Journal of Scientific Computing (2023) 95:30
https://doi.org/10.1007/s10915-023-02145-1

Large-Scale Bayesian Optimal Experimental Design with Derivative-Informed Projected Neural Network

Keyi Wu[1] · **Thomas O'Leary-Roseberry**[2] · **Peng Chen**[3] · **Omar Ghattas**[2,4]

Received: 19 January 2022 / Revised: 6 September 2022 / Accepted: 2 February 2023 /
Published online: 8 March 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
We address the solution of large-scale Bayesian optimal experimental design (OED) problems governed by partial differential equations (PDEs) with infinite-dimensional parameter fields. The OED problem seeks to find sensor locations that maximize the expected information gain (EIG) in the solution of the underlying Bayesian inverse problem. Computation of the EIG is usually prohibitive for PDE-based OED problems. To make the evaluation of the EIG tractable, we approximate the (PDE-based) parameter-to-observable map with a derivative-informed projected neural network (DIPNet) surrogate, which exploits the geometry, smoothness, and intrinsic low-dimensionality of the map using a small and dimension-independent number of PDE solves. The surrogate is then deployed within a greedy algorithm-based solution of the OED problem such that no further PDE solves are required. We analyze the EIG approximation error in terms of the generalization error of the DIPNet and show they are of the same order. Finally, the efficiency and accuracy of the method are demonstrated via numerical experiments on OED problems governed by inverse scattering and inverse reactive transport with up to 16,641 uncertain parameters and 100 experimental design variables, where we observe up to three orders of magnitude speedup relative to a reference double loop Monte Carlo method.

✉ Keyi Wu
keyiwu.w@gmail.com

Thomas O'Leary-Roseberry
tom.olearyroseberry@utexas.edu

Peng Chen
peng@cc.gatech.edu

Omar Ghattas
omar@oden.utexas.edu

[1] Department of Mathematics, The University of Texas at Austin, 2515 Speedway, Austin, TX 78712, USA

[2] Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin, 201 E 24th St, Austin, TX 78712, USA

[3] School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA

[4] Departments of Geological Sciences and Mechanical Engineering, The University of Texas at Austin, Austin 78712, TX, USA

Keywords Bayesian optimal experimental design · Bayesian inverse problem · Expected information gain · Neural network · Surrogate · Greedy algorithm

1 Introduction

In modeling natural or engineered systems, uncertainties are often present due to the lack of knowledge or intrinsic variability of the system. Uncertainties may arise from sources as varied as initial and boundary conditions, material properties and other coefficients, external source terms, interaction and coupling terms, and geometries; for simplicity of exposition, we refer to all of these as *parameters*. Uncertainties in prior knowledge of these parameters can be reduced by incorporating indirect observational or experimental data on the system state or related quantities of interest into the forward model via solution of a Bayesian inverse problem. Prior knowledge can be incorporated through a prior distribution on the uncertain parameters. The data are typically noisy because of limited measurement precision, which induces a likelihood of the data conditioned on the given parameters. Uncertainties of the parameters are then reduced by the data and quantified by a posterior distribution, which is a joint distribution of the prior and the likelihood, given by Bayes' rule.

Large amounts of informative data can reduce uncertainties in the parameters, and thus posterior predictions, significantly. However, the data are often sparse or limited due to the cost of their acquisition. In such cases it is critical to design the acquisition process or experiment in an optimal way so that as much information as possible can be gained from the acquired data, or the uncertainty in the parameters or posterior predictions can be reduced as much as possible. Experimental design variables can include what, when, and where to measure, which sources to use to excite the system, and under which conditions should the experiments be conducted. This is known as the optimal experimental design (OED) problem [1], or Bayesian OED in the context of Bayesian inference. OED problems arise across numerous fields including geophysical exploration, medical imaging, nondestructive evaluation, drug testing, materials characterization, and earth system data assimilation, to name just a few. For example, two notable uses of OED include optimal observing system design in oceanography [2] and optimal sensor placement for tsunami early warning [3].

The challenges to solving OED problems in these and other fields are manifold. The models underlying the systems of interest typically take the form of partial differential equations (PDEs) and can be large-scale, complex, nonlinear, dynamic, multiscale, and coupled. The uncertain parameters may depend on both space and time, and are often characterized by infinite-dimensional random fields and/or stochastic processes. The PDE models can be extremely expensive to solve for each realization of the infinite-dimensional uncertain parameters. The computation of the OED objective involves high-dimensional (after discretization) integration with respect to (w.r.t.) the uncertain parameters, and thus require a large number of PDE solves. Finally, the OED objective will need to be evaluated numerous times, especially when the experimental design variables are high-dimensional or when they represent discrete decisions.

To address these computational challenges, different classes of methods have been developed by exploiting (1) sparsity via polynomial chaos approximation of parameter-to-observable maps [4–6], (2) Laplace approximation of the posterior [7–11], (3) intrinsic low dimensionality by low-rank approximation of (prior-preconditioned and data-informed) operators [7, 12–16], (4) decomposibility by offline (for PDE-constrained approximation)–online (for design optimization) decomposition [17, 18], and (5) surrogate models of the

PDEs, parameter-to-observable map, or posterior distribution by model reduction [19, 20] and deep learning [21–23].

Here, we consider the Bayesian OED problem for optimal sensor placement governed by large-scale and possibly nonlinear PDEs with infinite-dimensional uncertain parameters. We use the expected information gain (EIG), also known as mutual information, as the optimality criterion for the OED problem. The optimization problem is combinatorial: we seek the combination of sensors, selected from a set of candidate locations, that maximizes the EIG. The EIG is an average of the Kullback–Leibler (KL) divergence between the posterior and the prior distributions over all realizations of the data. This involves a double integral: one integral of the likelihood function w.r.t. the prior distribution to compute the normalization constant or model evidence for each data realization, and one integral w.r.t. the data distribution. To evaluate the two integrals we adopt a double-loop Monte Carlo (DLMC) method that requires the computation of the parameter-to-observable map at each of the parameter and data samples. Since the likelihood can be rather complex and highly locally supported in the parameter space, the number of parameter samples from the prior distribution needed to capture the likelihood well with relatively accurate sample average approximation of the normalization constant can be extremely large. The requirement to evaluate the PDE-constrained parameter-to-observable map at each of the large number of samples leads to numerous PDE solves, which is prohibitive when the PDEs are expensive to solve. To tackle this challenge, we construct a derivative-informed projected neural network (DIPNet) [24–26] surrogate of the parameter-to-observable map that exploits the intrinsic low dimensionality of both the parameter and the data spaces. This intrinsic low dimensionality is due to the correlation of the high-dimensional parameters, the smoothing property of the underlying PDE solution, and redundant information contained in the data from all of the candidate sensors. In particular, the low-dimensional subspace of the parameter space can be detected via low rank approximations of derivatives of the parameter-to-observable map, such as the Jacobian, Gauss-Newton Hessian, or higher-order derivatives. This property has been observed and exploited across a wide spectrum of Bayesian inverse problems [27–42] and Bayesian optimal experimental design [7, 17, 18]. See [43] for analysis of model elliptic, parabolic, and hyperbolic problems, and a lengthy list of complex inverse problems that have been found numerically to exhibit this property.

This intrinsic low-dimensionality of parameter and data spaces, along with smoothness of the parameter-to-observable map, allow us to construct an accurate (over parameter space) DIPNet surrogate with a limited and dimension-independent number of training data pairs, each requiring a PDE solve. Once trained, the DIPNet surrogate is deployed in the OED problem, which is solved without further PDE solution, resulting in very large reductions in computing time. Under suitable assumptions, we provide an analysis of the error propagated from the DIPNet approximation to the approximation of the normalization constant and the EIG. To solve the combinatorial optimization problem of sensor selection, we use a greedy algorithm developed in our previous work [17, 18]. We demonstrate the efficiency and accuracy of our computational method by conducting two numerical experiments with infinite-dimensional parameter fields: OED for inverse scattering (with an acoustic Helmholtz forward problem) and inverse reactive transport (with a nonlinear advection–diffusion-reaction forward problem).

The rest of the paper is organized as follows. The setup of the problems including Bayesian inversion, EIG, sensor design matrix, and Bayesian OED are presented in Sect. 2. Section 3 is devoted to presentation of the computational methods including DLMC, DIPNet and its induced error analysis, and the greedy optimization algorithm. Results for the two OED numerical experiments are provided in Sect. 4, followed by conclusions in Sect. 5.

2 Problem Setup

2.1 Bayesian Inverse Problems

Let $\mathcal{D} \subset \mathbb{R}^{n_x}$ denote a physical domain in dimension $n_x = 1, 2, 3$. We consider the problem of inferring an uncertain parameter field m defined in the physical domain $\mathcal{D}$ from noisy data $\mathbf{y}$ and a complex model represented by PDEs. Let $\mathbf{y} \in \mathbb{R}^{n_y}$ denote the noisy data vector of dimension $n_y \in \mathbb{N}$, given by

$$\mathbf{y} = \mathcal{F}(m) + \boldsymbol{\varepsilon}, \tag{1}$$

which is contaminated by the additive Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Gamma}_{\text{n}})$ with zero mean and covariance $\boldsymbol{\Gamma}_{\text{n}} \subset \mathbb{R}^{n_y \times n_y}$. Specifically, $\mathbf{y}$ is obtained from observation of the solution of the PDE model at n_y sensor locations. $\mathcal{F}$ is the parameter-to-observable map which depends on the solution of the PDE and an observation operator that extracts the solution values at the n_y locations.

We consider the above inverse problem in a Bayesian framework. First, we assume that m lies in an infinite-dimensional real separable Hilbert space $\mathcal{M}$, e.g., $\mathcal{M} = \mathcal{L}^2(\mathcal{D})$ of square integrable functions defined in $\mathcal{D}$. Moreover, we assume that m follows a Gaussian prior measure $\mu_{\text{pr}} = \mathcal{N}(m_{\text{pr}}, \mathcal{C}_{\text{pr}})$ with mean $m_{\text{pr}} \in \mathcal{M}$ and covariance operator $\mathcal{C}_{\text{pr}}$, a strictly positive, self-adjoint, and trace-class operator. As one example, we consider $\mathcal{C}_{\text{pr}} = \mathcal{A}^{-2}$, where $\mathcal{A} = -\gamma\Delta + \delta I$ is a Laplacian-like operator with prescribed homogeneous Neumann boundary condition, with Laplacian Δ, identity I, and positive constants $\gamma, \delta > 0$; see [29, 44, 45] for more details. Given the Gaussian observation noise, the likelihood of the data $\mathbf{y}$ for the parameter $m \in \mathcal{M}$ satisfies

$$\pi_{\text{like}}(\mathbf{y}|m) \propto \exp\left(-\Phi(m, \mathbf{y})\right), \tag{2}$$

where

$$\Phi(m, \mathbf{y}) := \frac{1}{2}\|\mathbf{y} - \mathcal{F}(m)\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} \tag{3}$$

is known as a potential function. By Bayes' rule, the posterior measure, denoted as $\mu_{\text{post}}(m|\mathbf{y})$, is given by the Radon-Nikodym derivative as

$$\frac{d\mu_{\text{post}}(m|\mathbf{y})}{d\mu_{\text{pr}}(m)} = \frac{1}{\pi(\mathbf{y})}\pi_{\text{like}}(\mathbf{y}|m), \tag{4}$$

where $\pi(\mathbf{y})$ is the so-called normalization constant or model evidence, given by

$$\pi(\mathbf{y}) = \int_{\mathcal{M}} \pi_{\text{like}}(\mathbf{y}|m) d\mu_{\text{pr}}(m). \tag{5}$$

This expression is often computationally intractable because of the infinite-dimensional integral, which involves a (possibly large-scale) PDE solve for each realization m.

2.2 Expected Information Gain

To measure the information gained from the data $\mathbf{y}$ in the inference of the parameter m, we consider a Kullback–Leibler (KL) divergence between the posterior and the prior, defined as

$$D_{\text{KL}}(\mu_{\text{post}}(\cdot|\mathbf{y})\|\mu_{\text{pr}}) := \int_{\mathcal{M}} \ln\left(\frac{d\mu_{\text{post}}(m|\mathbf{y})}{d\mu_{\text{pr}}(m)}\right) d\mu_{\text{post}}(m|\mathbf{y}), \tag{6}$$

which is random since the data $\mathbf{y}$ is random. We consider a widely used optimality criterion, expected information gain (EIG), which is the KL divergence averaged over all realizations of the data, defined as

$$\begin{aligned}\Psi &:= \mathbb{E}_{\mathbf{y}}\left[D_{\mathrm{KL}}(\mu_{\mathrm{post}}(\cdot|\mathbf{y})\|\mu_{\mathrm{pr}})\right] \\ &= \int_{\mathcal{Y}} D_{\mathrm{KL}}(\mu_{\mathrm{post}}(\cdot|\mathbf{y})\|\mu_{\mathrm{pr}})\,\pi(\mathbf{y})\,d\mathbf{y} \\ &= \int_{\mathcal{Y}}\int_{\mathcal{M}} \ln\left(\frac{d\mu_{\mathrm{post}}(m|\mathbf{y})}{d\mu_{\mathrm{pr}}(m)}\right) d\mu_{\mathrm{post}}(m|\mathbf{y})\,\pi(\mathbf{y})\,d\mathbf{y} \\ &= \int_{\mathcal{M}}\int_{\mathcal{Y}} \ln\left(\frac{\pi_{\mathrm{like}}(\mathbf{y}|m)}{\pi(\mathbf{y})}\right)\pi_{\mathrm{like}}(\mathbf{y}|m)\,d\mathbf{y}\,d\mu_{\mathrm{pr}}(m),\end{aligned} \tag{7}$$

where the last equality follows Bayes' rule (4) and the Fubini theorem under the assumption of proper integrability.

2.3 Optimal Experimental Design

We consider the OED problem for optimally acquiring data to maximize the expected information gained in the parameter inference. The experimental of design seeks to choose r sensor locations out of d candidates $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_d\}$ represented by a design matrix $\mathbf{W} \in \mathbb{R}^{r\times d} \in \mathcal{W}$, namely, if the i-th sensor is placed at $\boldsymbol{x}_j$, then $\mathbf{W}_{ij} = 1$, otherwise $\mathbf{W}_{ij} = 0$:

$$\mathcal{W} := \left\{ \mathbf{W} \in \mathbb{R}^{r\times d} : \mathbf{W}_{ij} \in \{0,1\}, \sum_{i=1}^{r} \mathbf{W}_{ij} \in \{0,1\}, \sum_{j=1}^{d} \mathbf{W}_{ij} = 1 \right\}. \tag{8}$$

Let $\mathcal{F}_d : \mathcal{M} \mapsto \mathbb{R}^d$ denote the parameter-to-observable map and $\boldsymbol{\varepsilon}_d \in \mathbb{R}^d$ denote the additive noise, both using all d candidate sensors; then we have

$$\mathcal{F} = \mathbf{W}\mathcal{F}_d \quad \text{and} \quad \boldsymbol{\varepsilon} = \mathbf{W}\boldsymbol{\varepsilon}_d. \tag{9}$$

Then the likelihood (2) for a specific design $\mathbf{W}$ is given by

$$\pi_{\mathrm{like}}(\mathbf{y}|m, \mathbf{W}) \propto \exp\left(-\frac{1}{2}\|\mathbf{y} - \mathbf{W}\mathcal{F}_d(m)\|^2_{\boldsymbol{\Gamma}_{\mathrm{n}}^{-1}}\right), \tag{10}$$

and the normalization constant also depends on $\mathbf{W}$ as

$$\pi(\mathbf{y}, \mathbf{W}) = \int_{\mathcal{Y}} \pi_{\mathrm{like}}(\mathbf{y}|m, \mathbf{W}) d\mu_{\mathrm{pr}}(m). \tag{11}$$

From Sect. 2.2, we can see that the EIG Ψ depends on the design matrix $\mathbf{W}$ through the likelihood function $\pi_{\mathrm{like}}(\mathbf{y}|m, \mathbf{W})$. To this end, we formulate the OED problem to find an optimal design matrix $\mathbf{W}^*$ such that

$$\mathbf{W}^* = \arg\max_{\mathbf{W}\in\mathcal{W}} \Psi(\mathbf{W}), \tag{12}$$

with the $\mathbf{W}$-dependent EIG given by

$$\Psi(\mathbf{W}) = \int_{\mathcal{M}}\int_{\mathcal{Y}} \ln\left(\frac{\pi_{\mathrm{like}}(\mathbf{y}|m, \mathbf{W})}{\pi(\mathbf{y}, \mathbf{W})}\right)\pi_{\mathrm{like}}(\mathbf{y}|m, \mathbf{W})\,d\mathbf{y}\,d\mu_{\mathrm{pr}}(m). \tag{13}$$

2.4 Finite-Dimensional Approximation

To facilitate the presentation of our computational methods, we make a finite-dimensional approximation of the parameter field by using a finite element discretization. Let $\mathcal{M}_n \subset \mathcal{M}$ denote a subspace of $\mathcal{M}$ spanned by n piecewise continuous Lagrange polynomial basis functions $\{\psi_j\}_{j=1}^n$ over a mesh with elements of size h. Then the discrete parameter $m_h \in \mathcal{M}_n$ is given by

$$m_h = \sum_{j=1}^{n} m_j \psi_j. \tag{14}$$

The Bayesian inverse problem is then stated for the finite-dimensional coefficient vector $\mathbf{m} = (m_1, \ldots, m_n)^T$ of m_h, with n possibly very large. The prior distribution of the discrete parameter $\mathbf{m}$ is Gaussian $\mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{\Gamma}_{\text{pr}})$ with $\mathbf{m}_{\text{pr}}$ representing the coefficient vector of the discretized prior mean of m_{pr}, and $\mathbf{\Gamma}_{\text{pr}}$ representing the covariance matrix corresponding to $\mathcal{C}_{\text{pr}} = \mathcal{A}^{-2}$, given by

$$\mathbf{\Gamma}_{\text{pr}} = \mathbf{A}^{-1}\mathbf{M}\mathbf{A}^{-1}, \tag{15}$$

where $\mathbf{A}$ is the finite element matrix of the Laplacian-like operator $\mathcal{A}$, and $\mathbf{M}$ is the mass matrix. Moreover, let $\mathbf{F}_d : \mathbb{R}^n \to \mathbb{R}^d$ denote the discretized parameter-to-observable map corresponding to $\mathcal{F}_d$, we have $\mathbf{F} = \mathbf{W}\mathbf{F}_d$ as in (9). Then the likelihood function corresponding to (10) for the discrete parameter $\mathbf{m}$ is given by

$$\pi_{\text{like}}(\mathbf{y}|\mathbf{m}, \mathbf{W}) \propto \exp\left(-\frac{1}{2}\|\mathbf{y} - \mathbf{W}\mathbf{F}_d(\mathbf{m})\|^2_{\mathbf{\Gamma}_{\text{n}}^{-1}}\right). \tag{16}$$

3 Computational Methods

3.1 Double-Loop Monte Carlo Estimator

To solve the OED problem (12), we need to evaluate the EIG repeatedly for each given design $\mathbf{W}$. The double integrals in the EIG expression can be computed by a double-loop Monte Carlo (DLMC) estimator Ψ^{dl} defined as

$$\Psi^{dl}(\mathbf{W}) := \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \log\left(\frac{\pi_{\text{like}}(\mathbf{y}_i|\mathbf{m}_i, \mathbf{W})}{\hat{\pi}(\mathbf{y}_i, \mathbf{W})}\right), \tag{17}$$

where $\mathbf{m}_i$, $i = 1, \ldots, n_{\text{out}}$, are i.i.d. samples from prior $\mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{\Gamma}_{\text{pr}})$ in the outer loop and $\mathbf{y}_i = \mathbf{F}(\mathbf{m}_i) + \boldsymbol{\epsilon}_i$ are the realizations of the data with i.i.d. noise $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{\Gamma}_{\text{n}})$. $\hat{\pi}(\mathbf{y}_i, \mathbf{W})$ is a Monte Carlo estimator of the normalization constant $\pi(\mathbf{y}_i, \mathbf{W})$ with n_{in} samples in the inner loop, given by

$$\hat{\pi}(\mathbf{y}_i, \mathbf{W}) := \frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \pi_{\text{like}}(\mathbf{y}_i|\mathbf{m}_{i,j}, \mathbf{W}), \tag{18}$$

where $\mathbf{m}_{i,j}$, $j = 1, \ldots, n_{\text{in}}$, are i.i.d. samples from the prior $\mathcal{N}(\mathbf{m}_{\text{pr}}, \mathbf{\Gamma}_{\text{pr}})$.

For complex posterior distributions, e.g., high-dimensional, locally supported, multi-modal, non-Gaussian, etc., evaluation of the normalization constant is often intractable, i.e., a prohibitively large number of samples n_{in} is needed. As one particular example, when the posterior of $\mathbf{m}$ for data $\mathbf{y}_i$ generated at sample $\mathbf{m}_i$ concentrates in a very small region far away from the mean of the prior, the likelihood $\pi_{\text{like}}(\mathbf{y}_i|\mathbf{m}_{i,j}, \mathbf{W})$ is extremely small for most

samples $\mathbf{m}_{i,j}$, which which leads to a requirement of a large number of samples to evaluate $\hat{\pi}(\mathbf{y}_i, \mathbf{W})$ with relatively small estimation error. This is usually prohibitive, since one evaluation of the parameter-to-observable map, and thus one solution of the large-scale PDE model, is required for each of $n_{\text{out}} \times n_{\text{in}}$ samples. This $n_{\text{out}} \times n_{\text{in}}$ PDE solves are required for each design matrix $\mathbf{W}$ at each optimization iteration.

3.2 Derivative-Informed Projected Neural Networks

Recent research has motivated the deployment of neural networks as surrogates for parametric PDE mappings [24, 46–53]. These surrogates can be used to accelerate the computation of the EIG within OED problems. Specifically, to reduce the prohibitive computational cost, we build a surrogate for the parameter-to-observable map $\mathbf{F}_d : \mathbb{R}^n \mapsto \mathbb{R}^d$ at all candidate sensor locations by a derivative-informed projected neural network (DIPNet) [24–26]. Often, PDE-constrained high-dimensional parametric maps, such as the parameter-to-observable map $\mathbf{F}_d$, admit low-dimensional structure due to the correlation of the high-dimensional parameters, the regularizing property of the underlying PDE solution, and/or redundant information in the data from all candidate sensors. When this is the case, the DIPNet can exploit this low-dimensional structure and parametrize a parsimonious map between the most informed subspaces of the input parameter and the output observables. The dimensions of the input and output subspaces are referred to as the "information dimension" of the map, which is often significantly smaller than the parameter and data dimensions. The architectural strategy that we employ exploits compressibility of the map, by first reducing the input and output dimensionality via projection to informed reduced bases of the inputs and outputs. A neural network is then used to construct a low-dimensional nonlinear mapping between the two reduced bases. Error bounds for the effects of basis truncation, and parametrization by neural network are studied in [24, 26, 46].

For the input parameter dimension reduction, we use a vector generalization of an active subspace (AS) [54], which is spanned by the generalized eigenvectors (input reduced basis) corresponding to the r_M largest eigenvalues of the eigenvalue problem

$$\left[\int_{\mathcal{M}_n} \nabla_{\mathbf{m}} \mathbf{F}_d(\mathbf{m})^T \nabla_{\mathbf{m}} \mathbf{F}_d(\mathbf{m}) d\mu_{\text{pr}}(\mathbf{m})\right] v_i = \lambda_i^{AS} \boldsymbol{\Gamma}_{\text{pr}}^{-1} v_i, \tag{19}$$

where the eigenvectors v_i are ordered by the decreasing generalized eigenvalues λ_i^{AS}, $i = 1, \ldots, r_M$. For the output data dimension reduction, we use a proper orthogonal decomposition (POD)[55, 56], which uses the eigenvectors (output reduced basis) corresponding to the first r_F eigenvalues of the expected observable outer product matrix,

$$\left[\int_{\mathcal{M}_n} \mathbf{F}_d(\mathbf{m}) \mathbf{F}_d(\mathbf{m})^T d\mu_{\text{pr}}(\mathbf{m})\right] \phi_i = \lambda_i^{POD} \phi_i, \tag{20}$$

where the eigenvectors ϕ_i are ordered by the decreasing eigenvalues λ_i^{POD}, $i = 1, \ldots, r_F$. When the eigenvalues of AS and POD both decay quickly, the mapping $\mathbf{m} \mapsto \mathbf{F}_d(\mathbf{m})$ can be well approximated when $\mathbf{m}$ and $\mathbf{F}_d$ are projected to the corresponding subspaces with small r_M and r_F; in this case approximation error bounds for reduced basis representation of the mapping are given by the trailing eigenvalues of the systems (19),(20). This allows one to detect appropriate "breadth" for the neural network via the direct computation of the associated eigenvalue problems, removing the need for ad-hoc neural network hyperparameter search for appropriate breadth. The neural network surrogate $\tilde{\mathbf{F}}_d$ of the map F_d then has the

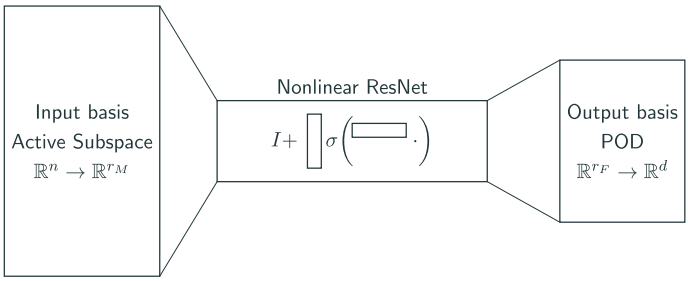

Fig. 1 A schematic representation of the derivative-informed projected neural network using ResNet as the nonlinear mapping between the reduced subspace using active subspaces for input parameters and POD for output observables

form

$$\tilde{\mathbf{F}}_d(\mathbf{m}, [\mathbf{w}, \mathbf{b}]) = \Phi_{r_F} \mathbf{f}_r(\mathbf{V}_{r_M}^T \mathbf{m}, \mathbf{w}) + \mathbf{b}, \tag{21}$$

where $\Phi_{r_F} \in \mathbb{R}^{d \times r_F}$ represents the POD reduced basis for the output, $\mathbf{V}_{r_M} \in \mathbb{R}^{n \times r_M}$ represents the AS reduced basis for the input, $\mathbf{f}_r \in \mathbb{R}^{r_F}$ is the neural network mapping between the two bases parametrized by weights $\mathbf{w}$ and bias $\mathbf{b}$. Since the reduced basis dimensions r_F, r_M are chosen based on spectral decay of the AS and POD operators, we can choose them to be the same; for convenience we denote the reduced basis dimension instead by r. The remaining difficulty is how to properly parametrize and train the neural network mapping. While the use of the reduced basis representation for the approximating map allows one to detect appropriate breadth for the neural network by avoiding complex neural network hyperparameter searches, and associated nonconvex neural network trainings, how to choose appropriate depth for the network is still an open question. While neural network approximation theory suggests deeper networks have richer representative capacities, in practice, for many architectures, adding depth eventually diminishes performance in what is known as the "peaking phenomenon"[57]. In general finding appropriate depth for e.g., fully-connected feedforward neural networks requires re-training from scratch different networks with differing depths. In order to avoid this issue we employ an adaptively constructed residual network (ResNet) neural network parametrization of the mapping between the two reduced bases. This adaptive construction procedure is motivated by recent approximation theory that conceives of ResNets as discretizations of sequentially minimizing control flows [58], where such maps are proven to be universal approximators of L^p functions on compact sets. A schematic for our neural network architecture can be seen in Fig. 1.

This strategy adaptively constructs and trains a sequence of low-rank ResNet layers, where for convenience we take $r = r_M = r_F$ or otherwise employ a restriction or prolongation layer to enforce dimensional compatibility. The ResNet hidden neurons at layer $i + 1$ have the form

$$z_{i+1} = z_i + w_{i,2}\sigma(w_{i,1}z_i + b_i), \tag{22}$$

with $z_{i+1}, z_i, b_i \in \mathbb{R}^r, w_{i,2}, w_{i,1}^T \in \mathbb{R}^{r \times k}$, where the parameter $k < r$ is referred to as the layer rank, and it is chosen to be smaller than r in order to impose a compressed representation of the ResNet latent space update (22). This choice is guided by the "well function" property in [58]. The ResNet weights $\mathbf{w} = [(w_{i,2}, w_{i,1}, b_i)]_{i=0}^{\text{depth}}$ consist of all of the coefficient arrays in each layer. Given appropriate reduced bases with dimension r, the ResNet mapping between the reduced bases is trained adaptively, one layer at a time, until over-fitting is detected in training validation metrics. When this is the case, a final global end-to-end training is

employed using a stochastic Newton optimizer [59]. This architectural strategy is able to achieve high generalizability for few and (input–output) dimension-independent data, for more information on this strategy, please see [26].

By removing the dependence of the input-to-output map on their high-dimensional and uninformed subspaces (complement to the low-dimensional and informed subspaces), we can construct a neural network of small input and output size that requires few training data. Since these architectures are able to achieve high generalization accuracy with limited training data for parametric PDE maps, they are especially well suited to scalable EIG approximation, since they can be efficiently queried many times at no cost in PDE solves, and require few high fidelity PDE solutions for their construction.

3.3 DLMC with DIPNet Surrogate

We propose to train a DIPNet surrogate $\tilde{\mathbf{F}}_d$ for the parameter-to-observable map $\mathbf{F}_d$, so that $\hat{\pi}(\mathbf{y}_i, \mathbf{W})$ can be approximated as

$$\tilde{\pi}(\mathbf{y}_i, \mathbf{W}) = \frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \exp\left(-\frac{1}{2} \|\mathbf{y}_i - \mathbf{W}\tilde{\mathbf{F}}_d(\mathbf{m}_{i,j})\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} \right), \tag{23}$$

where we omitted a constant $\frac{1}{(2\pi)^{n_{\text{out}}/2} \det(\boldsymbol{\Gamma}_{\text{n}})^{1/2}}$ since it appears in both the numerator and denominator of the argument of the log in the expression for the EIG. To this end, we can formulate the approximate EIG with the DIPNet surrogate as

$$\Psi^{nn} := \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \log\left(\frac{\pi_{\text{like}}(\mathbf{y}_i|\mathbf{m}_i, \mathbf{W})}{\tilde{\pi}(\mathbf{y}_i, \mathbf{W})} \right) \tag{24}$$

$$= -\frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \frac{1}{2} \|\mathbf{W}\boldsymbol{\varepsilon}_{d,i}\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} \tag{25}$$

$$- \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \log\left(\frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \exp\left(-\frac{1}{2} \|\mathbf{y}_i - \mathbf{W}\tilde{\mathbf{F}}_d(\mathbf{m}_{i,j})\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} \right) \right), \tag{26}$$

where $\boldsymbol{\varepsilon}_{d,i}$ are i.i.d. observation noise. Thanks to the DIPNet surrogate, the EIG can be evaluated at negligible cost (relating to PDE solver cost) for each given $\mathbf{W}$, and does not require any PDE solves.

3.4 Error Analysis

Theorem 1 *We assume that the parameter-to-observable map $\mathbf{F}_d$ and its surrogate $\tilde{\mathbf{F}}_d$ are bounded as*

$$\mathbb{E}_{\mathbf{m}\sim\mu_{pr}}[\|\mathbf{F}_d(\mathbf{m})\|^2] < \infty \quad \textit{and} \quad \mathbb{E}_{\mathbf{m}\sim\mu_{pr}}[\|\tilde{\mathbf{F}}_d(\mathbf{m})\|^2] < \infty. \tag{27}$$

Moreover, we assume that the generalization error of the DIPNet surrogate can be bounded by ε, i.e.,

$$\mathbb{E}_{\mathbf{m}\sim\mu_{pr}}[\|\mathbf{F}_d(\mathbf{m}) - \tilde{\mathbf{F}}_d(\mathbf{m})\|^2] \leq \varepsilon^2. \tag{28}$$

Then the error in the approximation of the normalization constant by the DIPNet surrogate can be bounded by

$$|\hat{\pi}(\mathbf{y}_i, \mathbf{W}) - \tilde{\pi}(\mathbf{y}_i, \mathbf{W})| \leq C_i\varepsilon, \tag{29}$$

for sufficiently large n_{in} and some constants $0 < C_i < \infty$, $i = 1, \dots, n_{out}$. Moreover, the approximation error for the EIG can be bounded by

$$|\Psi^{dl}(\mathbf{W}) - \Psi^{nn}(\mathbf{W})| \leq C\varepsilon \tag{30}$$

for some constant $0 < C < \infty$.

Proof For notational simplicity, we omit the dependence of $\hat{\pi}$ and $\tilde{\pi}$ on $\mathbf{W}$ and write $\mathbf{F} = \mathbf{W}\mathbf{F}_d$ and $\tilde{\mathbf{F}} = \mathbf{W}\tilde{\mathbf{F}}_d$. Note that the bounds (27) and (28) also hold for $\mathbf{F}$ and $\tilde{\mathbf{F}}$ since $\mathbf{F}$ and $\tilde{\mathbf{F}}$ are selection of some entries of $\mathbf{F}_d$ and $\tilde{\mathbf{F}}_d$. By definition of $\hat{\pi}$ in (18) and $\tilde{\pi}$ in (23), and the fact that $|e^{-x} - e^{-x'}| \leq |x - x'|$ for any $x, x' > 0$, we have

$$\begin{aligned}
&|\hat{\pi}(\mathbf{y}_i) - \tilde{\pi}(\mathbf{y}_i)|, \\
&\leq \frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \frac{1}{2} \left| \|\mathbf{y}_i - \mathbf{F}(\mathbf{m}_{i,j})\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} - \|\mathbf{y}_i - \tilde{\mathbf{F}}(\mathbf{m}_{i,j})\|^2_{\boldsymbol{\Gamma}_{\text{n}}^{-1}} \right|, \\
&= \frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \frac{1}{2} \left| \left(2\mathbf{y}_i - \mathbf{F}(\mathbf{m}_{i,j}) - \tilde{\mathbf{F}}(\mathbf{m}_{i,j})\right)^T \boldsymbol{\Gamma}_{\text{n}}^{-1} \left(\mathbf{F}(\mathbf{m}_{i,j}) - \tilde{\mathbf{F}}(\mathbf{m}_{i,j})\right) \right|, \\
&\leq \frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \|\boldsymbol{\Gamma}_{\text{n}}^{-1}\| (2\|\mathbf{y}_i\| + \|\mathbf{F}(\mathbf{m}_{i,j})\| + \|\tilde{\mathbf{F}}(\mathbf{m}_{i,j})\|) \|\mathbf{F}(\mathbf{m}_{i,j}) - \tilde{\mathbf{F}}(\mathbf{m}_{i,j})\|, \\
&\leq C_i \left(\frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \|\mathbf{F}(\mathbf{m}_{i,j}) - \tilde{\mathbf{F}}(\mathbf{m}_{i,j})\|^2 \right)^{1/2},
\end{aligned}$$

where we used the Cauchy–Schwarz inequality in the last inequality with C_i given by

$$C_i = \|\boldsymbol{\Gamma}_{\text{n}}^{-1}\| \left(4\|\mathbf{y}_i\|^2 + \left(\frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \|\mathbf{F}(\mathbf{m}_{i,j})\|^2 \right)^{1/2} + \left(\frac{1}{n_{\text{in}}} \sum_{j=1}^{n_{\text{in}}} \|\tilde{\mathbf{F}}(\mathbf{m}_{i,j})\|^2 \right)^{1/2} \right). \tag{31}$$

For sufficiently large n_{in}, we have that $C_i < \infty$ by $\|\mathbf{y}_i\| < \infty$ and the assumption (27). Moreover, the error bound (29) holds by the assumption (28).

By the definition of the EIGs (17) and (24), we have

$$|\Psi^{dl}(\mathbf{W}) - \Psi^{nn}(\mathbf{W})| \leq \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} |\log(\hat{\pi}(\mathbf{y}_i, \mathbf{W})) - \log(\tilde{\pi}(\mathbf{y}_i, \mathbf{W}))|. \tag{32}$$

For sufficiently large n_{in}, we have that the normalization constants $\hat{\pi}(\mathbf{y}_i, \mathbf{W})$ and $\tilde{\pi}(\mathbf{y}_i, \mathbf{W})$ are bounded away from zero, i.e.,

$$\hat{\pi}(\mathbf{y}_i, \mathbf{W}), \tilde{\pi}(\mathbf{y}_i, \mathbf{W}) \geq c_i, \tag{33}$$

for some constant $c_i > 0$. Then we have

$$|\Psi^{dl}(\mathbf{W}) - \Psi^{nn}(\mathbf{W})| \leq \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \frac{1}{c_i} |\hat{\pi}(\mathbf{y}_i, \mathbf{W}) - \tilde{\pi}(\mathbf{y}_i, \mathbf{W})| \leq \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \frac{C_i}{c_i} \varepsilon, \tag{34}$$

where we used the bound (29), which implies the bound (30) with constant

$$C = \frac{1}{n_{\text{out}}} \sum_{i=1}^{n_{\text{out}}} \frac{C_i}{c_i}. \tag{35}$$

□

3.5 Greedy Algorithm

With the DIPNet surrogate, the evaluation of the DLMC EIG Ψ^{nn} defined in (26) does not involve any PDE solves. Thus to solve the optimization problem

$$\mathbf{W}^* = \arg\max_{\mathbf{W}\in\mathcal{W}} \Psi^{nn}(\mathbf{W}), \tag{36}$$

we can directly use a greedy algorithm that requires evaluations of the EIG, not its derivative w.r.t. $\mathbf{W}$. Let S_d denote the set of all d candidate sensors; we wish to choose r sensors from S_d that maximize the approximate EIG (approximated with the DIPNet surrogate). At the first step with $t = 1$, we select the sensor $v^* \in S_d$ corresponding to the maximum approximate EIG and set $S^* = \{v^*\}$. Then at step $t = 2, \ldots, r$, with $t - 1$ sensors selected in S^*, we choose the t-th sensor $v^* \in S_d \backslash S^*$ corresponding to the maximum approximate EIG evaluated with t sensors $S^* \cup \{v^*\}$; see Algorithm 1 for the greedy optimization process, which can achieve (quasi-optimal) experimental designs with an approximation guarantee under suitable assumptions on the incremental information gain of an additional sensor; see [60] and references therein. Note that at each step the approximate EIG can be evaluated in parallel for each sensor choice $S^* \cup \{v\}$ with $v \in S_d \setminus S^*$.

Algorithm 1 Greedy algorithm to solve (36)

Require: data $\{\mathbf{y}_i\}_{i=1}^{N_s}$ generated from the prior samples $\{\boldsymbol{m}_i\}_{i=1}^{N_s}$, d sensor candidates set S_d, sensor budget r, optimal sensor set $S^* = \emptyset$
Ensure: optimal sensor set S^*
1: **for** $t = 1, \ldots, r$ **do**
2: $S \Leftarrow S_d \setminus S^*$
3: **for** $v \in S$ **do**
4: $\mathbf{W}_v$ is the design matrix of sensor choice $S^* \cup \{v\}$
5: evaluate $\Psi^{nn}(\mathbf{W}_v)$
6: **end for**
7: $v^* \Leftarrow \arg\max_{v\in S} \Psi^{nn}(\mathbf{W}_v)$
8: $S^* \Leftarrow S^* \cup \{v^*\}$
9: **end for**

4 Numerical Results

In this section, we present numerical results for OED problems involving a Helmholtz acoustic inverse scattering problem and an advection-reaction-diffusion inverse transport problem to illustrate the efficiency and accuracy of our method. We compare the approximated normalization constant and EIG of our method with 1) the DLMC truth computed by a large number of Monte Carlo samples; and 2) the DLMC sampled at the same computational cost (number of PDE solves) as our method using DIPNet training.

Both PDE problems are discretized using the finite element library `FEniCS` [61]. The construction of training data and reduced bases (active subspace and proper orthogonal

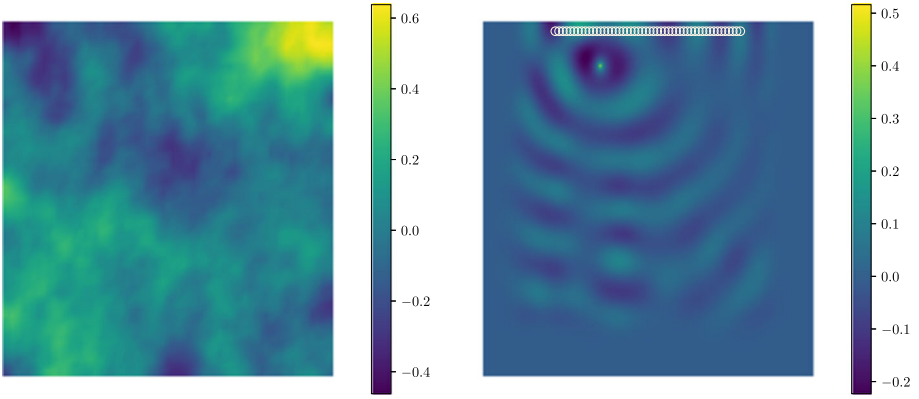

Fig. 2 A random sample of the parameter field **m** (left) and the corresponding solution **u** with candidate observation sensor locations marked in circles (right) for the Helmholtz problem

decomposition) is implemented in `hIPPYflow` [62], a library for dimension reduced PDE surrogate construction, building on top of PDE adjoints implemented in `hIPPYlib` [63]. The DIPNet neural network surrogates are constructed in `TensorFlow` [64], and are adaptively trained using a combination of Adam [65], and a Newton method, LRSFN, which improves local convergence and generalization [26, 59, 66].

4.1 Helmholtz Problem

For the first numerical experiment, we consider an inverse wave scattering problem modeled by the Helmholtz equation with uncertain medium in the two-dimensional physical domain $\Omega = (0, 3)^2$

$$-\Delta u - e^{2m}k^2 u = f \quad \text{in } \Omega, \tag{37a}$$

$$\text{PML boundary condition on } \partial\Omega \setminus \Gamma_{\text{top}}, \tag{37b}$$

$$\nabla u \cdot \mathbf{n} = 0 \text{ on } \Gamma_{\text{top}}, \tag{37c}$$

$$\mathcal{F}_d(m) = [u(\boldsymbol{x}_i, m)] \quad \text{at } \boldsymbol{x}_i \in \Omega. \tag{37d}$$

where u is the total wave field, $k \approx 4.55$ is the wavenumber, and e^{2m} models the uncertainty of the medium, with the parameter m a log-prefactor of the squared wavenumber. The right hand side f is a point source located at $\mathbf{x} = (0.775, 2.5)$. The perfectly matched layer (PML) boundary condition approximates a semi-infinite domain. The candidate sensor locations $\boldsymbol{x}_i$ are linearly spaced in the line segment between the edge points $(0.1, 2.9)$ and $(2.9, 2.9)$, with coordinates $\{(0.1 + 2i/35, 2.9)\}_{i=0}^{49}$, as shown in Fig. 2. The prior distribution for the uncertain parameter m is Gaussian $\mu_{\text{pr}} = \mathcal{N}(m_{\text{pr}}, \mathcal{C}_{\text{pr}})$ with zero mean $m_{\text{pr}} = 0$ and covariance $\mathcal{C}_{\text{pr}} = (5.0I - \Delta)^{-2}$. The mesh used for this problem is uniform of size 128×128. We use quadratic elements for the discretization of the wave field u and linear elements for the parameter m, leading to a discrete parameter **m** of dimension 16, 641. The dimension of the wave field u is 66049, the wave is more than sufficiently resolved in regards to the Nyquist sampling criteria for wave problems.

A sample of the parameter field m and the PDE solution u is shown in Fig. 2 with all 50 candidate sensor locations marked in circles.

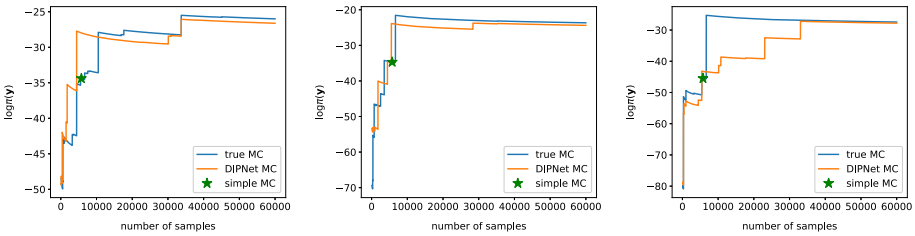

Fig. 3 The approximation of the log normalization constant with increasing numbers of samples without (true MC) and with (DIPNet MC) surrogate at 3 random designs. Green stars indicate 5757 samples (for simple MC), which is the same computational cost as DIPNet

The network has 10 low-rank residual layers, each with a layer rank of 10. For this numerical example we demonstrate the effects of using different breadths in the neural network representation, in each case the ResNet learns a mapping from the first r_M basis vectors for active subspace to the first r_M basis vectors of POD. In the case that $r_M > 50$ we use a linear restriction layer to reduce the ResNet latent representation to the 50 dimensional output. For the majority of the numerical results, we employ a $r_M = 50$ dimensional network. The neural network is trained adaptively using 4915 training samples, and 1228 validation samples. Using 512 independent testing samples, the DIPNet surrogate was 81.56% ℓ^2 accurate measured as a percentage by

$$\ell^2 \text{ accuacy} = 100\left(1 - \frac{\|\tilde{\mathbf{F}}_d - \mathbf{F}_d\|_2}{\|\mathbf{F}_d\|_2}\right). \tag{38}$$

For more details on this neural network architecture and training, see [26]. The computational cost of the 50 dimensional active subspace projector using 128 samples is equivalent to the cost of 842 additional training data; since the problem is linear the additional linear adjoint computations are comparable to the costs of the training data generation. As we will see in the next example, when the PDE is nonlinear the additional active subspace computations are marginal. Thus for fair comparison with the same computational cost of PDE solves, we use $4915 + 842 = 5757$ samples for simple MC.

To test the efficiency and accuracy of our DIPNet surrogate, we first compute the log normalization constant $\log \pi(\mathbf{y})$ with our DIPNet surrogate for given observation data $\mathbf{y}$ generated by $\mathbf{y} = \mathbf{W}\mathbf{F}_d(\mathbf{m}) + \boldsymbol{\epsilon}$. $\mathbf{m}$ is a random sample from the prior. We use in total 60000 random samples for Monte Carlo (MC) to compute the normalization constant as the ground truth. Figure 3 shows the $\log \pi(\mathbf{y})$ comparison for three random designs $\mathbf{W}$ that choose 15 sensors out of 50 candidates. We can see that DIPNet surrogate converges to a value close to the ground truth MC reference, while for the (simple) MC with 5757 samples, the approximated value indicated by the green star is much less accurate than the DIPNet surrogate using 60000 samples with similar cost in PDE solves. Note that the DIPNet training and evaluation cost for this small size neural network is negligible compared to PDE solves.

The left and middle figures in Fig. 4 illustrate the sample distributions of the relative approximation errors for the log normalization constant $\log \pi(\mathbf{y})$ and the EIG Ψ ($n_{\text{out}} = 200$) with (DIPNet MC with 60000 samples) and without (simple MC with 5757 samples) the DIPNet surrogate, compared to the true MC with 60000 samples. These results show that using DIPNet surrogate we can approximate $\log \pi(\mathbf{y})$ and Ψ much more accurately with less bias compared to the simple MC. The sample distributions of EIG at 200 random designs compared to the design chosen by the greedy optimization using DIPNet surrogate

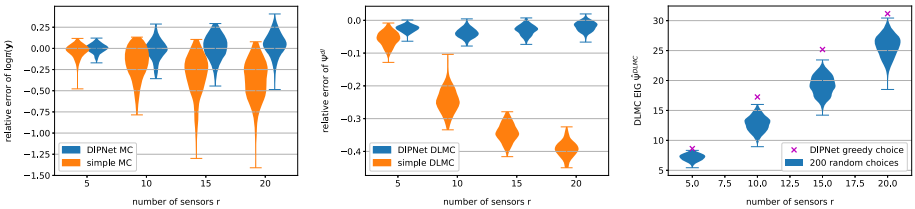

Fig. 4 Sample distributions for 200 random designs of the relative approximation errors (compared to true MC) for the log normalization constant $\log \pi(\mathbf{y})$ (left) and EIG Ψ (middle) by DIPNet MC and simple MC with different number of sensors r. Right: Blue filled areas represent the sample distributions of the true DLMC EIG Ψ^{dl} for 200 random designs. Pink crosses is the true DLMC EIG Ψ^{dl} of designs chosen by the greedy optimization using DIPNet surrogates

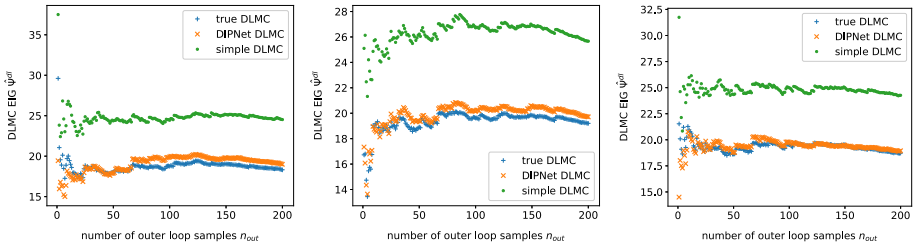

Fig. 5 EIG of true DLMC ($n_{\text{in}} = 60000$), DIPNet DLMC ($n_{\text{in}} = 60000$), and simple DLMC ($n_{\text{in}} = 5757$) with increasing number of outer loop samples n_{out} at 3 random designs

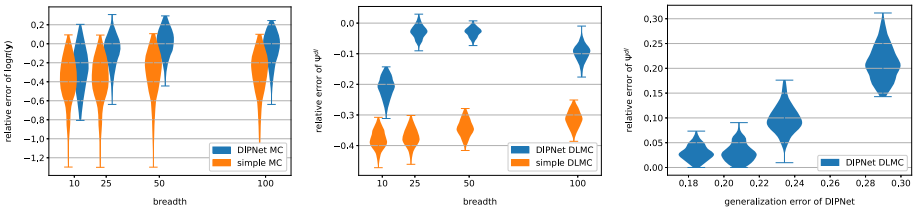

Fig. 6 Sample distributions for 200 random designs of the relative approximation errors (compared to true MC) for the log normalization constant $\log \pi(\mathbf{y})$ (left) and EIG Ψ (middle) by DIPNet MC and simple MC with increasing breadth. Right: Sample distributions for 200 random designs of the relative approximation errors (compared to true MC) against the corresponding DIPNet generalization error at breadth (from left to right) = 50, 100, 25, 10

for different number of sensors are shown in the right figure, from which we can see that our method can always chose better designs with larger EIG values than all the random designs.

Figure 5 gives approximate values of the EIG with increasing number of outer loop samples n_{out} using: the DIPNet surrogate with inner loop $n_{\text{in}} = 60000$, simple DLMC with inner loop $n_{\text{in}} = 5757$, and the truth computed with inner loop $n_{\text{in}} = 60000$. We can see that the approximate values by the DIPNet surrogates are almost the same as the truth, while simple DLMC results are very inaccurate.

To show the effectiveness of truncated rank (breadth) for DIPNet surrogate, we evaluate the log normalization constant $\log \pi(\mathbf{y})$ and EIG Ψ with breadth $= 10, 25, 50, 100$ and compared with true MC and simple MC in Fig. 6. We can see that with increasing breadth, the relative error is decreasing, but gets worse when breadth reaches 100. With breadth $= 100$, the difficulties of neural network training start to dominate and diminish the accuracy. We

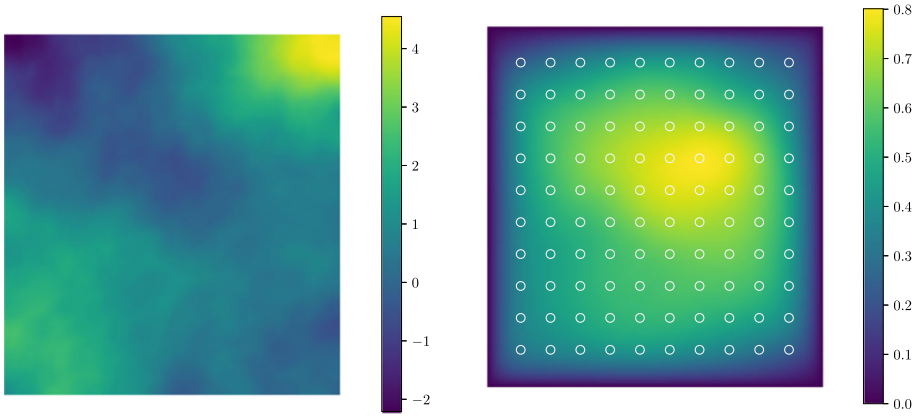

Fig. 7 A random sample of the parameter field **m** (left) and the corresponding solution **u** with candidate observation sensor locations marked in circles (right) for the advection–diffusion-reaction problem

can also see it in the right part of the figure. The relative error of EIG approximation reduces (close to) linearly with respect to the generalization error of the DIPNet approximation of the observables with network breadth $= 10, 25, 50$, which confirms the error analysis in Theorem 1. However, when the breadth increases to 100, the neural network becomes less accurate (without using more training data), leading to less accurate EIG approximation.

4.2 Advection–Diffusion-Reaction Problem

For the second numerical experiment we consider an OED problem for an advection–diffusion-reaction equation with a cubic nonlinear reaction term. The uncertain parameter m appears as a log-coefficient of the cubic nonlinear reaction term. The PDE is defined in a domain $\Omega = (0, 1)^2$ as

$$-\nabla \cdot (k\nabla u) + \mathbf{v} \cdot \nabla u + e^m u^3 = f \quad \text{in } \Omega, \tag{39a}$$

$$u = 0 \text{ on } \partial\Omega, \tag{39b}$$

$$\mathcal{F}_d(m) = [u(\boldsymbol{x}_i, m)] \quad \text{at } \boldsymbol{x}_i \in \Omega. \tag{39c}$$

Here $k = 0.01$ is the diffusion coefficient. The volumetric forcing function f is a smoothed Gaussian bump located at $\boldsymbol{x} = (0.7, 0.7)$,

$$f(\mathbf{x}) = \max\left(0.5, e^{-25(x_1-0.7)^2-25(x_2-0.7)^2}\right). \tag{40}$$

The velocity field $\mathbf{v}$ is a solution of a steady-state Navier Stokes equation with shearing boundary conditions driving the flow (see the Appendix in [24] for more information on the flow). The candidate sensor locations are located in a linearly spaced mesh-grid of points in $(0.1, 0.9) \times (0.1, 0.9)$, with coordinates $\{(0.1i, 0.1j), i, j = 1, 2, \ldots, 9\}$. The prior distribution for the uncertain parameter m is a mean zero Gaussian with covariance $\mathcal{C}_{\text{pr}} = (I - 0.1\Delta)^{-2}$. The mesh used for this problem is uniform of size 128×128. We use linear elements for both u and m, leading to a discrete parameter of dimension 16, 641. Figure 7 gives a prior sample of the parameter field m and the solution u with all 100 candidate sensor locations in white circles.

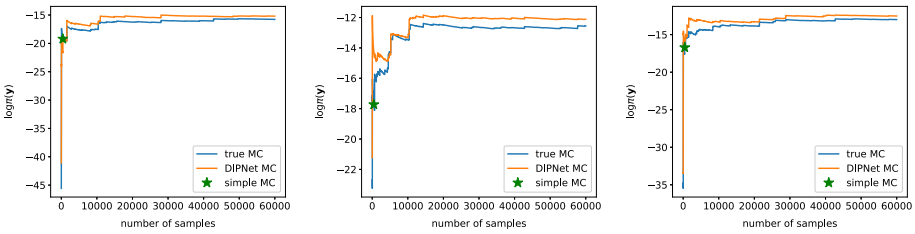

Fig. 8 The approximation of the log normalization constant with increasing numbers of samples without (true MC) and with (DIPNet MC) surrogate at 3 random designs. Green stars indicate 443 samples (simple MC), having the same computational cost as DIPNet

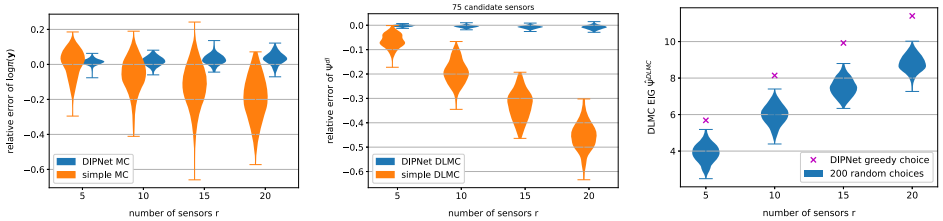

Fig. 9 Sample distributions of the relative errors (compared to the truth MC) in approximating the log normalization constant $\log \pi(\mathbf{y})$ (left) and EIG Ψ (middle) by DIPNet MC and simple MC with different number of sensors r; Right: Blue filled areas represent the sample distributions of the true DLMC EIG Ψ^{dl} for 200 random designs. Pink crosses are the true DLMC EIG Ψ^{dl} of designs chosen by the greedy optimization using DIPNet surrogates

The neural network surrogate is trained adaptively using 409 training samples and 102 validation samples. Using 512 independent testing samples the DIPNet network was $97.13\%\ell^2$ accurate (see equation 38). The network has 20 low-rank residual layers, each with a layer rank of 10, the breadth of the network is $r_M = r_F = 25$. The computational cost of the 25 dimensional active subspace projector using 256 samples is equivalent to the cost of 34 additional training data. As was noted before when the PDE is nonlinear the linear adjoint-based derivative computations become much less of a computational burden. Thus we use $409 + 34 = 443$ samples for simple MC for fair comparison.

We first examine the log normalization constant $\log \pi(\mathbf{y})$ computed with our 409 PDE-solve-based DIPNet surrogate compared against the truth computed with 60000 MC samples and the simple MC computed with 443 samples. Figure 8 shows the $\log \pi(\mathbf{y})$ comparison for three random designs that select 15 sensors out of 100 candidates. We can see that DIPNet MC converges to a value close to the true MC curves, while the simple MC's green star computed with the same number of PDE solves (443) as DIPNet MC, has much worse accuracy.

The left figure of Fig. 9 shows the relative errors for $\log \pi(\mathbf{y})$ computed with the DIPNet surrogate and simple MC using 443 samples based on the true MC with 60000 samples, for 200 random designs. We see again that DIPNet gives better accuracy with less bias than simple MC.

Figure 10 shows the EIG approximations of three random designs with increasing number of outer loop samples n_{out} using the DIPNet MC with 60000 (inner loop) samples, simple MC with 443 samples, and true MC with 60000 samples. We can see that the values of DIPNet MC are quite close to the true MC, while simple MC is far off. Relative errors of the EIG Ψ ($n_{\text{out}} = 100$) computed with the DIPNet surrogate, the simple MC for 200 random designs is given in the middle figure of Fig. 9. With the DIPNet DLMC Ψ^{nn}, we can use the

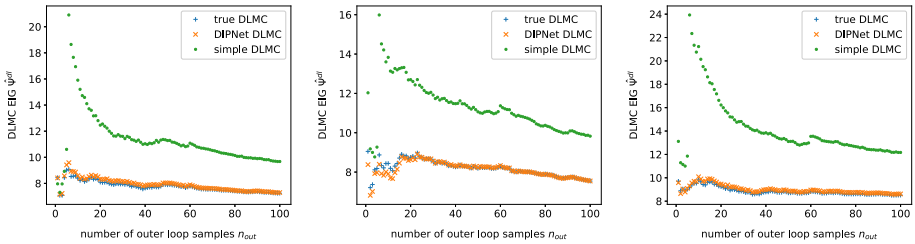

Fig. 10 EIG of true DLMC ($n_{\text{in}} = 60000$), DIPNet DLMC ($n_{\text{in}} = 60000$) and simple DLMC ($n_{\text{in}} = 443$) with increasing number of outer loop samples n_{out} at 3 random designs

greedy algorithm to find the optimal designs. The DIPNet greedy designs are presented as the pink crosses in the right figure of Fig. 9. We can see that the designs chosen by the greedy algorithm have much larger EIG values than all 200 random designs.

5 Conclusions

We have developed a computational method based on DIPNet surrogates for solving large-scale PDE-constrained Bayesian OED problems to determine optimal sensor locations (using the EIG criterion) to best infer infinite-dimensional parameters. We exploited the intrinsic low dimensionality of the parameter and data spaces and constructed a DIPNet surrogate for the parameter-to-observable map. The surrogate was used repeatedly in the evaluation of the normalization constant and the EIG. We presented error analysis for the approximation of the normalization constant and the EIG, showing that the errors are of the same order as the DIPNet RMS approximation error. Moreover, we used a greedy algorithm to solve the combinatorial optimization problem for sensor selection. The computational efficiency and accuracy of our approach are demonstrated by two numerical experiments. Future work will focus on gradient-based optimization also using the derivative information of the DIPNet w.r.t. both the parameter and the design variables, on the use of different optimality criteria such as A-optimality or D-optimality, and on exploring new network architectures for intrinsically high-dimensional Bayesian OED problems.

Funding This research was partially funded by DOE ASCR DE-SC0019303 and DE-SC0021239, DOD MURI FA9550-21-1-0084, and NSF DMS-2012453.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

1. Uciński, D.: Optimal Measurement Methods for Distributed Parameter System Identification. CRC Press, Boca Raton (2005)
2. Loose, N., Heimbach, P.: Leveraging uncertainty quantification to design ocean climate observing systems. J. Adv. Model. Earth Syst., pp 1–29 (2021)

3. Ferrolino, A.R., Lope, J.E.C., Mendoza, R.G.: Optimal location of sensors for early detection of tsunami waves. In: International Conference on Computational Science, pp. 562–575 (2020). Springer
4. Huan, X., Marzouk, Y.M.: Simulation-based optimal Bayesian experimental design for nonlinear systems. J. Comput. Phys. **232**(1), 288–317 (2013). https://doi.org/10.1016/j.jcp.2012.08.013
5. Huan, X., Marzouk, Y.M.: Gradient-based stochastic optimization methods in Bayesian experimental design. Int. J. Uncertain. Quantif. **4**(6), 479–510 (2014)
6. Huan, X., Marzouk, Y.M.: Sequential bayesian optimal experimental design via approximate dynamic programming. arXiv preprint arXiv:1604.08320 (2016)
7. Alexanderian, A., Petra, N., Stadler, G., Ghattas, O.: A fast and scalable method for A-optimal design of experiments for infinite-dimensional Bayesian nonlinear inverse problems. SIAM J. Sci. Comput. **38**(1), 243–272 (2016). https://doi.org/10.1137/140992564
8. Beck, J., Dia, B.M., Espath, L.F., Long, Q., Tempone, R.: Fast bayesian experimental design: Laplace-based importance sampling for the expected information gain. Comput. Methods Appl. Mech. Eng. **334**, 523–553 (2018). https://doi.org/10.1016/j.cma.2018.01.053
9. Long, Q., Scavino, M., Tempone, R., Wang, S.: Fast estimation of expected information gains for Bayesian experimental designs based on Laplace approximations. Comput. Methods Appl. Mech. Eng. **259**, 24–39 (2013)
10. Long, Q., Motamed, M., Tempone, R.: Fast bayesian optimal experimental design for seismic source inversion. Comput. Methods Appl. Mech. Eng. **291**, 123–145 (2015). https://doi.org/10.1016/j.cma.2015.03.021
11. Beck, J., Mansour Dia, B., Espath, L., Tempone, R.: Multilevel double loop Monte Carlo and stochastic collocation methods with importance sampling for Bayesian optimal experimental design. Int. J. Numer. Methods Eng. **121**(15), 3482–3503 (2020)
12. Alexanderian, A., Petra, N., Stadler, G., Ghattas, O.: A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized ℓ_0-sparsification. SIAM J. Sci. Comput. **36**(5), 2122–2148 (2014). https://doi.org/10.1137/130933381
13. Alexanderian, A., Gloor, P.J., Ghattas, O.: On Bayesian A-and D-optimal experimental designs in infinite dimensions. Bayesian Anal. **11**(3), 671–695 (2016). https://doi.org/10.1214/15-BA969
14. Saibaba, A.K., Alexanderian, A., Ipsen, I.C.: Randomized matrix-free trace and log-determinant estimators. Numerische Mathematik **137**(2), 353–395 (2017)
15. Crestel, B., Alexanderian, A., Stadler, G., Ghattas, O.: A-optimal encoding weights for nonlinear inverse problems, with application to the Helmholtz inverse problem. Inverse Prob. **33**(7), 074008 (2017)
16. Attia, A., Alexanderian, A., Saibaba, A.K.: Goal-oriented optimal design of experiments for large-scale Bayesian linear inverse problems. Inverse Prob. **34**(9), 095009 (2018)
17. Wu, K., Chen, P., Ghattas, O.: A fast and scalable computational framework for large-scale and high-dimensional Bayesian optimal experimental design. arXiv preprint arXiv:2010.15196, to appear in SIAM Journal on Scientific Computing (2020)
18. Wu, K., Chen, P., Ghattas, O.: An efficient method for goal-oriented linear bayesian optimal experimental design: Application to optimal sensor placement. arXiv preprint arXiv:2102.06627, to appear in SIAM/AMS Journal on Uncertainty Quantification (2021)
19. Aretz-Nellesen, N., Chen, P., Grepl, M.A., Veroy, K.: A-optimal experimental design for hyper-parameterized linear Bayesian inverse problems. Numer. Math. Adv. Appl. ENUMATH 2020 (2020)
20. Aretz, N., Chen, P., Veroy, K.: Sensor selection for hyper-parameterized linear Bayesian inverse problems. PAMM **20**(S1), 202000357 (2021)
21. Foster, A., Jankowiak, M., Bingham, E., Horsfall, P., Teh, Y.W., Rainforth, T., Goodman, N.: Variational Bayesian optimal experimental design. In: Wallach, H., Larochelle, H., Beygelzimer, A., d' Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. **32**, pp. 14036–14047. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper/2019/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf
22. Kleinegesse, S., Gutmann, M.U.: Bayesian experimental design for implicit models by mutual information neural estimation. In: International Conference on Machine Learning, pp. 5316–5326 (2020). PMLR
23. Shen, W., Huan, X.: Bayesian sequential optimal experimental design for nonlinear models using policy gradient reinforcement learning. arXiv preprint arXiv:2110.15335 (2021)
24. O'Leary-Roseberry, T., Villa, U., Chen, P., Ghattas, O.: Derivative-informed projected neural networks for high-dimensional parametric maps governed by pdes. Comput. Methods Appl. Mech. Eng. **388**, 114199 (2022)
25. O'Leary-Roseberry, T.: Efficient and dimension independent methods for neural network surrogate construction and training. PhD thesis, The University of Texas at Austin (2020)
26. O'Leary-Roseberry, T., Du, X., Chaudhuri, A., Martins, J.R., Willcox, K., Ghattas, O.: Adaptive projected residual networks for learning parametric maps from sparse data. arXiv preprint arXiv:2112.07096 (2021)

27. Flath, P.H., Wilcox, L.C., Akçelik, V., Hill, J., van Bloemen Waanders, B., Ghattas, O.: Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations. SIAM J. Sci. Comput. **33**(1), 407–432 (2011). https://doi.org/10.1137/090780717
28. Bui-Thanh, T., Burstedde, C., Ghattas, O., Martin, J., Stadler, G., Wilcox, L.C.: Extreme-scale UQ for Bayesian inverse problems governed by PDEs. In: SC12: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (2012)
29. Bui-Thanh, T., Ghattas, O., Martin, J., Stadler, G.: A computational framework for infinite-dimensional Bayesian inverse problems Part I: The linearized case, with application to global seismic inversion. SIAM J. Sci. Comput. **35**(6), 2494–2523 (2013). https://doi.org/10.1137/12089586X
30. Bui-Thanh, T., Ghattas, O.: An analysis of infinite dimensional Bayesian inverse shape acoustic scattering and its numerical approximation. SIAM/ASA J. Uncertain. Quantif. **2**(1), 203–222 (2014). https://doi.org/10.1137/120894877
31. Kalmikov, A.G., Heimbach, P.: A Hessian-based method for uncertainty quantification in global ocean state estimation. SIAM J. Sci. Comput. **36**(5), 267–295 (2014)
32. Hesse, M., Stadler, G.: Joint inversion in coupled quasistatic poroelasticity. J. Geophys. Res. Solid Earth **119**(2), 1425–1445 (2014)
33. Isaac, T., Petra, N., Stadler, G., Ghattas, O.: Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet. J. Comput. Phys. **296**, 348–368 (2015). https://doi.org/10.1016/j.jcp.2015.04.047
34. Cui, T., Law, K.J.H., Marzouk, Y.M.: Dimension-independent likelihood-informed MCMC. J. Comput. Phys. **304**, 109–137 (2016)
35. Chen, P., Villa, U., Ghattas, O.: Hessian-based adaptive sparse quadrature for infinite-dimensional Bayesian inverse problems. Comput. Methods Appl. Mech. Eng. **327**, 147–172 (2017)
36. Beskos, A., Girolami, M., Lan, S., Farrell, P.E., Stuart, A.M.: Geometric MCMC for infinite-dimensional inverse problems. J. Comput. Phys. **335**, 327–351 (2017)
37. Zahm, O., Cui, T., Law, K., Spantini, A., Marzouk, Y.: Certified dimension reduction in nonlinear Bayesian inverse problems. arXiv preprint arXiv:1807.03712 (2018)
38. Brennan, M., Bigoni, D., Zahm, O., Spantini, A., Marzouk, Y.: Greedy inference with structure-exploiting lazy maps. Adv. Neural Inf. Process. Syst. **33** (2020)
39. Chen, P., Wu, K., Chen, J., O'Leary-Roseberry, T., Ghattas, O.: Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions. Adv. Neural Inf. Process, Syst (2019)
40. Chen, P., Ghattas, O.: Projected Stein variational gradient descent. Adv. Neural Inf. Process, Syst (2020)
41. Subramanian, S., Scheufele, K., Mehl, M., Biros, G.: Where did the tumor start? An inverse solver with sparse localization for tumor growth models. Inverse Prob. **36**(4), 045006 (2020). https://doi.org/10.1088/1361-6420/ab649c
42. Babaniyi, O., Nicholson, R., Villa, U., Petra, N.: Inferring the basal sliding coefficient field for the Stokes ice sheet model under rheological uncertainty. Cryosphere **15**(4), 1731–1750 (2021)
43. Ghattas, O., Willcox, K.: Learning physics-based models from data: perspectives from inverse problems and model reduction. Acta Numer. **30**, 445–554 (2021). https://doi.org/10.1017/S0962492921000064
44. Stuart, A.M.: Inverse problems: a Bayesian perspective. Acta Numer. **19**, 451–559 (2010). https://doi.org/10.1017/S0962492910000061
45. Petra, N., Martin, J., Stadler, G., Ghattas, O.: A computational framework for infinite-dimensional Bayesian inverse problems: Part II. Stochastic Newton MCMC with application to ice sheet flow inverse problems. SIAM J. Sci. Comput. **36**(4), 1525–1555 (2014)
46. Bhattacharya, K., Hosseini, B., Kovachki, N.B., Stuart, A.M.: Model reduction and neural networks for parametric pdes. arXiv preprint arXiv:2005.03180 (2020)
47. Fresca, S., Manzoni, A.: POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized pdes by proper orthogonal decomposition. Comput. Methods Appl. Mech. Eng. **388**, 114181 (2022)
48. Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., Anandkumar, A.: Neural operator: Learning maps between function spaces. arXiv preprint arXiv:2108.08481 (2021)
49. Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A.: Neural operator: Graph kernel network for partial differential equations. arXiv preprint arXiv:2003.03485 (2020)
50. Lu, L., Jin, P., Karniadakis, G.E.: Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. arXiv preprint arXiv:1910.03193 (2019)

51. O'Leary-Roseberry, T., Chen, P., Villa, U., Ghattas, O.: Derivative-Informed Neural Operator: An Efficient Framework for High-Dimensional Parametric Derivative Learning. arXiv preprint arXiv:2206.10745 (2022)
52. Nelsen, N.H., Stuart, A.M.: The random feature model for input-output maps between banach spaces. SIAM J. Sci. Comput. **43**(5), 3212–3243 (2021)
53. Nguyen, H.V., Bui-Thanh, T.: Model-constrained deep learning approaches for inverse problems. arXiv preprint arXiv:2105.12033 (2021)
54. Zahm, O., Constantine, P.G., Prieur, C., Marzouk, Y.M.: Gradient-based dimension reduction of multivariate vector-valued functions. SIAM J. Sci. Comput. **42**(1), 534–558 (2020)
55. Manzoni, A., Negri, F., Quarteroni, A.: Dimensionality reduction of parameter-dependent problems through proper orthogonal decomposition. Ann. Math. Sci. Appl. **1**(2), 341–377 (2016)
56. Quarteroni, A., Manzoni, A., Negri, F.: Reduced Basis Methods for Partial Differential Equations: An Introduction vol. **92**. Springer (2015)
57. Hughes, G.: On the mean accuracy of statistical pattern recognizers. IEEE Trans. Inf. Theory **14**(1), 55–63 (1968)
58. Li, Q., Lin, T., Shen, Z.: Deep learning via dynamical systems: An approximation perspective. J. Eur, Math, Soc. (2022)
59. O'Leary-Roseberry, T., Alger, N., Ghattas, O.: Low rank saddle free Newton: A scalable method for stochastic nonconvex optimization. arXiv preprint arXiv:2002.02881 (2020)
60. Jagalur-Mohan, J., Marzouk, Y.: Batch greedy maximization of non-submodular functions: guarantees and applications to experimental design. J. Mach. Learn. Res. **22**(252), 1–62 (2021)
61. Alnæs, M.S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The FEniCS project version 1.5. Archive of Numerical Software **3**(100) (2015). https://doi.org/10.11588/ans.2015.100.20553
62. O'Leary-Roseberry, T., Villa, U.: hippyflow: Dimension reduced surrogate construction for parametric PDE maps in Python (2021). https://doi.org/10.5281/zenodo.4608729
63. Villa, U., Petra, N., Ghattas, O.: hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs; Part I: Deterministic inversion and linearized Bayesian inference. ACM Trans, Math, Softw. (2021)
64. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pp. 265–283 (2016)
65. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
66. O'Leary-Roseberry, T., Alger, N., Ghattas, O.: Inexact Newton methods for stochastic nonconvex optimization with applications to neural network training. arXiv preprint arXiv:1905.06738 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.