Journal of Machine Learning Research 26 (2025) 1-68 Submitted 5/24; Revised 2/25; Published 4/25

Derivative-Informed Neural Operator Acceleration of Geometric MCMC for Infinite-Dimensional Bayesian Inverse Problems

Lianghao Cao* LIANGHAO@CALTECH.EDU
Department of Computing and Mathematical Sciences
California Institute of Technology
Pasadena, CA 91125, USA.

Thomas O'Leary-Roseberry† TOM.OLEARYROSEBERRY@UTEXAS.EDU
Omar Ghattas†,‡ OMAR@ODEN.UTEXAS.EDU
† *Oden Institute for Computational Engineering and Sciences*
‡ *Walker Department of Mechanical Engineering*
The University of Texas at Austin
Austin, TX 78712, USA.

Editor: Animashree Anandkumar

Abstract

We propose an operator learning approach to accelerate geometric Markov chain Monte Carlo (MCMC) for solving infinite-dimensional Bayesian inverse problems (BIPs). While geometric MCMC employs high-quality proposals that adapt to posterior local geometry, it requires repeated computations of gradients and Hessians of the log-likelihood, which becomes prohibitive when the parameter-to-observable (PtO) map is defined through expensive-to-solve parametric partial differential equations (PDEs). We consider a delayed-acceptance geometric MCMC method driven by a neural operator surrogate of the PtO map, where the proposal exploits fast surrogate predictions of the log-likelihood and, simultaneously, its gradient and Hessian. To achieve a substantial speedup, the surrogate must accurately approximate the PtO map and its Jacobian, which often demands a prohibitively large number of PtO map samples via conventional operator learning methods. In this work, we present an extension of derivative-informed operator learning [O'Leary-Roseberry et al., *J. Comput. Phys.*, 496 (2024)] that uses joint samples of the PtO map and its Jacobian. This leads to derivative-informed neural operator (DINO) surrogates that accurately predict the observables and posterior local geometry at a significantly lower training cost than conventional methods. Cost and error analysis for reduced basis DINO surrogates are provided. Numerical studies demonstrate that DINO-driven MCMC generates effective posterior samples 3–9 times faster than geometric MCMC and 60–97 times faster than prior geometry-based MCMC. Furthermore, the training cost of DINO surrogates breaks even compared to geometric MCMC after just 10–25 effective posterior samples.

Keywords: Inverse problem, scientific machine learning, uncertainty quantification, MCMC, neural operator

*. Corresponding author

1. Introduction

1.1 Quality–cost trade-off in MCMC for Bayesian inverse problems

Continuum models of physical systems arising in scientific and engineering problems, such as those governed by partial differential equations (PDEs), often contain unspecified or uncertain parameters in the form of spatially and temporally varying functions. Given sparse and noisy observations of the system, infinite-dimensional inverse problems aim to infer the parameter function at which model predictions of observations (i.e., the *observables*) best explain the observed data. A Bayesian formulation is often adopted to rigorously account for various uncertainties in inverse problems, whose solutions are represented as probability distributions of parameters (i.e., the *posterior*). Bayesian inverse problems (BIPs) are of great practical importance due to their ability to enhance the predictability and reliability of computational models for better design, control, and more general decision-making (Biegler et al., 2010; Oden et al., 2017; Kouri and Shapiro, 2018; Ghattas and Willcox, 2021; Huan et al., 2024).

Markov chain Monte Carlo (MCMC) methods based on the Metropolis–Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) construct Markov chains whose stationary distributions are the Bayesian posterior (Robert and Casella, 2004; Roberts and Rosenthal, 2004). These methods are regarded as the gold standard for rigorous solutions of nonlinear BIPs due to their algorithmic simplicity and asymptotic posterior sampling consistency. A fundamental challenge in designing an efficient MCMC method for BIPs is to optimize the balance between the quality of generated Markov chains and the associated computational cost. The quality of a Markov chain can be quantified by, e.g., its effective sample size and mixing time. The computational cost consists of proposal sampling and acceptance probability computation at each chain position; the latter often involves evaluating the nonlinear *parameter-to-observable (PtO) map* via solving large-scale parametric PDEs.

To generate high-quality Markov chains for posterior sampling, an MCMC method must be capable of agile exploration of the posterior landscape. This often requires intelligent MCMC proposal designs using either (i) the local curvature of the posterior landscape or (ii) a surrogate PtO map. These two approaches represent two major but mostly distinct developments of MCMC methods for infinite-dimensional BIPs: (i) *geometric MCMC* methods with proposals that adapt to posterior local geometry (Girolami and Calderhead, 2011; Martin et al., 2012; Law, 2014; Bui-Thanh and Girolami, 2014; Lan et al., 2016; Beskos et al., 2017; Lan, 2019) and (ii) *delayed-acceptance (DA) MCMC* methods with proposals informed by the surrogate posterior (Christen and Fox, 2005; Efendiev et al., 2006; Lykkegaard et al., 2023). However, high quality comes at a high cost. For geometric MCMC, exploiting posterior local geometry requires computing the *Jacobian* of the PtO map through solutions of sensitivity or adjoint problems of the PDEs at each chain position. In practice, the accumulated cost of these linear solves often overwhelms their benefit. Moreover, for DA MCMC, constructing a data-driven surrogate of the nonlinear PtO map with infinite-dimensional parameter space may necessitate a large number of offline (prior to MCMC) model solutions to achieve a substantial online (during MCMC) acceleration.

Remark 1 *Here, we use Jacobian as a generic term that refers to the derivative of the observable vector with respect to the parameter function. However, there are multiple definitions of differentiability for nonlinear mappings in function spaces. We provide precise names and definitions for the derivatives and Jacobians of the PtO map in Sections 2.3 and 3.3.*

1.2 Geometric MCMC driven by derivative-informed neural operator

We consider the use of neural operator surrogates of the PtO map to design MCMC methods based on the following observation: an ideal quality–cost trade-off in an MCMC method can be achieved by a surrogate PtO map that is fast and accurate in predicting both the observables (for DA MCMC) and the posterior local geometry (for geometric MCMC). Neural operators (Kovachki et al., 2023, 2024), i.e., nonlinear mappings between function spaces constructed using neural networks, have the potential to provide a good approximation of the PtO map in infinite-dimensional BIPs. Notably, the Jacobian of a neural operator can be extracted through automatic differentiation at a low cost.

On the other hand, conventional operator learning methods using input–output samples of the target mapping (i.e., supervised learning) do not enforce direct control of the Jacobian approximation error; thus, the training cost for achieving a small Jacobian approximation error can be prohibitively high for large-scale PDE-constrained target mappings. As a result, neural operator surrogates often struggle to accelerate gradient-based optimization in high or infinite dimensions, where the surrogate-predicted gradient of the optimization objective function substitutes the model-predicted gradient; see, e.g., Luo et al. (2023, Section 4.2.2). Similarly, we expect the neural operator surrogate constructed via conventional operator learning to struggle in accelerating geometric MCMC.

In this work, we propose an efficient geometric MCMC method leveraging derivative-informed neural operator (DINO, O'Leary-Roseberry et al. 2024). Compared to conventional operator learning with input–output error control, derivative-informed operator learning additionally enforces Jacobian error control. In the setting of BIPs, we generate samples of the PtO map and its Jacobian to train a DINO PtO map surrogate during the offline phase. This surrogate can achieve significantly higher accuracy in predicting both the observables and the Jacobian at a similar training cost as conventional operator learning methods. During the online phase, we deploy the trained DINO surrogate in a delayed-acceptance geometric MCMC method, where both the DINO prediction of posterior local geometry and the DINO PtO map contribute to generating high-quality Markov chains for posterior sampling at a considerably lower cost than conventional geometric MCMC methods.

We provide rigorous comparisons of our method with various baseline MCMC methods for solving challenging BIPs, such as coefficient inversion for a nonlinear diffusion–reaction PDE and inference of a heterogeneous hyperelastic material property. Our numerical results show that

1. DINO-driven MCMC generates effective posterior samples 60–97 times faster than MCMC based on prior geometry (precondition Crank–Nicolson, Cotter et al. 2013), 3–9 times faster than geometric MCMC, and 18–40 times faster than using a conventionally trained neural operator surrogate. When accounting for the training sample generation cost, the

training cost of DINO surrogates breaks even after collecting just 10–25 effective posterior samples compared to geometric MCMC.

2. Derivative-informed operator learning achieves a surrogate accuracy of the PtO map Jacobian similar to the surrogate accuracy of the PtO map itself. This is achieved at 16–25 times lower cost in training sample generation than the conventional operator learning method. In our nonlinear diffusion–reaction numerical examples, we observe an estimated 166 times difference in training sample generation cost between the two operator learning methods to achieve an acceleration of geometric MCMC measured by the speed of effective posterior sample generation.

1.3 Literature review

In this subsection, we cover literature relevant to our work on neural operator surrogates and MCMC for BIPs.

1.3.1 Neural operator surrogates

Constructing neural operator surrogates involves (i) using neural networks to design an architecture that maps between function spaces and allows for sufficient expressivity and (ii) approximating a target mapping by training the neural networks via supervised or semi-supervised learning. The key feature of a neural operator is that its architecture and learning scheme are independent of any particular discretization of the input and output space. We use the term neural operator when input or output belongs to a function space, as it necessitates a neural operator architecture and learning scheme. In this subsection, we briefly overview concepts related to neural operator surrogates relevant to our work.

Neural operator architectures. The architecture most relevant to this work is reduced basis neural operators that use neural networks to learn the finite-dimensional nonlinear mapping between coefficients of input and output reduced bases. Choices of reduced bases include but are not limited to (i) proper orthogonal decomposition (POD-NN, Hesthaven and Ubbiali 2018) or principal component analysis (PCA-Net, Bhattacharya et al. 2021), (ii) active subspace or derivative-informed subspace (DIP-Net, O'Leary-Roseberry et al. 2022a,b), (iii) learned neural network representation of output reduced bases (DeepONet, Lu et al. 2021), and (iv) variational auto-encoders (VANO, Seidman et al. 2023). Other architectures include the Fourier neural operator (FNO, Li et al. 2021) and its variants (Cao et al., 2024b; Lanthaler et al., 2024; Li et al., 2020a,b). See empirical comparisons of neural operator architectures for learning solution operators of PDEs by de Hoop et al. (2022); Lu et al. (2022).

Operator learning objective. The operator learning objective function is typically designed to control approximate error in the Bochner norm of nonlinear mappings between function spaces (Kovachki et al., 2023). When the objective function is approximated using samples, it leads to a loss function for empirical risk minimization. There are efforts to enhance the loss function using spatial evaluations of strong-form PDE residual, notably for FNO (PINO, Li et al. 2024) and DeepONet (PI-DeepONet, Wang et al. 2021) architectures. The focus of this work is DINO (O'Leary-Roseberry et al., 2024). Its operator learning

objective function is designed to control approximation error in high-dimensional Sobolev spaces of nonlinear mappings. This objective leads to neural operator surrogates that are accurate in both input–output and Jacobian evaluations. DINOs have been successfully deployed in optimization under uncertainty (Luo et al., 2023), optimal experimental design (Go and Chen, 2025, 2024), and surrogate-driven measure transport (Cao et al., 2024a), demonstrating notable improvements compared to conventional operator learning methods. Recent work successfully applied derivative-informed learning to the DeepONet architecture (Qiu et al., 2024).

Jacobian vs. spatial derivative. We emphasize the distinction between the Jacobian of the neural operator and the spatial derivative of neural operator output. Evaluating the Jacobian requires differentiating the nonlinear mappings from a parameter to the PDE solution. It often requires repeatedly solving direct or adjoint sensitivity problems with different right-hand side vectors (Ghattas and Willcox, 2021). Controlling approximation error in the Jacobian is challenging in the operator learning setting. On the other hand, controlling approximation error in the spatial derivatives of spatially varying output functions can be implemented straightforwardly using a Sobolev norm over the spatial domain (e.g., $H^1(\Omega)$ norm where Ω is a spatial domain) for output error measure.

1.3.2 MCMC for Bayesian inverse problems

We briefly overview three aspects of MCMC methods for infinite-dimensional BIPs relevant to this work. Technical descriptions of some of these methods can be found in Section 2.

Scalability. During computation, the posterior is approximated on a discretized finite-dimensional subspace of the parameter function space via, e.g., the Galerkin method. Many popular MCMC methods that target the discretized posterior, such as random walk Metropolis, suffer from a deterioration in sampling performance as the discretization dimension increases. A class of *dimension-independent MCMC methods* has emerged (Cotter et al., 2013; Hairer et al., 2014; Law, 2014; Cui et al., 2016; Bui-Thanh and Nguyen, 2016; Rudolf and Sprungk, 2018) that seeks first to design MCMC methods that are well-posed on function spaces and then discretize for computation.

Exploiting posterior geometry. A class of *geometric MCMC* methods gained attention in the last decade due to their information geometric approaches to proposal design. Relevant developments in this area include but are not limited to (i) proposals employing fixed or averaged posterior geometry, such as likelihood-informed subspace (Cui et al., 2016), active subspace (Constantine et al., 2016), variational and Laplace approximation (Pinski et al., 2015; Rudolf and Sprungk, 2018; Petra et al., 2014; Kim et al., 2023), and adaptive dimension reduction (Lan, 2019); (ii) proposals that adapt to posterior local geometry, such as Riemannian manifold MCMC using the Jacobian and Hessian of the PtO map (Girolami and Calderhead, 2011; Bui-Thanh and Girolami, 2014), stochastic Newton MCMC using the Jacobian and the low-rank approximation of the Hessian (Martin et al., 2012), Gaussian process emulation of geometric quantities (Lan et al., 2016), and dimension-independent geometric MCMC using the Jacobian, i.e., using a simplified manifold (Law, 2014; Beskos et al., 2017; Lan, 2019). Other notable developments include proposal designs using lo-

cal likelihood approximations (Patrick R. Conrad and Smith, 2016) and transport maps constructed by low-fidelity model solutions (Peherstorfer and Marzouk, 2019).

Multi-fidelity acceleration. Cheap-to-evaluate low-fidelity models can help alleviate the cost of MCMC due to high-fidelity model solutions via *delayed acceptance (DA) MCMC* (Christen and Fox, 2005; Efendiev et al., 2006; Lykkegaard et al., 2023). It uses a proposal given by the Markov chain transition rule of an MCMC method targeting the surrogate posterior, leading to a two-stage (Christen and Fox, 2005) or multi-stage (Lykkegaard et al., 2023) procedure for single or multiple surrogate models. *Multilevel MCMC* methods (Hoang et al., 2013; Latz et al., 2018; Dodwell et al., 2019; Cui et al., 2024) employ a hierarchy of discretizations of a PDE model to reduce the overall computational cost of MCMC.

1.4 Contributions

The main contributions of this work are summarized as follows.

Formulation and analysis of DINO. We present an extended formulation of the derivative-informed operator learning proposed by O'Leary-Roseberry et al. (2024). In particular, we establish suitable function space settings, i.e., the H^1_μ Sobolev space with Gaussian measure (Section 3.2), for derivative-informed operator learning that can be extended beyond the confines of inverse problems and particular choices of neural operator architecture. We also provide (i) a cost analysis for training data generation based on PDE models (Section 4.3) and (ii) theoretical results on the neural operator approximation error of reduced basis DINO surrogate based on a Poincaré inequality for nonlinear mappings between function spaces (Section 4.4).

Efficient DINO-driven geometric MCMC. We propose an efficient MCMC method (Algorithm 1) for infinite-dimensional BIPs via a synthesis of ideas from reduced basis DINO surrogates, DA MCMC, and dimension-independent geometric MCMC. The method employs a proposal that adapts to DINO-predicted posterior local geometry within a delayed acceptance procedure. Compared to conventional geometric MCMC, our method leads to significant cost reduction due to (i) no online forward or adjoint sensitivity solves, (ii) fewer online PDE solves necessary for posterior consistency, and (iii) reduced need for prior sampling. At the same time, our numerical examples show that the method produces high-quality Markov chains typical of a geometric MCMC method, leading to substantial speedups in posterior sampling.

Detailed numerical studies and open-sourced software. We provide detailed numerical studies of our methods and other baseline MCMC methods using two infinite-dimensional BIPs: coefficient inversion for a nonlinear diffusion-reaction PDE and inference of a heterogeneous hyperelastic material property. The software and implementation of the numerical studies are publicly available in the following GitHub repository:

`https://github.com/dinoSciML/geometric_mcmc`.

1.5 Layout of the paper

In Section 2, we introduce concepts in BIPs and MCMC, including precision definitions of differentiability, posterior local approximation, dimension-independent geometric MCMC, and delayed acceptance MCMC. In Section 3, we present a derivative-informed operator learning method with error control in the H^1_μ Sobolev space with Gaussian measure. In Section 4, we formulate the derivative-informed training of reduced basis DINO, discuss its computational cost, and provide error analysis for different choices of reduced bases. In Section 5, we detail the process of generating proposals that adapt to posterior local geometry with a trained neural operator surrogate and the resulting MCMC acceptance probability computation. In Section 6, we explain the setup for our numerical examples, including an extensive list of baseline and reference MCMC methods, Markov chain diagnostics, efficiency metrics, and software. In Sections 7 and 8, we showcase and analyze numerical results. The appendices include detailed mathematical proofs and supplementary numerical results.

2. Preliminaries: Bayesian inverse problems and MCMC

2.1 Notations

- $\langle \cdot, \cdot \rangle_{\mathscr{X}}$ denotes the inner-product on a Hilbert space $\mathscr{X}$ and $\|\cdot\|_{\mathscr{X}}$ denotes the inner-product induced norm. The subscript is omitted when $\mathscr{X}$ is an Euclidean space.

- $\langle x_1, x_2 \rangle_{\mathcal{T}} \coloneqq \langle \mathcal{T}^{1/2} x_1, \mathcal{T}^{1/2} x_2 \rangle_{\mathscr{X}}$ and $\|x\|_{\mathcal{T}} \coloneqq \sqrt{\langle \mathcal{T}^{1/2} x, \mathcal{T}^{1/2} x \rangle_{\mathscr{X}}}$ denote the inner-product and norm weighted by a positive and self-adjoint operator $\mathcal{T} : \mathscr{X} \to \mathscr{X}$ and $\mathcal{T}^{1/2}$ denotes the square root of $\mathcal{T}$. The square root is not explicitly required during computation.

- $B(\mathscr{X}_1, \mathscr{X}_2)$ denotes the Banach space of bounded and linear operators between two Hilbert spaces $\mathscr{X}_1$ and $\mathscr{X}_2$ equipped with the operator norm. We use $B(\mathscr{X})$ for $B(\mathscr{X}, \mathscr{X})$.

- $\mathrm{HS}(\mathscr{X}_1, \mathscr{X}_2) \subseteq B(\mathscr{X}_1, \mathscr{X}_2)$ denotes the Banach space of Hilbert–Schmidt operators. $\|\cdot\|_{\mathrm{HS}}$ denotes the Hilbert–Schmidt norm. We use $\mathrm{HS}(\mathscr{X})$ for $\mathrm{HS}(\mathscr{X}, \mathscr{X})$.

- $B_1^+(\mathscr{X}) \subseteq \mathrm{HS}(\mathscr{X})$ denotes the set of positive, self-adjoint, and trace class operators on a Hilbert space $\mathscr{X}$.

- $(\mathscr{X}, \mathcal{B}(\mathscr{X}))$ denotes a measurable space with $\mathcal{B}(\cdot)$ being the Borel σ-algebra generated by open sets. $\mathscr{P}(\mathscr{X})$ denotes the set of probability measures defined on $(\mathscr{X}, \mathcal{B}(\mathscr{X}))$.

- $\nu(\mathrm{d}x)$ denotes a measure on $(\mathscr{X}, \mathcal{B}(\mathscr{X}))$ in the sense that $\nu(\mathscr{A}) = \int_{\mathscr{A}} \nu(\mathrm{d}x)$, where $\mathscr{A} \in \mathcal{B}(\mathscr{X})$ and x is a dummy variable for integration. The expression $\nu_1(\mathrm{d}x)/\nu_2(\mathrm{d}x)$ denotes the Radon–Nikodym derivative between two measures $\nu_1, \nu_2 \in \mathscr{P}(\mathscr{X})$ at $x \in \mathscr{X}$.

- We use capital letters to denote random variables, i.e., $X \sim \nu \in \mathscr{P}(\mathscr{X})$. Both matrices and random vectors are denoted using bold and capitalized letters; they can be distinguished based on the context.

2.2 Nonlinear Bayesian inverse problem

Let $\mathscr{M}$ be a separable Hilbert space. We refer to $\mathscr{M}$ as the *parameter space*. Let $\boldsymbol{\mathcal{G}} : \mathscr{M} \to \mathbb{R}^{d_y}$ be a nonlinear *parameter-to-observable (PtO) map* that represents model predictions of the *observable vector*. We refer to the $\mathbb{R}^{d_y}$, $d_y \in \mathbb{N}$, as the *observable space*. Let $\boldsymbol{y} \in \mathbb{R}^{d_y}$ denote a set of observed data. We assume that $\boldsymbol{y}$ is given by a model-predicted observable vector at unknown parameter $m \in \mathscr{M}$ corrupted by unknown additive noise $\boldsymbol{n} \in \mathbb{R}^{d_y}$:

$$\boldsymbol{y} = \boldsymbol{\mathcal{G}}(m) + \boldsymbol{n}\,, \quad \boldsymbol{n} \overset{\text{i.i.d.}}{\sim} \pi_n\,, \qquad \text{(Data model)} \tag{1}$$

where $\pi_n \in \mathscr{P}(\mathbb{R}^{d_y})$ is the noise probability density[1]. The inverse problem is to recover m given data $\boldsymbol{y}$.

Under the Bayesian approach to inverse problems, we assume prior knowledge of the parameter represented by a *prior distribution* $\mu \in \mathscr{P}(\mathscr{M})$. We are interested in characterizing the *posterior distributions* $\mu^{\boldsymbol{y}} \in \mathscr{P}(\mathscr{M})$ representing our updated knowledge of the parameter after acquiring data $\boldsymbol{y}$. The posterior is defined by Bayes' rule using a Radon–Nikodym (RN) derivative:

$$\frac{\mu^{\boldsymbol{y}}(\mathrm{d}m)}{\mu(\mathrm{d}m)} \propto \pi_n(\boldsymbol{y} - \boldsymbol{\mathcal{G}}(m))\,. \qquad \text{(Bayes' rule)} \tag{2}$$

We adopt the following assumptions, often employed for infinite-dimensional BIPs, e.g., when $\mathscr{M}$ consists of spatially or temporally varying functions.

Assumption 1 (Gaussian noise) *The noise distribution is given by $\pi_n \coloneqq \mathcal{N}(\mathbf{0}, \boldsymbol{C}_n)$ with covariance matrix $\boldsymbol{C}_n \in B_1^+(\mathbb{R}^{d_y})$. This leads to the following form of the Bayes' rule:*

$$\frac{\mu^{\boldsymbol{y}}(\mathrm{d}m)}{\mu(\mathrm{d}m)} \coloneqq \frac{1}{z(\boldsymbol{y})} \exp\left(-\Phi^{\boldsymbol{y}}(m)\right)\,, \quad \Phi^{\boldsymbol{y}}(m) \coloneqq \frac{1}{2}\left\|\boldsymbol{y} - \boldsymbol{\mathcal{G}}(m)\right\|^2_{\boldsymbol{C}_n^{-1}} \quad \mu\text{-a.e.}\,, \tag{3}$$

where $\Phi^{\boldsymbol{y}} : \mathscr{M} \to \mathbb{R}$ is the data misfit and $z(\boldsymbol{y}) \coloneqq \mathbb{E}_{M\sim\mu}\left[\Phi^{\boldsymbol{y}}(M)\right]$ is the normalization constant.

Assumption 2 (Gaussian prior) *The prior distribution is given by $\mu \coloneqq \mathcal{N}(0, \mathcal{C}_{\mathrm{pr}})$ with covariance operator $\mathcal{C}_{\mathrm{pr}} \in B_1^+(\mathscr{M})$.*

Assumption 3 (Well-posedness, Stuart 2010, Corollary 4.4) *The PtO map $\boldsymbol{\mathcal{G}}$ is μ-a.e. well-defined, sufficiently bounded, and locally Lipshitz continuous, which implies that the Bayesian inversion is well-posed.*

2.3 The Cameron–Martin space and differentiability

We consider two Hilbert spaces with prior and noise covariance inverse-weighted inner products on the observable and parameter space. These spaces are known as the Cameron–Martin (CM) space of μ and π_n.

$$\text{Parameter CM space } \mathscr{H}_\mu : \quad \left(\left\{m \in \mathscr{M} \;\middle|\; \|m\|_{\mathcal{C}_{\mathrm{pr}}^{-1}} < \infty\right\}, \langle\cdot,\cdot\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}}\right),$$

$$\text{Observable CM space } \mathscr{Y} : \quad \left(\mathbb{R}^{d_y}, \langle\cdot,\cdot\rangle_{\boldsymbol{C}_n^{-1}}\right).$$

1. For finite-dimensional distributions, we assume their probability densities exist and do not distinguish between densities and measures.

We have the continuous embedding $\mathscr{H}_\mu \hookrightarrow \mathscr{M}$, i.e., there exists $c > 0$ such that $\|m\|_{\mathscr{M}} \leq c \|m\|_{\mathcal{C}_{\text{pr}}^{-1}}$ for all $m \in \mathscr{H}_\mu$. However, the converse is not true when $\mathscr{M}$ has infinite dimensions, e.g., $\|M\|_{\mathcal{C}_{\text{pr}}^{-1}} = \infty$ and $\|M\|_{\mathscr{M}} < \infty$ a.s. for $M \sim \mu$. The observable CM space $\mathscr{Y}$ is isomorphic to $\mathbb{R}^{d_y}$ under the identity map, yet the weighted inner product of $\mathscr{Y}$ is often preferred in the context of inverse problems as it reflects our confidence in the observed data.

The CM space plays a major role in understanding the equivalence of measures due to the linear transformations of infinite-dimensional Gaussian random functions, and it will be applied extensively in this work. We refer to Sullivan (2015, Section 2.7), Bogachev (1998), and Stuart (2010) for detailed references on Gaussian measures and the CM space.

In addition to Assumptions 1 to 3, we assume the directional differentiability of the PtO map along the parameter CM space $\mathscr{H}_\mu$.

Assumption 4 (stochastic Gâteaux differentiability, Bogachev 1998, 5.2.3) *There exists a mapping $D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}} : \mathscr{M} \to \text{HS}(\mathscr{H}_\mu, \mathscr{Y})$ such that for any $\delta m \in \mathscr{H}_\mu$,*

$$\lim_{t\to 0} \left\| t^{-1} \left(\boldsymbol{\mathcal{G}}(m + t\delta m) - \boldsymbol{\mathcal{G}}(m) \right) - D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\delta m \right\|_{\boldsymbol{C}_n^{-1}} = 0 \quad \mu\text{-a.e.}$$

The mapping $D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}$ is called the stochastic derivative of $\boldsymbol{\mathcal{G}}$.

The stochastic Gâteaux differentiability in Assumption 4 is weaker than the typical Gâteaux differentiability assumption that requires directional differentiability along the whole parameter space $\mathscr{M}$ given in Definition 2.

Definition 2 (μ-a.e. Gâteaux differentiability) *There exists a mapping $D\boldsymbol{\mathcal{G}} : \mathscr{M} \to \text{HS}(\mathscr{M}, \mathbb{R}^{d_y})$ such that for any $\delta m \in \mathscr{M}$,*

$$\lim_{t\to 0} \left\| t^{-1} \left(\boldsymbol{\mathcal{G}}(m + t\delta m) - \boldsymbol{\mathcal{G}}(m) \right) - D\boldsymbol{\mathcal{G}}(m)\delta m \right\| = 0 \quad \mu\text{-a.e.}$$

The mapping $D\boldsymbol{\mathcal{G}}$ is called the Gâteaux derivative of $\boldsymbol{\mathcal{G}}$.

Suppose $D\boldsymbol{\mathcal{G}}$ exists, we have $D\boldsymbol{\mathcal{G}}(m)|_{\mathscr{H}_\mu} = D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)$ μ-a.e. Additionally, the stochastic derivative carries over the parameter regularity given by the prior distribution μ, making it the more natural derivative definition for BIPs; see Appendix A. Stochastic differentiability is used in our derivative-informed operator learning formulation; see Section 3.2. The stochastic derivative is often called the Malliavin derivative (Nualart and Nualart, 2018) or the H-derivative (Kac and Cheung, 2002) in different contexts.

2.4 Local Gaussian approximation of the posterior

We consider a linear expansion of the nonlinear PtO map $\boldsymbol{\mathcal{G}}$ at a given $m \in \mathscr{M}$:

$$\boldsymbol{\mathcal{G}}(\cdot) \approx \boldsymbol{\mathcal{G}}(m) + D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)(\cdot - m)\,.$$

Replacing the PtO map in (3) using the linear expansion, we obtain a local Gaussian approximation to the posterior in closed form (Stuart, 2010, Section 6.4). It is a conditional probability distribution $\mathcal{Q}_{\text{local}} : \mathscr{M} \times \mathcal{B}(\mathscr{M}) \to [0, 1]$ given by:

$$\mu^{\boldsymbol{y}} \approx \mathcal{Q}_{\text{local}}(m, \cdot) = \mathcal{N}(-D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m), \mathcal{C}_{\text{post}}(m))\,, \tag{4}$$

where the negative mean $D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}} : \mathscr{M} \to \mathscr{H}_\mu$ is the $\mathscr{H}_\mu$-Riesz representation of the stochastic derivative of the data misfit,

$$D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m) \coloneqq D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^* \left(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}\right) , \tag{5}$$

and the covariance $\mathcal{C}_{\text{post}} : \mathscr{M} \to B_1^+(\mathscr{M})$ is given by

$$\mathcal{C}_{\text{post}}(m) \coloneqq (\mathcal{I}_{\mathscr{H}_\mu} + \mathcal{H}(m))^{-1}\mathcal{C}_{\text{pr}} , \quad \mathcal{H}(m) \coloneqq D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^* D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m) , \tag{6}$$

where $\mathcal{I}_{\mathscr{H}_\mu}$ is the identity map on $\mathscr{H}_\mu$, and $\mathcal{H} : \mathscr{M} \to B_1^+(\mathscr{H}_\mu)$ is a Gauss–Newton approximation to the stochastic Hessian of the data misfit in (53). The mappings $D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}$ and $\mathcal{H}$ are often known as the *prior-preconditioned gradient (ppg)* and the *prior-preconditioned Gauss-Newton Hessian (ppGNH)* in the context of inverse problems. Their connections to conventional definitions are shown in Appendix B.

2.5 The Metropolis–Hastings algorithm

We use the *Metropolis–Hastings (MH) algorithm* to sample from the posterior $\mu^{\boldsymbol{y}}$ defined over infinite-dimensional $\mathscr{M}$. The MH algorithm is a procedure for generating reversible Markov chains $\{m_j \in \mathscr{M}\}_{j=1}^{\infty}$ with a stationary distribution of $\mu^{\boldsymbol{y}}$. The algorithm prescribes a Markov chain transition rule (i.e., the law of $m_j \mapsto m_{j+1}$) based on a proposal distribution and an accept-reject move. The proposal, denoted by $\mathcal{Q} : \mathscr{M} \times \mathcal{B}(\mathscr{M}) \to [0, 1]$, is a conditional probability distribution. The proposal and the posterior jointly define a set of transition rates[2] between two positions in $\mathscr{M}$ as measures (unnormalized) on the product space $\mathscr{M} \times \mathscr{M}$:

$$\nu(\,\mathrm{d}m,\, \mathrm{d}m^\dagger) \coloneqq \mathcal{Q}(m,\, \mathrm{d}m^\dagger)\mu^{\boldsymbol{y}}(\,\mathrm{d}m) , \quad \nu^T(\,\mathrm{d}m,\, \mathrm{d}m^\dagger) \coloneqq \mathcal{Q}(m^\dagger,\, \mathrm{d}m)\mu^{\boldsymbol{y}}(\,\mathrm{d}m^\dagger) .$$

The ratio of transition rates, $\rho : \mathscr{M} \times \mathscr{M} \to [0, \infty)$, is given by an RN derivative:

$$\rho(m, m^\dagger) \coloneqq \frac{\nu^T(\mathrm{d}m, \mathrm{d}m^\dagger)}{\nu(\mathrm{d}m, \mathrm{d}m^\dagger)} = \frac{\mathcal{Q}(m^\dagger,\, \mathrm{d}m)\exp(-\Phi^{\boldsymbol{y}}(m^\dagger))\mu(\,\mathrm{d}m^\dagger)}{\mathcal{Q}(m,\, \mathrm{d}m^\dagger)\exp(-\Phi^{\boldsymbol{y}}(m))\mu(\,\mathrm{d}m)} , \tag{7}$$

where we use a change of measure in (2) to represent $\mu^{\boldsymbol{y}}$.

The accept-reject move is executed as follows. At a chain position $m_j \in \mathscr{M}$, we sample a proposed move $m^\dagger \overset{\text{i.i.d.}}{\sim} \mathcal{Q}(m_j, \cdot)$. The next position is set to $m_{j+1} = m^\dagger$, i.e., acceptance, with probability $\alpha(m_j, m^\dagger) \in [0, 1]$ given by

$$\alpha(m_j, m^\dagger) \coloneqq \min\{1, \rho(m_j, m^\dagger)\} .$$

Alternatively, we set $m_{j+1} = m_j$, i.e., rejection, with probability $1-\alpha(m_j, m^\dagger)$. See Figure 1 (*left*) for a schematic of the MH algorithm.

2.6 Dimension-independent MCMC

A dimension-independent MCMC method using the MH algorithm employs a proposal with a well-defined transition rate ratio ρ in (7). According to the RN theorem (Sullivan, 2015,

2. These transition rates should not be confused with the Markov chain transition rule of the MH algorithm.

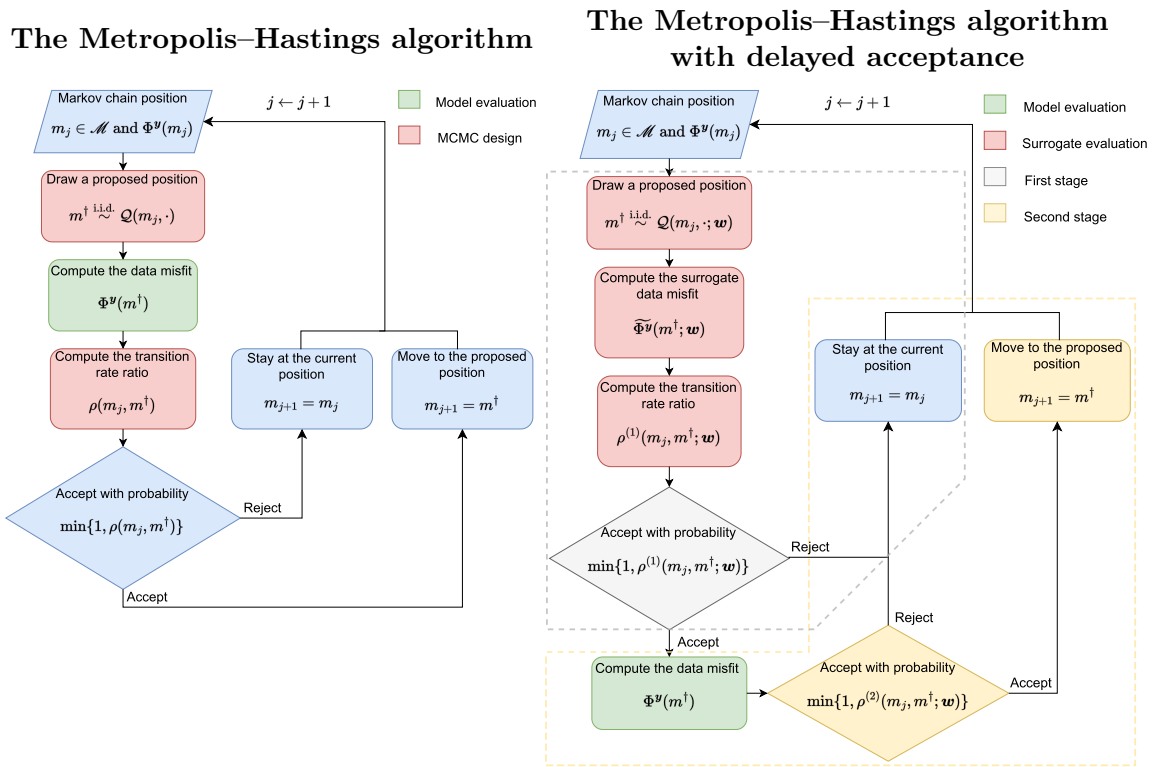

Figure 1: (*left*) A schematic of the MH algorithm for sampling from the posterior distribution $\mu^{\boldsymbol{y}}$ as described in Section 2.5. (*right*) A schematic of the MH algorithm with delayed acceptance enabled by a surrogate PtO map $\widetilde{\boldsymbol{\mathcal{G}}}(\cdot; \boldsymbol{w})$ parameterized by $\boldsymbol{w}$. See Section 2.8 for a detailed description of the components of this algorithm.

Theorem 2.29), ρ is well-defined if $\nu^T \ll \nu$, where $\ll$ denotes the absolute continuity of measures. In particular, Tierney (1998) shows that MH rejects all proposed moves if ν^T and ν are mutually singular. In finite dimensions, $\nu^T \ll \nu$ holds for most proposal choices (e.g., Gaussian random walk), and one typically expresses ν and ν^T as probability densities. However, measures on infinite-dimensional spaces tend to be mutually singular (see, e.g., Sullivan 2015, Theorem 2.51), and their probability densities do not exist (Sullivan, 2015, Theorem 2.38). As a result, a finite-dimensional MH algorithm targeting a discretized infinite-dimensional sampling problem often leads to deteriorating sampling performance when the discretization is refined (Hairer et al., 2014). For example, the conventional Gaussian random walk proposal given by $\mathcal{Q}_{\text{RW}}(m, \cdot) := \mathcal{N}(m, s\mathcal{C}_{\text{pr}})$, $s > 0$, fails to be dimension-independent because it leads to an ill-defined ρ in (7); see Stuart 2010, Example 5.3, Hairer et al. 2014, Section 2.4, and Rudolf and Sprungk 2018, Section 3.3.

The building block for a dimension-independent MCMC is the preconditioned Crank–Nicolson (pCN, Cotter et al. 2013) proposal, which is reversible with respect to the prior:

$$\mathcal{Q}_{\text{pCN}}(m,\cdot) := \mathcal{N}(sm,(1-s^2)\mathcal{C}_{\text{pr}})\,, \quad s\leq 1\,, \qquad \text{(The pCN proposal)} \tag{8a}$$

$$\mathcal{Q}_{\text{pCN}}(m^\dagger,\mathrm{d}m)\mu(\mathrm{d}m^\dagger) = \mathcal{Q}_{\text{pCN}}(m,\mathrm{d}m^\dagger)\mu(\mathrm{d}m)\,. \qquad \text{(Prior reversibility)} \tag{8b}$$

The equivalence of measure in (8b) leads to a well-defined transition rate ratio of the form:

$$\rho_{\text{pCN}}(m,m^\dagger) = \exp\left(\Phi^{\boldsymbol{y}}(m) - \Phi^{\boldsymbol{y}}(m^\dagger)\right)\,.$$

The pCN proposal is used for deriving the acceptance probability for proposals that include posterior local geometry (Beskos et al., 2017; Rudolf and Sprungk, 2018; Lan, 2019). In particular, these geometry-informed proposals are designed to possess well-defined and close-formed RN derivatives with respect to the pCN proposal. The existence of these RN derivatives leads to dimension-independent geometric MCMC methods. The closed forms of these RN derivatives make evaluations of the acceptance probability straightforward.

2.7 Geometric MCMC

We consider the simplified manifold Metropolis-adjusted Langevin algorithm, or mMALA, introduced by Beskos et al. 2017. It originates from the following Langevin stochastic differential equation (SDE) on $\mathscr{M}$, preconditioned by a position-dependent, positive, and self-adjoint trace class operator $\mathcal{K}:\mathscr{M}\to B_1^+(\mathscr{M})$:

$$\mathrm{d}M_t = -\frac{1}{2}\mathcal{K}(M_t)\mathcal{C}_{\text{pr}}^{-1}\left(M_t + D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(M_t)\right)\,\mathrm{d}t + \mathcal{K}(M_t)^{1/2}\,\mathrm{d}W_t\,, \quad t\geq 0\,, \tag{9}$$

where W_t is a cylindrical Wiener process on $\mathscr{M}$. This SDE can be derived using a local reference measure of the form:

$$\mathcal{Q}_{\text{local}}(m,\cdot) = \mathcal{N}(\mathcal{M}(m),\mathcal{K}(m))\,, \tag{10}$$

where $\mathcal{M}:\mathscr{M}\to\mathscr{H}_\mu$ outputs the mean of the reference, which does not appear in the SDE (Beskos et al., 2017, Section 3.1). We assume $\mathcal{K}(m)$ leads to the equivalence of measure between μ and $\mathcal{N}(0,\mathcal{K}(m))$ μ-a.e. Discretizing the above SDE in time using a semi-implicit Euler scheme with a step size $\triangle t\in\mathbb{R}_+$ leads to the following proposal:

$$\mathcal{Q}_{\text{mMALA}}(m,\cdot) \coloneqq \mathcal{N}\left(sm + (1-s)\,\mathcal{A}(m), \left(1-s^2\right)\mathcal{K}(m)\right)\,, \tag{11a}$$

$$\mathcal{A}(m) \coloneqq m - \mathcal{K}(m)\mathcal{C}_{\text{pr}}^{-1}\left(m + D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m)\right)\,, \quad s = \frac{4-\triangle t}{4+\triangle t}\,. \tag{11b}$$

The mMALA and pCN proposals are equivalent in measure μ-a.e. (Beskos et al., 2017, Theorem 3.5), and the mMALA proposal has a well-defined transition rate ratio $\rho_{\text{mMALA}}:\mathscr{M}\times\mathscr{M}\to\mathbb{R}_+$ with a closed form given by:

$$\begin{aligned}\rho_{\text{mMALA}}(m_1,m_2) &\coloneqq \exp\left(\Phi^{\boldsymbol{y}}(m_1) - \Phi^{\boldsymbol{y}}(m_2)\right)\frac{\mathcal{Q}_{\text{mMALA}}(m_2,\,\mathrm{d}m_1)\mu(\,\mathrm{d}m_2)}{\mathcal{Q}_{\text{mMALA}}(m_1,\,\mathrm{d}m_2)\mu(\,\mathrm{d}m_1)}\\ &= \exp\left(\Phi^{\boldsymbol{y}}(m_1) - \Phi^{\boldsymbol{y}}(m_2)\right)\frac{\rho_0(m_2,m_1)}{\rho_0(m_1,m_2)}\,,\end{aligned}$$

where $\rho_0 : \mathscr{M} \times \mathscr{M} \to \mathbb{R}_+$ is the RN derivative between the mMALA and pCN proposals:

$$\begin{aligned}\rho_0(m_1, m_2) &:= \frac{\mathcal{Q}_{\mathrm{mMALA}}(m_1, \mathrm{d}m_2)}{\mathcal{Q}_{\mathrm{pCN}}(m_1, \mathrm{d}m_2)} \\ &= \exp\left(-\frac{\triangle t}{8}\left\|\mathcal{A}(m_1)\right\|^2_{\mathcal{K}(m_1)^{-1}} + \frac{\sqrt{\triangle t}}{2}\left\langle \mathcal{A}(m_1), \widehat{m}\right\rangle_{\mathcal{K}(m_1)^{-1}}\right) \\ &\quad \times \det\nolimits_{\mathscr{M}}\left(\mathcal{C}_{\mathrm{pr}}^{1/2}\mathcal{K}(m_1)^{-1}\mathcal{C}_{\mathrm{pr}}^{1/2}\right)^{1/2} \exp\left(-\frac{1}{2}\left\|\widehat{m}\right\|^2_{\left(\mathcal{K}(m_1)^{-1}-\mathcal{C}_{\mathrm{pr}}^{-1}\right)}\right),\end{aligned} \tag{12}$$

where $\widehat{m} = (m_2 - s m_1)/\sqrt{1-s^2}$ and $\det_{\mathscr{M}} : B_1^+(\mathscr{M}) \to \mathbb{R}$ is the operator determinant given by eigenvalue product (Gohberg et al., 2012, Theorem 6.1).

The mMALA proposal allows flexibility in designing $\mathcal{K}(m)$ and removing or reducing the ppg $D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}$. For example, the mMALA proposal reduces to the pCN proposal in (8a) when $\mathcal{K}(m) = \mathcal{C}_{\mathrm{pr}}$ and the ppg is removed. We may choose $\mathcal{K}(m)$ such that the local reference measure adapts to the posterior local geometry at each Markov chain position, such as $\mathcal{K}(m) \approx \mathcal{C}_{\mathrm{post}}(m)$ using the local Gaussian approximation in (4). The proposals incorporating posterior local geometry lead to a class of MCMC methods referred to as *dimension-independent geometric MCMC methods*.

2.8 Delayed acceptance MCMC

The DA MCMC method follows the MH algorithm targeting $\mu^{\boldsymbol{y}}$ with a special choice of the proposal distribution: the Markov chain transition rule (i.e., the law of $m_j \to m_{j+1}$) of MH targeting the surrogate posterior $\widetilde{\mu^{\boldsymbol{y}}} \in \mathscr{P}(\mathscr{M})$ defined by a surrogate PtO map $\widetilde{\boldsymbol{\mathcal{G}}} \approx \boldsymbol{\mathcal{G}}$. The surrogate data misfit and posterior are given by

$$\widetilde{\Phi^{\boldsymbol{y}}}(m) := \frac{1}{2}\left\|\boldsymbol{y} - \widetilde{\boldsymbol{\mathcal{G}}}(m)\right\|^2_{\boldsymbol{C}_n^{-1}}, \qquad \text{(Surrogate data misfit)} \tag{13a}$$

$$\frac{\widetilde{\mu^{\boldsymbol{y}}}(\mathrm{d}m)}{\mu(\mathrm{d}m)} \propto \exp(-\widetilde{\Phi^{\boldsymbol{y}}}(m)). \qquad \text{(Surrogate posterior)} \tag{13b}$$

The DA procedure has two stages at each chain position m_j with a proposed move $m^\dagger \overset{\text{i.i.d.}}{\sim} \mathcal{Q}(m, \cdot)$.

1. A pass-reject move is performed based on the transition rate ratio in (7) using surrogate data misfit evaluations:

$$\rho^{(1)}(m_j, m^\dagger) = \frac{\mathcal{Q}(m^\dagger, \mathrm{d}m_j)\exp(-\widetilde{\Phi^{\boldsymbol{y}}}(m^\dagger))\mu(\mathrm{d}m^\dagger)}{\mathcal{Q}(m_j, \mathrm{d}m^\dagger)\exp(-\widetilde{\Phi^{\boldsymbol{y}}}(m_j))\mu(\mathrm{d}m_j)}.$$

   The proposed move $m^\dagger$ is passed to the second stage with probability $\alpha^{(1)}(m_j, m^\dagger) := \min\{1, \rho^{(1)}(m_j, m^\dagger)\}$. Otherwise, the second stage is skipped and set $m_{j+1} = m_j$ (i.e., rejection) with probability $1 - \alpha^{(1)}(m_j, m^\dagger)$.

2. An accept-reject move of MH is performed based on a proposal $\mathcal{Q}_{\mathrm{DA}}(m, \mathrm{d}m^\dagger)$ prescribed by the first stage pass-reject move:

$$\mathcal{Q}_{\mathrm{DA}}(m, \mathrm{d}m^\dagger) = \alpha^{(1)}(m_j, m^\dagger)\mathcal{Q}(m, \mathrm{d}m^\dagger) + \left(1 - \alpha^{(1)}(m_j, m^\dagger)\right)\delta_{m_j}(\mathrm{d}m^\dagger),$$

where δ_{m_j} is the Dirac mass concentrated on m_j. Since $\mathcal{Q}_{\mathrm{DA}}$ is reversible with respect to $\widetilde{\mu^{\boldsymbol{y}}}$, we have $\mathcal{Q}_{\mathrm{DA}}(m, \mathrm{d}m^\dagger)\widetilde{\mu^{\boldsymbol{y}}}(\mathrm{d}m) = \mathcal{Q}_{\mathrm{DA}}(m^\dagger, \mathrm{d}m)\widetilde{\mu^{\boldsymbol{y}}}(\mathrm{d}m^\dagger)$. As a result, the transition rate ratio for $\mathcal{Q}_{\mathrm{DA}}$ is given by

$$\rho^{(2)}(m_j, m^\dagger) = \frac{\exp(-\widetilde{\Phi^{\boldsymbol{y}}}(m_j))\exp(-\Phi^{\boldsymbol{y}}(m^\dagger))}{\exp(-\widetilde{\Phi^{\boldsymbol{y}}}(m^\dagger))\exp(-\Phi^{\boldsymbol{y}}(m_j))}. \tag{14}$$

See Figure 1 (right) for a schematic of DA MCMC.

The DA procedure allows proposed moves to be rejected solely based on surrogate evaluations in the first stage. This feature potentially significantly reduces the evaluation counts of the PtO map during posterior sampling. On the other hand, the efficiency of DA MCMC relies heavily on the quality of the surrogate approximation. When the surrogate PtO map is accurate, most rejections occur during the first stage without model solutions, and most proposed moves passed to the second stage are accepted. Higher surrogate approximation error leads to more frequent second-stage rejection, thus increasing the average computational cost per Markov chain sample and deteriorating posterior sampling efficiency. See Appendix C for additional discussion on surrogate approximation in DA MCMC.

3. Operator learning in H^1_μ Sobolev space with Gaussian measure

We consider an operator learning problem of optimizing the weight $\boldsymbol{w} \in \mathbb{R}^{d_w}$ of an operator surrogate $\widetilde{\boldsymbol{\mathcal{G}}}(\cdot; \boldsymbol{w}) : \mathscr{M} \to \mathscr{Y}$ so that $\widetilde{\boldsymbol{\mathcal{G}}}$ is close to the PtO map $\boldsymbol{\mathcal{G}}$ measured by certain metric. When the operator surrogate $\widetilde{\boldsymbol{\mathcal{G}}}$ is represented using a neural network, the weight $\boldsymbol{w}$ consists of the tunable parameters of neural networks. In Section 3.1, we introduce the conventional operator learning method. We present our derivative-informed operator learning method in Section 3.2. In Section 3.3, we discuss matrix representations of Hilbert–Schmidt operators for efficient operator learning.

3.1 Operator learning in L^2_μ Bochner space

The typical operator learning method approximates the PtO map in the $L^2_\mu(\mathscr{M}; \mathscr{Y})$ Bochner space, or L^2_μ for short. It is defined by:

$$L^2_\mu(\mathscr{M}; \mathscr{Y}) \coloneqq \left\{ \boldsymbol{\mathcal{T}} : \mathscr{M} \to \mathscr{Y} \;\middle|\; \|\boldsymbol{\mathcal{T}}\|_{L^2_\mu} < \infty \right\}, \qquad (L^2_\mu \text{ Definition})$$

$$\|\boldsymbol{\mathcal{T}}\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} \coloneqq \left(\mathbb{E}_{M\sim\mu}\left[\|\boldsymbol{\mathcal{T}}(M)\|^2_{\boldsymbol{C}_n^{-1}} \right] \right)^{1/2}. \qquad (L^2_\mu \text{ norm})$$

The operator learning objective function is designed to control the approximation error in $L^2_\mu(\mathscr{M}; \mathscr{Y})$:

$$\boldsymbol{w}^\ddagger = \operatorname*{arg\,min}_{\boldsymbol{w}\in\mathbb{R}^{d_w}} \mathcal{L}^\infty_{L^2_\mu}(\boldsymbol{w}), \qquad \text{(Operator learning objective)} \tag{15a}$$

$$\mathcal{L}^\infty_{L^2_\mu}(\boldsymbol{w}) \coloneqq \frac{1}{2}\left\| \boldsymbol{\mathcal{G}} - \widetilde{\boldsymbol{\mathcal{G}}}(\cdot; \boldsymbol{w}) \right\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})}. \qquad \text{(Error control in } L^2_\mu(\mathscr{M}; \mathscr{Y})) \tag{15b}$$

The objective $\mathcal{L}^{\infty}_{L^2_\mu}$ can be estimated via input–output pairs $\{m_j, \boldsymbol{\mathcal{G}}(m_j)\}_{j=1}^{n_t}$ with $m_j \overset{\text{i.i.d.}}{\sim} \mu$, which leads to a loss function $\mathcal{L}^{n_t}_{L^2_\mu}$ defined as follows:

$$\mathcal{L}^{\infty}_{L^2_\mu}(\boldsymbol{w}) \approx \mathcal{L}^{n_t}_{L^2_\mu}(\boldsymbol{w}; \{m_j\}_{j=1}^{n_t}) \coloneqq \frac{1}{2n_t} \sum_{j=1}^{n_t} \left\| \boldsymbol{\mathcal{G}}(m_j) - \widetilde{\boldsymbol{\mathcal{G}}}(m_j; \boldsymbol{w}) \right\|^2_{\boldsymbol{C}_n^{-1}} . \tag{16}$$

The operator surrogate can be constructed via finding $\boldsymbol{w}^{\ddagger}$ that minimizes the loss $\mathcal{L}^{n_t}_{L^2_\mu}$.

3.2 Operator learning in H^1_μ Sobolev space with Gaussian measure

In this work, we are interested in designing MCMC methods using operator surrogate that requires small approximation errors in both operator evaluations (for approximating ppg in (5) and efficient DA procedure) and its derivative evaluations (for approximating the ppg and ppGNH in (5) and (6)). Therefore, we consider controlling the operator surrogate error in the H^1 *Sobolev space with Gaussian measure*, or H^1_μ for short. It is a Hilbert space of nonlinear mappings with an inner product-induced norm that measures the distance between nonlinear mappings using *the discrepancy in their stochastic derivative evaluations* in addition to the discrepancy in the mappings:

$$H^1_\mu(\mathscr{M}; \mathscr{Y}) \coloneqq \left\{ \boldsymbol{\mathcal{T}} \in L^2_\mu(\mathscr{M}; \mathscr{Y}) \;\middle|\; \left\| D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{T}}(M) \right\|_{L^2_\mu(\mathscr{M}; \mathrm{HS}(\mathscr{H}_\mu, \mathscr{Y}))} < \infty \right\} , \qquad (H^1_\mu \text{ definition})$$

$$\|\boldsymbol{\mathcal{T}}\|_{H^1_\mu(\mathscr{M}; \mathscr{Y})} \coloneqq \left(\|\boldsymbol{\mathcal{T}}\|^2_{L^2_\mu(\mathscr{M}; \mathscr{Y})} + \left\| D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{T}} \right\|^2_{L^2_\mu(\mathscr{M}; \mathrm{HS}(\mathscr{H}_\mu, \mathscr{Y}))} \right)^{1/2} , \qquad (H^1_\mu \text{ norm})$$

$$\left\| D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{T}} \right\|_{L^2_\mu(\mathscr{M}; \mathrm{HS}(\mathscr{H}_\mu, \mathscr{Y}))} \coloneqq \left(\mathbb{E}_{M \sim \mu} \left[\left\| D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{T}}(M) \right\|^2_{\mathrm{HS}(\mathscr{H}_\mu, \mathscr{Y})} \right] \right)^{1/2} . \qquad (\text{Semi-norm})$$

See Bogachev (1998) and references therein for a detailed discussion on the definition and properties of $H^1_\mu(\mathscr{M}; \mathscr{Y})$. The following logarithmic Sobolev (Theorem 3) and Poincaré (Theorem 4) inequalities hold on $H^1_\mu(\mathscr{M}; \mathscr{Y})$, which are essential for establishing approximation error bounds on operator surrogate and Bayesian inversion. In particular, we have a Poincaré constant of 1 on $H^1_\mu(\mathscr{M}; \mathscr{Y})$.

Theorem 3 (Logarithmic Sobolev inequality, Bogachev 1998, 5.5.1) *If* $\mathcal{S} \in H^1_\mu(\mathscr{M}) \coloneqq H^1_\mu(\mathscr{M}; \mathbb{R})$*, then the following inequality holds*

$$\begin{aligned} \mathbb{E}_{M \sim \mu} \left[\mathcal{S}(M)^2 \ln \left(|\mathcal{S}(M)| \right) \right] \leq{}& \mathbb{E}_{M \sim \mu} \left[\left\| D_{\mathscr{H}_\mu} \mathcal{S}(M) \right\|^2_{\mathscr{H}_\mu} \right] \\ &+ \frac{1}{2} \mathbb{E}_{M \sim \mu} \left[\mathcal{S}(M)^2 \right] \ln \left(\mathbb{E}_{M \sim \mu} \left[\ln \left(\mathcal{S}(M)^2 \right) \right] \right) . \end{aligned}$$

where $D_{\mathscr{H}_\mu} \mathcal{S}$ *is the* $\mathscr{H}_\mu$*-Riesz representation of the stochastic derivative of* $\mathcal{S}$*.*

Theorem 4 (Poincaré inequality, Bogachev 1998, 5.5.6) *If* $\boldsymbol{\mathcal{T}} \in H^1_\mu(\mathscr{M}; \mathscr{Y})$*, then*

$$\|\boldsymbol{\mathcal{T}} - \mathbb{E}_{M \sim \mu}[\boldsymbol{\mathcal{T}}(M)]\|^2_{L^2_\mu(\mathscr{M}; \mathscr{Y})} \leq \left\| D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{T}} \right\|^2_{L^2_\mu(\mathscr{M}; \mathrm{HS}(\mathscr{H}_\mu, \mathscr{Y}))} .$$

The operator learning problem with error control in $H^1_\mu(\mathscr{M};\mathscr{Y})$ is formulated as

$$\boldsymbol{w}^\ddagger = \arg\min_{\boldsymbol{w}\in\mathbb{R}^{d_w}} \mathcal{L}^\infty_{H^1_\mu}(\boldsymbol{w})\,, \qquad \text{(Operator learning objective)} \tag{17a}$$

$$\mathcal{L}^\infty_{H^1_\mu} \coloneqq \frac{1}{2}\left\|\boldsymbol{\mathcal{G}} - \widetilde{\boldsymbol{\mathcal{G}}}(\cdot;\boldsymbol{w})\right\|^2_{H^1_\mu(\mathscr{M};\mathscr{Y})}\,. \qquad \text{(Error control in } H^1_\mu(\mathscr{M};\mathscr{Y})) \tag{17b}$$

The operator learning objective $\mathcal{L}^\infty_{H^1_\mu}$ can be estimated via joint samples of the operator evaluations and stochastic derivative evaluations $\{\, m_j \overset{\text{i.i.d.}}{\sim} \mu,\, \boldsymbol{\mathcal{G}}(m_j),\, D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m_j)\}_{j=1}^{n_t}$, which leads to a loss function $\mathcal{L}^{n_t}_{H^1_\mu}$ defined as follows:

$$\begin{aligned}\mathcal{L}^\infty_{H^1_\mu}(\boldsymbol{w}) \approx \mathcal{L}^{n_t}_{H^1_\mu}(\boldsymbol{w};\{m_j\}_{j=1}^{n_t}) \coloneqq \frac{1}{2n_t}\sum_{j=1}^{n_t}\Bigg(&\left\|\boldsymbol{\mathcal{G}}(m_j) - \widetilde{\boldsymbol{\mathcal{G}}}(m_j;\boldsymbol{w})\right\|^2_{\boldsymbol{C}_n^{-1}} \\ &+ \left\|D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m_j) - D_{\mathscr{H}_\mu}\widetilde{\boldsymbol{\mathcal{G}}}(m_j;\boldsymbol{w})\right\|^2_{\mathrm{HS}(\mathscr{H}_\mu,\mathscr{Y})}\Bigg)\,.\end{aligned} \tag{18}$$

In the context of neural network-based operator learning, we refer to the resulting operator surrogates as *derivative-informed neural operators* (DINOs).

3.3 Matrix representations of Hilbert–Schmidt operators

We consider a matrix representation of the stochastic derivative to generate training samples. For an arbitrary pair of orthonormal basis (ONB) on the parameter and observable CM spaces

$$\mathscr{H}_\mu\text{-ONB} \;:\; \{\psi_k\}_{k=1}^\infty\,, \quad \mathscr{Y}\text{-ONB} \;:\; \{\boldsymbol{v}_j\}_{j=1}^{d_y}\,,$$

we define a *Jacobian*, denoted by $\boldsymbol{J} : \mathscr{M} \to\in \mathrm{HS}(l^2,\mathbb{R}^{d_y})$ where l^2 denotes the Hilbert space of squared-summable sequences, using an isometric isomorphism between $\mathrm{HS}(l^2,\mathbb{R}^{d_y})$ and $\mathrm{HS}(\mathscr{H}_\mu,\mathscr{Y})$ defined by the bases:

$$(\boldsymbol{J}(m))_{jk} \coloneqq \boldsymbol{v}_j^T\boldsymbol{C}_n^{-1}D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\psi_k\,, \qquad \text{(Bijective linear mapping)} \tag{19a}$$

$$\left(\boldsymbol{J}(m)^T\right)_{jk} \coloneqq \left\langle \psi_j, D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^*\boldsymbol{v}_k\right\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}}\,, \qquad \text{(Jacobian matrix transpose)} \tag{19b}$$

$$\|\boldsymbol{J}(m)\|_F = \left\|D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\right\|_{\mathrm{HS}(\mathscr{H}_\mu;\mathscr{Y})}\,, \qquad \text{(Isometry)} \tag{19c}$$

where $\|\cdot\|_F$ is the Frobenius norm. The Frobenius and HS inner products are the same on $\mathrm{HS}(l^2,\mathbb{R}^{d_y})$, and $\boldsymbol{J}(m)$ can be interpreted as a matrix via its components defined in (19a) and (19b). Importantly, the isometry in (19c) is independent of the choice of basis, while the mapping between $\boldsymbol{J}$ and $D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}$ in (19a) depends on the choice of basis.

With the matrix representation of the stochastic derivative, training samples can be generated as Jacobian matrices (i.e., Jacobian evaluations) at each parameter sample. Furthermore, we can estimate the derivative approximation error at each parameter sample via the Frobenius norm of the error in the Jacobian matrices. However, the size of Jacobian matrices can be problematic in numerical computation. Assume that numerical computation is performed in a discretized parameter space $\mathscr{M}^h \subset \mathscr{M}$ using the Galerkin method, where $\mathscr{M}^h$ is isomorphic to $\mathbb{R}^{d_m}$. Then, the discretized Jacobian $\boldsymbol{J}^h$ outputs $\mathbb{R}^{d_y\times d_m}$ matrices,

i.e., $\boldsymbol{J}^h : \mathscr{M}^h \to \mathbb{R}^{d_y \times d_m}$; thus, storing and learning the Jacobian matrices generated at a large number of parameter samples can be intractable for large-scale problems. As a result, dimension reduction of the parameter space is essential for derivative-informed H^1_μ operator learning.

O'Leary-Roseberry et al. (2024) argue that restricting the derivative-informed operator learning using a pre-determined rank-r reduced basis $\{\psi_j\}_{j=1}^r$ with $r \ll d_m$ leads to tractable and accurate learning of the derivative for a wide range of PDE models. We adopt this strategy in this work. The following section describes DINO with error control in H^1_μ that extends reduced basis derivative-informed operator learning to our setting.

4. Reduced basis derivative-informed neural operator

Assume we have a set of reduced $\mathscr{H}_\mu$-ONBs of rank r denoted by $\{\psi_j\}_{j=1}^r$. They define a pair of linear encoders $\Psi_r^* \in \mathrm{HS}(\mathscr{H}_\mu, \mathbb{R}^r)$ and decoders $\Psi_r \in \mathrm{HS}(\mathbb{R}^r, \mathscr{H}_\mu)$ on $\mathscr{H}_\mu$ with $\Psi_r^*\Psi_r = \boldsymbol{I}_r \in \mathbb{R}^{r\times r}$, where $\boldsymbol{I}_r$ is the identity matrix:

$$\Psi_r^* : \mathscr{H}_\mu \ni m \mapsto \sum_{j=1}^r \langle m, \psi_j\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}} \boldsymbol{e}_j \in \mathbb{R}^r\,, \qquad \text{(Parameter encoder)} \tag{20a}$$

$$\Psi_r : \mathbb{R}^r \ni \boldsymbol{m}_r \mapsto \sum_{j=1}^r (\boldsymbol{m}_r)_j \psi_j \in \mathscr{H}_\mu\,, \qquad \text{(Parameter decoder)} \tag{20b}$$

where Ψ_r^* is the adjoint of Ψ_r and $\boldsymbol{e}_j$ is the unit vector along the j-th coordinate. Using the matrix representation of the HS operator introduced in Section 3.3, the linear encoder and decoder may be represented as

$$\Psi_r = \begin{bmatrix} | & | & & | \\ \psi_1 & \psi_2 & \cdots & \psi_r \\ | & | & & | \end{bmatrix}, \quad \Psi_r^* = \begin{bmatrix} — & \langle\psi_1,\cdot\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}} & — \\ — & \langle\psi_2,\cdot\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}} & — \\ & \vdots & \\ — & \langle\psi_r,\cdot\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}} & — \end{bmatrix}.$$

We extend the range and domain of the encoder and decoder from $\mathscr{H}_\mu$ to $\mathscr{M}$ and define a projection $\mathcal{P}_r$ on $\mathscr{M}$ as follows:

$$\mathcal{P}_r \coloneqq \Psi_r\Psi_r^* : \mathscr{M} \to \mathrm{span}(\{\psi_j\}_{j=1}^r)\,. \tag{21}$$

We emphasize that the $\mathscr{M}$-adjoint of Ψ_r is $\Psi_r^*\mathcal{C}_{\mathrm{pr}}$, and the $\mathscr{M}$-adjoint of Ψ_r^* is $\mathcal{C}_{\mathrm{pr}}^{-1}\Psi_r$. Similarly, let $\boldsymbol{V} \in \mathrm{HS}(\mathbb{R}^{d_y}, \mathscr{Y})$ be a matrix with columns consist of $\mathscr{Y}$-ONB vectors $\{\boldsymbol{v}_j\}_{j=1}^{d_y}$:

$$\boldsymbol{V} = \begin{bmatrix} | & | & & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_{d_y} \\ | & | & & | \end{bmatrix}. \qquad \text{(Observable basis)}$$

We note that $\boldsymbol{V}^* = \boldsymbol{V}^T\boldsymbol{C}_n^{-1}$, where $\boldsymbol{V}^T$ is the matrix transpose of $\boldsymbol{V}$.

We parameterize the operator surrogate using a neural network $\boldsymbol{f}_{\mathrm{NN}} : \mathbb{R}^r \times \mathbb{R}^{d_w} \to \mathbb{R}^{d_y}$:

$$\widetilde{\boldsymbol{\mathcal{G}}}(m;\boldsymbol{w}) \coloneqq \boldsymbol{V}\boldsymbol{f}_{\mathrm{NN}}(\Psi_r^* m, \boldsymbol{w})\,. \qquad \text{(Reduced basis neural operator)} \tag{22}$$

The neural network represents the nonlinear mapping from the coefficients of reduced $\mathscr{H}_\mu$-ONBs $\{\psi_j\}_{j=1}^r$ to the coefficients of $\mathscr{Y}$-ONBs $\{\boldsymbol{v}_j\}_{j=1}^{d_y}$.

Remark 5 *In this work, we restrict our attention to parameter dimension reduction only. We acknowledge that reducing the observables is essential for many BIPs, such as those with high-resolution image data and time-evolving data. Recent work by Baptista et al. (2022) studies optimal joint parameter and data dimension reduction in the context of BIPs based on logarithmic Sobolev inequality, which can be readily applied to our setting due to Theorem 3. However, there are many more practical considerations when jointly reducing the input and output dimensions for efficient DINO training. For this reason, we reserve dimension reduction of the observables for future work.*

The stochastic derivative of the operator surrogate and its adjoint can be expressed using the surrogate reduced Jacobian $\widetilde{\boldsymbol{J}_r}(\cdot;\boldsymbol{w}) : \mathscr{M} \to \mathbb{R}^{d_y \times r}$ through the neural network Jacobian $\partial_{\boldsymbol{m}_r} \boldsymbol{f}_{\mathrm{NN}}(\cdot,\boldsymbol{w}) : \mathbb{R}^r \to \mathbb{R}^{d_y \times r}$:

$$\widetilde{\boldsymbol{J}_r}(m;\boldsymbol{w}) \coloneqq \partial_{\boldsymbol{m}_r} \boldsymbol{f}_{\mathrm{NN}}(\Psi_r^* m, \boldsymbol{w}), \quad \begin{cases} D_{\mathscr{H}_\mu} \widetilde{\boldsymbol{\mathcal{G}}}(m;\boldsymbol{w}) = \boldsymbol{V} \widetilde{\boldsymbol{J}_r}(m;\boldsymbol{w}) \Psi_r^*, \\ D_{\mathscr{H}_\mu} \widetilde{\boldsymbol{\mathcal{G}}}(m;\boldsymbol{w})^* = \Psi_r \widetilde{\boldsymbol{J}_r}(m;\boldsymbol{w})^T \boldsymbol{V}^*. \end{cases}$$

Using the reduced basis architecture in (22) with $\mathscr{H}_\mu$ and $\mathscr{Y}$-ONBs and the isometric isomorphism in (19a), the derivative-informed H^1_μ operator learning objective in (17) can be reduced as follows:

$$\mathcal{L}^\infty_{H^1_\mu}(\boldsymbol{w}) \propto \frac{1}{2} \mathbb{E}_{M\sim\mu} \Bigg[\Big\| \underbrace{\boldsymbol{V}^* \boldsymbol{\mathcal{G}}(M) - \boldsymbol{f}_{\mathrm{NN}}(\Psi_r^* M, \boldsymbol{w})}_{\in \mathbb{R}^{d_y}} \Big\|^2 + \Big\| \underbrace{\boldsymbol{V}^* D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{G}}(M) \Psi_r - \partial_{\boldsymbol{m}_r} \boldsymbol{f}_{\mathrm{NN}}(\Psi_r^* M, \boldsymbol{w})}_{\boldsymbol{J}_r(M) - \widetilde{\boldsymbol{J}_r}(M;\boldsymbol{w}) \in \mathbb{R}^{d_y \times r}} \Big\|_F^2 \Bigg],$$

where constant terms independent of $\boldsymbol{w}$ are eliminated and the reduced Jacobian $\boldsymbol{J}_r : \mathscr{M} \to \mathbb{R}^{d_y \times r}$ of the PtO map is given by

$$\boldsymbol{J}_r(m) \coloneqq \boldsymbol{V}^* D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{G}}(m) \Psi_r. \qquad \text{(Reduced Jacobian matrix)}$$

The reduced loss function can now be estimated via joints samples $\{m_j \overset{\text{i.i.d.}}{\sim} \mu,\ \boldsymbol{V}^* \boldsymbol{\mathcal{G}}(m_j),\ \boldsymbol{J}_r(m_j)\}_{j=1}^{n_t}$.

4.1 A brief summary

We emphasize the following important points about our operator learning formulation:

1. The derivative-informed operator learning $\boldsymbol{\mathcal{G}} \approx \widetilde{\boldsymbol{\mathcal{G}}}(\cdot;\boldsymbol{w})$ is formulated as an approximation problem in $H^1_\mu(\mathscr{M};\mathscr{Y})$, a Sobolev space of nonlinear mappings between two separable Hilbert spaces $\mathscr{M}$ and $\mathscr{Y}$. While the parameter space $\mathscr{M}$ has infinite dimensions and the observable CM space $\mathscr{Y}$ has finite dimensions in our setting, the learning formulation is general and can be applied to infinite-dimensional output spaces. However, the stochastic derivative must remain an HS operator when the output space becomes a separable Hilbert space.

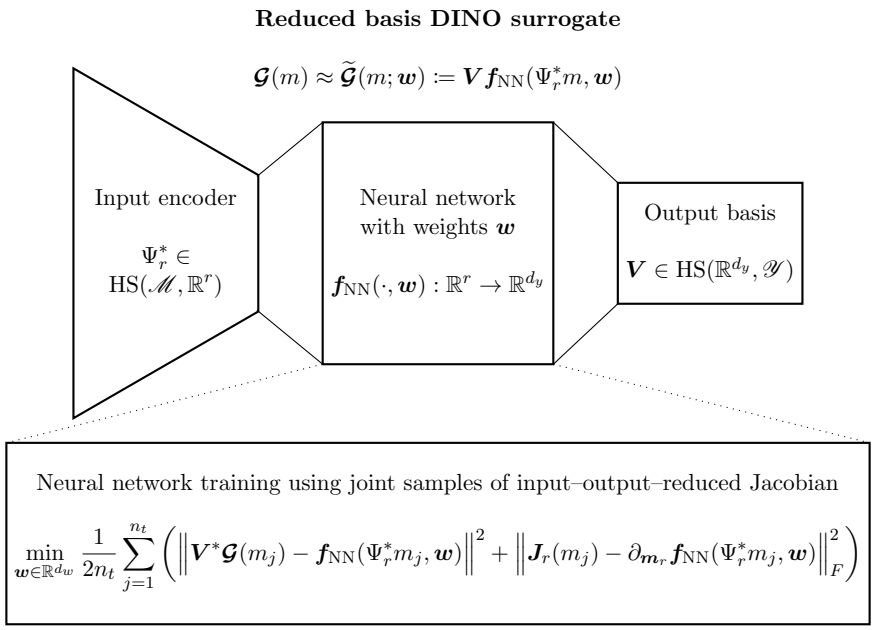

Figure 2: A schematic of reduced basis DINO architecture and learning for surrogate approximation $\widetilde{\boldsymbol{\mathcal{G}}} \approx \boldsymbol{\mathcal{G}}$ in $H^1_\mu(\mathscr{M};\mathscr{Y})$.

2. The neural network in the reduced basis DINO learns the mapping from the reduced coefficient vector of the parameter $\Psi_r^* m \in \mathbb{R}^r$ to the model-predicted coefficient vector of the observables $\boldsymbol{V}^*\boldsymbol{\mathcal{G}}(m) \in \mathbb{R}^{d_y}$.

$$\boldsymbol{f}_{\mathrm{NN}}(\cdot, \boldsymbol{w}) \approx \Psi_r^* m \mapsto \boldsymbol{V}^*\boldsymbol{\mathcal{G}}(m)\,. \qquad \text{(Neural network approximation)}$$

3. The neural network Jacobian of the reduced basis DINO learns the nonlinear mapping from the reduced coefficient vector of the parameter $\Psi_r^* m \in \mathbb{R}^r$ to the model-predicted reduced Jacobian matrix $\boldsymbol{J}_r(m) \in \mathbb{R}^{d_y \times r}$:

$$\partial_{\boldsymbol{m}_r}\boldsymbol{f}_{\mathrm{NN}}(\cdot, \boldsymbol{w}) \approx \Psi_r^* m \mapsto \boldsymbol{J}_r(m)\,. \qquad \text{(Neural network Jacobian approximation)}$$

4. As a result of the reduced basis architecture, the training sample storage and training cost are independent of the discretization dimension of the parameter space.

5. The derivative-informed H^1_μ operator learning often achieves better generalization at the small sample size regime compared to the conventional learning as it suffers less from overfitting. When the target mapping is undersampled, fitting only the input-output data points tends to interpolate without generalization. However, using derivative-informed learning, the neural network also learns the curvature of the target mapping around these data points. Consequently, the trained neural network generalizes well locally around each data point and achieves better generalization over the entire input space.

Remark 6 *In the following presentation, we often omit the notation for dependency on the neural network parameter $\boldsymbol{w}$ and use the tilde symbol $\widetilde{\cdot}$ when referring to the quantities computed via surrogate evaluations.*

4.2 Derivative and prior-based reduced bases

This subsection describes two types of reduced bases that can be used to construct DINO. The first type is based on the derivative-informed subspace (DIS, Constantine et al. 2014; Zahm et al. 2020; Cui and Zahm 2021; O'Leary-Roseberry et al. 2022b). The reduced bases for the derivative-informed subspace can be found by the following eigenvalue problem in $\mathscr{H}_\mu$ for the ppGNH (6):

$$\text{Find } \left\{(\lambda_j^{\text{DIS}}, \psi_j^{\text{DIS}}) \in \mathbb{R}_+ \times \mathscr{H}_\mu\right\}_{j=1}^{\infty} \text{ with decreasing } \lambda_j^{\text{DIS}} \text{ such that} \\ \begin{cases} \left(\mathbb{E}_{M\sim\mu}\left[\mathcal{H}(M)\right] - \lambda_j^{\text{DIS}} \mathcal{I}_{\mathscr{H}_\mu}\right) \psi_j^{\text{DIS}} = 0\,, & j \in \mathbb{N}\,; \\ \left\langle \psi_j^{\text{DIS}}, \psi_k^{\text{DIS}} \right\rangle_{\mathcal{C}_{\text{pr}}^{-1}} = \delta_{jk} & j,k \in \mathbb{N}\,. \end{cases} \tag{23}$$

We select the first r bases that correspond to the r largest eigenvalues to form encoders and decoders. During numerical computation, a Monte Carlo estimate of the expected ppGNH $\mathbb{E}_{M\sim\mu}[\mathcal{H}(M)]$ is computed at a set of prior samples $m_j \overset{\text{i.i.d.}}{\sim} \mu$, $j = 1, \ldots, n_{\text{DIS}}$:

$$\widehat{\mathcal{H}}(\{m_j\}_{j=1}^{n_{\text{DIS}}}) \coloneqq \frac{1}{n_{\text{DIS}}} \sum_{j=1}^{n_{\text{DIS}}} \mathcal{H}(m_j) \approx \mathbb{E}_{M\sim\mu}[\mathcal{H}(M)]\,. \tag{24}$$

The eigenvalue problem in (23) is solved to obtain the eigenpairs $\left\{(\widehat{\lambda_j^{\text{DIS}}}, \widehat{\psi_j^{\text{DIS}}})\right\}_{j=1}^{r}$ of $\widehat{\mathcal{H}}$, which gives the following DIS approximation of the expected ppGNH:

$$\mathbb{E}_{M\sim\mu}[\mathcal{H}(M)] \approx \widehat{\Psi_r^{\text{DIS}}} \widehat{\mathbf{\Lambda}_r^{\text{DIS}}} \widehat{\Psi_r^{\text{DIS}}}^*, \tag{25}$$

where the linear encoder and decoder are defined as in (20) and $\widehat{\mathbf{\Lambda}_r^{\text{DIS}}} \in \mathbb{R}^{r\times r}$ is a diagonal matrix consists of the eigenvalues.

The second type of reduced bases is based on the Karhunen–Loève expansion (KLE) of the prior distribution:

$$M = \sum_{j=1}^{\infty} \sqrt{\lambda_j^{\text{KLE}}} \Xi_j \eta_j \sim \mu\,, \quad \Xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1)\,, \quad \langle \eta_j, \eta_k \rangle_{\mathscr{M}} = \delta_{jk}\,,$$

where $\{(\lambda_j^{\text{KLE}}, \eta_j) \in \mathbb{R}_+ \times \mathscr{M}\}_{j=1}^{\infty}$ are eigenpairs of the prior covariance $\mathcal{C}_{\text{pr}}$ with $\mathscr{M}$-orthonormal eigenbases and decreasing eigenvalues. We refer to the r-dimensional subspace spanned by $\{\eta_j\}_{j=1}^r$ as the rank-r KLE subspace or simply the KLE subspace. A set of reduced $\mathscr{H}_\mu$-ONBs of the KLE subspace $\{\psi_j^{\text{KLE}}\}_{j=1}^r$ can be found by

$$\psi_j^{\text{KLE}} = \sqrt{\lambda_j^{\text{KLE}}} \eta_j\,, \quad 0 \le j \le r\,.$$

The KLE reduced bases $\{\psi_j^{\text{KLE}}\}_{j=1}^r$ can be computed with high precision for some representations of the $\mathcal{C}_{\text{pr}}$, notably Laplacian inverse or bi-Laplacian inverse Matérn covariances for Gaussian random functions (Bui-Thanh et al., 2013; Villa et al., 2021). More generally, the KLE reduced bases can be approximated from samples.

4.3 Training sample generation and cost analysis for PDE models

We describe a training sample generation procedure and its cost analysis when the PtO map $\boldsymbol{\mathcal{G}}$ is defined through a PDE. In particular, we consider an abstract variational residual form of the PDE as follows:

$$\text{Given } m \in \mathscr{M} \text{ find } u \in \mathscr{U} \text{ such that } \mathcal{R}(u, m) = 0 \in \mathscr{V}\,, \qquad \text{(PDE model)} \tag{26}$$

where $\mathscr{U}$ and $\mathscr{V}$ are Hilbert spaces corresponding to the spaces of PDE state and residual, and $\mathcal{R} : \mathscr{U} \times \mathscr{M} \to \mathscr{V}$ is the PDE residual operator. The residual space $\mathscr{V}$ is the dual space of the *space of adjoint variable* in the context of PDE-constrained optimization (Antil and Leykekhman, 2018; Manzoni et al., 2021), and the two spaces are identical when $\mathscr{V}$ is a Hilbert space. While the numerical examples in this work focus on steady-state problems where $\mathscr{U}$ is a Sobolev space defined over a spatial domain, e.g., $H^1(\Omega)$ where Ω is a spatial domain, our methodology is general. It can be applied to, e.g., time-evolving problems where $\mathscr{U}$ is a time-evolving space, e.g., $L^2([0,T]; H^1(\Omega))$ where T is a terminal time.

Assume the PDE solution operator $\mathcal{F} : \mathscr{M} \to \mathscr{U}$ is composed with a linear observation operator $\boldsymbol{\mathcal{O}} \in \mathrm{HS}(\mathscr{U}, \mathscr{Y})$ to define the PtO map $\boldsymbol{\mathcal{G}}$:

$$\boldsymbol{\mathcal{G}} \coloneqq \boldsymbol{\mathcal{O}} \circ \mathcal{F}\,, \quad \mathcal{R}(\mathcal{F}(m), m) = 0 \quad \mu\text{-a.e.} \qquad \text{(PDE-constrained PtO map)}$$

Each evaluation of the PtO map requires solving a PDE in (26). The action and the adjoint action of the stochastic derivative of the PtO map are given by the stochastic derivative of the PDE solution operator $D_{\mathscr{H}_\mu}\mathcal{F}(m) \in B(\mathscr{H}_\mu, \mathscr{U})$:

$$D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\delta m = \left(\boldsymbol{\mathcal{O}} \circ D_{\mathscr{H}_\mu}\mathcal{F}(m)\right)\delta m\,, \quad D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^*\delta \boldsymbol{y} = \left(D_{\mathscr{H}_\mu}\mathcal{F}(m)^* \circ \boldsymbol{\mathcal{O}}^*\right)\delta \boldsymbol{y}\,,$$

where the action of the derivative is given by the partial Gâteaux derivatives of the residual with respect to the PDE state and the parameter (in the direction of $\mathscr{H}_\mu$), denoted by $\partial_{\mathscr{U}}\mathcal{R}(\mathcal{F}(m), m) \in B(\mathscr{U}, \mathscr{V})$ and $\partial_{\mathscr{H}_\mu}\mathcal{R}(\mathcal{F}(m), m) \in B(\mathscr{H}_\mu, \mathscr{V})$ respectively. In particular, the implicit function theorem (Ciarlet, 2013) implies the following relations:

$$D_{\mathscr{H}_\mu}\mathcal{F}(m)\delta m = -\underbrace{\left(\partial_{\mathscr{U}}\mathcal{R}(\mathcal{F}(m), m)\right)^{-1}}_{\delta v \mapsto \delta u}\underbrace{\partial_{\mathscr{H}_\mu}\mathcal{R}(\mathcal{F}(m), m)}_{\delta m \mapsto \delta v}\delta m\,, \qquad \text{(Direct sensitivity)}$$

$$D_{\mathscr{H}_\mu}\mathcal{F}(m)^*\delta u = -\underbrace{\partial_{\mathscr{H}_\mu}\mathcal{R}(\mathcal{F}(m), m)^*}_{\delta v \mapsto \delta m}\underbrace{\left(\partial_{\mathscr{U}}\mathcal{R}(\mathcal{F}(m), m)^*\right)^{-1}}_{\delta u \mapsto \delta v}\delta u\,, \qquad \text{(Adjoint sensitivity)}$$

where $\delta v \in \mathscr{V}$ indicates a variation in the PDE residual or, equivalently, an adjoint variable. Evaluating the action of $D_{\mathscr{H}_\mu}\mathcal{F}(m)$ requires solving the linearized PDE problem for $\delta v \mapsto \delta u$, and evaluating its adjoint action $D_{\mathscr{H}_\mu}\mathcal{F}(m)^*$ requires solving the linear adjoint problem for $\delta u \mapsto \delta v$; see, e.g., Ghattas and Willcox 2021, Section 5 and Plessix 2006.

The associated computational cost for generating n_t training samples at parameter samples $m_j \overset{\text{i.i.d.}}{\sim} \mu$, $j = 1 \dots n_t$, can be decomposed as follows:

	$1\times$	**Cost of reduced bases estimation**
$+$	$n_t\times$	**Cost of a PDE solve**
$+$	$n_t\times$	**Cost of evaluating the reduced Jacobian $\boldsymbol{J}_r$**
$=$		**Cost of sample generation for DINO training**

When compared to L^2_μ training of an operator surrogate, DINO training requires additionally forming reduced Jacobian matrices $\boldsymbol{J}_r(m_j) \in \mathbb{R}^{d_y \times r}$ at each parameter sample m_j via rows or columns. In Table 1 and the following paragraph, we provide a simple cost analysis for this task when the parameter and the state space are discretized.

For time-evolving problems, we assume the state space $\mathscr{U}$ is discretized such that $\mathscr{U}^h$ is isomorphic to $\mathbb{R}^{d_t \times d_u}$, where d_t is the dimension of the temporal discretization and d_u is the dimension of the spatial discretization. We assume such discretization leads to d_t systems of equations (linear PDE) or d_t iterative systems of equations (nonlinear PDE) of size $d_u \times d_u$. For steady-state problems, we take $d_t = 1$. When a direct solver is used, the cost of factorizing systems of equations for a typical PDE problem is $O(d_t d_u^{3/2})$ and $O(d_t d_u^2)$ for 2D and 3D spatial domains, while back-substitution has a cost of $O(d_t d_u \ln d_u)$ and $O(d_t d_u^{4/3})$ for 2D and 3D spatial domains (Davis et al., 2016). The factorization used for solving a linear PDE can be reused to form $\boldsymbol{J}_r(m_j)$ via back-substitution, making the additional cost of H^1_μ training sample generation scale much slower with d_u compared to the cost of L^2_μ training. When an iterative solver is used, one can reuse preconditioners for a linear PDE to form $\boldsymbol{J}_r(m_j)$, but their cost analysis should be performed case-by-case. For nonlinear PDEs, one needs to solve one linear system of equations with $\min\{r, d_y\}$ different right-hand side vectors to form $\boldsymbol{J}_r(m_j)$, which is potentially much cheaper than solving a highly nonlinear PDE problem via iterative methods such as the Newton–Rapshon method.

Forming a reduced Jacobian matrix $\boldsymbol{J}_r(m_j) \in \mathbb{R}^{d_y \times r}$	
Linearized forward sensitivity:	$\text{column}_l(\boldsymbol{J}_r(m_j)) = \boldsymbol{V}^* D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{G}}(m_j)\psi_l\,, \quad l = 1 \dots r\,.$
Adjoint sensitivity:	$\text{row}_k(\boldsymbol{J}_r(m_j)) = \Psi_r^* D_{\mathscr{H}_\mu} \boldsymbol{\mathcal{G}}(m_j)^* \boldsymbol{v}_k\,, \quad k = 1 \dots d_y\,.$

Linear	**Solver**	**Operation** ($d_t = 1$ for steady-state problems)
✓	Direct	$d_t \times \min\{d_y, r\}\times$ Back-substitution (**note: significant cost saving from reusing factorization**)
	Iterative	$d_t \times \min\{d_y, r\}\times$ Iterative solve (**note: significant cost saving from reusing preconditioner**)
✗	Direct	$d_t\times$ Factorization + $d_t \times \min\{d_y, r\}\times$ Back-substitution
	Iterative	$d_t\times$ Preconditioner build + $d_t \times \min\{d_y, r\}\times$ Iterative solve

Table 1: The cost analysis of forming reduced Jacobian matrix $\boldsymbol{J}_r(m)$ at a parameter sample m_j given parameter reduced bases $\{\psi_j\}_{j=1}^\infty$ and observable basis $\{\boldsymbol{v}\}_{j=1}^{d_y}$.

4.4 Neural operator approximation error

This subsection briefly discusses approximation error for reduced basis DINO surrogates. To the best of our knowledge, there are no existing theoretical studies on H^1_μ approximation error with input dimension reduction. We focus on theoretical results that isolate various

sources of L^2_μ approximation error under the assumption that the true mapping lives in H^1_μ and comment on the relation between neural network size and the L^2_μ approximation error.

Understanding the L^2_μ approximation error of the operator surrogate is important as it is closely linked to the efficiency of the DA procedure. The connection between the second stage acceptance probability and the L^2_μ approximation error is discussed in Appendix C.

Here, we provide results on the $L^2_\mu(\mathscr{M};\mathscr{Y})$ approximation error of the DIS and KLE reduced basis neural operators. Our results show that a reduced basis architecture leads to approximation error contributions due to truncation and neural network approximation of the optimal reduced mapping. This mapping, denoted by $\boldsymbol{\mathcal{G}}_r$, can be defined explicitly (Zahm et al., 2020, Proposition 2.3) for a given pair of linear encoder Ψ_r^* and decoder Ψ_r constructed as in (20). Let $\mathcal{P}_r \coloneqq \Psi_r\Psi_r^*$ be a projection on $\mathscr{M}$ as in (21). We have

$$\boldsymbol{\mathcal{G}}_r(m) \coloneqq \mathbb{E}_{M\sim\mu}\left[\boldsymbol{\mathcal{G}}\left(\mathcal{P}_r m + (\mathcal{I}_{\mathscr{M}} - \mathcal{P}_r)M\right)\right], \qquad \text{(Subspace } L^2_\mu \text{ projection)} \tag{27a}$$

$$\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_r\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} = \inf_{\substack{\boldsymbol{\mathcal{T}}:\mathscr{M}\to\mathscr{Y}\\ \text{Borel func.}}} \|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{T}}\circ\mathcal{P}_r\|_{L^2_\mu(\mathscr{M};\mathscr{Y})}\,. \qquad \text{(Optimal reduced mapping)} \tag{27b}$$

The following propositions extend the results on DIS and KLE subspace by Zahm et al. (2020, Proposition 2.6 and 3.1) to a function space setting using the Poincaré inequality in Theorem 4. The proofs are provided in Appendix D.

Proposition 7 (L^2_μ approximation error, DIS) *Assume $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M};\mathscr{Y})$. Let $\{(\lambda_j^{\mathrm{DIS}}, \psi_j^{\mathrm{DIS}})\}_{j=1}^\infty$ and $\{(\widehat{\lambda_j^{\mathrm{DIS}}}, \widehat{\psi_j^{\mathrm{DIS}}})\}_{j=1}^\infty$ be the $\mathscr{H}_\mu$-orthonormal eigenpairs of the expected ppGNH $\mathbb{E}_{M\sim\mu}[\mathcal{H}(M)]$ in (23) and its estimator $\widehat{\mathcal{H}}$ in (24) with decreasing eigenvalues. Consider a reduced basis neural operator constructed using a linear encoder $\widehat{\Psi_r^{\mathrm{DIS}}}^* \in \mathrm{HS}(\mathscr{M},\mathbb{R}^{d_y})$ based on $\{\widehat{\psi_j^{\mathrm{DIS}}}\}_{j=1}^r$ as in (20) and any $\mathscr{Y}$-orthonormal basis $\boldsymbol{V} \in \mathrm{HS}(\mathbb{R}^{d_y},\mathscr{Y})$:*

$$\widetilde{\boldsymbol{\mathcal{G}}}(\cdot;\boldsymbol{w}) \coloneqq \boldsymbol{V}\circ \boldsymbol{f}_{\mathrm{NN}}(\cdot,\boldsymbol{w}) \circ \widehat{\Psi_r^{\mathrm{DIS}}}^*\,.$$

The following upper bound holds for the $L^2_\mu(\mathscr{M};\mathscr{Y})$ approximation error of $\widetilde{\boldsymbol{\mathcal{G}}}$ to $\boldsymbol{\mathcal{G}}$:

$$\left\|\boldsymbol{\mathcal{G}} - \widetilde{\boldsymbol{\mathcal{G}}}(\cdot;\boldsymbol{w})\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} \le \overbrace{\left\|\boldsymbol{f}_{\mathrm{NN}}(\cdot,\boldsymbol{w}) - \boldsymbol{V}^*\circ\boldsymbol{\mathcal{G}}_r\circ\widehat{\Psi_r^{\mathrm{DIS}}}\right\|_{L^2_{\mathcal{N}(\mathbf{0},\boldsymbol{I}_r)}(\mathbb{R}^r;\mathbb{R}^{d_y})}}^{\text{Neural network error}} + \Bigg(\underbrace{\sum_{j=r+1}^\infty \lambda_j^{\mathrm{DIS}}}_{\text{Basis truncation error}} + \underbrace{2r\left\|\mathbb{E}_{M\sim\mu}[\mathcal{H}(M)] - \widehat{\mathcal{H}}\right\|_{B(\mathscr{H}_\mu)}}_{\text{Sampling error}}\Bigg)^{1/2},$$

where $\widehat{\Psi_r^{\mathrm{DIS}}} \in \mathrm{HS}(\mathbb{R}^r,\mathscr{M})$ is the linear decoder based on $\{\widehat{\psi_j^{\mathrm{DIS}}}\}_{j=1}^r$ as in (20), and $\boldsymbol{\mathcal{G}}_r$ is the optimal reduced mapping of $\boldsymbol{\mathcal{G}}$ in (27).

Proposition 8 (L^2_μ approximation error, KLE) *Assume that $\boldsymbol{\mathcal{G}} \in H^1_\mu(\mathscr{M};\mathscr{Y})$ is Lipshitz continuous with a Lipschitz constant $c_{\boldsymbol{\mathcal{G}}} \ge 0$, i.e.,*

$$\|\boldsymbol{\mathcal{G}}(m_1) - \boldsymbol{\mathcal{G}}(m_2)\|_{\boldsymbol{C}_n^{-1}} \le c_{\boldsymbol{\mathcal{G}}}\|m_1 - m_2\|_{\mathscr{M}} \quad \forall m_1, m_2 \in \mathscr{M}\,.$$

Let $\{(\lambda_j^{\mathrm{KLE}}, \eta_j)\}_{j=1}^{\infty}$ *be the* $\mathscr{M}$*-orthonormal eigenpairs of* $\mathcal{C}_{\mathrm{pr}}$ *with decreasing eigenvalues. Consider a reduced basis neural operator constructed as* (22) *using a linear encoder* $\Psi_r^{\mathrm{KLE}^*} \in \mathrm{HS}(\mathscr{M}, \mathbb{R}^{d_y})$ *based on* $\left\{\psi_j^{\mathrm{KLE}} := \sqrt{\lambda_j^{\mathrm{KLE}}}\eta_j\right\}_{j=1}^{r}$ *as in* (20) *and any* $\mathscr{Y}$*-orthonormal basis* $\boldsymbol{V} \in \mathrm{HS}(\mathbb{R}^{d_y}, \mathscr{Y})$*:*

$$\widetilde{\boldsymbol{\mathcal{G}}}(\cdot; \boldsymbol{w}) := \boldsymbol{V} \circ \boldsymbol{f}_{\mathrm{NN}}(\cdot, \boldsymbol{w}) \circ \Psi_r^{\mathrm{KLE}^*}.$$

The following upper bound holds for $L^2_\mu(\mathscr{M}; \mathscr{Y})$ *approximation error of* $\widetilde{\boldsymbol{\mathcal{G}}}$ *to* $\boldsymbol{\mathcal{G}}$*:*

$$\left\|\boldsymbol{\mathcal{G}} - \widetilde{\boldsymbol{\mathcal{G}}}(\cdot; \boldsymbol{w})\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} \leq \underbrace{\left\|\boldsymbol{f}_{\mathrm{NN}} - \boldsymbol{V}^* \circ \boldsymbol{\mathcal{G}}_r \circ \Psi_r^{\mathrm{KLE}}\right\|_{L^2_{\mathcal{N}(\mathbf{0}, \boldsymbol{I}_r)}(\mathbb{R}^r; \mathbb{R}^{d_y})}}_{\textit{Neural network error}} + \underbrace{c_{\boldsymbol{\mathcal{G}}}\Big(\sum_{j=r+1}^{\infty} \left(\lambda_j^{\mathrm{KLE}}\right)^2\Big)^{1/2}}_{\textit{Basis truncation error}},$$

where $\Psi_r^{\mathrm{KLE}} \in \mathrm{HS}(\mathbb{R}^r, \mathscr{M})$ *is the linear decoder based on* $\{\psi_j^{\mathrm{KLE}}\}_{j=1}^r$ *as in* (20) *and* $\boldsymbol{\mathcal{G}}_r$ *is the optimal reduced mapping of* $\boldsymbol{\mathcal{G}}$ *in* (27)*. Additionally, we have*

$$\sum_{j=r+1}^{\infty} \lambda_j^{\mathrm{DIS}} \leq c_{\boldsymbol{\mathcal{G}}}^2 \sum_{j=r+1}^{\infty} \left(\lambda_j^{\mathrm{KLE}}\right)^2. \tag{28}$$

where $\{\lambda_j^{\mathrm{DIS}}\}_{j=1}^{\infty}$ *consists of decreasing eigenvalues of the expected ppGNH in* (23)*.*

Here, we briefly discuss the advantages and limitations of KLE and DIS parameter dimension reduction based on Propositions 7 and 8. KLE is relatively more straightforward to compute yet often leads to a larger basis truncation error compared to DIS as shown in (28). To compute DIS reduced bases, one needs to generate full Jacobian matrices at parameter samples, which may incur high computational and memory costs. On the other hand, a low number of samples may lead to considerable sampling error. This cost–accuracy trade-off needs to be balanced in DIS computation. Furthermore, the basis truncation error of DIS can still be significant if the eigenvalues of the expected ppGNH decay slowly. In these cases, alternative dimension reduction methods should be considered.

Furthermore, universal approximation theories of neural networks can help us understand the expressiveness of the neural network architecture (e.g., width, breadth, and activation functions) used in reduced basis neural operator surrogates. An important question is the neural network size, measured by the size of the weight d_w, needed to achieve a given neural network error tolerance. An exponential convergence in $L^2_{\mathcal{N}(\mathbf{0}, \boldsymbol{I}_r)}(\mathbb{R}^r; \mathbb{R})$ for approximating certain analytic functions by deep neural networks with the ReLU activation function is established by Schwab and Zech (2023, Theorem 4.7). Their theoretical results can be directly applied to the neural network error in our setting by stacking $\mathbb{R}^{d_y}$ of these deep neural networks to form an output space of $\mathbb{R}^{d_y}$, given that the optimal reduced mapping $\boldsymbol{\mathcal{G}}_r$ is sufficiently regular. Using this construction, the convergence rate derived by Schwab and Zech is scaled linearly by d_y.

5. Geometric MCMC via reduced basis neural operator

This section derives dimension-independent geometric MCMC methods with proposals entirely generated by a trained reduced basis neural operator. This work focuses on the

mMALA method introduced in Section 2.7 and approximates all components in the mMALA proposal using the surrogate. We note that the derivation in this section is similar to the DR-∞-mMALA method by Lan (2019), except that our derivation (i) does not involve prior covariance factorization[3], (ii) does not distinguish between KLE and DIS reduced bases, and (iii) involves the reduced basis neural operator surrogate introduced in (22).

5.1 Surrogate approximation

Given a trained neural network $\boldsymbol{f}_{\mathrm{NN}}(\cdot;\boldsymbol{w}^{\ddagger})$ as in (22), we can approximate the data misfit in (3) with $\widetilde{\Phi^{\boldsymbol{y}}}(\cdot;\boldsymbol{w}^{\ddagger}) \approx \Phi^{\boldsymbol{y}}$,

$$\widetilde{\Phi^{\boldsymbol{y}}}(m) \equiv \widetilde{\Phi_r^{\boldsymbol{y}}}(\Psi_r^* m)\,, \qquad \text{(Data misfit)} \tag{29a}$$

$$\widetilde{\Phi_r^{\boldsymbol{y}}}(\boldsymbol{m}_r) \coloneqq \frac{1}{2}\left\|\boldsymbol{V}^*\boldsymbol{y} - \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{m}_r)\right\|^2\,, \qquad \text{(Reduced data misfit)} \tag{29b}$$

the ppg in (5) with $D_{\mathscr{H}_\mu}\widetilde{\Phi^{\boldsymbol{y}}}(\cdot;\boldsymbol{w}^{\ddagger}) \approx D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}$,

$$D_{\mathscr{H}_\mu}\widetilde{\Phi^{\boldsymbol{y}}}(m) \equiv \Psi_r \widetilde{\boldsymbol{g}_r}(\Psi_r^* m)\,. \qquad \text{(ppg)} \tag{30a}$$

$$\mathbb{R}^r \ni \widetilde{\boldsymbol{g}_r}(\boldsymbol{m}_r) \coloneqq \partial_{\boldsymbol{m}_r}\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{m}_r)^T\left(\boldsymbol{V}^*\boldsymbol{y} - \boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{m}_r)\right)\,, \qquad \text{(Reduced ppg)} \tag{30b}$$

and the ppGNH in (6) with $\widetilde{\mathcal{H}}(\cdot;\boldsymbol{w}^{\ddagger}) \approx \mathcal{H}$,

$$\widetilde{\mathcal{H}}(m) \equiv \Psi_r \widetilde{\boldsymbol{H}_r}(\Psi_r^* m)\Psi_r^* \qquad \text{(ppGNH)} \tag{31a}$$

$$\mathbb{R}^{r\times r} \ni \widetilde{\boldsymbol{H}_r}(\boldsymbol{m}_r) \coloneqq (\partial_{\boldsymbol{m}_r}\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{m}_r))^T\partial_{\boldsymbol{m}_r}\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{m}_r)\,. \qquad \text{(Reduced ppGNH)} \tag{31b}$$

5.2 Surrogate prediction of posterior local geometry

An eigendecomposition of the surrogate reduced ppGNH $\widetilde{\boldsymbol{H}_r}(\boldsymbol{m}_r)$ with $\boldsymbol{m}_r = \Psi_r^* m$ is computed at the beginning of each step in the MH algorithm. Such an eigendecomposition is necessary for fast evaluations of multiple terms in the transition rate ratio (12). We denote the eigendecomposition of the surrogate reduced ppGNH as

$$\widetilde{\boldsymbol{H}_r}(\boldsymbol{m}_r) = \widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r)\widetilde{\boldsymbol{D}_r}(\boldsymbol{m}_r)\widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r)^T\,, \quad \begin{cases} \widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r)^T\widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r) = \boldsymbol{I}_r\,; \\ \left(\widetilde{\boldsymbol{D}_r}(\boldsymbol{m}_r)\right)_{jk} = \widetilde{d}_j(\boldsymbol{m}_r)\delta_{jk}\,. \end{cases} \tag{32}$$

where $\widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r) \in \mathbb{R}^{r\times r}$ is a rotation matrix in $\mathbb{R}^r$ with columns consist of eigenvectors and $\widetilde{\boldsymbol{D}_r}(\boldsymbol{m}_r) \in \mathbb{R}^{r\times r}$ is a diagonal matrix consists of eigenvalues $\{\widetilde{d}_j(\boldsymbol{m}_r)\}_{j=1}^r$. The rotation matrix nonlinearly depends on the parameter m through $\boldsymbol{m}_r$, thus leading to a pair of position-dependent linear decoder and encoder as follows:

$$\widetilde{\Psi_r}(\boldsymbol{m}_r) \coloneqq \Psi_r\widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r) \qquad \text{(Position-dependent linear decoder)} \tag{33a}$$

$$\widetilde{\Psi_r}(\boldsymbol{m}_r)^* \coloneqq \widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_r)^T\Psi_r^* \qquad \text{(Position-dependent linear encoder)} \tag{33b}$$

where the adjoint is taken in $\mathscr{H}_\mu$ similar to (20). The basis functions extracted from the position-dependent linear encoder and decoder represent the dominant directions of the surrogate posterior local geometry.

3. We acknowledge that Lan (2019) utilizes $\mathscr{H}_\mu$-orthonormal reduced bases, which means that the DR-∞-mMALA algorithm can be implemented without using prior factorization.

Remark 9 *In the following presentation, we often omit the notation of position dependency for the encoder, decoder, rotation matrix, and eigenvalues of the ppGNH when there is no ambiguity. Moreover, we adopt the index notation of diagonal matrices as in* (32) *to explicitly reveal its structure.*

The covariance of the local Gaussian approximation of the posterior in (4) can be approximated as follows:

$$\widetilde{\mathcal{C}_{\text{post}}}(m) = \mathcal{C}_{\text{pr}} - \widetilde{\Psi_r}\left(\frac{\widetilde{d}_j}{\widetilde{d}_j+1}\delta_{jk}\right)\widetilde{\Psi_r}^*\mathcal{C}_{\text{pr}}\,, \quad \widetilde{\mathcal{C}_{\text{post}}}(m)^{-1} = \mathcal{C}_{\text{pr}}^{-1} + \mathcal{C}_{\text{pr}}^{-1}\widetilde{\Psi_r}(\widetilde{d}_j\delta_{jk})\widetilde{\Psi_r}^*\,. \quad (34)$$

5.3 Sampling from the surrogate mMALA proposal

We consider an approximation to the mMALA proposal using the reduced basis neural operator with $\mathcal{K}(m) = \widetilde{\mathcal{C}_{\text{post}}}(m;\boldsymbol{w}^\dagger)$. By (34) and (11), we arrive at the following surrogate mMALA proposal:

$$\widetilde{\mathcal{Q}_{\text{mMALA}}}(m,\cdot) = \mathcal{N}\left(sm + (1-s)\,\widetilde{\mathcal{A}}(m), \left(1-s^2\right)\widetilde{\mathcal{C}_{\text{post}}}(m)\right)\,, \quad s = \frac{4-\triangle t}{4+\triangle t}\,,$$
$$\widetilde{\mathcal{A}}(m) = \widetilde{\Psi_r}\left(\frac{\widetilde{d}_j}{\widetilde{d}_j+1}\delta_{jk}\right)\widetilde{\Psi_r}^* m - \widetilde{\Psi_r}\left(\frac{1}{\widetilde{d}_j+1}\delta_{jk}\right)\widetilde{\boldsymbol{P}_r}^T\widetilde{\boldsymbol{g}_r}\,.$$

To sample from the surrogate mMALA proposal, we consider the following lemma:

Lemma 10 *Let $M \sim \mathcal{N}(0,\mathcal{C}_{\text{pr}})$, $m \in \mathscr{M}$ and $\mathcal{T} \in B(\mathscr{M})$. We have $m+\mathcal{T}M \sim \mathcal{N}(m, \mathcal{T}\mathcal{C}_{\text{pr}}\mathcal{T}^*)$. Moreover, if Ψ_r and Ψ_r^* are a set of linear encoder and decoder defined using reduced $\mathscr{H}_\mu$-ONBs of rank r as in* (20)*, then*

$$\Psi_r^* M \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_r) \text{ and } (\mathcal{I}_{\mathscr{M}} - \Psi_r\Psi_r^*)M \perp \Psi_r\Psi_r^* M\,,$$

where $\perp$ denotes pairwise independency of random elements.

Based on Lemma 10, we derive the following proposition for sampling the surrogate mMALA proposal by splitting the proposal into two parts: a position-dependent one for the r-dimensional coefficients in the reduced bases and the pCN proposal in the complementary subspace of $\mathscr{M}$, $\text{Range}(\mathcal{I}_{\mathscr{M}} - \Psi_r\Psi_r^*)$.

Proposition 11 *Given $m \in \mathscr{M}$ and $\triangle t > 0$, define two conditional distributions with $s \coloneqq (4-\triangle t)/(4+\triangle t)$:*

1. $M_\perp^\dagger \sim \mathcal{Q}_{\text{pCN}}(m,\cdot)$ following the pCN proposal distribution in (8a) *given by*

$$M_\perp^\dagger \coloneqq sm + \sqrt{1-s^2}M\,, \quad M \sim \mu\,.$$

2. $\boldsymbol{M}_r^\dagger \sim \pi_r(\cdot|\boldsymbol{m}_r = \Psi_r^* m)$, a r-dimensional conditional random vector given by

$$\begin{aligned}\boldsymbol{M}_r^\dagger \coloneqq\ & \widetilde{\boldsymbol{P}_r}\left(\frac{\widetilde{d}_j+s}{\widetilde{d}_j+1}\delta_{jk}\right)\widetilde{\boldsymbol{P}_r}^T\boldsymbol{m}_r - \widetilde{\boldsymbol{P}_r}\left(\frac{1-s}{\widetilde{d}_j+1}\delta_{jk}\right)\widetilde{\boldsymbol{P}_r}^T\widetilde{\boldsymbol{g}_r} \\ & + \widetilde{\boldsymbol{P}_r}\left(\left(\frac{1-s^2}{\widetilde{d}_j+1}\right)^{1/2}\delta_{jk}\right)\boldsymbol{\Xi}\,.\end{aligned} \quad (35)$$

where $\boldsymbol{\Xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_r)$ *is independent of* M*. We have*

$$(\mathcal{I}_{\mathscr{M}} - \Psi_r \Psi_r^*) M_\perp^\dagger + \Psi_r \boldsymbol{M}_r^\dagger \sim \widetilde{\mathcal{Q}_{\mathrm{mMALA}}}(m, \cdot).$$

See proofs of Lemma 10 and Proposition 11 in Appendix E. While using an operator surrogate for the position-dependent proposal sampling is novel, the idea of proposal splitting is common in dimension-independent MCMC methods; see, e.g., Cui et al. (2015, 2016); Beskos et al. (2017); Lan (2019).

5.4 Evaluating acceptance probabilities

The RN derivative $\widetilde{\rho_0}(m_1, m_2; \boldsymbol{w}^\ddagger)$ between the surrogate mMALA proposal $\widetilde{\mathcal{Q}_{\mathrm{mMALA}}}(m_1, \mathrm{d}m_2)$ and the pCN proposal $\mathcal{Q}_{\mathrm{pCN}}(m_1, \mathrm{d}m_2)$ can be efficiently evaluated using the trained neural network. Due to Proposition 11, $\widetilde{\rho_0}$ is constant in $\mathrm{Range}(\mathcal{I}_{\mathscr{M}} - \Psi_r \Psi_r^*)$ and can be reduced to a function in $\mathbb{R}^r$ denoted as $\widetilde{\rho_{0,r}}(\cdot, \cdot; \boldsymbol{w}^\ddagger) : \mathbb{R}^r \times \mathbb{R}^r \to \mathbb{R}_+$:

$$\widetilde{\rho_0}(m_1, m_2; \boldsymbol{w}^\ddagger) \equiv \widetilde{\rho_{0,r}}(\Psi_r^* m_1, \Psi_r^* m_2; \boldsymbol{w}^\ddagger)\,. \qquad \text{(Reduced density w.r.t. pCN)}$$

The form of $\widetilde{\rho_{0,r}}$ is given by

$$\begin{aligned}\widetilde{\rho_{0,r}}(\boldsymbol{m}_1, \boldsymbol{m}_2) \coloneqq \exp\Big(&- \frac{\triangle t}{8} \Big\| \widetilde{\boldsymbol{H}_r} \boldsymbol{m}_1 - \widetilde{\boldsymbol{g}_r} \Big\|^2_{(\widetilde{\boldsymbol{H}_r} + \boldsymbol{I}_r)^{-1})} \\ &+ \frac{\sqrt{\triangle t}}{2} \widehat{\boldsymbol{m}}^T \Big(\widetilde{\boldsymbol{H}_r} \boldsymbol{m}_1 - \widetilde{\boldsymbol{g}_r} \Big) - \frac{1}{2} \|\widehat{\boldsymbol{m}}\|^2_{\widetilde{\boldsymbol{H}_r}} \Big) + \det\Big((\widetilde{\boldsymbol{H}_r} + \boldsymbol{I}_r)^{1/2} \Big)\,,\end{aligned}$$

where $\widehat{\boldsymbol{m}} \coloneqq (\boldsymbol{m}_2 - s\boldsymbol{m}_1)/\sqrt{1 - s^2}$. Here, the reduced ppg $\widetilde{\boldsymbol{g}_r}$ defined in (30b) and the reduced ppGNH $\widetilde{\boldsymbol{H}_r}$ defined in (31b) are evaluated at $\boldsymbol{m}_1 = \Psi_r^* m_1$ through the trained neural network.

For the DA MCMC introduced in Section 2.8, only the surrogate data misfit enters the first stage transition rate ratio, and, thus, it can also be reduced to $\mathbb{R}^r$:

$$\rho^{(1)}(m_1, m_2; \boldsymbol{w}^\ddagger) \equiv \rho_r^{(1)}\Big(\Psi_r^* m_1, \Psi_r^* m_2; \boldsymbol{w}^\ddagger \Big)\,, \tag{36a}$$

$$\rho_r^{(1)}(\boldsymbol{m}_1, \boldsymbol{m}_2) \coloneqq \exp\Big(\widetilde{\Phi_r^{\boldsymbol{y}}}(\boldsymbol{m}_1) - \widetilde{\Phi_r^{\boldsymbol{y}}}(\boldsymbol{m}_2) \Big) \frac{\widetilde{\rho_{0,r}}(\boldsymbol{m}_2, \boldsymbol{m}_1)}{\widetilde{\rho_{0,r}}(\boldsymbol{m}_1, \boldsymbol{m}_2)}\,. \tag{36b}$$

When combined with the proposal splitting in Proposition 11, the first stage in the DA procedure can be performed entirely in reduced coefficient space $\mathbb{R}^r$, and prior sampling can be avoided until entering the second stage due to Lemma 10. In the second stage, the true data misfit evaluated at the full proposal is required to maintain the posterior sampling consistency of MCMC:

$$\rho^{(2)}(m_1, m_2; \boldsymbol{w}^\ddagger) = \frac{\exp(-\widetilde{\Phi_r^{\boldsymbol{y}}}(\Psi_r^* m_j)) \exp(-\Phi^{\boldsymbol{y}}(m^\dagger))}{\exp(-\widetilde{\Phi_r^{\boldsymbol{y}}}(\Psi_r^* m^\dagger)) \exp(-\Phi^{\boldsymbol{y}}(m_j))}\,. \tag{37}$$

We note that (36b) allows for proposal rejection without true data misfit evaluation nor prior sampling. The additional cost reduction in prior sampling due to our choice of reduced basis architecture can be important, for example, when the prior is defined through Whittle–Matérn Gaussian random fields (Whittle, 1954) that require solving fractional stochastic PDEs to sample. In Algorithm 1, we summarize the procedure for our DA geometric MCMC method via a reduced basis neural operator surrogate.

Algorithm 1: Markov chain transition of surrogate-driven DA mMALA at the j-th position $m_j \in \mathscr{M}$

Input: (i) a trained neural network $\boldsymbol{f}_{\text{NN}}(\cdot, \boldsymbol{w}^\ddagger)$, (ii) a pair of parameter encoder Ψ_r^* and decoder Ψ_r, (iii) an observable basis $\boldsymbol{V}$, and (iv) a step size $\triangle t$.

Known at m_j**:** (i) the data misfit value $\Phi^{\boldsymbol{y}}(m_j)$, (ii) the surrogate reduced data misfit value $\widetilde{\Phi_r^{\boldsymbol{y}}}(\boldsymbol{m}_{r,j})$, where $\boldsymbol{m}_{r,j} \coloneqq \Psi_r^* m_j$ (iii) the surrogate reduced ppg $\widetilde{\boldsymbol{g}_r}(\boldsymbol{m}_{r,j})$, (iv) the surrogate reduced ppGNH eigendecomposition $\Big(\widetilde{\boldsymbol{D}_r}(\boldsymbol{m}_{r,j}), \widetilde{\boldsymbol{P}_r}(\boldsymbol{m}_{r,j})\Big)$.

Output: The next position $m_{j+1} \in \mathscr{M}$.

Sample $\boldsymbol{\xi}_r \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_r)$;
Compute $\boldsymbol{m}_r^\dagger$ using $\boldsymbol{\xi}_r$ via (35); ▷ `Reduced proposal sampling`
Evaluate $\boldsymbol{f}_{\text{NN}}$ and $\partial_{\boldsymbol{m}_r} \boldsymbol{f}_{\text{NN}}$ at $\boldsymbol{m}_r^\dagger$;
Evaluate $\widetilde{\Phi_r^{\boldsymbol{y}}}$, $\widetilde{\boldsymbol{g}_r}$, $\widetilde{\boldsymbol{D}_r}$, and $\widetilde{\boldsymbol{P}_r}$ at $\boldsymbol{m}_r^\dagger$; ▷ $\mathbb{R}^{r\times r}$ `Hermitian eigenvalue problem`
Compute $\alpha^{(1)} = \min\{1, \rho_r^{(1)}(\boldsymbol{m}_{r,j}, \boldsymbol{m}_r^\dagger)\}$ via (36b);
if $\alpha^{(1)} < \xi_1$ *where* $\xi_1 \overset{\text{i.i.d.}}{\sim} \text{Uniform}([0,1])$ **then**
  return m_j; ▷ `First stage rejection`
else
  Sample prior $m_\perp^\dagger \overset{\text{i.i.d.}}{\sim} \mu$; ▷ `Prior sampling in second stage`
  Compute $m^\dagger = \Psi_r \boldsymbol{m}_r^\dagger + m_\perp^\dagger - \Psi_r \Psi_r^* m_\perp^\dagger$; ▷ `Assemble the full proposal`
  Evaluate the data misfit $\Phi^{\boldsymbol{y}}(m^\dagger)$; ▷ `Model evaluation in second stage`
  Compute $\alpha^{(2)} = \min\{1, \rho^{(2)}(m_j, m^\dagger)\}$ via (37)
  if $\alpha^{(2)} < \xi_2$ *where* $\xi_2 \overset{\text{i.i.d.}}{\sim} \text{Uniform}([0,1])$ **then**
    return m_j; ▷ `Second stage rejection`
  else
    return $m^\dagger$; ▷ `Second stage acceptance`
  end
end

6. Numerical examples: Baseline, chain diagnostics, efficiency metrics, and software

The proposed DINO-driven geometric MCMC method is studied on two PDE-constrained BIPs in Sections 7 and 8. In this section, we briefly introduce baseline MCMC methods to assess the efficiency of our proposed MCMC method. Then, we specify two diagnostics for assessing the quality of Markov chains for posterior sampling. Next, we introduce two metrics that quantify the relative efficiency of two MCMC methods.

6.1 Baseline and reference MCMC methods

A table of baseline MCMC methods is listed in Table 2. The mMALA method can be deduced from the generic mMALA proposal in (11) with $\mathcal{K}(m) = (\mathcal{I}_{\mathscr{H}_\mu} + \mathcal{H}(m))^{-1}\mathcal{C}_{\text{pr}}$

Posterior geometry information	**Name**	**Gauss–Newton Hessian (6) (approximation)**	**Gradient (5)**
None	pCN	✗	✗
	MALA	✗	✓
Fixed	LA-pCN	$\mathcal{H}(m_{\text{MAP}})$ in (6) and (38a)	✗
	DIS-mMALA	$\widehat{\Psi_r^{\text{DIS}}}\widehat{\mathbf{\Lambda}_r^{\text{DIS}}}\widehat{\Psi_r^{\text{DIS}}}^*$ in (25)	✓
Position -dependent	mMALA	$\mathcal{H}$ in (6)	✓

Table 2: A list of baseline dimension-independent MCMC methods used in our numerical examples.

Operator learning objective function	**Name**	**Reduced bases**	**Delayed acceptance**	**Reference method (39)**
Conventional $L^2_\mu(\mathscr{M};\mathscr{Y})$	NO-mMALA	DIS $r = 200$	✗	r-mMALA
	DA-NO-mMALA		✓	DA-r-mMALA
Derivative-informed $H^1_\mu(\mathscr{M};\mathscr{Y})$	DINO-mMALA		✗	r-mMALA
	DA-DINO-mMALA		✓	DA-r-mMALA

Table 3: A list of operator surrogate-based dimension-independent geometric MCMC methods used in our numerical examples. We additionally introduce two reference methods, r-mMALA and DA-r-mMALA, that isolate the effects of reduced basis architecture in the empirical performance of surrogate-driven MCMC.

as in (6). This method is the same as ∞-mMALA by Beskos et al. (2017); Lan (2019). The DIS-mMALA method uses the DIS approximation of $\mathbb{E}_{M\sim\mu}[\mathcal{H}(m)]$ in (25) as a fixed approximation to $\mathcal{H}$. This method is the same as DR-∞-mMALA by Lan (2019) with a fixed DIS reduced basis, similar to gpCN by Rudolf and Sprungk (2018) except that gpCN does not include ppg, similar to LI-Langevin by Cui et al. (2016) except that (i) DIS-mMALA do not require prior covariance factorization and (ii) DIS-mMALA has only one step size parameter. The LA-pCN method (Pinski et al., 2015; Kim et al., 2023) utilizes the Laplace approximation to the posterior, which requires solving the following deterministic inverse problem for the maximum a posteriori probability (MAP) estimate (Dashti et al., 2013; Villa et al., 2021), denoted by $m_{\text{MAP}} \in \mathscr{H}_\mu$, to construct a proposal:

$$m_{\text{MAP}} \coloneqq \operatorname*{arg\,min}_{m\in\mathscr{M}} \left(\Phi^{\boldsymbol{y}}(m) + \frac{1}{2}\|m\|^2_{\mathcal{C}_{\text{pr}}^{-1}}\right), \quad \mu^{\boldsymbol{y}} \approx \mathcal{N}(m_{\text{MAP}}, \mathcal{C}_{\text{post}}(m_{\text{MAP}})), \tag{38a}$$

$$\mathcal{Q}_{\text{LA-pCN}}(m,\cdot) \coloneqq \mathcal{N}\left(m_{\text{MAP}} - sm, (1-s^2)\mathcal{C}_{\text{post}}(m_{\text{MAP}})\right). \tag{38b}$$

In Table 3, we provide a list of MCMC methods driven by reduced basis neural operators detailed in Section 5. The mMALA proposal approximated by both the L^2_μ-trained NO and

H^1_μ-trained DINO with and without the DA procedure is studied in our numerical examples, and their efficiency is compared with the baseline methods.

When computationally feasible, we complement the numerical results of our proposed methods with those of two reference MCMC methods: r-mMALA (reduced mMALA) and DA-r-MALA (reduced mMALA with delayed acceptance). These reference methods are designed to isolate the effects of the reduced basis architecture of DINO on the performance of the proposed MCMC methods. They are defined via the following sample average approximation to the optimal reduced mapping (27) of the PtO map and its reduced Jacobian:

$$\boldsymbol{\mathcal{G}}(m) \approx \sum_{j=1}^{n_{\text{rm}}} \boldsymbol{\mathcal{G}}\left(\Psi_r\Psi_r^* m + (\mathcal{I}_{\mathscr{M}} - \Psi_r\Psi_r^*)m_j\right) , \quad m_j \overset{\text{i.i.d.}}{\sim} \mu\,, \tag{39a}$$

$$\boldsymbol{J}_r(m) \approx \sum_{j=1}^{n_{\text{rm}}} \boldsymbol{V}^* D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}\left(\Psi_r\Psi_r^* m + (\mathcal{I}_{\mathscr{M}} - \Psi_r\Psi_r^*)m_j\right)\Psi_r^* , \quad m_j \overset{\text{i.i.d.}}{\sim} \mu\,. \tag{39b}$$

The r-mMALA reference method replaces the PtO map and its stochastic derivative in the mMALA proposal using (39). The DA-r-mMALA reference method additionally includes the DA procedure using (39a). We take $n_{\text{rm}} = 20$ in Section 7.

6.2 Markov chain diagnostics

We focus on two diagnostics that help us understand the quality of Markov chains generated by MCMC: the multivariate potential scale reduction factor (MPSRF) and the effective sample size percentage (ESS%). In this subsection, we assume access to n_c number of independent Markov chains generated by the same MCMC method targeting the same posterior. Each chain has n_s samples (after burn-in), denoted by $\{\{m_{j,k}\}_{j=1}^{n_s}\}_{k=1}^{n_c}$.

6.2.1 Wasserstein multivariate potential scale reduction factor

The MPSRF (Brooks and Gelman, 1998) is a diagnostic for the convergence of MCMC. It compares the multi-chain mean of the empirical covariance within each chain, denoted as $\widehat{\mathcal{W}}_s \in B(\mathscr{M})$, and the empirical covariance across all chains, denoted as $\widehat{\mathcal{V}}_s \in B(\mathscr{M})$. They are given by

$$\widehat{\mathcal{W}}_s \coloneqq \frac{1}{n_c(n_s-1)} \sum_{k=1}^{n_c}\sum_{j=1}^{n_s} \left\langle m_{j,k} - \overline{m}_k, \cdot \right\rangle_{\mathscr{M}} (m_{j,k} - \overline{m}_k) \,, \tag{40a}$$

$$\widehat{\mathcal{V}}_s \coloneqq \frac{n_s - 1}{n_s}\widehat{\mathcal{W}}_s + \frac{n_c+1}{n_c(n_c-1)} \sum_{k=1}^{n_c} \left\langle \overline{m_k} - \overline{m}, \cdot \right\rangle_{\mathscr{M}} (\overline{m_k} - \overline{m}) \,, \tag{40b}$$

where $\overline{m_k}$ is the mean of samples in each chain labeled by $k = 1, \ldots, n_{\text{chain}}$ and $\overline{m}$ is the mean of samples in all chains. We expect the difference in $\widehat{\mathcal{W}}_s$ and $\widehat{\mathcal{V}}_s \in B(\mathscr{M})$ to be small as the chains become longer.

In this work, we propose to use the 2-Wasserstein distance for Gaussian measures (Dowson and Landau, 1982) to compare the distance between the two covariance operators. We

refer to this diagnostic as Wasserstein MPSRF:

$$\begin{aligned}\widehat{R}_w &= \mathrm{Wass}_2\left(\mathcal{N}(0,\widehat{\mathcal{W}}_s),\mathcal{N}(0,\widehat{\mathcal{V}}_s)\right)\\ &= \mathrm{Tr}_{\mathscr{M}}\left(\widehat{\mathcal{W}}_s+\widehat{\mathcal{V}}_s-2(\widehat{\mathcal{W}}_s^{1/2}\widehat{\mathcal{V}}_s\widehat{\mathcal{W}}_s^{1/2})^{1/2}\right).\end{aligned} \qquad \text{(Wasserstein MPSRF)}$$

The faster $\widehat{R}_w$ decays as a function of the chain length, the faster the pool of chains converges to their stationary distribution (i.e., faster mixing time). While the Wasserstein MPSRF is uncommon for MCMC convergence diagnostics, we find it useful for comparing the performance of MCMC methods in function spaces; see Appendix F for additional discussions on this diagnostic.

6.2.2 Effective sample size percentage distribution

The ESS% (Gelman et al., 2014) is the estimated percentage of independent samples from a pool of Markov chains. When each sample resides in 1D, such a metric is estimated using the autocorrelation function, denoted by $\mathrm{AC}(t,k)$, of samples that are t positions apart within the k-th Markov chain:

$$\mathrm{ESS}\% = \frac{1}{1+2\sum_{t=1}^{2n'+1}\mathrm{MAC}(t)}, \qquad \text{(Effective sample size percentage)} \quad (41a)$$

$$\mathrm{MACT}(t) = 1-\frac{\widehat{w}_s-\frac{1}{n_c}\sum_{k=1}^{n_c}\mathrm{AC}(t,k)}{\widehat{v}_s}, \qquad \text{(Multichain autocorrelation time)} \quad (41b)$$

where MACT is the multi-chain AC estimate, $\widehat{w}_s$ and $\widehat{v}_s$ is the 1D version of (40). The index $n'\in\mathbb{Z}_+$ is chosen to be the largest integer so that the sum of MACT evaluated at neighboring positions is positive.

The simulated MCMC samples $m_{j,k}^h$ belong to a discretized function space $\mathscr{M}^h$ of dimension d_m. Following Beskos et al. (2017) and Lan (2019), we estimate this 1D diagnostic metric for each degree of freedom (DoF) in the discretized space. This leads to a distribution of d_m ESS% estimates, visualized using the violin plot.

6.3 Comparing efficiency of MCMC methods

In this subsection, we introduce two metrics that measure the relative efficiency of a pair of MCMC methods: effective sampling speedup and total effective sampling speedup.

6.3.1 Effective sampling speedup

The effective sampling speed quantifies the efficiency of an MCMC method regarding its speed of effective sample generation:

$$\text{Effective sampling speed} = \frac{\text{Median(ESS\%)}}{\text{Cost of 100 Markov chain samples}}. \quad (42)$$

Instead of directly computing this quantity, we use it to compare the relative efficiency of posterior sampling for different pairs of MCMC methods. The relative efficiency is measured by *the effective sampling speedup*, or speedup for short. It is given by the ratio between the effective sampling speed of the two methods.

6.3.2 Total effective sampling speedup

Training sample generation is an important part of the total computational cost for DINO-accelerated geometric MCMC methods, especially when the PtO map is expensive to evaluate. The cost of training sample generation is a fixed offline cost, while the cost of posterior sampling scales with the number of MCMC samples. To incorporate both the offline training cost and the online MCMC cost, we introduce an efficiency metric called *the total effective sampling speed* as a function of the effective sample size n_{ess} required for an MCMC run:

$$\text{Total effective sampling speed}(n_{\text{ess}}) = \frac{n_{\text{ess}}}{\text{Total cost}(n_{\text{ess}})}, \tag{43}$$

$$\text{Total cost}(n_{\text{ess}}) = \text{Offline cost} \ + \ \text{Cost of 100 Markov chain samples} \ \times \frac{n_{\text{ess}}}{\text{Median(ESS\%)}}.$$

For DINO-mMALA and DA-DINO-mMALA, the offline cost is the sum of training sample generation and neural network training cost. The total effective sampling speed converges to the effective sampling speed when the offline cost is negligible compared to the cost of the MCMC run, i.e., $n_{\text{ess}} \to \infty$. Instead of directly computing this quantity, we use it to compare the relative efficiency of posterior sampling for different pairs of MCMC methods. The relative efficiency is measured by *the total effective sampling speed*, or total speedup for short. It is given by the ratio between the total effective sampling speed of the two methods at a given n_{ess}.

Remark 12 *In our numerical examples, we do not include the computational cost of step size tuning and burn-in in the total cost. We found that these costs can be significantly reduced if the Markov chain's initial position is sampled from the Laplace approximation of the posterior instead of the prior. See Appendix F for the step size tuning and chain initialization procedure.*

6.4 Software

Our numerical examples are implemented through `geometric_mcmc`, a software library for solving PDE-constrained Bayesian inverse problems using MCMC. It is built on other open-source projects, mainly (i) `FEniCS` (Alnæs et al., 2015; Logg et al., 2012) for finite element discretization, solves, and symbolic differentiation of PDE residual operators, and (ii) `hIPPYlib` (Villa et al., 2018, 2021) for all components related to inverse problems, e.g. the prior, adjoint solves, Laplace approximation, and some baseline MCMC methods (LA-pCN, MALA, and pCN). We also used `hIPPYflow` (O'Leary-Roseberry and Villa, 2021) for reduced basis estimation and training sample generation, and (iv) `dino` (O'Leary-Roseberry, 2023) for derivative-informed operator learning.

7. Numerical example: Coefficient inversion for a nonlinear diffusion–reaction PDE

We consider the following steady-state nonlinear diffusion–reaction equation in the unit square:

$$\begin{aligned} -\nabla \cdot \exp(m(\boldsymbol{x}))\nabla u(\boldsymbol{x}) + u(\boldsymbol{x})^3 &= 0\,, && \boldsymbol{x} \in (0,1)^2\,; \\ u(\boldsymbol{x}) &= 0\,, && \boldsymbol{x} \in \Gamma_{\text{bottom}}\,; \\ u(\boldsymbol{x}) &= 1\,, && \boldsymbol{x} \in \Gamma_{\text{top}}\,; \\ \exp(m(\boldsymbol{x}))\nabla u(\boldsymbol{x}) \cdot \boldsymbol{n} &= 0\,, && \boldsymbol{x} \in \Gamma_{\text{left}} \cup \Gamma_{\text{right}}\,; \end{aligned}$$

where Γ_{bottom}, Γ_{top}, Γ_{left}, and Γ_{right} are the boundaries the unit square. The inverse problem is to invert for the log-coefficient field m given noise-corrupted discrete observations of the PDE state variable u at a set of spatial positions.

This section is organized as follows. We introduce the prior, the PtO map, and the setting for Bayesian inversion in Sections 7.1 to 7.3. Then, we present the specifications and results on training operator surrogates with conventional L^2_μ operator learning and the derivative-informed H^1_μ operator learning in Section 7.4. Next, we showcase and analyze MCMC results in Section 7.5. In Section 7.5.1, we discuss results on the baseline methods listed in Table 2. In Section 7.5.2, we discuss results on NO-mMALA and DINO-mMALA listed in Table 3 to understand the quality of surrogate mMALA proposals. In Section 7.5.3, we discuss results on DA-NO-mMALA and DA-DINO-mMALA listed in Table 3.

7.1 The prior distribution

We consider the following parameter space and prior distribution

$$\begin{aligned} \mathscr{M} &\coloneqq L^2((0,1)^2)\,, && \text{(Parameter space)} \\ \mu &\coloneqq \mathcal{N}(0, (-0.03\Delta + 3.33\mathcal{I}_{\mathscr{M}})^{-2})\,, && \text{(The prior distribution)} \end{aligned}$$

where $-\Delta : H^1((0,1)^2) \to H^1((0,1)^2)'$ is the weak Laplace operator with a Robin boundary condition for eliminating boundary effects (Villa et al., 2021, Equation 37). The resulting Gaussian random field has pointwise variance and spatial correlation lengths of around 9 and 0.1, respectively. We approximate $\mathscr{M}$ using a finite element space $\mathscr{M}^h$ constructed by linear triangular elements with 1681 DoFs. A visualization of prior samples is provided in Figure 3.

7.2 The parameter-to-observable map

We consider a symmetric variational formulation of the PDE problem, and define the following Hilbert spaces following the notation in Section 4.3:

$$\begin{aligned} \mathscr{U} &\coloneqq \left\{ u \in H^1((0,1)^2) \;\middle|\; u|_{\Gamma_t} = 0 \wedge u|_{\Gamma_b} = 0 \right\}\,; && \text{(State space)} \\ \mathscr{V} &\coloneqq \mathscr{U}'\,, && \text{(Residual space)} \end{aligned}$$

where $\mathscr{U}'$ denotes the dual space of $\mathscr{U}$. To enforce the inhomogeneous Dirichlet boundary condition, we decompose the PDE solution u into $u = u_0 + \boldsymbol{x}^T \boldsymbol{e}_2$, where $\boldsymbol{e}_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ and $u_0 \in \mathscr{U}$ is the PDE state with the homogenous Dirichlet boundary condition.

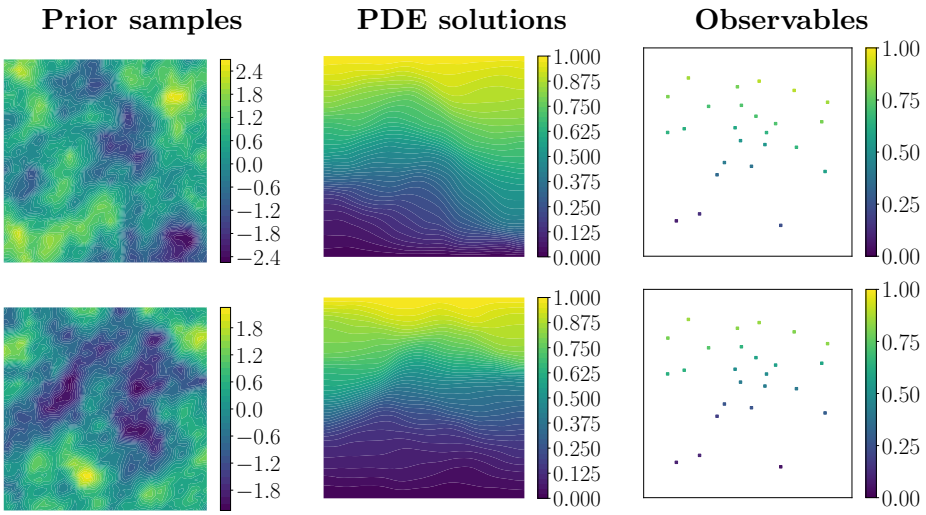

Figure 3: Visualizations of prior samples (1681 DoFs), PDE solutions (3362 DoFs), and predicted observables ($\mathbb{R}^{25}$) for coefficient inversion in a nonlinear diffusion–reaction PDE.

The residual operator for this PDE problem, denoted as $\mathcal{R} : \mathscr{U} \times \mathscr{M} \to \mathscr{V}$, can be defined by its action on an arbitrary test function $p \in \mathscr{U}$

$$\begin{aligned}\langle \mathcal{R}(u_0, m), p\rangle_{\mathscr{U}'\times\mathscr{U}} \coloneqq \int_{(0,1)^2} \Big(&\exp(m(\boldsymbol{x}))\,(\nabla u_0(\boldsymbol{x}) + \boldsymbol{e}_2)\cdot\nabla p(\boldsymbol{x}) \\ &+ \left(u_0(\boldsymbol{x}) + \boldsymbol{x}^T\boldsymbol{e}_2\right)^3 p(\boldsymbol{x})\Big)\,\mathrm{d}\boldsymbol{x}\,.\end{aligned} \tag{45}$$

The effective PDE solution operator is defined as a nonlinear parameter-to-state map $\mathcal{F} : \mathscr{M} \ni m \mapsto u_0 \in \mathscr{U}$ where $\mathcal{R}(u_0, m) = 0$. We approximate $\mathscr{U}$ using a finite element space $\mathscr{U}^h$ constructed by quadratic triangular elements with 3362 DoFs. Evaluating the discretized PDE solution operator involves solving the discretized residual norm minimization problem via the Newton–Raphson method in $\mathscr{U}^h$.

We define the observation operator $\boldsymbol{\mathcal{O}}$ using 25 randomly-sampled discrete interior points $\{\boldsymbol{x}_{\text{obs}}^{(j)}\}_{j=1}^{d_y}$ with $d_y = 25$:

$$\boldsymbol{\mathcal{O}}(u_0) = \left[\int_{\boldsymbol{B}_\epsilon(\boldsymbol{x}_{\text{obs}}^{(1)})} u(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \dots \quad \int_{\boldsymbol{B}_\epsilon(\boldsymbol{x}_{\text{obs}}^{(25)})} u(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\right]^T, \quad \text{(Observation operator)} \tag{46}$$

where $\boldsymbol{B}_\epsilon(\boldsymbol{x}) \subset (0,1)^2$ is a ball around $\boldsymbol{x}$ with a small radius $\epsilon > 0$. The PtO map is $\boldsymbol{\mathcal{G}} \coloneqq \boldsymbol{\mathcal{O}} \circ \mathcal{F}$. Samples of the PtO map are visualized in Figure 3.

7.3 Bayesian inverse problem settings

We generate synthetic data for our BIP using an out-of-distribution piecewise-constant parameter field, following the examples from, e.g., Cui et al. (2016); Lan (2019). The model-predicted observable at the synthetic parameter field is then corrupted with 2% additive white noise, which leads to a noise covariance matrix of identity scaled by $v_n = 1.7 \times 10^{-4}$:

$$\pi_n = \mathcal{N}(\boldsymbol{0}, v_n \boldsymbol{I})\,. \qquad \text{(Noise distribution)}$$

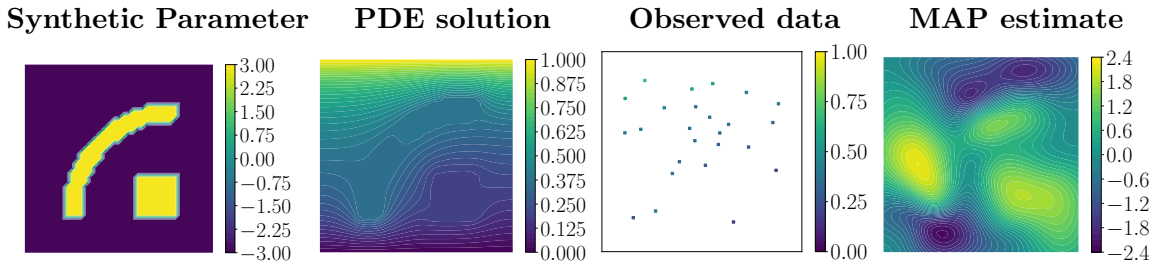

Figure 4: Visualization of the BIP setting and the MAP estimate for coefficient inversion in a nonlinear diffusion–reaction PDE.

The synthetic data, its generating parameter and PDE solution, and the MAP estimate are visualized in Figure 4.

7.4 Neural operator surrogates

We follow the procedure described in Section 4.3 for generating samples for neural network training and testing. Recall that each training sample consists of an i.i.d. random parameter field, a model-predicted observable coefficient vector, and a reduced Jacobian matrix. We compute DIS reduced bases of dimension $r = 200$ using $n_{\text{DIS}} = 1000$ of the generated samples as specified in (24). In particular, the full Jacobian matrix is generated instead of the reduced Jacobian matrix for the first 1000 samples. Then, a generalized eigenvalue problem is solved to compute the DIS reduced bases (Villa et al., 2021; O'Leary-Roseberry and Villa, 2021). Selected DIS basis functions are visualized in Figure 19. Forming reduced Jacobian matrices via columns using a direct solver with reused factorization takes 25% of the computing time for solving the nonlinear PDEs using a direct solver, estimated on average over sample generation.

We employ a simple feed-forward neural network with six hidden layers, each with 400 hidden neurons and a GELU activation function. The neural network is trained using either the conventional L^2_μ or derivative-informed H^1_μ operator learning objective. For each training method, we use a varying number of training samples n_t for the loss function specified in (16) and (18), with $n_t = 125, 250, \ldots, 16000$.

The generalization errors of the observable vector prediction and the reduced Jacobian matrix prediction are estimated using 5000 testing samples. The two types of errors are measured using the relative L^2_μ error defined as follows:

$$\mathcal{E}_{\text{Obs}}(\boldsymbol{\mathcal{G}}, \widetilde{\boldsymbol{\mathcal{G}}}) \coloneqq \sqrt{\mathbb{E}_{M\sim\mu}\left[\frac{\left\|\boldsymbol{\mathcal{G}}(M) - \widetilde{\boldsymbol{\mathcal{G}}}(M)\right\|^2_{\boldsymbol{C}_n^{-1}}}{\|\boldsymbol{\mathcal{G}}(M)\|^2_{\boldsymbol{C}_n^{-1}}}\right]}, \tag{47a}$$

$$\mathcal{E}_{\text{Jac}}(\boldsymbol{J}_r, \widetilde{\boldsymbol{J}_r}) \coloneqq \sqrt{\mathbb{E}_{M\sim\mu}\left[\frac{\left\|\boldsymbol{J}_r(M) - \widetilde{\boldsymbol{J}_r}(M)\right\|^2_F}{\|\boldsymbol{J}_r(M)\|^2_F}\right]}. \tag{47b}$$

The generalization accuracy is defined by $(1-\mathcal{E})\times 100\%$. In Figure 5, we plot the estimated errors as a function of training sample generation cost, measured relative to the averaged cost of one nonlinear PDE solve. For L^2_μ-trained neural operators (NOs), we discount the cost of forming reduced Jacobian matrices; thus, its relative cost is the same as the number of training samples. The error plot includes the relative cost of forming reduced Jacobian matrices for derivative-informed H^1_μ operator learning. Additionally, we provide generalization accuracy values on the error plot.

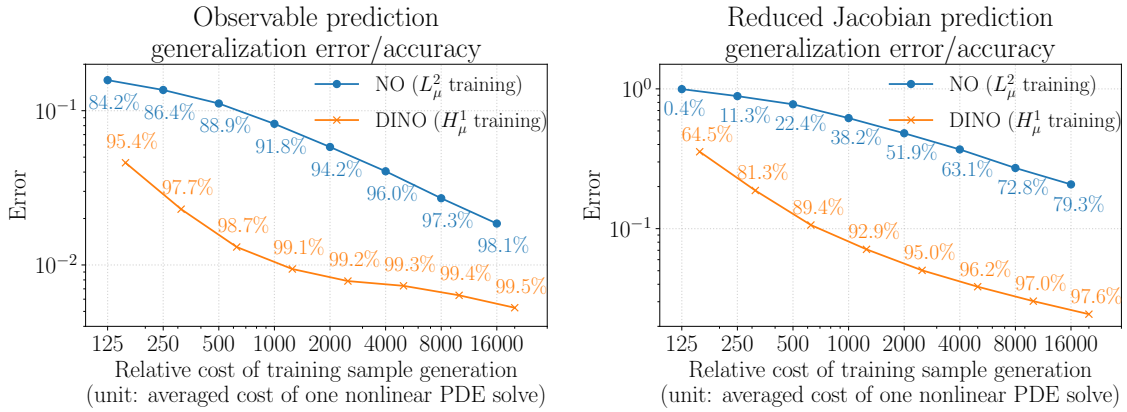

Figure 5: The generalization error and accuracy (47) for predicting the observable vector and the reduced Jacobian matrix via L^2_μ-trained neural operators and H^1_μ-trained DINOs for coefficient inversion in a nonlinear diffusion–reaction PDE. The error is plotted as a function of training sample generation cost, measured relative to the averaged cost of one nonlinear PDE solve.

The plot shows that the derivative-informed H^1_μ operator learning significantly improves the quality of the surrogate at the same training sample generation cost compared to the conventional L^2_μ operator learning. Here is a list of important takeaways from this plot:

- To achieve the same generalization accuracy for predicting the observables or the reduced Jacobian matrix, the derivative-informed H^1_μ operator learning is at least 25 times more efficient than the conventional L^2_μ operator learning measured by training sample generation cost.

- In the small training sample size regime, e.g., $n_t < 1000$, derivative-informed H^1_μ operator learning leads to a much higher generalization error reduction rate for both observable prediction and reduced Jacobian prediction.

- To achieve a similar efficiency in posterior sampling compared to mMALA, the operator surrogate needs an estimated 90% generalization accuracy in reduced Jacobian prediction (see Figure 10). The conventional L^2_μ operator learning may at least (estimated via extrapolation) demand the cost of around 116000 nonlinear PDE solves, while derivative-informed H^1_μ operator learning requires the cost of around 700 nonlinear PDE solves—two orders of magnitude difference in computational cost.

These numerical results indicate that the derivative-informed H^1_μ operator learning provides a much superior cost-accuracy trade-off compared to the conventional L^2_μ operator learning. The superiority of derivative-informed learning is more pronounced for large-scale PDE systems since one typically cannot afford to solve these systems at a large number of parameter samples. The superiority of derivative-informed H^1_μ operator learning is decisive when one expects the trained operator to possess an accurate derivative with respect to a high or infinite-dimensional input.

7.5 MCMC results

We present numerical results on the efficiency of DINO-mMALA and DA-DINO-mMALA methods compared to the baseline and reference MCMC methods. For each method, we collect $n_c = 10$ Markov chains, each with $n_s = 19000$ samples. The step size parameter $\triangle t$ and initialization are chosen by the procedure detailed in Appendix F. The statistics of the MCMC runs and posterior visualization are provided in Appendix G.

7.5.1 The baseline MCMC methods

The diagnostics for the baseline MCMC methods in Table 2 are visualized in Figure 6. The diagnostics show that mMALA produces Markov chains with the most effective posterior samples and the fastest mixing speed among the baseline methods. When comparing methods using the same type of posterior geometry information (see Table 2), MALA is inferior to pCN, and DIS-mMALA is inferior to LA-pCN, even though both MALA and DIS-mMALA include gradient information in their proposal distributions. This result shows the importance of accurate posterior local geometry information (i.e., data misfit Hessians) in MCMC proposal design, as including data misfit gradients alone may compromise the chain quality.

Based on a comparison of the median of ESS%, mMALA outperformed pCN and LA-pCN, yielding 15.7 and 2.3 times more effective samples, respectively. Yet, each MCMC sample generated by mMALA incurs approximately 2.3 times[4] higher computational costs than pCN and LA-pCN. Consequently, LA-pCN and mMALA achieve equivalent speeds in generating effective posterior samples. The effective sampling speedups of mMALA against other baseline methods are provided in Table 4.

7.5.2 Geometric MCMC with surrogate proposals

For this part of the numerical results, we focus on the quality of the operator surrogate proposal by switching off the DA procedure and using the model-predicted data misfit to compute acceptance probability during MCMC runs. In Figure 7, we visualize the diagnostics of Markov chains generated by NO-mMALA, DINO-mMALA, r-mMALA, and baseline methods.

The diagnostics show that DINO-mMALA at $n_t = 2000$ surpasses LA-pCN regarding ESS and mixing speed of MCMC chains. By $n_t = 16000$, the ESS% of DINO-mMALA

4. Major contributing factors to the extra cost at each sample m can be roughly decomposed into (1) forming a discretized Jacobian matrix $\boldsymbol{J}^h(m)$ via adjoint solves, (2) solving the eigenvalue problem for the operator $\boldsymbol{J}(m)\boldsymbol{J}(m)^T \in \mathbb{R}^{25\times 25}$, and (3) forming the decoder via actions of the prior covariance operator on the encoder.

Baseline \ Speedup	mMALA	DINO-mMALA $n_t = 2000$	DINO-mMALA $n_t = 16000$	DA-DINO-mMALA $n_t = 2000$	DA-DINO-mMALA $n_t = 16000$
pCN	6.8	11.9	14.3	26.5	59.8
MALA	9.4	16.5	19.6	30.6	82.7
LA-pCN	1	1.8	2.1	3.9	8.8
DIS-mMALA	5.1	8.9	10.7	19.9	44.9
NO-mMALA $n_t = 16000$	4.8	8.4	10.1	18.7	42.2
DA-NO-mMALA $n_t = 16000$	5	9	10.5	19.5	44

Table 4: The effective sampling speedup of mMALA, DINO-mMALA, and DA-DINO-mMALA against other baseline MCMC methods for coefficient inversion in a nonlinear diffusion–reaction PDE. The speedup measures the relative speed of generating effective samples for an MCMC method compared against another MCMC method; see (42).

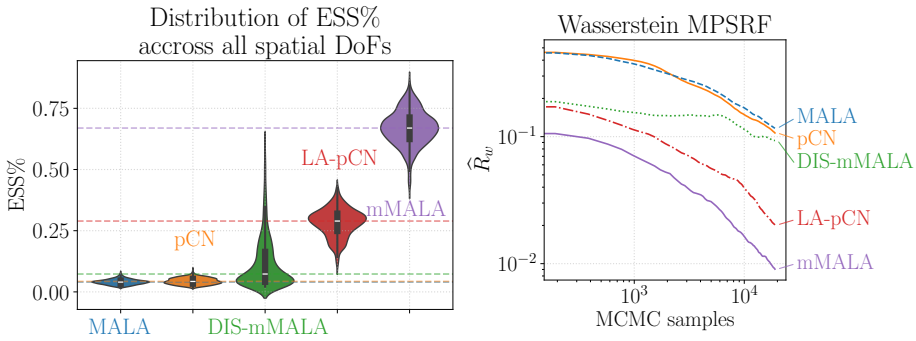

Figure 6: Visualization of the diagnostics (see Section 6.2) of MCMC chains generated by baseline methods listed in Table 2 for coefficient inversion in a nonlinear diffusion–reaction PDE. (*left*) The violin plot of the ESS% distributions over 1,681 spatial DoFs for the discretized parameter space. (*left*) The Wasserstein MPSRF as a function of the Markov chain position.

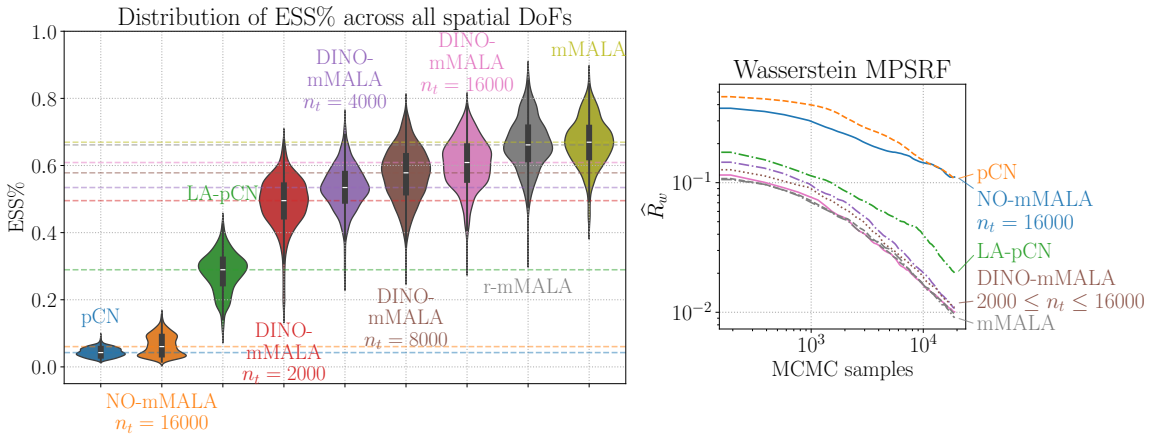

Figure 7: The diagnostics (see Section 6.2) of Markov chains generated by DINO-mMALA, NO-mMALA, and other baseline and reference MCMC methods in Tables 2 and 3 for coefficient inversion in a nonlinear diffusion–reaction PDE. The symbol n_t denotes the training sample size. (*left*) The ESS% diagnostic. (*right*) The Wasserstein MPSRF diagnostic.

is close to the baseline mMALA. Measured by the median ESS%, DINO-mMALA retains 91% of mMALA's ESS despite errors specified in Proposition 7. Estimated by the median ESS% of r-mMALA, the basis truncation and sampling error accounts for 13% of this reduction in ESS%. Accounting for the extra computational cost of mMALA at each MCMC sample, DINO-mMALA at $n_t = 16000$ generates effective samples more than twice as fast as mMALA. See the speedups of DINO-mMALA against other methods in Table 4. These results suggest that DINO-predicted posterior local geometry is sufficiently accurate for enhancing the trade-off between chain quality and computational cost in geometric MCMC.

The diagnostics of chains generated by NO-mMALA at $n_t = 16000$ implies that MCMC driven by L^2_μ-trained NOs lead to low posterior sampling efficiency. While the median ESS% of NO-mMALA is 1.4 times that of pCN, it is only 11% of the median ESS% of DINO-mMALA at $n_t = 16000$. Recall that L^2_μ-trained NO at $n_t = 16000$ has an estimated 98.1% and 79.3% generalization accuracy in observable and reduced Jacobian prediction; see Figure 5. However, to surpass LA-pCN in terms of median ESS%, the reduced Jaocbian prediction accuracy needs to be around 90% (i.e., $n_t \approx 1000$ for H^1_μ-trained DINO), which would require L^2_μ-trained NO at least $n_t = 116000$ to achieve; see comments in Section 7.4.

7.5.3 Delayed acceptance geometric MCMC with surrogate proposals

In Figure 8 (*left*) and Figure 9, we visualize the diagnostics of the Markov chains generated by DA geometric MCMC with operator surrogate proposal; see Algorithm 1. By $n_t = 2000$, DA-DINO-mMALA outperforms LA-pCN regarding ESS and mixing speed. According to the median ESS%, DINO-mMALA retains 73% of mMALA's ESS. Estimated by the ESS% of DA-r-mMALA, the basis truncation error accounts for approximately 36% of this reduction in ESS%. After including the extra computational cost of mMALA at each MCMC sample and the cost reduction of the DA procedure, DA-DINO-mMALA at $n_t = 16000$ generates effective samples around 9 times faster than mMALA.

For MCMC driven by L^2_μ-trained NOs, DA-NO-mMALA at $n_t = 16000$ achieves ESS% and mixing speed similar to pCN. When accounting for the cost reduction of the DA procedure, DA-NO-mMALA at $n_t = 16000$ generates effective samples 1.4 times faster than pCN. However, it is still 5 times slower than LA-pCN and mMALA. Furthermore, it is 19.5 and 44 times slower than DA-DINO-mMALA at $n_t = 2000$ and 16000. See the speedups of DA-DINO-mMALA and DA-NO-mMALA against other methods in Table 4.

In Figure 8 (*right*), we visualize the proposal acceptance rate in the first and second stages of the DA procedure for both DA-DINO-mMALA and DA-NO-mMALA. Recall that a low acceptance rate in the first stage leads to a low computational cost, and the acceptance rate in the second stage is correlated to the accuracy of the surrogate approximation. The plot shows that DA-DINO-mMALA has a high second-stage acceptance rate, improving consistently as the training sample size grows. When comparing between L^2_μ-trained NO and H^1_μ-trained DINO, the second-stage acceptance rate for DA-NO-mMALA is half of the rate for DA-DINO-mMALA, and the first-stage acceptance rate for DA-NO-mMALA is around 3 times the rate for DA-DINO-mMALA.

In Figure 10, we plot the total effective sampling speedups of DA-DINO-mMALA against DA-NO-mMALA at $n_t = 16000$, pCN, LA-pCN and mMALA. Recall from (43) that the total speedups include the offline cost of surrogate construction, and it is a function of the ESS requested for an MCMC run. The total speedups against LA-pCN and mMALA show that if one aims to collect more than just 10 effective posterior samples, it is more cost-efficient to use DA-DINO-mMALA with $n_t = 1000$. On the other hand, the asymptotic speedup as $n_{\text{ess}} \to \infty$ (i.e., the effective sampling speedup in Table 4) at $n_t = 1000$ is relatively small. A better asymptotic speedup requires a better surrogate trained with more samples. At $n_t = 16000$, an asymptotic speedup of 8.8 against LA-pCN and mMALA is achieved, and one needs to collect just 66 effective posterior samples to break even the offline training cost.

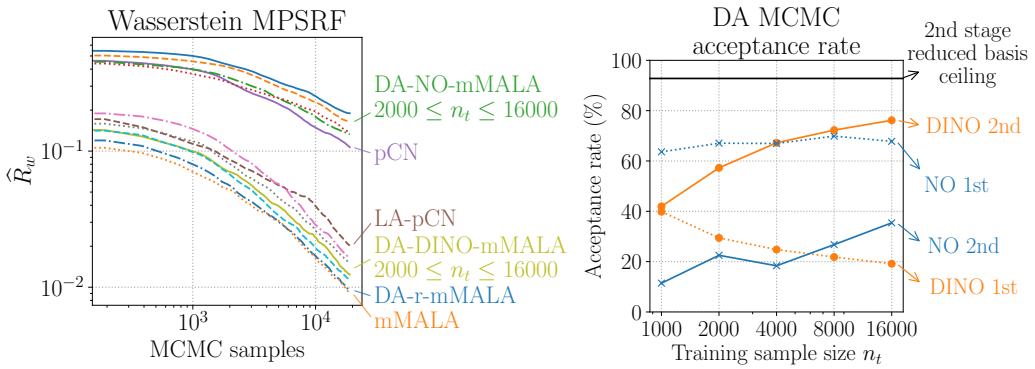

Figure 8: (*left*) The Wasserstein MPSRF diagnostic (see Section 6.2.1) of Markov chains generated by DA-DINO-mMALA, DA-NO-mMALA, and other baseline and reference MCMC methods for coefficient inversion in a nonlinear diffusion–reaction PDE. (*right*) The proposal acceptance rate in the first and second stages of the DA procedure as a function of training sample size. The reduced basis ceiling indicates the estimated (via DA-r-mMALA in Table 3) highest second-stage acceptance rate for the reduced basis architecture with $r = 200$ DIS reduced bases.

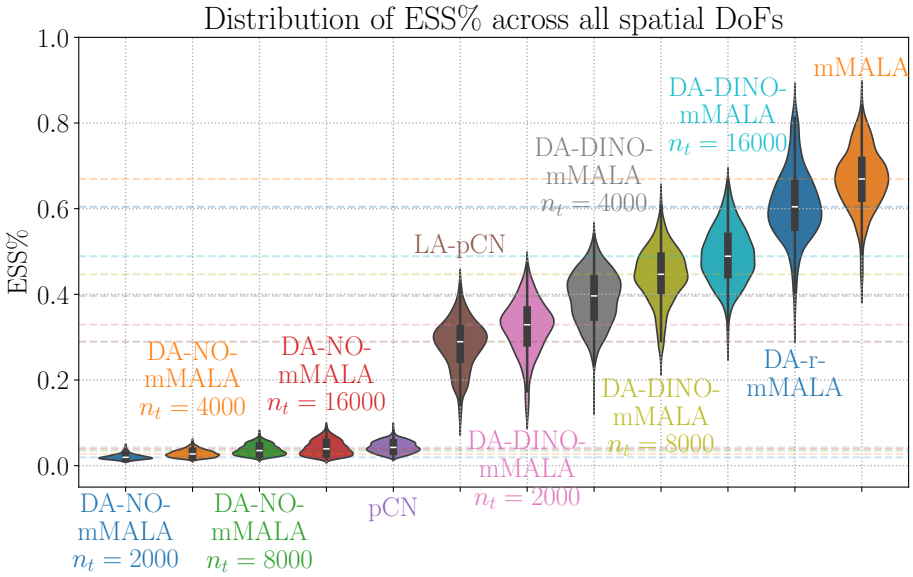

Figure 9: The ESS% diagnostic (see Section 6.2.2) of Markov chains generated by DA-DINO-mMALA, DA-NO-mMALA, and other baseline and reference MCMC methods for coefficient inversion in a nonlinear diffusion–reaction PDE. The symbol n_t denotes the training sample size for operator learning.

8. Numerical example: Inference of a heterogeneous hyperelastic material property

In this section, we consider an experimental scenario where a rectangular thin film of hyperelastic material is stretched on two opposite edges. The inverse problem aims to recover Young's modulus field characterizing spatially varying material strength from measurements of the material deformation.

This section is organized as follows. We introduce the material deformation model, the prior, the PtO map, and the setting for Bayesian inversion in Sections 8.1 to 8.4. Then, we showcase and analyze MCMC results in Section 8.6. In Section 8.6.1, we discuss results on the baseline MCMC methods listed in Table 2. In Section 8.6.2, we discuss results on DA-NO-mMALA and DA-DINO-mMALA listed in Table 3.

8.1 The neo-Hookean model for hyperelastic material deformation

Let $\Omega = (0,1) \times (0,2)$ be a normalized reference configuration for the hyperelastic material that follows the plane strain assumption. The material coordinates $\boldsymbol{x} \in \Omega$ of the reference configuration are mapped to the spatial coordinates $\boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ of the deformed configuration, where $\boldsymbol{u} : \Omega \to \mathbb{R}^2$ is the material displacement. Internal forces are developed as the material deforms. These internal forces depend on the underlying stored internal energy; for a hyperelastic material, the strain energy $\mathcal{W}_e$ depends on the deformation gradient, i.e., $\mathcal{W}_e = \mathcal{W}_e(\boldsymbol{F})$ where $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}$ and $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ is the identity matrix. We consider the neo-Hookean model for the strain energy density (Marsden and Hughes, 1994; Ogden, 1997;

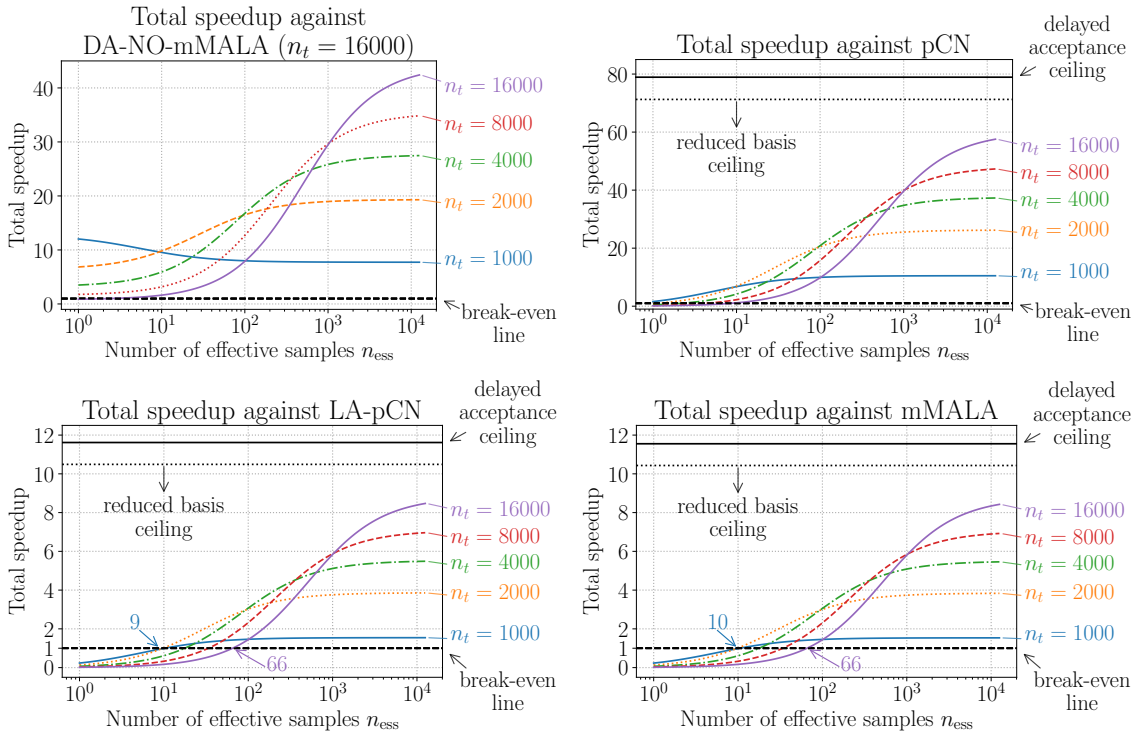

Figure 10: Total effective sampling speedups of DA-DINO-mMALA against DA-NO-mMALA, pCN, LA-pCN, mMALA as a function of ESS collected in an MCMC run for coefficient inversion in a nonlinear diffusion–reaction PDE. The total speedup in (43) compares the relative speed of an MCMC method for generating effective samples when including all computational costs, offline (e.g., training and MAP estimate) and online (MCMC). The break-even line indicates an equal total effective sampling speed of the two MCMC methods. The reduced basis ceiling indicates the estimated (via DA-r-mMALA in Table 3) optimal speedup for the reduced basis architecture with $r = 200$ DIS reduced bases. The delayed acceptance ceiling indicates the estimated asymptotic total speedup when the operator surrogate has no error (i.e., model-evaluated PtO map and reduced Jacobian), which leads to 100% second stage acceptance rate. The symbol n_t denotes the training sample size.

Gonzalez and Stuart, 2008):

$$\mathcal{W}_e(\boldsymbol{F}) = \frac{\mu}{2}(\mathrm{tr}(\boldsymbol{F}^T\boldsymbol{F}) - 3) + \frac{\lambda}{2}\left(\ln\det(\boldsymbol{F})\right)^2 - \mu \ln\det(\boldsymbol{F})\,. \tag{48}$$

Here, λ and μ are the Lamé parameters, which are assumed to be related to Young's modulus of elasticity, E, and Poisson ratio, ν, as follows:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}\,, \qquad \mu = \frac{E}{2(1+\nu)}\,. \tag{49}$$

We assume Poisson ratio $\nu = 0.4$, and an uncertain and spatially-varying Young's modulus, $E : \Omega \to (E_{\min}, E_{\max})$ with $0 < E_{\min} < E_{\max}$. We represent E through a parameter field $m : \Omega \to \mathbb{R}$ as follows

$$E(m(\boldsymbol{x})) = \frac{1}{2}\left(E_{\max} - E_{\min}\right)\left(\mathrm{erf}(m(\boldsymbol{x})) + 1\right) + E_{\min}\,,$$

where $\mathrm{erf} : \mathbb{R} \to (-1, 1)$ is the error function.

The first Piola–Kirchhoff stress tensor is given by $\boldsymbol{P}_e(m, \boldsymbol{F}) = 2\partial\mathcal{W}_e(m, \boldsymbol{F})/\partial\boldsymbol{F}$. Assuming a quasi-static model with negligible body forces, the balance of linear momentum leads to the following nonlinear PDE:

$$\begin{aligned}
\nabla \cdot \boldsymbol{P}_e(m(\boldsymbol{x}), \boldsymbol{F}(\boldsymbol{x})) &= \boldsymbol{0}\,, && \boldsymbol{x} \in \Omega\,; && \text{(50a)}\\
\boldsymbol{u}(\boldsymbol{x}) &= \boldsymbol{0}\,, && \boldsymbol{x} \in \Gamma_{\text{left}}\,; && \text{(50b)}\\
\boldsymbol{u}(\boldsymbol{x}) &= 3/2\,, && \boldsymbol{x} \in \Gamma_{\text{right}}\,; && \text{(50c)}\\
\boldsymbol{P}_e(m(\boldsymbol{x}), \boldsymbol{F}(\boldsymbol{x})) \cdot \boldsymbol{n} &= \boldsymbol{0}\,, && \boldsymbol{x} \in \Gamma_{\text{top}} \cup \Gamma_{\text{bottom}}; && \text{(50d)}
\end{aligned}$$

where Γ_t, Γ_r, Γ_b, and Γ_l denote the material domain's top, right, bottom, and left boundary. Notice that the stretching is enforced as a Dirichlet boundary condition, and the strain specified on the Γ_{right} is 0.75.

8.2 The prior distribution

The normalized Young's modulus follows a prior distribution defined through a Gaussian random field $M \sim \mu$ with $E_{\min} = 1$ and $E_{\max} = 7$:

$$\begin{aligned}
\mathscr{M} &\coloneqq L^2(\Omega)\,, && \text{(Parameter space)}\\
\mu &\coloneqq \mathcal{N}(0, (-0.3\nabla \cdot \boldsymbol{A}\nabla + 3.3\mathcal{I}_{\mathscr{M}})^{-2})\,, && \text{(The prior distribution)}
\end{aligned}$$

where the differential operator is equipped with a Robin boundary for eliminating boundary effects, and $\boldsymbol{A} \in \mathbb{R}^{2\times 2}$ is a symmetric positive definite anisotropic tensor given by

$$\boldsymbol{A} = \begin{bmatrix} \theta_1 \sin(\alpha)^2 & (\theta_1 - \theta_2)\sin(\alpha)\cos(\alpha) \\ (\theta_1 - \theta_2)\sin(\alpha)\cos(\alpha) & \theta_2 \cos(\alpha)^2 \end{bmatrix}\,,$$

where $\theta_1 = 2$, $\theta_2 = 1/2$, $\alpha = \arctan(2)$. The resulting Gaussian random field has a pointwise variance of 1 and a spatial correlation of 2 and 1/2 perpendicular and along the left bottom to the right top diagonal of the spatial domain. We approximate $\mathscr{M}$ using a finite element space $\mathscr{M}^h$ constructed by linear triangular finite elements with 2145 DoFs. Prior samples are visualized in Figure 11.

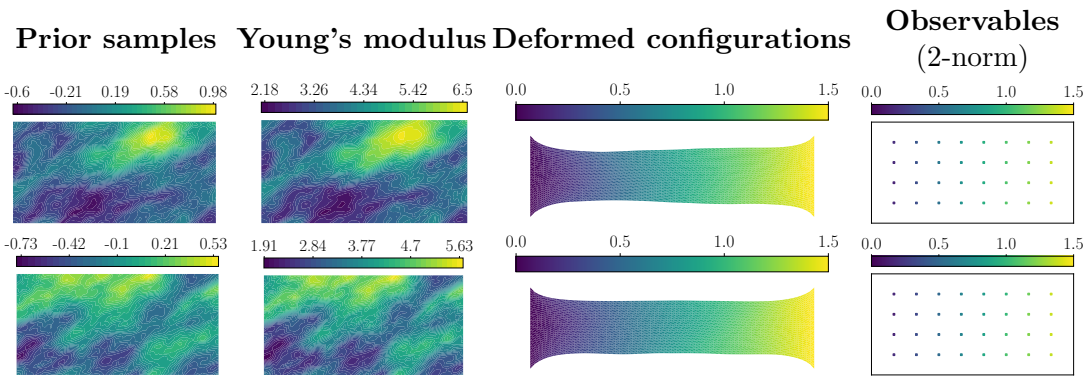

Figure 11: Visualizations of prior samples (2145 DoFs), deformed configuration (16770 DoFs), and predicted observables (in $\mathbb{R}^{64}$) for hyperelastic material property discovery.

8.3 The parameter-to-observable map

We consider a symmetric variational formulation for hyperelastic material deformation and define the following Hilbert spaces following the notation in Section 4.3:

$$\mathscr{U} := \left\{ \boldsymbol{u} \in H^1(\Omega; \mathbb{R}^2) \;\middle|\; \boldsymbol{u}|_{\Gamma_l} = \boldsymbol{0} \wedge \boldsymbol{u}|_{\Gamma_r} = \boldsymbol{0} \right\}; \qquad \text{(State space)}$$

$$\mathscr{V} := \mathscr{U}', \qquad \text{(Residual space)}$$

where $\mathscr{U}'$ denotes the dual space of $\mathscr{U}$. To enforce the inhomogeneous Dirichlet boundary condition, we decompose the displacement $\boldsymbol{u}$ into $\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{B}\boldsymbol{x}$, where $\boldsymbol{B} = \begin{bmatrix} 3/4 & 0 \\ 0 & 1 \end{bmatrix}$ and $\boldsymbol{u}_0 \in \mathscr{U}$ is the PDE state with the homogenous Dirichlet boundary condition.

The residual operator $\mathcal{R} : \mathscr{U} \times \mathscr{M} \to \mathscr{V}$ is defined by its action on an arbitrary test function $\boldsymbol{p} \in \mathscr{U}$:

$$\langle \mathcal{R}(\boldsymbol{u}_0, m), p \rangle_{\mathscr{U}' \times \mathscr{U}} := \int_\Omega \boldsymbol{P}_e \left(m(\boldsymbol{x}), \boldsymbol{I} + \nabla \boldsymbol{u}_0(\boldsymbol{x}) + \boldsymbol{B} \right) \nabla \boldsymbol{p}(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \,. \tag{51}$$

The effective PDE solution operator $\mathcal{F} : \mathscr{M} \ni m \mapsto \boldsymbol{u}_0 \in \mathscr{U}$ satisfies $\mathcal{R}(\mathcal{F}(m), m) = 0$. We approximate $\mathscr{U}$ using a finite element space $\mathscr{U}^h$ constructed by quadratic triangular elements with 16770 DoFs. Evaluating the discretized PDE solution operator involves solving the discretized residual norm minimization problem via the Newton–Raphson method in $\mathscr{U}^h$.

We define a observation operator $\boldsymbol{\mathcal{O}} \in \mathscr{U} \to \mathbb{R}^{64}$ using 32 equally spaced discrete interior points $\{\boldsymbol{x}_{\text{obs}}^{(j)}\}_{j=1}^{32}$ similar to (46). The PtO map is $\boldsymbol{\mathcal{G}} := \boldsymbol{\mathcal{O}} \circ \mathcal{F}$. We visualize the output of the PtO map in Figure 11.

8.4 Bayesian inverse problem settings

We generate synthetic data for our BIP using a prior sample. The model-predicted observables at the synthetic parameter field are corrupted with 1% additive white noise, which has a noise covariance matrix of identity scaled by $v_n = 1.8 \times 10^{-4}$. The synthetic data, its generating parameter and PDE solution, and the MAP estimate are visualized in Figure 12.

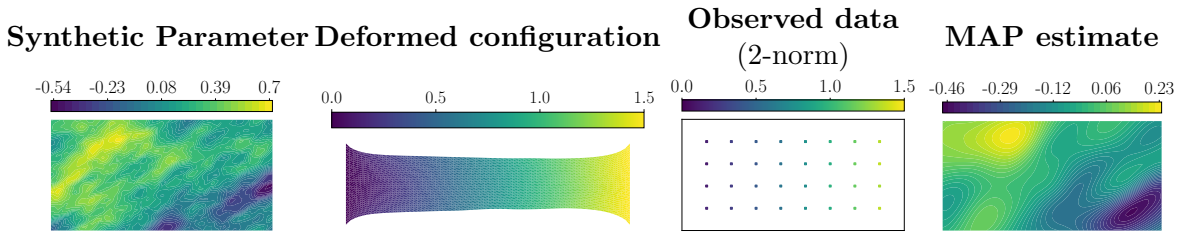

Figure 12: Visualization of the BIP setting and the MAP estimate for hyperelastic material property discovery.

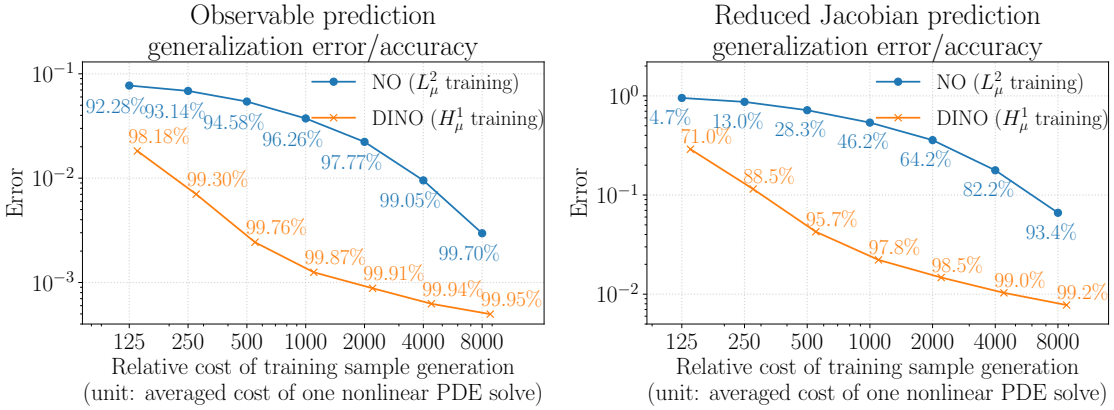

Figure 13: The generalization error and accuracy (47) of observable prediction and reduced Jacobian prediction made by L^2_μ-trained neural operators and H^1_μ-trained DINOs for inference of a heterogeneous hyperelastic material property. The error is plotted as a function of training sample generation cost, measured relative to the averaged cost of one nonlinear PDE solve.

8.5 Neural operator surrogates

We follow the procedure described in Section 4.3 for generating samples for neural network training and testing. We compute DIS reduced bases of dimension $r = 200$ using $n_{\text{DIS}} = 500$ of the generated samples as specified in (24). Selected DIS basis functions are visualized in Figure 21. Forming reduced Jacobian matrices via columns using a direct solver with reused factorization takes 10% of the computing time for solving the nonlinear PDEs using a direct solver, estimated on average over sample generation. Note that the relative cost of forming reduced Jacobian is low mainly because a large number of Newton–Raphson iterations are needed to solve the PDE.

We use a simple feed-forward neural network architecture with six hidden layers, each with 400 hidden neurons and a GELU activation function, trained using $n_t = 125, 250, \ldots,$ 8000 samples. We estimate the generalization errors of the trained neural networks using 2500 testing samples. In Figure 13, we plot the estimated errors as a function of training sample generation cost, measured relative to the averaged cost of one nonlinear PDE solve.

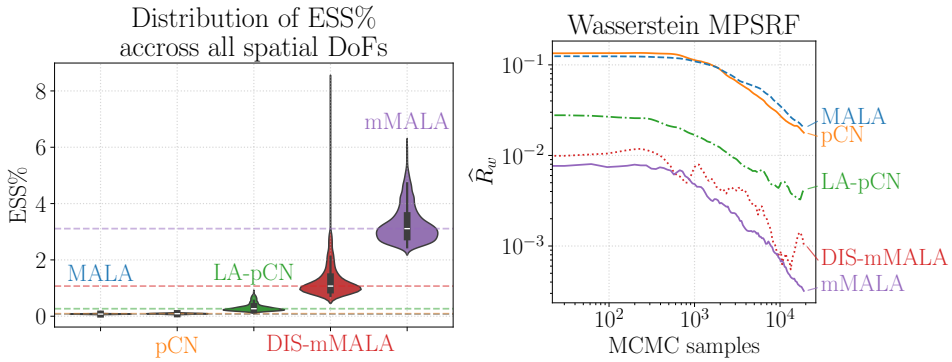

Figure 14: Visualization of the diagnostics (see Section 6.2) of MCMC chains generated by baseline MCMC methods listed in Table 2 for inference of a heterogeneous hyperelastic material property. (*left*) The violin plot of the ESS% distributions over 2,145 spatial DoFs for the discretized parameter space. We note that the ESS% of DIS-mMALA is larger than that of mMALA at only 5 spatial DoFs. (*left*) The Wasserstein MPSRF as a function of the Markov chain position.

The plot leads us to similar conclusions outlined in Section 7.4. It shows that the derivative-informed H^1_μ operator learning significantly enhances the quality of the neural operator surrogate at the same training sample generation cost compared to the conventional L^2_μ operator learning. Notably, achieving comparable observable and reduced Jacobian prediction generalization accuracy requires approximately 16 times fewer training samples with H^1_μ training.

8.6 MCMC results

We present numerical results on the efficiency of DA-DINO-mMALA compared to the baseline MCMC methods. For each method, we collect $n_c = 10$ Markov chains with different initialization, each with $n_s = 19000$ samples. The step size parameter $\triangle t$ and initialization are chosen carefully according to the procedure detailed in Appendix F. The statistics of the MCMC runs and posterior visualization are provided in Appendix G.

8.6.1 The baseline MCMC methods

In Figure 14, we visualize the diagnostics for the baseline MCMC methods in Table 2. The diagnostics show that mMALA produces Markov chains with the most effective samples and the fastest mixing time among the baseline methods. When comparing methods with the same type of posterior geometry information (see Table 2), MALA is slightly inferior to pCN, and LA-pCN is much inferior to DIS-mMALA.

Comparing the median of ESS%, mMALA produces 35 and 3 times more effective samples than pCN and DIS-mMALA. Notably, the ESS% of DIS-mMALA is larger than that of mMALA for just 5 DoFs out of 2145. Moreover, due to the large number of iterative solves required for each PtO map evaluation, each Markov chain sample generated by mMALA is only around 1.18 and 1.05 times more computationally costly than pCN and DIS-mMALA.

Speedup / **Baseline**	mMALA	DA-DINO-mMALA $n_t = 500$	DA-DINO-mMALA $n_t = 4000$
pCN	29.6	72.7	96.2
MALA	38.8	93.2	126.4
LA-pCN	9.9	24.4	32.3
DIS-mMALA	2.8	6.8	9
mMALA	1	2.5	3.3
DA-NO-mMALA $n_t = 4000$	5.5	13.5	17.9

Table 5: The effective sampling speedup of mMALA, DINO-mMALA, and DA-DINO-mMALA against other baseline MCMC methods for inference of a heterogeneous hyperelastic material property. The speedup measures the relative speed of generating effective samples for an MCMC method compared against another MCMC method; see (42).

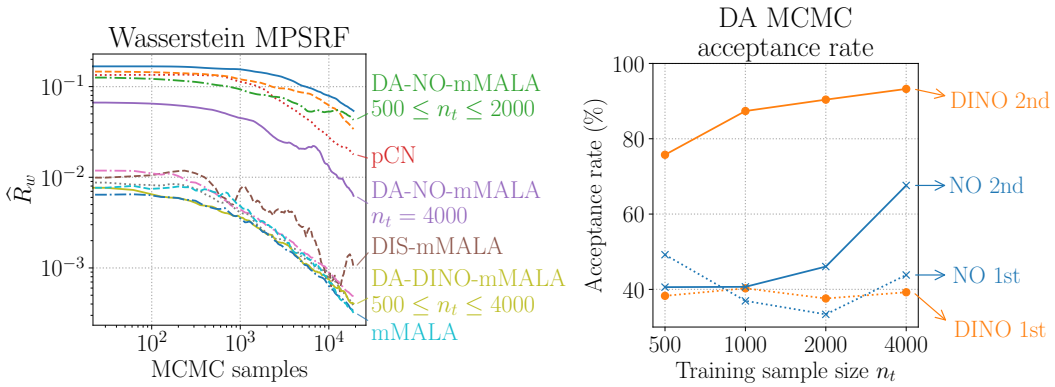

Figure 15: (*left*) The Wasserstein MPSRF diagnostic (see Section 6.2.1) of Markov chains generated by DA-DINO-mMALA, DA-NO-mMALA, and other baseline MCMC methods for inference of a heterogeneous hyperelastic material property. (*right*) The proposal acceptance rate in the first and second stages of the DA procedure as a function of training sample size.

The effective sampling speedups (42) of mMALA against other baseline MCMC methods are provided in Table 5.

8.6.2 Delayed acceptance geometric MCMC with surrogate proposals

In Figure 15 (*left*) and Figure 16, we visualize the diagnostics of the Markov chains generated by the surrogate-driven geometric MCMC with DA. The diagnostics show that DA-DINO-mMALA at $n_t = 500$ and beyond outperforms DIS-mMALA regarding the ESS and mixing speed of MCMC chains. Furthermore, the ESS% of DA-DINO-mMALA plateaued at around $n_t = 1000$, meaning the ESS% fluctuation is dominated by (i) imperfect step size tuning, (ii) finite chain length, and (iii) finite Markov chain pool sizes. Accounting for the extra computational cost of mMALA at each MCMC sample and the cost reduction of the DA procedure, DA-DINO-mMALA at $n_t = 4000$ generates effective samples around 3.3 times faster than mMALA.

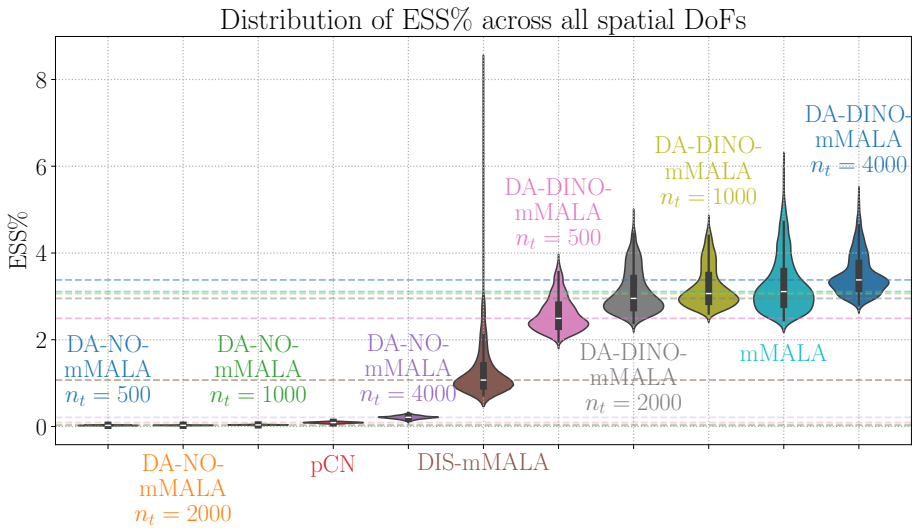

Figure 16: The ESS% diagnostic (see Section 6.2.2) of Markov chains generated by DA-DINO-mMALA, DA-NO-mMALA, and other baseline MCMC methods for inference of a heterogeneous hyperelastic material property. The symbol n_t denotes the training sample size.

Driven by L^2_μ-trained NOs, DA-NO-mMALA at $n_t = 4000$ surpasses pCN regarding the ESS and mixing speed. After including the cost reduction of the DA procedure, DA-NO-mMALA at $n_t = 4000$ generates effective samples 5 and 2 times faster than pCN and LA-pCN. However, it is still 2 and 5.5 times slower than DIS-mMALA and mMALA. Furthermore, it is 13.5 and 17.9 times slower than DA-DINO-mMALA at $n_t = 500$ and 4000. See the speedups of DA-DINO-mMALA and DA-NO-mMALA against other methods in Table 4.

In Figure 15 (*right*), we visualize the proposal acceptance rate in the first and second stages of the DA procedure for both DA-DINO-mMALA and DA-NO-mMALA. The plot shows that DA-DINO-mMALA has a high second-stage acceptance rate, improving consistently as the training sample size grows. The second-stage acceptance rate for DA-NO-mMALA is 1.3–2.3 times the rate for DA-DINO-mMALA. These results affirm that H^1_μ-trained DINO leads to higher quality Markov chains for posterior sampling.

In Figure 17, we plot the total effective sampling speedups of DA-DINO-mMALA against DA-NO-mMALA at $n_t = 4000$, pCN, LA-pCN, and mMALA. The total speedups against LA-pCN and mMALA show that if one aims to collect more than 25 effective posterior samples, switching to DA-DINO-mMALA with $n_t = 500$ is more cost-efficient than using mMALA. On the other hand, the asymptotic speedup at $n_t = 500$ is relatively small. We achieve an asymptotic speedup of 3.3 against mMALA, and one only needs to collect 171 effective posterior samples to break even the offline cost of surrogate training.

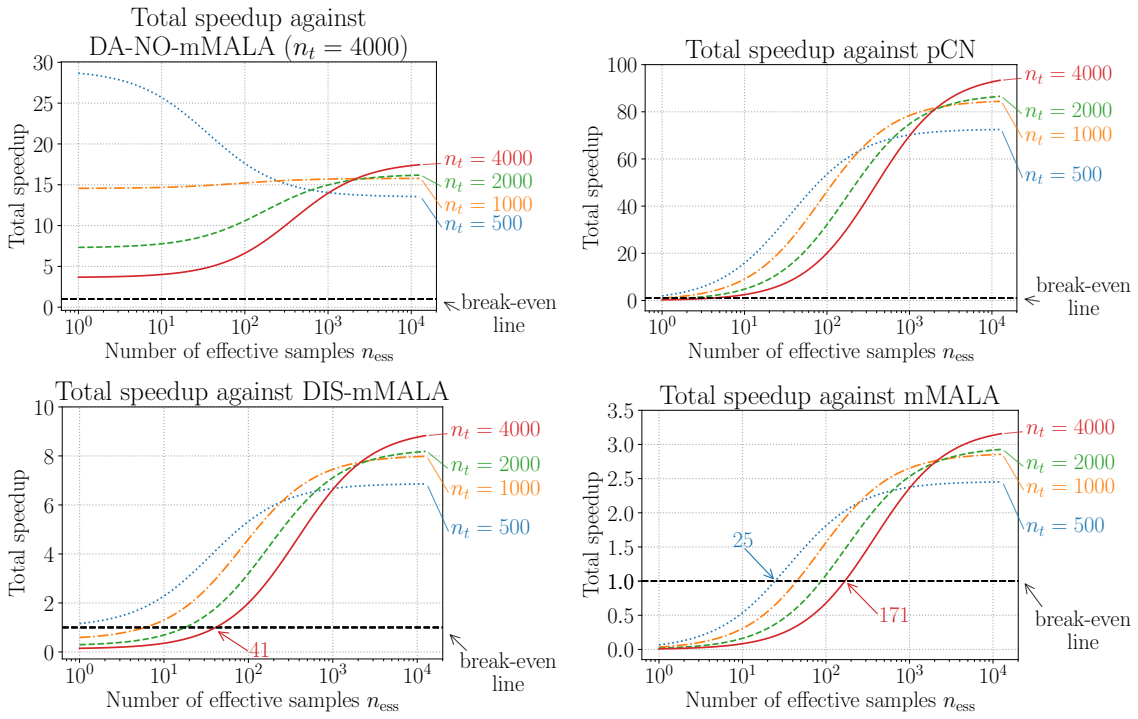

Figure 17: Total effective sampling speedups of DA-DINO-mMALA against DA-NO-mMALA, pCN, DIS-mMALA, and mMALA as a function of ESS collected in an MCMC run for inference of a heterogeneous hyperelastic material property. The total speedup in (43) compares the relative speed of an MCMC method for generating effective samples when considering all computational costs, offline (e.g., training and MAP estimate) and online (MCMC). The break-even line indicates an equal total effective sampling speed of the two MCMC methods. The symbol n_t denotes the training sample size.

9. Conclusion

In this work, we propose deploying a neural operator surrogate of the PtO map to accelerate geometric MCMC and obtain fast and consistent solutions to infinite-dimensional Bayesian inverse problems. The method represents a synthesis of ideas from reduced basis DINO, DA MCMC, and dimension-independent geometric MCMC with the goal of designing an MCMC proposal that adapts to DINO-predicted posterior local geometry within a delayed acceptance procedure. Compared to conventional geometric MCMC, this surrogate-driven geometric MCMC method leads to significant cost reduction, as it requires no online forward or adjoint sensitivity solves, fewer model evaluations, and fewer instances of prior sampling. Our numerical results show that our proposed method can produce high-quality Markov chains typical of a geometric MCMC method at a much lower cost, leading to substantial speedups in posterior sample generation. In particular, our numerical examples show that DA-DINO-mMALA generates effective posterior samples 60–97 times faster than pCN and 3–9 times faster than mMALA. Moreover, the training cost of DINO surrogates breaks even after collecting just 10–25 effective posterior samples compared to mMALA.

Our derivative-informed operator learning formulation is the key to enabling surrogate acceleration of geometric MCMC. We present an operator learning objective in H^1_μ Sobolev space with Gaussian measure that controls error in approximating the stochastic derivative of the PtO map. This formulation is naturally equipped with the Poincaré inequality for nonlinear mappings on function spaces, which is used to derive a L^2_μ approximation error bound for the reduced basis neural operator surrogate consisting of three sources of error: (i) neural network approximation of the optimal reduced mapping, (ii) basis truncation error, and (iii) sampling error when applicable. Our numerical examples show that derivative-informed H^1_μ operator learning achieves similar generalization accuracy in predicting the observable vector and the reduced Jacobian of the PtO map with at least 16–25 times fewer training sample generation cost than conventional L^2_μ operator learning. For coefficient inversion in a nonlinear diffusion–reaction PDE, we observe an estimated 166 times difference in training sample generation cost between derivative-informed H^1_μ and conventional L^2_μ operator learning for achieving an acceleration of mMALA.

We believe that additional work in the following directions may help improve the capability of our proposed MCMC method. First, our method can benefit from various adaptive techniques in proposal design and surrogate fine-tuning for MCMC; see, e.g., Lan (2019) and Yan and Zhou (2020). Second, the numerical studies in this work focus on nonlinear steady-state PDE problems, while time-dependent problems often have unique challenges, such as memory cost and adjoint consistency. Detailed studies on time-dependent problems could broaden the insight into the proposed method.

Acknowledgments

This work was partially supported by the National Science Foundation under awards OAC-2313033 and DMS-234643, and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under awards DE-SC0021239 and DE-SC0023171, and the Air Force Office of Scientific Research under MURI grant FA9550-21-1-0084. The work of Lianghao Cao was partially supported by a Department of Defense Vannevar Bush

Faculty Fellowship held by Andrew M. Stuart, and by the SciAI Center, funded by the Office of Naval Research, under Grant Number N00014-23-1-2729. This work benefited from discussions with Dingcheng Luo and Jakob Zech.

References

Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E Rognes, and Garth N Wells. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

Harbir Antil and Dmitriy Leykekhman. *A Brief Introduction to PDE-Constrained Optimization*, pages 3–40. Springer New York, New York, NY, 2018. doi: 10.1007/978-1-4939-8636-1_1.

Ricardo Baptista, Youssef Marzouk, and Olivier Zahm. Gradient-based data and parameter dimension reduction for Bayesian models: An information theoretic perspective. *arXiv preprint, arXiv.2207.08670*, 2022.

Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E. Farrell, and Andrew M. Stuart. Geometric MCMC for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2016.12.041.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural network for parametric PDEs. *The SMAI Journal of computational mathematics*, 7:121–157, 2021. doi: 10.5802/smai-jcm.74.

Lorenz Biegler, George Biros, Omar Ghattas, Matthias Heinkenschloss, David Keyes, Bani Mallick, Youssef Marzouk, Luis Tenorio, Bart van Bloemen Waanders, and Karen Willcox, editors. *Large-Scale Inverse Problems and Quantification of Uncertainty*. John Wiley & Sons, Ltd, 2010. ISBN 9780470685853. doi: 10.1002/9780470685853.

Vladimir Igorevich Bogachev. *Gaussian measures*, volume 62 of *Mathematical Surveys and Monographs*. American Mathematical Society, 1998.

Stephen P. Brooks and Andrew Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7(4):434–455, 1998. doi: 10.1080/10618600.1998.10474787.

T Bui-Thanh and M Girolami. Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo. *Inverse Problems*, 30(11):114014, oct 2014. doi: 10.1088/0266-5611/30/11/114014.

Tan Bui-Thanh and Quoc P. Nguyen. FEM-based discretization-invariant MCMC methods for PDE-constrained Bayesian inverse problems. *Inverse Problems and Imaging*, 10(4): 943–975, 2016. ISSN 1930-8337. doi: 10.3934/ipi.2016028.

Tan Bui-Thanh, Omar Ghattas, James Martin, and Georg Stadler. A computational framework for infinite-dimensional Bayesian inverse problems part I: The linearized case, with

application to global seismic inversion. *SIAM Journal on Scientific Computing*, 35:A2494–A2523, 2013. doi: 10.1137/12089586X.

Lianghao Cao, Thomas O'Leary-Roseberry, Prashant K. Jha, J. Tinsley Oden, and Omar Ghattas. Residual-based error correction for neural operator accelerated infinite-dimensional Bayesian inverse problems. *Journal of Computational Physics*, 486:112104, 2023. doi: 10.1016/j.jcp.2023.112104.

Lianghao Cao, Joshua Chen, Michael Brennan, Thomas O'Leary-Roseberry, Youssef Marzouk, and Omar Ghattas. LazyDINO: Fast, scalable, and efficiently amortized Bayesian inversion via structure-exploiting and surrogate-driven measure transport. *arXiv preprint*, 2024a. doi: 10.48550/arXiv.2411.12726.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024b. doi: 10.1038/s42256-024-00844-4.

J. Andrés Christen and Colin Fox. Markov chain Monte Carlo using an approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005. doi: 10.1198/106186005X76983.

Philippe G. Ciarlet. *Linear and Nonlinear Functional Analysis with Applications*. the Society for Industrial and Applied Mathematics, 2013.

Paul G. Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014. doi: 10.1137/130916138.

Paul G. Constantine, Carson Kent, and Tan Bui-Thanh. Accelerating Markov chain Monte Carlo with active subspaces. *SIAM Journal on Scientific Computing*, 38(5):A2779–A2805, 2016. doi: 10.1137/15M1042127.

S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White. MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28:424–446, 2013. doi: 10.1214/13-STS421.

Tiangang Cui and Olivier Zahm. Data-free likelihood-informed dimension reduction of Bayesian inverse problems. *Inverse Problems*, 37(4):045009, mar 2021. doi: 10.1088/1361-6420/abeafb.

Tiangang Cui, Youssef M. Marzouk, and Karen E. Willcox. Data-driven model reduction for the Bayesian solution of inverse problems. *International Journal for Numerical Methods in Engineering*, 102(5):966–990, 2015. doi: 10.1002/nme.4748.

Tiangang Cui, Kody J.H. Law, and Youssef M. Marzouk. Dimension-independent likelihood-informed MCMC. *Journal of Computational Physics*, 304:109–137, 2016. ISSN 0021-9991. doi: 10.1016/j.jcp.2015.10.008.

Tiangang Cui, Gianluca Detommaso, and Robert Scheichl. Multilevel dimension-independent likelihood-informed MCMC for large-scale inverse problems. *Inverse Problems*, 40(3):035005, feb 2024. doi: 10.1088/1361-6420/ad1e2c.

Giuseppe Da Prato and Jerzy Zabczyk. *Second order partial differential equations in Hilbert spaces*, volume 293. Cambridge University Press, 2002.

M Dashti, K J H Law, A M Stuart, and J Voss. MAP estimators and their consistency in Bayesian nonparametric inverse problems. *Inverse Problems*, 29(9):095017, sep 2013. doi: 10.1088/0266-5611/29/9/095017.

Timothy A Davis, Sivasankaran Rajamanickam, and Wissam M Sid-Lakhdar. A survey of direct methods for sparse linear systems. *Acta Numerica*, 25:383–566, 2016.

Maarten V. de Hoop, Daniel Zhengyu, Huang, Elizabeth Qian, and Andrew M. Stuart. The cost-accuracy trade-off in operator learning with neural networks. *Journal of Machine Learning*, 1(3):299–341, 2022. ISSN 2790-2048. doi: 10.4208/jml.220509.

T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup. Multilevel Markov chain Monte Carlo. *SIAM Review*, 61(3):509–545, 2019. doi: 10.1137/19M126966X.

D.C Dowson and B.V Landau. The Fréchet distance between multivariate normal distributions. *Journal of Multivariate Analysis*, 12(3):450–455, 1982. ISSN 0047-259X. doi: 10.1016/0047-259X(82)90077-X.

Y. Efendiev, T. Hou, and W. Luo. Preconditioning Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing*, 28(2):776–803, 2006. doi: 10.1137/050628568.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC texts in statistical science. CRC Press, Boca Raton, Florida, 3rd edition, 2014. ISBN 9781439840955.

Omar Ghattas and Karen Willcox. Learning physics-based models from data: Perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554, 2021. doi: 10.1017/S0962492921000064.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011. doi: 10.1111/j.1467-9868.2010.00765.x.

Jinwoo Go and Peng Chen. Sequential infinite-dimensional Bayesian optimal experimental design with derivative-informed latent attention neural operator. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2409.09141.

Jinwoo Go and Peng Chen. Accurate, scalable, and efficient bayesian optimal experimental design with derivative-informed neural operators. *Computer Methods in Applied Mechanics and Engineering*, 438:117845, 2025. ISSN 0045-7825. doi: 10.1016/j.cma.2025.117845.

Israel Gohberg, Seymour Goldberg, and Nahum Krupnik. *Traces and Determinants of Linear Operators*, volume 116 of *Operator Theory: Advances and Applications*. Birkhäuser Basel, 2012. doi: 10.1007/978-3-0348-8401-3.

Oscar Gonzalez and Andrew M Stuart. *A first course in continuum mechanics*, volume 42. Cambridge University Press, 2008.

Martin Hairer, Andrew M. Stuart, and Sebastian J. Vollmer. Spectral gaps for a Metropolis–Hastings algorithm in infinite dimensions. *The Annals of Applied Probability*, 24(6):2455–2490, 2014. ISSN 10505164.

W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 00063444.

J.S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018. ISSN 0021-9991. doi: doi.org/10.1016/j.jcp.2018.02.037.

Viet Ha Hoang, Christoph Schwab, and Andrew M Stuart. Complexity analysis of accelerated MCMC methods for Bayesian inversion. *Inverse Problems*, 29(8):085010, jul 2013. doi: 10.1088/0266-5611/29/8/085010.

Xun Huan, Jayanth Jagalur, and Youssef Marzouk. Optimal experimental design: Formulations and computations. *Acta Numerica*, 33:715–840, 2024. doi: 10.1017/S0962492924000023.

Victor Kac and Pokman Cheung. *Quantum Calculus*, volume 113 of *Universitext*. Springer New York, NY, 2002. doi: 10.1007/978-1-4613-0071-7.

Ki-Tae Kim, Umberto Villa, Matthew Parno, Youssef Marzouk, Omar Ghattas, and Noemi Petra. hIPPYlib-MUQ: A Bayesian inference software framework for integration of data with complex predictive models under uncertainty. *ACM Trans. Math. Softw.*, 49(2), jun 2023. ISSN 0098-3500. doi: 10.1145/3580278.

Drew P Kouri and Alexander Shapiro. Optimization of PDEs with uncertain inputs. *Frontiers in PDE-constrained optimization*, pages 41–81, 2018.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24 (89):1–97, 2023. URL `http://jmlr.org/papers/v24/21-1524.html`.

Nikola B. Kovachki, Samuel Lanthaler, and Andrew M. Stuart. Operator learning: Algorithms and analysis. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2402.15715.

Remi R. Lam, Olivier Zahm, Youssef M. Marzouk, and Karen E. Willcox. Multifidelity dimension reduction via active subspaces. *SIAM Journal on Scientific Computing*, 42(2): A929–A956, 2020. doi: 10.1137/18M1214123.

Shiwei Lan. Adaptive dimension reduction to accelerate infinite-dimensional geometric Markov chain Monte Carlo. *Journal of Computational Physics*, 392:71–95, 2019. ISSN 0021-9991. doi: 10.1016/j.jcp.2019.04.043.

Shiwei Lan, Tan Bui-Thanh, Mike Christie, and Mark Girolami. Emulation of higher-order tensors in manifold Monte Carlo methods for Bayesian inverse problems. *Journal of Computational Physics*, 308:81–101, 2016. ISSN 0021-9991. doi: 10.1016/j.jcp.2015.12.032.

Samuel Lanthaler, Zongyi Li, and Andrew M. Stuart. Nonlocality and nonlinearity implies universality in operator learning. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2304.13221.

Jonas Latz, Iason Papaioannou, and Elisabeth Ullmann. Multilevel sequential2 Monte Carlo for Bayesian inverse problems. *Journal of Computational Physics*, 368:154–178, 2018. ISSN 0021-9991. doi: 10.1016/j.jcp.2018.04.014.

K.J.H. Law. Proposals which speed up function-space MCMC. *Journal of Computational and Applied Mathematics*, 262:127–138, 2014. ISSN 0377-0427. doi: 10.1016/j.cam.2013.07.026. Selected Papers from NUMDIFF-13.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint, arXiv.2003.03485*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6755–6766. Curran Associates, Inc., 2020b. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/4b21cf96d4cf612f239a6c322b10c8fe-Paper.pdf`.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint*, 2021. doi: 10.48550/arXiv.2010.08895.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, feb 2024. doi: 10.1145/3648506.

Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.

Lu Lu, Pengzhan Jin, Guofei Pang, and George Em Karniadakis. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 2021.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022. ISSN 0045-7825. doi: 10.1016/j.cma.2022.114778.

Dingcheng Luo, Thomas O'Leary-Roseberry, Peng Chen, and Omar Ghattas. Efficient PDE-constrained optimization under high-dimensional uncertainty using derivative-informed neural operators. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2305.20053.

M. B. Lykkegaard, T. J. Dodwell, C. Fox, G. Mingas, and R. Scheichl. Multilevel delayed acceptance MCMC. *SIAM/ASA Journal on Uncertainty Quantification*, 11(1):1–30, 2023. doi: 10.1137/22M1476770.

Andrea Manzoni, Alfio Quarteroni, and Sandro Salsa. *Optimal control of partial differential equations*. Applied Mathematical Sciences. Springer Cham, 2021. doi: 10.1007/978-3-030-77226-0.

Jerrold E Marsden and Thomas JR Hughes. *Mathematical foundations of elasticity*. Courier Corporation, 1994.

James Martin, Lucas C. Wilcox, Carsten Burstedde, and Omar 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. doi: 10.1137/110845598.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 06 1953. ISSN 0021-9606. doi: 10.1063/1.1699114.

David Nualart and Eulalia Nualart. *Introduction to Malliavin Calculus*, volume 9 of *Institute of Mathematical Statistics Textbooks*. Cambridge University Press, 2018. doi: 10.1017/9781139856485.

John Tinsley Oden, Ivo Babuška, and Danial Faghihi. Predictive computational science: Computer predictions in the presence of uncertainty. In *Encyclopedia of Computational Mechanics*, pages 1–26. John Wiley & Sons, Ltd, 2nd edition, 2017. ISBN 9781119176817. doi: 10.1002/9781119176817.ecm2101.

R.W. Ogden. *Non-linear Elastic Deformations*. Dover Civil and Mechanical Engineering. Dover Publications, 1997. ISBN 9780486696485.

Thomas O'Leary-Roseberry. *dino: Derivative-informed neural operator, an efficient framework for high-dimensional parametric derivative learning*, v0.2.0 edition, 2023. URL `https://github.com/tomoleary/dino`.

Thomas O'Leary-Roseberry and Umberto Villa. *hIPPYflow: Dimension reduced surrogate construction for parametric PDE maps in Python*, 2021. URL `https://github.com/hippylib/hippyflow`.

Thomas O'Leary-Roseberry, Peng Chen, Umberto Villa, and Omar Ghattas. Derivative-Informed Neural Operator: An efficient framework for high-dimensional parametric derivative learning. *Journal of Computational Physics*, 496:112555, 2024. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.112555.

Thomas O'Leary-Roseberry, Xiaosong Du, Anirban Chaudhuri, Joaquim R.R.A. Martins, Karen Willcox, and Omar Ghattas. Learning high-dimensional parametric maps via reduced basis adaptive residual networks. *Computer Methods in Applied Mechanics and Engineering*, 402:115730, 2022a. ISSN 0045-7825. doi: 10.1016/j.cma.2022.115730.

Thomas O'Leary-Roseberry, Umberto Villa, Peng Chen, and Omar Ghattas. Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs. *Computer Methods in Applied Mechanics and Engineering*, 388:114199, 2022b. ISSN 0045-7825. doi: 10.1016/j.cma.2021.114199.

Natesh S. Pillai Patrick R. Conrad, Youssef M. Marzouk and Aaron Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016. doi: 10.1080/01621459.2015.1096787.

Benjamin Peherstorfer and Youssef Marzouk. A transport-based multifidelity preconditioner for Markov chain Monte Carlo. *Advances in Computational Mathematics*, 45:2321–2348, 2019.

Noemi Petra, James Martin, Georg Stadler, and Omar Ghattas. A computational framework for infinite-dimensional Bayesian inverse problems, Part II: Stochastic Newton MCMC with application to ice sheet flow inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1525–A1555, 2014. doi: 10.1137/130934805.

F. J. Pinski, G. Simpson, A. M. Stuart, and H. Weber. Algorithms for Kullback–Leibler approximation of probability measures in infinite dimensions. *SIAM Journal on Scientific Computing*, 37(6):A2733–A2757, 2015. doi: 10.1137/14098171X.

R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 11 2006. ISSN 0956-540X. doi: 10.1111/j.1365-246X.2006.02978.x.

Yuan Qiu, Nolan Bridges, and Peng Chen. Derivative-enhanced deep operator network. *arXiv preprint*, 2024. doi: 10.48550/arXiv.2402.19242.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York, New York, 2nd edition, 2004. ISBN 0387212396.

Gareth O. Roberts and Jeffrey S. Rosenthal. General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1(none):20–71, 2004. doi: 10.1214/154957804100000024.

Daniel Rudolf and Björn Sprungk. On a generalization of the preconditioned Crank–Nicolson Metropolis algorithm. *Foundations of Computational Mathematics*, 18: 309–343, 4 2018. ISSN 16153383. doi: 10.1007/s10208-016-9340-x.

Christoph Schwab and Jakob Zech. Deep learning in high dimension: Neural network expression rates for analytic functions in $\boldsymbol{L^2(\mathbb{R}^d, \gamma\_d)}$. *SIAM/ASA Journal on Uncertainty Quantification*, 11(1):199–234, 2023. doi: 10.1137/21M1462738.

Jacob H. Seidman, Georgios Kissas, George J. Pappas, and Paris Perdikaris. Variational autoencoding neural operators. *arXiv preprint*, 2023. doi: 10.48550/arXiv.2302.10351.

A M Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–459, 2010. ISSN 09624929. doi: 10.1017/S0962492910000061.

T. J. Sullivan. *Introduction to Uncertainty Quantification*. Texts in Applied Mathematics. Springer Cham, 2015. doi: 10.1007/978-3-319-23395-6.

Luke Tierney. A note on Metropolis–Hastings kernels for general state spaces. *The Annals of Applied Probability*, 8(1):1–9, 1998. doi: 10.1214/aoap/1027961031.

Umberto Villa, Noemi Petra, and Omar Ghattas. hIPPYlib: An extensible software framework for large-scale inverse problems. *The Journal of Open Source Software*, 3:940, 2018.

Umberto Villa, Noemi Petra, and Omar Ghattas. hIPPYlib: An extensible software framework for large-scale inverse problems governed by PDEs: Part I: Deterministic inversion and linearized Bayesian inference. *ACM Transactions on Mathematical Software*, 47(2), April 2021. ISSN 0098-3500. doi: 10.1145/3428447.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40):eabi8605, 2021. doi: 10.1126/sciadv.abi8605.

P. Whittle. On stationary processes in the plane. *Biometrika*, 41(3/4):434–449, 1954. ISSN 00063444.

Liang Yan and Tao Zhou. An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems. *Communications in Computational Physics*, 28 (5):2180–2205, 2020. doi: 10.4208/cicp.OA-2020-0186.

Olivier Zahm, Paul G. Constantine, Clémentine Prieur, and Youssef M. Marzouk. Gradient-based dimension reduction of multivariate vector-valued functions. *SIAM Journal on Scientific Computing*, 42(1):A534–A558, 2020. doi: 10.1137/18M1221837.

Appendix A. Stochastic Gâteaux differentiability

From Assumption 4 and Definition 2, it is clear that μ-a.e. Gâteaux differentiability implies stochastic Gâteaux differentiability. Here, we provide a case where the reverse cannot be true. The following lemma uses the Cameron–Martin and Feldman–Hájek theorem (Sullivan, 2015, Theorem 2.51) to establish that μ-a.e. Gâteaux differentiability requires a more regular forward operator $\boldsymbol{\mathcal{G}}$ than the μ-a.e. well-definedness specified in Assumption 3.

Lemma 13 *Assume $\boldsymbol{\mathcal{G}}$ is well-defined μ-a.e. and ill-defined on all sets $\mathscr{A} \subset \mathscr{M}$ with $\mu(\mathscr{A}) = 0$. Then, stochastic Gâteaux differentiability does not imply μ-a.e. Gâtueax differentiability. In particular, $\boldsymbol{\mathcal{G}}$ is not Gâteaux differentiable μ-a.e.*

Proof We focus on the term $\boldsymbol{\mathcal{G}}(M+t\widehat{m})$ for $t > 0$ and $M \sim \mu$ in the definition of the Gâteuax ($\widehat{m} \in \mathscr{M}$) and stochastic Gâteuax ($\widehat{m} \in \mathscr{H}_\mu$) derivative. Let $\mathscr{N}_\mu \coloneqq \{\mathscr{A} \subset \mathscr{M} \,|\, \mu(\mathscr{A}) = 0\}$ be the null set of μ and $M + t\widehat{m} \sim \nu(\cdot; t\widehat{m})$. We have two scenarios listed as follows.

(i) $\mathscr{N}_\mu = \mathscr{N}_{\nu(\cdot;t\widehat{m})}$ for all $t > 0$ if and only if $\widehat{m} \in \mathscr{H}_\mu$, i.e., the null sets are shift invariant.

(ii) There exists a set $\mathscr{E}_t \subset \mathscr{M}$ such that $\mu(\mathscr{E}_t) = 0$ and $\nu(\mathscr{E}_t; t\widehat{m}) = 1$ for all $t > 0$ and $\widehat{m} \in \mathscr{M} \setminus \mathscr{H}_\mu$, i.e., the shifted distributions have disjoint probability concentration.

Due to (1), the term $\boldsymbol{\mathcal{G}}(M + t\widehat{m})$ is well-defined a.s. for all $t > 0$ and $\widehat{m} \in \mathscr{H}_\mu$, thus the limiting sequence within the stochastic derivative definition is well-defined. Due to (2), $M + t\widehat{m} \in \mathscr{E}_t$ a.s. for $\widehat{m} \in \mathscr{M} \setminus \mathscr{H}_\mu$, in which case $\boldsymbol{\mathcal{G}}(M + t\widehat{m})$ is ill-defined and the limiting sequence within the Gâtueax derivative definition is ill-defined. ■

Appendix B. The gradient and the Gauss–Newton Hessian of the data misfit

In this section, we show the connection of our definitions of the ppGNH $\mathcal{H}$ in (6) and the ppg $D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m)$ in (5) to the conventional definitions. Assume the Gâteaux derivative $D\boldsymbol{\mathcal{G}}(m)$ exists, then the following relation holds μ-a.e.

$$D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m) = D\boldsymbol{\mathcal{G}}(m)|_{\mathscr{H}_\mu}\,, \quad D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^* = \mathcal{C}_{\mathrm{pr}} D\boldsymbol{\mathcal{G}}(m)^* \boldsymbol{C}_n^{-1}\,. \tag{52}$$

We have the following $\mathscr{M}$-Risez representation of the gradient, i.e., Gâteaux derivative of the data misfit:

$$D\Phi^{\boldsymbol{y}}(m) \coloneqq D\boldsymbol{\mathcal{G}}(m)^* \boldsymbol{C}_n^{-1}(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y})\,.$$

By the definition of the ppg and (52)

$$D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m) \coloneqq \mathcal{C}_{\mathrm{pr}} D\Phi^{\boldsymbol{y}}(m) \implies D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m) = D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^*(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y})\,.$$

We now show that $D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m)$ is indeed the $\mathscr{H}_\mu$-Riesz representation of the data misfit stochastic derivative. Suppose the stochastic derivative of the data misfit is given by $\mathcal{T} \in \mathscr{H}_\mu'$, where $\mathscr{H}_\mu'$ is the dual space of $\mathscr{H}_\mu$. By the chain rule, for all $\delta m \in \mathscr{H}_\mu$ we have

$$\begin{aligned}
\mathcal{T}(m)\delta m &= \left(D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\delta m\right)^T \boldsymbol{C}_n^{-1}\left(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}\right) && \text{(Chain rule)} \\
&= \left\langle D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^*(\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}), \delta m\right\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}} && \text{(Definition of adjoint on HS}(\mathscr{H}_\mu, \mathscr{Y})\text{)} \\
&= \left\langle D_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m), \delta m\right\rangle_{\mathcal{C}_{\mathrm{pr}}^{-1}}\,.
\end{aligned}$$

Therefore, our definition of the ppg is identical to the conventional definition. Similarly, we have the following definition of the Gauss–Newton Hessian using the Gâteaux derivative:

$$\mathcal{H}_{\mathrm{GN}}(m) \coloneqq D\boldsymbol{\mathcal{G}}(m)\boldsymbol{C}_n^{-1} D\boldsymbol{\mathcal{G}}(m)\,. \qquad \text{(Gauss–Newton Hessian)}$$

By the definition of the ppGNH and (52), we have

$$\mathcal{H}(m) \coloneqq \mathcal{C}_{\text{pr}}\mathcal{H}_{\text{GN}}(m) \implies \mathcal{H}(m) = D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)^* D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\,.$$

To see that $\mathcal{H}(m)$ is the Gauss–Newton Hessian under the stochastic derivative assumption, notice that the stochastic Hessian of the data misfit $D^2_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m) \in \text{HS}(\mathscr{H}_\mu)$ is given by

$$D^2_{\mathscr{H}_\mu}\Phi^{\boldsymbol{y}}(m)\delta m \coloneqq \left(D^2_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\delta m\right)^* (\boldsymbol{\mathcal{G}}(m) - \boldsymbol{y}) + \mathcal{H}(m)\delta m \quad \forall \delta m \in \mathscr{H}_\mu\,, \tag{53}$$

where $D^2_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m) \in \text{HS}(\mathscr{H}_\mu, \text{HS}(\mathscr{H}_\mu, \mathscr{Y}))$ is the stochastic Hessian of the forward operator. Assuming the data misfit term is relatively small in regions with high posterior probability, one may drop the term involving the Hessian of the PtO map and still retain a reasonable approximation to the data misfit Hessian. This makes the ppGNH $\mathcal{H}$ an approximation to the stochastic Hessian of the data misfit.

Appendix C. Delayed acceptance and neural operator approximation error

Recall from Section 2.8 that the proposal acceptance rate in the second stage of the DA MCMC reflects the quality of the generated Markov chain, and the second stage acceptance probability is closely related to the error in surrogate data misfit evaluation, denoted by $\mathcal{E}_{\text{misfit}} : \mathscr{M} \to \mathbb{R}$:

$$\ln \rho^{(2)}(m_j, m^\dagger) = \mathcal{E}_{\text{misfit}}(m_j) - \mathcal{E}_{\text{misfit}}(m^\dagger)\,, \quad \mathcal{E}_{\text{misfit}}(m) \coloneqq \Phi^{\boldsymbol{y}}(m) - \widetilde{\Phi^{\boldsymbol{y}}}(m) \quad \mu\text{-a.e..}$$

where $\rho^{(2)}$ is the transition rate ratio of DA MCMC defined in (14). The arguments of $\rho^{(2)}$, namely m_j and $m^\dagger$, are coupled through the proposal and the first stage of DA MCMC, thus analyzing $\rho^{(2)}$ is not straightforward and is not the focus of this work. However, the error analysis for surrogate data misfit evaluation provides insights into the behavior of $\rho^{(2)}$ and the efficiency of DA MCMC related to the operator surrogate approximation error. In particular, the $L^1_{\mu^{\boldsymbol{y}}}(\mathscr{M})$ approximation error of the surrogate data misfit (averaged over the true posterior) is controlled by the surrogate approximation error (averaged over the prior):

$$\begin{aligned} \|\mathcal{E}_{\text{misfit}}\|_{L^1_{\mu^{\boldsymbol{y}}}(\mathscr{M})} &\coloneqq \mathbb{E}_{M\sim\mu^{\boldsymbol{y}}}\left|\Phi^{\boldsymbol{y}}(M) - \widetilde{\Phi^{\boldsymbol{y}}}(M)\right| \\ &\leq c_{\text{misfit}}(\boldsymbol{\mathcal{G}}, \widetilde{\boldsymbol{\mathcal{G}}}, \boldsymbol{y}, \boldsymbol{C}_n^{-1}\mu)\left\|\boldsymbol{\mathcal{G}} - \widetilde{\boldsymbol{\mathcal{G}}}\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})}\,, \end{aligned} \tag{54}$$

where $c_{\text{misfit}} > 0$ is a constant given by

$$c_{\text{misfit}}(\boldsymbol{\mathcal{G}}, \widetilde{\boldsymbol{\mathcal{G}}}, \boldsymbol{y}, \boldsymbol{C}_n^{-1}, \mu) = \left\|\exp(-\widetilde{\Phi^{\boldsymbol{y}}}(\cdot))\right\|_{L^\infty_\mu(\mathscr{M})}\left\|\frac{1}{2}\left(\boldsymbol{\mathcal{G}}(\cdot) + \widetilde{\boldsymbol{\mathcal{G}}}(\cdot)\right) - \boldsymbol{y}\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})}\,.$$

See proof by Cao et al. (2023, Theorem 1).

Appendix D. Proofs of Propositions 7 and 8

Proof [Proposition 7] We decompose the operator surrogate approximation error into two parts using a triangle inequality:

$$\left\|\boldsymbol{\mathcal{G}}-\widetilde{\boldsymbol{\mathcal{G}}}\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} \leq \|\boldsymbol{\mathcal{G}}-\boldsymbol{\mathcal{G}}_r\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} + \left\|\boldsymbol{\mathcal{G}}_r-\widetilde{\boldsymbol{\mathcal{G}}}\right\|_{L^2_\mu(\mathscr{M};\mathscr{Y})} .$$

We examine the second term on the right-hand side of the inequality. First, we have $\Psi_r^* M \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_r)$ for any linear encoder Ψ_r^* defined as in (20); see (57). Second, notice that $\boldsymbol{\mathcal{G}}_r(m) \equiv \boldsymbol{\mathcal{G}}_r(\widehat{\Psi_r^{\mathrm{DIS}}}\widehat{\Psi_r^{\mathrm{DIS}}}^* m)$. Consequently, the neural network error is given by

$$\begin{aligned}\left\|\widetilde{\boldsymbol{\mathcal{G}}}-\boldsymbol{\mathcal{G}}_r\right\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} &= \mathbb{E}_{\boldsymbol{M}_r\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_r)}\left[\left\|\boldsymbol{V}\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{M}_r)-\boldsymbol{\mathcal{G}}_r\left(\widehat{\Psi_r^{\mathrm{DIS}}}\boldsymbol{M}_r\right)\right\|^2_{\boldsymbol{C}_n^{-1}}\right]\\ &= \mathbb{E}_{\boldsymbol{M}_r\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_r)}\left[\left\|\boldsymbol{f}_{\mathrm{NN}}(\boldsymbol{M}_r)-\boldsymbol{V}^*\boldsymbol{\mathcal{G}}_r\left(\widehat{\Psi_r^{\mathrm{DIS}}}\boldsymbol{M}_r\right)\right\|^2\right] .\end{aligned}$$

We further decompose the first term on the right-hand side of the triangle inequality.

Part I: Subspace Poincaré inequality. We follow Zahm et al. 2020, Propositions 2.4. Due to the Poincaré inequality for H^1_μ (Theorem 4 and Bogachev 1998, 5.5.6), for any $\mathcal{S} \in H^1_\mu(\mathscr{M}) \coloneqq H^1_\mu(\mathscr{M};\mathbb{R})$ and any pair of encoder Ψ_r^* and decoder Ψ_r as defined in (20), we have:

$$\|\mathcal{S}-\mathcal{S}_r\|^2_{L^2_\mu(\mathscr{M})} \leq \left\|(\mathcal{I}_{\mathscr{H}_\mu}-\Psi_r\Psi_r^*)D_{\mathscr{H}}\mathcal{S}\right\|^2_{L^2_\mu(\mathscr{M};\mathscr{H}_\mu)} ,$$

where $\mathcal{S}_r$ is the $L^2_\mu(\mathscr{M})$-optimal reduced mapping of $\mathcal{S}$ for the given encoder and decoder, $D_{\mathscr{H}_\mu}\mathcal{S} \in L^2_\mu(\mathscr{M};\mathscr{H}_\mu)$ is the $\mathscr{H}_\mu$-representation of the stochastic derivative of $\mathcal{S}$. The key to extend the results by Zahm et al. 2020, Propositions 2.4 is to show that the mapping for any $m' \in \mathscr{M}$

$$f : m \mapsto \mathcal{S}\left(\Psi_r\Psi_r^* m' + (\mathcal{I}_{\mathscr{M}}-\Psi_r\Psi_r^*)m\right)$$

has a stochastic derivative of the following form via the chain rule:

$$D_{\mathscr{H}_\mu} f(m) = (\mathcal{I}_{\mathscr{H}_\mu}-\Psi_r\Psi_r^*)D_{\mathscr{H}_\mu}\mathcal{S}(\Psi_r\Psi_r^* m' + (\mathcal{I}_{\mathscr{M}}-\Psi_r\Psi_r^*)m) .$$

The $H^1_\mu(\mathscr{M})$-Poincaré inequality is applied to f, which leads to the subspace Poincaré inequality

Part II: Error upper bound. We follow Zahm et al. 2020, Proposition 2.5 to our setting. Due to the subspace Poincaré inequality, for any pair of encoder Ψ_r^* and decoder Ψ_r as defined by (20), we have

$$\|\boldsymbol{\mathcal{G}}-\boldsymbol{\mathcal{G}}_r\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} \leq \mathrm{Tr}_{\mathscr{H}_\mu}(\mathcal{H}_A) - \mathrm{Tr}\left(\Psi_r^*\mathcal{H}_A\Psi_r\right) .$$

where $\boldsymbol{\mathcal{G}}_r$ is the $L^2_\mu(\mathscr{M};\mathscr{Y})$-optimal reduced mapping for the given Ψ_r^* and Ψ_r. The key to extend the results by Zahm et al. 2020, Proposition 2.5 is to define $\mathcal{S}^{(j)} \coloneqq \boldsymbol{v}_j^T\boldsymbol{C}_n^{-1}\boldsymbol{\mathcal{G}}$ where $\{\boldsymbol{v}_j\}_{j=1}^{d_y}$ is a $\mathscr{Y}$-ONB.

$$\|\boldsymbol{\mathcal{G}}-\boldsymbol{\mathcal{G}}_r\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} = \sum_{j=1}^{d_y}\left\|\mathcal{S}^{(j)}-\mathcal{S}_r^{(j)}\right\|^2_{L^2_\mu(\mathscr{M})} .$$

Applying the subspace inequality to $\mathcal{S}^j$ and the transformation between trace and HS norm (Gohberg et al., 2012, Theorem 7.3), we have

$$\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_r\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} \leq \sum_{j=1}^{d_y} \mathrm{Tr}_{\mathscr{H}_\mu}\Big((\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r\Psi_r^*) \tag{55}$$

$$\mathbb{E}_{M\sim\mu}\left[D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}^*\boldsymbol{v}_j\boldsymbol{v}_j^T\boldsymbol{C}_n^{-1}D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}\right](\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r\Psi_r^*)\Big)$$

$$= \mathrm{Tr}_{\mathscr{H}_\mu}\left((\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r\Psi_r^*)\mathcal{H}_A(\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r\Psi_r^*)\right). \tag{56}$$

Part III: Sampling error. We follow a line of arguments presented in Lam et al. (2020). Let $\{\lambda_j^{\mathrm{DIS}} \in \mathbb{R}_+, \psi_j^{\mathrm{DIS}} \in \mathscr{H}_\mu\}_{j=1}^\infty$ and $\{\widehat{\lambda_j^{\mathrm{DIS}}} \in \mathbb{R}_+, \widehat{\psi_j^{\mathrm{DIS}}} \in \mathscr{H}_\mu\}_{j=1}^\infty$ denote the eigendeomposition of $\mathcal{H}_A \coloneqq \mathbb{E}_{M\sim\mu}[\mathcal{H}(m)]$ and $\widehat{\mathcal{H}}$ with decreasing eigenvalues and $\mathscr{H}_\mu$-orthonormal basis. Let $\Psi_r^{\mathrm{DIS}}, \widehat{\Psi_r^{\mathrm{DIS}}} \in \mathrm{HS}(\mathbb{R}^r, \mathscr{H}_\mu)$ be the linear decoder defined using the first r eigenbases. Then, we can deduce the optimal low-rank approximation of $\mathcal{H}_A$ and $\widehat{\mathcal{H}}$ using the Courant min–max principle:

$$\Psi_r^{\mathrm{DIS}} \in \underset{\substack{\mathcal{U}_r\in\mathrm{HS}(\mathbb{R}^r,\mathscr{H}_\mu)\\ \mathcal{U}_r^*\mathcal{U}_r=\boldsymbol{I}_r}}{\arg\max}\ \mathrm{Tr}\left(\mathcal{U}_r^*\mathcal{H}_A\mathcal{U}_r\right), \quad \widehat{\Psi_r^{\mathrm{DIS}}} \in \underset{\substack{\mathcal{U}_r\in\mathrm{HS}(\mathbb{R}^r,\mathscr{H}_\mu)\\ \mathcal{U}_r^*\mathcal{U}_r=\boldsymbol{I}_r}}{\arg\max}\ \mathrm{Tr}\left(\mathcal{U}_r^*\widehat{\mathcal{H}}\mathcal{U}_r\right).$$

Assume $\widehat{\mathcal{H}} - \mathcal{H}$ can be decomposed to $\mathcal{V}\mathcal{D}\mathcal{V}^*$, where $\mathcal{V} \in \mathrm{HS}(l^2, \mathscr{H}_\mu)$ has columns consisting of $\mathscr{H}_\mu$-orthonormal eigenbases and $\mathcal{D} \in l^2$ consists of eigenvalues, the cyclic property of trace leads to

$$\mathrm{Tr}(\mathcal{U}_r^*(\widehat{\mathcal{H}} - \mathcal{H}_A)\mathcal{U}_r) = \mathrm{Tr}(\mathcal{U}_r^*\mathcal{V}\mathcal{D}\mathcal{V}^*\mathcal{U}_r)$$

$$\leq \left\|\mathcal{H} - \widehat{\mathcal{H}}\right\|_{B(\mathscr{H}_\mu)} \mathrm{Tr}_{\mathscr{H}_\mu}(\mathcal{V}^*\mathcal{U}_r\mathcal{U}_r^*\mathcal{V}) = r\left\|\mathcal{H}_A - \widehat{\mathcal{H}}\right\|_{B(\mathscr{H}_\mu)},$$

where $\mathcal{U}_r \in \mathrm{HS}(\mathbb{R}^r, \mathscr{H}_\mu)$ is arbitrary and has columns consisting of $\mathscr{H}_\mu$-orthonormal reduced bases. Applying the inequality above twice, we have the following upper bound of the approximation error to the optimal reduced mapping given the pair of decoder and encoder $\widehat{\Psi_r^{\mathrm{DIS}}}$ and $\widehat{\Psi_r^{\mathrm{DIS}}}^*$:

$$\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_r\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} \leq \mathrm{Tr}_{\mathscr{H}_\mu}(\mathcal{H}_A) - \mathrm{Tr}\left(\widehat{\Psi_r^{\mathrm{DIS}}}^*\mathcal{H}_A\widehat{\Psi_r^{\mathrm{DIS}}}^*\right)$$

$$\leq \mathrm{Tr}_{\mathscr{H}_\mu}(\mathcal{H}_A) - \mathrm{Tr}\left(\Psi_r^{\mathrm{DIS}}\mathcal{H}_A(\Psi_r^{\mathrm{DIS}})^*\right) + 2r\left\|\mathcal{H}_A - \widehat{\mathcal{H}}\right\|_{B(\mathscr{H}_\mu)}$$

$$= \sum_{j=r+1}^{\infty}\lambda_j^{\mathrm{DIS}} + 2r\left\|\mathcal{H}_A - \widehat{\mathcal{H}}\right\|_{B(\mathscr{H}_\mu)}.$$

■

Proof [Proposition 8] We follow the same arguments by Zahm et al. 2020, Porposition 3.1. From (56) and the definition of KLE, the approximation error for the optimal reduced

mapping defined by the KLE reduced bases $\Psi_r^{\mathrm{KLE}} \in \mathrm{HS}(\mathbb{R}^r, \mathscr{H}_\mu)$ is given by

$$\begin{aligned}\|\boldsymbol{\mathcal{G}} - \boldsymbol{\mathcal{G}}_r\|^2_{L^2_\mu(\mathscr{M};\mathscr{Y})} &\leq \mathrm{Tr}_{\mathscr{H}_\mu}\left((\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r^{\mathrm{KLE}}\Psi_r^{\mathrm{KLE}^*})\mathcal{H}_A(\mathcal{I}_{\mathscr{H}_\mu} - \Psi_r^{\mathrm{KLE}}\Psi_r^{\mathrm{KLE}^*})\right)\\ &\leq \|\mathcal{H}_A\|_{B(\mathscr{H}_\mu)}\,\mathbb{E}_{M\sim\mu}\left[\left\|\left(\mathcal{I}_{\mathscr{M}} - \Psi_r^{\mathrm{KLE}}\Psi_r^{\mathrm{KLE}^*}\right)M\right\|^2_{\mathscr{M}}\right]\\ &= \|\mathcal{H}_A\|_{B(\mathscr{H}_\mu)}\sum_{j=r+1}^{\infty}\left(\lambda_j^{\mathrm{KLE}}\right)^2 .\end{aligned}$$

By Assumption 4, we have the following bound μ-a.e.

$$\left\|D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(m)\right\|_{B(\mathscr{H},\mathscr{Y})} \leq \sup_{\substack{\delta m\in\mathscr{H}_\mu\\ \|\delta m\|_{\mathscr{H}_\mu}=1}} \lim_{t\to 0}\left\|t^{-1}\left(\boldsymbol{\mathcal{G}}(m+t\delta m) - \boldsymbol{\mathcal{G}}(m)\right)\right\|_{\boldsymbol{C}_n^{-1}} \leq c_{\boldsymbol{\mathcal{G}}}\,.$$

Thus we have

$$\|\mathcal{H}_A\|_{B(\mathscr{H}_\mu)} = \sup_{\substack{\delta m\in\mathscr{H}_\mu\\ \|\delta m\|_{\mathscr{H}_\mu}=1}} \mathbb{E}_{M\sim\mu}\left[\left\|D_{\mathscr{H}_\mu}\boldsymbol{\mathcal{G}}(M)\delta m\right\|^2_{\boldsymbol{C}_n^{-1}}\right] \leq c^2_{\boldsymbol{\mathcal{G}}}\,.$$

Due to (56), the minimized upper bound is achieved when $\Psi_r = \Psi_r^{\mathrm{DIS}}$. Therefore, we have

$$\sum_{j=r+1}^{\infty}\lambda_j^{\mathrm{DIS}} \leq \|\mathcal{H}_A\|_{B(\mathscr{H}_\mu)}\sum_{j=r+1}^{\infty}\left(\lambda_j^{\mathrm{KLE}}\right)^2 \leq c^2_{\boldsymbol{\mathcal{G}}}\sum_{j=r+1}^{\infty}\left(\lambda_j^{\mathrm{KLE}}\right)^2 .$$

■

Appendix E. Proofs of Lemma 10 and Proposition 11

Proof [Lemma 10] The statement on transforming Gaussian random elements is standard; see, e.g., Da Prato and Zabczyk 2002, Proposition 1.2.3. Since the $\mathscr{M}$-adjoint of Ψ_r^* is $\mathcal{C}_{\mathrm{pr}}^{-1}\Psi_r$, we have

$$\Psi^* M \sim \mathcal{N}(0, \Psi^*\mathcal{C}_{\mathrm{pr}}\mathcal{C}_{\mathrm{pr}}^{-1}\Psi_r) = \mathcal{N}(0, \boldsymbol{I}_r)\,. \tag{57}$$

We focus on the statement on the independence of two random elements $M_r \perp M_\perp$ given by:

$$\begin{cases} M_r = \Psi_r\Psi_r^* M \\ M_\perp \coloneqq (\mathcal{I}_{\mathscr{M}} - \Psi_r\Psi_r^*)M \end{cases}, \quad M\sim\mu\,.$$

We examine the characteristic function for the joint random element $X = (M_r, M_\perp)$ and equip the product space $\mathscr{M}\times\mathscr{M}$ with an extended inner product:

$$\langle (t_r, t_\perp), (s_r, s_\perp)\rangle_{\mathscr{M}\times\mathscr{M}} = \langle t_r, s_r\rangle_{\mathscr{M}} + \langle t_\perp, s_\perp\rangle_{\mathscr{M}}\,.$$

We have the following form for the characteristic function of X:

$$\begin{aligned}
&\mathbb{E}_{M\sim\mu}\left[\exp\left(i\left\langle \Psi_r\Psi_r^* M, t_r\right\rangle_{\mathscr{M}}\right)\exp\left(i\left\langle \left(\mathcal{I}_{\mathscr{M}}-\Psi_r\Psi_r^*\right)M, t_\perp\right\rangle_{\mathscr{M}}\right)\right] \\
&= \mathbb{E}_{M\sim\mu}\left[\exp\left(i\left\langle M, (\Psi_r\Psi_r^*)^*(t_r-t_\perp)+t_\perp\right\rangle_{\mathscr{M}}\right)\right] && \text{(Def. of } \mathscr{M}\text{-adjoint)} \\
&= \exp\left(\left\langle \mathcal{C}_{\mathrm{pr}}\left((\Psi_r\Psi_r^*)^*(t_r-t_\perp)+t_\perp\right), (\Psi_r\Psi_r^*)^*(t_r-t_\perp)+t_\perp\right\rangle_{\mathscr{M}}\right) && \text{(Def. of charac. func.)} \\
&= \exp\left(\left\langle \Psi_r\Psi_r^*\mathcal{C}_{\mathrm{pr}}(t_r-t_\perp)+\mathcal{C}_{\mathrm{pr}}t_\perp, (\Psi_r\Psi_r^*)^*(t_r-t_\perp)+t_\perp\right\rangle_{\mathscr{M}}\right) && \text{(Explicit } \mathscr{M}\text{-adjoint)} \\
&= \exp\left(\left\langle \Psi_r\Psi_r^*\mathcal{C}_{\mathrm{pr}}t_r, t_r\right\rangle_{\mathscr{M}}\right)\exp\left(\left\langle \left(\mathcal{I}_{\mathscr{M}}-\Psi_r\Psi_r^*\right)\mathcal{C}_{\mathrm{pr}}t_\perp, t_\perp\right\rangle_{\mathscr{M}}\right) && \text{(Cancel cross terms)}
\end{aligned}$$

Therefore, by the definition of the characteristic function for Gaussian measures, we have

$$X\sim\mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\Psi_r\Psi_r^*\mathcal{C}_{\mathrm{pr}} & 0\\ 0 & \left(\mathcal{I}_{\mathscr{M}}-\Psi_r\Psi_r^*\right)\mathcal{C}_{\mathrm{pr}}\end{bmatrix}\right),$$

and thus M_r and $M_\perp$ are independently distributed. ■

Proof [Proposition 11] We show that the surrogate mMALA proposal $\widetilde{\mathcal{Q}_{\mathrm{mMALA}}}(m,\cdot)$ at a given m can be defined through a deterministic transformation of the prior following Lemma 10. Consider $\widetilde{\mathcal{T}}(m)\in B(\mathscr{M})$ given by

$$\widetilde{\mathcal{T}}(m)=\mathcal{I}_{\mathscr{M}}+\widetilde{\Psi_r}\left(\left((\widetilde{d}_j+1)^{1/2}-1\right)\delta_{jk}\right)\widetilde{\Psi_r}^*,$$

where $\widetilde{\Psi_r}$, $\widetilde{\Psi_r}^*$, and $\widetilde{d}_j$ are defined in Section 5.2. The covariance of the local Gaussian approximation of the posterior in (34) can be expressed as

$$\widetilde{\mathcal{C}_{\mathrm{post}}}(m)=\widetilde{\mathcal{T}}(m)\mathcal{C}_{\mathrm{pr}}\widetilde{\mathcal{T}}(m)^*. \tag{58}$$

The key to validating (58) is to take the adjoint of $\widetilde{\Psi_r}$ and $\widetilde{\Psi_r}^*$ in $\mathscr{M}$ when taking the adjoint of $\mathcal{T}(m)$. In particular, the $\mathscr{M}$-adjoint of $\widetilde{\mathcal{T}}(m)$ is given by

$$\widetilde{\mathcal{T}}(m)^*=\mathcal{I}_{\mathscr{M}}+\mathcal{C}_{\mathrm{pr}}^{-1}\widetilde{\Psi_r}\left(\left((\widetilde{d}_j+1)^{1/2}-1\right)\delta_{jk}\right)\widetilde{\Psi_r}^*\mathcal{C}_{\mathrm{pr}}.$$

The covariance transformation given by $\widetilde{\mathcal{T}}(m)$ leads to

$$M^\dagger\sim\widetilde{\mathcal{Q}_{\mathrm{mMALA}}}(m,\cdot)\quad\text{and}\quad M^\dagger=sm+(1-s)\widetilde{\mathcal{A}}(m)+\sqrt{1-s^2}\widetilde{\mathcal{T}}(m)M,\quad M\sim\mu.$$

Since the M can be independently decomposed into two parts using the encoder Ψ_r^* and decoder Ψ_r due to Lemma 10, the proposal distribution can also be decomposed into two parts independently. ■

Appendix F. Diagnostics, tuning and initialization

F.1 On the MPSRF diagnostic

The MPSRF is typically defined as follows:

$$\widehat{R}=\sqrt{\max_{\substack{m\in\mathscr{M}\text{ s.t.}\\ \|m\|_{\mathscr{M}}=1}}\frac{\left\langle m,\widehat{\mathcal{W}}_s m\right\rangle_{\mathscr{M}}}{\left\langle m,\widehat{\mathcal{V}}_s m\right\rangle_{\mathscr{M}}}}.\qquad\text{(Conventional MPSRF)}$$

where $\widehat{\mathcal{W}}_s$ and $\widehat{\mathcal{V}}_s$ are defined in (40). However, this quantity is not well-defined since the two empirical covariance operators are singular for a finite sample size n_s. During computation on a discretized parameter space, a pool of long MCMC chains is often needed to estimate this quantity. Moreover, the MPSRF characterizes Markov chains along a single slice of the parameter space, which is sufficient for monitoring convergence but insufficient for comparing the quality of Markov chains generated by different MCMC methods.

F.2 On tuning and initialization

We provide the procedure for determining the step size parameter $\triangle t$ for a given Bayesian inverse problem. The procedure is consistent across all MCMC algorithms in Tables 2 and 3. It is designed to maximize the sampling performance of MCMC methods while maintaining uniform behaviors of the MCMC chain across different regions of the parameter space.

First, we choose a list of candidate values for $\triangle t$. Then, we generate an MCMC chain $\{m_j\}_{j=1}^{n_s}$ with $n_s = 5000$ samples (after burn-in) for each candidate value. Using these samples, we compute the acceptance rate (AR) and the mean square jump (MSJ) given by

$$\mathrm{AR} \coloneqq \frac{n_{\mathrm{accept}}}{n_s} \times 100\%\,, \quad \mathrm{MSJ} \coloneqq \frac{1}{n_s - 1} \sum_{j=1}^{n_s - 1} \|m_{j+1} - m_j\|_{\mathscr{M}}^2\,,$$

where n_{accept} is the number of accepted proposal samples in the MCMC chain. We down-select the candidate values by choosing a maximum step size $\triangle t_{\max} > 0$ such that AR monotonically decreases and MSJ monotonically increases as a function of $\triangle t \in [0, \triangle t_{\max}]$. Finally, we choose a tuned step size value from the remaining candidates that maximizes the median of the single chain version of the ESS% in (41).

To make the step size tuning more efficient, we initialize the chains with samples from the Laplace approximation to reduce the number of burn-in samples. Once the step size tuning is complete, we initialize the subsequent MCMC runs using samples obtained from step size tuning.

Remark 14 *Due to the high non-linearity of Bayesian inversion in our numerical examples, the step size tuning procedure introduced above is often ineffective for MALA and LA-pCN. In particular, the ARs for MCMC chains initialized at different positions have significant variations when the tuned step size is employed. In such cases, the MCMC chains are often stuck in local regions with low ARs; see Figure 18. We reduce the step size to eliminate this behavior until the AR variations are within* $\pm 5\%$ *of the averaged value.*

Appendix G. Supplementary materials for the numerical examples

In Figure 19, we visualize selected DIS and KLE basis functions for coefficient inversion in a nonlinear diffusion–reaction PDE. In Table 6, we list the statistics for the MCMC runs. In Figure 20, we visualize the posterior samples, mean estimate via MCMC using mMALA, the absolute error between MAP estimate and mean estimate, pointwise variance estimate via MCMC using mMALA, and the absolute error of pointwise variance estimate from LA and MCMC. The same visualization and statistics for hyperelastic material deformation are provided in Figures 21 and 22 and Table 7. We emphasize that the large error between the MCMC mean and MAP estimates indicates a non-Gaussian posterior distribution.

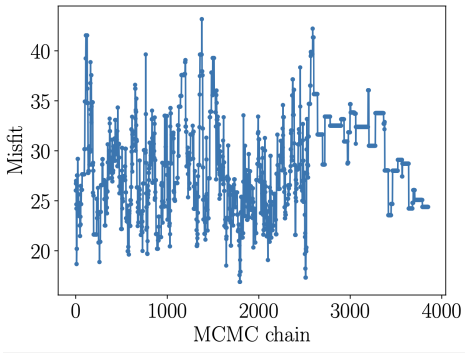

Figure 18: The trace plot of data misfit values along a Markov chain generated by LA-pCN at large step size for coefficient inversion in a nonlinear diffusion–reaction PDE. The Markov chain behaves drastically differently in different parts of the chain, leading to a large bias in posterior estimation.

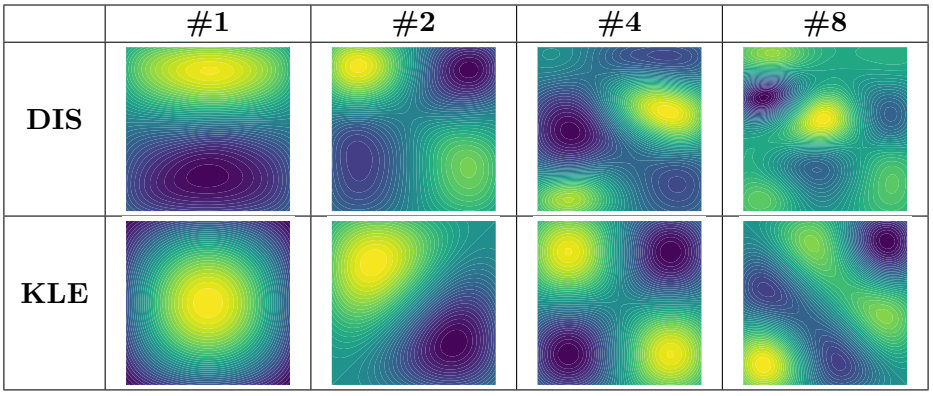

Figure 19: Visualization of selected DIS and KLE basis functions for coefficient inversion in a nonlinear diffusion–reaction PDE in Section 7

Name	**Step size** $\triangle t$ $\times 10^{-2}$	**Acceptance rate** %	**Mean square jump** $\times 10^{-3}$
pCN	1	16	1
MALA	0.36	39	0.96
LA-pCN	2.3	60	6.0
DIS-mMALA	11	44	11
mMALA	15	20	12
NO-mMALA $n_t = 16 \times 10^3$	1.8	26	2.1
DINO-mMALA $n_t = (2, 4, 8, 16) \times 10^3$	$(12, 12, 13, 15)$	$(19, 20, 20, 17)$	$(9.9, 10, 11, 11)$
DA-NO-mMALA $n_t = (1, 2, 4, 8, 16) \times 10^3$	$(1, 1, 1.3, 1.5, 1.5, 1.8)$	1st: $(63, 67, 67, 70, 68)$ 2nd: $(11, 23, 28, 27, 35)$	$(0.23, 0.45, 0.61, 0.87, 1.1)$
DA-DINO-mMALA $n_t = (1, 2, 4, 8, 16) \times 10^3$	$(11, 11, 12, 13, 14)$	1st: $(40, 29, 25, 22, 19)$ 2nd: $(42, 57, 67, 72, 76)$	$(4.6, 6.8, 7.9, 8.7, 9.2)$

Table 6: The statistics of MCMC runs for coefficient inversion in a nonlinear diffusion–reaction PDE in Section 7.

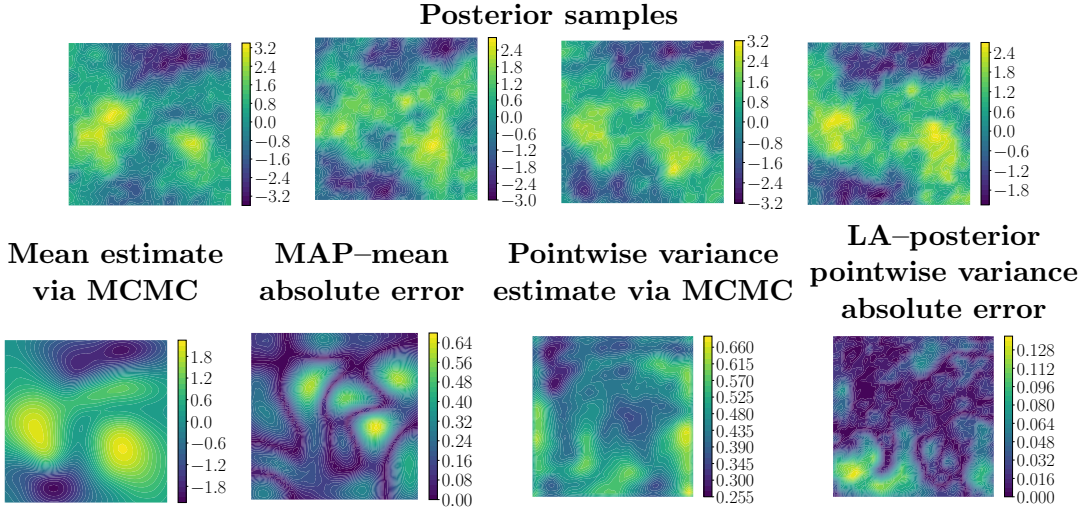

Figure 20: Visualization of relevant statistics from Markov chains collected using mMALA for coefficient inversion in a nonlinear diffusion–reaction PDE in Section 7.

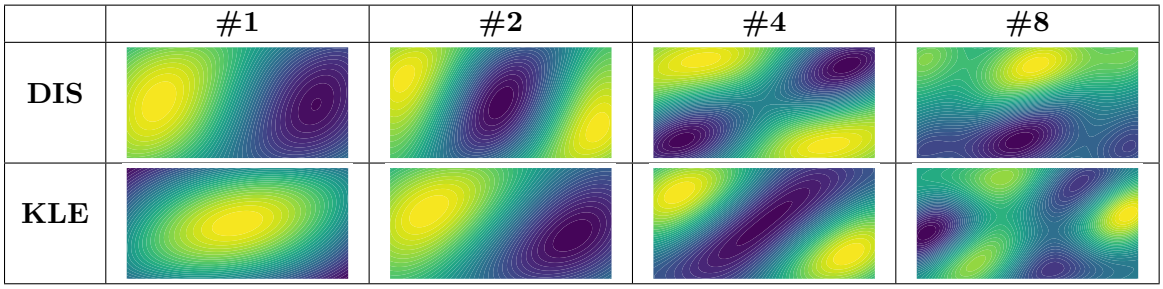

Figure 21: Visualization of selected DIS and KLE basis functions for inference of a heterogeneous hyperelastic material property in Section 8

Name	**Step size** $\triangle t$ $\times 10^{-2}$	**Acceptance rate** %	**Mean square jump** $\times 10^{-3}$
pCN	1.0	30	0.43
MALA	0.075	78	0.40
LA-pCN	5.0	75	2.6
DIS-mMALA	150	64	26
mMALA	100	34	14
DA-NO-mMALA $n_t = (5, 10, 20, 40) \times 10^2$	$(0.5, 1, 2, 4)$	1st: $(49, 37, 33, 44)$ 2nd: $(41, 41, 46, 68)$	$(0.12, 0.16, 0.27, 0.90)$
DA-DINO-mMALA $n_t = (5, 10, 20, 40) \times 10^2$	$(65, 65, 70, 70)$	1st: $(38, 40, 38, 39)$ 2nd: $(76, 87, 90, 93)$	$(9.4, 11, 11, 12)$

Table 7: The statistics of MCMC runs for inference of a heterogeneous hyperelastic material property in Section 8.

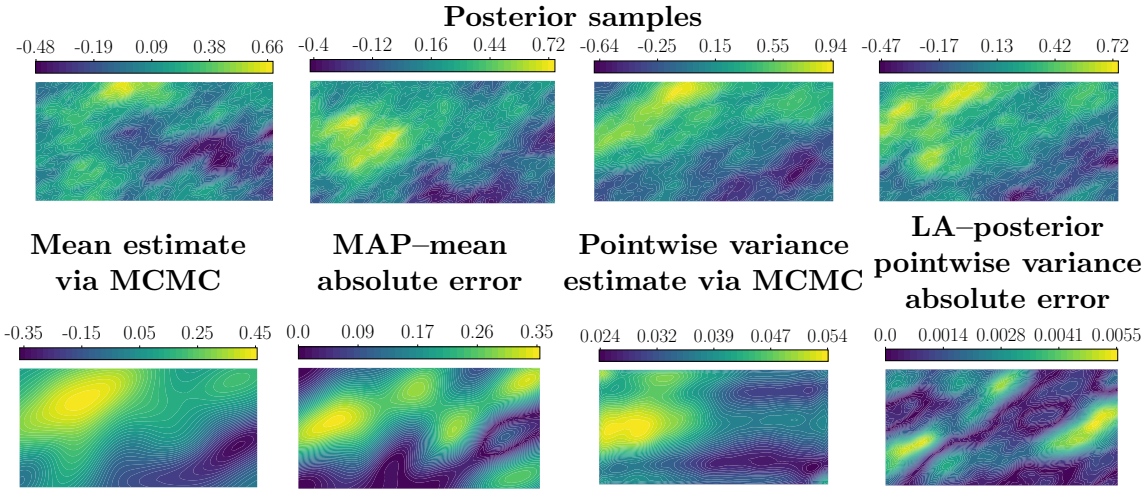

Figure 22: Visualization of relevant statistics from Markov chains collected using mMALA for inference of a heterogeneous hyperelastic material property in Section 8.