Reduced-order model-based variational inference with normalizing flows for Bayesian elliptic inverse problems

Zhizhang Wu[a], Cheng Zhang[b,*], Zhiwen Zhang[a,*]

[a]*Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China.*
[b]*School of Mathematical Sciences and Center for Statistical Science, Peking University, Beijing, China.*

Abstract

We propose a reduced-order model-based variational inference method with normalizing flows for Bayesian elliptic inverse problems. The coefficient of the elliptic PDE is represented by a finite number of parameters. We aim to estimate the posterior distributions of these parameters in the framework of variational inference, and the approximation of the posterior distribution is constructed through a normalizing flow. Moreover, as data of inputs and outputs of the forward problem are accumulated during the training of the normalizing flow, we can naturally exploit the low-dimensional intrinsic structure of the forward elliptic PDE using reduced-order models based on these data, in which information of the true posterior is incorporated. We construct a low-dimensional set of data-driven basis functions in the solution space using proper orthogonal decomposition and train a neural network that maps the parameters to the coefficients of these data-driven basis functions. Then the surrogate forward map, which is the combination of the reduced-order model and the parameter-to-coefficient neural network, significantly accelerates the inversion, as repetitive evaluations of the forward problem are needed in the training of the normalizing flow. We present numerical examples to show the accuracy and efficiency of the proposed method.

Keyword: Bayesian elliptic inverse problems; variational inference; normalizing flows; reduced-order models; proper orthogonal decomposition.

AMS subject classifications. 35R30, 35R60, 62F15, 68T07, 78M34.

1. Introduction

Inverse problems are frequently encountered in science and engineering communities, in which we need to infer problem-specific parameters or inputs from indirect and noisy observations. However, inverse problems are often nonlinear (even if the forward problems are linear) and ill-posed (or unstable) in the sense that either the existence and uniqueness of the solution to the inverse problem are not guaranteed or the dependence of the parameters

*Corresponding author

Email addresses: wuzz@hku.hk (Zhizhang Wu), chengzhang@math.pku.edu.cn (Cheng Zhang), zhangzw@hku.hk (Zhiwen Zhang)

on the data (and noise) is sensitive. As a result, pointwise estimates of the parameters may be erroneous and misleading, and additional regularization is often required. On the other hand, the Bayesian approach provides an alternative to inverse problems [21, 28, 10, 27, 7, 23]. In the Bayesian framework, we aim to infer the posterior distribution of the unknowns conditioned on observations, where regularization is naturally imposed in the form of an appropriate prior distribution. Therefore, Bayesian inversion provides a principled way of uncertainty quantification in the presence of data and noise. We refer interested readers to [35] for a comprehensive review of the Bayesian approach to inverse problems.

As the posterior is generally intractable due to the complexity of the system, one often resorts to computational approximation approaches. Two classical approximation schemes are the Markov chain Monte Carlo (MCMC) method and the variational inference method. In a typical MCMC method, samples from the posterior distribution are generated by simulating a Markov chain whose stationary distribution is the desired posterior distribution. Although MCMC asymptotically converges to the posterior, its efficiency heavily depends on a good balance among many factors, e.g., its computational cost, acceptance probability, and mixing property, making it challenging to solve complicated and large systems in practice using MCMC algorithms.

Variational inference (VI) [20, 38, 8] is an alternative approximate Bayesian inference method that is growing in popularity. Unlike MCMC methods, VI seeks the best approximation to the true posterior from a family of tractable distributions. By transforming the inference problem into an optimization problem, VI tends to be faster and easier to scale to large data and complicated models [8]. In recent years, many efforts have been made to apply VI to solve Bayesian inverse problems [36, 3, 32]. However, VI can also introduce a large bias if the variational family of distributions is insufficiently flexible [26, 40]. The success of variational methods, therefore, largely relies on the construction of appropriate tractable variational family of distributions and efficient optimization procedures [15].

A powerful framework for building flexible approximating distributions is normalizing flows (NFs) [33, 11, 22, 31]. Starting from a simple base distribution with a tractable probability density function, NFs apply a sequence of invertible transformations, often parameterized by neural networks, to obtain a more flexible distribution. These flow-based approximating distributions enjoy many advantages such as efficient sampling, exact likelihood evaluation, and low-variance Monte Carlo gradient estimates when the base distribution is reparameterizible, making them ideal for variational inference. While efficient, current NFs rely on stochastic optimization for training that requires repetitive evaluations of the target posterior and gradients. 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, but the solution to an elliptic PDE is also involved in generating the data and evaluating the posterior in variational inference with normalizing flows. These computational challenges make normalizing flow based variational inference methods costly to use for Bayesian inverse problems [17].

To break this computational bottleneck, we resort to model reduction [5, 6, 18], which

aims to construct an intrinsic low-dimensional surrogate model that is both accurate and efficient. In particular, we consider the proper orthogonal decomposition (POD) method [37], one of the most commonly used reduced-order model basis generation methods that is widely applied to inverse problems [9, 12, 24] and optimization problems [16, 41]. In inverse problems and optimization problems, the first few steps of the iteration provides a natural collection of data of inputs and outputs of the forward problem. Then the POD method takes these solution data as input data and returns an ordered set of orthogonal basis functions that best approximates the input data in the least square sense. We then obtain a reduced-order model by truncating the basis functions. Provided that the forward problem has an intrinsic low-dimensional structure, a small number of basis functions are sufficient to accurately approximate the solution to the forward problem. In other words, a low-dimensional reduced-order model can be constructed to solve the forward problem both accurately and efficiently. Therefore, we can greatly enhance the efficiency of solving the inverse problems and the optimization problems by applying this fast and accurate reduced-order model to the repetitive evaluation of the forward problem in the subsequent iteration.

In this work, we consider the variational inference with normalizing flows for Bayesian elliptic inverse problems. We follow the idea of reduced-order models and adopt a reduced-order model-based and data-driven strategy proposed in [24, 25], which can significantly reduce the computational cost of training normalizing flows. First, the data of inputs and outputs of the forward problem are naturally collected in the first few steps of training the normalizing flow. Then, a set of data-driven basis functions are constructed using the POD method based on these data to achieve significant dimension reduction in the solution space. Since the data that are used for constructing the data-driven basis functions are collected in the process of training the approximate posterior, information of the true posterior is incorporated in these data. Moreover, the solution space of the elliptic equation has an intrinsic low-dimensional structure [4, 24, 25]. Thus, we can construct a low-dimensional reduced-order model that solves the forward elliptic PDE both accurately and efficiently, where the solution of the elliptic problem is a linear combination of the POD basis functions. To further enhance efficiency, we use the same solution data to train a neural network that maps the parameters to the coefficients of the POD basis functions. With the reduced-order model and the parameter-to-coefficient neural network, we are able to construct a fast and accurate surrogate forward map. Finally, we apply this surrogate forward map to the repetitive evaluation of the forward elliptic problem in the subsequent steps of training the normalizing flow. In this way, this reduced-order model-based strategy can significantly reduce the computational cost of training normalizing flows for our Bayesian inverse problem.

The rest of the paper is organized as follows. We first describe the setting of the forward model problems and the Bayesian inversion problems in Section 2. Then, we present the reduced-order model-based variational inference with normalizing flows for solving the Bayesian elliptic inverse problems in Section 3, where the variational inference, the normalizing flow, the model-based and data-driven dimension reduction, and implementation details of the proposed methods will be discussed. We present numerical experiments to study the

accuracy and efficiency of the proposed methods in Section 4. Concluding remarks are made in Section 5.

2. Setting of the problems

2.1. Forward problems

In this paper, we consider a classical inverse problem where we infer the diffusion coefficient in an elliptic PDE, which is commonly used for modeling isothermal steady flow in porous media, hydrology and reservoir simulation. For illustrative purposes, we consider the following elliptic PDEs with random coefficients $a(\mathbf{x}, \omega)$:

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

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

where $D \in \mathbb{R}^d$ is a bounded spatial domain, Ω is a sample space and $f \in L^2(D)$. We assume that $a(\mathbf{x}, \omega)$ is almost surely uniformly elliptic, i.e., there exist $a_{\min}, a_{\max} > 0$ such that

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

In general, we assume that $a(\mathbf{x}, \omega)$ takes a parametric form. For instance, we may parameterize $a(\mathbf{x}, \omega)$ as

$$\log a(\mathbf{x}, \omega) = \bar{a}(\mathbf{x}) + \sum_{j=1}^{r} a_j(\mathbf{x})\xi_j(\omega), \tag{4}$$

where $\xi_m(\omega)$'s are random variables and $\bar{a}(\mathbf{x})$, $a_m(\mathbf{x})$'s are spatial basis functions. 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 forward map

$$\begin{aligned} &\mathcal{F} : \mathcal{W} \subset \mathbb{R}^r \to \mathcal{U} = H_0^1(\Omega) : \\ &\boldsymbol{\xi} = (\xi_1, \ldots, \xi_r)^T \mapsto u(\mathbf{x}, \omega) = u(\mathbf{x}, \boldsymbol{\xi}(\omega)), \end{aligned} \tag{5}$$

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

The past few decades have witnessed significant development of efficient numerical methods for solving elliptic PDEs with random coefficients; see e.g. [14, 2, 30, 1] and the references therein. One can quantify the uncertainty in the stochastic problem by solving the forward problem of random elliptic PDEs. A more challenging case is that the random elliptic PDEs involve multiscale features and/or the dimension of random inputs is high, which leads to high computational costs. In recent years, some data-driven methods and multiscale model reduction methods were developed to solve multiscale elliptic PDEs with random coefficients. We refer the interested readers to [39, 1, 42, 19, 25] and the references therein for related works.

2.2. Bayesian inverse problems

While the forward problem of the elliptic PDE problem (1)–(2) assigns an input $\boldsymbol{\xi} \in \mathcal{W}$ to an output $u \in \mathcal{U}$, we consider in the work the inverse problem of (1)–(2), in which we would like 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 measured at a finite number of discrete locations with noise, which is the measurement data denoted by $\mathbf{y} \in \mathbb{R}^m$ in the form of

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

where the operator $\mathcal{G} : \mathbb{R}^r \to \mathbb{R}^m$ is the 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 framework of Bayesian inversion, the parameter $\boldsymbol{\xi}$ is treated as a random variable (vector) with a prior distribution $p_{\boldsymbol{\xi}}(\boldsymbol{\xi})$. Equation (6) and the distribution of $\boldsymbol{\eta}$ together give the likelihood $p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi})$. For simplicity and concreteness, we assume in this paper that $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_m)$ is a zero-mean Gaussian with diagonal covariance $\sigma^2 \boldsymbol{I}_m$, so that

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

According to the Bayes' rule, the posterior distribution of $\boldsymbol{\xi}$ conditioned on the data $\mathbf{y}$ is given by the following formula:

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

Then, the Bayesian inversion can be performed by estimating the posterior via, e.g., the MCMC method.

As mentioned in Section 1, it is challenging to use MCMC type methods for solving the Bayesian inverse problem of complicated and large systems. Moreover, the complicated forward problem (1)–(2) poses another challenge for the Bayesian elliptic inverse problem. 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)–(2) for each random coefficient to compute the likelihood function (7) whenever a new $\boldsymbol{\xi}$ comes at hand, which forms the computational bottleneck for the Bayesian elliptic inverse problem. To address these challenges, we propose a reduced-order model-based variational inference with normalizing flows to obtain an accurate approximation of the posterior of Bayesian elliptic inverse problems while keeping the overall computational cost low at the same time.

3. Reduced-order model-based variational inference with normalizing flows

In the following, we use the notational convention that $(\boldsymbol{v})_j$ denotes the j-th element of $\boldsymbol{v}$ for any vector $\boldsymbol{v}$.

3.1. Variational inference

In the framework of variational inference [33], we approximate the posterior distribution $p_{\boldsymbol{\xi}|\mathbf{y}}(\boldsymbol{\xi}|\mathbf{y})$ by $q_{\boldsymbol{\theta}}(\boldsymbol{\xi})$, where $\boldsymbol{\theta}$ denotes the trainable parameters in the variational approximation. In the following, we drop the subscripts of $p_{\boldsymbol{\xi}|\mathbf{y}}, p_{\boldsymbol{\xi}}$ for brevity of notation. Naturally, we want to construct $q_{\boldsymbol{\theta}}(\boldsymbol{\xi})$ by minimizing the KL-divergence $\mathbb{D}_{\mathrm{KL}}\big[q_{\boldsymbol{\theta}}(\boldsymbol{\xi})||p(\boldsymbol{\xi}|\mathbf{y})\big]$. However, as the exact posterior $p(\boldsymbol{\xi}|\mathbf{y})$ is unknown, we cannot compute the KL-divergence directly. Instead, we can derive that

$$\mathbb{D}_{\mathrm{KL}}\big[q_{\boldsymbol{\theta}}(\boldsymbol{\xi})||p(\boldsymbol{\xi}|\mathbf{y})\big] = \log p(\mathbf{y}) + \mathscr{F}(\boldsymbol{\theta};\mathbf{y}), \tag{9}$$

where

$$\begin{aligned}\mathscr{F}(\boldsymbol{\theta};\mathbf{y}) &= \mathbb{E}_{q_{\boldsymbol{\theta}}}\big[\log q_{\boldsymbol{\theta}}(\boldsymbol{\xi})\big] - \mathbb{E}_{q_{\boldsymbol{\theta}}}\big[\log p(\mathbf{y},\boldsymbol{\xi})\big] \\ &= \mathbb{E}_{q_{\boldsymbol{\theta}}}\big[\log q_{\boldsymbol{\theta}}(\boldsymbol{\xi})\big] - \mathbb{E}_{q_{\boldsymbol{\theta}}}\big[\log p(\mathbf{y}|\boldsymbol{\xi})\big] - \mathbb{E}_{q_{\boldsymbol{\theta}}}\big[\log p(\boldsymbol{\xi})\big],\end{aligned} \tag{10}$$

and $-\mathscr{F}(\boldsymbol{\theta};\mathbf{y})$ is called the evidence lower bound (ELBO). In the following, we shall drop the dependence of $\mathscr{F}(\boldsymbol{\theta};\mathbf{y})$ on $\mathbf{y}$ for notational brevity. Since $\log p(\mathbf{y})$ is fixed w.r.t. $q_{\boldsymbol{\theta}}(\boldsymbol{\xi})$, minimizing $\mathbb{D}_{\mathrm{KL}}[q_{\boldsymbol{\theta}}(\boldsymbol{\xi})||p(\boldsymbol{\xi}|\mathbf{y})]$ is equivalent to minimizing $\mathscr{F}(\boldsymbol{\theta})$. Moreover, we can minimize $\mathscr{F}(\boldsymbol{\theta})$ using the stochastic backpropagation method.

3.2. Normalizing flows

We construct the approximation $q_{\boldsymbol{\theta}}(\boldsymbol{\xi})$ using the normalizing flow approach, which refers to the transformation of a probability density through a sequence of invertible mappings [33]. We consider the finite flow, which means the sequence of invertible mappings is of finite length. We obtain $\boldsymbol{\xi}_K$ by transforming $\boldsymbol{\xi}_0$ with initial distribution q_0 through K invertible mapping f_k, $k = 1, \ldots, K$, i.e.,

$$\boldsymbol{\xi}_K = f_K \circ \cdots \circ f_1(\boldsymbol{\xi}_0). \tag{11}$$

Let $q_K(\boldsymbol{\xi}_K)$ be the probability density of $\boldsymbol{\xi}_K$. Then

$$\log q_K(\boldsymbol{\xi}_K) = \log q_0 - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}} \right|, \tag{12}$$

where $\det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}}$ is the determinant of the Jacobian matrix of f_k w.r.t. $\boldsymbol{\xi}_{k-1}$, $k = 1, \ldots, K$. The law of the unconscious statistician (LOTUS) indicates that any expectation $\mathbb{E}_{q_K}[g(\boldsymbol{\xi})]$ can be written as an expectation under q_0 as follows:

$$\mathbb{E}_{q_K}\big[g(\boldsymbol{\xi})\big] = \mathbb{E}_{q_0}\big[g(f_K \circ \cdots \circ f_1(\boldsymbol{\xi}_0))\big]. \tag{13}$$

In practice, we usually take the initial density $q_0 \sim \mathcal{N}\big(\boldsymbol{\mu}_0, \mathrm{diag}(\boldsymbol{\sigma}_0)\big)$ and construct $q_0(\boldsymbol{\xi}_0)$ by the following formula

$$\boldsymbol{\xi}_0 = f_0(\hat{\boldsymbol{\xi}}_0) = \boldsymbol{\sigma}_0 \odot \hat{\boldsymbol{\xi}}_0 + \boldsymbol{\mu}_0,$$

where $\boldsymbol{\sigma}_0, \boldsymbol{\mu}_0 \in \mathbb{R}^r$, $\odot$ is the elementwise multiplication, and $\hat{\boldsymbol{\xi}}_0 \sim \hat{q}_0$, with $\hat{q}_0$ the density of standard Gaussian distribution. Then $\log q_0 = \log \hat{q}_0 - \sum_{j=1}^r \log |(\boldsymbol{\sigma}_0)_j|$.

We take $\boldsymbol{\xi} := \boldsymbol{\xi}_K$ and $q_{\boldsymbol{\theta}}(\boldsymbol{\xi}) := q_K(\boldsymbol{\xi}_K)$. Then

$$\begin{aligned}\mathscr{F}(\boldsymbol{\theta}) =& \mathbb{E}_{\hat{q}_0}[\log \hat{q}_0] - \mathbb{E}_{\hat{q}_0}\left[\sum_{j=1}^r \log |(\boldsymbol{\sigma}_0)_j|\right] - \mathbb{E}_{\hat{q}_0}\left[\sum_{k=1}^K \log \left|\det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}}\right|\right] \\ & - \mathbb{E}_{\hat{q}_0}[\log p(\mathbf{y}|\boldsymbol{\xi}_K)] - \mathbb{E}_{\hat{q}_0}[\log p(\boldsymbol{\xi}_K)], \end{aligned} \tag{14}$$

where $\boldsymbol{\theta} = \{\boldsymbol{\sigma}_0, \boldsymbol{\mu}_0, \text{all trainable parameters in } f_k, k = 1, \ldots, K\}$. The expectation $\mathbb{E}_{\hat{q}_0}$ in (14) is computed using the mini-batch Monte Carlo estimation. Hence the gradient of $\mathscr{F}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ can be computed via

$$\begin{aligned}\nabla_{\boldsymbol{\theta}}\mathscr{F}(\boldsymbol{\theta}) =& \mathbb{E}_{\hat{q}_0}[\nabla_{\boldsymbol{\theta}} \log \hat{q}_0] - \mathbb{E}_{\hat{q}_0}\left[\sum_{j=1}^r \nabla_{\boldsymbol{\theta}} \log |(\boldsymbol{\sigma}_0)_j|\right] - \mathbb{E}_{\hat{q}_0}\left[\sum_{k=1}^K \nabla_{\boldsymbol{\theta}} \log \left|\det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}}\right|\right] \\ & - \mathbb{E}_{\hat{q}_0}[\nabla_{\boldsymbol{\theta}} \log p(\mathbf{y}|\boldsymbol{\xi}_K)] - \mathbb{E}_{\hat{q}_0}[\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\xi}_K)]. \end{aligned} \tag{15}$$

In the following, we introduce three useful types of normalizing flows that will be used in this paper.

Planar flows. We first introduce the planar flow [33]. Let

$$f_k(\boldsymbol{\xi}) = \boldsymbol{\xi} + \mathbf{u}_k h(\mathbf{w}_k^T \boldsymbol{\xi} + b_k), \tag{16}$$

where $\mathbf{u}_k, \mathbf{w}_k \in \mathbb{R}^m, b_k \in \mathbb{R}$ and h is an element-wise nonlinear function. Let

$$\psi_k(\boldsymbol{\xi}) = h'(\mathbf{w}_k^T \boldsymbol{\xi} + b_k)\mathbf{w}_k. \tag{17}$$

Then

$$\log \left|\det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}}\right| = \left|1 + \mathbf{u}_k^T \psi_k(\boldsymbol{\xi}_{k-1})\right|. \tag{18}$$

In our numerical experiments, h will be chosen to be tanh nonlinearity.

Real-valued non-volume preserving transformations. We can also consider the real-valued non-volume preserving (RealNVP) transformations [11]. At each layer k of RealNVP, the invertible mapping f_k is constructed as follows. Let $S \subset \{1, \ldots, r\}$ and $S^c = \{1, \ldots, r\} \backslash S$. Then

$$(\boldsymbol{\xi}_k)_{S^c} = (\boldsymbol{\xi}_{k-1})_{S^c}, \tag{19}$$

$$(\boldsymbol{\xi}_k)_S = (\boldsymbol{\xi}_{k-1})_S \odot \exp\left(\boldsymbol{\beta}((\boldsymbol{\xi}_{k-1})_{S^c})\right) + \boldsymbol{\alpha}((\boldsymbol{\xi}_{k-1})_{S^c}), \quad j \in S, \tag{20}$$

where $(\boldsymbol{\xi})_S$ is the entries of $\boldsymbol{\xi}$ with indices in S, and $\boldsymbol{\alpha} : \mathbb{R}^{r-|S^c|} \to \mathbb{R}^{|S|}$, $\boldsymbol{\beta} : \mathbb{R}^{r-|S^c|} \to \mathbb{R}^{|S|}$ are neural networks. The logarithm of the determinant of the Jacobian matrix f_k is

$$\log \left|\det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}}\right| = \sum_j (\boldsymbol{\beta})_j. \tag{21}$$

In our numerical experiments, the network structure of $\boldsymbol{\alpha}$ is Linear $\to$ ReLU $\to$ Linear, and the network structure of $\boldsymbol{\beta}$ is Linear $\to$ tanh $\to$ Linear $\to$ log sigmoid. S and S^c are partitioned as sets of odd and even numbers at the first layer, respectively. Through each layer of the flow, S and S^c are interchanged alternatively.

Inverse autoregressive flows. We will also study the inverse autoregressive flows (IAF) [22]. We first describe the autoregressive structure. Consider a series $\mathbf{z} = \{z_j\}_{j=1}^J$. Let $\mathbf{f}(\mathbf{z}) = (f_1, \ldots, f_J)$ be a vector-valued autoregressive function of $\mathbf{z}$. The autoregressive structure of $\mathbf{f}$ means that $\partial f_j/\partial z_i = 0$ for $i \geq j$. As to IAF, at each layer, f_k is constructed in the following way:

$$\boldsymbol{\alpha}, \boldsymbol{\beta} \leftarrow \text{autoregressiveNN}(\boldsymbol{\xi}_{k-1}), \tag{22}$$

$$\boldsymbol{\gamma} = \text{sigmoid}(\boldsymbol{\beta}), \tag{23}$$

$$\boldsymbol{\xi}_k = \boldsymbol{\gamma} \odot \boldsymbol{\xi}_{k-1} + (1 - \boldsymbol{\gamma}) \odot \boldsymbol{\alpha}, \tag{24}$$

where $\boldsymbol{\alpha} : \mathbb{R}^r \to \mathbb{R}^r$ and $\boldsymbol{\beta} : \mathbb{R}^r \to \mathbb{R}^r$ are autoregressive neural networks. The logarithm of the determinant of the Jacobian matrix f_k is

$$\log \left| \det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}} \right| = \sum_j \log(\boldsymbol{\gamma})_j. \tag{25}$$

In our numerical experiments, the structure of the autoregressive neural network [13] is Masked Linear $\to$ ReLU $\to$ Masked Linear. Also, note that the order of the input of the autoregressive neural network plays a role. Hence in our experiments, the order of indices is taken to be $1 \to \cdots \to r$ at the first layer and it is reversed through each layer of the flow.

3.3. A reduced-order model-based surrogate forward map

At each optimization step, we need to evaluate the likelihood $p(\mathbf{y}|\boldsymbol{\xi}_K)$ for N times, where N is the mini-batch size in the Monte Carlo estimation of $\mathscr{F}$. In other words, we need to solve the forward problem for N times at each optimization step. Therefore, it will significantly enhance the efficiency of the Bayesian inversion if we can accelerate the solution of the forward problem. In this paper, we adopt the reduced-order model-based and data-driven strategy proposed in [24, 25], which is constructing a surrogate forward map by using POD for model reduction and then training a parameter-to-coefficient map using a neural network. We refer interested readers to [5, 37] for more details of the POD method and to [25] for other types of constructions of the parameter-to-solution map. To make this paper self-contained, we introduce the main idea of this strategy in this subsection.

Assume that we use the full-order model to solve the forward problem for the first M_1 optimization steps. Then we have NM_1 pairs of data $\{(\boldsymbol{\xi}_K^j, u(\mathbf{x}, \boldsymbol{\xi}_K^j))\}_{j=1}^{NM_1}$, where $u(\mathbf{x}, \boldsymbol{\xi}_K^j)) = \mathcal{F}(\boldsymbol{\xi}_K^j)$. With $\{u(\mathbf{x}, \boldsymbol{\xi}_K^j)\}_{j=1}^{NM_1}$ the snapshot ensemble, the POD method can extract a set of orthogonal basis functions $\{\psi_j(\mathbf{x})\}_{j=1}^m$ that optimally approximate the snapshot ensemble

in the L^2 sense. To be concrete, $\{\psi_j(\mathbf{x})\}_{j=1}^m$ is the solution to the following constrained optimization problem:

$$\min_{\{\psi_j(\mathbf{x})\}_{j=1}^m} \sum_{j=1}^{NM_1} \|u(\cdot,\boldsymbol{\xi}_K^j) - \sum_{i=1}^m < u(\cdot,\boldsymbol{\xi}_K^j), \psi_i(\cdot) >_D \psi_i(\cdot)\|_{L^2(D)}^2, \tag{26}$$

$$\text{s.t.} \quad (\psi_i, \psi_j) = \delta_{ij}, \quad i,j = 1,\ldots,m, \tag{27}$$

where $< \cdot,\cdot >_D$ is the L^2 inner product on $L^2(D)$. To solve (26)–(27), the methods of snapshots [34] computes the correlation matrix A corresponding to the snapshots with entries

$$A_{ij} =< u(\cdot,\boldsymbol{\xi}_K^i), u(\cdot,\boldsymbol{\xi}_K^j) >_D, \quad i,j = 1,\ldots,NM_1.$$

Then A is semi-positive-definite. Let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{NM_1} \geq 0$ be the eigenvalues of A in the descending order and $\boldsymbol{\phi}_1,\ldots,\boldsymbol{\phi}_{NM_1}$ be the corresponding eigenvectors. Then the POD basis can be constructed as

$$\psi_j = \frac{1}{\sqrt{\lambda_j}} \sum_{i=1}^{NM_1} (\boldsymbol{\phi}_j)_i u(\cdot,\boldsymbol{\xi}_K^i), \tag{28}$$

where $j = 1,\ldots,m$ and $\lambda_j > 0$. Moreover, the approximation error is

$$\sum_{j=1}^{NM_1} \|u(\cdot,\boldsymbol{\xi}_K^j) - \sum_{i=1}^m < u(\cdot,\boldsymbol{\xi}_K^j), \psi_i(\cdot) >_D \psi_i(\cdot)\|_{L^2(D)}^2 = \sum_{j=m+1}^{NM_1} \lambda_j. \tag{29}$$

Since the solution space of the elliptic problem has an intrinsic low-dimensional structure [4, 24, 25], the eigenvalues $\{\lambda_j\}_{j=1}^{NM_1}$ decay fast, and thus we can choose a small m, with $m \ll NM_1$, so that the m POD basis functions corresponding to the first m eigenvalues are sufficient to construct an accurate reduced-order model. In our numerical experiments, we pre-choose a small fixed number m when constructing the reduced-order model. Note that $\{\boldsymbol{\xi}_K^j\}_{j=1}^{NM_1}$ are sampled from the normalizing flow in the first M_1 steps and the normalizing flow starts to approximate the true posterior in the first M_1 steps of the training. Hence the data set $\{\boldsymbol{\xi}_K^j\}_{j=1}^{NM_1}$ provides information of the true posterior. Therefore, we can expect that the reduced-order model constructed from $\{(\boldsymbol{\xi}_K^j, u(\mathbf{x},\boldsymbol{\xi}_K^j))\}_{j=1}^{NM_1}$ is accurate enough for the repetitive evaluation of the forward problem in the Bayesian inversion.

With the constructed basis functions $\{\psi_j(\mathbf{x})\}_{j=1}^m$, each snapshot $u(\mathbf{x},\boldsymbol{\xi}_K^j)$ can be approximated by

$$u(\mathbf{x},\boldsymbol{\xi}_K^j) \approx \widetilde{u}(\mathbf{x},\boldsymbol{\xi}_K^j) := \sum_{i=1}^m c_i(\boldsymbol{\xi}_K^j)\psi_i(\mathbf{x}), \tag{30}$$

where

$$c_i(\boldsymbol{\xi}_K^j) =< u(\cdot,\boldsymbol{\xi}_K^j), \psi_i(\cdot) >_D . \tag{31}$$

Let $\mathbf{c}^j = (c_1(\boldsymbol{\xi}_K^j), \ldots, c_m(\boldsymbol{\xi}_K^j))^T$. Then, we can train a neural network $\boldsymbol{\varphi} : \boldsymbol{\xi} \mapsto \mathbf{c}$ with input data $\{\boldsymbol{\xi}_K^j\}_{j=1}^{NM_1}$ and output data $\{\mathbf{c}^j\}_{j=1}^{NM_1}$. Given any $\boldsymbol{\xi} \in \mathbb{R}^r$, let $\mathbf{c}_{\boldsymbol{\varphi}}(\boldsymbol{\xi}) = \boldsymbol{\varphi}(\boldsymbol{\xi})$. Then the reduced-order model-based surrogate forward map $\widetilde{\mathcal{F}}$ can be constructed as

$$\widetilde{\mathcal{F}}(\boldsymbol{\xi}) = \sum_{j=1}^{m} (\mathbf{c}_{\boldsymbol{\varphi}}(\boldsymbol{\xi}))_j \psi_j. \tag{32}$$

Let $\widetilde{\mathcal{G}}$ be the composition of $\widetilde{\mathcal{F}}$ and the discretized observation operator. We can obtain the solution at the observation points using $\widetilde{\mathcal{G}}$. In our numerical experiments, we choose the neural network $\boldsymbol{\varphi}$ to be a fully connected neural network.

3.4. Algorithm

The procedure of our method can be summarized in the following Algorithm 1.

Algorithm 1 Reduced-order model-based variational inference with normalizing flows

Input: obervations $\mathbf{y}$, prior distribution $p(\boldsymbol{\xi})$, mini-batch size N.
for $i = 1 : M_1$ **do**
  Draw $\{\hat{\boldsymbol{\xi}}_0^{i,n}\}_{n=1}^N \sim \hat{q}_0$.
  Compute $\boldsymbol{\xi}_K^{i,n} = f_K \circ \cdots \circ f_1 \circ f_0(\hat{\boldsymbol{\xi}}_0^{i,n})$ for $n = 1, \ldots N$.
  Compute the solution at observation points $\mathcal{G}(\boldsymbol{\xi}_K^{i,n})$ for $n = 1, \ldots N$.
  Compute $\mathscr{F}(\mathbf{y})$ in (14) using Monte Carlo estimation with $\{\mathcal{G}(\boldsymbol{\xi}_K^{i,n})\}_{n=1}^N$, i.e.,

$$\mathscr{F}(\boldsymbol{\theta}) \approx \frac{1}{N} \sum_{n=1}^{N} \left(\log \hat{q}_0(\hat{\boldsymbol{\xi}}_0^{i,n}) - \sum_{j=1}^{r} \log |(\boldsymbol{\sigma}_0)_j| - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial \boldsymbol{\xi}_{k-1}} (\boldsymbol{\xi}_{k-1}^{i,n}) \right| \right.$$
$$\left. - \log p(\mathbf{y}|\boldsymbol{\xi}_K^{i,n}) - \log p(\boldsymbol{\xi}_K^{i,n}) \right)$$

  $\Delta\boldsymbol{\theta}_i \propto -\nabla_{\boldsymbol{\theta}} \mathscr{F}(\mathbf{y})$.
end for
Train a surrogate forward map $\widetilde{\mathcal{F}}$ and then obtain $\widetilde{\mathcal{G}}$ using $\{\boldsymbol{\xi}_K^{i,n}, n = 1, \ldots, N\}_{i=1}^{M_1}$ and $\{\mathcal{F}(\boldsymbol{\xi}_K^{i,n}), n = 1, \ldots, N\}_{i=1}^{M_1}$ based on the reduced-order model and the parameter-to-coefficient map described in Section 3.3.
for $i = M_1 + 1 : M_1 + M_2$ **do**
  Draw $\{\hat{\boldsymbol{\xi}}_0^{i,n}\}_{n=1}^N \sim q_0$.
  Compute $\boldsymbol{\xi}_K^{i,n} = f_K \circ \cdots \circ f_1 \circ f_0(\hat{\boldsymbol{\xi}}_0^{i,n})$ for $n = 1, \ldots N$.
  Compute the solution at observation points $\widetilde{\mathcal{G}}(\boldsymbol{\xi}_K^{i,n})$ for $n = 1, \ldots N$.
  Compute $\mathscr{F}(\mathbf{y})$ in (14) using Monte Carlo estimation with $\{\widetilde{\mathcal{G}}(\boldsymbol{\xi}_K^{i,n})\}_{n=1}^N$.
  $\Delta\boldsymbol{\theta}_i \propto -\nabla_{\boldsymbol{\theta}} \mathscr{F}(\mathbf{y})$.
end for
Output: $q_{\boldsymbol{\theta}} := q_K$

4. Numerical Experiments and Results

In this section, we show numerical examples to demonstrate the performance of the proposed method for Bayesian elliptic inverse problems. We consider the following model problem:

$$-\nabla \cdot (a(\mathbf{x},\omega)u(\mathbf{x},\omega)) = 0, \quad \mathbf{x} = (x_1, x_2) \in [0,1]^2, \tag{33}$$

$$\left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{x_1=0} = \left.\frac{\partial u}{\partial \boldsymbol{n}}\right|_{x_1=1} = 0, \quad u|_{x_2=0} = x_1, \quad u|_{x_2=1} = 1 - x_1. \tag{34}$$

A Gaussian prior with zero mean and covariance function

$$c(\mathbf{x},\mathbf{x}') = \sigma_a^2 \exp\left(-\frac{\|x_1 - x_1'\|^2}{2\ell_1^2} - \frac{\|x_2 - x_2'\|^2}{2\ell_2^2}\right) \tag{35}$$

is assumed on $\log a(\mathbf{x},\omega)$, where we choose $\sigma_a = 1$. The logarithm of the diffusion coefficient is approximated by a truncated Karhunen-Loève (KL) expansion

$$\log(a(\mathbf{x},\omega)) = \sum_{j=1}^{r} \xi_j \lambda_j v_j(\mathbf{x}), \tag{36}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_r)^T$ are r i.i.d. Gaussian random variables and $\{\lambda_j, v_j(\mathbf{x}), j = 1, \ldots, r\}$ are the eigenvalues and eigenfunctions of the covariance function (35). We set the true values of $\boldsymbol{\xi}$ to be $\boldsymbol{\xi}^* = (1, \ldots, 1)^T$. We also impose a zero-mean Gaussian prior on $\boldsymbol{\xi}$, i.e., $p(\boldsymbol{\xi}) \sim \mathcal{N}(0, \sigma_{\boldsymbol{\xi}}^2 \boldsymbol{I}_r)$, where we choose $\sigma_{\boldsymbol{\xi}} = 0.5$.

We want to obtain the posterior distributions of $\boldsymbol{\xi}$. For the reference solution, we use the Hamiltonian Monte Carlo (HMC) method [29] since it is asymptotically correct as the number of samples increases. More details of the HMC method will be included in Appendix A. For the HMC method, the number of leap-frog steps is 10 and the time step size is Δt. We run 5 sampling processes in parallel. In each sampling process, we draw 5000 samples, the first 1000 of which are discarded. In total, we collect 20000 samples in the HMC method. For the numerical solution, we use Algorithm 1 with the three types of normalizing flows, i.e., the Planar flow, RealNVP, and IAF. The flow lengths are $K = 32$ for the Planar flow, $K = 10$ for the RealNVP, and $K = 5$ for the IAF, respectively. Recall that the first M_1 training steps in Algorithm 1 are implemented using the full-order model (FOM) for the computation of $\mathscr{F}$ and that the subsequent M_2 training steps in Algorithm 1 are implemented using the reduced-order model (ROM) for the computation of $\mathscr{F}$.

All the computations are performed using the CPU-version Pytorch on a laptop with a CPU of 2.60 GHz and 16GB RAM.

4.1. Homogeneous case

We first consider the homogeneous case where $\ell_1 = \ell_2 = 0.2$ in the covariance function (35), and we choose $r = 20$ in (36). The observation data are generated on a 31×31 uniform

grid with triangular finite element basis functions. Measurements are collected on a 11×11 uniform grid. We list in Table 1 the values of some parameters of Algorithm 1.

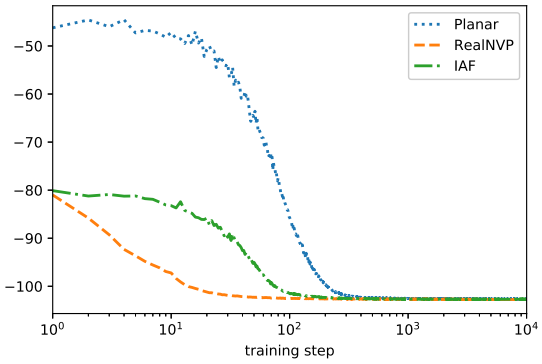
Table 1: Values of some parameters in Algorithm 1 in the homogeneous case

parameters	values
Number of POD basis functions m in (30)	20
mini-batch size N	1000
Number of training steps using FOM M_1	10
Number of training steps using ROM M_2	10000
learning rate	0.001

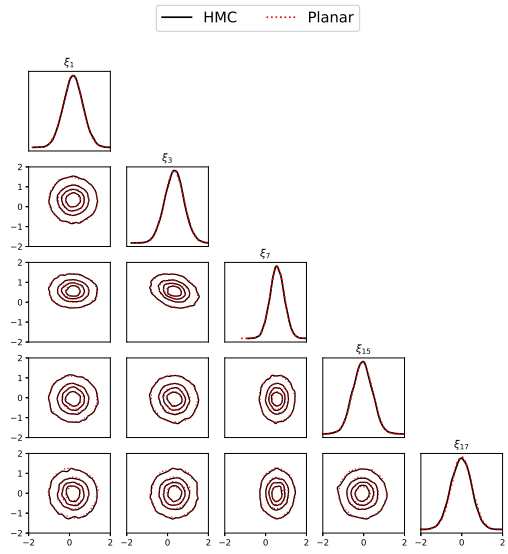
Example 4.1. We choose the noise level $\sigma = 0.1$ in observation data (7). In this case, the time step size for HMC is chosen to be $\Delta t = 0.16$. We show in Table 2 the average training times per training step using FOM and ROM, respectively. We also show the losses and the marginal posterior distributions of the normalizing flows in Figure 1. We can see from Table 2 that there is a great speed-up of the training when we use the surrogate forward map. Moreover, the results in Figure 1 show that our algorithm works well with all three types of the normalizing flows for this problem in the homogeneous case with $\sigma = 0.1$.

Table 2: Average training times (sec) per training step using FOM and ROM in the homogeneous case with $\sigma = 0.1$

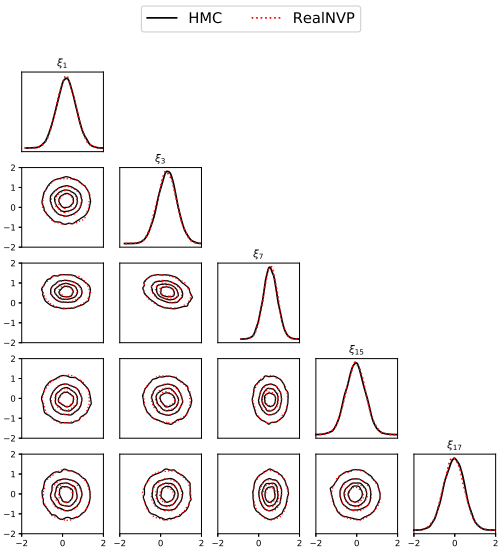
	Planar	RealNVP	IAF
FOM	98.36	98.37	99.67
ROM	0.04100	0.02762	0.02465



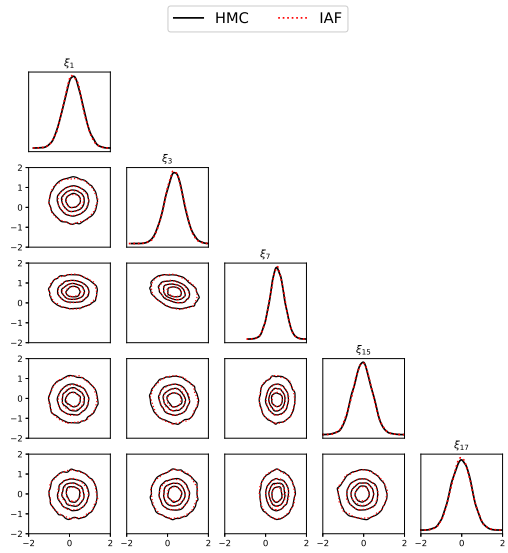
(a) Losses



(b) Marginal posterior distributions of Planar



(c) Marginal posterior distributions of RealNVP



(d) Marginal posterior distributions of IAF

Figure 1: Losses and marginal posterior distributions of the normalizing flows in the homogeneous case with $\sigma = 0.1$

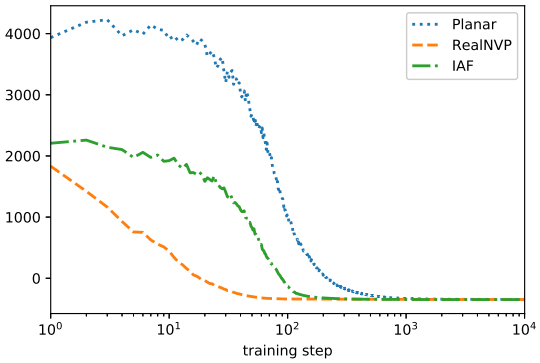
Example 4.2. We then choose the noise level $\sigma = 0.01$ in the observation data (7). In this case, the time step size for HMC is chosen to be $\Delta t = 0.016$. We show in Table 3 the average training times per training step using FOM and ROM, respectively. We also show the losses and the marginal posterior distributions of the normalizing flows in Figure 2. We can still observe a great acceleration of the training when the surrogate forward map is used. On the

other hand, Figure 2 shows that our algorithm also works well with all three types of the normalizing flows for this problem in the homogeneous case with $\sigma = 0.01$, but the results using RealNVP and IAF are slightly better than that using the Planar flow (see e.g. the posterior distributions of ξ_8 and ξ_{15}).

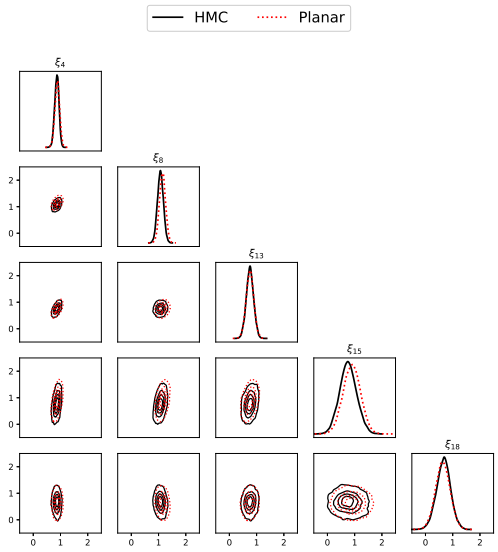
Note that in this case the noise level in the observation data $\sigma = 0.01$ is small and thus the observation data in this case provide more information about $\boldsymbol{\xi}^*$ than those in the case where $\sigma = 0.1$. As a result, the posterior distributions of $\boldsymbol{\xi}$ are more localized around $\boldsymbol{\xi}^*$. So a more accurate forward map for $\boldsymbol{\xi}$ around $\boldsymbol{\xi}^*$ is needed to obtain the posterior distributions of $\boldsymbol{\xi}$ in this case. It can be seen from Figure 2 (a) that the losses of IAF and RealNVP are smaller than that of the Planar flow in the first M_1 steps. Therefore $\{\boldsymbol{\xi}_K^{i,n}, n = 1, \ldots, N\}_{i=1}^{M_1}$, which are the training data collected in the first M_1 steps for the construction of the surrogate forward map, are closer to $\boldsymbol{\xi}^*$ in the training using IAF and RealNVP than in the training using the Planar flow. Thus a more accurate surrogate forward map for $\boldsymbol{\xi}$ around $\boldsymbol{\xi}^*$ is constructed for IAF and RealNVP than for the Planar flow. As a consequence, the results using IAF and RealNVP are slightly better than that using the Planar flow.

Table 3: Average training times (sec) per training step using FOM and ROM in the homogeneous case with $\sigma = 0.01$

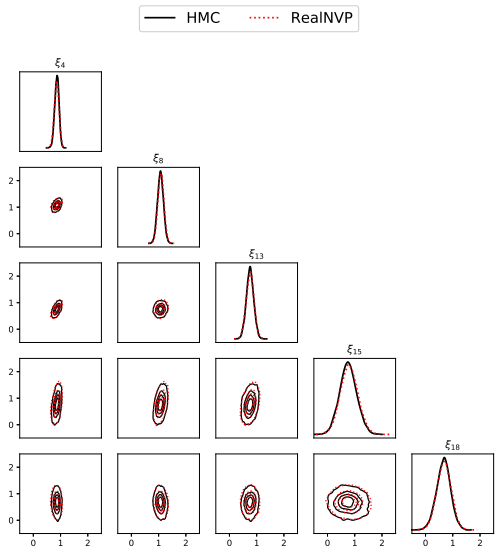
	Planar	RealNVP	IAF
FOM	97.30	99.00	98.15
ROM	0.04137	0.02703	0.02471



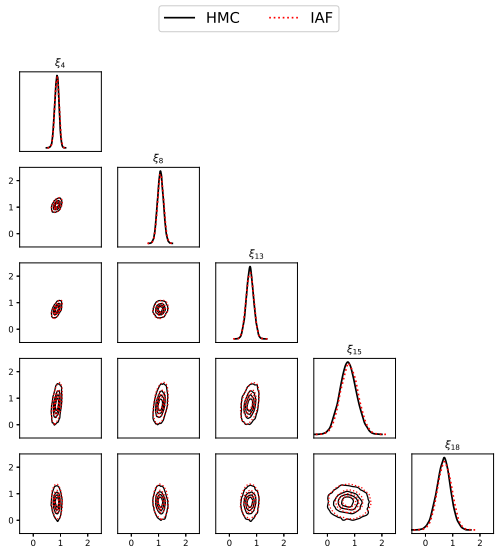
(a) Losses



(b) Marginal posterior distributions of Planar



(c) Marginal posterior distributions of RealNVP



(d) Marginal posterior distributions of IAF

Figure 2: Losses and marginal posterior distributions of the normalizing flows in the homogeneous case with $\sigma = 0.01$

4.2. Heterogeneous case

We then consider the heterogeneous case where $\ell_1 = 0.1$ and $\ell_2 = 0.4$ in (35), and we choose $r = 30$ in (36). The observation data are generated on a 61×61 uniform grid with triangular finite element basis functions. Measurements are collected on a 21×21 uniform grid. We list in Table 4 the values of some parameters in Algorithm 1.

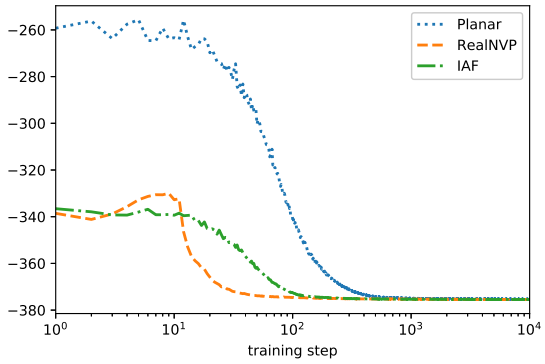
Table 4: Values of some parameters in Algorithm 1 in the heterogeneous case

parameters	values
Number of POD basis functions m in (30)	30
mini-batch size N	1000
Number of training steps using FOM M_1	10
Number of training steps using ROM M_2	10000
learning rate	0.001

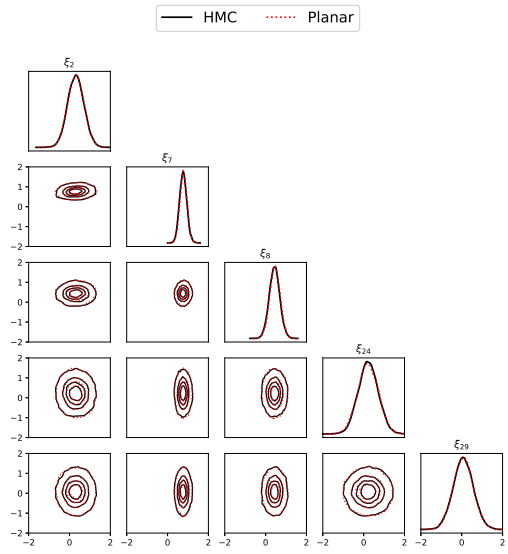
Example 4.3. We choose the noise level $\sigma = 0.1$ in observation data (7). In this case, the time step size for HMC is chosen to be $\Delta t = 0.128$. We show in Table 5 the average training times per training step using FOM and ROM, respectively. We also show the losses and marginal posterior distributions of the normalizing flows in Figure 3. We can still see from Table 5 that the training is greatly accelerated when the surrogate forward map is used. Figure 3 also shows that our algorithm works well with all three types of the normalizing flows for this problem in the heterogeneous case with $\sigma = 0.1$.

Table 5: Average training times (sec) per training step using FOM and ROM in the heterogeneous case with $\sigma = 0.1$

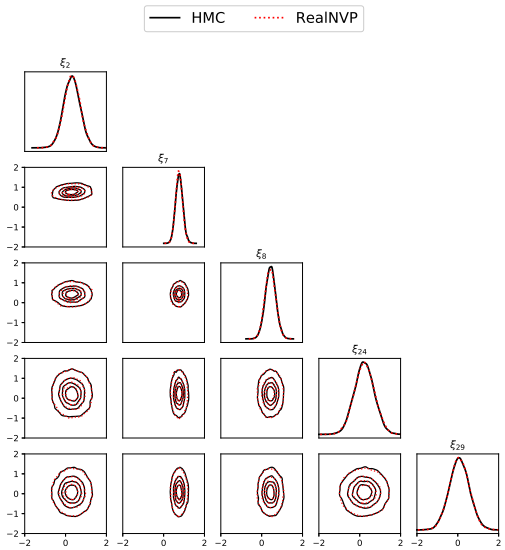
	Planar	RealNVP	IAF
FOM	349.8	350.6	350.7
ROM	0.1067	0.09256	0.08726



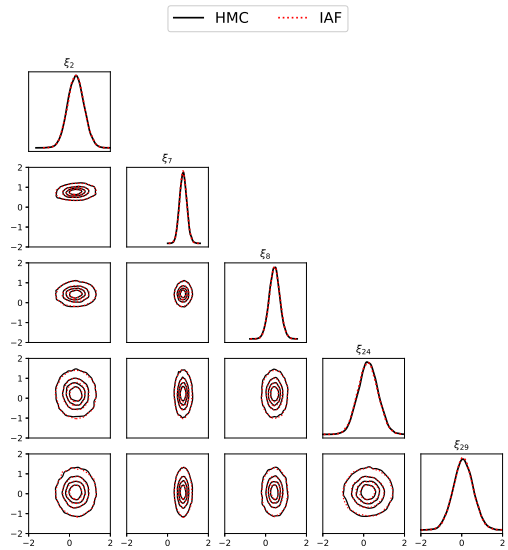
(a) Losses



(b) Marginal posterior distributions of Planar



(c) Marginal posterior distributions of RealNVP



(d) Marginal posterior distributions of IAF

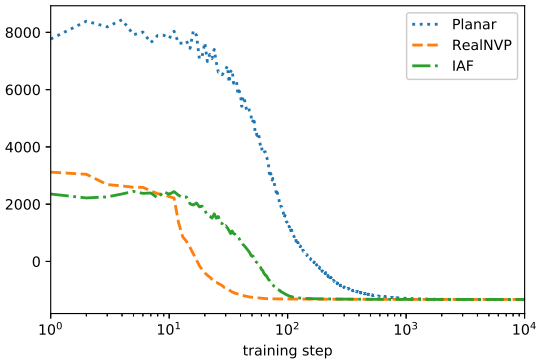
Figure 3: Losses and marginal posterior distributions of the normalizing flows in the heterogeneous case with $\sigma = 0.1$

Example 4.4. We then choose the noise level $\sigma = 0.01$ in the observation data (7). In this case, the time step size for HMC is chosen to be $\Delta t = 0.016$. We show in Table 6 the average training times per step using FOM and ROM, respectively. We also show the losses and the marginal posterior distributions of the normalizing flows in Figure 4. A great speed-up of the training is still observed when using the surrogate forward map. Figure 4 also shows that

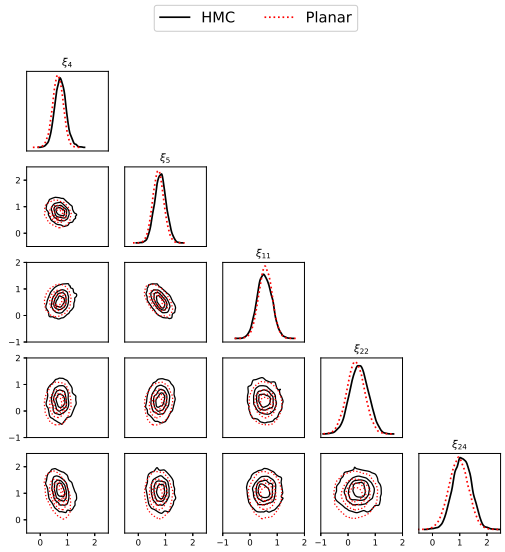
our algorithm works well with all three types of the normalizing flows for this problem in the heterogeneous case with $\sigma = 0.01$, and the results using RealNVP and IAF are still slightly better than that using the Planar flow. The reason is similar to that in the homogeneous case with $\sigma = 0.01$.

Table 6: Average training times (sec) per training step using FOM and ROM in the heterogeneous case with $\sigma = 0.01$

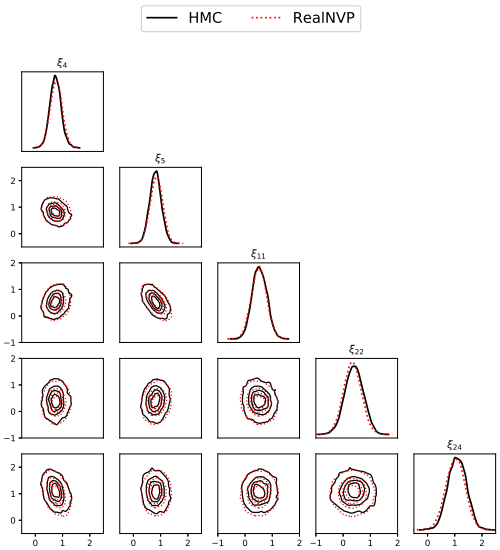
	Planar	RealNVP	IAF
FOM	349.5	348.2	351.3
ROM	0.1069	0.09165	0.08709



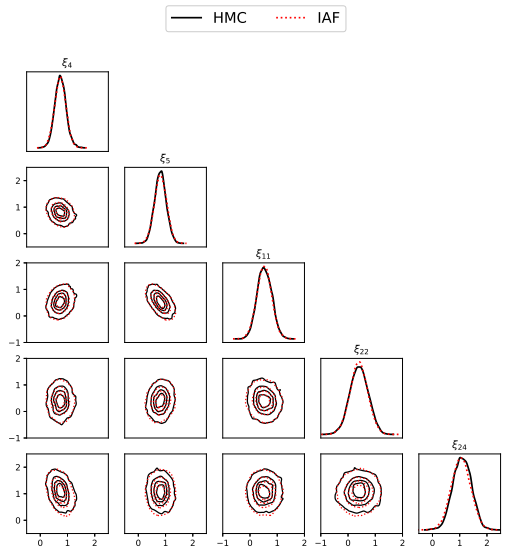
(a) Losses



(b) Marginal posterior distributions of Planar



(c) Marginal posterior distributions of RealNVP



(d) Marginal posterior distributions of IAF

Figure 4: Losses and marginal posterior distributions of the normalizing flows in the heterogeneous case with $\sigma = 0.01$

In summary, the first M_1 training steps of the training provide a natural collection of the solution data that are used for constructing the reduced-order model. Since these solution data are collected in the process of the training of the posterior distributions, some posterior information of the unknowns is incorporated in these solution data, which makes the constructed reduced-order model accurate enough for the Bayesian inverse problem. Moreover,

the training is greatly accelerated once the reduced-order model is used for the computation of the loss function. Our numerical results in the Examples 4.1–4.4 confirm the above advantages of our algorithm.

5. Conclusion

In this paper, we propose a reduced-order model-based variational inference method with normalizing flows for Bayesian elliptic inverse problems. We study the case where the coefficient of the elliptic PDE is represented by a finite number of parameters and we obtain the posterior distributions of these parameters in the framework of variational inference, where the posterior distribution is constructed by normalizing flows.

In our method, the first few steps of training the normalizing flow provide a natural collection of data of inputs and outputs of the forward elliptic problem. We can then construct a fast and accurate reduced-order model using the POD method based on these data since these data provide information of the true posterior and the underlying elliptic problem has a low-dimensional structure, which allows us to achieve significant dimension reduction in approximating the solution space. To further enhance efficiency, we train a parameter-to-coefficient map using a neural network and combine it with the reduced-order model to obtain a surrogate forward map. The surrogate forward map significantly reduces the computational cost in computing the solutions of the elliptic PDEs with random coefficients and thus accelerates the repetitive evaluations of the forward problem that are needed in the training of the normalizing flow. We present numerical results to demonstrate the accuracy and efficiency of the proposed method for solving elliptic Bayesian inverse problems. We find that the the reduced-order model-based surrogate forward map can greatly reduce the computational cost in the training process and the variational inference approach with a normalization flow can accurately compute the posterior distributions of the parameters in the random elliptic PDEs.

Acknowledgements

The research of C. Zhang is supported by 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, the National Natural Science Foundation of China (project 12171406), the Outstanding Young Researcher award of HKU (2020-21), an R&D Funding Scheme from the HKU-SCF FinTech Academy and Seed Funding for Strategic Interdisciplinary Research Scheme 2021/22 (HKU). The computations were performed using research computing facilities offered by Information Technology Services, the University of Hong Kong.

Appendix A. 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 [29] 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

$$\frac{d\boldsymbol{\xi}}{dt} = \frac{\partial H}{\partial \boldsymbol{\zeta}}, \qquad \frac{d\boldsymbol{\zeta}}{dt} = -\frac{\partial H}{\partial \boldsymbol{\xi}}. \tag{A.1}$$

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 (A.1) 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

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

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

$$\alpha_{\text{HMC}} = \min\Big(1, \exp[-H(\boldsymbol{\xi}^*, \boldsymbol{\zeta}^*) + H(\boldsymbol{\xi}, \boldsymbol{\zeta})]\Big). \tag{A.3}$$

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

$$p(\boldsymbol{\xi}, \boldsymbol{\zeta}) \propto \exp\left(-U(\boldsymbol{\xi}) - \frac{1}{2}\boldsymbol{\zeta}^T M^{-1}\boldsymbol{\zeta}\right). \tag{A.4}$$

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 (A.1), and the discretization error in (A.2) 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 the Bayesian elliptic inverse problem, we need repetitive computation of solution $u(\mathbf{x}, \boldsymbol{\xi})$ to the elliptic PDE in order to evaluate the potential energy

$$U(\boldsymbol{\xi}) = -\log p_{\mathbf{y}|\boldsymbol{\xi}}(\mathbf{y}|\boldsymbol{\xi}) - \log p_{\boldsymbol{\xi}}(\boldsymbol{\xi}) \tag{A.5}$$

in the Hamiltonian, and the gradient with respect to the parameter $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$, so that we can get HMC proposals as in (A.2). Note that the key to the evaluation of $\nabla_{\boldsymbol{\xi}} U(\boldsymbol{\xi})$ is the evaluation of derivatives $\frac{\partial u(\mathbf{x},\boldsymbol{\xi})}{\partial \xi_j} = u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})$, for $j = 1, ..., r$, where $u(\mathbf{x}, \boldsymbol{\xi})$ is the solution of the elliptic PDE problem (1)–(2). Moreover, $u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})$ satisfies

$$-\nabla \cdot \big(a(\mathbf{x}, \boldsymbol{\xi}) \nabla u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi})\big) = \nabla \cdot \big(a_{\xi_j}(\mathbf{x}, \boldsymbol{\xi}) \nabla u(\mathbf{x}, \boldsymbol{\xi})\big), \quad \mathbf{x} \in D, \tag{A.6}$$

$$u_{\xi_j}(\mathbf{x}, \boldsymbol{\xi}) = 0, \qquad \mathbf{x} \in \partial D, \tag{A.7}$$

which is the same elliptic PDE as (1)–(2) with a right-hand side that depends on the solution $u(\mathbf{x}, \boldsymbol{\xi})$ 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.

References

[1] A. Abdulle, A. Barth, and C. Schwab. Multilevel Monte Carlo methods for stochastic elliptic multiscale PDEs. *Multiscale Modeling & Simulation*, 11(4):1033–1070, 2013.

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

[3] D.A. Barajas-Solano and A.M. Tartakovsky. Approximate Bayesian model inversion for PDEs with heterogeneous and state-dependent coefficients. *Journal of Computational Physics*, 395:247–262, 2019.

[4] Mario Bebendorf and Wolfgang Hackbusch. Existence of H-matrix approximants to the inverse FE-matrix of elliptic operators with l^∞-coefficients. *Numer. Math.*, 95:1–28, 2003.

[5] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531, 2015.

[6] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox. *Model reduction and approximation: theory and algorithms*. SIAM, 2017.

[7] A. Beskos, A. Jasra, E. Muzaffer, and A. Stuart. Sequential Monte Carlo methods for Bayesian elliptic inverse problems. *Statistics and Computing*, 25(4):727–737, 2015.

[8] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[9] T. Cui, Y. M. Marzouk, and K. Willcox. Data-driven model reduction for the bayesian solution of inverse problems. *Int. J. Numer. Methods Eng.*, 102(5):966–990, 2015.

[10] M. Dashti and A. Stuart. Uncertainty Quantification and Weak Approximation of an Elliptic Inverse Problem. *SIAM Journal on Numerical Analysis*, 49(6):2524–2542, 2011.

[11] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[12] D. Garmatter, B. Haasdonk, and B. Harrach. A reduced basis landweber method for nonlinear inverse problems. *Inverse Probl.*, 32(3):035001, 2016.

[13] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

[14] R. Ghanem and P. Spanos. *Stochastic finite elements: a spectral approach.* Springer-Verlag, New York, 1991.

[15] R. Giordano, T. Broderick, and M. I. Jordan. Covariances, robustness, and variational Bayes. *Journal of Machine Learning Research*, 19:1981–2029, 2018.

[16] C. Gogu. Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. *Int. J. Numer. Methods Eng.*, 101(4):281–304, 2015.

[17] Paul Hagemann, Johannes Hertrich, and Gabriele Steidl. Stochastic normalizing flows for inverse problems: A Markov Chains viewpoint. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1162–1190, 2022.

[18] J. S. Hesthaven, G. Rozza, and B. Stamm. *Certified reduced basis methods for parametrized partial differential equations*, volume 590. Springer, 2016.

[19] T. Hou, D. Ma, and Z. Zhang. A model reduction method for multiscale elliptic PDEs with random coefficients using an optimization approach. *Multiscale Modeling & Simulation*, 17(2):826–853, 2019.

[20] M. I. Jordan, Z. Ghahramani, T. Jaakkola, and L. Saul. Introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[21] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer, New York, 2005.

[22] D.P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751, 2016.

[23] S. Lan. Adaptive dimension reduction to accelerate infinite-dimensional geometric Markov Chain Monte Carlo. *Journal of Computational Physics*, 392:71–95, 2019.

[24] S. Li, C. Zhang, Z. Zhang, and H. Zhao. A data-driven and model-based accelerated Hamiltonian Monte Carlo method for Bayesian elliptic inverse problems. *arXiv preprint arXiv:2104.13070*, 2021.

[25] S. Li, Z. Zhang, and H. Zhao. A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction. *SIAM Journal on Multiscale Modeling & Simulation*, 18 (3):1242–1271, 2020.

[26] D.J.C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[27] J. Martin, L. 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, 2012.

[28] A. Mondal, Y. Efendiev, B. Mallick, and A. Datta-Gupta. Bayesian uncertainty quantification for flows in heterogeneous porous media using reversible jump Markov chain Monte Carlo methods. *Advances in Water Resources*, 33(3):241–256, 2010.

[29] Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011.

[30] F. Nobile, R. Tempone, and C. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numer. Anal.*, 46:2309–2345, 2008.

[31] G. Papamakarios, E. Nalisnick, D. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *ArXiv Preprint arXiv:1912.02762*, 2019.

[32] Jan Povala, Ieva Kazlauskaite, Eky Febrianto, Fehmi Cirak, and Mark Girolami. Variational Bayesian approximation of inverse problems using sparse precision matrices. *Computer Methods in Applied Mechanics and Engineering*, 393:114712, 2022.

[33] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[34] Lawrence Sirovich. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Quart. Appl. Math.*, 45(3):561–571, 1987.

[35] A. Stuart. Inverse problems: a Bayesian perspective. *Acta numerica*, 19:451–559, 2010.

[36] Panagiotis Tsilifis, Ilias Bilionis, Ioannis Katsounaros, and Nicholas Zabaras. Computationally efficient variational approximations for Bayesian inverse problems. *Journal of Verification, Validation and Uncertainty Quantification*, 1(3), 07 2016. 031004.

[37] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 4(4):1–29, 2013.

[38] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Maching Learning*, 1(1-2):1–305, 2008.

[39] J. Wan and N. Zabaras. A probabilistic graphical model approach to stochastic multiscale partial differential equations. *Journal of Computational Physics*, 250:477–510, 2013.

[40] B. Wang and D.M. Titterington. Inadequacy of interval estimates corresponding to variational Bayesian approximations. In R.G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 373–380, 2005.

[41] M. J Zahr and C. Farhat. Progressive construction of a parametric reduced-order model for pde-constrained optimization. *Int. J. Numer. Methods Eng.*, 102(5):1111–1135, 2015.

[42] Z. Zhang, M. Ci, and T. Y. Hou. A multiscale data-driven stochastic method for elliptic PDEs with random coefficients. *SIAM Multiscale Model. Simul.*, 13:173–204, 2015.