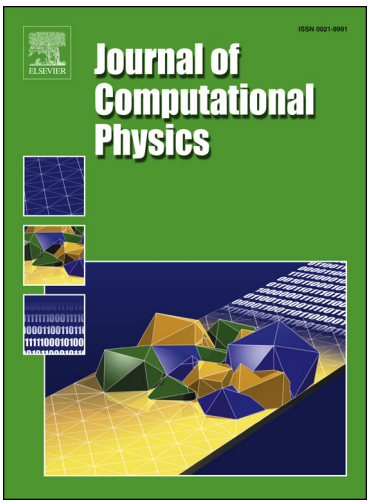
Journal Pre-proof



Residual-Based Error Correction for Neural Operator Accelerated Infinite-Dimensional Bayesian Inverse Problems

Lianghao Cao, Thomas O'Leary-Roseberry, Prashant K. Jha, J. Tinsley Oden and Omar Ghattas

PII: S0021-9991(23)00199-7
DOI: https://doi.org/10.1016/j.jcp.2023.112104
Reference: YJCPH 112104

To appear in: *Journal of Computational Physics*

Received date: 18 October 2022
Revised date: 22 March 2023
Accepted date: 26 March 2023

Please cite this article as: L. Cao, T. O'Leary-Roseberry, P.K. Jha et al., Residual-Based Error Correction for Neural Operator Accelerated Infinite-Dimensional Bayesian Inverse Problems, *Journal of Computational Physics*, 112104, doi: https://doi.org/10.1016/j.jcp.2023.112104.

Highlights

- We deploy neural operators with error correction as parameter-to-state surrogates in infinite-dimensional Bayesian inference problems.
- The neural operator prediction is corrected by solving a linear variational problem defined by the PDE residual.
- Correction steps can lead to a quadratic reduction of the approximation error of neural operators.
- An *a priori* bound is derived, showing that the approximation error of neural operators controls the error in posterior distributions.
- Numerical examples, including the deformation of hyperelastic materials, show reliably improved inference results from neural operators with error correction.

Residual-Based Error Correction for Neural Operator Accelerated Infinite-Dimensional Bayesian Inverse Problems

Lianghao Cao*, Thomas O'Leary-Roseberry, Prashant K. Jha, J. Tinsley Oden, Omar Ghattas

Oden Institute for Computational Sciences and Engineering, The University of Texas at Austin, 201 E. 24th Street, C0200, Austin, TX 78712, United States of America.

Abstract

We explore using neural operators, or neural network representations of nonlinear maps between function spaces, to accelerate infinite-dimensional Bayesian inverse problems (BIPs) with models governed by nonlinear parametric partial differential equations (PDEs). Neural operators have gained significant attention in recent years for their ability to approximate the parameter-to-solution maps defined by PDEs using as training data solutions of PDEs at a limited number of parameter samples. The computational cost of BIPs can be drastically reduced if the large number of PDE solves required for posterior characterization are replaced with evaluations of trained neural operators. However, reducing error in the resulting BIP solutions via reducing the approximation error of the neural operators in training can be challenging and unreliable. We provide an *a priori* error bound result that implies certain BIPs can be ill-conditioned to the approximation error of neural operators, thus leading to inaccessible accuracy requirements in training. To reliably deploy neural operators in BIPs, we consider a strategy for enhancing the performance of neural operators: correcting the prediction of a trained neural operator by solving a linear variational problem based on the PDE residual. We show that a trained neural operator with error correction can achieve a quadratic reduction of its approximation error, all while retaining substantial computational speedups of posterior sampling when models are governed by highly nonlinear PDEs. The strategy is applied to two numerical examples of BIPs based on a nonlinear reaction–diffusion problem and deformation of hyperelastic materials. We demonstrate that posterior representations of the two BIPs produced using trained neural operators are greatly and consistently enhanced by error correction.

Keywords: uncertainty quantification, partial differential equations, machine learning, neural networks, operator learning, error analysis

Contents

*Corresponding authors

Email addresses: lianghao@oden.utexas.edu (Lianghao Cao), tom.olearyroseberry@utexas.edu (Thomas O'Leary-Roseberry), prashant.jha@austin.utexas.edu (Prashant K. Jha), oden@oden.utexas.edu (J. Tinsley Oden), omar@oden.utexas.edu (Omar Ghattas)

1. Introduction

Many mathematical models of physical systems are governed by parametric partial differential equations (PDEs), where the *states* of the systems are described by spatially and/or temporally–varying functions of PDE solutions, such as the evolution of temperature fields modeled by the heat equation and material deformation modeled by the nonlinear elasticity equation. The parameters, such as thermal conductivity and Young's modulus, of these models, often characterize properties of the physical systems and cannot be directly determined; one has to solve *inverse problems* for that purpose, where the parameters are inferred from sparse and noisy observations of the states. To account for uncertainties in observations and our prior knowledge of the parameters, represented by *prior probability distributions*, in the solutions of inverse problems, they are often formulated via Bayes' rule, or as *Bayesian inverse problems*, for which solutions are probability distributions of the parameters conditioned on the observations, or *posterior probability distributions*. In some scenarios, our prior knowledge of the parameters requires them to be treated as functions, leading to *infinite-dimensional Bayesian inverse problems*. These scenarios arise, for example, when the parameters are possibly spatially varying with uncertain spatial structures. Bayesian inverse

problems are fundamental to constructing predictive models [1–4], and the need for inferring parameters as functions can be found in many areas of engineering, sciences, and medicine [5–10].

For models governed by large-scale highly nonlinear parametric PDEs, numerical simulations are computationally expensive as it involves solving high-dimensional linear systems in an iterative manner many times to obtain solutions with desired accuracy [11]. In these cases, solving infinite-dimensional Bayesian inverse problems can be intractable, as numerically approximating infinite-dimensional posterior distributions with complex structures requires an untenable number of numerical solutions at different parameters, i.e., these problems suffer from the *curse of dimensionality*. Many mathematical and numerical techniques are developed to mitigate the computational burden of these problems. Examples of these techniques are (i) advanced sampling methods exploiting the intrinsic low-dimensionality [12, 13] or derivatives [14–16] of posterior distributions, (ii) direct posterior construction and statistical computation via Laplace approximation [17, 18], deterministic quadrature [19, 20], or transport maps [21–24], and (iii) surrogate modeling using polynomial approximation [25, 26] or model order reduction [27–29] combined with multilevel or multifidelity methods [30–32].

Neural operators, or neural network representations of nonlinear maps between function spaces, have gained significant interest in recent years for their ability to represent the parameter-to-state maps defined by nonlinear parametric PDEs, and approximate these maps using a limited number of PDE solutions at samples of different parameters [33–44]. Notable neural operators include POD-NN [44], DeepONet [38], Fourier neural operator [45], and derivative-informed reduced basis neural networks [39]. The problem of approximating nonlinear maps is often referred to as the *operator learning problem*, and numerically solving the operator learning problem by optimizing the neural network weights is referred to as *training*. Neural operators are fast-to-evaluate and offer an alternative to the existing surrogate modeling techniques for accelerating the posterior characterization of infinite-dimensional Bayesian inverse problems by replacing the nonlinear PDE solves with evaluations of trained neural operators. We explore this alternative surrogate modeling approach using neural operators in this work.

1.1. Neural operators as surrogate models: Advantages and limitations

The direct deployment of trained neural operators as surrogates of the nonlinear PDE-based model transfers most of the computational cost from posterior characterization to the offline generation of training samples and neural network training. Moreover, in contrast to some of the surrogate modeling approaches that approximate the parameter-to-observation or parameter-to-likelihood maps [25, 46], neural operators approximate the parameter-to-state map, or *learn the physical laws*. As a result, they can be used as surrogates for a class of different Bayesian inverse problems with models governed by the same PDEs but with different types of observations and noise models, thus further amortizing the cost of surrogate construction.

While the drastic reduction of computational cost is advantageous, the accuracy of trained neural operators as well as the accuracy of the resulting posterior characterization produced by them needs to be examined. In theory, there are universal approximation results, such as those for DeepONet [38], Fourier neural operators [35], and reduced basis architectures [33, 40], that imply the existence of neural operators that approximate a given nonlinear map between function spaces within certain classes arbitrarily well. In practice, however, constructing and training neural operators to satisfy a given accuracy can be challenging and unreliable. One often observes an empirical accuracy ceiling – enriching training data and enhancing the representation power of network operators via increasing the inner-layer dimensions or the depth of neural networks, as often suggested by universal approximation theories, do not guarantee improved performance. In fact, in certain cases, increasing training data or depth of networks can lead to degraded performance. These behaviors are contrary to some other approximation methods, such as the finite element method with hp-refinement and surrogate modeling using polynomial approximation or model order reduction, for which theoretical results are well-connected to numerical implementation for controlling and reducing approximation errors [47–50]. The unreliability of neural operator performance improvement via training results from several confounding reasons discussed in this work. It is demonstrated via empirical studies in recent work by de Hoop et. al [51], where neural operator performance, measured by their cost–accuracy trade-off, for

approximating the parameter-to-state maps of various nonlinear parametric PDEs are provided.

1.2. Neural operator approximation error in Bayesian inverse problems: Estimation and correction

The approximation error of a trained neural operator in the operator learning problem propagates to the error in the solutions of Bayesian inverse problems when the trained neural operator is employed as a surrogate. By deriving an *a priori* bound, we demonstrate that the approximation error of a trained neural operator controls the error in the posterior distributions defined using the trained neural operator. Additionally, the bounding constant shows that Bayesian inverse problems can be ill-conditioned to the approximation error of neural operators in many scenarios, such as when the prior is uninformative, data is high-dimensional, noise corruption is small, or the models are inadequate. Our theoretical result suggests that for many challenging Bayesian inverse problems, posing accuracy requirements on their solutions may lead to significantly tighter accuracy requirements for neural operator training that are practically inaccessible.

In this work, we consider a strategy for reliably deploying a trained neural operator as a surrogate in, but not limited to, infinite-dimensional Bayesian inverse problems. This strategy is inspired by a recent work by Jha and Oden [52] on extending the goal-oriented *a posteriori* error estimation techniques [53–59] to accelerate Bayesian calibration of high-fidelity models with a calibrated low-fidelity model. Instead of directly using the prediction of the trained neural operator at a given parameter for likelihood evaluation, we first solve a linear *error correction* problem based on the PDE residual evaluated at the neural operator prediction and then use the obtained solution for likelihood evaluation. We show that solving this error-correction problem is equivalent to generating one Newton iteration under some mild conditions, and a trained neural operator with error correction can achieve global, i.e., over the prior distribution, quadratic error reduction when the approximation error of the trained neural operator is relatively small. We expect that the significant accuracy improvement of a trained neural operator from the error correction leads to vital accuracy improvement of the posterior characterization for challenging Bayesian inverse problems. The improvement in the accuracy of posterior characterization is achieved while retaining substantial computational speedups proportional to the expected number of iterative linear solves within a nonlinear PDE solve at parameters sampled from the posterior distribution,

Two numerical examples are provided to showcase the proposed strategy's utility. In the first example, we consider the inference of an uncertain coefficient field in an equilibrium nonlinear reaction–diffusion problem with a cubic reaction term from discrete state observations. The second example concerns the inference of Young's modulus, as a spatially varying field, of a hyperelastic material from discrete observations of its displacement in response to an external force. For both examples, trained neural operators fail to recover all distinctive features of the posterior predictive means despite reaching their empirical accuracy ceilings. In contrast, the error-corrected neural operators are consistently successful in such tasks.

1.3. Related works

Next, we discuss related works on error correction in surrogate modeling approaches for Bayesian inverse problems. To the best of our knowledge, the existing works mainly focus on building data-driven models of the approximation error of surrogate parameter-to-observation maps. The sampling-based techniques for error correction presented in these works are different from the residual-based approach proposed in this work. The term model error correction sometime refers to numerical methods for representing model inadequacy, which is beyond the scope of this work.

In the context of model order reduction, Arridge et. al [60] proposed an offline sampling approach for constructing a normal approximation for the joint probability distribution of the error in surrogate-predicted observations and the parameter over the prior distribution. The probability distribution of the error conditioned on the parameter can be directly used to correct likelihood evaluations defined using an additive Gaussian noise model. This approach simplifies the conditional dependence of the error on the parameter, leading to unreliable performance as pointed out by Manzoni et. al [61], who proposed two alternative error models: one based on radial basis interpolation and the other on linear regression models. Cui et. al [62] presented two methods for adaptively constructing error models during posterior sampling using delayed-acceptance Metropolis–Hastings: one is similar to that of Arridge et. al but with posterior samples, and the other is a zeroth order error correction using the error evaluated at the current Markov chain position.

Additionally, correcting errors in neural network surrogates is explored by Yan and Zhou [63] for large-scale Bayesian inverse problems. They propose a strategy based on a predictor–corrector scheme using two neural networks. The predictor is a deep neural network surrogate of the parameter-to-observable map constructed offline. The corrector is a shallow neural network that takes the prediction of the surrogate as input and produces a corrected prediction. The corrector is trained using a few model simulations produced during posterior characterization.

1.4. Layout of the paper

The layout of the paper is as follows. Section 2 introduces infinite-dimensional Bayesian inverse problems and their numerical solutions in an abstract Hilbert space setting. Section 3 presents the operator learning problem associated with neural operator approximation of nonlinear mappings in function spaces. The sources and reduction of approximation errors in neural network training are discussed. A result on *a priori* bound of the error in the posterior distributions of the Bayesian inverse problem using the operator learning error is provided and interpreted. Section 4 introduces the residual-based error correction problem and discusses its conditional equivalency to a Newton-step problem. Then the error-corrected neural operator is proposed, and computational cost analysis for its use as a surrogate for posterior sampling is provided. Connections of the error-correction problem to goal-oriented *a posteriori* error estimation techniques are also taken up in the same section. Section 5 provides the physical, mathematical, and numerical settings for the two numerical examples of infinite-dimensional Bayesian inverse problems. The empirical accuracy of neural operators and error-corrected neural operators at different training data sizes is presented. Posterior mean estimates generated by the model, trained neural operators, and neural operators with error correction are visualized and examined to understand the accuracy of posterior sampling. The empirical and asymptotic cost analysis results for the posterior sampling are also showcased. The concluding remarks are given in Section 6.

2. Preliminaries

This section introduces infinite-dimensional Bayesian inverse problems in an abstract Hilbert space setting. We refer to [17, 64, 65] and references therein for a more detailed analysis and numerical implementation of infinite-dimensional Bayesian inverse problems. For general treatments of Bayesian inference problems, see [66, 67]. For a reference on probability theory in infinite-dimensional Hilbert spaces, see [68]. For references on the formulation and numerical solutions of partial differential equations (PDEs), see [69, 70]

2.1. Models governed by parametric partial differential equations

Consider a mathematical model that predicts the state $u \in \mathcal{U}$ of a physical system given a parameter $m \in \mathcal{M}$. We assume that the model is governed by PDEs, and $\mathcal{U}$ and $\mathcal{M}$ are infinite-dimensional separable real Hilbert spaces endowed with inner products $(\cdot,\cdot)_{\mathcal{U}}$ and $(\cdot,\cdot)_{\mathcal{M}}$, respectively. The state space $\mathcal{U}$ is a Sobolev space defined over a bounded, open, and sufficiently regular spatial domain $\Omega_u \subset \mathbb{R}^3$. It either consists of functions with ranges in a vector space of dimension $d_s \leq 3$, such as $H^1(\Omega_u;\mathbb{R}^{d_s})$, or time-evolving functions, such as $L^2(0,T;H^1(\Omega_u;\mathbb{R}^{d_s}))$ with $T>0$. The former is appropriate for boundary value problems (BVPs), while the latter is appropriate for initial and boundary value problems (IBVPs). We assume $\mathcal{M}$ consists of spatially-varying scalar-valued functions defined over a set $\Omega_m \subset \overline{\Omega_u}$. The parameter m may appear in the PDEs' boundary conditions, initial conditions, forcing terms, or coefficients.

We specify the model as an abstract nonlinear variational problem as follows. Let $\mathcal{U}_0 \subseteq \mathcal{U}$ be a closed subspace that satisfies the homogenized strongly enforced boundary and initial conditions of the PDEs. Let the solution set $\mathcal{V}_u \subseteq \mathcal{U}$ be an affine space of $\mathcal{U}_0$ that satisfies the strongly enforced boundary conditions and initial conditions that possibly depend on m. The abstract nonlinear variational problem can be written as,

$$\text{Given } m \in \mathcal{M}, \text{ find } u \in \mathcal{V}_u \text{ such that} \quad \mathcal{R}(u,m) = 0 \quad \in \mathcal{U}_0^*\,, \tag{1}$$

where $\mathcal{R} : \mathcal{U} \times \mathcal{M} \to \mathcal{U}_0^*$ is a residual operator associated with the variational form, and $\mathcal{U}_0^*$ is the dual space of the space of test functions $\mathcal{U}_0$. We assume that the residual operator is possibly nonlinear with respect to

both the parameter and state, and the nonlinear variational problem has a unique solution for any $m \in \mathcal{M}$. As a result, we can define a solution operator $\mathcal{F} : \mathcal{M} \to \mathcal{V}_u$, or the *forward operator*, of the model, i.e.,

$$\mathcal{R}(\mathcal{F}(m), m) \equiv 0 \quad \forall m \in \mathcal{M}\,. \tag{2}$$

2.2. Infinite-dimensional Bayesian inverse problems

Let $\boldsymbol{y} \in \mathbb{R}^{n_y}$ denote a set of discrete and noisy observations of the physical system described by the state $u \in \mathcal{U}$. We assume that the state u and observations $\boldsymbol{y}$ are connected via a possibly nonlinear observation operator $\boldsymbol{\mathcal{B}} : \mathcal{U} \to \mathbb{R}^{n_y}$ and a linear additive noise model[1],

$$\boldsymbol{y} = \boldsymbol{\mathcal{B}}(u) + \boldsymbol{n}\,, \tag{3}$$

where $\boldsymbol{n}$ is an unknown noise vector that corrupts the observed data. We assume it is a realization of a random vector $\boldsymbol{N}$ with a probability distribution of $\nu_{\boldsymbol{N}}$ and density of $\pi_{\boldsymbol{N}}$.

Under the Bayesian framework of inverse problems, the model parameter is considered epistemically uncertain and a $\mathcal{M}$-valued random function denoted by M. Its probability distribution ν_M, called *the prior distribution*, incorporates our prior knowledge of the parameter. This leads to the following data model,

$$\boldsymbol{Y} = (\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M) + \boldsymbol{N}\,, \quad M \sim \nu_M, \quad \boldsymbol{N} \sim \nu_{\boldsymbol{N}}\,, \tag{4}$$

where the data is considered a random vector $\boldsymbol{Y}$ due to the influence of the measurement uncertainty and parameter uncertainty, represented by $\boldsymbol{N}$ and M, respectively. In the context of the Bayesian inverse problem, the forward operator $\mathcal{F}$ is also referred to as the *parameter-to-state map*.

Given a particular set of observed data $\boldsymbol{y}^*$, the goal of the Bayesian inverse problem is to construct or sample from the distribution of the parameter conditioned on the observed data $\boldsymbol{y}^*$, or the *posterior distribution*, denoted by $\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)$. The posterior and prior are related via the *likelihood function*, $\mathcal{L}(\cdot; \boldsymbol{y}^*) : \mathcal{M} \to \mathbb{R}_+$, according to *Bayes' rule*,

$$\frac{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_M}(m) = \frac{1}{Z(\boldsymbol{y}^*)} \underbrace{\pi_{\boldsymbol{N}}\left(\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m)\right)}_{=:\mathcal{L}(m;\boldsymbol{y}^*)} \quad \text{a.s.}\,, \tag{5}$$

where $\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)/\,\mathrm{d}\nu_M$ is the Radon–Nikodym derivative of the posterior distribution with respect to the prior distribution, and $Z(\boldsymbol{y}^*) \coloneqq \mathbb{E}_{M\sim\nu_M}\left[\mathcal{L}(M; \boldsymbol{y}^*)\right]$ is the marginal likelihood or model evidence. The likelihood function evaluated at $m \in \mathcal{M}$ returns the probability of observing $\boldsymbol{y}^*$ at m under the assumptions of the data model in (4). Each evaluation of the likelihood function requires solving the model, i.e., evaluating the forward operator. The resulting posterior distribution encodes the additional knowledge of the parameter based on the information of the physical system contained in the observed data.

The prior is often defined as a Gaussian measure $\nu_M \coloneqq \mathcal{N}(m_{\mathrm{pr}}, \mathcal{C}_{\mathrm{pr}})$, where $m_{\mathrm{pr}} \in \mathcal{M}$ is the mean and $\mathcal{C}_{\mathrm{pr}} : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ is a covariance operator. The covariance operator can be defined using an inverse elliptic operator with a Robin boundary condition, which can be expressed in the strong form as

$$\mathcal{C}_{\mathrm{pr}} = \begin{cases} (-\alpha\nabla \cdot \boldsymbol{\Theta}\nabla + \beta)^{-d} & \text{in } \Omega\,, \\ \boldsymbol{\Theta} n \cdot \nabla + \gamma & \text{on } \partial\Omega\,, \end{cases} \tag{6}$$

where n is the outward normal vector, and the negative exponent d is chosen sufficiently large to ensure bounded pointwise variance of the covariance operator and the well-posedness of the Bayesian inverse problem. The hyperparameters of the prior, namely, $\alpha, \beta, d, \boldsymbol{\Theta}, \gamma$, control its properties. For Ω_m with a dimension larger than one, i.e., M is a Gaussian random field, the orthonormal matrix $\boldsymbol{\Theta}$ controls the anisotropy, and if $\boldsymbol{\Theta}$ is the identity matrix, one has isotropic random fields. The parameters α and β together control the random function's pointwise variance and correlation length. Finally, the value of $\gamma \propto \sqrt{\alpha\beta}$ is often chosen to minimize boundary artifacts.

[1] Alternatives to the linear additive noise model, such as a multiplicative noise model or a mixture of both, do not affect the error correction approach introduced in this work. See, e.g., [71] for investigations of alternative noise models.

Remark 1. *Here we distinguish between the physical parameter, denoted as $p \in \mathcal{M}$, and the model parameters $m \in \mathcal{M}$ that is ignored in the description above. While the physical parameter is often the target for inversion, they may have additional constraints so that the solution operator defined by the map between the physical parameter and model solution is only well-defined in a subset $\mathcal{D} \subset \mathcal{M}$. If the uncertain physical parameter $P \sim \nu_P = \mathcal{N}(m_p, \mathcal{C}_p)$ follows a Gaussian distribution and is used as the prior distribution in (5), then the condition $\nu_P(\mathcal{D}) = 1$ for the well-posedness of Bayesian inverse problems might not be satisfied. If this is the case, such constraints can be enforced by introducing a deterministic coupling of the physical parameter with the model parameter using a smooth function ψ such that $\psi(\mathcal{M}) \subseteq \mathcal{D}$ and*

$$P = \psi(M)\,, \quad M \sim \mathcal{N}(m_{pr}, \mathcal{C}_{pr})\,. \tag{7}$$

For example, if P is strictly positive, such as the thermal conductivity field and Young's modulus in Section 5, then $\psi(\cdot) := \exp(\cdot)$ is typically used. Under this assumption, we can formulate Bayesian inverse problems based on a well-understood Gaussian prior distribution, from which we can easily produce numerical samples. On the other hand, it is often the case that one has prior knowledge on P but not M; thus, this assumption requires designing suitable ψ, m_{pr}, and $\mathcal{C}_{pr}$ to reflect our uncertainty in P.

2.3. Numerical solutions of Bayesian inverse problems

We consider a finite-dimensional approximation of the abstract Bayesian inverse problem introduced above. We start from a finite-dimensional approximation of the function spaces $\mathcal{U}$ and $\mathcal{M}$, denoted by $\mathcal{U}^h \subset \mathcal{U}$ and $\mathcal{M}^h \subset \mathcal{M}$, respectively. The finite-dimensional approximation of $\mathcal{M}$ is accomplished by a Galerkin approximation using a set of basis functions $\{\psi_j \in \mathcal{M}\}_{j=1}^{d_m}$. For the state u defined over Ω_u, e.g., $H^1(\Omega_u; \mathbb{R}^{d_s})$, we assume a similar approximation using a set of basis functions $\{\phi_j \in \mathcal{U}\}_{j=1}^{d_g}$ with typically a consistent discretization. For a time-evolving Sobolev space, additionally, a time discretization $0 = t_1 < t_2 < \cdots < t_{d_t} = T$ is required, and a discrete-time integration rule needs to be specified. We use $d_u = d_g d_s$ for BVPs and $d_u = d_g d_t d_s$ for IBVPs to represent the total degrees-of-freedom of $\mathcal{U}^h$.

The residual $\mathcal{R}(u, m)$ can be estimated using numerical techniques such as the finite element method, with particular choices of the basis, test space discretization, and functional evaluations. The forward operator evaluations can then be numerically computed using iterative methods for nonlinear equations, such as fixed point iteration. We denote the forward operator associated with a computer simulation of the model as $\mathcal{F}^h : \mathcal{M}^h \to \mathcal{U}^h$.

The nonlinear Bayesian inverse problem can be solved numerically with a combination of the approximated forward operator $\mathcal{F}^h$, the prior ν_M^h represented in the finite-dimensional bases, i.e., its samples are in $\mathcal{M}^h$, and a method for sampling from the posterior distribution $\nu_{M|\boldsymbol{Y}}^h(|\boldsymbol{y}^*)$ represented in the finite-dimensional bases. For generating numerical results later in Section 5, we restrict ourselves to sampling methods that require the evaluation of the likelihood function but not the gradient and Hessian estimates of the posterior that often yield increased computational efficiency[2]. Here we consider a version of the Markov chain Monte Carlo (MCMC) method called the *preconditioned Crank–Nicolson* (pCN) algorithm [72], a dimension-independent method that is applicable when the likelihood function is of the form:

$$\mathcal{L}(m; \boldsymbol{y}^*) \propto \exp(-\Phi(m))\,, \tag{8}$$

where $\Phi(\cdot) : \mathcal{M} \to \mathbb{R}_+$ is referred to as the *potential*. One of the examples of such a likelihood function is that of normally distributed noise with $\nu_{\boldsymbol{N}} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_N})$ in which case we have

$$\Phi(m) = \frac{1}{2} \left\| \boldsymbol{y}^* - (\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m) \right\|_{\boldsymbol{C_N}^{-1}}^2\,, \tag{9}$$

[2]We acknowledge this is a strong restriction for infinite-dimensional Bayesian inverse problems. The restriction is due to the limitation of neural operators in a general setting. In most cases, the loss function of neural operator training is uninformed of the gradient of the forward operator, leading to poor performances in gradient estimations. See [73] for how to incorporate high-dimensional derivative information in neural operator training.

where $\|\cdot\|_{\boldsymbol{C_N}^{-1}}$ denotes the discrete l^2-norm weighted by the inverse of the noise covariance matrix $\boldsymbol{C_N}^{-1}$. The pCN algorithm is described in Algorithm 1. The procedure for evaluating the potential in (9) for BVPs is illustrated in Figure 1.

Algorithm 1: The preconditioned Crank–Nicolson (pCN) algorithm.

Result: A Markov chain $\{m_j \in \mathcal{M}\}_{j=1}^{n_{\text{chain}}}$ with a stationary distribution of $\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)$ defined by (5).
Input: (1) an initial guess $m_0 \in \nu_M$, (2) a mixing parameter $\beta_{\text{pCN}} \in (0,1)$
$k = 0$;
$\Phi_0 = \Phi(m_0; \boldsymbol{y}^*)$;
while $k < n_{chain}$ **do**
  Sample from prior, $\hat{m} \sim \nu_M$;
  Generate the proposal parameter, $m_{\text{p}} = \sqrt{1-\beta_{\text{pCN}}^2}m_k + \beta_{\text{pCN}}\hat{m}$;
  Evaluate the potential, $\Phi_p = \Phi(m_p)$;
  if $\exp(\Phi_k - \Phi_p) \geq r \sim \mathit{Uniform}([0,1])$ **then**
    $m_{k+1} = m_{\text{p}}$;
    $\Phi_{k+1} = \Phi_{\text{p}}$;
  else
    $m_{k+1} = m_k$;
    $\Phi_{k+1} = \Phi_k$;
  end
  $k \leftarrow k+1$;
end

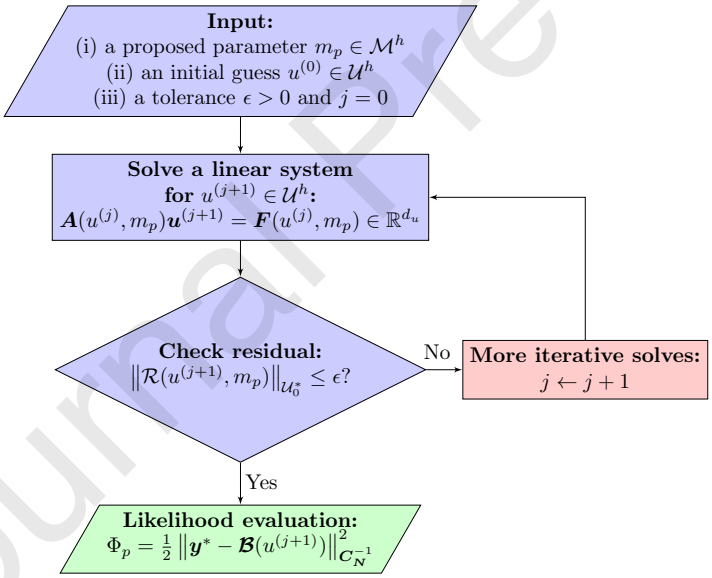


Figure 1: The procedure for numerically evaluating the potential function at a proposed parameter $m_p \in \mathcal{M}^h$ through numerically solving a nonlinear BVP. An iterative scheme is often employed to solve the nonlinear BVP until its residual norm is reduced below a given tolerance. A high-dimensional linear system is solved within each iteration to reduce the residual norm. For a highly nonlinear BVP, many iterations are required to sufficiently reduce the residual norm, making such an iterative process computationally expensive.

MCMC algorithms such as pCN generate Markov chains $\{m_j \in \mathcal{M}^h\}_{j=1}^{n_{\text{chain}}}$ that are used for further analysis. A large portion of the chains must be discarded, or "burned", as they are influenced by initial samples of the chains that might be far from the support of the posterior. Second, the rest of the samples

along each chain are correlated to certain degrees, leading to a much smaller effective sample size, i.e., the number of independently and identically distributed (i.i.d.) samples from the posterior distribution. They are typically much smaller than the actual sample size in the burned Markov chains. For problems with highly localized posterior and highly nonlinear PDEs, the computational cost, measured by the total number of iterative solves for forward operator evaluations, for generating a given number of effective samples from the posterior can be intractably high if performed without introducing advanced algorithmic or numerical techniques.

3. Neural operators and approximation errors

The high computational cost of infinite-dimensional Bayesian inverse problems motivates the development of *surrogates* of the forward operator $\mathcal{F}$ that are fast-to-evaluate and constructed offline, i.e., before receiving observation data and posterior sampling. Employing surrogate forward operators may lead to a significant speedup of posterior sampling, yet typically leads to a trade-off in the accuracy of posterior representations.

In what follows, we introduce the operator learning problems associated with the learning of nonlinear maps between function spaces $\mathcal{M}$ and $\mathcal{U}$ using neural networks, or *neural operators*, which is appropriate for models based on PDEs introduced in Section 2. We review some theoretical and practical aspects of neural operators and their training, focusing on sources and reduction of approximation errors of neural operators in the operator learning setting as well as the propagation of the approximation error in Bayesian inverse problems.

3.1. Operator learning with neural networks

We consider a general class of *operator learning problems*, where we wish to determine a nonlinear map $\widetilde{\mathcal{F}}_{\boldsymbol{w}} : \mathcal{M} \to \mathcal{U}$ parameterized by $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^{d_w}$ such that $\widetilde{\mathcal{F}}_{\boldsymbol{w}}$ is *closest* to a forward operator $\mathcal{F}$ via the following optimization problem:

$$\inf_{\boldsymbol{w}\in\mathcal{W}} \mathcal{J}(\boldsymbol{w}) \coloneqq \left\| \mathcal{F} - \widetilde{\mathcal{F}}_{\boldsymbol{w}} \right\|_{\mathcal{T}}, \tag{10}$$

where $\mathcal{T}$ is the suitable Bochner space for the forward operator $\mathcal{F}$. It is often defined as

$$\mathcal{T} \coloneqq L^p(\mathcal{M}, \nu_M; \mathcal{U}), \quad \|\mathcal{G}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} = \begin{cases} \left(\mathbb{E}_{M\sim\nu_M}\left[\|\mathcal{G}(M)\|_{\mathcal{U}}^p\right]\right)^{1/p}, & p \in [1,\infty), \\ \operatorname{ess\,sup}_{M\sim\nu_M} \|\mathcal{G}(M)\|_{\mathcal{U}}, & p = \infty, \end{cases} \tag{11}$$

where the choice of p depends on the regularity of the forward operator $\mathcal{F}$. The choice $p = 2$ is often taken in practice, and additional learning of derivatives may be included via generalizing to $\mathcal{T} = W^{1,p}(\mathcal{M}, \nu_M; \mathcal{U})$, as in [73].

A neural operator approximates a nonlinear map, such as the forward operator $\mathcal{F}$ for PDEs, by "learning" a neural network representation of the map. The neural network takes as its inputs a finite-dimensional representation of the parameter m, i.e., the degrees-of-freedom of $\mathcal{M}^h$, and produces output as a finite-dimensional representation of $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m) \in \mathcal{U}^h$ through a sequence of compositions of nonlinear functions, i.e., activation functions, acting on affine operations. The coefficient arrays in the affine operations correspond to the neural network weights $\boldsymbol{w}$, which are found by solving the operator learning problem, or *training*, to be specified later. The choice of affine layers and activation functions decides the architecture of the neural operator. Many different architectures are used in practice; typical choices for the affine operations include fully-connected dense layers, residual neural networks (ResNets), and convolution layers, while typical activation functions include ReLU, sigmoid, tanh, softplus, etc. Neural operators can include additional operations such as Fourier transforms [36].

Many classes of neural networks are known to be universal approximators of different classes of functions. Universal approximation results for neural operators in [33, 35, 38, 40] establish the following result that we state in a formal and generalized way. For a nonlinear mapping $\mathcal{G}$ belong to some subset of a Bochner space

$(\mathcal{T}, \|\cdot\|_{\mathcal{T}})$ and any desired error tolerance $\epsilon > 0$, there exists a neural network architecture (e.g. number of layers, breadth, etc.) with weights $\boldsymbol{w}^\dagger$ that defines an operator $\widetilde{\mathcal{F}}_{\boldsymbol{w}^\dagger}$ such that

$$\left\| \mathcal{G} - \widetilde{\mathcal{F}}_{\boldsymbol{w}^\dagger} \right\|_{\mathcal{T}} < \epsilon \,. \tag{12}$$

Such results establish that complex high-dimensional parametric maps can, *in theory*, be learned directly via neural networks arbitrarily well.

For a candidate neural network architecture, the loss function $\mathcal{J}$ can be numerically estimated via sample average approximation, leading to the following empirical risk minimization problem, or *neural operator training problem*, for finite p:

$$\min_{\boldsymbol{w}\in\mathcal{W}} \widetilde{\mathcal{J}}(\boldsymbol{w}; \{m_j\}_{j=1}^{n_{\text{train}}}) \coloneqq \frac{1}{n_{\text{train}}} \sum_{j=1}^{n_{\text{train}}} \left\| u_j - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m_j; \boldsymbol{w}) \right\|_{\mathcal{U}}^p \,, \quad \left\{ \left(m_j, u_j = \mathcal{F}^h(m_j)\right) | m_j \sim \nu_M^h \right\}_{j=1}^{n_{\text{train}}} \,, \tag{13}$$

where samples of training data are generated prior to the training. In some settings, one may be able to incorporate additional information via physics constraints for the optimization problem, perhaps making the optimization problem more difficult [41–43, 45], but more informed.

For many models for multiscale complex systems, fine discretization may be required to resolve necessary physics, which leads to large costs in evaluating the forward operator. In this case, a fundamental issue in neural operator learning arises: one is faced with learning very high-dimensional nonlinear maps from limited training data. This issue makes neural operator learning fundamentally different from typical machine learning due to the limitations of sparse training data. In order to address this issue, the input and output spaces of neural operators are often restricted to some finite-dimensional reduced bases of $\mathcal{M}$ and $\mathcal{U}$. Different architectures of neural operator incorporate different classical reduced basis representations such as proper orthogonal decomposition (POD) [33, 34, 39, 40], Fourier representation [36], multipole graph representations [37], and derivative sensitivity bases [39, 40, 73] among others. These reduced basis representations offer a scalable means of learning structured maps between infinite-dimensional spaces, by taking advantage of their compact representations in specific bases, if such representations exist. From a numerical point of view, this also mitigates direct dependence on the possibly enormous degrees-of-freedom d_u and d_m of the discretized state and parameter.

In the case that a sufficiently accurate network operator $\widetilde{\mathcal{F}}_{\boldsymbol{w}^\dagger}$ (i.e. specific architecture and weights) in a well-designed reduced basis representation is found, one can substitute this surrogate for the forward operator in the likelihood function,

$$\mathcal{L}(m; \boldsymbol{y}^*) \mapsto \widetilde{\mathcal{L}}(m; \boldsymbol{y}^*) \coloneqq \pi_{\boldsymbol{N}} \left(\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}^\dagger})(m) \right) \,, \tag{14}$$

resulting in significant computational speedups for infinite-dimensional Bayesian inverse problems with models governed by nonlinear PDEs; see Figure 2 for this procedure in contrast to the one using the full PDE model in Figure 1. We note that the same neural operator can also be deployed to accelerate Bayesian inverse problems governed by the same model but defined by possibly a variety of different noise models, observation operators, or data, which leads to additional computational cost reduction via the amortization of the model deployment in many different problems. Neural operators have also observed success in their deployment as surrogates for accelerating so-called "outer-loop" problems, such as inverse problems [36], Bayesian optimal experimental design [74], PDE-constrained optimization [75], etc., where models governed by PDEs need to be solved repeatedly at different samples of input variables.

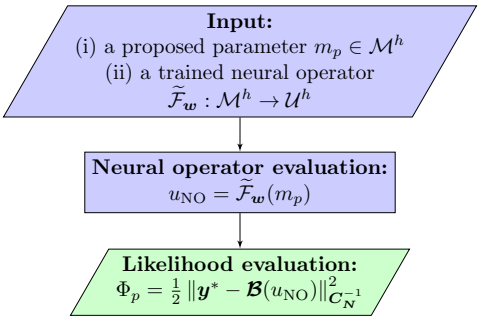


Figure 2: The procedure for numerically evaluating the potential function at a proposed parameter $m_p \in \mathcal{M}^h$ using a trained neural operator $\widetilde{\mathcal{F}}_{\boldsymbol{w}}$. While training a neural operator requires numerically evaluating the PDE solution operator $\mathcal{F}^h$ many times for training data generation, using a trained neural operator in likelihood evaluations significantly reduces the computational cost required for MCMC sampling of posterior distributions.

The hope with neural operators is that the ability to learn very high-dimensional complex nonlinear parametric maps via fast-to-evaluate surrogates provides an alternative approach for making the infinite-dimensional outer-loop problems governed by nonlinear parametric PDEs tractable due to the significant reduction in per-iteration costs. In many settings, however, sufficient accuracy may be out of reach due to some mathematical and numerical issues inherent to neural network construction and training.

3.2. Sources and reduction of the approximation errors

While approximation theories posit the existence of arbitrarily accurate neural networks for operator learning problems, reliably realizing them in practice remains a major obstacle in machine learning research. In particular, one often observes empirical ceilings in the approximation accuracy of neural operators, typically measured by numerically estimating a relative accuracy percentage metric given by

$$100\left(1-\sqrt{\mathbb{E}_{M\sim\nu_M}\left[\frac{\left\|\mathcal{F}(M)-\widetilde{\mathcal{F}}_{\boldsymbol{w}}(M)\right\|^2_{\mathcal{U}}}{\|\mathcal{F}(M)\|^2_{\mathcal{U}}}\right]}\right). \tag{15}$$

The accuracy metric above is often called the *generalization accuracy*. The empirical accuracy ceiling is often persistently observed in the regime where both the error in estimating the loss function (e.g., statistical errors due to finite samples) and truncation error (e.g., reduced bases representation) are asymptotically small. This is due to many confounding issues that we discuss here in brief.

The universal approximation theoretic understanding is typically disconnected from the way neural operators are constructed in practice. Many universal approximation results hinge on density arguments, i.e., certain classes of neural networks are capable of approximating, e.g., polynomials or simple functions to arbitrary accuracy. These density arguments are then used to tie error bounds to well-known approximation results for the class of functions that neural networks are dense in. The theoretical construction of the neural networks used in density arguments may lead to infinitely broad or deep networks in the limit [76–80].

In practice, however, neural network performance eventually degrades as the network gets too broad or deep, and thus neural operators are not constructed in this way. Instead one has to employ a combination of physical intuition, architecture search, and model selection techniques to produce a neural network that will work suitably for the target setting; see, e.g., [81, 82]. The resulting neural network is then calibrated via an empirical risk minimization on finite sample data using an optimizer that searches for local minimizers. The empirical risk minimization problem is typically nonconvex, in which case finding global minimizers is NP-hard. The empirical risk minimization problem over finite samples in (13) introduces both statistical sampling error, as well as optimization error since the local minimizer may be significantly worse than the global minimizer for that particular neural network.

Altogether these different errors lead to a challenging situation where, while neural operators show significant promise in learning complex parametric maps from training data up to a certain accuracy, eventually one cannot continue to improve the empirical accuracy reliably, contrary to other approximation methods such as the finite element method with hp-refinement [47, 48, 83] or surrogate modeling approaches such as polynomial approximation, model order reduction, and Gaussian process approximation [49, 50, 84]. In fact, at a certain point adding more representation power can make the approximation worse, and more training data does not improve the generalization accuracy. For a relatively comprehensive empirical comparison of some neural operators that demonstrates the diminishing marginal returns described above, see recent work by de Hoop et. al in [51].

3.3. Propagation of the approximation errors in Bayesian inverse problems

If one employs a trained neural operator in place of the 'true' PDE solution operator in a Bayesian inverse problem, the approximation error of the neural operator will lead to errors in the resulting posterior distribution. Depending on the conditioning of the particular Bayesian inverse problem of interest, the empirical accuracy ceiling of neural operator training might not be sufficient for reaching the accuracy requirement of the Bayesian inverse problem.

Let $\mathcal{E} \in L^p(\mathcal{M}, \nu_M; \mathcal{U})$ be the approximation error of the operator learning problem,

$$\mathcal{E}(m) \coloneqq \mathcal{F}(m) - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)\,, \quad \nu_M\text{-a.e.}\,. \tag{16}$$

The goal of the following analysis is to find *a priori* bound on the error in the posterior distribution using the error in the operator learning problem, i.e., $\|\mathcal{E}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}$, and obtain an understanding of the bounding constant, i.e., the conditioning of Bayesian inverse problems with respect to the operator learning error. We consider the approximate data random variable $\widetilde{\boldsymbol{Y}}$ as well as the approximate posterior distribution $\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)$ defined using a learned operator $\widetilde{\mathcal{F}}_{\boldsymbol{w}}$ as follows:

$$\widetilde{\boldsymbol{Y}} = (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(M) + \boldsymbol{N} \implies \frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_M}(m) = \frac{1}{\widetilde{Z}(\boldsymbol{y}^*)} \underbrace{\pi_{\boldsymbol{N}}\left(\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(m)\right)}_{=:\widetilde{\mathcal{L}}(m;\boldsymbol{y}^*)} \quad \text{a.s.}, \tag{17}$$

where $\widetilde{Z}(\boldsymbol{y}^*) = \mathbb{E}_{M\sim\nu_M}[\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)]$ is the approximate marginal likelihood. The error in the posterior distribution referred to as $\mathcal{E}_{\text{post}}$ can be represented by the Kullback–Leibler (KL) divergence as follows:

$$\mathcal{E}_{\text{post}} \coloneqq D_{KL}(\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)||\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)) = \mathbb{E}_{M\sim\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}\left[\ln\left(\frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}(M)\right)\right]. \tag{18}$$

The following theorem suggests that for Bayesian inverse problems with sufficiently well-behaved observation operators and normally distributed additive noise, the error in the operator learning problem controls the error of Bayesian inverse problems in an abstract Hilbert space setting introduced in Section 2 and Section 3.1.

Theorem 1 (Operator learning errors in Bayesian inverse problems)**.** *Assume $\mathcal{F}, \widetilde{\mathcal{F}}_{\boldsymbol{w}} \in L^p(\mathcal{M}, \nu_M; \mathcal{U})$, $p \in [2, \infty]$. Assume $\boldsymbol{\mathcal{B}}$ satisfies*

$$\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m)\|_2 \le c_B \|\mathcal{F}(m)\|_{\mathcal{U}}\,, \quad \left\|(\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(m)\right\|_2 \le \widetilde{c_B} \|\mathcal{F}(m)\|_{\mathcal{U}}\,, \quad \nu_M\text{-a.e.}\,,$$
$$\left\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m) - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(m)\right\|_2 \le c_L \|\mathcal{E}(m)\|_{\mathcal{U}}\,, \quad \nu_M\text{-a.e.}\,,$$

for constants $c_B, \widetilde{c_B} > 0$ and $c_L \ge 0$. Assume $\nu_{\boldsymbol{N}} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_N})$ is normally distributed. Assume $\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)$, $\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)$, and ν_M are pairwise equivalent. For a given set of observation data $\boldsymbol{y}^* \in \mathbb{R}^{n_y}$, we have

$$\mathcal{E}_{post} \le c_{BIP} \|\mathcal{E}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}\,, \quad c_{BIP} = c_1\,(c_2(1) + c_3(p))\,c_L\,,$$

where the constants $c_1, c_2, c_3 > 0$ are defined by

$$c_1 = \frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\left((\boldsymbol{\mathcal{B}}\circ\mathcal{F})(\cdot)+(\boldsymbol{\mathcal{B}}\circ\widetilde{\mathcal{F}}_{\boldsymbol{w}})(\cdot)-2\boldsymbol{y}^*\right)\right\|_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})}, \tag{19}$$

$$c_2(p) = \left\|\exp\left(-\frac{1}{2}\left\|\boldsymbol{y}^*-(\boldsymbol{\mathcal{B}}\circ\widetilde{\mathcal{F}}_{\boldsymbol{w}})(\cdot)\right\|^2_{\boldsymbol{C}_{\boldsymbol{N}}^{-1}}\right)\right\|^{-1}_{L^p(\mathcal{M},\nu_M)}, \tag{20}$$

$$c_3(p) = \frac{c_2(1)}{c_2(q)} \in [1, c_2(1)], \quad q = \begin{cases}\infty, & p = 2;\\ p/(p-2), & p\in(2,\infty);\\ 1, & p=\infty.\end{cases} \tag{21}$$

Proof. We only provide a sketch of proof here for $p \in [2,\infty)$. The detailed proof is provided in Appendix A. The following transformation can be made to $\mathcal{E}_{\text{post}}$

$$\mathcal{E}_{\text{post}} = \underbrace{\ln\left(\frac{Z(\boldsymbol{y}^*)}{\widetilde{Z}(\boldsymbol{y}^*)}\right)}_{\text{Ⓐ}} + \underbrace{\frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\mathbb{E}_{M\sim\nu_M}\left[\ln\left(\frac{\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)}{\mathcal{L}(M;\boldsymbol{y}^*)}\right)\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right]}_{\text{Ⓑ}}.$$

We seek to bound the two terms. By applying Cauchy–Schwarz inequality and Minkowski inequality, we have

$$\left\|\Phi-\widetilde{\Phi}\right\|_{L^{p^*}(\mathcal{M},\nu_M)} \le c_1 c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}, \quad p^*\in[1,p/2],$$

$$c_1 \le \frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2\left(c_B\left\|\mathcal{F}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} + \widetilde{c_B}\left\|\widetilde{\mathcal{F}}_{\boldsymbol{w}}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} + 2\left\|\boldsymbol{y}^*\right\|_2\right) < \infty,$$

where $\Phi(m)$,$\widetilde{\Phi}(m)$ are the potentials defined with $\mathcal{F}$,$\widetilde{\mathcal{F}}_{\boldsymbol{w}}$ as in (9). Using Jensen's, Hölder's, and Minkowski inequalities, we have

$$c_2(1)^{-1} \ge \exp\left(-\frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2\left(\left\|\boldsymbol{y}^*\right\|_2 + \widetilde{c_B}\left\|\widetilde{\mathcal{F}}_{\boldsymbol{w}}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}\right)\right) > 0,$$

where $c_2 = c_2(p)$ as a function of p is defined in (19). The second term in $\mathcal{E}_{\text{post}}$ can be bounded by the following term using Jensen's and Hölder's inequalities:

$$\text{Ⓑ} \le c_3(p)\left\|\Phi-\widetilde{\Phi}\right\|_{L^{p^*}(\mathcal{M},\nu_M)} \le c_1 c_3(p) c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}.$$

The term involving normalization constants can be bounded using inequalities $\log(1+x) \le x, \forall x \ge 0$, $|e^{-x_1}-e^{-x_2}| \le |x_1-x_2|, \forall x_1, x_2 \ge 0$, and Hölder's inequality,

$$\begin{aligned}\text{Ⓐ} &\le c_2(1)\left|\mathbb{E}_{M\sim\nu_M}\left[\exp(-\Phi(m)) - \exp(-\widetilde{\Phi}(m))\right]\right| \\ &\le c_2(1)\left\|\Phi-\widetilde{\Phi}\right\|_{L^{p/2}(\mathcal{M},\nu_M)} \le c_1 c_2(1) c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}.\end{aligned}$$

Therefore, the bound $\mathcal{E}_{\text{post}} \le c_{\text{BIP}}\left\|\mathcal{E}\right\|_{L^p(M,\nu_M;\mathcal{U})}$ holds for $c_{\text{BIP}} = c_1\left(c_2(1)+c_3(p)\right)c_L$. □

Remark 2. *Conditions under which the equality sign holds in the error bound in Theorem 1: the observation operator is linear and the operator learning error $\mathcal{E}$ is in the kernel of $\boldsymbol{\mathcal{B}}$, i.e., $(\boldsymbol{\mathcal{B}}\circ\mathcal{E})(m) = 0$, ν_M-a.e., and model discrepancy does not pollute the predicted observation. In this case, $c_L = 0$ and the equality holds with $\mathcal{E}_{post} = 0$; see [85] for a detailed discussion.*

We note that the constants c_1 and c_L are associated with the error in the evaluation of the potential as defined in (9), while c_2 and c_3 are associated with the error in the evaluation of the marginal likelihood. The magnitude of the bounding constant is, generally speaking, dictated by the magnitude of the data misfit term $\left\|\boldsymbol{y}^* - \boldsymbol{y}^{\text{pred}}\right\|_{\boldsymbol{C}_{\boldsymbol{N}}^{-1}}$, with $\boldsymbol{y}^{\text{pred}}$ being the predicted data, over the prior distribution for both the model and the surrogate. This constant is large for challenging Bayesian inverse problems that have uninformative prior, high-dimensional data, small noise corruption, or inadequate models. In these cases, the posterior distributions are typically highly localized and/or much different than the prior distributions. This is somewhat expected as neural operators are trained with data generated from prior distributions, and the performance of neural operators is likely to deteriorate when posterior sampling is concentrated in areas of the parameter space where training samples are relatively sparse. Additionally, Bayesian inverse problems are susceptible to the approximation error originating from overfitting to training samples, regardless of whether they adequately resolve the region of the parameter space with high likelihood values.

This result implies that significant magnification of the approximation error, when propagated through challenging Bayesian inverse problems, is likely. It further implies that the empirically observed accuracy ceiling for neural operator training might not be sufficient for the accuracy requirements of these types of Bayesian inverse problems. We note that this is often not the case for conventional surrogate modeling techniques, for which similar results are derived with the additional assumption that the error in Bayesian inverse problem solutions generated via a surrogate model asymptotically diminishes due to the asymptotic diminishing of the surrogate approximation error as $n_{\text{train}} \to \infty$; see [84, 86–88].

4. Residual-based error correction of neural operator predictions

Methods for estimating error, and subsequently correcting the error, in approximations of solutions to the forward problems fall into the category of *a posteriori* error estimation and typically involve computing residuals representing the degree to which approximate solutions fail to satisfy the forward problems; i.e., residuals evaluated at the approximate solutions. In this section, we propose a strategy, following earlier works on *a posteriori* error estimation techniques in [52, 54], that enhances the application of neural operators in infinite-dimensional Bayesian inverse problems using the PDE residual.

In particular, we ask, given a neural operator prediction $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m) \in \mathcal{U}$ at $m \in \mathcal{M}$ that is reasonably close to the true solution $\mathcal{F}(m) \in \mathcal{V}_u$, if it is possible to cheaply compute a *corrected* solution, $u_C \in \mathcal{V}_u$, that has a significantly smaller approximation error, so that the mapping $m \to u_C(m)$ can more reliably achieve the accuracy requirements for deployment in Bayesian inverse problems. It turns out that if the residual operator $\mathcal{R}$ as in (1) has a sufficiently regular derivative, a correction can be computed by solving a linear variational problem based on $\mathcal{R}$ at $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)$, and this correction may lead to quadratic error reduction. For highly nonlinear problems, computing one correction step is inexpensive relative to evaluating the forward operator; thus this approach retains substantial speedups for Bayesian inverse problems.

In what follows, we first introduce the linear variational problem associated with the error correction. We show that it is a Newton step under some conditions, and thus the Newton–Kantorovich theorem can be directly applied to understand the error reduction. Then an application of this general procedure for neural operators is discussed. We discuss the possibility of conditions on the approximation error of a trained neural operator that ensures global quadratic error reduction of the correction step. Next, we make a connection between the error correction problem and *a posteriori* error estimation techniques for estimating modeling error. Lastly, an analysis of the computational cost of MCMC sampling of the posterior using an error-corrected neural operator is given.

4.1. The residual-based error correction problem

Consider an expansion of the residual operator $\mathcal{R}$ in the state space $\mathcal{U}$, assuming for now that $\mathcal{R}$ is at least twice differentiable with respect to the state variable. For an arbitrary pair $(u, m) \in \mathcal{U} \times \mathcal{M}$, the first and second order derivatives with respect to the state variable, $\delta_u \mathcal{R}(u, m) : \mathcal{U} \to \mathcal{U}_0^*$ and $\delta_u^2 \mathcal{R}(u, m) : \mathcal{U} \times \mathcal{U} \to \mathcal{U}_0^*$,

are given by,

$$\begin{aligned}\delta_u\mathcal{R}(u,m)v &= \lim_{\theta\to 0}\frac{1}{\theta}\left(\mathcal{R}(u+\theta v,m)-\mathcal{R}(u,m)\right)\,, \quad \forall v\in\mathcal{U}\,,\\ \delta_u^2\mathcal{R}(u,m)(v,q) &= \lim_{\theta\to 0}\frac{1}{\theta}\left(\delta_u\mathcal{R}(u+\theta q,m)v-\delta_u\mathcal{R}(u,m)v\right)\,, \quad \forall v,q\in\mathcal{U}\,.\end{aligned} \tag{22}$$

The Taylor series expansion of $\mathcal{R}(\mathcal{F}(m),m)$ in terms of u involving up to second-order derivative of $\mathcal{R}$ is given by

$$\begin{aligned}\underbrace{\mathcal{R}(\mathcal{F}(m),m)}_{=0} &= \mathcal{R}(u,m)+\delta_u\mathcal{R}(u,m)(\mathcal{F}(m)-u)\\ &\quad+\int_0^1\delta_u^2\mathcal{R}(u+s(\mathcal{F}(m)-u),m)(\mathcal{F}(m)-u,\mathcal{F}(m)-u)(1-s)\,\mathrm{d}s.\end{aligned} \tag{23}$$

In the regime where the higher order terms are small, such as where $\|\mathcal{F}(m)-u\|_{\mathcal{U}}$ is small, the following approximation can be made

$$\begin{aligned}\delta_u\mathcal{R}(u,m)\mathcal{F}(m) &= -\mathcal{R}(u,m)+\delta_u\mathcal{R}(u,m)u+O(\|\mathcal{F}(m)-u\|_{\mathcal{U}}^2)\\ &\approx -\mathcal{R}(u,m)+\delta_u\mathcal{R}(u,m)u\,.\end{aligned} \tag{24}$$

This leads to the following linear variational problem, which we refer to as the *residual-based error correction problem*, to compute the new estimation u_C of $\mathcal{F}(m)$ given an initial estimation $u\in\mathcal{U}$:

$$\text{Given } m\in\mathcal{M} \text{ and } u\in\mathcal{U},\ \text{find } u_C\in\mathcal{V}_u \text{ such that} \quad \delta_u\mathcal{R}(u,m)u_C=-\mathcal{R}(u,m)+\delta_u\mathcal{R}(u,m)u\,. \tag{25}$$

Assuming the existence of a unique solution to this linear variational problem under Lax–Milgram theorem [89, pg. 310], solving this problem is equivalent, under some additional conditions, to generate a single Newton iteration for the nonlinear equation (1) reformulated as

$$\text{Given } m\in\mathcal{M},\ \text{find } v\in\mathcal{U}_0 \text{ such that} \quad \widetilde{\mathcal{R}}(v,m)=0\,, \quad \widetilde{\mathcal{R}}(v,m)\coloneqq\mathcal{R}(v+u_L,m)\,, \tag{26}$$

for a given[3] $u_L\in\mathcal{V}_u$. The Newton iteration $\{v_j\in\mathcal{U}_0\}_{j=0}^{\infty}$ is given by

$$v_{j+1}=v_j-\delta_v\widetilde{\mathcal{R}}(v_j,m)^{-1}\widetilde{\mathcal{R}}(v_j,m)\,, \quad j>0\,. \tag{27}$$

Consider one Newton step with $u-u_L-u_\perp$, where $\mathcal{U}_0^\perp$ is the orthogonal complement of $\mathcal{U}_0$, and $u_\perp$ is the orthogonal projection of $u-u_L$ to $\mathcal{U}_0^\perp$.

$$v_C=u-u_L-u_\perp-\delta_v\widetilde{\mathcal{R}}(u-u_L-u_\perp,m)^{-1}\widetilde{\mathcal{R}}(u-u_L-u_\perp,m)\,,$$

The equivalency $v_C=u_C-u_L$ can be established when, for example, (i) $\mathcal{U}=\mathcal{U}_0=\mathcal{V}_u$, (ii) $u\in\mathcal{V}_u$, i.e., u has the correct strongly enforced boundary and initial conditions with $u_\perp=0$, or (iii) $\mathcal{R}(u+\mathcal{U}_0^\perp,m)\equiv\mathcal{R}(u,m)$, i.e., the residual operator is invariant to changes in the strongly enforced boundary and initial conditions of u.

If the equivalency between the Newton step problem and the error correction problem can be established, the latter can be understood in the setting of the Newton–Kantorovich theorem in Banach spaces; see, [89, 90]

[3]The so-called lifting function u_L is simply a part of the mathematical construction of boundary value problems. It allows us to weakly formulate certain PDEs in symmetric form (i.e., the trial and test spaces are the same) under strongly enforced boundary conditions. A lifting function exists due to the surjective trace operator $\gamma_0: H^s(\Omega)\overset{\text{onto}}{\to}H^{s-1/2}(\partial\Omega)$ for any $s>1/2$ [69], and a strongly-enforced boundary condition is often specified using an element of $H^{s-1/2}(\partial\Omega)$. While, in theory, one can explicitly find a lifting function $u_L\in H^s(\Omega)$ for a given strongly enforced boundary condition, it is not necessary for the analysis and numerical implementation in this work.

and references therein. The theorem gives sufficient conditions for the quadratic convergence of a Newton iteration starting at u for solving the fixed point problem associated with (26). It implies that within a regime in which u is sufficiently close to $\mathcal{F}(m)$, the solution to the error correction problem is guaranteed a quadratic error reduction, i.e.,

$$\|\mathcal{F}(m) - u_C\|_{\mathcal{U}} \leq c \, \|\mathcal{F}(m) - u\|_{\mathcal{U}}^2 \tag{28}$$

for some constant $c > 0$. Consequently, we expect u_C to have a much smaller error than u in such a regime.

We remark that in the rare cases where the equivalency of the two problems cannot be established, one may still retain the conditional quadratic error reduction property for the error correction problem, but perhaps with stronger conditions than those stated by the Newton–Kantorovich theorem. This situation may arise in practice, for example, when both conditions for equivalency suggested above fail, and the numerical implementation of the orthogonal projection from $\mathcal{U}$ to $\mathcal{V}_u$ is not straightforward. The stronger conditions for quadratic error reduction should incorporate additional sensitivity factors for when the initial position of a Newton iteration orthogonally deviates from the domain of the Jacobian operator of the nonlinear system.

4.2. Correcting neural operator predictions: Global quadratic reduction of approximation error

For well-designed and well-trained neural operators, we expect they are within the regime where the approximation error is small. However, in such a regime, the reduction of the approximation error via neural network construction or training is ineffective, as discussed in Section 3.2. On the other hand, solving a residual-based error correction problem returns a significant reduction in approximation error that may not be reliably achieved in the training phase of neural operators. This reduction of the approximation errors may lead to crucial improvements in the quality of the solutions to Bayesian inverse problems accelerated by neural operators. To this end, we consider the following linear problem that returns a neural operator with error correction $\widetilde{\mathcal{F}}_C : \mathcal{M} \ni m \mapsto u_C \in \mathcal{V}_u$:

$$\begin{aligned} &\text{Given } m \in \mathcal{M}, \text{ find } u_C \in \mathcal{V}_u \text{ such that} \\ &\qquad \delta_u \mathcal{R}(\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m), m) u_C = -\mathcal{R}(\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m), m) + \delta_u \mathcal{R}(\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m), m) \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m), \end{aligned} \tag{29}$$

i.e., we let $u = \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)$ in (25). The mapping $\widetilde{\mathcal{F}}_C$ is then used in likelihood evaluations at m for solving Bayesian inverse problems:

$$\mathcal{L}(m; \boldsymbol{y}) \mapsto \widetilde{\mathcal{L}}_C(m; \boldsymbol{y}^*) \coloneqq \pi_N(\boldsymbol{y}^* - \boldsymbol{\mathcal{B}}(\widetilde{\mathcal{F}}_C(m))) \, . \tag{30}$$

See Figure 3 for this procedure in contrast to the ones in Figures 1 and 2.

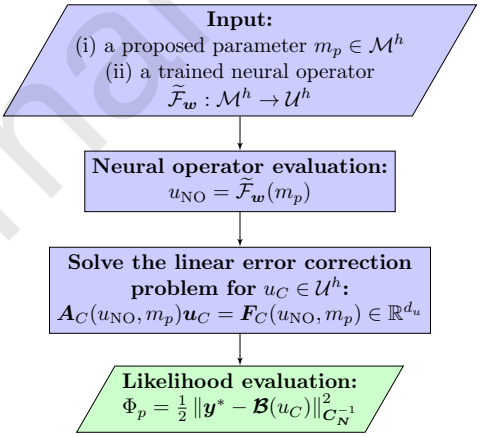


Figure 3: The procedure for numerically evaluating the potential function at a proposed parameter $m_p \in \mathcal{M}^h$ using a trained neural operator with error correction $\widetilde{\mathcal{F}}_{\boldsymbol{w}}$. By evaluating the likelihood function using the solution to a linear error correction problem based on the prediction of a trained neural operator, we expect the evaluation to be much more accurate than using merely the neural operator prediction. This accuracy improvement comes with a price: one has to solve a high-dimensional linear system at each likelihood evaluation. However, this computational cost is mild compared to the iterative solve shown in Figure 1.

Next we discuss a sufficient condition on the approximation error of a trained neural operator for global quadratic error reduction via the error correction problem. Under the framework of the Newton–Kantorovich theorem, if the following two conditions are satisfied at states that are nearby $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)$ for all $m \in \mathcal{M}$: (i) the neural operator error correction problems have unique solutions under the Lax–Milgram theorem, and (ii) the residual derivative is locally Lipschitz continuous with respect to the state variables, then for $\mathcal{F} \in L^\infty(\mathcal{M}, \nu_M; \mathcal{U})$ there exists a global convergence radius upper bound $r_C > 0$ that depends on the properties of $\mathcal{R}$ such that

$$\|\mathcal{E}\|_{L^\infty(\mathcal{M},\nu_M;\mathcal{U})} < r_C \implies \|\mathcal{E}_C\|_{L^\infty(\mathcal{M},\nu_M;\mathcal{U})} \leq c_R \|\mathcal{E}\|^2_{L^\infty(\mathcal{M},\nu_M;\mathcal{U})} , \tag{31}$$

where $\mathcal{E}_C(m) \coloneqq \widetilde{\mathcal{F}}_C(m) - \mathcal{F}(m)$, ν_M-a.e., and $c_R > 0$ is a constant. We state the full form of this corollary in Appendix B.

We note that even though the global convergence radius upper bound is enforced in the L^∞ Bochner norm, which may not be compatible with the operator learning setting, the global quadratic error reduction is observed in practice for neural operators trained in the L^2 Bochner norm with pre-asymptotic accuracy; see, e.g., Figure 5 and Figure 12 in our numerical examples to be specified in Section 5. We expect that these results are natural consequences of the reduced basis representation built into the neural operator architectures, in which the operator learning problem can be essentially represented in finite-dimensional coefficient spaces, even though the optimization is typically performed in the $L^2(\mathcal{M}, \nu_M; \mathcal{U})$ topology. Moreover, when the global convergence radius upper bound is small, trained neural operators with pre-asymptotic accuracy may not be sufficient for a guaranteed global quadratic error reduction via error correction. In these cases, performance-improving techniques for neural operator construction and training can be crucial. Even though these techniques may not produce neural operators with arbitrary accuracy, they may be able to reduce the approximation error so that its norm is smaller than the global convergence radius upper bound for a guaranteed quadratic error reduction according to the Newton–Kantorovich theorem.

Remark 3. *Here we briefly discuss the limitation of the proposed strategy for error reduction in likelihood evaluations. First, to formulate the error correction problem in* (25) *using the residual derivative, we require the residual operator $\mathcal{R}$ to be Fréchet differentiable with respect to the state variable near $\widetilde{\mathcal{F}}(m)$ over the prior distribution. Second, if the posterior distribution happens to concentrate on the region of the parameter space where the radius of convergence around $\mathcal{F}(m)$ for the Newton iteration is very small, the residual-based error correction may lead to overall worse likelihood evaluations during posterior sampling due to the possible divergence of the Newton iteration. In this situation, the sampling-based error correction strategies introduced in Section 1.3 could be superior in the consistency of error reduction, particularly when the marginal cost (e.g., generating more training data) of improving the generalization accuracy of a neural operator is high. The computational cost of the proposed strategy is discussed in Section 4.4.*

4.3. Connection to goal-oriented a posteriori error estimation

The procedure we described to correct the predictions of a trained neural operator is based on the estimation of modeling error using the goal-oriented a posteriori error estimates that provide computationally inexpensive estimates of the error in the quantities of interests (QoIs), see [52–59, 91–94]. Originally, a goal-oriented error estimation technique was developed [53, 55, 57, 91–93] to perform mesh adaption based on the specific measure of error – error in QoIs – as opposed to the energy norm. Techniques were developed to adapt the mesh to control the error in QoIs which is not necessarily possible by relying on the energy norm. It was soon realized that goal-oriented error estimates can be used to estimate the modeling error – error in predictions via the fine (high-fidelity) and coarse (low-fidelity) models – further expanding the utility of the technique; see [52, 54]. Here we briefly cover the topic of estimation of modeling error to relate to the development in previous subsections.

Suppose $u \in \mathcal{U}$ is a solution to the variational problem, $\mathcal{R}(u) = 0 \in \mathcal{U}^*$, assuming test and trial spaces are the same for simplicity, and $p \in \mathcal{U}$ is the solution to the dual problem: $-\langle \delta_u \mathcal{R}(u)v, p\rangle = \langle \delta_u \mathcal{Q}(u), v\rangle$

$\forall v \in \mathcal{U}$, where $\langle \cdot, \cdot \rangle$ denotes a duality pairing between $\mathcal{U}^*$ and $\mathcal{U}$. Here $\mathcal{Q} : \mathcal{U} \to \mathbb{R}$ is a differentiable QoI functional. For any arbitrary functions $\widetilde{u}, \widetilde{p} \in \mathcal{U}$ the following holds,

$$\mathcal{Q}(u) - \mathcal{Q}(\widetilde{u}) = \langle \mathcal{R}(\widetilde{u}), p \rangle + r(u, p, \widetilde{u}, \widetilde{p}) \approx \langle \mathcal{R}(\widetilde{u}), p \rangle, \tag{32}$$

where r is the remainder term involving derivatives of residual and errors in forward and dual solutions, $e = u - \widetilde{u}$ and $\varepsilon = p - \widetilde{p}$, respectively. When e and ε are sufficiently small, r can be ignored. We note that several versions of such estimates as the equation above can be derived; see [54]. While estimates like the above provide an approximation of the QoI error, they still require access to the pair of the forward and dual solutions, (u, p). Because u can be written as $u = \widetilde{u} + e$, if the error e can be approximately computed, referring to the approximation of e by $\hat{e}$, then u can be approximated using $u \approx \widetilde{u} + \hat{e}$. The approximation $\hat{e}$ of the error e can be obtained by solving the following linear variational problem:

$$\text{Find } \hat{e} \in \mathcal{U} \text{ such that} \quad \delta_u \mathcal{R}(\widetilde{u})\hat{e} = -\mathcal{R}(\widetilde{u}) \qquad \in \mathcal{U}^* \,. \tag{33}$$

The same steps can be used to also compute p approximately following the discussion in the earlier references.

The above problem is formally derived via the linearization of the nonlinear variational problem similar to the one presented in Section 4.1. The steps discussed above are somewhat similar to estimating the error $u_C - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m) \approx \mathcal{F}(m) - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)$ given a neural operator prediction $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(m)$ proposed in Section 4.2. However, one needs to also consider the error in $\mathcal{U}^{\perp}$ for when $\mathcal{U}_0 \subset \mathcal{U}$, as the neural operator prediction u may not satisfy the strongly enforced boundary or initial conditions. Note that the error correction problem is essentially an extended error estimation problem, where one solves for the error in the affine space $\mathcal{U}_0 - \widetilde{u}$, thus similar to an error estimate that includes the error of $\widetilde{u}$ at those boundaries.

4.4. Discussion of computational costs

In this subsection, we briefly discuss the computational costs when the error-corrected neural operator is deployed as a surrogate of the forward operator in Bayesian inverse problems, in particular the expected computational speedups in MCMC sampling of the posterior. The neural operator construction and training have offline costs due to training data generation, building or finding appropriate architecture, and training the neural operator. The cost of generating training data is the same as the cost of solving the PDEs and sampling from the prior; here we neglect the cost due to the latter. The additional computation may be required for finding reduced bases that involve expectations, as is done in the network strategies discussed in [39, 40], which we use in our numerical examples presented in Section 5. The cost of constructing an appropriate architecture is more abstract, since this procedure may involve repeating many different training runs for different architectural hyperparameters.

In the case that reduced basis neural operator methods are used, the training cost can be made independent of the discretization dimensions d_m, d_u, and instead be made to scale with the dimension of the reduced bases, since in this case the training data can be projected into these bases in a pre-training step. For this reason, the cost of neural network training can be ignored asymptotically for very high-dimensional discretization. We note that neural network training may represent a significant cost in the pre-asymptotic regime, i.e. on the order of the cost of training data generation, depending on the type of architecture used and the number of iterations required to train a sufficiently accurate neural operator.

The cost of posterior sampling to acquire a Markov chain of size n_{chain}, according to our settings in Section 2.2, is dictated by the cost of solving the governing PDEs for likelihood evaluations. The total cost is a sum of the online posterior sampling cost and the offline cost:

$$\text{Total Cost } = (n_{\text{chain}} \times \text{Evaluation Cost}) + \text{Offline Cost}\,. \tag{34}$$

The speedups realized in practice by using a neural operator are measured by the ratio of the total cost of MCMC sampling using the forward operator to the total cost of MCMC sampling using a neural operator,

$$\text{Speedup}_{NO} = \frac{(n_{\text{chain}} \times \text{Cost}_{PDE})}{(n_{\text{chain}} \times \text{Cost}_{NO}) + \text{Offline Cost}}\,. \tag{35}$$

In the regime where a large number of likelihood evaluations are required for generating a Markov chain of length n_{chain}, the offline cost of neural operator training is likely negligible in relation to the online cost of posterior sampling. Assuming the acceptance rates are the same for Markov chains generated by the forward operator with a neural operator as its surrogate, the speedup of the neural operator asymptotically becomes the ratio between the averaged evaluation cost for the forward operator and the neural operator over the posterior distribution.

$$\text{Asymptotic Speedup }_{NO} \approx \frac{\text{Cost}_{PDE}}{\text{Cost}_{NO}}\,. \tag{36}$$

In general, the cost ratio of the PDE solves to the neural operator can be very high for highly nonlinear PDEs with fine discretization, i.e., large d_u and d_m, particularly if the neural network architecture is designed to be invariant to the discretization dimensions.

When a neural operator with error correction is used as a surrogate of the forward operator in posterior sampling, the asymptotic speedup is approximately equal to the number of iterations, denoted as $N_{\text{nonlinear}}$, within a given iterative scheme for nonlinear equations used for evaluating the forward operator, averaged over the posterior distribution. We assume a linear problem of a similar size and numerical properties to the error correction problem is solved at each iteration. Under this assumption, we may assume that an error correction step and a linear solve within the iterative scheme have a similar computational cost. Therefore, the asymptotic speedup is given by

$$\text{Asymptotic Speedup}_{ECNO} = \frac{\text{Cost}_{PDE}}{\text{Cost}_{NO} + \text{Cost}_{EC}} \approx \frac{N_{\text{nonlinear}} \times \text{Cost}_{EC}}{\text{Cost}_{NO} + \text{Cost}_{EC}} \approx N_{\text{nonlinear}}\,. \tag{37}$$

The proposed method thus yields favorable speedups for models governed by highly nonlinear PDEs with fine discretization, where evaluating the forward operator requires many iterative solves, and the posterior sampling requires many queries of the forward operator.

5. Numerical examples

In this section, we demonstrate through numerical examples how the residual-based error correction both improves the accuracy of posterior representations of challenging Bayesian inverse problems relative to using a trained neural operator alone, and offers a computational advantage over directly using the expensive forward model in likelihood evaluations. The first example concerns inferring the uncertain coefficient field in a nonlinear reaction–diffusion problem with a cubic reaction term. The second example involves the inference of Young's modulus as a spatially-varying uncertain field for a hyperelastic neo-Hookean material undergoing deformation. For each of the two examples, we generate samples from the posterior distributions for likelihood functions defined using (i) model predictions, (ii) predictions by three trained neural operators whose measured accuracy is close to the ceiling for a given architecture with an increasing number of training samples, and (iii) error-corrected predictions of these neural operators. We visualize and compare the accuracy of their posterior predictive means and analyze the computational cost.

In what follows, we first discuss the neural operator architecture and training as well as the software used in our calculations. Then the physical, mathematical, and numerical settings of the two examples are provided. Lastly, we present the visualization of posterior predictive means and cost analysis.

5.1. Derivative-informed reduced basis neural operator

We consider a derivative-informed reduced basis neural network [39, 40, 95] for constructing neural operators in the two numerical examples. This neural network architecture uses both derivative and principal information of the solution map to construct appropriate reduced bases for the inputs and outputs. This strategy exploits the compactness of the forward operator, if it exists, to construct a parsimonious and mesh-independent neural network approximation, making it a suitable strategy for learning discrete representations of mappings between function spaces. The input reduced basis uses derivative information to detect subspaces of the inputs that the outputs are most sensitive in expectation over ν_M. The output

reduced basis is proper orthogonal decomposition (POD), an optimal linear basis for learning operators in the L^2 norm [96, 97]. The neural operator learns a nonlinear coefficient mapping between the two subspaces via an adaptively constructed and trained ResNet [40]. These networks provide a reliable way of detecting appropriate breadth for a map by detecting useful bases directly from the map instead of during neural network training. The adaptive training adds layers in a way that only marginally perturbs the existing coefficient mapping, allowing one to adaptively detect appropriate nonlinearity (depth) until overfitting is detected. We only consider this network since it gives a good practical performance, outperforming other conventional reduced basis strategies and overparametrized networks, while suffering from the same limitations that are typical of neural operators in general: limitations in achieving high accuracy given more data, and approximation power. It is thus general enough to demonstrate the efficacy of our proposed method.

For both numerical examples, we use networks with reduced basis dimensions of 50, and 10 nonlinear ResNet layers, each with a layer rank of 10, and softplus activation functions. The networks are trained and constructed adaptively, one ResNet layer at a time, using the Adam optimizer with a learning rate of 10^{-3}, and gradient batch size of 32. Each adaptive layer training problem executes 100 epochs, starting from 2 layers to 10, with one final end-to-end training, accounting for a total of 1000 epochs. The efficacy of this adaptive approach is demonstrated in [40]. The networks are trained for varying availability of training data, which are reported when appropriate. For both problems, a derivative sensitivity basis of rank 50 is computed via samples using matrix-free randomized algorithms,

$$\mathbb{E}_{M\sim\nu_M}\left[\delta\mathcal{F}^*(M)\delta\mathcal{F}(M)\right] \approx \frac{1}{n_{\text{sample}}}\sum_{j=1}^{n_{\text{sample}}}\delta\mathcal{F}^*(m_j)\delta\mathcal{F}(m_j)\,, \tag{38}$$

where $\delta\mathcal{F}^*(\cdot)$ denotes the adjoint of the derivative of $\mathcal{F}$. In both cases, $n_{\text{sample}} = 256$ samples are used, and the additional computational costs are taken into account when computing the offline cost of neural operators. The matrix-free actions of $\delta\mathcal{F}^*(M)\delta\mathcal{F}(M)$ only require solving linearized PDEs similar to that of the error correction problem and can be computed at training data sample points, thus ensuring a reasonable computational efficiency of the matrix-free action evaluations.

Remark 4. *Unlike [40], we do not employ the additional stochastic Newton optimizer [98, 99] since this is not typically used in neural operator learning. However, we observe substantially the performance improvement of neural operators trained with this optimizer. Moreover, unlike [73] this network does not include derivative approximations during the formulation of the operator learning problem and training, and the derivative information is only used in the architecture. For more information on the neural network architectures employed in the two numerical examples, see [39, 40].*

5.2. Software

The following software is used to implement the numerical examples: FEniCS [100] for finite element method, hIPPYlib [101] for prior and posterior sampling, and hIPPYflow [102] in combination with TensorFlow [103] for neural operator construction and training. We note that the derivative of the residual operator is implicitly derived and implemented using the automatic differentiation of weak forms supported by FEniCS and Unified Form Language (UFL) [104].

5.3. Inferring a coefficient field of a nonlinear reaction–diffusion problem

Here we introduce the setting for a demonstrative Bayesian inverse problem. The model is governed by an equilibrium reaction–diffusion equation with a nonlinear cubic reaction term. The model solutions are driven by the Dirichlet boundary conditions at the top and bottom of the square domain $\Omega = (0,1)^2$ and the coefficient field $\kappa : \Omega \to \mathbb{R}_+$:

$$-\nabla\cdot\kappa(\boldsymbol{x})\nabla u(\boldsymbol{x}) + u(\boldsymbol{x})^3 = 0, \qquad \boldsymbol{x}\in\Omega\,; \tag{39a}$$

$$\kappa(\boldsymbol{x})\nabla u(\boldsymbol{x})\cdot\boldsymbol{n} = 0, \qquad \boldsymbol{x}\in\Gamma_l\cup\Gamma_r\,; \tag{39b}$$

$$u(\boldsymbol{x}) = 1, \qquad \boldsymbol{x}\in\Gamma_t\,; \tag{39c}$$

$$u(\boldsymbol{x}) = 0, \qquad \boldsymbol{x}\in\Gamma_b\,, \tag{39d}$$

where Γ_t, Γ_r, Γ_b, and Γ_l denote the top, right, bottom, and left boundary of the domain, and $\boldsymbol{n}$ is the outward normal vector. We assume prior knowledge of an epistemically uncertain and spatially-varying coefficient field $K : \Omega \to \mathbb{R}_+$, following a log-normal prior distribution:

$$K = \exp(M)\,, \qquad M \sim \nu_M \coloneqq \mathcal{N}(0, \mathcal{C}_{\text{pr}})\,, \tag{40}$$

where M is the parameter random variable through which we represent the uncertainty in κ. The prior distribution ν_M is specified with a covariance constructed according to (6) with hyperparameters of $d = 2$, $\alpha = 0.08$, $\beta = 2$, and $\boldsymbol{\Theta} = \boldsymbol{I}$. They correspond to isotropic Gaussian random fields with a pointwise variance of approximately 1, correlation length of approximately 0.2, and small boundary artifacts. We visualize several samples of the prior as well as their corresponding coefficient fields and model solutions via the finite element method, to be specified later, in Figure 4.

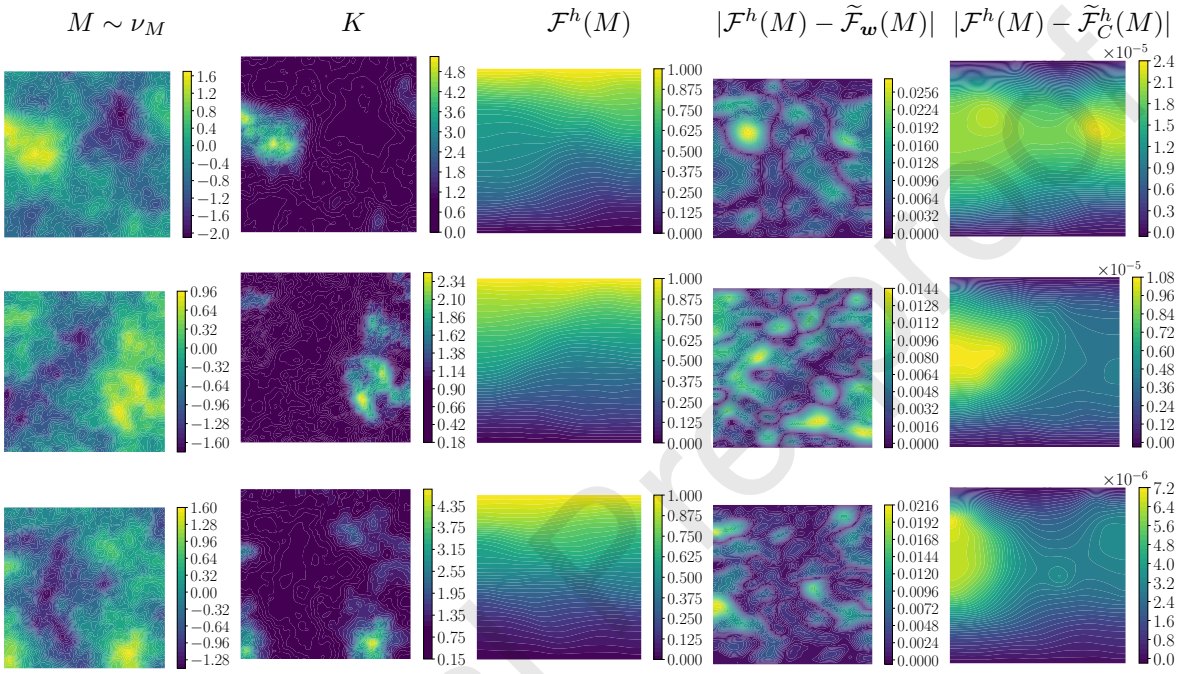


Figure 4: Visualization of the prior samples, model solutions, and neural operator performance with and without error correction for the reaction–diffusion problem introduced in Section 5.3. From left to right, we have (i) three Gaussian random fields, m_j, $j = 1, 2, 3$, sampled from the prior distribution ν_M with approximately a pointwise variance of 1 and correlation length of 20% of the domain size length, each occupies a row, (ii) the corresponding coefficient field samples, κ_j, defined by $\kappa_j = \exp(m_j)$, (iii) the corresponding finite element solutions of the reaction–diffusion problem, $\mathcal{F}^h(m_j)$, (iv) the absolute prediction errors at m_j for the *best performing* neural operator ($\sim 90\%$ accurate as shown in Figure 5), $|\mathcal{F}^h(m_j) - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m_j)|$, and (v) the absolute errors at m_j for the error-corrected predictions based on the best performing neural operator.

We consider the following parameter space, state space, and solution set for the nonlinear PDE problem above:

$$\begin{gathered} \mathcal{M} \coloneqq L^2(\Omega)\,, \quad \mathcal{U} \coloneqq H^1(\Omega)\,, \\ \mathcal{U}_0 \coloneqq \{u \in H^1(\Omega) : u|_{\Gamma_b} = 0 \text{ and } u|_{\Gamma_t} = 0\}\,, \quad \mathcal{V}_u \coloneqq \{u \in H^1(\Omega) : u|_{\Gamma_b} = 0 \text{ and } u|_{\Gamma_t} = 1\}\,, \end{gathered} \tag{41}$$

where the restriction to the boundary is defined with the trace operator. The function spaces are equipped with the Sobolev norm. The variational problem for the model is given by:

$$\begin{gathered} \text{Given } m \in \mathcal{M},\ \text{find } u \in \mathcal{V}_u \text{ such that} \\ \langle \mathcal{R}(u, m), v\rangle \coloneqq \int_\Omega \exp(m) \nabla u(\boldsymbol{x}) \cdot \nabla v(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} + \int_\Omega u(\boldsymbol{x})^3 v(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x} = 0\,, \quad \forall v \in \mathcal{U}_0\,. \end{gathered} \tag{42}$$

5.3.1. Numerical approximation and neural operator performance

The numerical evaluation of the forward operator, $\mathcal{F}^h$, sampling of the prior distribution, ν_M^h, and the residual-based error correction problem is implemented via the finite element method. In particular, the domain Ω is discretized with 64×64 cells of uniform linear and parabolic Lagrangian triangular elements that form the finite element spaces $\mathcal{M}^h \subset \mathcal{M}$ and $\mathcal{U}^h \subset \mathcal{U}$, respectively. The state finite element space has $d_u = 16641$ degrees of freedom, and the parameter finite element space has $d_m = 4225$. The variational problems of the model, prior sampling, and error correction are then approximated and solved in these finite element spaces. We employ the Newton iteration for solving the nonlinear reaction–diffusion problem, which takes an average of 2.5 iterations to converge for parameter samples generated from the prior distribution.

Applying neural operator construction and training specified in Section 5.1 to the reaction–diffusion problem, we produced 7 neural operators using increasing number of training samples, $n_{\text{train}} = 100$, 201, 403, 806, 1382, 3225, 6912, assuming $p = 2$ in the Bochner norm. In Figure 5, the accuracy of the trained neural operators at different numbers of training samples is shown. The accuracy number is computed according to (15) using 512 samples from the prior distribution that are unseen during training. The accuracy ceiling around 90% is reached at $n_{\text{train}} = 1382$.

For each trained neural operator, the accuracy for the error-corrected neural operators using the same 512 samples is also computed and shown in Figure 5. The error-corrected mappings for all 6 trained operators are close to 100% accurate. The visualization of absolution errors for the predictions by the best performing neural operator with $n_{\text{train}} = 6912$ and its error-corrected predictions at samples from the prior distribution are shown in Figure 4. We observe that the error correction step leads to a drop of maximum absolute pointwise error from the order of 10^{-2} to the order of 10^{-5}.

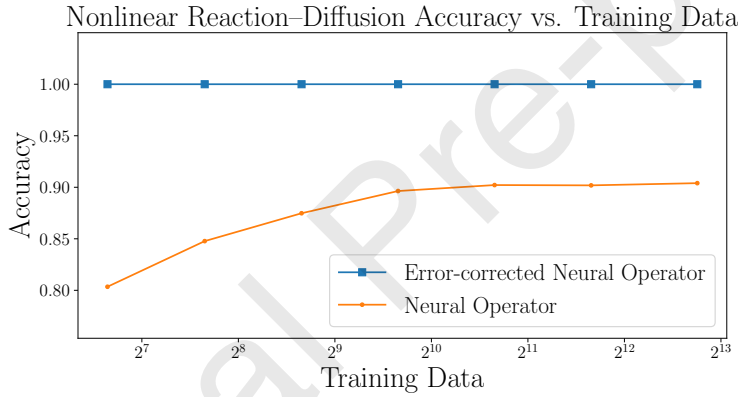


Figure 5: A study of the generalization accuracy, as defined in (15), of 7 neural operators trained using a varying number of training samples for the nonlinear reaction–diffusion problem. The neural operators are constructed using a derivative-informed projected ResNet; see Section 5.1 for its construction and training details. Generalization accuracy is computed using 512 data unseen during training. An empirical accuracy ceiling of $\sim 90\%$ is reached for the given neural network architecture.

Remark 5. *For both numerical examples in Sections 5.2 and 5.3, we additionally compute the $L^2(\Omega)$ generalization accuracy of the trained neural operators with visualizations presented in Appendix C. The $L^2(\Omega)$ generalization accuracy replaces $\|\cdot\|_{\mathcal{U}}$ in (15) with $\|\cdot\|_{L^2(\Omega)}$. To compute the $L^2(\Omega)$ generalization accuracy, one discards the terms involving spatial derivatives during the computation of the state norm $\|\cdot\|_{\mathcal{U}}$ in (15). In most existing studies of neural operators, this $L^2(\Omega)$ generalization accuracy is the only accuracy metric used for model validation and testing. However, the $L^2(\Omega)$ generalization accuracy does not necessarily reflect the accuracy for the operator learning problem (10) when the output space $\mathcal{U}$ is the solution space of PDEs with bounded derivatives.*

5.3.2. Bayesian inverse problem setting

We consider a set of synthetic observation data $\boldsymbol{y}^*$ generated according to the data model in (4) for the reaction–diffusion problem at a synthetic parameter fields m^*. It has distinctive curvatures generated using a Rosenbrock function. We visualize the synthetic parameter m^* as well as its corresponding coefficient field κ^* and model solution $u^* = \mathcal{F}^h(m^*)$ in Figure 6.

To complete the data model, we define an observation operator and a noise distribution. Here we consider a linear observation operator that extracts discrete observations of the model solution at a uniform grid of 10×10 points in the domain Ω. Let $\{\boldsymbol{x}_j\}_{j=1}^{100}$ denote the observation points. Given a function $u \in \mathcal{U}$, the observation operator $\boldsymbol{\mathcal{B}}(u) \in \mathbb{R}^{100}$ returns local averages of a given state around the observation points:

$$\boldsymbol{\mathcal{B}}(u) = \left[|B_r(\boldsymbol{x}_1)|^{-1} \int_{B_r(\boldsymbol{x}_1)} u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \quad \cdots \quad |B_r(\boldsymbol{x}_{100})|^{-1} \int_{B_r(\boldsymbol{x}_{100})} u(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \right], \tag{43}$$

where $B_r(\boldsymbol{x}_j)$ is a circle with a small radius $r > 0$ centered at the observation point and $|B_r(\boldsymbol{x}_j)|$ is the size of the circle. We assume the observation is corrupted with white noise, i.e., $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, with a standard deviation of $\sigma = 0.0073$ that is 1% of the maximum value in $\boldsymbol{\mathcal{B}}(u^*)$.

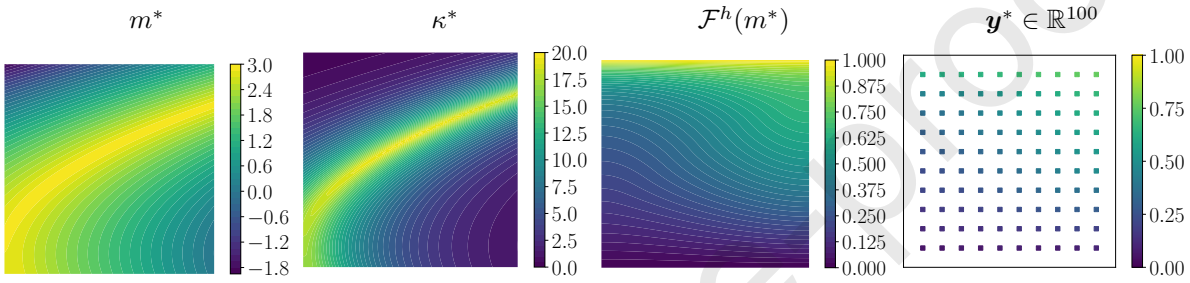


Figure 6: Visualization of the setting for a synthetic Bayesian inverse problem based on the nonlinear Poisson problem introduced in Section 5.3.2. From left to right, we have (i) the synthetic parameter field, m^*, defined using a Rosenbrock function, (ii) the corresponding synthetic coefficient field, κ^*, (iii) the finite element solution at m^*, $\mathcal{F}^h(m^*)$, (iv) the synthetic observed data, $\boldsymbol{y}^*$, extracted from locally averaged values of $\mathcal{F}^h(m^*)$ at a 10×10 grid of observation points, corrupted by a randomly sampled additive white noise with a standard deviation of 0.0073.

5.3.3. Posterior visualization and cost analysis

To visualize and compare posterior distributions with likelihood functions evaluated with the model via the finite element method, neural operators, and error-corrected neural operators, we generate samples from these posterior distributions via the pCN algorithm introduced in Section 2.3, and visualize their posterior predictive means of the coefficient field by sample average approximation,

$$\kappa_{\text{mean}} \approx \frac{1}{n_{\text{post}}} \sum_{j=1}^{n_{\text{post}}} \exp(m_j) \,, \tag{44}$$

where $\{m_j \in \mathcal{M}^h\}_{j=1}^{n_{\text{post}}}$ are posterior samples. For each posterior distribution, 8 MCMC chains are constructed with a mixing parameter of $\beta_{\text{pCN}} = 0.03$, and samples of total $n_{\text{post}} = 120,000$ are collected. While the mixing of the chains is seen to be rapid, a conservative burn-in rate of 25% is used. The average sample acceptance rate for MCMC sampling using the model is around 20%.

In Figure 7, we visualize the model-generated posterior predictive mean estimate of the coefficient field alongside the ones generated by the three best-performing neural operators. We observe that the estimates by these neural operators are unable to recover the distinctive curvature of the coefficient field captured by the model estimate, even though they are near the accuracy ceiling of neural operator training, where the accuracy increment is quickly diminishing with respect to the size of the training data. In Figure 8, we provide the same visualization of the estimates generated by the three neural operators with error correction.

We observe that they produce estimates that are much more consistently similar to the ones generated by the model.

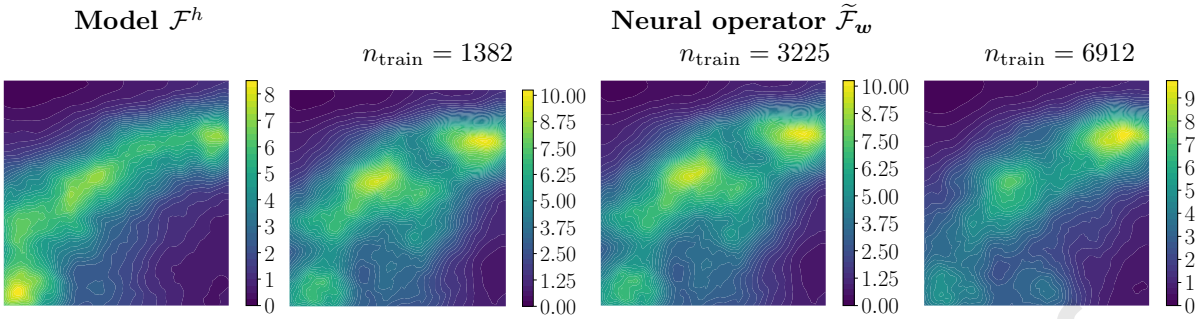


Figure 7: Visualization of posterior predictive mean estimates in (44) of the coefficient field κ for a synthetic Bayesian inverse problem introduced in Section 5.3.2. From left to right, we have the estimates by (i) the model via the finite element method, (2) the neural operator trained with $n_{\text{train}} = 1382$, (3) the neural operator trained with $n_{\text{train}} = 3225$, (4) and the neural operator trained with $n_{\text{train}} = 6912$. The accuracy of the neural operators is around 90% as shown in Figure 5.

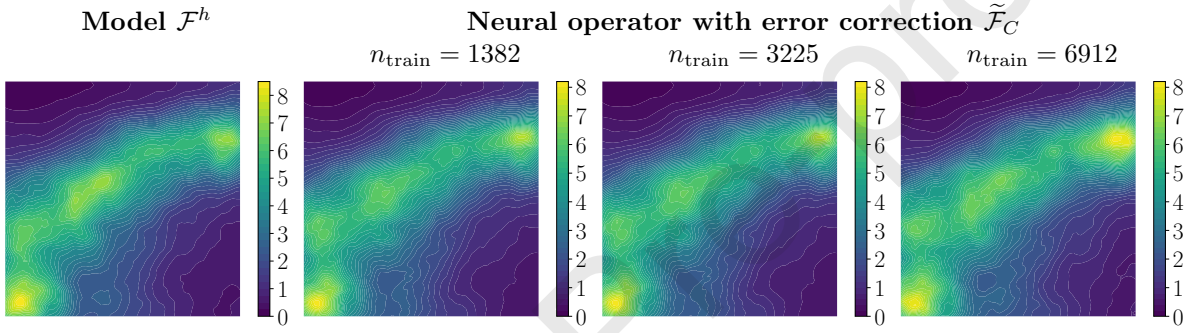


Figure 8: Visualization of posterior predictive mean estimates of the coefficient field in (44) for a synthetic Bayesian inverse problem introduced in Section 5.3.2. From left to right, we have the estimates by (i) the model via the finite element method, (2) the neural operator trained with $n_{\text{train}} = 1382$ with error correction, (3) the neural operator trained with $n_{\text{train}} = 3225$ with error correction, (4) and the neural operator trained with $n_{\text{train}} = 6912$ with error correction. The accuracy of the neural operators with error correction is close to 100% as shown in Figure 5.

In Figure 9, we visualize the observed and asymptotic speedups for the posterior sampling using the 7 trained neural operators with or without the error correction. As defined in Section 4.4, the asymptotic speedup assumes $n_{\text{chain}} \to \infty$ in the sense that a long Markov chain is generated or repetitive use of the trained neural operators in different problems. As a result, the offline costs of the neural operator construction and training are neglected. The asymptotic speedup of the error-corrected neural operators is about 2.5, which is the number of Newton iterations for solving the nonlinear problem, averaged over the posterior distribution. The asymptotic speedup of the neural operators is over two orders of magnitude. The observed speedups additionally account for the offline cost of reduced basis approximation, training data generation, and optimization. The finite n_{chain} used for the posterior sampling presented above is used to compute the observed speedups. As a result, the observed speedups for the neural operator decay substantially as the number of training data increases. We observe almost an order of magnitude drop of speedup for $n_{\text{train}} = 6912$ compared to $n_{\text{train}} = 100$. A similar decay is observed for the error-corrected neural operators, but the dominant cost remains the cost of solving the linear systems associated with the error-correction steps.

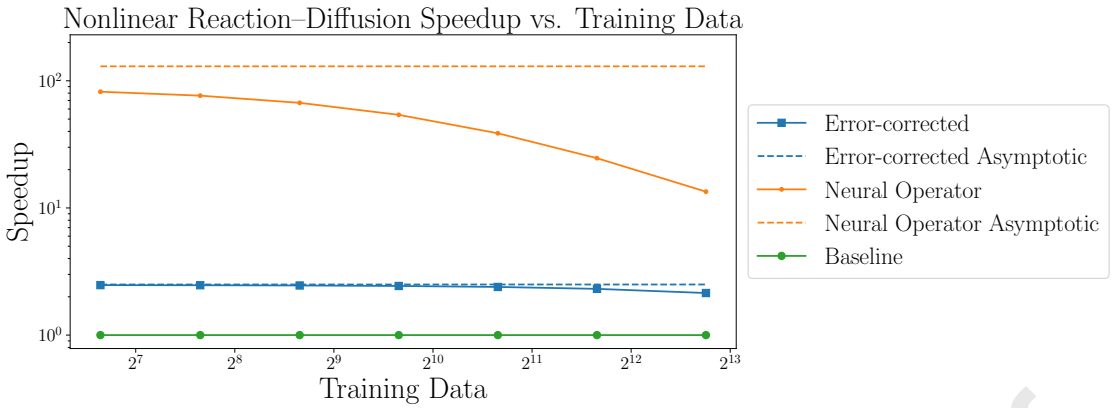


Figure 9: The observed and asymptotic speedups, as defined in Section 4.4, for the posterior sampling via pCN for the neural operators and the error-corrected neural operators for the nonlinear reaction–diffusion problem. The asymptotic speedups assume $n_{\text{chain}} \to \infty$, thus neglecting the offline cost of neural operator construction and training. The observed speedups additionally account for the offline cost and the total number of iterative solves within generated Markov chains used for the posterior visualization in Figure 7 and 8.

5.4. Hyperelastic material properties discovery

Here we introduce the physical and mathematical setting for hyperelastic material properties discovery. This problem has attracted many research interests for its essential role in various engineering and medical applications [105–108]. We consider an experimental scenario where a square thin film of a hyperelastic material is fixed on one edge, with a traction force applied on the opposite edge, leading to a deformation of the material. We attempt to infer the material properties as spatially-varying functions from noisy measurements of material displacement at discrete positions via Bayes' rule. The schematic of the problem setup is shown in Figure 10.

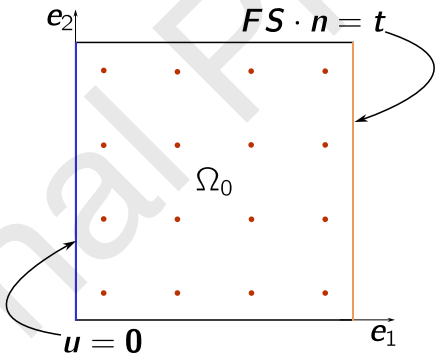


Figure 10: Schematics of a numerical experiment involving deformation of a hyperelastic material. Here the red dots are the points on the reference domain at which displacement (both components) is observed. On the left edge, the material is clamped so that the displacement is zero at this edge. Traction is prescribed as a function $\boldsymbol{t} = \boldsymbol{t}(\boldsymbol{X})$ on the right edge. The top and bottom edges are traction free.

5.4.1. A model for hyperelastic material deformation

Let $\Omega_0 = (0, 1)^2$ be a normalized unit square material domain for the material of interest under the thin film approximation, and $\boldsymbol{X} \in \Omega_0$ denote the material point. The current configuration of the material after deformation is represented by a map $\boldsymbol{\chi} : \Omega \to \mathbb{R}^2$. The material point $\boldsymbol{X}$ is mapped to a spatial point $\boldsymbol{x} = \boldsymbol{\chi}(\boldsymbol{X}) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X})$, where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{X})$ is the displacement of material points. Internal forces are developed within the body as material deforms relative to a given reference configuration. These internal forces depend on the underlying stored internal energy; for a hyperelastic material, there is a strain energy

density, $W = W(\boldsymbol{X}, \boldsymbol{C})$, as a function of material coordinate $\boldsymbol{X}$ and the right Cauchy–Green strain tensor $\boldsymbol{C} = \boldsymbol{F}^T\boldsymbol{F}$, $\boldsymbol{F} = \nabla\boldsymbol{\chi} = \boldsymbol{I} + \nabla\boldsymbol{u}$ being the deformation gradient. The stress in the reference configuration, $\boldsymbol{S} = \boldsymbol{S}(\boldsymbol{X}, \boldsymbol{C})$ is the *second Piola–Kirchhoff stress tensor* given by

$$\boldsymbol{S}(\boldsymbol{X}, \boldsymbol{C}) = 2\frac{\partial W(\boldsymbol{X}, \boldsymbol{C})}{\partial \boldsymbol{C}}\,. \tag{45}$$

We consider the model for neo-Hookean materials [109, 110] for which strain energy density function takes the form

$$W(\boldsymbol{X}, \boldsymbol{C}) = \frac{\mu(\boldsymbol{X})}{2}(\mathrm{tr}(\boldsymbol{C}) - 3) + \frac{\lambda(\boldsymbol{X})}{2}\left(\ln(J)\right)^2 - \mu(\boldsymbol{X})\ln(J)\,, \tag{46}$$

where $\mathrm{tr}(\boldsymbol{C})$ is the trace of the second order tensor $\boldsymbol{C}$, and J is the determinant of the deformation gradient $\boldsymbol{F}$. Here λ and μ are the so-called *Lamé parameters* which we assume to be related to Young's modulus of elasticity, E, and Poisson ratio, ν, as follows:

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}\,, \qquad \mu = \frac{E}{2(1+\nu)}\,. \tag{47}$$

In what follows, we assume prior knowledge of (i) Poisson ratio $\nu = 0.4$, and (ii) an epistemically uncertain and spatially-varying Young's modulus, $E : \Omega_0 \to \mathbb{R}_+$, that follows a log-normal prior distribution:

$$E = \exp(M) + E_L\,, \quad M \sim \nu_M \coloneqq \mathcal{N}(m_{\mathrm{pr}}, \mathcal{C}_{\mathrm{pr}})\,, \tag{48}$$

where $E_L > 0$ is a lower bound on the pointwise value of the random function, normalized to have a value of 1, and $m_{\mathrm{pr}} = 0.37$ is a constant over Ω_0. The Bayesian inverse problem aims to learn the material properties E through the parameter random field M from observed displacements via Bayes' rule.

The prior distribution of M is constructed according to (6) with hyperparameters of $d = 2$, $\alpha = 4/3$, $\beta = 0.12$, and $\boldsymbol{\Theta} = \boldsymbol{I}$. They correspond to Gaussian random fields with a pointwise variance of approximately 1, a correlation length of approximately 0.3, and small boundary artifacts. In Figure 11, we visualize several samples of the prior and their corresponding Young's modulus fields.

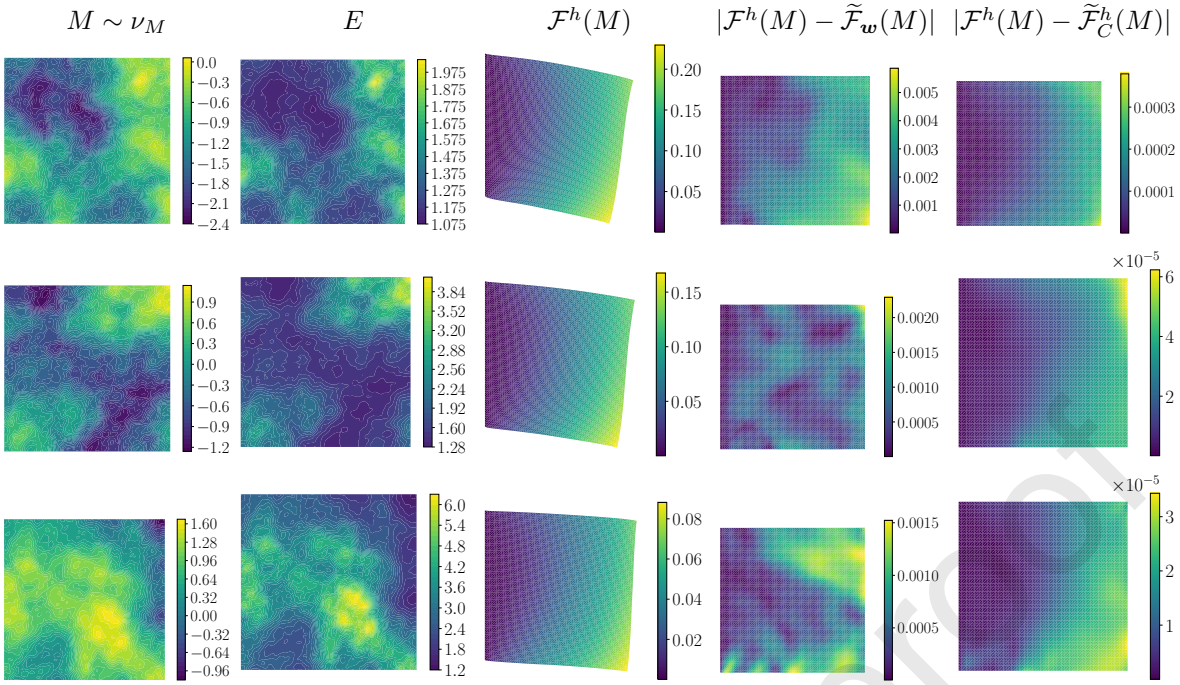


Figure 11: Visualization of the prior distribution, model solutions, and neural operator performance with and without error correction for the hyperelastic material deformation problem introduced in Section 5.4.1. From left to right, we have (i) three Gaussian random fields, m_j, $j = 1, 2, 3$, sampled from the prior distribution ν_M with approximately a pointwise variance of 1 and a correlation length of 30% domain side length, each occupies a row; (ii) the corresponding Young's modulus samples, E_j, defined by $E_j = \exp(m_j) + 1$, (viii) the corresponding finite element solutions of the current configurations, $\mathcal{F}^h(m_j)$, (iv) the absolute prediction errors at m_j for the best performing neural operator $\widetilde{\mathcal{F}}_{\boldsymbol{w}}$ ($\sim 95\%$ accurate as shown in Figure 12), $|\mathcal{F}^h(m_j) - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(m_j)|$, and (v) the absolute errors at m_j for the error-corrected predictions based on the best performing neural operator.

Assuming a quasi-static model in which external forces are applied very slowly so that dependence of displacement on time can be ignored and negligible body forces, the balance of linear momentum leads to the following nonlinear PDE for a given parameter $m : \Omega_0 \to \mathbb{R}$.

$$\nabla \cdot (\boldsymbol{F}(\boldsymbol{X})\boldsymbol{S}(\boldsymbol{X}, m(\boldsymbol{X}), \boldsymbol{C}(\boldsymbol{X}))) = \boldsymbol{0}\,, \qquad \boldsymbol{X} \in \Omega_0\,; \tag{49a}$$

$$\boldsymbol{u} = \boldsymbol{0}\,, \qquad \boldsymbol{X} \in \Gamma_l\,; \tag{49b}$$

$$\boldsymbol{F}(\boldsymbol{X})\boldsymbol{S}(\boldsymbol{X}, m(\boldsymbol{X}), \boldsymbol{C}(\boldsymbol{X})) \cdot \boldsymbol{n} = \boldsymbol{0}\,, \qquad \boldsymbol{X} \in \Gamma_t \cup \Gamma_b; \tag{49c}$$

$$\boldsymbol{F}(\boldsymbol{X})\boldsymbol{S}(\boldsymbol{X}, m(\boldsymbol{X}), \boldsymbol{C}(\boldsymbol{X})) \cdot \boldsymbol{n} = \boldsymbol{t}(\boldsymbol{X})\,, \qquad \boldsymbol{X} \in \Gamma_r\,. \tag{49d}$$

Here, Γ_t, Γ_r, Γ_b, and Γ_l denote the top, right, bottom, and left boundary of the material domain, and $\boldsymbol{t}$ is the traction given by

$$\boldsymbol{t}(\boldsymbol{X}) = a \exp\left(-\frac{|X_2 - 0.5|^2}{b}\right) \boldsymbol{e}_1 + c\left(1 + \frac{X_2}{d}\right) \boldsymbol{e}_2 \tag{50}$$

with $a = 0.06$, $b = 4$, $c = 0.03$, and $d = 10$. The applied traction on the right side combines shear and tensile forces.

We consider the following parameter space, state space, and solution set for the nonlinear PDE problem above:

$$\mathcal{M} \coloneqq L^2(\Omega_0)\,, \quad \mathcal{U} \coloneqq H^1(\Omega_0; \mathbb{R}^2)\,, \quad \mathcal{U}_0 = \mathcal{V}_u \coloneqq \{\boldsymbol{u} \in H^1(\Omega_0; \mathbb{R}^2) : \boldsymbol{u}|_{\Gamma_l} = \boldsymbol{0}\}\,, \tag{51}$$

where the restriction to the boundary is defined with the trace operator. The function spaces are equipped

with the Sobolev norm. The variational problem for the experimental scenario is:

$$\text{Given } m \in \mathcal{M}, \text{ find } \boldsymbol{u} \in \mathcal{V}_u \text{ such that}$$
$$\langle \mathcal{R}(\boldsymbol{u}, m), v\rangle \coloneqq \int_{\Omega_0} \boldsymbol{F}\boldsymbol{S}(\boldsymbol{X}, m(\boldsymbol{X})\boldsymbol{C}(\boldsymbol{X}) : \nabla \boldsymbol{v} \, \mathrm{d}\boldsymbol{X} - \int_{\Gamma_t} \boldsymbol{t} \cdot \boldsymbol{v} \, \mathrm{d}\boldsymbol{X} = 0\,, \quad \forall \boldsymbol{v} \in \mathcal{U}_0\,. \tag{52}$$

In Figure 11, we show the model predictions via the finite element method, to be specified in the following subsection, of the current configurations at random samples from the prior.

5.4.2. Numerical approximation and neural operator performance

The numerical evaluation of the forward operator, $\mathcal{F}^h$, sampling of the prior distribution, ν_M^h, and the residual-based error correction problem is implemented via the finite element method. In particular, the domain Ω_0 is discretized with 64×64 cells of uniform linear Lagrangian triangular elements, which forms finite element spaces $\mathcal{M}^h \subset \mathcal{M}$ and $\mathcal{U}^h \subset \mathcal{U}$ with 8450 and 4225 degrees of freedom, respectively. The variational problems of the model, prior sampling, and error correction are then approximated and solved in these finite element spaces. We employ the Newton iteration for solving the hyperelastic material deformation problem, which takes on average 8 iterations to converge for parameter samples generated from the prior distribution.

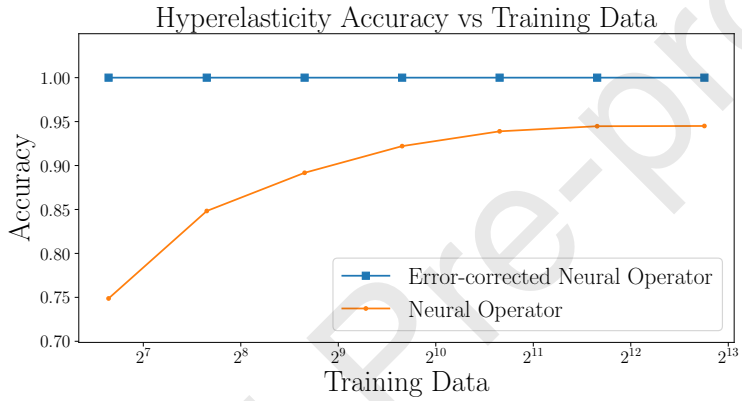


Figure 12: A study of the generalization accuracy, as defined in (15), of 7 neural operators trained using a varying number of training samples for the hyperelastic material deformation problem. The neural operators are constructed using a derivative-informed projected ResNet; see Section 5.1 for details on its construction and training. Generalization accuracy is computed using 512 data unseen during training. An empirical accuracy ceiling of $\sim$ 95% is reached for the given neural network architecture.

Applying neural operator construction and training specified in Section 5.1 to the hyperelastic material deformation problem, we produced 7 neural operators using a increasing number of training samples, $n_{\text{train}} =$ 100, 201, 403, 806, 1612, 3225, 6912, assuming $p = 2$ in the Bochner norm. In Figure 12, the accuracy of the trained neural operators at different numbers of training samples is shown. The accuracy number is computed according to (15) using 512 samples from the prior distribution that are unseen during training. The accuracy ceiling around 95% is reached at $n_{\text{train}} = 1612$, and a slight drop in accuracy is observed at $n_{\text{train}} = 6912$ samples.

For each trained neural operator, the accuracy for the error-corrected neural operator using the same 512 samples is also computed and shown in Figure 12. The error-corrected neural operator mapping for all 7 trained operators is close to 100% accurate.

The visualization of absolution errors for the predictions by the best performing neural operator with $n_{\text{train}} = 3225$ and its error-corrected predictions at samples from the prior distribution is shown in Figure 11. We observe that the error correction step leads to a drop of maximum absolute pointwise error from the order of 10^{-3} to the order of 10^{-4}. For prior samples that lead to small deformations, an additional order of magnitude drop in the maximum absolute pointwise error is observed.

5.4.3. Bayesian inverse problem setting

We consider a set of synthetic observation data $\boldsymbol{y}^*$ generated according to the data model in (4) for hyperelastic material deformation at a synthetic parameter field m^* generated by a sum of three Gaussian bumps with different diagonal covariance and weights. This parameter corresponds to a synthetic Young's modulus field of E^* that may represent a mixture of a small amount of spatially concentrated stiffer materials and a large amount of softer materials, e.g., cancer mass enclosed in healthy tissue. We visualize the synthetic parameter m^*, corresponding Young's modulus E^*, and displacement field $\boldsymbol{u}^* = \mathcal{F}^h(m^*)$ in Figure (13).

To complete the data model, we define an observation operator and a noise distribution. We, again, consider a linear observation operator that extracts discrete observations of the displacement vector at a uniform grid of 10×10 points in the reference domain Ω_0. This form of the observation operator is compatible with image analysis techniques such as digital image correlation; see, e.g., [111–117]. Let $\{\boldsymbol{X}_j\}_{j=1}^{100}$ denote the observation points. Given a displacement field $\boldsymbol{u} \in \mathcal{U}$, the observation operator $\boldsymbol{\mathcal{B}}(\boldsymbol{u}) \in \mathbb{R}^{2\times100}$ returns local averages of the displacements around the observation points:

$$\boldsymbol{\mathcal{B}}(\boldsymbol{u}) = \left[|B_r(\boldsymbol{X}_1)|^{-1} \int_{B_r(\boldsymbol{X}_1)} \boldsymbol{u}(\boldsymbol{X}) \, \mathrm{d}\boldsymbol{X} \quad \cdots \quad |B_r(\boldsymbol{X}_{100})|^{-1} \int_{B_r(\boldsymbol{X}_{100})} \boldsymbol{u}(\boldsymbol{X}) \, \mathrm{d}\boldsymbol{X} \right]. \tag{53}$$

We assume the observation is corrupted with white noise, i.e., $\boldsymbol{N} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, with a standard deviation of $\sigma = 0.0089$ that is 1% of the maximum values in $\boldsymbol{\mathcal{B}}(\boldsymbol{u}^*)$.

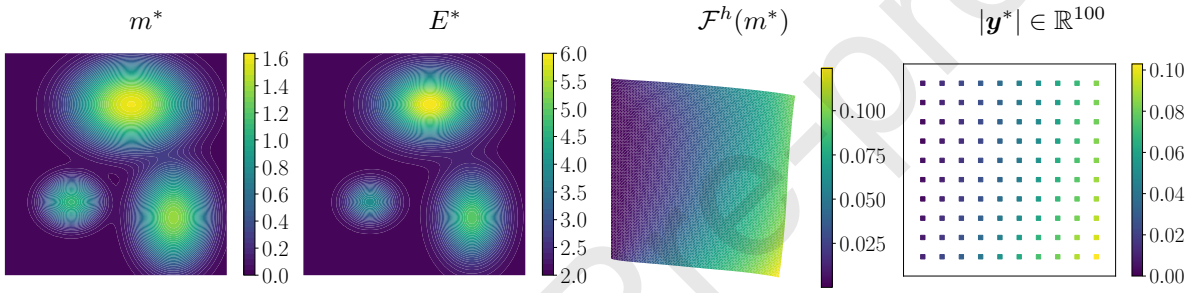


Figure 13: Visualization of a synthetic Bayesian inverse problem setting based on the hyperelastic material deformation introduced in Section 5.4.3. From left to right, we have (i) the synthetic parameter field, m^*, defined as the sum of three Gaussian bumps with different weights and diagonal covariances, (ii) the corresponding synthetic Young's modulus field, E^*, (iii) the finite element solution at m^*, $\mathcal{F}^h(m^*)$, (iv) the synthetic observed data, $\boldsymbol{y}^*$, extracted from locally averaged values of $\mathcal{F}^h(m^*)$ at a 10×10 grid of observation points, corrupted by a randomly sampled additive white noise with a standard deviation of 0.0089.

5.4.4. Posterior visualization and cost analysis

To visualize and compare posterior distributions defined with likelihood functions evaluated using the model via the finite element method, neural operators, and error-corrected neural operators, we generate samples from these posterior distributions via the pCN algorithms introduced in Section 2.3, and visualize the posterior predictive mean of Young's modulus by sample average approximation,

$$E_{\text{mean}} \approx \frac{1}{n_{\text{post}}} \sum_{j=1}^{n_{\text{post}}} (\exp(m_j) + 1), \tag{54}$$

where $\{m_j \in \mathcal{M}^h\}_{j=1}^{n_{\text{post}}}$ are the posterior samples. For each posterior distribution, 8 MCMC chains are constructed with a mixing parameter of $\beta_{\text{pCN}} = 0.05$ and collect, in total, posterior samples of $n_{\text{post}} = 120,000$. While the mixing of the chains is rapid, a conservative burn-in rate of 25% is used. The average sample acceptance ratio for MCMC sampling using the model is around 10%.

In Figure 14, we visualize the model-generated posterior predictive mean estimate of Young's modulus alongside the ones generated by the three best-performing neural operators. We observe that the estimate by the neural operator with $n_{\text{train}} = 1612$ completely misplaced two of the bumps clearly visible in the model

estimate. A doubling of training data and a slight increase of accuracy from $n_{\text{train}} = 1612$ to $n_{\text{train}} = 3225$ leads to a visually significant increase in the accuracy of the neural operator estimate. The neural operator with $n_{\text{train}} = 3225$ is able to qualitatively recover most features in the model estimate, with some difficulty in capturing the shape of the top bump and the location of the lower left bump. However, a doubling of training data from $n_{\text{train}} = 3225$ to $n_{\text{train}} = 6912$ in an attempt to further improve this result leads to a slight drop in the accuracy of the neural operator and a significant decrease in accuracy of the estimate. This observation reflects the unreliability of approximation error reduction of neural operators and thus the unreliability of the surrogate modeling approach for inverse problems based solely on trained neural operators.

In Figure 15, we provide the same visualization for the estimates produced by the same neural operators but with error correction. We observe that they generate estimates that are qualitatively much similar to the model-generated estimate, where all three bumps are captured with positions and magnitudes matching that of the model. These estimates are relatively consistent across the three neural operators, even though the three neural operators show drastically different accuracy in their estimates, demonstrating the robustness of our approach for error reduction of neural operators in Bayesian inverse problems.

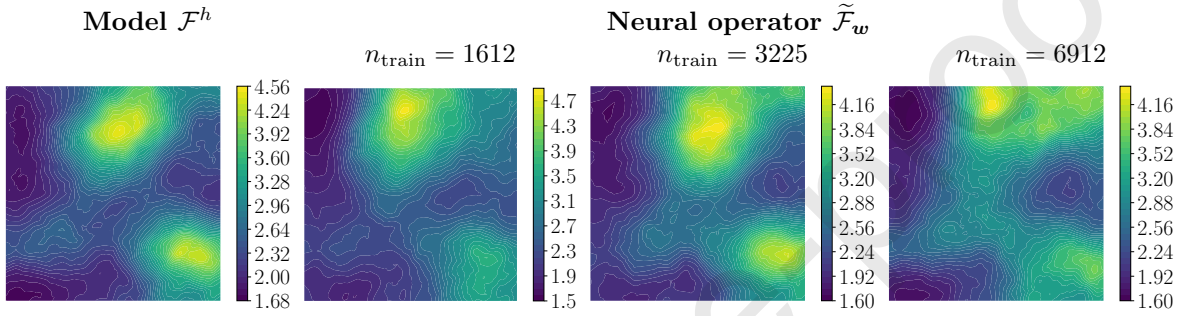


Figure 14: Visualization of the posterior predictive estimates of Young's modulus field in (54) for a synthetic Bayesian inverse problem introduced in Section 5.4.3. From left to right, we have the estimates by (i) the model via the finite element method, (2) the neural operator trained with $n_{\text{train}} = 1612$, (3) the neural operator trained with $n_{\text{train}} = 3225$, (4) and the neural operator trained with $n_{\text{train}} = 6912$.

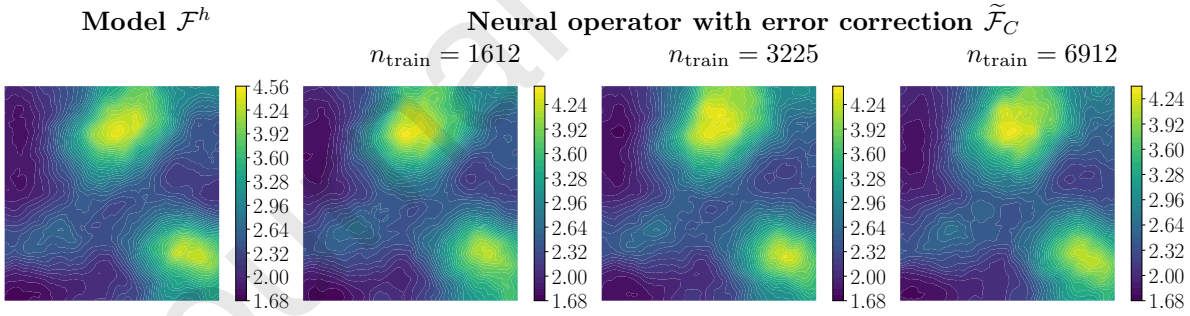


Figure 15: Visualization of posterior the predictive mean estimates of Young's modulus field in (54) for a synthetic Bayesian inverse problem introduced in Section 5.4.3. From left to right, we have the estimates by (i) the model via the finite element method, (2) the neural operator trained with $n_{\text{train}} = 1612$ with error correction, (3) the neural operator trained with $n_{\text{train}} = 3225$ with error correction, (4) and the neural operator trained with $n_{\text{train}} = 6912$ with error correction.

In Figure 16, we visualize the observed and asymptotic speedups for the posterior sampling using the 7 trained neural operators with or without the error correction. Similarly to the computational cost analysis for the reaction–diffusion problem, the asymptotic speedups here assume $n_{\text{chain}} \to \infty$, and the offline cost of the neural operator construction and training is neglected. The asymptotic speedup of the error-corrected

neural operators is about one order of magnitude, which is the number of Newton iterations for solving the nonlinear problem, averaged over the posterior distribution. The asymptotic speedup of the neural operators is over two orders of magnitude. The observed speedups additionally account for the offline cost of reduced basis approximation, training data generation, and optimization, and included a finite n_{chain} used for the posterior sampling presented above. The resulting observed speedups for the neural operators decay substantially as the number of training data increases and nearly drops an order of magnitude from $n_{\text{train}} = 100$ to $n_{\text{train}} = 6912$. The observed speedup at the latter is in the same order as the asymptotic speedup of the error-correction neural operators. A similar decay of the observed speedups is seen for the error-corrected neural operators. The dominant cost remains the cost of solving the linear systems associated with the error correction steps.

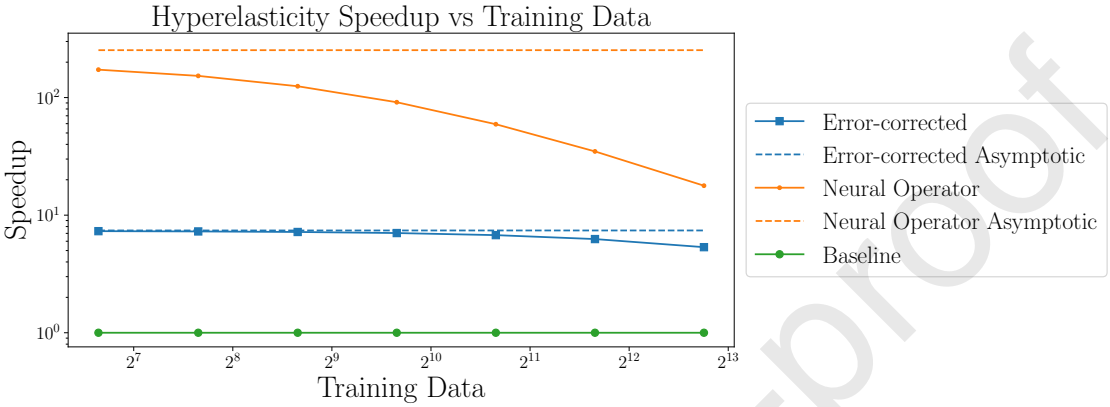


Figure 16: The observed and asymptotic speedups, as defined in Section 4.4, for the posterior sampling via pCN for both the neural operators and the error-corrected neural operators. The asymptotic speedups assume $n_{\text{chain}} \to \infty$, thus neglecting the offline cost of neural operator construction and training. The observed speedups additionally account for the offline costs and the total number of iterative solves within generated Markov chains used for the posterior visualization in Figure 14 and 15.

6. Conclusion and Outlook

In this work, we presented a residual-based error correction strategy for the reliable deployment of trained neural operators as a surrogate of the parameter-to-state map defined by nonlinear PDEs for accelerating infinite-dimensional Bayesian inverse problems. The strategy is motivated by the *a posterior* error estimation techniques applied to estimating modeling error. For a trained neural operator, we utilized its prediction at a given parameter to formulate and solve a linear variational problem, or an error-correction problem, based on the PDE residual and its derivative. The resulting solution can lead to a quadratic reduction of local error due to the equivalency of solving the error-correction problem and generation one Newton step under some mild conditions. We show that this can be extended to a global reduction, i.e., over the prior distribution, of approximation errors for well-trained neural operators.

The proposed strategy addresses an important issue facing the application of operator learning using neural networks: the unreliability of neural operator performance improvement or approximation error reduction via training. By deriving an *a priori* error bound, we demonstrate that the approximation error of trained neural operators controls the error in the posterior distributions of Bayesian inverse problems when used as surrogate parameter-to-state maps in likelihood functions. Furthermore, the resulting error in the posterior distributions may be significantly magnified for Bayesian inverse problems with uninformative prior distributions, high-dimensional data, small noise corruption, and inadequate models. In situations where the approximation error of a neural operator is persistent and not easily reduced to an acceptable level for the target Bayesian inverse problem via training, our proposed strategy offers an effective alternative of constructing error-corrected neural operators for achieving these accuracy requirements, all while retaining substantial computational speedups by leveraging the predictability of trained neural operators. For models

governed by large-scale highly nonlinear PDEs, where the costs of evaluating trained neural operators are relatively negligible, this strategy provides a great computational speedup for posterior characterization, which is approximately the expected number of iterative linear solves within a nonlinear PDE solve at parameters sampled from the posterior distribution,

We demonstrate the advantages of our proposed strategy through two numerical examples: inference of an uncertain coefficient field in an equilibrium nonlinear reaction–diffusion equation and hyperelastic material properties discovery. For both problems, the performance of trained neural operators shows diminishing improvement from $< 80\%$ accuracy with a small number of training samples to their empirical accuracy ceilings of $< 95\%$ with a much larger number of training samples, while the performance of error-corrected neural operators is consistent with near 100% accuracy. The visualization of the posterior predictive mean estimates suggests that trained neural operators alone as surrogates of the parameter-to-state map cannot reliably recover distinctive features of the physical parameters that are shown to be retrievable via inference with full-scale model solves, while the error-corrected neural operators as surrogates show consistency in its ability to recover these features.

Many other outer-loop problems in engineering, sciences, and medicine, such as optimal control and design under uncertainties, can also benefit from fast and accurate full-state predictions using error-corrected neural operators. Moreover, many of these outer-loop problems, including Bayesian inverse problems, are based on physical systems modeled by nonlinear time-evolving PDEs. Although the combination of residual-based error correction and neural operators is generally applicable to these outer-loop problems and models, thorough theoretical and numerical investigations are needed to understand its range of applications and limitations in these settings.

From our theoretical and numerical analysis of the computational speedups of error-corrected neural operators, it is clear that the dominant computational cost of the proposed strategy is the accumulated cost of solving the full linear error correction problem at each neural operator prediction. This burden can also be mitigated by incorporating approximation techniques such as model order reduction, developing problem-specific fast solvers, or even training a neural operator for the error correction problem. Other possible applications of the error-corrected neural operators are in multilevel or multifidelity methods for outer-loop problems.

Acknowledgement

The work was partially supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under awards DE-SC0019303, DE-SC0021239, and DE-SC0023171, and the Air Force Office of Scientific Research under award FA9550-21-1-0084. The authors thank Peng Chen, Jiaqi Li, and Barbara Wohlmuth for technical suggestions that helped improve this work. The authors thank Michael B. Giles for a detailed review of the manuscript and for pointing out mistakes in the original pre-print version.

References

[1] E. T. Jaynes, Probability theory: The logic of science, Cambridge University Press, 2003.

[2] L. Biegler, G. Biros, O. Ghattas, M. Heinkenschloss, D. Keyes, B. Mallick, Y. Marzouk, L. Tenorio, B. van Bloemen Waanders, K. Willcox (Eds.), Large-Scale Inverse Problems and Quantification of Uncertainty, John Wiley & Sons, Ltd, 2010.

[3] J. T. Oden, I. Babuška, D. Faghihi, Predictive computational science: Computer predictions in the presence of uncertainty, in: Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd, 2nd edition, 2017, pp. 1–26.

[4] O. Ghattas, K. Willcox, Learning physics-based models from data: Perspectives from inverse problems and model reduction, Acta Numerica 30 (2021) 445–554.

[5] J. Wang, N. Zabaras, Using Bayesian statistics in the estimation of heat source in radiation, International Journal of Heat and Mass Transfer 48 (2005) 15–29.

[6] T. Isaac, N. Petra, G. Stadler, O. Ghattas, Scalable and efficient algorithms for the propagation of uncertainty from data through inference to prediction for large-scale problems, with application to flow of the Antarctic ice sheet, Journal of Computational Physics 296 (2015) 348–368.

[7] H. Zhu, S. Li, S. Fomel, G. Stadler, O. Ghattas, A Bayesian approach to estimate uncertainty for full-waveform inversion using a priori information from depth migration, Geophysics 81 (2016) R307–R323.

[8] A. Alghamdi, M. A. Hesse, J. Chen, O. Ghattas, Bayesian poroelastic aquifer characterization from InSAR surface deformation data. Part I: Maximum a posteriori estimate, Water Resources Research 56 (2020).

[9] P. Chen, K. Wu, O. Ghattas, Bayesian inference of heterogeneous epidemic models: Application to COVID-19 spread accounting for long-term care facilities, Computer Methods in Applied Mechanics and Engineering 385 (2021).

[10] B. Liang, J. Tan, L. Lozenski, D. A. Hormuth, T. E. Yankeelov, U. Villa, D. Faghihi, Bayesian inference of tissue heterogeneity for individualized prediction of glioma growth, 2022.

[11] C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, Society for Industrial and Applied Mathematics, 1995.

[12] T. Cui, K. J. Law, Y. M. Marzouk, Dimension-independent likelihood-informed MCMC, Journal of Computational Physics 304 (2016) 109–137.

[13] P. G. Constantine, C. Kent, T. Bui-Thanh, Accelerating Markov chain Monte Carlo with active subspaces, SIAM Journal on Scientific Computing 38 (2016) A2779–A2805.

[14] A. M. Stuart, J. Voss, P. Wilberg, Conditional path sampling of SDEs and the Langevin MCMC method, Communications in Mathematical Sciences 2 (2004) 685–697.

[15] J. Martin, L. C. Wilcox, C. Burstedde, O. Ghattas, A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion, SIAM Journal on Scientific Computing 34 (2012) A1460–A1487.

[16] T. Bui-Thanh, M. Girolami, Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo, Inverse Problems 30 (2014) 114014.

[17] T. Bui-Thanh, O. Ghattas, J. Martin, G. Stadler, A computational framework for infinite-dimensional Bayesian inverse problems part I: The linearized case, with application to global seismic inversion, SIAM Journal on Scientific Computing 35 (2013) A2494–A2523.

[18] C. Schillings, B. Sprungk, P. Wacker, On the convergence of the Laplace approximation and noise-level-robustness of Laplace-based Monte Carlo methods for Bayesian inverse problems, Numerische Mathematik 145 (2020) 915–971.

[19] C. Schillings, C. Schwab, Sparse, adaptive Smolyak quadratures for Bayesian inverse problems, Inverse Problems 29 (2013).

[20] R. N. Gantner, C. Schwab, Computational higher order quasi-Monte Carlo integration, in: R. Cools, D. Nuyens (Eds.), Monte Carlo and Quasi-Monte Carlo Methods, Springer International Publishing, Cham, 2016, pp. 271–288.

[21] M. Parno, T. Moselhy, Y. Marzouk, A multiscale strategy for Bayesian inference using transport maps, SIAM/ASA Journal on Uncertainty Quantification 4 (2016) 1160–1190.

[22] P. Chen, K. Wu, J. Chen, T. O'Leary-Roseberry, O. Ghattas, Projected Stein variational Newton: A fast and scalable Bayesian inference method in high dimensions, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019.

[23] J. Zech, Y. Marzouk, Sparse approximation of triangular transports, Part II: The infinite-dimensional case, Constructive Approximation 55 (2022) 987–1036.

[24] Y. Wang, P. Chen, W. Li, Projected Wasserstein gradient descent for high-dimensional Bayesian inference, 2021.

[25] Y. M. Marzouk, H. N. Najm, L. A. Rahn, Stochastic spectral methods for efficient Bayesian solution of inverse problems, Journal of Computational Physics 224 (2007) 560–586.

[26] Y. M. Marzouk, H. N. Najm, Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, Journal of Computational Physics 228 (2009) 1862–1902.

[27] D. Galbally, K. Fidkowski, K. Willcox, O. Ghattas, Non-linear model reduction for uncertainty quantification in large-scale inverse problems, International Journal for Numerical Methods in Engineering 81 (2010) 1581–1608.

[28] C. Lieberman, K. Willcox, O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, SIAM Journal on Scientific Computing 32 (2010) 2523–2542.

[29] T. Cui, Y. M. Marzouk, K. E. Willcox, Data-driven model reduction for the Bayesian solution of inverse problems, International Journal for Numerical Methods in Engineering 102 (2015) 966–990.

[30] T. J. Dodwell, C. Ketelsen, R. Scheichl, A. L. Teckentrup, A hierarchical multilevel Markov Chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow, SIAM/ASA Journal on Uncertainty Quantification 3 (2015) 1075–1108.

[31] A. L. Teckentrup, P. Jantsch, C. G. Webster, M. Gunzburger, A multilevel stochastic collocation method for partial differential equations with random input data, SIAM/ASA Journal on Uncertainty Quantification 3 (2015) 1046–1074.

[32] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Review 60 (2018) 550–591.

[33] K. Bhattacharya, B. Hosseini, N. B. Kovachki, A. M. Stuart, Model reduction and neural networks for parametric PDEs, SMAI Journal of Computational Mathematics, Volume 7 (2021).

[34] S. Fresca, A. Manzoni, POD-DL-ROM: Enhancing deep learning–based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition, Computer Methods in Applied Mechanics and Engineering 388 (2022) 114181.

[35] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces, arXiv preprint arXiv:2108.08481 (2021).

[36] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, International Conference on Learning Representations (2021).

[37] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, Neural Information Processing Systems (2020).

[38] L. Lu, P. Jin, G. Pang, G. E. Karniadakis, DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, Nature Machine Intelligence (2021).

[39] T. O'Leary-Roseberry, U. Villa, P. Chen, O. Ghattas, Derivative-informed projected neural networks for high-dimensional

parametric maps governed by PDEs, Computer Methods in Applied Mechanics and Engineering 388 (2022) 114199.
[40] T. O'Leary-Roseberry, X. Du, A. Chaudhuri, J. R. Martins, K. Willcox, O. Ghattas, Learning high-dimensional parametric maps via reduced basis adaptive residual networks, arXiv preprint arXiv:2112.07096 (2021).
[41] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.
[42] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, Science advances 7 (2021) eabi8605.
[43] J. Yu, L. Lu, X. Meng, G. E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, Computer Methods in Applied Mechanics and Engineering 393 (2022) 114823.
[44] J. Hesthaven, S. Ubbiali, Non-intrusive reduced order modeling of nonlinear problems using neural networks, Journal of Computational Physics 363 (2018) 55–78.
[45] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, arXiv preprint arXiv:2111.03794 (2021).
[46] M. Järvenpää, M. U. Gutmann, A. Vehtari, P. Marttinen, Parallel Gaussian Process Surrogate Bayesian Inference with Noisy Likelihood Evaluations, Bayesian Analysis 16 (2021) 147 – 178.
[47] I. Babuška, M. Suri, The p and h-p versions of the finite element method, basic principles and properties, SIAM Review 36 (1994) 578–632.
[48] L. Demkowicz, Computing with hp-ADAPTIVE FINITE ELEMENTS: Volume 1 One and Two Dimensional Elliptic and Maxwell Problems, Chapman and Hall/CRC, 1st edition, 2006.
[49] O. G. Ernst, A. Mugler, H. J. Starkloff, E. Ullmann, On the convergence of generalized polynomial chaos expansions, ESAIM: Mathematical Modelling and Numerical Analysis 46 (2012) 317–339.
[50] P. Chen, A. Quarteroni, G. Rozza, Reduced basis methods for uncertainty quantification, SIAM/ASA Journal on Uncertainty Quantification 5 (2017) 813–869.
[51] M. De Hoop, D. Z. Huang, E. Qian, A. M. Stuart, The cost-accuracy trade-off in operator learning with neural networks, arXiv preprint arXiv:2203.13181 (2022).
[52] P. K. Jha, J. T. Oden, Goal-oriented a-posteriori estimation of model error as an aid to parameter estimation, Journal of Computational Physics 470 (2022) 111575.
[53] J. T. Oden, S. Prudhomme, Goal-oriented error estimation and adaptivity for the finite element method, Computers & mathematics with applications 41 (2001) 735–756.
[54] J. T. Oden, S. Prudhomme, Estimation of modeling error in computational mechanics, Journal of Computational Physics 182 (2002) 496–515.
[55] S. Prudhomme, J. T. Oden, On goal-oriented error estimation for elliptic problems: Application to the control of pointwise errors, Computer Methods in Applied Mechanics and Engineering 176 (1999) 313–331.
[56] S. Prudhomme, J. T. Oden, Computable error estimators and adaptive techniques for fluid flow problems, in: Error estimation and adaptive discretization methods in computational fluid dynamics, Springer, 2003, pp. 207–268.
[57] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, Acta numerica 10 (2001) 1–102.
[58] M. B. Giles, E. Süli, Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality, Acta numerica 11 (2002) 145–236.
[59] N. A. Pierce, M. B. Giles, Adjoint recovery of superconvergent functionals from PDE approximations, SIAM review 42 (2000) 247–264.
[60] S. R. Arridge, J. P. Kaipio, V. Kolehmainen, M. Schweiger, E. Somersalo, T. Tarvainen, M. Vauhkonen, Approximation errors and model reduction with an application in optical diffusion tomography, Inverse Problems 22 (2006) 175–195.
[61] A. Manzoni, S. Pagani, T. Lassila, Accurate solution of Bayesian inverse uncertainty quantification problems combining reduced basis methods and reduction error models, SIAM/ASA Journal on Uncertainty Quantification 4 (2016) 380–412.
[62] T. Cui, C. Fox, M. J. O'Sullivan, A posteriori stochastic correction of reduced models in delayed-acceptance MCMC, with application to multiphase subsurface inverse problems, International Journal for Numerical Methods in Engineering 118 (2019) 578–605.
[63] L. Yan, T. Zhou, An adaptive surrogate modeling based on deep neural networks for large-scale Bayesian inverse problems, 2019.
[64] A. M. Stuart, Inverse problems: A Bayesian perspective, Acta Numerica 19 (2010) 451–459.
[65] N. Petra, J. Martin, G. Stadler, O. Ghattas, A computational framework for infinite-dimensional Bayesian inverse problems, part ii: Stochastic Newton MCMC with application to ice sheet flow inverse problems, SIAM Journal on Scientific Computing 36 (2014) A1525–A1555.
[66] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, D. B. Rubin, Bayesian Data Analysis, Chapman & Hall/CRC texts in statistical science, CRC Press, Boca Raton, Florida, 3rd edition, 2014.
[67] C. P. Robert, G. Casella, Monte Carlo Statistical Methods, Springer texts in statistics, Springer, New York, New York, 2nd edition, 2004.
[68] G. Prato, An Introduction to Infinite-Dimensional Analysis, volume 1 of *Universitext*, Springer, Berlin, Heidelberg, 2006.
[69] J. T. Oden, J. N. Reddy, An introduction to the mathematical theory of finite elements, Dover Publications, 2012.
[70] A. Quarteroni, A. Valli, Numerical approximation of partial differential equations, volume 23 of *Springer Series in Computational Mathematics*, Springer, Berlin, Heidelberg, 2008.
[71] M. M. Dunlop, Multiplicative noise in Bayesian inverse problems: Well-posedness and consistency of MAP estimators, 2019.

[72] S. L. Cotter, G. O. Roberts, A. M. Stuart, D. White, MCMC methods for functions: Modifying old algorithms to make them faster, Statistical Science 28 (2013).

[73] T. O'Leary-Roseberry, P. Chen, U. Villa, O. Ghattas, Derivative-Informed Neural Operator: An efficient framework for high-dimensional parametric derivative learning, arXiv preprint arXiv:2206.10745 (2022).

[74] K. Wu, T. O'Leary-Roseberry, P. Chen, O. Ghattas, Large-scale Bayesian optimal experimental design with derivative-informed projected neural network, 2022.

[75] S. Wang, M. A. Bhouri, P. Perdikaris, Fast PDE-constrained optimization via self-supervised operator learning, arXiv preprint arXiv:2110.13297 (2021).

[76] G. Cybenko, Approximation by superpositions of a sigmoidal function, Mathematics of control, signals and systems 2 (1989) 303–314.

[77] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural networks 4 (1991) 251–257.

[78] Q. Li, T. Lin, Z. Shen, Deep learning via dynamical systems: An approximation perspective, Journal of the European Mathematical Society (2022).

[79] H. Lin, S. Jegelka, ResNet with one-neuron hidden layers is a universal approximator, Neural Information Processing Systems (2018).

[80] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The expressive power of neural networks: A view from the width, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6232–6240.

[81] U. Anders, O. Korn, Model selection in neural networks, Neural networks 12 (1999) 309–323.

[82] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, X. Wang, A comprehensive survey of neural architecture search: Challenges and solutions, ACM Computing Surveys (CSUR) 54 (2021) 1–34.

[83] L. Demkowicz, J. Oden, W. Rachowicz, O. Hardy, Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure, Computer Methods in Applied Mechanics and Engineering 77 (1989) 79–112.

[84] A. L. Teckentrup, Convergence of Gaussian process regression with estimated hyper-parameters and applications in Bayesian inverse problems, SIAM/ASA Journal on Uncertainty Quantification 8 (2020) 1310–1337.

[85] N. Cvetković, H. C. Lie, H. Bansal, K. Veroy-Grepl, Choosing observation operators to mitigate model error in bayesian inverse problems, 2023.

[86] L. Yan, Y. X. Zhang, Convergence analysis of surrogate-based methods for Bayesian inverse problems, Inverse Problems 33 (2017).

[87] Y. Marzouk, D. Xiu, A stochastic collocation approach to Bayesian inference in inverse problems, Communications in Computational Physics 6 (2009) 826–847.

[88] A. M. Stuart, A. L. Teckentrup, Posterior consistency for Gaussian process approximations of Bayesian posterior distributions, Mathematics of Computation 87 (2017) 721–753.

[89] P. G. Ciarlet, Linear and Nonlinear Functional Analysis with Applications, the Society for Industrial and Applied Mathematics, 2013.

[90] J. M. Ortega, The Newton–Kantorovich theorem, The American Mathematical Monthly 75 (1968) 658–660.

[91] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Computer methods in applied mechanics and engineering 142 (1997) 1–88.

[92] M. Ainsworth, J. T. Oden, A Posteriori Error Estimation in Finite Element Analysis, John Wiley & Sons, Inc., 2000.

[93] R. Rannacher, F.-T. Suttmeier, A feed-back approach to error control in finite element methods: Application to linear elasticity, Computational Mechanics 19 (1997) 434–446.

[94] K. G. van der Zee, J. T. Oden, S. Prudhomme, A. Hawkins-Daarud, Goal-oriented error estimation for Cahn–Hilliard models of binary phase transition, Numerical Methods for Partial Differential Equations 27 (2011) 160–196.

[95] T. O'Leary-Roseberry, Efficient and dimension independent methods for neural network surrogate construction and training, Ph.D. thesis, 2020.

[96] A. Manzoni, F. Negri, A. Quarteroni, Dimensionality reduction of parameter-dependent problems through proper orthogonal decomposition, Annals of Mathematical Sciences and Applications 1 (2016) 341–377.

[97] A. Quarteroni, A. Manzoni, F. Negri, Reduced basis methods for partial differential equations: An introduction, volume 92, Springer, 2015.

[98] T. O'Leary-Roseberry, N. Alger, O. Ghattas, Inexact Newton methods for stochastic non-convex optimization with applications to neural network training, arXiv preprint arXiv:1905.06738 (2019).

[99] T. O'Leary-Roseberry, N. Alger, O. Ghattas, Low Rank Saddle Free Newton: A scalable method for stochastic nonconvex optimization, arXiv preprint arXiv:2002.02881 (2020).

[100] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The FEniCS project version 1.5, Archive of Numerical Software 3 (2015).

[101] U. Villa, N. Petra, O. Ghattas, hIPPYlib: An extensible software framework for large-scale inverse problems, The Journal of Open Source Software 3 (2018) 940.

[102] T. O'Leary-Roseberry, U. Villa, hIPPYflow: Dimension reduced surrogate construction for parametric PDE maps in Python, 2021.

[103] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[104] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, G. N. Wells, Unified Form Language: A domain-specific language

for weak formulations of partial differential equations, ACM Trans. Math. Softw. 40 (2014).
[105] N. H. Gokhale, P. E. Barbone, A. A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: The compressible case, Inverse Problems 24 (2008) 045010.
[106] S. Goenezen, P. Barbone, A. A. Oberai, Solution of the nonlinear elasticity imaging inverse problem: The incompressible case, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1406–1420.
[107] J.-S. Affagard, P. Feissel, S. F. Bensamoun, Identification of hyperelastic properties of passive thigh muscle under compression with an inverse method from a displacement field measurement, Journal of Biomechanics 48 (2015) 4081–4086.
[108] Y. Mei, B. Stover, N. Afsar Kazerooni, A. Srinivasa, M. Hajhashemkhani, M. Hematiyan, S. Goenezen, A comparative study of two constitutive models within an inverse approach to determine the spatial stiffness distribution in soft materials, International Journal of Mechanical Sciences 140 (2018) 446–454.
[109] C. Jog, P. Motamarri, An energy-momentum conserving algorithm for nonlinear transient analysis within the framework of hybrid elements, Journal of Mechanics of Materials and Structures 4 (2009) 157–186.
[110] C. S. Jog, Continuum mechanics, volume 1, Cambridge University Press, 2015.
[111] L. Chevalier, S. Calloch, F. Hild, Y. Marco, Digital image correlation used to analyze the multiaxial behavior of rubber-like materials, European Journal of Mechanics-A/Solids 20 (2001) 169–187.
[112] K. M. Moerman, C. A. Holt, S. L. Evans, C. K. Simms, Digital image correlation and finite element modelling as a method to determine mechanical properties of human soft tissue in vivo, Journal of biomechanics 42 (2009) 1150–1153.
[113] N. McCormick, J. Lord, Digital image correlation, Materials Today 13 (2010) 52–54.
[114] D. L. B. R. Jurjo, C. Magluta, N. Roitman, P. B. Gonçalves, Analysis of the structural behavior of a membrane using digital image processing, Mechanical Systems and Signal Processing 54 (2015) 394–404.
[115] Z. Li, Z. Liu, An algorithm for obtaining real stress field of hyperelastic materials based on digital image correlation system, International Journal of Computational Materials Science and Engineering 6 (2017) 1850003.
[116] J. Ribeiro, H. Lopes, P. Martins, A hybrid method to characterise the mechanical behaviour of biological hyper-elastic tissues, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization 5 (2017) 157–164.
[117] M. Flaschel, S. Kumar, L. De Lorenzis, Unsupervised discovery of interpretable hyperelastic constitutive laws, Computer Methods in Applied Mechanics and Engineering 381 (2021) 113852.

Appendix A. The full statement and proof of Theorem 1

Theorem 1 (Operator learning errors in Bayesian inverse problems)**.** *Assume $\mathcal{U}$ and $\mathcal{M}$ are real-valued separable Hilbert spaces equipped with inner product–induced norm $\|\cdot\|_{\mathcal{U}}$ and $\|\cdot\|_{\mathcal{M}}$. Let ν_M be a probability measure on $\mathcal{M}$, and $\mathcal{F}, \widetilde{\mathcal{F}} \in L^p(\mathcal{M}, \nu_M; \mathcal{U})$, $p \in [2, \infty]$, where the Bochner space is equipped with the norm*

$$\|\mathcal{G}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} = \begin{cases} \left(\mathbb{E}_{M\sim\nu_M}\left[\|\mathcal{G}(M)\|_{\mathcal{U}}^p\right]\right)^{1/p} & p \in [1,\infty) \\ \operatorname{ess\,sup}_{M\sim\nu_M} \|\mathcal{G}(M)\|_{\mathcal{U}} & p = \infty\,. \end{cases}$$

Assume the observation operator $\boldsymbol{\mathcal{B}} : \mathcal{U} \to \mathbb{R}^{n_y}$ satisfies

$$\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m)\|_2 \le c_B \|\mathcal{F}(m)\|_{\mathcal{U}}\,, \quad \left\|(\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}})(m)\right\|_2 \le \widetilde{c_B} \|\mathcal{F}(m)\|_{\mathcal{U}}\,, \quad \nu_M\text{-a.e.}\,,$$
$$\left\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m) - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}})(m)\right\|_2 \le c_L \left\|\mathcal{F}(m) - \widetilde{\mathcal{F}}(m)\right\|_{\mathcal{U}}\,, \quad \nu_M\text{-a.e.}$$

Assume the additive noise $\mathbb{R}^{n_y} \ni \boldsymbol{N} \sim \nu_{\boldsymbol{N}} = \mathcal{N}(\boldsymbol{0}, \boldsymbol{C_N})$ is normally distributed with a probability density function $\pi_{\boldsymbol{N}}(\boldsymbol{x}) = ((2\pi)^{n_y} \det(\boldsymbol{C_N}))^{-1/2} \exp(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{C_N} \boldsymbol{x})$, $\boldsymbol{x} \in \mathbb{R}^{n_y}$.

For a set of observed data $\boldsymbol{y}^* \in \mathbb{R}^{n_y}$, we have

$$\mathcal{E}_{post} \le c \|\mathcal{E}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}\,, \quad c > 0\,,$$

where $\mathcal{E}_{post}$ is the Kullback–Leibler divergence between the posterior distribution defined by $\mathcal{F}$ and $\widetilde{\mathcal{F}}$.

$$\mathcal{E}_{post} := D_{KL}(\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)||\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*))\,, \quad \mathcal{E}(m) := \mathcal{F}(m) - \widetilde{\mathcal{F}}(m)\,, \quad \nu_M\text{-a.e.}\,,$$

and the divergence is governed by Bayes' rule

$$\frac{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_M}(m) = \frac{1}{Z(\boldsymbol{y}^*)}\underbrace{\pi_{\boldsymbol{N}}\left(\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}}\circ\mathcal{F})(m)\right)}_{=:\mathcal{L}(m;\boldsymbol{y}^*)} \quad a.s., \quad Z(\boldsymbol{y}^*) = \mathbb{E}_{M\sim\nu_M}[\mathcal{L}(M;\boldsymbol{y}^*)]\,,$$

$$\frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_M}(m) = \frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\underbrace{\pi_{\boldsymbol{N}}\left(\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}}\circ\widetilde{\mathcal{F}})(m)\right)}_{=:\widetilde{\mathcal{L}}(m;\boldsymbol{y}^*)} \quad a.s., \quad \widetilde{Z}(\boldsymbol{y}^*) = \mathbb{E}_{M\sim\nu_M}[\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)]\,,$$

1131 *where we assume* $\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*) \sim \nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*) \sim \nu_M\cdot$*, meaning that they are pairwise equivalent.*

Proof. Let us first dissect $\mathcal{E}_{\mathrm{post}}$ into two parts:

$$\begin{aligned}
\mathcal{E}_{\mathrm{post}} &= \mathbb{E}_{M\sim\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}\left[\ln\left(\frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}(M)\right)\right] && \text{(Def. of KL divergence)}\\
&= \mathbb{E}_{M\sim\nu_M}\left[\ln\left(\frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}(M)\right)\frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right] && \text{(Change of var. formula)}\\
&= \mathbb{E}_{M\sim\nu_M}\left[\ln\left(\frac{\mathrm{d}\nu_{M|\widetilde{\boldsymbol{Y}}}(\cdot|\boldsymbol{y}^*)}{\mathrm{d}\nu_M}(M)\frac{\mathrm{d}\nu_M}{\mathrm{d}\nu_{M|\boldsymbol{Y}}(\cdot|\boldsymbol{y}^*)}(M)\right)\frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right] && \text{(Mutually abs. continuous)}\\
&= \mathbb{E}_{M\sim\nu_M}\left[\ln\left(\frac{Z(\boldsymbol{y}^*)}{\widetilde{Z}(\boldsymbol{y}^*)}\frac{\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)}{\mathcal{L}(M;\boldsymbol{y}^*)}\right)\frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right] && \text{(Bayes' rule)}\\
&= \underbrace{\ln\left(\frac{Z(\boldsymbol{y}^*)}{\widetilde{Z}(\boldsymbol{y}^*)}\right)}_{\text{Ⓐ}} + \underbrace{\frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\mathbb{E}_{M\sim\nu_M}\left[\ln\left(\frac{\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)}{\mathcal{L}(M;\boldsymbol{y}^*)}\right)\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right]}_{\text{Ⓑ}}. && \text{(Def. of model evidence)}
\end{aligned}$$

The term that involves the likelihoods can be bounded by

$$\begin{aligned}
\text{Ⓑ} &= \frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\mathbb{E}_{M\sim\nu_M}\left[\left(\Phi(M)-\widetilde{\Phi}(M)\right)\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right] \quad \left(\begin{cases}\Phi(m) = \frac{1}{2}\left\|\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}}\circ\mathcal{F})(m)\right\|^2_{\boldsymbol{C}_{\boldsymbol{N}}^{-1}}\\ \widetilde{\Phi}(m) = \frac{1}{2}\left\|\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}}\circ\widetilde{\mathcal{F}})(m)\right\|^2_{\boldsymbol{C}_{\boldsymbol{N}}^{-1}}\end{cases}\right)\\
&\le \frac{1}{\widetilde{Z}(\boldsymbol{y}^*)}\mathbb{E}_{M\sim\nu_M}\left[\left|\Phi(M)-\widetilde{\Phi}(M)\right|\widetilde{\mathcal{L}}(M;\boldsymbol{y}^*)\right] \quad \text{(Jensen's ineq.)}\\
&\le \frac{\left\|\widetilde{\mathcal{L}}(\cdot;\boldsymbol{y}^*)\right\|_{L^{q^*}(\mathcal{M},\nu_{\mathcal{M}})}}{\widetilde{Z}(\boldsymbol{y}^*)}\left\|\Phi-\widetilde{\Phi}\right\|_{L^{p^*}(\mathcal{M},\nu_M)}. \quad \text{(Hölder's ineq., } q^* \text{ is the conj. exp. to } p^*\in[1,\infty])
\end{aligned}$$

We make several remarks on the last inequality. First, The validity of the inequality is conditional upon whether we can bound $\left\|\Phi-\widetilde{\Phi}\right\|_{L^{p^*}(\mathcal{M},\nu_M)}$ for some $p^*\in[1,\infty]$. Second, if such a p^* exists, the fraction term is larger than one if $p^*\in[1,\infty)$ and equal to one for $p^*=\infty$, as the following inequalities hold for any $q\in[1,\infty]$ due to inclusion of Lebesgue spaces defined on domains with finite measures ($\nu_M(\mathcal{M})=1$):

$$\widetilde{Z}(\boldsymbol{y}^*) = \left\|\widetilde{\mathcal{L}}(\cdot;\boldsymbol{y}^*)\right\|_{L^1(\mathcal{M},\nu_M)} \le \left\|\widetilde{\mathcal{L}}(\cdot;\boldsymbol{y}^*)\right\|_{L^q(\mathcal{M},\nu_M)} \le \left((2\pi)^{n_y}\det(\boldsymbol{C}_{\boldsymbol{N}})\right)^{-1/2}.$$

1132 Now we seek to

1133 1. bound $\mathcal{E}_P(m) \coloneqq \Phi(m) - \widetilde{\Phi}(m)$, ν_M-a.e., from above for some p^*, and

1134 2. bound $\widetilde{Z}(\boldsymbol{y}^*)$ from below.

From now on, we assume $p \in [2, \infty)$ as the extension to $p = \infty$ is straightforward.

First, we examine $\boldsymbol{\mathcal{B}} \circ \mathcal{F}$, $\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}$, and $\boldsymbol{\mathcal{E}}_B(m) \coloneqq (\boldsymbol{\mathcal{B}} \circ \mathcal{F})(m) - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}})(m)$, ν_M-a.e.,

$$\begin{aligned}
\|\boldsymbol{\mathcal{B}} \circ \mathcal{F}\|^p_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})} &= \mathbb{E}_{M\sim\nu_M}\left[\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M)\|_2^p\right] && \text{(Def. of Bochner space norm)}\\
&\le c_B^p \mathbb{E}_{M\sim\nu_M}\left[\|\mathcal{F}(M)\|_{\mathcal{U}}^p\right] && \text{(Property of } \boldsymbol{\mathcal{B}}\text{)}\\
&= c_B^p \|\mathcal{F}\|^p_{L^p(\mathcal{M},\nu_M;\mathcal{U})} < \infty\,. && \text{(Def. of Bochner space norm)}
\end{aligned}$$

We thus have $\boldsymbol{\mathcal{B}} \circ \mathcal{F}, \boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}} \in L^p(\mathcal{M}, \nu_M; \mathbb{R}^{n_y})$. Similarly,

$$\begin{aligned}
\|\boldsymbol{\mathcal{E}}_B\|^p_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})} &= \mathbb{E}_{M\sim\nu_M}\left[\left\|(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M) - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}})(M)\right\|_2^p\right] && \text{(Def. of Bochner space norm)}\\
&\le c_L^p \mathbb{E}_{M\sim\nu_M}\left[\left\|\mathcal{F}(M) - \widetilde{\mathcal{F}}(M)\right\|_{\mathcal{U}}^p\right] && \text{(Property of } \boldsymbol{\mathcal{B}}\text{)}\\
&= c_L^p \|\mathcal{E}\|^p_{L^p(\mathcal{M},\nu_M;\mathcal{U})}\,. && \text{(Def. of Bochner space norm)}
\end{aligned}$$

Now we have

$$\begin{aligned}
\|\mathcal{E}_P\|^{p/2}_{L^{p/2}(\mathcal{M},\nu_M)} &= \mathbb{E}_{M\sim\nu_M}\left[|\Phi(M) - \widetilde{\Phi}(M)|^{p/2}\right]\\
&= \mathbb{E}_{M\sim\nu_M}\left[\left|\left(\frac{1}{2}(\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M) + \frac{1}{2}(\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(M) - \boldsymbol{y}^*\right)^T \boldsymbol{C}_{\boldsymbol{N}}^{-1}\boldsymbol{\mathcal{E}}_B(M)\right|^{p/2}\right]\\
&\le \mathbb{E}_{M\sim\nu_M}\left[\left\|\frac{1}{2}\boldsymbol{C}_{\boldsymbol{N}}^{-1}\left((\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M) + (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(M) - 2\boldsymbol{y}^*\right)\right\|_2^{p/2} \|\boldsymbol{\mathcal{E}}_B(M)\|_2^{p/2}\right]\\
&\le \underbrace{\mathbb{E}_{M\sim\nu_M}\left[\left\|\frac{1}{2}\boldsymbol{C}_{\boldsymbol{N}}^{-1}((\boldsymbol{\mathcal{B}} \circ \mathcal{F})(M) + (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}_{\boldsymbol{w}})(M) - 2\boldsymbol{y}^*)\right\|_2^{p}\right]^{1/2}}_{c_1^p} \mathbb{E}_{M\sim\nu_M}\left[\|\boldsymbol{\mathcal{E}}_B(M)\|_2^p\right]^{1/2}\,.
\end{aligned}$$

Where we apply the Cauchy–Schwarz inequality for the two inequalities above. The first expectation above is bounded by

$$\begin{aligned}
c_1^p &\le \frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2^p \left(\|\boldsymbol{\mathcal{B}} \circ \mathcal{F}\|_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})} + \left\|\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}\right\|_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})} + 2\|\boldsymbol{y}^*\|_2\right)^p && \text{(Minkowski ineq.)}\\
&\le \frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2^p \left(c_B\|\mathcal{F}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} + \widetilde{c_B}\left\|\widetilde{\mathcal{F}}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} + 2\|\boldsymbol{y}^*\|_2\right)^p < \infty\,.
\end{aligned}$$

Consequently

$$\|\mathcal{E}_P\|_{L^{p^*}(\mathcal{M},\nu_M)} \le c_1\|\boldsymbol{\mathcal{E}}_B\|_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})} \le c_1 c_L\|\mathcal{E}\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})} < \infty\,, \quad p^* \in [1, p/2]\,.$$

Second, we examine the lower bound for $\widetilde{Z}(\boldsymbol{y}^*)$:

$$\begin{aligned}
c_2^{-1} \coloneqq ((2\pi)^{n_y}\det(\boldsymbol{C}_{\boldsymbol{N}}))^{1/2}\widetilde{Z}(\boldsymbol{y}^*) &= \mathbb{E}_{M\sim\nu_M}\left[\exp(-\widetilde{\Phi}(M))\right]\\
&\ge \exp\left(-\mathbb{E}_{M\sim\nu_M}\left[\widetilde{\Phi}(M)\cdot 1\right]\right) && \text{(Jen. ineq.)}\\
&\ge \exp\left(-\frac{1}{2}\mathbb{E}_{M\sim\nu_M}\left[\left\|\boldsymbol{y}^* - (\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}})(M)\right\|^p_{\boldsymbol{C}_{\boldsymbol{N}}^{-1}}\right]^{1/p}\right) && \text{(Höd. ineq.)}\\
&\ge \exp\left(-\frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2\left(\|\boldsymbol{y}^*\|_2 + \left\|\boldsymbol{\mathcal{B}} \circ \widetilde{\mathcal{F}}\right\|_{L^p(\mathcal{M},\nu_M;\mathbb{R}^{n_y})}\right)\right) && \text{(Min. ineq.)}\\
&\ge \exp\left(-\frac{1}{2}\left\|\boldsymbol{C}_{\boldsymbol{N}}^{-1}\right\|_2\left(\|\boldsymbol{y}^*\|_2 + \widetilde{c_B}\left\|\widetilde{\mathcal{F}}\right\|_{L^p(\mathcal{M},\nu_M;\mathcal{U})}\right)\right) > 0\,. && \text{(A.1)}
\end{aligned}$$

Therefore, for $q \in [p/(p-2), \infty]$ we have

$$c_3 \coloneqq \frac{\left\|\widetilde{\mathcal{L}}(\cdot; \boldsymbol{y}^*)\right\|_{L^q(\mathcal{M}, \nu_{\mathcal{M}})}}{\widetilde{Z}(\boldsymbol{y}^*)} \leq \frac{\left\|\widetilde{\mathcal{L}}(\cdot; \boldsymbol{y}^*)\right\|_{L^\infty(\mathcal{M}, \nu_{\mathcal{M}})}}{\widetilde{Z}(\boldsymbol{y}^*)} \leq c_2 < \infty \,.$$

Consequently, the term Ⓑ is bounded from above as follows

$$\text{Ⓑ} \leq c_1 c_3 c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M}, \nu_M; \mathcal{U})} \,.$$

Following, the term Ⓐ involving normalization constants can be bounded by

$$\begin{aligned}
\text{Ⓐ} &= \ln\left(1 + \frac{Z(\boldsymbol{y}^*) - \widetilde{Z}(\boldsymbol{y}^*)}{\widetilde{Z}(\boldsymbol{y}^*)}\right) \\
&\leq \ln\left(1 + \frac{|Z(\boldsymbol{y}^*) - \widetilde{Z}(\boldsymbol{y}^*)|}{\widetilde{Z}(\boldsymbol{y}^*)}\right) && (\ln(\cdot) \text{ monotonic incres.}) \\
&\leq \frac{1}{\widetilde{Z}(\boldsymbol{y}^*)} \left|Z(\boldsymbol{y}^*) - \widetilde{Z}(\boldsymbol{y}^*)\right| && (\log(1+x) \leq x \quad \forall x \geq 0) \\
&\leq \frac{1}{\left((2\pi)^{n_y} \det(\boldsymbol{C_N})\right)^{1/2} \widetilde{Z}(\boldsymbol{y}^*)} \underbrace{\left|\mathbb{E}_{M \sim \nu_M}\left[\exp(-\Phi(m)) - \exp(-\widetilde{\Phi}(m))\right]\right|}_{\text{Ⓓ}} \,.
\end{aligned}$$

Next, we estimate Ⓓ as follows

$$\begin{aligned}
\text{Ⓓ} &\leq \mathbb{E}_{M \sim \nu_M}\left[\left|\exp(-\Phi(m)) - \exp(-\widetilde{\Phi}(m)\right|\right] && (\text{Jensen's ineq.}) \\
&\leq \mathbb{E}_{M \sim \nu_M}\left[\left|\mathcal{E}_P \cdot 1\right|\right] && (|e^{-x_1} - e^{-x_2}| \leq |x_1 - x_2|,\ x_1, x_2 \geq 0) \\
&\leq \left\|\mathcal{E}_P\right\|_{L^{p/2}(\mathcal{M}, \nu_M)} && (\text{Hölder's ineq.}) \\
&\leq c_1 c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M}, \nu_M; \mathcal{U})} \,.
\end{aligned}$$

Combining the two inequalities above and noting the definition of a constant c_2 in (A.1), we have shown

$$\text{Ⓐ} \leq c_1 c_2 c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M}, \nu_M; \mathcal{U})} \,.$$

In conclusion, we have

$$\mathcal{E}_{\text{post}} \leq c_1 (c_2 + c_3) c_L \left\|\mathcal{E}\right\|_{L^p(\mathcal{M}, \nu_M; \mathcal{U})} \,.$$

□

Appendix B. The full statement of the corollary to the Newton–Kantorovich theorem

We state the corollary to the Newton–Kantorovich theorem assuming the equivalency between solving the error correction problem in the solution set $\mathcal{V}_u$ and generating one Newton step in $\mathcal{U}_0$ with a given "lifting" element $u_L \in \mathcal{V}_u$ for the nonlinear equation

$$\text{Given } m \in \mathcal{M}, \text{ find } v \in \mathcal{U}_0 \text{ such that } \quad \widetilde{\mathcal{R}}(v, m) = 0 \,, \quad \widetilde{\mathcal{R}}(v, m) \coloneqq \mathcal{R}(v + u_L, m) \quad \in \mathcal{U}_0^* \,.$$

We thus redefine the forward operator and the neural operator to $\mathcal{F}(\cdot) - u_L$ and $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(\cdot) - u_L - \widetilde{u}^\perp$, where $\widetilde{u}^\perp \in \mathcal{U}_0^\perp$ represent $\widetilde{\mathcal{F}}_{\boldsymbol{w}}(\cdot) - u_L$ projected to $\mathcal{U}_0^\perp$. Additionally, we define the space of bounded linear operator between two Banach spaces $\mathcal{X}$ and $\mathcal{Y}$ as $B(\mathcal{X}, \mathcal{Y})$ equipped with the operator norm,

$$\left\|\mathcal{G}\right\|_{B(\mathcal{X}, \mathcal{Y})} = \sup_{y \neq 0} \frac{\left\|\mathcal{G}(x)\right\|_{\mathcal{Y}}}{\left\|x\right\|_{\mathcal{X}}} \,.$$

The following result directly translates the Newton–Kantorovich theorem in Banach spaces as stated by Ciarlet [89] to our setting.

Corollary 2 (Global error reduction for residual-based error correction)**.** *Let $\mathcal{U}$ and $\mathcal{M}$ be real-valued separable Hilbert spaces equipped with inner products–induced norm $\|\cdot\|_{\mathcal{U}}$ and $\|\cdot\|_{\mathcal{M}}$. Let ν_M be a probability measure on $\mathcal{M}$ and $\widetilde{\mathcal{F}} \in L^\infty(\mathcal{M}, \nu_M; \mathcal{U})$. Assume there exists $\mathcal{F} \in L^\infty(\mathcal{M}, \nu_M; \mathcal{U})$ such that*

$$\mathcal{R}(\mathcal{F}(m), m) \equiv 0\,, \quad \nu_M\text{-a.e.}\,, \tag{B.1}$$

for $\mathcal{R} : \mathcal{U} \times \mathcal{M} \to \mathcal{U}^*$. Assume for any $m \in \mathcal{M}$, ν_M-a.e., there exists an open set $\mathcal{D}_m \subseteq \mathcal{U}$ with $\widetilde{\mathcal{F}}(m) \in \mathcal{D}_m$ such that $\mathcal{R}(\cdot, m) : \mathcal{D}_m \to \mathcal{U}^*$ is differentiable and its derivative at $\widetilde{\mathcal{F}}(m)$, $\delta_u \mathcal{R}(\widetilde{\mathcal{F}}(m), m) \in B(\mathcal{U}, \mathcal{U}^*)$, is bijective. Assume that there exists three constants c_1, c_2, c_3 such that

$$0 < c_1 c_2 c_3 \leq \frac{1}{2} \text{ and } B_r(\widetilde{\mathcal{F}}(m)) \subset \mathcal{D}_m, \text{ where } r \coloneqq \frac{1}{c_2 c_3}\,,$$

and

$$\begin{aligned}
\left\| \delta_u \mathcal{R}(\widetilde{\mathcal{F}}(\cdot), \cdot)^{-1} \mathcal{R}(\widetilde{\mathcal{F}}(\cdot), \cdot) \right\|_{L^\infty(\mathcal{M}, \nu; \mathcal{U})} &\leq c_1\,, \\
\left\| \delta_u \mathcal{R}(\widetilde{\mathcal{F}}(\cdot), \cdot)^{-1} \right\|_{L^\infty(\mathcal{M}, \nu_M; B(\mathcal{U}^*, \mathcal{U}))} &\leq c_2\,, \\
\|\delta_u \mathcal{R}(u_1, m) - \delta_u \mathcal{R}(u_2, m)\|_{B(\mathcal{U}, \mathcal{U}^*)} &\leq c_3 \|u_1 - u_2\|_{\mathcal{U}}\,, \quad \forall u_1, u_2 \in B_r(\widetilde{\mathcal{F}}(m))\,, \quad \nu_M\text{-a.e.}
\end{aligned}$$

Then for any $m \in \mathcal{M}$, ν_M-a.e., $\delta_u \mathcal{R}(u, m)$ is bijective at each $u \in B_r(\widetilde{\mathcal{F}}(m))$ and the sequence $\{u_j\}_{j=0}^\infty$ with $u_0 = \widetilde{\mathcal{F}}(m)$ defined as

$$u_{j+1} = u_j - \delta_u \mathcal{R}(u_j, m)^{-1} \mathcal{R}(u_j, m)\,, \quad k \geq 0\,,$$

is contained in the ball $B_{r_-}(\widetilde{\mathcal{F}}(m))$, where

$$r_- \coloneqq \frac{1 - \sqrt{1 - 2c_1 c_2 c_3}}{c_2 c_3} \leq r\,,$$

and converges to $\mathcal{F}(m)$. Besides, for each $j \geq 0$,

$$\begin{cases}
\|u_j - \mathcal{F}(\cdot)\|_{L^\infty(\mathcal{M}, \nu_M; \mathcal{U})} \leq \dfrac{r}{2^j} (\dfrac{r_-}{r})^{2^j}\,, & c_1 < \dfrac{1}{2c_2 c_3}\,, \\
\|u_j - \mathcal{F}(\cdot)\|_{L^\infty(\mathcal{M}, \nu_M; \mathcal{U})} \leq \dfrac{r}{2^j}\,, & c_1 = \dfrac{1}{2c_2 c_3}\,.
\end{cases}$$

Appendix C. Numerical examples: The $L^2(\Omega)$ generalization accuracy

As discussed in Remark 3, we present the $L^2(\Omega)$ generalization accuracy for the trained neural operators for both numerical examples in Figure C.17. The $L^2(\Omega)$ generalization accuracy is given by

$$100 \left(1 - \sqrt{\mathbb{E}_{M \sim \nu_M} \left[\frac{\left\| \mathcal{F}(M) - \widetilde{\mathcal{F}}_{\boldsymbol{w}}(M) \right\|_{L^2(\Omega)}^2}{\|\mathcal{F}(M)\|_{L^2(\Omega)}^2} \right]} \right)\,. \tag{C.1}$$

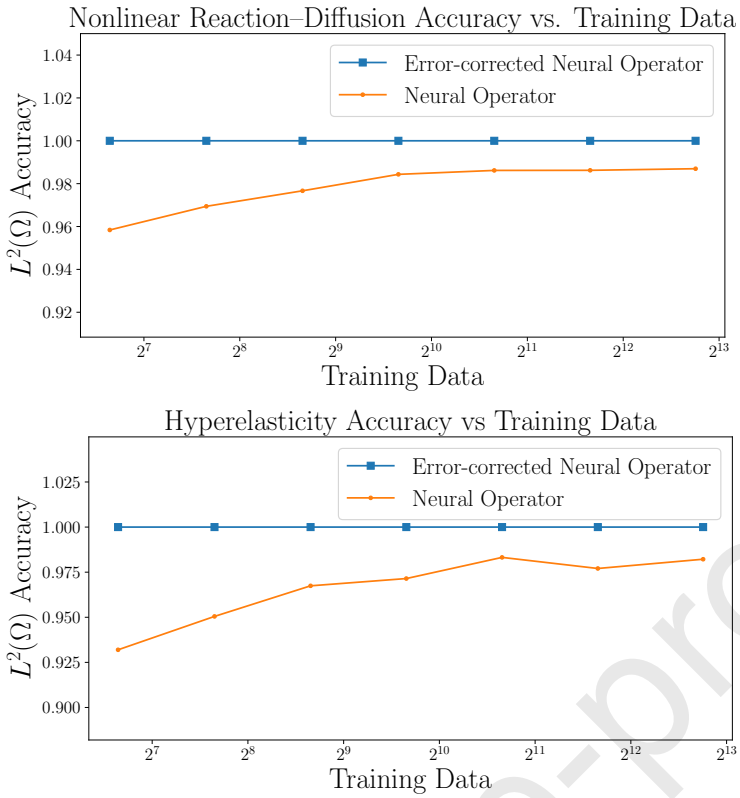


Figure C.17: A study of the $L^2(\Omega)$ generalization accuracy, as defined in (C.1), of the neural operators studied in Figures 5 and 12. The accuracy is computed using 512 data unseen during training.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

CRediT author statement

Lianghao Cao:
Conceptualization, Methodology, Software, Formal analysis, Data Curation, Investigation, Visualization, Writing - Original Draft

Thomas O'Leary-Roseberry:
Conceptualization, Methodology, Software, Formal analysis, Data Curation, Investigation, Visualization, Writing - Original Draft

Prashant K. Jha:
Conceptualization, Methodology, Software, Visualization, Investigation, Writing - Original Draft

J. Tinsley Oden:
Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition

Omar Ghattas:
Conceptualization, Methodology, Writing - Review & Editing, Funding acquisition