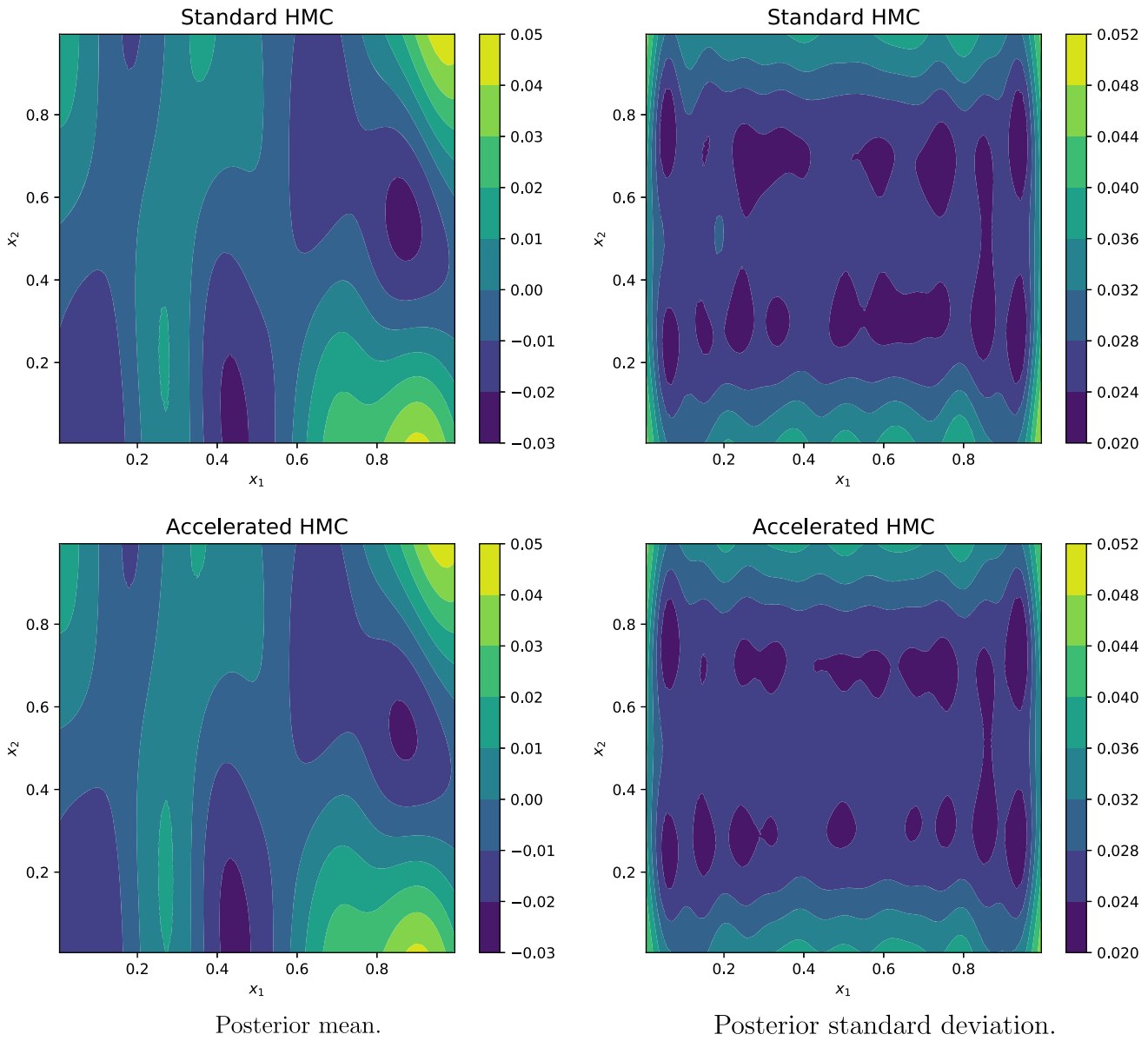
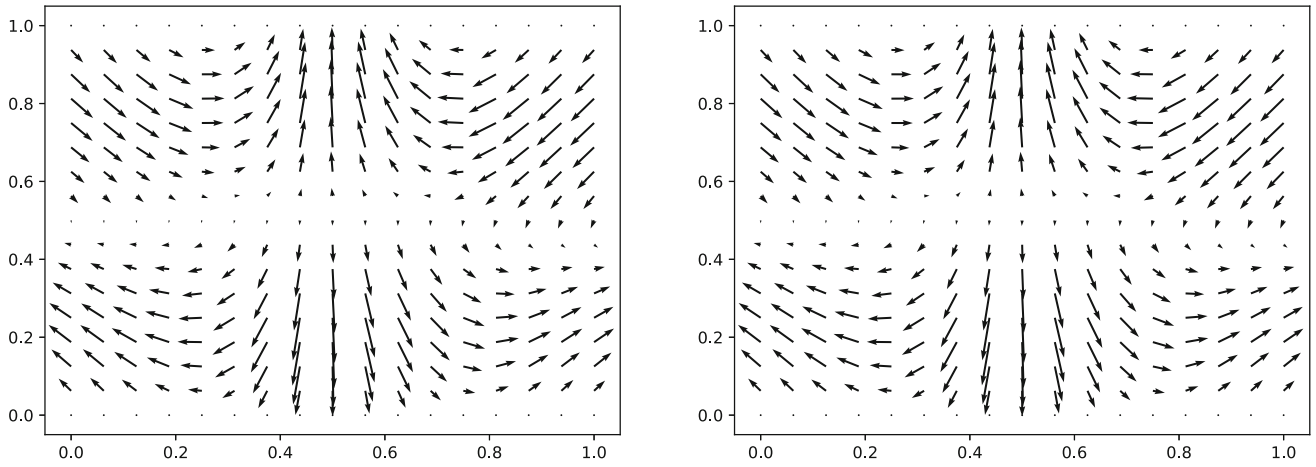


Fig. 3 Posterior statistics obtained by standard HMC and accelerated HMC

PDE solution and its derivatives for the Hamiltonian dynamics in proposing a new sample. Through numerical tests, we demonstrate that our method strikes a good balance between computation efficiency and exploration efficiency and provides an effective data and model-based approach for solving elliptic Bayesian inverse problems.



(a) The standard HMC method. (b) The accelerated HMC method.

Fig. 4 Partial derivative of $u(\mathbf{x}, \boldsymbol{\xi})$ with respect to ξ_1 and ξ_2

Acknowledgements The research of S. Li is partially supported by the Doris Chen Postgraduate Scholarship. The research of C. Zhang is partially supported by National Natural Science Foundation of China (projects 12201014 and 12292983), the Key Laboratory of Mathematics and Its Applications (LMAM) and the Key Laboratory of Mathematical Economics and Quantitative Finance (LMEQF) of Peking University. The research of Z. Zhang is supported by the Hong Kong RGC General Research Fund projects 17300318 and 17307921, National Natural Science Foundation of China (project 12171406), an R&D Funding Scheme from the HKU-SCF FinTech Academy, Seed Funding Programme for Basic Research (HKU), and the outstanding young researcher award of HKU (2020–21). The research of H. Zhao is partially supported by NSF grants DMS-2048877 and DMS-2012860. The computations were performed using research computing facilities offered by Information Technology Services, the University of Hong Kong.

Data availability and materials The data generated and analysed during the current study are available from the corresponding author on reasonable request.

Code availability The Python codes are published on GitHub.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Abdulle, A., Barth, A., Schwab, C.: Multilevel Monte Carlo methods for stochastic elliptic multiscale PDEs. Multiscale Model. Simul. **11**, 1033–1070 (2013)

Asokan, B.V., Zabaras, N.: A stochastic variational multiscale method for diffusion in heterogeneous random media. J. Comput. Phys. **218**, 654–676 (2006)

Babuska, I., Nobile, F., Tempone, R.: A stochastic collocation method for elliptic partial differential equations with random input data. SIAM J. Numer. Anal. **45**, 1005–1034 (2007)

Babuska, I., Tempone, R., Zouraris, G.: Galerkin finite element approximations of stochastic elliptic partial differential equations. SIAM J. Numer. Anal. **42**, 800–825 (2004)

Bachmayr, M., Cohen, A., DeVore, R., Migliorati, G.: Sparse polynomial approximation of parametric elliptic PDEs, Part II: lognormal coefficients. ESAIM: Math. Model. Numer. Anal. **51**, 341–363 (2017)

Barrault, M., Maday, Y., Nguyen, N.C., Patera, A.T.: An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. C.R. Math. **339**(9), 667–672 (2004)

Benner, P., Gugercin, S., Willcox, K.: A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev. **57**, 483–531 (2015)

Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**(1), 539–575 (1993)

Beskos, A., Jasra, A., Muzaffer, E., Stuart, A.: Sequential Monte Carlo methods for Bayesian elliptic inverse problems. Stat. Comput. **25**, 727–737 (2015)

Bryson, J., Zhao, H., Zhong, Y.: Intrinsic complexity and scaling laws: from random fields to random vectors. SIAM Multiscale Model. Simul. **17**(1), 460–481 (2019)

Chung, E., Efendiev, Y., Leung, W., Zhang, Z.: Cluster-based generalized multiscale finite element method for elliptic PDEs with random coefficients. J. Comput. Phys. **371**, 606–617 (2018)

Cohen, A., DeVore, R.: Approximation of high-dimensional parametric PDEs. Acta Numer. **24**, 1–159 (2015)

Dashti, M., Stuart, A.: Uncertainty quantification and weak approximation of an elliptic inverse problem. SIAM J. Numer. Anal. **49**, 2524–2542 (2011)

Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B **195**, 216–222 (1987)

Efendiev, Y., Kronsbein, C., Legoll, F.: Multilevel Monte Carlo approaches for numerical homogenization. Multiscale Model. Simul. **13**, 1107–1135 (2015)

Ghanem, R., Spanos, P.: Stochastic Finite Elements: A Spectral Approach. Springer-Verlag, New York (1991)

Girolami, M., Calderhead, B.: Riemann manifold Langevin and Hamiltonian Monte Carlo methods. J. R. Stat. Soc.: Series B (Stat. Methodol.) **73**(2), 123–214 (2011)

Givoli, D.: A tutorial on the adjoint method for inverse problems. Comput. Methods Appl. Mech. Eng. **380**, 113810 (2021)
Graham, I., Kuo, F., Nuyens, D., Scheichl, R., Sloan, I.: Quasi-Monte Carlo methods for elliptic PDEs with random coefficients and applications. J. Comput. Phys. **230**, 3668–3694 (2011)
Graham, I.G., Kuo, F.Y., Nichols, J.A., Scheichl, R., Schwab, C., Sloan, I.H.: Quasi-Monte Carlo finite element methods for elliptic PDEs with lognormal random coefficients. Numer. Math. **131**(2), 329–368 (2015)
Halko, N., Martinsson, P., Tropp, J.: Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. SIAM Rev. **53**, 217–288 (2011)
He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778 (2016)
Hoang, V., Schwab, C.: N-term wiener chaos approximation rates for elliptic PDEs with lognormal Gaussian random inputs. Math. Models Methods Appl. Sci. **24**, 797–826 (2014)
Holmes, P., Lumley, J., Berkooz, G.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, Cambridge (1998)
Hou, T., Liu, P., Zhang, Z.: A localized data-driven stochastic method for elliptic PDEs with random coefficients. Bull. Inst. Math. Acad. Sin. **1**, 179–216 (2016)
Hou, T., Ma, D., Zhang, Z.: A model reduction method for multiscale elliptic PDEs with random coefficients using an optimization approach. Multiscale Model. Simul. **17**, 826–853 (2019)
Kaipio, J., Somersalo, E.: Statistical and Computational Inverse Problems, vol. 160. Springer, New York (2005)
Kingma, D. P., Ba, J.: *Adam: A method for stochastic optimization*, arXiv preprint arXiv:1412.6980, (2014)
Lan, S.: Adaptive dimension reduction to accelerate infinite-dimensional geometric Markov Chain Monte Carlo. J. Comput. Phys. **392**, 71–95 (2019)
Lan, S., Bui-Thanh, T., Christie, M., Girolami, M.: Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems. J. Comput. Phys. **308**, 81–101 (2016)
Li, S., Zhang, Z., Zhao, H.: A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction. SIAM J. Multiscale Model. Simul. **18**(3), 1242–1271 (2020)
Martin, J., Wilcox, L., Burstedde, C., Ghattas, O.: A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. SIAM J. Sci. Comput. **34**, A1460–A1487 (2012)
Mondal, A., Efendiev, Y., Mallick, B., Datta-Gupta, A.: Bayesian uncertainty quantification for flows in heterogeneous porous media using reversible jump Markov chain Monte Carlo methods. Adv. Water Resour. **33**, 241–256 (2010)
Natterer, F.: Adjoint methods as applied to inverse problems. Encyclopedia Appl. Comput. Math. **1**, 33–36 (2015)
Neal, R.M.: MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo **2**, 2 (2011)
Nobile, F., Tempone, R., Webster, C.: A sparse grid stochastic collocation method for partial differential equations with random input data. SIAM J. Numer. Anal. **46**, 2309–2345 (2008)
Schwab, C., Todor, R.A.: Karhunen loève approximation of random fields by generalized fast multipole methods. J. Comput. Phys. **217**(1), 100–122 (2006)
Sirovich, L.: Turbulence and the dynamics of coherent structures. I. Coherent structures. Q. Appl. Math. **45**(3), 561–571 (1987)
Strathmann, H., Sejdinovic, D., Livingstone, S., Szabo, Z., Gretton, A.: Gradient-free Hamiltonian Monte Carlo with efficient kernel exponential families. In: Advances in Neural Information Processing Systems, pp 955–963 (2015)
Stuart, A.: Inverse problems: a Bayesian perspective. Acta Numer. **19**, 451–559 (2010)
Wald, I., Havran, V.: On building fast kd-trees for ray tracing, and on doing that in O (N log N). In: 2006 IEEE Symposium on Interactive Ray Tracing, IEEE, pp 61–69 (2006)
Wan, J., Zabaras, N.: A probabilistic graphical model approach to stochastic multiscale partial differential equations. J. Comput. Phys. **250**, 477–510 (2013)
Xiu, D., Karniadakis, G.: Modeling uncertainty in flow simulations via generalized polynomial chaos. J. Comput. Phys. **187**, 137–167 (2003)
Zhang, C., Shahbaba, B., Zhao, H.: Hamiltonian Monte Carlo acceleration using surrogate functions with random bases. Stat. Comput. **27**(6), 1473–1490 (2017)
Zhang, C., Shahbaba, B., Zhao, H.: Precomputing strategy for Hamiltonian Monte Carlo method base on regularity in parameter space. Comput. Stat. **32**, 253–279 (2017)
Zhang, C., Shahbaba, B., Zhao, H.: Variational Hamiltonian Monte Carlo via score matching. Bayesian Anal. **13**(2), 485–506 (2018)
Zhang, Z., Ci, M., Hou, T.Y.: A multiscale data-driven stochastic method for elliptic PDEs with random coefficients. SIAM Multiscale Model. Simul. **13**, 173–204 (2015)