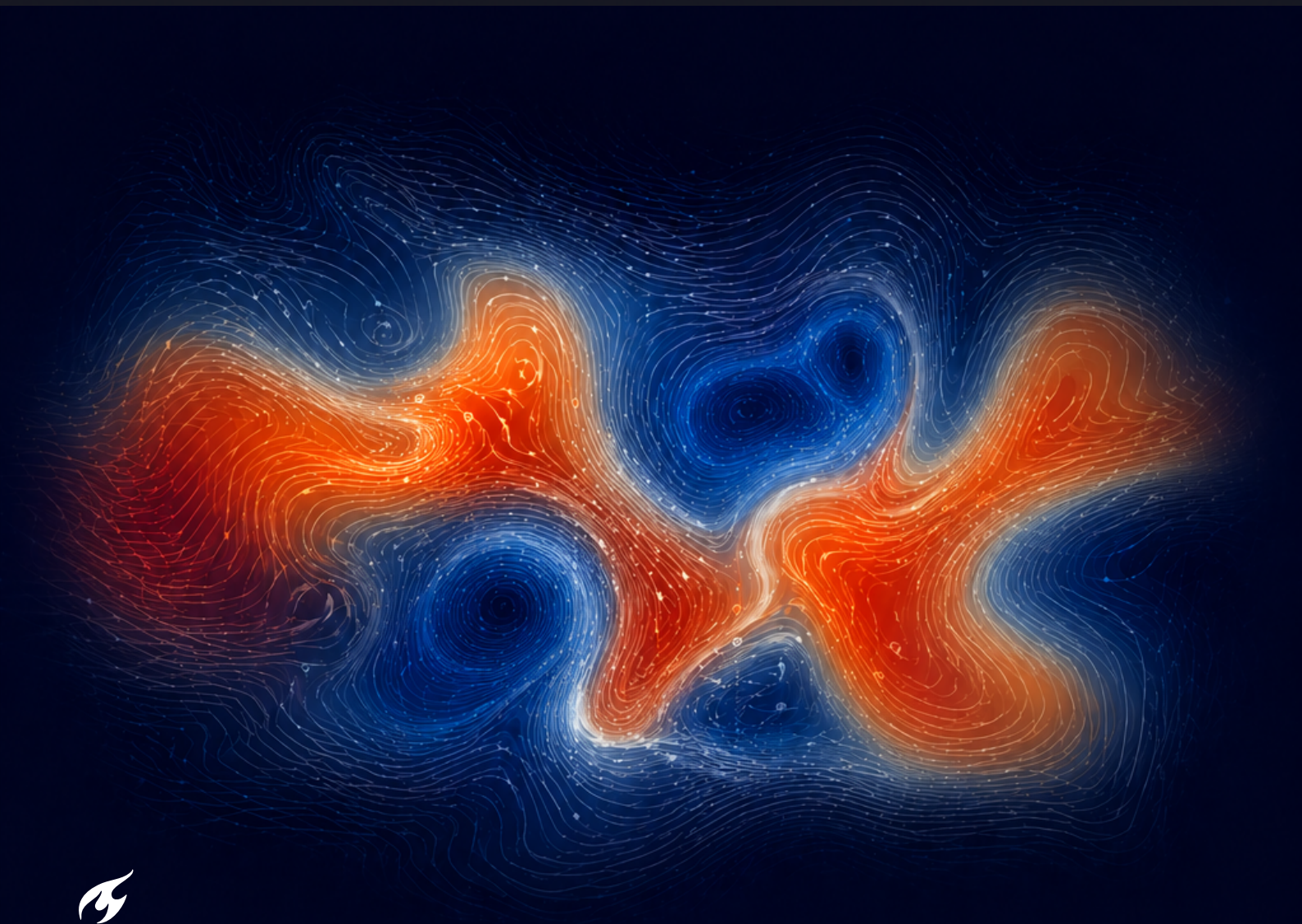
Bayesian Inverse Generative Neural Operator

Latent-Space Posterior Formulation for PDE-Constrained Inverse Problems

Henry Page



TU Delft

Bayesian Inverse Generative Neural Operator

Latent-Space Posterior Formulation for PDE-Constrained Inverse Problems

by

Henry Page

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Monday 29 June 2026 at 1:00 PM.

Project duration:	November 10, 2025 – June 29, 2026	
Graduation committee:	Dr. D.M.J. Tax,	Intelligent Systems, TU Delft
	Prof. dr. M.M. de Weerdt,	Software Technology, TU Delft
	Dr. J. Sun,	Intelligent Systems, TU Delft
	Dr. A. Heinlein,	Applied Mathematics, TU Delft

Faculty:	Electrical Engineering, Mathematics and Computer Science, TU Delft
Research group:	Pattern Recognition & Bioinformatics
Cover:	Generated using ChatGPT Images 2.0 (OpenAI)
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

Preface

After five years at TU Delft, three for my bachelor's and two for my master's, I have learned that the most interesting work tends to find you when you are not looking for it. I came into the master's programme wanting to work in machine learning broadly, without a clear sense of where it would lead. It was a lecture on physics-informed machine learning in the Alternative Learning Strategies course that pulled me in a direction I had not expected. One paper led to another, and the thesis you are holding (or scrolling through) is where that rabbit hole eventually led.

I am grateful to my supervisors, Dr. Jing Sun and Dr. Alexander Heinlein, for helping me turn that initial curiosity into something concrete. In the early months, they were patient with a research direction that took time to find its footing, and their feedback in the months that followed shaped every part of this work. I would also like to thank Dr. David Tax, my responsible advisor, whose perspective during our regular meetings often helped me see things I would have otherwise missed.

To my dad, thank you for supporting me at every step of my education and always encouraging me to try new things. I am immensely grateful for that. To my mum, いつも応援してくれたおかげで、ここまで頑張ることができました。本当にありがとう。To Nic, good luck. I hope you have as much fun at university studying physics as I had writing this.

To my friends, both here in the Netherlands and everywhere else, thank you for the conversations, the distractions, and the reminders that there is more to life than debugging code at two in the morning.

Henry Page
Delft, June 2026

Abstract

Inverse problems governed by partial differential equations (PDEs) are ill-posed, and responsible use of their solutions requires quantifying the uncertainty in recovered parameters. Neural operator methods for inverse problems offer fast surrogates for classical solvers, but placing posteriors over network weights is intractable at scale. This thesis extends the Inverse Generative Neural Operator (IGNO) to full Bayesian posterior sampling by adding a normalising flow prior term to the inversion objective and replacing gradient-based optimisation with the No-U-Turn Sampler (NUTS). The extension requires no retraining of any network component. We evaluate the method on four inverse problems spanning Darcy flow, electrical impedance tomography (EIT), and the viscous Burgers equation. On in-domain test instances, the posterior achieves 93% to 100% empirical coverage at the 95% nominal level across all four benchmarks and responds appropriately to changes in observation noise and sensor count. The posterior mean matches or improves on the maximum a posteriori (MAP) point estimate in every case. A Laplace approximation baseline, which fits a Gaussian posterior at the MAP estimate, fails on two of the four problems and does not consistently outperform NUTS on the two where it converges. Because the posterior formulation separates data, physics, and prior into additive terms, physical constraints can be incorporated during sampling alongside the data likelihood. Including PDE residuals as a virtual likelihood is most beneficial when observations alone leave the posterior under-determined, as demonstrated by EIT, where boundary-only measurements provide no direct information about the interior conductivity. The uncertainty estimates are unreliable for out-of-distribution coefficient fields. The learned prior pulls the posterior toward the training distribution, producing credible intervals that can be both narrow and wrong.

Contents

Abbreviations

The following table lists abbreviations used in this thesis.

Abbreviation	Expansion
CI	Credible Interval
CNN	Convolutional Neural Network
CRPS	Continuous Ranked Probability Score
CSRBF	Compactly Supported Radial Basis Function
DAIC	Delft AI Cluster
DGenNO	Deep Generative Neural Operator
DtN	Dirichlet-to-Neumann
EIT	Electrical Impedance Tomography
ELBO	Evidence Lower Bound
EM	Expectation-Maximisation
ESS	Effective Sample Size
FEM	Finite Element Method
GAN	Generative Adversarial Network
GMM	Gaussian Mixture Model
GP	Gaussian Process
GPU	Graphics Processing Unit
HMC	Hamiltonian Monte Carlo
IGNO	Inverse Generative Neural Operator
KL	Kullback-Leibler
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MH	Metropolis-Hastings
NF	Normalising Flow
NLL	Negative Log-Likelihood
NUTS	No-U-Turn Sampler
OOD	Out-Of-Distribution
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
rRMSE	Relative Root Mean Squared Error
SNR	Signal-to-Noise Ratio
VAE	Variational Autoencoder
VI	Variational Inference
WGAN	Wasserstein Generative Adversarial Network

Notation

The following table summarises the notation used throughout this thesis.

Symbol	Description	Introduced
Domain and spaces		
Ω	Spatial domain	Section 2.1
$\partial\Omega$	Boundary of the spatial domain	Section 2.1
$\mathcal{A}, \mathcal{U}$	Coefficient and solution function spaces	Section 2.1
Latent variables		
β_1	Latent variable for inversion, output of E_{θ_a}	Chapter 3
β_2	Encoded boundary condition, output of $E_{\theta_{bc}}$	Chapter 3
$\boldsymbol{\beta} = (\beta_1, \beta_2)$	Concatenated latent vector	Chapter 3
d_1, d_2	Latent dimensions of β_1 and β_2	Chapter 3
Architectural components		
E_{θ_a}	Coefficient encoder, maps $a \to \beta_1 \in [-1, 1]^{d_1}$	Chapter 3
$E_{\theta_{bc}}$	Boundary condition encoder, maps $bc \to \beta_2$	Chapter 3
G_{θ_u}	Solution decoder, maps $(\boldsymbol{\beta}, \boldsymbol{x}) \to u(\boldsymbol{x})$	Chapter 3
G_{θ_a}	Coefficient decoder, maps $\beta_1 \to a$ field	Chapter 3
F_ϕ	Normalising flow; models $p(\beta_1)$ as prior for sampling	Chapter 3
Objective terms		
$\mathcal{F}_{\text{data}}$	Data mismatch term	Section 3.3
$\mathcal{F}_{\text{pde}}$	PDE residual term (virtual likelihood)	Section 3.2
$\mathcal{F}_{\text{prior}}$	Normalising flow prior term	Section 3.4
$\mathcal{F}_{\text{rec}}$	Coefficient reconstruction term	Section 3.2
w_{data}, w_{pde}, w_{rec}	Loss weights	Section 3.2
σ_{data}, σ_{pde}, σ_{rec}	Noise standard deviations	Section 3.2
$\mathcal{L}_{\text{train}}$	Combined training loss	Section 3.2
N_c	Number of collocation points for weak-form residuals	Section 3.2
$\hat{R}_M$	Set of virtual PDE residual observations	Section 3.2
$U(\beta_1)$	Potential energy	Section 3.4
PDE operators		
$\mathcal{N}_a$	PDE differential operator, parameterised by a	Section 2.1
$\mathcal{B}$	Boundary operator	Section 2.1
$\mathcal{F}$	Forward solution operator; maps coefficient to solution fields	Section 2.1
Common symbols		
u_{obs}	Observed solution data	Section 2.1
$\tilde{a}$	Reconstructed coefficient field	Chapter 3
a^{true}	Ground-truth coefficient field	Section 4.2
$\tilde{a}^{\text{MAP}}$	MAP coefficient estimate	Section 4.3
$\tilde{a}^{\text{post}}$	Posterior mean coefficient field	Section 4.3
$\tilde{u}$	Decoded solution field	Chapter 3
u^{true}	Ground-truth solution field	Section 4.2
$\mathbb{R}$	Set of real numbers	
∇	Gradient (∇)	
$\nabla\cdot$	Divergence ($\nabla\cdot$)	
$\arg\min$	Argument that minimises a function	
Probability and statistics		
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2	
$\mathcal{GP}(\mu, C)$	Gaussian process with mean μ and covariance C	Section 4.2
$\mathbb{E}$	Expectation	
ρ	Spearman rank correlation	Section 4.3

1

Introduction

Inverse problems play a central role in scientific and engineering applications where internal properties of a physical system must be inferred from indirect, incomplete, noisy, or irregularly gridded measurements. Examples arise across diverse domains: seismic imaging seeks to reconstruct subsurface geological structures from surface sensor readings [1], medical imaging recovers tissue density distributions from X-ray attenuation measurements [2], material characterisation infers elastic and constitutive material properties from mechanical test and boundary field measurements [3, 4], and subsurface flow modelling infers permeability fields from sparse flow measurements [5].

The underlying physical behaviour of these systems is often governed by partial differential equations (PDEs) that relate unknown coefficient fields[1] to observable quantities. Recovering these coefficient fields from observations is fundamentally ill-posed: solutions may be non-unique and sensitive to small perturbations from measurement noise or model imperfection [8, 9].

Traditional PDE-constrained inversion methods address this challenge by formulating the estimation problem as an optimisation task that enforces consistency with both measurement data and governing physical laws [10]. These approaches typically require repeated forward and adjoint PDE solves, often leading to high computational cost and slow convergence, especially for large-scale problems or nonlinear models [11, 12]. These optimisation-based approaches produce point estimates without quantifying the inherent uncertainty in reconstructed coefficient fields. Since many coefficient fields can fit the observations equally well within measurement error, a single best-fit estimate is insufficient for responsible decision-making, which requires a full probabilistic characterisation of all coefficient fields consistent with the data. When model outputs inform decisions affecting health, safety, or the environment, practitioners have no way to judge whether a prediction is reliable or whether further investigation is needed. Without a measure of confidence, the cost of an incorrect prediction falls on the people affected by the decision, not on the model that produced it.

Bayesian inference provides a statistical framework for this characterisation, jointly accounting for measurement noise, modelling error, and prior knowledge to produce a probability distribution over coefficient fields consistent with the data [8]. In practice, however, each evaluation of this distribution requires a forward PDE solve, and sampling high-dimensional coefficient field spaces via Markov chain Monte Carlo (MCMC) demands many such solves [13].

Within scientific machine learning, physics-informed approaches embed physical knowledge directly into learned models, offering a data-driven alternative to classical solvers. Neural operators learn mappings between function spaces and, once trained, provide fast surrogates for PDE solutions that generalise across problem instances [14–16]. Recent lines of work combine operator learning with encoder-decoder architectures to map forward and inverse problems into low-dimensional latent spaces [6, 7,

[1] We use the term *coefficient field* to refer to a spatially varying function that enters the PDE as a coefficient, such as permeability or conductivity. These are the spatially distributed unknowns to be recovered from observations; the terminology follows [6, 7].

17, 18]. Reliable uncertainty quantification in this setting remains an open challenge. The Inverse Generative Neural Operator (IGNO) [7] achieves competitive accuracy on inverse problems requiring only a small set of coefficient samples and no paired solution data, but does not provide any measure of uncertainty. Existing approaches to uncertainty quantification in neural operator settings face tradeoffs between computational cost and calibration quality, as placing posteriors directly over operator weights is intractable at scale.

This thesis extends IGNO with Bayesian uncertainty quantification. By adding a learned prior term to the inversion objective and replacing gradient-based optimisation with MCMC sampling, the method produces a full probability distribution over plausible coefficient fields consistent with both the observations and the governing PDE, without retraining any network component.

This thesis investigates the following questions:

RQ1. Does Bayesian sampling in IGNO's latent space produce well-calibrated uncertainty estimates?

RQ2. What is the effect of including physics information during sampling? In particular, is the physics already sufficiently encoded in the learned latent representation, or does explicitly enforcing physical consistency during inference change reconstruction quality?

RQ3. How does the method perform on out-of-distribution coefficient fields, where the learned latent structure may no longer provide a reliable basis for inference?

We make two contributions: a Bayesian extension of IGNO's inversion procedure that enables full uncertainty quantification with no retraining, and experimental validation on four inverse problems spanning steady-state and time-dependent PDEs. Chapter 2 provides technical background and surveys related work. Chapter 3 presents the IGNO framework and our Bayesian extension. Chapter 4 reports experimental results. Chapter 5 discusses limitations and future directions. Chapter 6 concludes.

2

Background and related work

This chapter establishes the four foundational topics on which the contribution of this thesis rests: the structure of ill-posed inverse problems, neural operators as fast surrogates, Bayesian inference for uncertainty quantification, and normalising flows for encoding learned distributions. It then surveys how recent work has combined these elements for latent-space Bayesian inversion and identifies the gap that motivates our approach.

2.1. Inverse problems

We consider PDEs of the general form

$$\begin{aligned} \mathcal{N}_a[u](x) &= f(x), \quad x \in \Omega, \\ \mathcal{B}[u](x) &= g(x), \quad x \in \partial\Omega, \end{aligned}$$

where $\Omega \subset \mathbb{R}^n$ is a bounded domain, $u \in \mathcal{U}$ is the solution, $a \in \mathcal{A}$ is an unknown coefficient field, f is a source term, and g specifies the boundary values of u on $\partial\Omega$. Here $\mathcal{U}$ and $\mathcal{A}$ denote appropriate Banach spaces[1] over Ω. The differential operator $\mathcal{N}_a$ depends on the coefficient a, and the boundary operator $\mathcal{B}$ encodes the boundary condition. The forward (solution) operator $\mathcal{F} : \mathcal{A} \to \mathcal{U}$ maps coefficients to solutions via $u = \mathcal{F}(a)$. Given complete knowledge of a and appropriate boundary conditions, numerical methods (e.g. finite element methods) can compute u to desired accuracy [11].

The forward problem is typically well-posed in the Hadamard sense. For a given coefficient field a and boundary function g, the solution $u = \mathcal{F}(a)$ exists, is unique, and depends continuously on the input data [8]. Because $\mathcal{N}_a$ acts through derivatives, the solution at each point depends only on the coefficient in a neighbourhood of that point. For time-dependent problems, the solution at time t likewise depends only on the coefficient history up to t [12].

The inverse problem reverses this map: given observations of the solution field u, infer the coefficient field a. Although $\mathcal{N}_a$ acts locally, the inverse map is non-local. The coefficient a at any point influences the solution throughout Ω, so local observations carry global information about a. Similarly, in time-dependent problems, the measurement at time t reflects the integrated history of a, not an instantaneous value [12]. In practice, observations are sparse (measured at M sensor locations $\{x_i\}_{i=1}^M$, with $M \ll N_a$, where N_a is the number of degrees of freedom in the discretised coefficient field) and noisy (corrupted by measurement error $\xi \sim \mathcal{N}(0, \sigma^2)$) [5]. Formally, the inverse problem seeks a such that

$$u_{\text{obs}}(x_i) = \mathcal{F}(a)(x_i) + \xi_i, \quad i = 1, \dots, M.$$

Figure 2.1 depicts the forward and inverse problem setup.

The PDE coefficient recovery problems we consider involve a nonlinear forward operator. The coefficient a appears inside the differential operator $\mathcal{N}_a$, so the map $a \mapsto u = \mathcal{F}(a)$ is nonlinear in a [11]. This

[1] Loosely speaking, a Banach space is a complete normed space of functions [16].

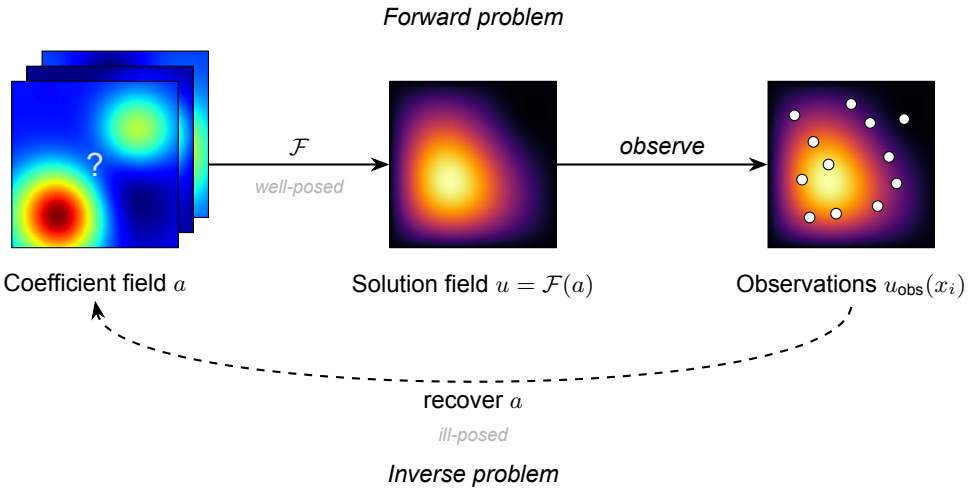


Figure 2.1: The forward operator $\mathcal{F}$ maps a coefficient field a to a solution field u, and the inverse problem seeks to recover a from sparse, noisy observations u_{obs}. Multiple visually distinct coefficient fields are stacked on the left to emphasise that the inverse map is not unique, as many different a can produce observations consistent with the same data.

distinguishes coefficient recovery from classical linear inverse problems such as deconvolution, where $\mathcal{F}$ is linear and regularisation methods admit closed-form solutions [19]. For general nonlinear forward operators, the data mismatch objective is non-convex, so gradient-based minimisation may converge to local rather than global minima [11].

Inverse problems are notoriously ill-posed, typically violating all three Hadamard conditions [8, 9]. Classical approaches address ill-posedness by augmenting the data mismatch objective with a penalty that favours certain solution properties. A common instance is Tikhonov regularisation, which seeks

$$a^* = \arg\min_a \left\{ \frac{1}{2} \|u_{\text{obs}} - \mathcal{F}(a)\|^2 + \frac{\alpha}{2} \|La\|^2 \right\}, \tag{2.1}$$

where $\alpha > 0$ is the regularisation parameter and L is a regularisation operator (e.g., $L = \nabla$ for smoothness). For an appropriate choice of α, it yields stable, unique solutions [19]. However, it produces a single point estimate with no quantification of uncertainty.

Tikhonov regularisation corresponds to MAP estimation under a Gaussian prior on a. The regularisation term $\frac{\alpha}{2}\|La\|^2$ is the negative log-density of this prior, so minimising the Tikhonov objective is equivalent to finding the MAP estimate under that prior [20]. This connection reveals that regularisation encodes prior information, and that the regularisation parameter α controls the strength of that prior belief. Chapter 3 identifies this same structure in IGNO's inversion objective and builds on it by adding a learned prior term for full posterior sampling.

The Bayesian approach to inverse problems reformulates inversion as computing a posterior measure over the function space of coefficient fields rather than seeking a single best estimate [8]. Rather than returning a single coefficient field, the posterior characterises the full distribution of fields consistent with the observed data, providing both a best estimate and a quantification of the remaining ambiguity. Section 2.3 formalises this approach. Figure 2.2 illustrates the resulting computational procedure.

Evaluating the likelihood for a single candidate coefficient field requires solving the forward PDE, and conventional MCMC methods require many such evaluations to adequately sample the posterior [12]. This computational bottleneck motivates the neural operator surrogates described next.

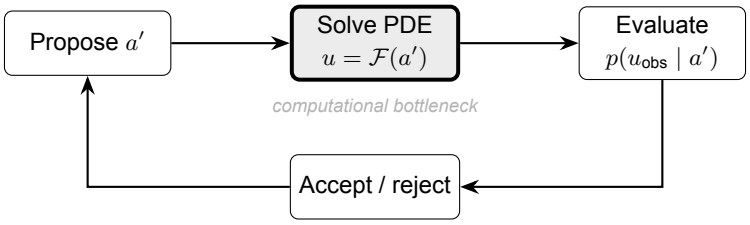


Figure 2.2: The computational procedure for Bayesian inversion. Each iteration proposes a candidate coefficient field a', solves the forward PDE to obtain the predicted solution, evaluates the likelihood by comparing the prediction to observations, and accepts or rejects the candidate.

2.2. Neural operators

Classical numerical methods solve the forward problem one instance at a time. For inverse problems requiring thousands of forward evaluations, this becomes computationally prohibitive. A neural operator is a parametric map $\mathcal{G}_\theta \colon \mathcal{A} \to \mathcal{U}$ between function spaces that is trained to approximate the solution operator $\mathcal{F}$ across a distribution of problem instances. Once trained, a single neural operator serves all inputs in the family, enabling forward simulation orders of magnitude faster than classical solvers. The operator acts on function-space inputs through pointwise evaluations at sensor locations, so it can be queried at arbitrary spatial resolutions without retraining [16].

The theoretical foundation for neural operators is the universal approximation theorem for operators, which shows that neural networks can approximate continuous nonlinear operators between function spaces to arbitrary accuracy [21].

DeepONet [14] splits operator learning into two sub-networks, one that encodes the coefficient function and one that encodes the query location (Figure 2.3). To approximate $u(\boldsymbol{x}) = \mathcal{F}(a)(\boldsymbol{x})$, a *branch* network $\mathbf{b}_\theta \colon \mathcal{A} \to \mathbb{R}^p$ encodes the coefficient function a, and a *trunk* network $\mathbf{t}_\psi \colon \Omega \to \mathbb{R}^p$ encodes the query location $\boldsymbol{x}$, where θ and ψ denote the trainable parameters of the branch and trunk networks, respectively. The output is their inner product:

$$\mathcal{G}_\theta(a)(\boldsymbol{x}) = \mathbf{b}_\theta(a)^\top \mathbf{t}_\psi(\boldsymbol{x}) = \sum_{k=1}^{p} b_k(a)\, t_k(\boldsymbol{x}). \tag{2.2}$$

The branch network processes the coefficient function via pointwise evaluations at N sensors, $\mathbf{b}_\theta([a(\boldsymbol{x}_1), \ldots, a(\boldsymbol{x}_N)])$.

Training minimises the squared L^2 error over a dataset $\{(a^{(i)}, u^{(i)})\}_{i=1}^{N_{\text{train}}}$ of solved PDE instances:

$$\min_{\theta,\psi} \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N_x} \left| u^{(i)}(\boldsymbol{x}_j) - \mathcal{G}_\theta(a^{(i)})(\boldsymbol{x}_j) \right|^2,$$

where $\{\boldsymbol{x}_j\}$ are query points sampled from Ω. A key property is that the trunk network accepts continuous spatial coordinates as input, so $u(\boldsymbol{x})$ can be evaluated at arbitrary query points $\boldsymbol{x}$ without retraining. This makes DeepONet *discretisation-invariant* on the output side. CNN-based operator methods require both input sensors and output locations to lie on equispaced grids [14], whereas DeepONet places no constraints on the locations of output evaluations. The branch network, however, requires the coefficient function to be sampled at a fixed set of sensor locations.

MultiONet extends DeepONet by aggregating branch-trunk inner products across multiple hidden layers rather than relying solely on the final layer [6]. For a branch network and trunk network each with l hidden layers, the MultiONet output is

$$\mathcal{G}(a)(\boldsymbol{x}) = \frac{1}{l} \sum_{k=1}^{l} w^{(k)} \left(\mathbf{b}^{(k)}(a) \cdot \mathbf{t}^{(k)}(\boldsymbol{x}) \right) + b_0, \tag{2.3}$$

where $\mathbf{b}^{(k)}$ and $\mathbf{t}^{(k)}$ are the k-th hidden layer outputs of the branch and trunk networks, $w^{(k)}$ are trainable scalar weights, and b_0 is a scalar bias. Setting $l = 1$, $w^{(1)} = 1$, and $b_0 = 0$ reduces Eq. 2.3 to the original DeepONet output of Eq. 2.2. Other neural operator architectures exist, notably the Fourier Neural

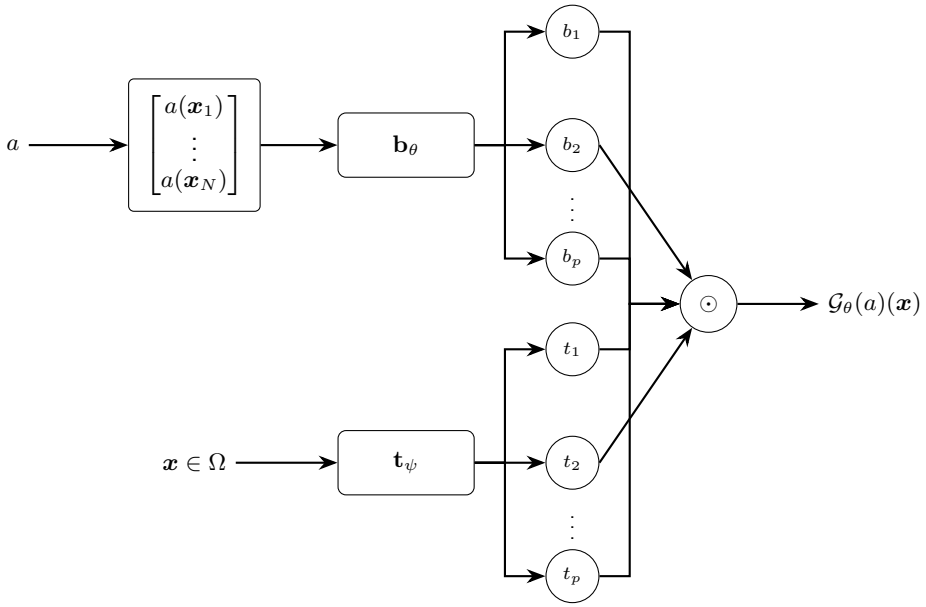


Figure 2.3: Architecture of DeepONet. The branch network $\mathbf{b}_\theta$ encodes the coefficient function a via pointwise evaluations at N fixed sensor locations and produces a p-dimensional output. The trunk network $\mathbf{t}_\psi$ encodes the query location $\boldsymbol{x} \in \Omega$. The operator output $\mathcal{G}_\theta(a)(\boldsymbol{x})$ is obtained as the inner product of the two outputs (Eq. 2.2).

Operator [15], which learns mappings in spectral space. We use the DeepONet family because its branch-trunk factorisation is useful for the latent-space generative modelling approach we build upon.

Neural operators provide fast, differentiable forward evaluation, but they are deterministic: for a given coefficient field, they return a single solution with no measure of uncertainty. One option is to place Bayesian posteriors over the network weights. Physics-informed neural networks (PINNs) encode PDE residuals directly into the training loss, enabling forward and inverse problem solving without simulation data, but they are instance-specific and require retraining for each new coefficient function [22]. Yang et al. extended PINNs to Bayesian inference (B-PINNs) by placing priors on the network weights and estimating the posterior via HMC or variational inference (VI) [23], which fits a simpler approximate distribution to the posterior instead of sampling from it directly [24]. Their experiments show that HMC produces accurate posterior samples, but VI with an approximate posterior that treats each weight as independent[2] yields "unreasonable uncertainties" [23]. Among neural operators, B-DeepONet applies Langevin sampling, another gradient-based MCMC variant, to the full parameter distribution, but weight-space inference scales poorly with the number of network parameters [25]. VB-DeepONet uses VI to reduce computational cost, but mean-field approximations struggle to capture complex posterior structure, consistent with the limitation Yang et al. observed for PINNs [26]. These results show that weight-space Bayesian inference in neural operators is computationally feasible only with approximations that break uncertainty calibration, motivating a different strategy in which the posterior is placed over a low-dimensional latent representation rather than the network weights themselves.

2.3. Bayesian inference

The inverse problem of Section 2.1 was stated as an optimisation problem. The Bayesian formulation instead treats the unknown coefficient field a as a random variable. Prior to observing data, our beliefs about a are encoded in a prior distribution $p(a)$. After observing data u_{obs}, we update our beliefs via Bayes' theorem:

$$p(a \mid u_{\text{obs}}) = \frac{p(u_{\text{obs}} \mid a)\, p(a)}{p(u_{\text{obs}})}. \tag{2.4}$$

[2]This is the *mean-field* approximation: the joint posterior over all parameters is replaced by a product of independent one-dimensional Gaussians, one per parameter. The resulting distribution cannot represent correlations between parameters and typically underestimates the true posterior spread [24].

The denominator is a normalising constant,

$$p(u_{\text{obs}}) = \int p(u_{\text{obs}} \mid a)\, p(a)\, da,$$

and the posterior $p(a \mid u_{\text{obs}})$ is the target density, quantifying the probability of every coefficient field consistent with both the observed data and prior knowledge.

Assuming independent homoscedastic Gaussian measurement noise at each sensor location, where $u_{\text{obs}}(x_i) = \mathcal{F}(a)(x_i) + \xi_i$ with $\xi_i \sim \mathcal{N}(0, \sigma^2_{\text{data}})$, each observation conditioned on a follows

$$u_{\text{obs}}(x_i) \mid a \;\sim\; \mathcal{N}\big(\mathcal{F}(a)(x_i),\, \sigma^2_{\text{data}}\big)\,.$$

Evaluating the Gaussian density gives the likelihood of a single observation:

$$p(u_{\text{obs}}(x_i) \mid a) = \frac{1}{\sqrt{2\pi\sigma^2_{\text{data}}}} \exp\left(-\frac{(u_{\text{obs}}(x_i) - \mathcal{F}(a)(x_i))^2}{2\sigma^2_{\text{data}}}\right).$$

Since the noise terms are independent, the joint likelihood over all M observations is the product of the individual likelihoods:

$$p(u_{\text{obs}} \mid a) = \prod_{i=1}^{M} \frac{1}{\sqrt{2\pi\sigma^2_{\text{data}}}} \exp\left(-\frac{|u_{\text{obs}}(x_i) - \mathcal{F}(a)(x_i)|^2}{2\sigma^2_{\text{data}}}\right).$$

Taking the negative logarithm,

$$\begin{aligned}
-\log p(u_{\text{obs}} \mid a) &= -\sum_{i=1}^{M} \log p(u_{\text{obs}}(x_i) \mid a) \\
&= \sum_{i=1}^{M} \left(\frac{|u_{\text{obs}}(x_i) - \mathcal{F}(a)(x_i)|^2}{2\sigma^2_{\text{data}}} + \frac{1}{2}\log(2\pi\sigma^2_{\text{data}})\right) \\
&= \frac{1}{2\sigma^2_{\text{data}}} \sum_{i=1}^{M} |u_{\text{obs}}(x_i) - \mathcal{F}(a)(x_i)|^2 + \frac{M}{2}\log(2\pi\sigma^2_{\text{data}}).
\end{aligned} \tag{2.5}$$

The second term does not depend on a and does not affect optimisation or sampling. The first term is the weighted sum of squared residuals with weight $1/(2\sigma^2_{\text{data}})$, which is the data mismatch in regularised inversion.

Taking the negative logarithm of Bayes' theorem (Eq. 2.4) separates the posterior into independent terms:

$$-\log p(a \mid u_{\text{obs}}) = \underbrace{-\log p(u_{\text{obs}} \mid a)}_{\text{data mismatch}} \underbrace{-\log p(a)}_{\text{prior}} + \log p(u_{\text{obs}}). \tag{2.6}$$

The last term is the logarithm of the normalising constant in Eq. 2.4 and does not depend on a. When $p(a)$ is Gaussian, minimising Eq. 2.6 recovers the Tikhonov objective of Eq. 2.1.

For optimisation, the prior acts only as a regulariser shaping the loss landscape, but posterior sampling additionally requires that the prior make the posterior integrable as a probability distribution. If the prior spreads probability too thinly, the posterior may not integrate to a finite value and therefore cannot be sampled from [8]. An informative prior simultaneously regularises the inverse problem and guarantees a well-defined posterior [12]. Its influence relative to the likelihood depends on the data. Sparse or noisy observations allow the prior to dominate, while abundant, high-quality data concentrate the posterior near the likelihood peak [8, 12].

Computing the posterior analytically is intractable for nonlinear forward operators, and the normalising constant $p(u_{\text{obs}})$ involves an integral over all possible coefficient fields. MCMC methods circumvent this by constructing a Markov chain whose stationary distribution equals the target posterior $p(a \mid u_{\text{obs}})$, requiring only unnormalised density evaluations [27, 28]. Even when the prior $p(a)$ and the noise

model $p(u_{\text{obs}} \mid a)$ are both Gaussian, the posterior $p(a \mid u_{\text{obs}}) \propto p(u_{\text{obs}} \mid a)\, p(a)$ is generally non-Gaussian whenever the forward operator $\mathcal{F}$ is nonlinear. The likelihood $p(u_{\text{obs}} \mid a) = \mathcal{N}(\mathcal{F}(a), \sigma^2_{\text{data}} I)$ is a Gaussian in u_{obs} but a nonlinear, non-Gaussian function of a through the map $a \mapsto \mathcal{F}(a)$ [8, 29]. A direct consequence is that methods which parametrise the approximate posterior as a Gaussian family, including mean-field variational inference and Laplace approximations, are in general misspecified and tend to underestimate posterior spread, producing overconfident uncertainty estimates [17]. MCMC methods, by contrast, make no parametric assumption on the posterior shape and are asymptotically exact; this is the main theoretical motivation for the sampling approach developed in Section 3.4.

2.4. Normalising flows

The previous section established that posterior sampling requires an informative prior with a tractable density. Normalising flows construct a bijective, differentiable transformation $f : \mathbb{R}^d \to \mathbb{R}^d$ that maps a complex data distribution $p_x(x)$ to a simple base distribution $p_z(z)$ [30, 31]. Since f is bijective, the inverse f^{-1} maps from the base distribution back to the data distribution. Given $z = f(x)$, the change of variables formula gives the data density:

$$p_x(x) = p_z(f(x)) \left| \det \frac{\partial f}{\partial x}(x) \right|. \tag{2.7}$$

The map f, its inverse f^{-1}, and the log-Jacobian determinant $\log |\det \partial f / \partial x|$ must therefore all be cheap to evaluate.

Dinh et al. [30] achieve this by constructing f as a composition of K invertible layers, $f = f_K \circ \cdots \circ f_1$, each designed so that its Jacobian determinant is cheap to evaluate individually. By the chain rule, the log-Jacobian determinant decomposes as a sum over layers:

$$\log \left| \det \frac{\partial f}{\partial x}(x) \right| = \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial h_{k-1}}(h_{k-1}) \right|,$$

where $h_0 = x$ and $h_k = f_k(h_{k-1})$. In the coupling layer [30], the input is partitioned into two groups. One group passes through unchanged, while the other is transformed by an arbitrary function conditioned on the first. Because the first group is an identity mapping, the Jacobian of each layer is triangular, and its determinant reduces to a product of diagonal entries. Expressivity comes from stacking many such layers and alternating which group is transformed. The specific coupling layer architecture used in this work is described in Appendix E.

Because the change of variables formula provides exact log-likelihoods, a normalising flow with parameters ϕ can be trained by maximum likelihood. Given samples $\{x^{(i)}\}_{i=1}^{N}$ from a data distribution, training maximises the mean log-likelihood:

$$\mathcal{L}(\phi) = \frac{1}{N} \sum_{i=1}^{N} \left(\log p_z(f_\phi(x^{(i)})) + \log \left| \det \frac{\partial f_\phi}{\partial x}(x^{(i)}) \right| \right).$$

A trained flow supports both exact density evaluation (compute $\log p_x(x)$ via a forward pass) and sampling (draw $z \sim p_z$, compute $x = f^{-1}(z)$). Both operations are exact, making normalising flows suitable as learned prior distributions in Bayesian inference [32, 33]. The log-prior density and its gradient with respect to the input are available in closed form at any point, as required by gradient-based MCMC samplers. This contrasts with variational autoencoders, where the marginal likelihood requires a lower-bound approximation and cannot be evaluated exactly [34]. The base distribution and flow architecture used in this work are described in Appendix E.

2.5. Latent-space approaches to Bayesian inversion

Recent work has explored Bayesian inversion in the latent space of a learned generative model rather than in the original high-dimensional parameter space. The central motivation, articulated by Meng et al. [35], is that Bayesian inference in a generative model's latent space is far more tractable than in the

original parameter space, because the surrogate compresses the problem to a small number of latent variables while preserving the structure needed for posterior sampling.

These methods share a common template but differ along two axes that determine the quality of the resulting posterior, namely how the posterior is approximated and what role physics plays during inference as opposed to training alone. Table 2.1 summarises the resulting landscape.

Table 2.1: Comparison of latent-space Bayesian methods for PDE inverse problems.

Method	Gen. model	Prior	Posterior	Retrain?	Phys. in post.?	Exact?[a]
OL-GAN [36]	WGAN-GP	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	MH-MCMC	No	Data only	Yes
PI-GAN+MCMC [35]	PI-GAN	$\mathcal{N}(\mathbf{0}, \mathbf{I})$	MCMC (NUTS)	No	PINN/DeepONet	Yes
GAN-Flow [32]	WGAN-GP	GAN	NF-VI	Yes (NF)	Forward model	No
iDeepONet [33]	Inv. DeepONet	GMM	Semi-analytic	No	PI-DeepONet	No
PDDLVM [17]	VAE-like	Specified	Mean-field VI	Partial	PDE residual	No
Ours	**IGNO**	**NF**	**MCMC (NUTS)**	**No**	**Weak-form res.**	**Yes**

[a]Refers to whether the inference method is asymptotically exact (MCMC). For GAN-based methods, the generator itself only approximates the true distribution, so the posterior sampled by MCMC is exact only with respect to the learned model.

Several methods use MCMC to sample the posterior exactly with respect to the learned model. OL-GAN [36] learns the joint distribution of PDE coefficients and solutions via a Wasserstein GAN with a DeepONet-style generator, then runs Metropolis-Hastings in the latent space. Physics enters only through the training dataset of solved forward problem instances, so the governing equations play no role during posterior inference. Meng et al. [35] take the opposite approach, embedding physics-informed residuals from a PINN or DeepONet surrogate directly into the MCMC likelihood so that the PDE is enforced during sampling. Both methods use isotropic Gaussian priors on the latent variables, which may not capture the true encoded distribution. Their subsequent work replaced MCMC with normalising flow-based variational inference, improving scalability at the cost of exact sampling [37]. The tractability of latent-space MCMC also depends on the dimensionality of the representation, as discussed below.

Other methods trade the asymptotic exactness of MCMC for lower computational cost. Dasgupta et al. [32] use a WGAN as a data-driven prior and fit a normalising flow as a variational approximation to the posterior. The authors report being "unsuccessful in obtaining convergence of the Markov chains" when attempting MCMC at 512-dimensional latent spaces, motivating a VI-based approach. While this sidesteps the convergence issue, the reverse KL divergence objective used for VI is mode-seeking, potentially missing posterior modes and underestimating uncertainty in tails [24, 32], and the posterior network must be retrained for each new observation. Kaltenbach et al. [33] make the branch network of a DeepONet invertible using RealNVP coupling blocks and fit a Gaussian mixture as the prior, obtaining a semi-analytic posterior at the cost of restricting it to this parametric family. The Physics-Driven Deep Latent Variable Model [17] uses mean-field variational inference, which is "known to be overconfident" [17] because the independence assumption cannot represent correlations in the true posterior. For all three methods, physics enters only during training, not during posterior inference.

To the best of our knowledge, no existing method combines a learned prior, exact posterior sampling, and physics constraints during inference. We extend IGNO [7] to full Bayesian inference by adding a normalising flow prior term and replacing deterministic optimisation with MCMC sampling, as detailed in Section 3.4.

3

Bayesian IGNO

This chapter describes the architecture, training procedure, and deterministic inversion objective of IGNO [7], before presenting the Bayesian extension that enables full posterior sampling.

3.1. Architecture

The IGNO architecture is illustrated in Figure 3.1, with the inference configuration shown in Figure 3.2. The coefficient encoder E_{θ_a} maps a coefficient field $a \in \mathcal{A}$ to a latent representation $\beta_1 \in \mathbb{R}^{d_1}$, where $d_1 \ll \dim(\mathcal{A})$ is a problem-dependent hyperparameter (see Appendix F for concrete values). This β_1 is the variable we optimise or sample during inversion. The boundary condition encoder $E_{\theta_{bc}} : \text{BC} \to \mathbb{R}^{d_2}$ maps boundary condition information into a fixed-dimensional representation $\beta_2 \in \mathbb{R}^{d_2}$; its form is problem-dependent and need not be a neural network. Two MultiONet-based decoders reconstruct physical fields from these latent representations. The solution decoder G_{θ_u} predicts the solution $u(\boldsymbol{x})$ at spatial location $\boldsymbol{x} \in \Omega$ given the concatenated latent variable $\boldsymbol{\beta} = (\beta_1, \beta_2)$, mapping $\mathbb{R}^{d_1+d_2} \times \Omega \to \mathbb{R}$:

$$\tilde{u}(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x}) = G_{\theta_u}((\beta_1, \beta_2), \boldsymbol{x}).$$

The coefficient decoder $G_{\theta_a} : \mathbb{R}^{d_1} \to \mathcal{A}$ reconstructs the coefficient field from the latent representation via $\tilde{a} = G_{\theta_a}(\beta_1)$. Both decoders use the MultiONet architecture and are discretisation-invariant. A normalising flow $F_\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}$ learns a bijective mapping between the latent distribution and a fixed base distribution (see Appendix E for architecture details). In the original IGNO framework, the flow is used only for initialisation via inverse sampling.

3.2. Training

IGNO is trained on a dataset of coefficient fields $\{a^{(i)}\}_{i=1}^{N_{\text{train}}}$ following the probabilistic framework introduced by DGenNO [6]. DGenNO introduces latent variables and derives the training objective from the variational EM framework [38]. The latent variable β_1 defines conditional distributions over both the coefficient field and the solution field. The coefficient decoder defines a Gaussian likelihood for continuous problems,

$$p(a \mid \beta_1) = \mathcal{N}(G_{\theta_a}(\beta_1),\, \sigma^2_{\text{rec}} I),$$

For piecewise-constant problems where $a(\boldsymbol{x}_i) \in \{k_{\text{low}}, k_{\text{high}}\}$, the coefficient decoder instead defines a Bernoulli likelihood. Let $p_i = \text{sigmoid}(G_{\theta_a}(\beta_1, \boldsymbol{x}_i))$ denote the predicted probability that location $\boldsymbol{x}_i$ belongs to the high-permeability phase, $a(\boldsymbol{x}_i) = k_{\text{high}}$. With binary labels $a_i^* = (a(\boldsymbol{x}_i) - k_{\text{low}})/(k_{\text{high}} - k_{\text{low}}) \in \{0, 1\}$, the likelihood is

$$p(a \mid \beta_1) = \prod_i p_i^{a_i^*} (1 - p_i)^{1 - a_i^*}. \tag{3.1}$$

The solution decoder is deterministic, mapping each latent code to a single solution field, formalised as

$$p(u \mid \boldsymbol{\beta}) = \delta(u(\boldsymbol{x}) - G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})).$$

The third component is a virtual likelihood for PDE residuals, following the construction introduced by Kaltenbach and Koutsourelakis [39] and extended by Rixner and Koutsourelakis [40]. The weak-form residuals $r_{w_j}(a, \tilde{u})$ are computed via the ParticleWNN framework [41], which replaces strong-form collocation with integration against compactly supported test functions; full derivations for each benchmark appear in Appendix A. The coefficient field a enters the integrand only as a pointwise multiplier, so no derivatives of a are required and a may be discontinuous. The residuals $r_{w_j}(a, \tilde{u})$, where $\tilde{u} = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$, are treated as data virtually observed to equal zero, written $\hat{R}_M = \{\hat{r}_j = 0\}_{j=1}^{N_c}$. Each residual is modelled as an independent observation of zero under a Gaussian noise model with variance σ^2_{pde}, giving the virtual likelihood

$$\begin{aligned} p(\hat{R}_M \mid a, \beta_1) &= \prod_{j=1}^{N_c} p(\hat{r}_j = 0 \mid a, \tilde{u}) \\ &= \prod_{j=1}^{N_c} \frac{1}{\sqrt{2\pi\sigma^2_{\mathsf{pde}}}} \exp\left(-\frac{r_{w_j}(a, \tilde{u})^2}{2\sigma^2_{\mathsf{pde}}}\right), \end{aligned} \tag{3.2}$$

where $w_{\mathsf{pde}} = 1/(2\sigma^2_{\mathsf{pde}})$ and $\tilde{u}$ is a deterministic function of β_1. Together with the reconstruction likelihood and prior, this defines a joint probability that requires marginalising over β_1:

$$p(a, \hat{R}_M) = \int p(\hat{R}_M \mid a, \beta_1)\, p(a \mid \beta_1)\, p(\beta_1)\, d\beta_1. \tag{3.3}$$

Rather than optimising a variational distribution for each training instance, the encoder maps coefficient fields to a single latent point via the degenerate posterior $q(\beta_1 \mid a) = \delta(\beta_1 - E_{\theta_a}(a))$ [6]. Training maximises $\log p(a, \hat{R}_M)$. Since the integral in Eq. 3.3 is intractable, DGenNO uses $q(\beta_1 \mid a)$ as an auxiliary density, following the variational EM framework [38], to derive the training objective:

$$\begin{aligned} \log p(a, \hat{R}_M) &\geq \mathbb{E}_q\left[\log \frac{p(\hat{R}_M \mid a, \beta_1)\, p(a \mid \beta_1)\, p(\beta_1)}{q(\beta_1 \mid a)}\right] \\ &= \mathbb{E}_q\Big[\log p(\hat{R}_M \mid a, \beta_1)\Big] + \mathbb{E}_q[\log p(a \mid \beta_1)] - \mathsf{KL}(q(\beta_1 \mid a) \,\|\, p(\beta_1))\,. \end{aligned} \tag{3.4}$$

Under the degenerate posterior $q(\beta_1 \mid a) = \delta(\beta_1 - E_{\theta_a}(a))$, each expectation collapses to a point evaluation at $\beta_1 = E_{\theta_a}(a)$. Under a uniform prior on β_1, Zang et al. argue that the KL term does not depend on the network parameters and drops from the optimisation[1] [6]. Each remaining term in Eq. 3.4 is then a log-likelihood evaluated at the point mass $\beta_1 = E_{\theta_a}(a)$. The reconstruction likelihood is the Gaussian density $\mathcal{N}(G_{\theta_a}(\beta_1), \sigma^2_{\mathsf{rec}} I)$, which satisfies

$$p(a \mid \beta_1) \propto \exp\big(-w_{\mathsf{rec}} \|a - G_{\theta_a}(\beta_1)\|^2\big)\,,$$

where $w_{\mathsf{rec}} = 1/(2\sigma^2_{\mathsf{rec}})$ and the proportionality constant, the Gaussian normalisation, does not depend on β_1 or the network parameters. The term $\mathcal{F}_{\mathsf{rec}} = w_{\mathsf{rec}} \|a - G_{\theta_a}(\beta_1)\|^2$ is therefore a weighted squared L^2 reconstruction error. Similarly, the virtual likelihood (Eq. 3.2) satisfies

$$p(\hat{R}_M \mid a, \beta_1) \propto \exp\left(-w_{\mathsf{pde}} \sum_{j=1}^{N_c} r_{w_j}(a, \tilde{u})^2\right),$$

so $\mathcal{F}_{\mathsf{pde}} = w_{\mathsf{pde}} \sum_{j=1}^{N_c} r_{w_j}(a, \tilde{u})^2$ is a weighted sum of squared weak-form residuals. Combining these, the training loss is

$$\mathcal{L}_{\mathsf{train}} = \mathcal{F}_{\mathsf{pde}} + \mathcal{F}_{\mathsf{rec}},$$

[1] Strictly, $\mathsf{KL}(\delta_x \| p)$ is infinite for a point mass against any continuous density. The continuous prior assigns probability zero to the singleton on which the point mass concentrates all its mass, so the KL divergence is infinite. The ELBO in Eq. 3.4 is therefore not finite. However, since the KL term does not depend on the network parameters, the infinite term is identical for all parameter values and has no effect when comparing objectives. What is effectively optimised is the joint log-likelihood evaluated at the encoder output.

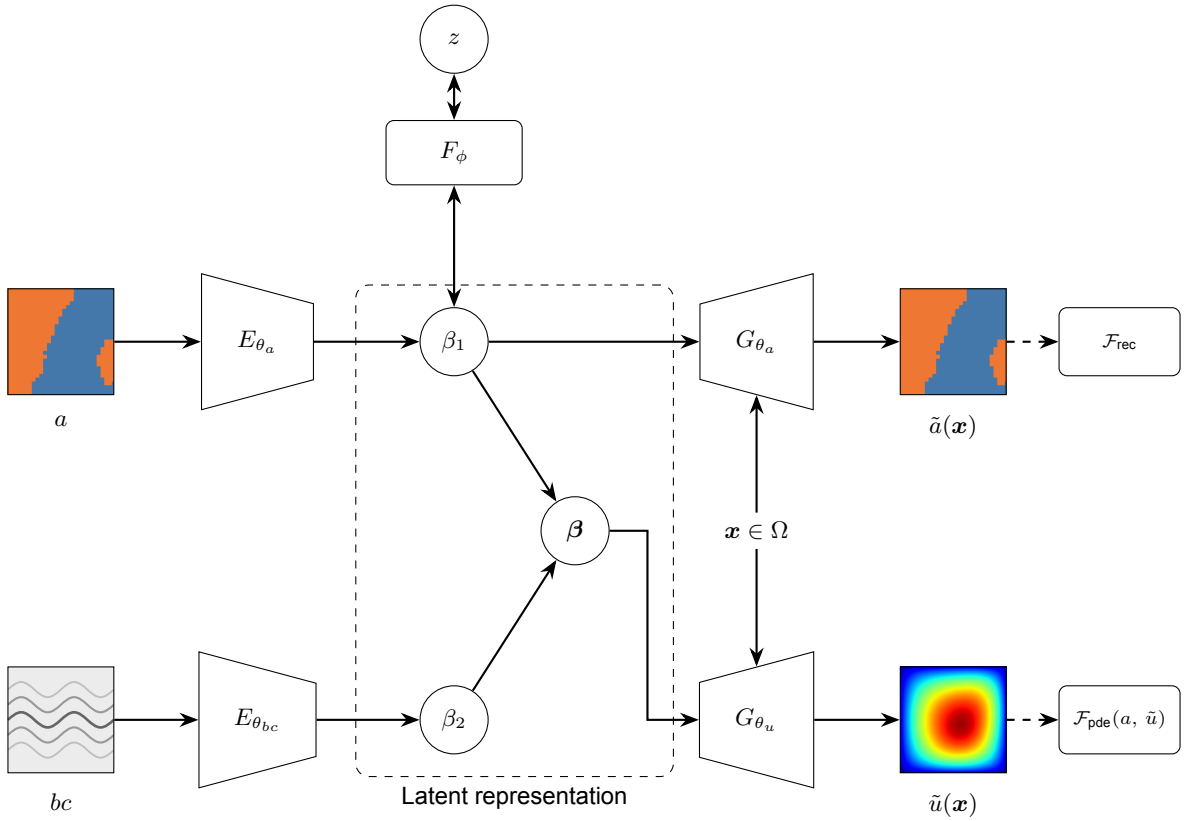


Figure 3.1: IGNO training. The coefficient encoder E_{θ_a} maps each training coefficient field a to a latent code β_1, and the boundary condition encoder $E_{\theta_{bc}}$ maps boundary conditions to β_2. The concatenated latent $\boldsymbol{\beta} = (\beta_1, \beta_2)$ is passed to the solution decoder G_{θ_u}, which produces $\tilde{u}(\boldsymbol{x})$ at query location $\boldsymbol{x} \in \Omega$. The coefficient decoder G_{θ_a} reconstructs $\tilde{a}(\boldsymbol{x})$ from β_1 alone. Dashed arrows indicate loss computation: $\mathcal{F}_{\text{rec}}$ measures the reconstruction error between $\tilde{a}$ and the true a, while $\mathcal{F}_{\text{pde}}$ evaluates weak-form PDE residuals using the true a in the PDE integrand. All components are trained jointly. The normalising flow F_ϕ learns the distribution of encoded β_1 values via a bijective mapping to the base variable z.

where $\tilde{a} = G_{\theta_a}(\beta_1)$, $\tilde{u} = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$, and $\beta_1 = E_{\theta_a}(a)$. Minimising $\mathcal{L}_{\text{train}}$ is equivalent to maximising the bound in Eq. 3.4. Substituting the definitions of $\mathcal{F}_{\text{pde}}$ and $\mathcal{F}_{\text{rec}}$,

$$\mathcal{L}_{\text{train}} = w_{\text{pde}} \sum_{j=1}^{N_c} r_{w_j}(a, \tilde{u})^2 + w_{\text{rec}} \|a - G_{\theta_a}(\beta_1)\|^2.$$

In practice, this objective is a weighted sum of PDE residual and reconstruction losses, optimised jointly via Adam over all encoder and decoder parameters rather than the alternating VB-EM updates of DGenNO [6]. Because the PDE residual provides supervision through the governing equations themselves, training requires no precomputed solutions.

IGNO extends DGenNO in two ways [7]. First, it introduces the boundary condition encoder $E_{\theta_{bc}}$, enabling problems where the data are not pointwise solution observations but operator-valued measurements. In such problems, multiple boundary conditions are applied to the domain, and the observed data are the resulting boundary responses. The encoder maps each boundary condition to a latent representation β_2, so that a single trained model can process all boundary conditions by varying β_2 while sharing the same coefficient latent β_1 (see Section 4.1 for a concrete instance). Second, it adds a normalising flow F_ϕ trained on the encoded latent representations (see Appendix E).

The trained components form a generative system in which β_1 is not merely a compressed representation of a in the sense of a dimensionality reduction $a \to \beta_1 \to a$. Instead, β_1 acts as a shared generator of both a and u, with the decoders jointly mapping a single latent code to a physically consistent coefficient–solution pair through the bidirectional structure $a \leftarrow \beta_1 \to u$ [6]. The normalising flow F_ϕ models the distribution of β_1 values induced by the training data, which IGNO uses to produce a

statistically informed initialisation for inversion by drawing many samples from the flow and taking their mean,

$$\beta_{1,\text{init}} = \frac{1}{S}\sum_{s=1}^{S} F_\phi^{-1}(z_s), \quad z_s \sim p_z.$$

This mean-of-samples initialisation differs from the single-sample draw described in the original IGNO algorithm [7] and provides a more stable starting point by averaging out flow sampling noise. After training, all components are fixed and no retraining occurs when solving new inverse problems. Inversion thus reduces to finding the β_1 whose decoded outputs best match the observed data.

3.3. Deterministic inversion

Given observations u_{obs}, IGNO seeks β_1^* via gradient-based optimisation. The objective combines data mismatch and PDE residual:

$$\beta_1^* = \arg\min_{\beta_1} \mathcal{F}_{\text{data}}(\beta_1) + \mathcal{F}_{\text{pde}}(\beta_1), \tag{3.5}$$

where

$$\mathcal{F}_{\text{data}}(\beta_1) = w_{\text{data}} \sum_{i=1}^{M} \left| u_{\text{obs}}(\boldsymbol{x}_i) - G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x}_i) \right|^2, \tag{3.6}$$

$$\mathcal{F}_{\text{pde}}(\beta_1) = w_{\text{pde}} \sum_{j=1}^{N_c} r_{w_j}(\tilde{a}, \tilde{u})^2.$$

The weak-form residuals r_{w_j} in $\mathcal{F}_{\text{pde}}$ are as defined in Section 3.2, with the coefficient and solution fields now provided by $\tilde{a} = G_{\theta_a}(\beta_1)$ and $\tilde{u} = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$ evaluated at the current β_1. Optimisation proceeds via Adam with gradients computed through automatic differentiation [42]. The full algorithm and per-problem hyperparameters are given in Appendix F.

Since both $\tilde{a}$ and $\tilde{u}$ are deterministic functions of β_1 (with β_2 determined by the boundary conditions), the likelihoods from Section 3.2 simplify during inversion. The data likelihood depends on β_1 through $\tilde{u}$, and with $w_{\text{data}} = 1/(2\sigma_{\text{data}}^2)$ as in Eq. 2.5, we have $p(u_{\text{obs}} \mid \beta_1) \propto \exp(-\mathcal{F}_{\text{data}}(\beta_1))$. The virtual likelihood (Eq. 3.2), which conditions on both a and β_1 during training, reduces during inversion to depend on β_1 alone since $\tilde{a} = G_{\theta_a}(\beta_1)$, with $p(\hat{R}_M \mid \beta_1) \propto \exp(-\mathcal{F}_{\text{pde}}(\beta_1))$.

After optimisation, IGNO returns β_1^*, the inferred coefficient $\tilde{a}^* = G_{\theta_a}(\beta_1^*)$, and predicted solution $\tilde{u}^*(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}^*, \boldsymbol{x})$. This is a single point estimate with no uncertainty quantification. The inference diagram in Figure 3.2 illustrates the Bayesian extension. The deterministic case follows the same data flow but without the flow prior term $\mathcal{F}_{\text{prior}}$, and with β_1 optimised rather than sampled.

3.4. Posterior formulation

We now extend the inversion to full Bayesian posterior sampling by adding an explicit prior from the normalising flow and replacing optimisation with MCMC, using IGNO's already-trained components with no retraining.

The inversion objective in Eq. 3.5 inherits probabilistic structure from IGNO's training framework. As established in Section 3.3, both the data likelihood and the virtual likelihood (Eq. 3.2) depend on β_1 alone during inversion, with $p(u_{\text{obs}} \mid \beta_1) \propto \exp(-\mathcal{F}_{\text{data}}(\beta_1))$ and $p(\hat{R}_M \mid \beta_1) \propto \exp(-\mathcal{F}_{\text{pde}}(\beta_1))$. Under a uniform prior on β_1, the inversion objective is equivalent to maximum likelihood estimation over these two terms. Since $\beta_1 \in [-1, 1]^{d_1}$ by construction, a uniform prior on this bounded domain already yields a well-defined posterior. However, such a prior is uninformative, treating all latent codes as equally plausible regardless of whether they correspond to physically realistic coefficient fields.

We obtain a proper prior from IGNO's already-trained normalising flow. Applying the change-of-variables formula (Eq. 2.7) to F_ϕ gives an exact prior density on β_1:

$$p(\beta_1) = p_z(F_\phi(\beta_1)) \left| \det J_{F_\phi}(\beta_1) \right|,$$

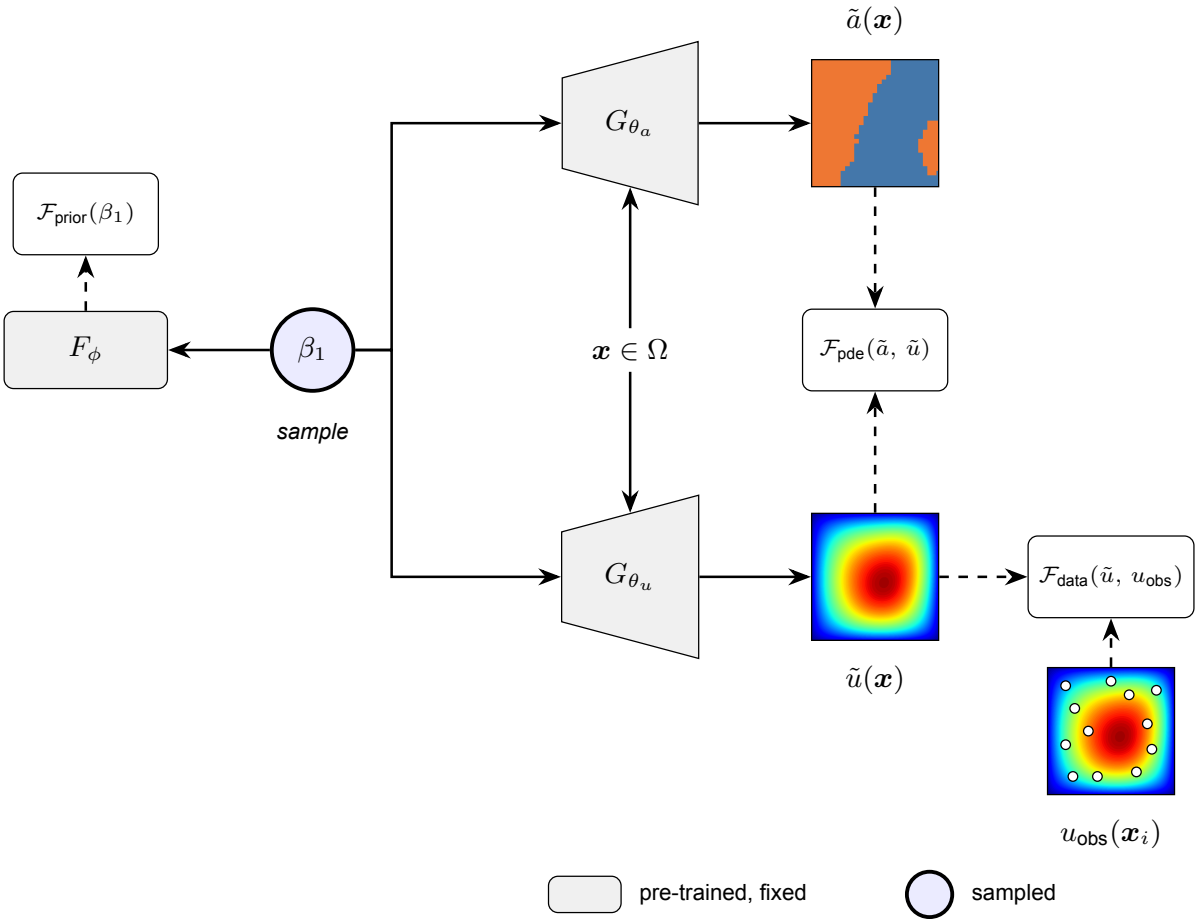


Figure 3.2: Bayesian inference. All network components are pre-trained and fixed. The latent variable β_1 is sampled via MCMC. The data loss $\mathcal{F}_{\text{data}}$ compares the predicted solution $\tilde{u}$ against observed data u_{obs} at sensor locations, while $\mathcal{F}_{\text{pde}}$ enforces PDE consistency using the decoded coefficient $\tilde{a}$. The flow prior $\mathcal{F}_{\text{prior}}$ evaluates the learned normalising flow density at the current β_1. Coefficient and boundary condition encoders are not used during inversion.

where p_z is the base distribution density (see Appendix E) and $J_{F_\phi}(\beta_1) = \frac{\partial F_\phi}{\partial \beta_1}(\beta_1)$ is the Jacobian. Taking the negative logarithm yields:

$$\mathcal{F}_{\text{prior}}(\beta_1) = -\log p_z(F_\phi(\beta_1)) - \log \left|\det J_{F_\phi}(\beta_1)\right|.$$

This is the only new term we add to IGNO's objective for Bayesian sampling.

In IGNO, the flow served only for initialisation (Section 3.3), contributing no prior term to the objective. For MCMC sampling, we repurpose it as a prior. Adding $\mathcal{F}_{\text{prior}}$ replaces the uninformative uniform prior with a learned density that concentrates mass on the region of latent space where realistic coefficient fields live, regularising β_1 towards the training distribution and improving MCMC mixing.

3.5. Full posterior derivation

The preceding section established three probabilistic components, namely the data likelihood $p(u_{\text{obs}} \mid \beta_1)$, the virtual likelihood $p(\hat{R}_M \mid \beta_1)$, and the normalising flow prior $p(\beta_1)$. These combine into two natural posterior formulations depending on whether one conditions on PDE residuals in addition to the observations [6].

The first formulation conditions only on the observed data. By Bayes' theorem,

$$p(\beta_1 \mid u_{\text{obs}}) \propto p(u_{\text{obs}} \mid \beta_1) \cdot p(\beta_1). \tag{3.7}$$

MCMC requires the posterior only up to its normalisation, so the potential energy is defined as the negative log of the unnormalised posterior in Eq. 3.7 [28]. Substituting the proportional form of the data

likelihood and $\mathcal{F}_{\text{prior}} = -\log p(\beta_1)$ gives the data-only potential energy

$$U_{\text{data}}(\beta_1) = \mathcal{F}_{\text{data}}(\beta_1) + \mathcal{F}_{\text{prior}}(\beta_1),$$

so that $p(\beta_1 \mid u_{\text{obs}}) = \frac{1}{Z_{\text{data}}} \exp(-U_{\text{data}}(\beta_1))$ with $Z_{\text{data}} = \int \exp(-U_{\text{data}}(\beta_1))\, d\beta_1$. This formulation relies solely on the observation likelihood and the learned prior, without enforcing PDE consistency during sampling.

One can also condition on the virtual observables $\hat{R}_M$, incorporating PDE residuals as an additional constraint. The joint posterior is

$$p(\beta_1 \mid u_{\text{obs}}, \hat{R}_M) \propto p(u_{\text{obs}} \mid \beta_1) \cdot p(\hat{R}_M \mid \beta_1) \cdot p(\beta_1),$$

where $p(\hat{R}_M \mid \beta_1)$ is the virtual likelihood from Eq. 3.2. Including this term, the full potential energy is

$$U(\beta_1) = \mathcal{F}_{\text{data}}(\beta_1) + \mathcal{F}_{\text{pde}}(\beta_1) + \mathcal{F}_{\text{prior}}(\beta_1),$$

so that $p(\beta_1 \mid u_{\text{obs}}, \hat{R}_M) = \frac{1}{Z} \exp(-U(\beta_1))$ with $Z = \int \exp(-U(\beta_1))\, d\beta_1$.

Both formulations give valid, well-posed posteriors. The data-only formulation avoids the cost of evaluating PDE residuals at each MCMC step, while the joint formulation additionally enforces physical consistency during sampling. The relative accuracy and calibration of the two formulations depends on the problem setting and is evaluated empirically in Chapter 4.

Both posteriors are well-posed under the conditions for Bayesian inverse problems established by Stuart [8]. The first condition is that the normalisation Z is finite and positive, so that the posterior is a proper probability distribution. This holds because β_1 is restricted to $[-1, 1]^{d_1}$ (Appendix E) and each term in U is a loss computed from neural network outputs, so we integrate a finite-valued function over a finite volume. Positivity of Z follows from $\exp(-U(\beta_1)) > 0$ for all β_1. The second condition is that small perturbations in the observed data produce at most proportionally small changes in the posterior. This holds because only $\mathcal{F}_{\text{data}}$ depends on u_{obs}, and since $\mathcal{F}_{\text{data}}$ is a squared difference between predictions and observations (Eq. 3.6), a small change in u_{obs} produces a correspondingly small change in the potential. These conditions hold even under a uniform prior on $[-1, 1]^{d_1}$, since the potential is bounded and the domain is finite. The flow prior's value is not well-posedness but informativeness. A uniform prior assigns equal density to all latent codes in the hypercube, including regions far from the training distribution where the decoded fields are unreliable. The flow prior concentrates mass on the manifold that the training data actually occupies, producing better-calibrated posteriors and faster MCMC mixing.

We do not need to compute Z for MCMC sampling. Our goal is to generate samples $\{\beta_1^{(i)}\}_{i=1}^N$ from this posterior. These samples encode the full uncertainty. Decoding each $\beta_1^{(i)}$ via $\tilde{a}^{(i)} = G_{\theta_a}(\beta_1^{(i)})$ and $\tilde{u}^{(i)}(\cdot) = G_{\theta_u}(\beta_1^{(i)}, \beta_2, \cdot)$ yields an ensemble of plausible coefficient and solution fields consistent with observations and prior knowledge.

We sample with NUTS [43]. Each NUTS step requires $\nabla_{\beta_1} U$, which decomposes into the gradients of the individual terms and is computed via automatic differentiation. The full sampling procedure is given in Algorithm 1.

4

Experiments

4.1. Problem formulation

We apply our method to four benchmark inverse problems spanning three measurement modalities (Table 4.1), following the experimental setup of [6, 7]. In solution-based problems, we observe u at M sensor locations corrupted by additive Gaussian noise at a given signal-to-noise ratio (see Eq. 4.2). In operator-based problems, we observe operator-valued data such as Dirichlet-to-Neumann maps from multiple boundary conditions. In all cases, synthetic test observations are generated by solving the governing PDE with a finite element method and adding noise at the specified level.

The first two benchmarks are Darcy flow problems governed by

$$-\nabla \cdot (k(\boldsymbol{x})\nabla p(\boldsymbol{x})) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega = [0,1]^2, \qquad p = 0 \text{ on } \partial\Omega,$$

where k is the unknown permeability field, p is the pressure, and $f = 10$ is a constant source term. Measurements consist of pressure values at $M = 100$ randomly placed interior sensors. In the *continuous* variant, permeability fields are $k(\boldsymbol{x}) = 2.1 + \sin(\omega_1 x_1) + \cos(\omega_2 x_2)$ with $(\omega_1, \omega_2) \sim \mathcal{U}(0, 7\pi/4)^2$. For OOD evaluation, $(\omega_1, \omega_2) \sim \mathcal{U}(7\pi/4, 2\pi)^2$. In the *piecewise-constant* variant, $k \in \{5, 10\}$ with phase geometry determined by a thresholded Gaussian process $\mathcal{GP}(0, (-\Delta + 9I)^{-2})$[1]. The OOD test set draws phase geometry from $\mathcal{GP}(0, (-\Delta + 16I)^{-2})$, producing finer-scale patterns with the same coefficient values. For this variant, the coefficient field is binary, and the coefficient decoder models it as a classification problem trained with a cross-entropy loss rather than predicting permeability values directly. Both Darcy variants and their coefficient distributions follow [6, 7], with the coefficient encoder receiving the permeability field evaluated on a 29×29 uniform grid.

The third benchmark is the electrical impedance tomography (EIT) problem, an inverse problem with an indirect observation model:

$$-\nabla \cdot (\gamma(\boldsymbol{x})\nabla u(\boldsymbol{x})) = 0, \quad \boldsymbol{x} \in \Omega = [0,1]^2, \qquad u = g \text{ on } \partial\Omega, \tag{4.1}$$

where $\gamma > 0$ is the unknown conductivity. Unlike the Darcy problems, where the solution field p is observed at interior sensors, EIT recovers γ from measurements taken only at the domain boundary. The measurement procedure works as follows: a voltage g is applied on the boundary, this boundary condition together with the interior conductivity γ determines the electric potential u throughout the domain via Eq. 4.1, and the resulting electrical current leaving the boundary is recorded at sensor locations. Mathematically, this boundary current is $\gamma \nabla u \cdot \vec{n}$, where $\vec{n}$ is the outward unit normal. The mapping from an applied boundary voltage g to the resulting boundary current defines the Dirichlet-to-Neumann map, $\Lambda_\gamma \colon g \mapsto \gamma \frac{\partial u}{\partial \vec{n}}\big|_{\partial\Omega}$. Because different conductivity fields route current differently, the DtN map encodes information about the interior conductivity γ. To extract sufficient information, $L = 20$ distinct voltage patterns are applied, each defined as $g_l(\boldsymbol{x}) = \cos(2\pi(x_1 \cos\theta_l + x_2 \sin\theta_l))$ with angles

[1] A random field is sampled from the GP and thresholded at zero: $k(\boldsymbol{x}) = 10$ where the sample is positive, $k(\boldsymbol{x}) = 5$ otherwise [6, 15].

Table 4.1: Dataset summary for each benchmark. N_{train} is the number of training coefficient fields, d_1 is the latent dimension of β_1, and M is the number of sensor locations per observation. Latent dimensions were chosen empirically per problem.

Problem	N_{train}	d_1	Encoder input	M (sensors)
Darcy continuous	1000	6	29×29 grid	100
Darcy piecewise	10000	200	29×29 grid	100
EIT	1000	6	32×32 grid	124×20 BCs
Burgers	1000	16	128 points	100

$\theta_l = \pi l/20$ for $l = 1, \ldots, 20$, and the current response to each pattern is recorded at $M = 124$ uniformly spaced boundary sensors.

Conductivity fields take the form $\gamma(\boldsymbol{x}) = \sum_{k=1}^{K} \exp(c_k \sin(k\pi x_1) \sin(k\pi x_2))$, where K is drawn uniformly from $\{1, 2, 3, 4\}$ and $c_k \sim \mathcal{U}[-1, 1]$. For OOD evaluation, $c_k \sim \mathcal{U}[1, 1.5]$. This benchmark follows [7], with the coefficient encoder receiving the conductivity field on a 32×32 uniform grid.

The fourth benchmark is the viscous Burgers equation:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \lambda \frac{\partial^2 u}{\partial x^2}, \quad x \in [-1, 1],\ t \in [0, 1], \qquad u(\pm 1, t) = 0,$$

with viscosity $\lambda = 0.1/\pi$. The unknown is the initial condition $a(x) = u(x, 0)$, sampled from $\mathcal{GP}(0, 49^2(-\Delta + 49I)^{-2})$ via a truncated sine series $a(x) = \sum_{k=1}^{K} c_k \sin(k\pi x)$. For OOD evaluation, initial conditions are drawn from $\mathcal{GP}(0, 36^2(-\Delta + 36I)^{-2})$, which yields different smoothness and amplitude characteristics. Observations consist of $M = 100$ solution values at randomly sampled space-time locations (x_i, t_i). The boundary conditions $u = 0$ at $x = \pm 1$ are enforced by a mollifier, and the coefficient encoder takes the initial condition evaluated at 128 uniform spatial points as input [6]. Reference solutions are computed with the Chebfun spectral solver [44] rather than FEM. In contrast to the steady-state problems above, the inverse problem here requires recovering an initial condition from partial observations of the solution over time.

The observation noise level is characterised by the signal-to-noise ratio

$$\text{SNR} = 10 \log_{10} \left(\frac{\frac{1}{M} \sum_{i=1}^{M} u(\boldsymbol{x}_i)^2}{\sigma_{\text{data}}^2} \right), \tag{4.2}$$

where σ_{data} is the noise standard deviation. For EIT, noise is applied independently to each of the L boundary condition responses. Model architectures, hyperparameters, and optimisation configurations for each problem are detailed in Appendix F.

4.2. Experimental setup

For each benchmark, we run 4 NUTS chains initialised at the mode of the learned prior, $F_\phi^{-1}(\mathbf{0})$, where $\boldsymbol{z} = \mathbf{0}$ is the mode of the base distribution (details in Appendix E). Per-problem warmup lengths, sampling configurations, and mass matrix choices are given in Table B.1 in Appendix B.

For the baseline and physics-constraint experiments, we evaluate on three randomly selected test coefficient fields. The noise and sensor-count sensitivity analyses fix a single test instance to isolate the effect of each varied quantity. To account for variability from observation noise realisations, sensor placement, and MCMC trajectory randomness, every experiment is repeated with three random seeds. Because the pre-trained neural networks are approximate forward models, σ_{data} must account for model discrepancy as well as measurement noise (discussed further in Chapter 5). We calibrate σ_{data} per problem using either the MAP residual at sensor locations or short pilot MCMC runs with coverage-based selection. Per-problem values and the full calibration procedure are given in Appendix B. Reported metrics are cross-instance means $\pm$ standard deviations, where each instance value is first averaged over the three seeds. Reconstruction quality is measured by the relative root mean squared error (rRMSE),

$$\text{rRMSE} = \sqrt{\frac{\sum_i (\tilde{a}(\boldsymbol{x}_i) - a^{\text{true}}(\boldsymbol{x}_i))^2}{\sum_i (a^{\text{true}}(\boldsymbol{x}_i))^2}},$$

for the continuous Darcy, EIT, and Burgers problems, and by the cross-correlation indicator I_{corr} [7] for the piecewise-constant Darcy problem. Given reconstructed and true fields rescaled to $\{0,1\}$ via $\tilde{a} = (a-5)/5$, the indicator is

$$I_{\mathrm{corr}} = \frac{\sum_i (\tilde{a}^{\mathrm{true}}(\boldsymbol{x}_i))^2 \, (\tilde{a}(\boldsymbol{x}_i))^2}{\sqrt{\sum_i (\tilde{a}^{\mathrm{true}}(\boldsymbol{x}_i))^2} \, \sqrt{\sum_i (\tilde{a}(\boldsymbol{x}_i))^2}},$$

where the sums run over all grid locations. For each posterior sample, the coefficient decoder outputs logits that are mapped through a sigmoid and thresholded at 0.5 to obtain a binary coefficient field $\tilde{a}^{(s)}(\boldsymbol{x}_i) \in \{k_{\mathsf{low}}, k_{\mathsf{high}}\}$. The posterior mean field $\bar{\tilde{a}}(\boldsymbol{x}_i) = \frac{1}{S}\sum_{s=1}^{S} \tilde{a}^{(s)}(\boldsymbol{x}_i)$ is then thresholded at the midpoint $(k_{\mathsf{low}} + k_{\mathsf{high}})/2$ to obtain the final binary reconstruction, and I_{corr} is computed on this field. For the Laplace MAP estimate, the same sigmoid-and-threshold pipeline is applied to the single decoded output.

Uncertainty quantification is assessed through four metrics. The continuous ranked probability score (CRPS) [45] is a strictly proper scoring rule[2] that jointly penalises bias and lack of sharpness, with lower values indicating better calibrated predictions. For S posterior samples at grid location $\boldsymbol{x}_j$, the CRPS is estimated as

$$\widehat{\mathrm{CRPS}}_j = \frac{1}{S}\sum_{s=1}^{S} \left| \tilde{a}^{(s)}(\boldsymbol{x}_j) - a^{\mathrm{true}}(\boldsymbol{x}_j) \right| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S} \left| \tilde{a}^{(s)}(\boldsymbol{x}_j) - \tilde{a}^{(s')}(\boldsymbol{x}_j) \right|,$$

where $\tilde{a}^{(s)}$ denotes the coefficient field reconstructed from the s-th posterior sample. The first term measures the mean absolute error against the truth and the second measures the spread among posterior samples. The reported CRPS is the spatial average $\frac{1}{N}\sum_{j=1}^{N} \widehat{\mathrm{CRPS}}_j$.

Let $Q_p(\boldsymbol{x}_j)$ denote the p-th quantile of the posterior samples at grid point $\boldsymbol{x}_j$. The empirical credible interval coverage at level $1-\alpha$ is the fraction of grid points for which the truth falls inside the equal-tailed credible interval,

$$C_{1-\alpha} = \frac{1}{N}\sum_{j=1}^{N} \mathbf{1}\big(Q_{\alpha/2}(\boldsymbol{x}_j) \le a^{\mathrm{true}}(\boldsymbol{x}_j) \le Q_{1-\alpha/2}(\boldsymbol{x}_j)\big),$$

and the mean credible interval width at the same level is

$$W_{1-\alpha} = \frac{1}{N}\sum_{j=1}^{N} \big(Q_{1-\alpha/2}(\boldsymbol{x}_j) - Q_{\alpha/2}(\boldsymbol{x}_j)\big).$$

We report $C_{0.95}$ and $W_{0.95}$ (i.e., $\alpha = 0.05$) as the primary coverage and width metrics. The nominal coverage level $1-\alpha$ is the target fraction of grid points that should fall inside the credible interval if the posterior is perfectly calibrated, while the empirical coverage $C_{1-\alpha}$ is the fraction actually observed. The calibration curve evaluates coverage at K nominal levels,

$$\mathrm{Cal} = \{(1-\alpha_k,\ C_{1-\alpha_k})\}_{k=1}^{K},$$

with $K = 10$ levels equally spaced from 10% to 95%. A well-calibrated posterior lies on the diagonal where empirical coverage equals the nominal level.

Two additional diagnostics assess the quality and consistency of the posterior. The mean Gaussian negative log-likelihood (NLL) over the spatial evaluation grid,

$$\mathrm{NLL} = \frac{1}{N}\sum_{i=1}^{N} \frac{(\tilde{a}(\boldsymbol{x}_i) - a^{\mathrm{true}}(\boldsymbol{x}_i))^2}{2\,\hat{\sigma}^2(\boldsymbol{x}_i)} + \frac{1}{2}\log\big(2\pi\,\hat{\sigma}^2(\boldsymbol{x}_i)\big),$$

where $\hat{\sigma}^2(\boldsymbol{x}_i)$ denotes the pointwise posterior variance and N is the number of grid points, is a strictly proper scoring rule [45]. The first term penalises prediction errors inversely weighted by the posterior variance, so underconfident posteriors (large $\hat{\sigma}^2$) reduce this penalty but inflate the second term, while

[2] A scoring rule is *strictly proper* if the only way to achieve the best possible expected score is to report the true predictive distribution. This guarantees that improving the score always means improving the quality of the uncertainty estimate.

overconfident posteriors (small $\hat{\sigma}^2$) do the opposite. The NLL is therefore minimised only when the posterior variance matches the true squared error, rewarding both accurate point predictions and well-calibrated uncertainty. For the piecewise Darcy problem, where the coefficient field is binary, the Gaussian log score is replaced by a Bernoulli NLL. Each posterior sample is decoded and thresholded at the midpoint $(k_{\text{low}} + k_{\text{high}})/2$, yielding the per-pixel probability $\hat{p}(\boldsymbol{x}_i) = \frac{1}{S}\sum_{s=1}^{S} \mathbf{1}\big(\tilde{a}^{(s)}(\boldsymbol{x}_i) \geq (k_{\text{low}} + k_{\text{high}})/2\big)$ and the binary ground truth $y_i = (a^{\text{true}}(\boldsymbol{x}_i) - k_{\text{low}})/(k_{\text{high}} - k_{\text{low}}) \in \{0, 1\}$. The mean Bernoulli negative log-likelihood,

$$\text{NLL}_{\text{Bernoulli}} = -\frac{1}{N}\sum_{i=1}^{N}\big(y_i \log \hat{p}(\boldsymbol{x}_i) + (1 - y_i)\log(1 - \hat{p}(\boldsymbol{x}_i))\big), \tag{4.3}$$

is also a strictly proper scoring rule [45]. The Spearman rank correlation ρ between the pointwise absolute error $|\tilde{a}(\boldsymbol{x}_i) - a^{\text{true}}(\boldsymbol{x}_i)|$ and the posterior standard deviation $\hat{\sigma}(\boldsymbol{x}_i)$ measures whether the posterior uncertainty is spatially aligned with the reconstruction error.

MCMC convergence is monitored via three diagnostics. The split-$\hat{R}$ statistic assesses convergence by comparing within-chain and between-chain variance [46]. Each chain is split in half, and m denotes the total number of resulting half-chains, each of length n [47]. The statistic is

$$\hat{R} = \sqrt{\frac{\widehat{\text{var}}^{+}(\beta_{1,k})}{W}},$$

where $\beta_{1,k}$ is a single latent component, $W = \frac{1}{m}\sum_{j=1}^{m} s_j^2$ is the pooled within-chain variance, $B = \frac{n}{m-1}\sum_{j=1}^{m}(\bar{\beta}_{1,k,j} - \bar{\bar{\beta}}_{1,k})^2$ is the between-chain variance, and $\widehat{\text{var}}^{+}(\beta_{1,k}) = \frac{n-1}{n}W + \frac{1}{n}B$. At convergence $W \approx \widehat{\text{var}}^{+}$ and $\hat{R} \to 1$. Values below 1.01 indicate adequate convergence[3] [48]. The effective sample size (ESS) estimates the number of independent samples equivalent to the correlated chain, computed by dividing the total sample count by the integrated autocorrelation time [48]. We report the minimum ESS across all d_1 latent dimensions as the bottleneck diagnostic, with $\text{ESS}_{\min} \geq 50$ indicating adequate sampling and $\text{ESS}_{\min} < 10$ indicating that the chains have not explored the posterior. Divergent transitions signal that the sampler failed to accurately simulate Hamiltonian dynamics in regions of high curvature [49].

As a baseline for uncertainty quantification, we compare NUTS against a Laplace approximation [24], which fits a Gaussian $q(\beta_1) = \mathcal{N}(\hat{\beta}_{1\text{MAP}},\, H^{-1})$ at the MAP estimate by approximating the negative log-posterior as locally quadratic. Full details of the MAP optimisation, Hessian computation, and sampling procedure are given in Appendix C. The MAP point estimate $\tilde{a}^{\text{MAP}}$ shown in field plots throughout this section is the mode of the Laplace approximation, obtained by the deterministic inversion procedure described in Appendix F.

4.3. Posterior inference

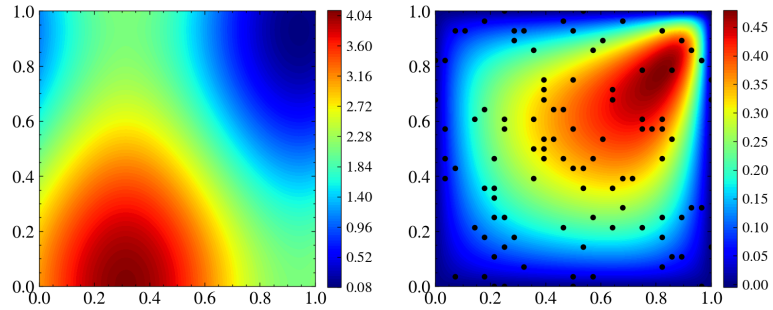
We first examine whether Bayesian IGNO produces meaningful uncertainty estimates on in-domain test instances. All field plots and per-instance diagnostics in this section correspond to a single representative test instance; a full posterior gallery covering additional instances appears in Appendix D. All experiments in this section use noiseless observations.

For the continuous Darcy problem (Figure 4.1, Figure 4.2), the posterior achieves 97% coverage at the nominal 95% level (Table 4.3), with the posterior narrowing CI widths by a factor of 6 relative to the unconditional prior distribution[4]. The Spearman rank correlation between pointwise absolute error and posterior standard deviation is $\rho = 0.31$ (Table 4.4), indicating weak spatial alignment between uncertainty and error.

For the piecewise-constant Darcy problem (Figure 4.3, Figure 4.4), the posterior mean yields $I_{\text{corr}} = 0.91$ after thresholding, with 100% coverage and a mean CI width of 4.3 on a field taking values in $\{5, 10\}$.

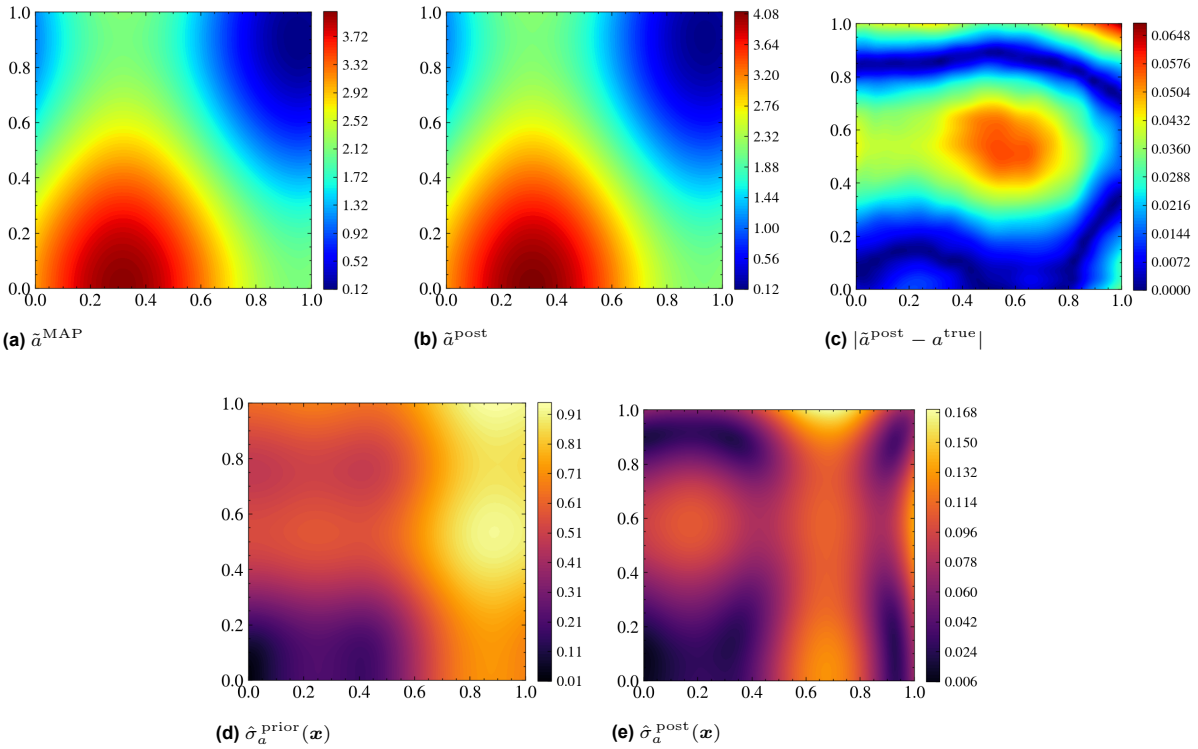
[3] A chain that "mixes well" traverses the full support of the target distribution within a reasonable number of iterations, so that successive samples are nearly independent. Poor mixing means the chain remains trapped in a subregion of the posterior and has not yet converged to the stationary distribution.

[4] The unconditional prior distribution is the distribution over coefficient fields obtained by drawing latent samples from the flow prior and decoding them through G_{θ_a}, without conditioning on any observed data. It represents the variability of the model's reconstructions before any measurements are incorporated.



(a) a^{true} **(b)** u^{true}

Figure 4.1: Continuous Darcy ground-truth fields. Ground-truth permeability a^{true} and pressure field u^{true} with sensor locations (black dots).



(a) $\tilde{a}^{\mathrm{MAP}}$ **(b)** $\tilde{a}^{\mathrm{post}}$ **(c)** $|\tilde{a}^{\mathrm{post}} - a^{\mathrm{true}}|$

(d) $\hat{\sigma}_a^{\mathrm{prior}}(\boldsymbol{x})$ **(e)** $\hat{\sigma}_a^{\mathrm{post}}(\boldsymbol{x})$

Figure 4.2: Continuous Darcy coefficient reconstruction and uncertainty. Top row: Laplace MAP estimate $\tilde{a}^{\mathrm{MAP}}$, posterior mean $\tilde{a}^{\mathrm{post}}$, and pointwise absolute error $|\tilde{a}^{\mathrm{post}} - a^{\mathrm{true}}|$. Bottom row: unconditional prior standard deviation and posterior standard deviation.

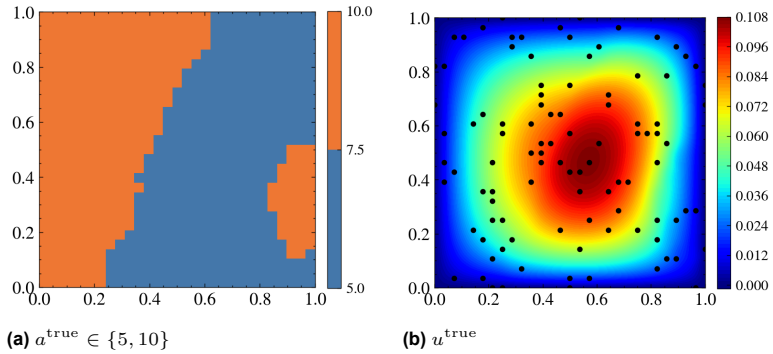
Table 4.2: Effect of coefficient contrast ratio on posterior inference for the piecewise Darcy problem. All variants use the same architecture, latent dimension, and 100 sensors. The 20:1 and 200:1 models were trained with 2× and 3× the baseline training epochs respectively. Values are cross-instance means ± standard deviations across three test instances, each averaged over three seeds. At 200:1 contrast the posterior collapses to the prior, with CI width spanning the full coefficient range.

Contrast	I_{corr}	CRPS	Cov. 95%	CI width	n_{div}
2:1	0.91±0.067	0.41±0.27	1.00±0.00	4.3±0.41	0
20:1	0.73±0.13	17±6.3	0.99±0.003	81±2.9	310
200:1	0.69±0.15	210±18	1.00±0.00	995±0.0	13

The unconditional prior CI width is 5.0 (the full range of the binary field), so the posterior narrows only modestly.

The large CI width is expected for a binary field. At any pixel where posterior samples disagree on the phase, the sample-averaged field takes intermediate values between 5 and 10, and the 95% credible interval covers most of the $[5, 10]$ range. Pointwise summaries such as the posterior mean and credible interval are therefore misleading for this problem, because the underlying uncertainty is discrete rather than continuous. The sample gallery in Figure D.4 is more informative. Individual samples decode to sharp binary fields, but the phase boundaries shift from sample to sample, so the posterior uncertainty is better understood as uncertainty over boundary locations than over pixel values.

The Spearman correlation between pointwise error and posterior standard deviation is the strongest across all four benchmarks ($\rho = 0.94$, Table 4.4), consistent with the spatial structure of the binary phase problem. Regions near phase boundaries are both harder to reconstruct and carry more posterior uncertainty, so error and uncertainty concentrate in the same locations. The Spearman correlation should therefore not be taken as informative for this problem, since it is driven by the binary contrast between flat interior regions and phase boundaries.



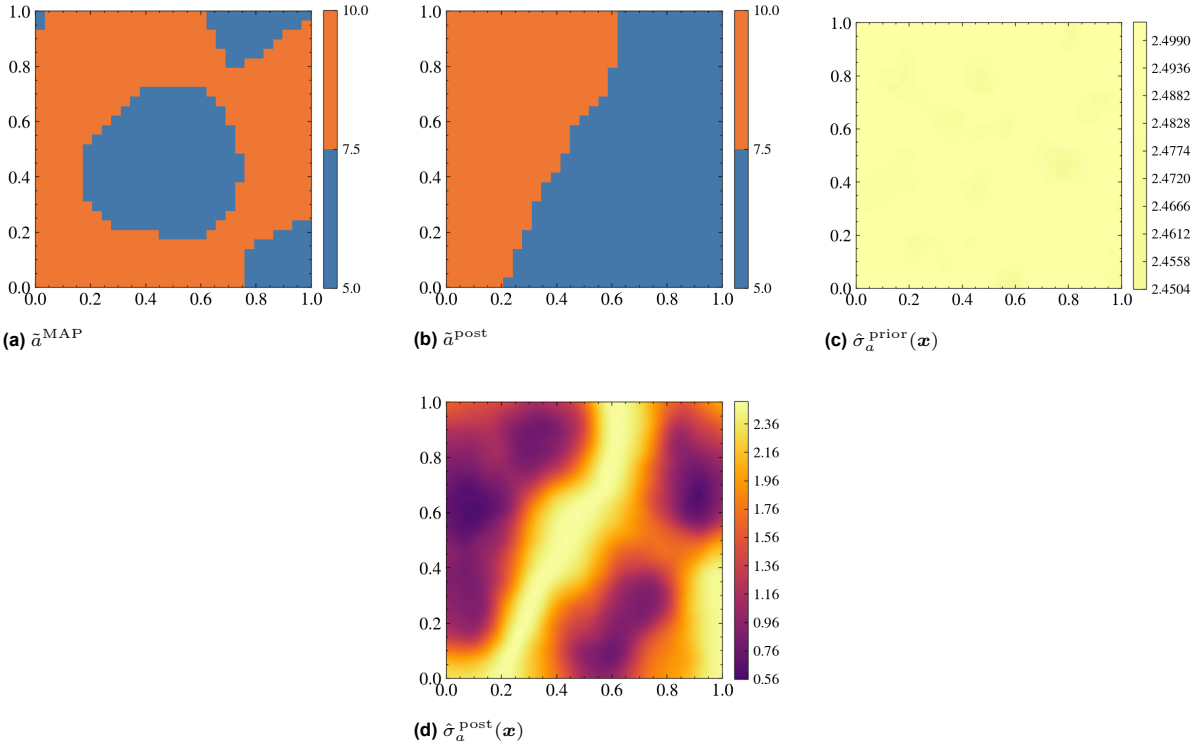
(a) $a^{\text{true}} \in \{5, 10\}$ **(b)** u^{true}

Figure 4.3: Piecewise-constant Darcy ground-truth fields. Ground-truth binary permeability $a^{\text{true}} \in \{5, 10\}$ and pressure field u^{true} with sensor locations (black dots).

We also train and evaluate piecewise Darcy variants at coefficient contrasts of 20:1 ($k \in \{5, 100\}$) and 200:1 ($k \in \{5, 1000\}$), keeping all other settings identical. The forward model progressively fails to learn the higher-contrast PDE. The relative L2 error of the solution decoder on held-out test data rises from 0.025 at 2:1 contrast to 0.59 at 20:1 and 0.96 at 200:1, despite doubling and tripling the number of training epochs respectively. At 200:1 contrast, the decoder effectively cannot predict the solution field.

The posterior metrics mirror the training-stage degradation (Table 4.2). At 20:1 contrast, I_{corr} degrades from 0.91 to 0.73 and NUTS produces approximately 310 divergent transitions per run on average, indicating that the posterior geometry becomes difficult to explore. At 200:1 contrast, the posterior collapses to the prior: the CI width of 995 equals the full range of the $\{5, 1000\}$ coefficient field, and the relative L2 error of the posterior-mean solution field is 0.95. The sampler itself remains healthy at both contrast levels, with $\text{ESS}_{\min} > 8000$ and $\hat{R} < 1.001$. The degradation reflects the solution decoder's inability to forward-solve the high-contrast PDE, not a failure of the MCMC sampling.

For EIT (Figure 4.5), the posterior concentrates sharply, with a 33× reduction in CI width relative to



(a) $\tilde{a}^{\mathrm{MAP}}$ **(b)** $\tilde{a}^{\mathrm{post}}$ **(c)** $\hat{\sigma}_a^{\mathrm{prior}}(\boldsymbol{x})$

(d) $\hat{\sigma}_a^{\mathrm{post}}(\boldsymbol{x})$

Figure 4.4: Piecewise-constant Darcy coefficient reconstruction and uncertainty. Top row: Laplace MAP estimate $\tilde{a}^{\mathrm{MAP}}$, posterior mean $\tilde{a}^{\mathrm{post}}$, and unconditional prior standard deviation. Bottom row: posterior standard deviation. High standard deviation near phase boundaries reflects the sensitivity of the decoded binary field to small latent perturbations.

the unconditional prior (Table 4.3). Coverage is 98% at the nominal 95% level. The posterior standard deviation increases towards the interior of the domain, and the standard deviation at each grid point correlates strongly with its distance to the nearest boundary sensor (Pearson $r = 0.94$, Figure 4.6). The Spearman correlation between pointwise error and posterior standard deviation is $\rho = 0.29$ (Table 4.4).

For the Burgers problem (Figure 4.7), the posterior achieves 93% coverage at the nominal 95% level (Table 4.3), with credible intervals narrowing by a factor of 10$\times$ relative to the prior. The Spearman correlation between pointwise error and posterior standard deviation is $\rho = 0.27$ (Table 4.4), indicating weak spatial alignment. Cross-instance variance is the highest of any benchmark (Table 4.3), consistent with the more complex posterior geometry of the time-dependent PDE and its larger latent dimension.

Because Burgers is the only benchmark with a one-dimensional coefficient field, the prior standard deviation can be visualised directly as a function of x (Figure 4.7c). The characteristic double-hump shape is consistent with the sine-series representation of the training distribution: each basis function $\sin(k\pi x)$ equals zero at $x = 0$ and $x = \pm 1$ for every k, so all training initial conditions pass through zero at these three points and the prior variance vanishes there. The variance peaks near $x \approx \pm 0.5$, where the low-frequency sine modes that dominate the GP have their largest values. The learned prior reproduces this structure, confirming that the normalising flow and coefficient decoder together capture the training distribution.

The Laplace approximation produces valid uncertainty estimates on two of the four problems, is unreliable on a third, and fails entirely on the fourth (Table 4.3). For piecewise Darcy, the Hessian is non-positive-definite, so no Laplace samples can be drawn. For Burgers, the Laplace approximation succeeds on most runs with coverage of 0.99, but at least one run per test instance produces a non-positive-definite Hessian, making the method unreliable in practice. On continuous Darcy, the Laplace posterior achieves comparable or better pointwise metrics than NUTS, with lower CRPS (0.029 vs 0.047) and narrower CI widths (0.34 vs 0.38), though NUTS achieves a better NLL (-2.0 vs -1.4). The large cross-seed variance on this problem (Table 4.3) means neither method dominates conclusively. For EIT, the Laplace posterior achieves full coverage but is oversmoothed, with CI widths of 0.61

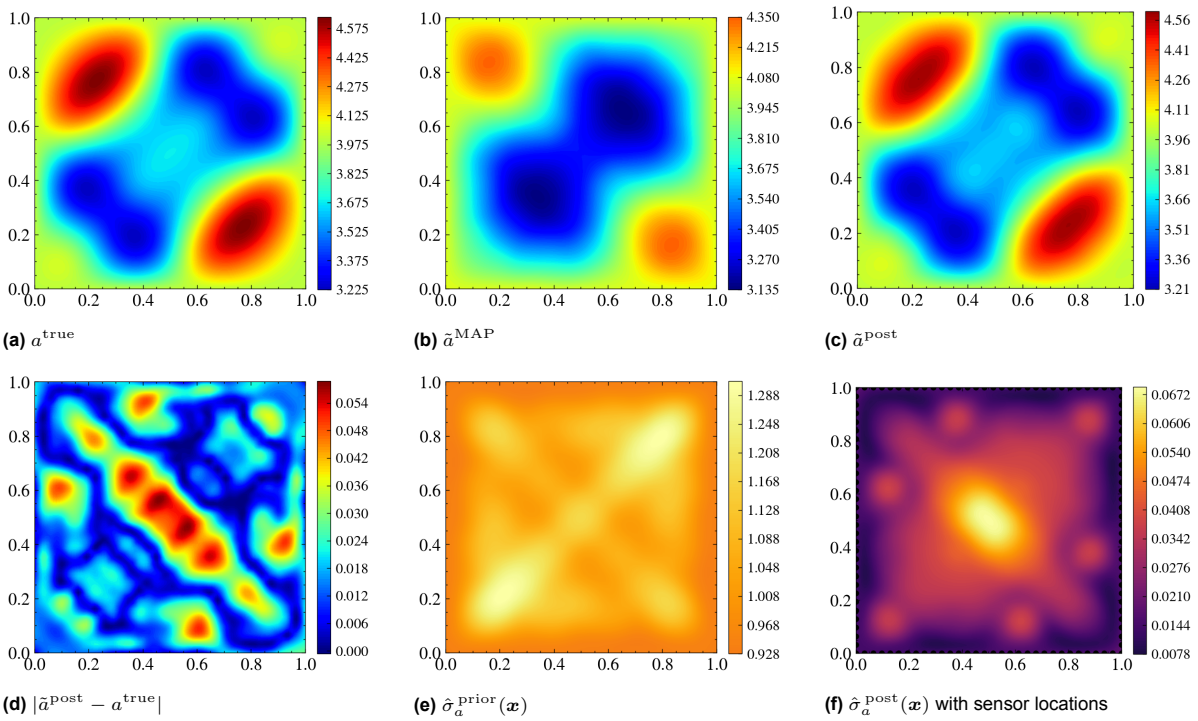


Figure 4.5: EIT conductivity field reconstruction. Top row: ground-truth conductivity a^{true}, Laplace MAP estimate $\tilde{a}^{\text{MAP}}$, and posterior mean $\tilde{a}^{\text{post}}$. Bottom row: pointwise absolute error $|\tilde{a}^{\text{post}} - a^{\text{true}}|$, unconditional prior standard deviation, and posterior standard deviation with boundary sensor locations (black dots). Uncertainty increases towards the interior of the domain, away from the boundary sensors.

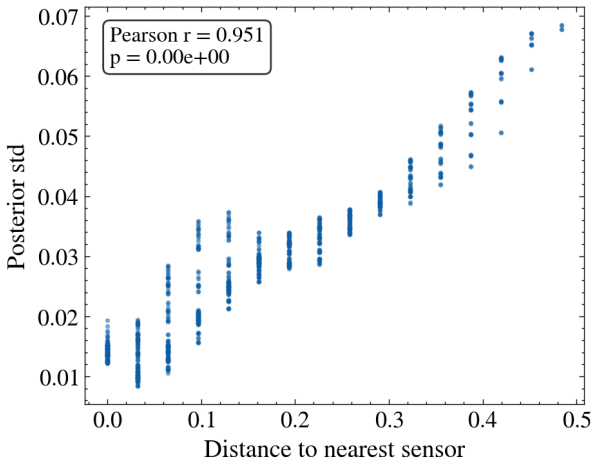
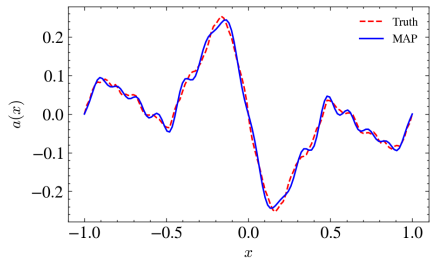
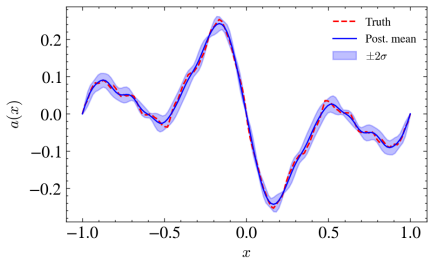


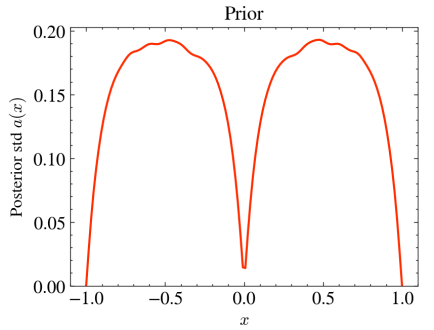
Figure 4.6: Posterior standard deviation versus distance to the nearest boundary sensor for each grid point in the EIT problem (Pearson $r = 0.94$, $p = 0.006$). Uncertainty increases monotonically with distance from the boundary, where all measurements are collected.



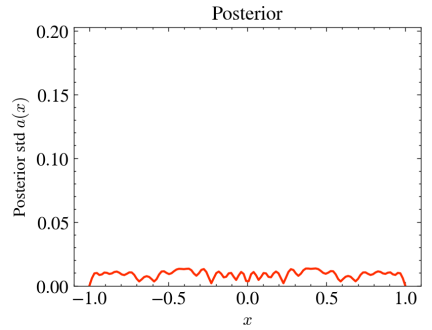
(a) a^{true} vs. $\tilde{a}^{\mathrm{MAP}}$



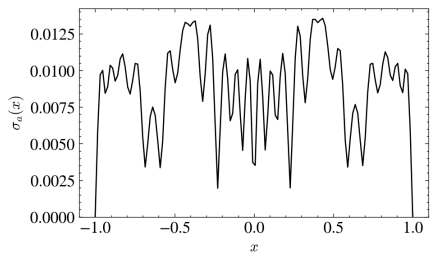
(b) a^{true} vs. $\tilde{a}^{\mathrm{post}} \pm 2\hat{\sigma}_a$



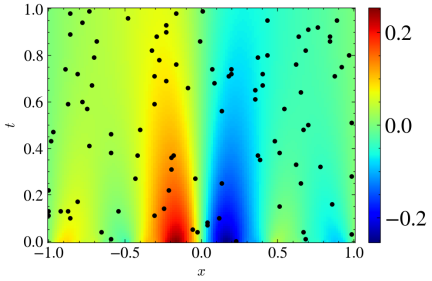
(c) $\hat{\sigma}_a^{\mathrm{prior}}(x)$



(d) $\hat{\sigma}_a^{\mathrm{post}}(x)$



(e) $\hat{\sigma}_a^{\mathrm{post}}(x)$



(f) $u^{\mathrm{true}}(x,t)$

Figure 4.7: Burgers reconstruction. Top row: ground-truth initial condition $a^{\mathrm{true}}(x)$ overlaid with the Laplace MAP estimate and with the posterior mean and $\pm 2\hat{\sigma}_a$ credible band. Middle row: unconditional prior standard deviation and posterior standard deviation. Bottom row: pointwise posterior standard deviation and ground-truth space-time solution $u^{\mathrm{true}}(x,t)$ with sensor locations (black dots).

Table 4.3: Posterior inference metrics from sampling. Post. mean is the reconstruction error of the posterior mean field, obtained by decoding each posterior sample and averaging in function space. CRPS is the continuous ranked probability score (lower is better). NLL is the mean Gaussian log score (lower is better), except for piecewise Darcy which uses the Bernoulli log score (Eq. 4.3). All error values are rRMSE (lower is better) except piecewise Darcy, which reports I_{corr} (higher is better). Values are cross-instance means $\pm$ standard deviations across three test instances, each averaged over three seeds. NUTS matches or outperforms Laplace on NLL across all problems where Laplace converges, while Laplace fails entirely on piecewise Darcy.

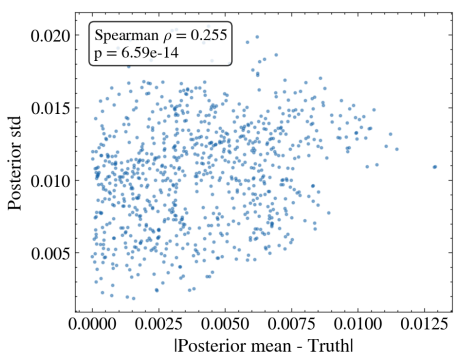
Problem	Method	Post. mean	CRPS	NLL	Cov. 95%	CI width
Darcy continuous	NUTS	0.033±0.042	0.047±0.058	**−2.0±1.5**	**0.97±0.04**	0.38±0.36
	Laplace	**0.024±0.010**	**0.029±0.010**	−1.4±0.37	1.00±0.00	**0.34±0.093**
Darcy piecewise	NUTS	0.91±0.067	0.41±0.27	0.28±0.15	1.00±0.00	4.3±0.41
	Laplace	—	—	—	—	—
EIT	NUTS	**0.0054±0.0002**	**0.012±0.0002**	**−2.4±0.023**	**0.98±0.026**	**0.11±0.015**
	Laplace	0.010±0.0024	0.037±0.0036	−0.94±0.10	1.00±0.00	0.61±0.053
Burgers	NUTS	0.12±0.068	**0.0086±0.0030**	**−3.1±0.36**	**0.93±0.023**	**0.062±0.014**
	Laplace	**0.11±0.024**	0.012±0.0033	−2.6±0.26	0.99±0.00	0.14±0.032

Table 4.4: Spearman rank correlation ρ between pointwise absolute error and posterior standard deviation. Higher values indicate stronger spatial alignment between uncertainty and reconstruction error. Values are cross-instance means $\pm$ standard deviations. NUTS achieves positive spatial alignment on all four benchmarks, whereas Laplace produces anti-correlated uncertainty on EIT.

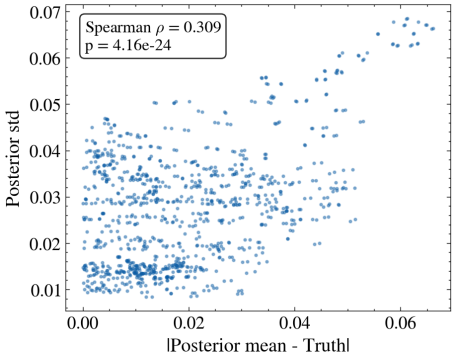
Problem	NUTS ρ	Laplace ρ
Darcy continuous	0.31 ± 0.083	$\mathbf{0.53 \pm 0.22}$
Darcy piecewise	0.94 ± 0.091	—
EIT	$\mathbf{0.29 \pm 0.017}$	-0.49 ± 0.064
Burgers	$\mathbf{0.27 \pm 0.071}$	0.21 ± 0.076

versus 0.11 for NUTS and a worse NLL (-0.94 vs -2.4). Overall, the Laplace approximation is orders of magnitude faster than NUTS but is unreliable on two of the four problems, and where it succeeds, it does not consistently outperform NUTS across all metrics.

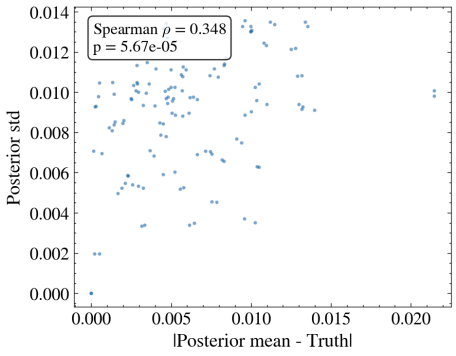
Figure 4.8 shows the Spearman rank correlation between pointwise absolute error and posterior standard deviation across three of the four problems (Table 4.4). Figure 4.9 shows that the two Darcy problems exhibit overcoverage at all nominal levels, with the piecewise variant showing extreme overcoverage (empirical coverage exceeds 85% even at the 10% nominal level). This conservatism is preferable to the overconfidence of variational approaches [17]. The EIT and Burgers calibration curves track the ideal line closely.



(a) Darcy continuous

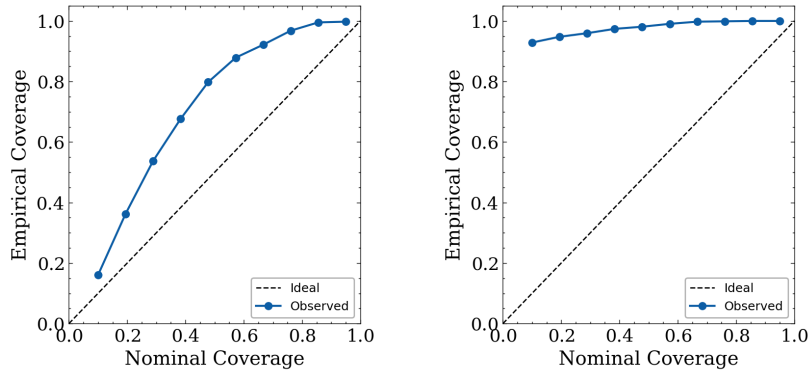


(b) EIT



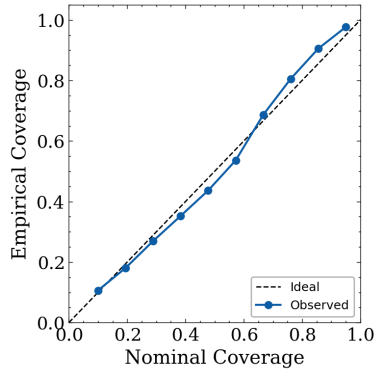
(c) Burgers

Figure 4.8: Pointwise absolute error versus posterior standard deviation for three of the four benchmarks. Each point represents one spatial grid location. The Spearman rank correlation ρ is annotated. Positive correlation indicates that the posterior assigns higher uncertainty where the reconstruction error is larger. The piecewise Darcy problem is omitted because its binary coefficient field makes this correlation trivially high.

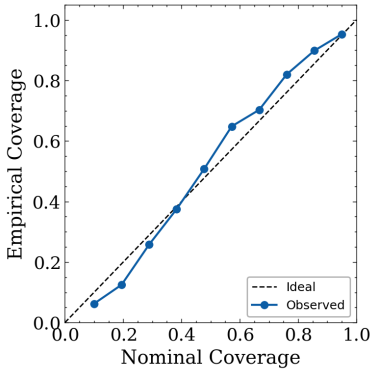


(a) Darcy continuous

(b) Darcy piecewise



(c) EIT



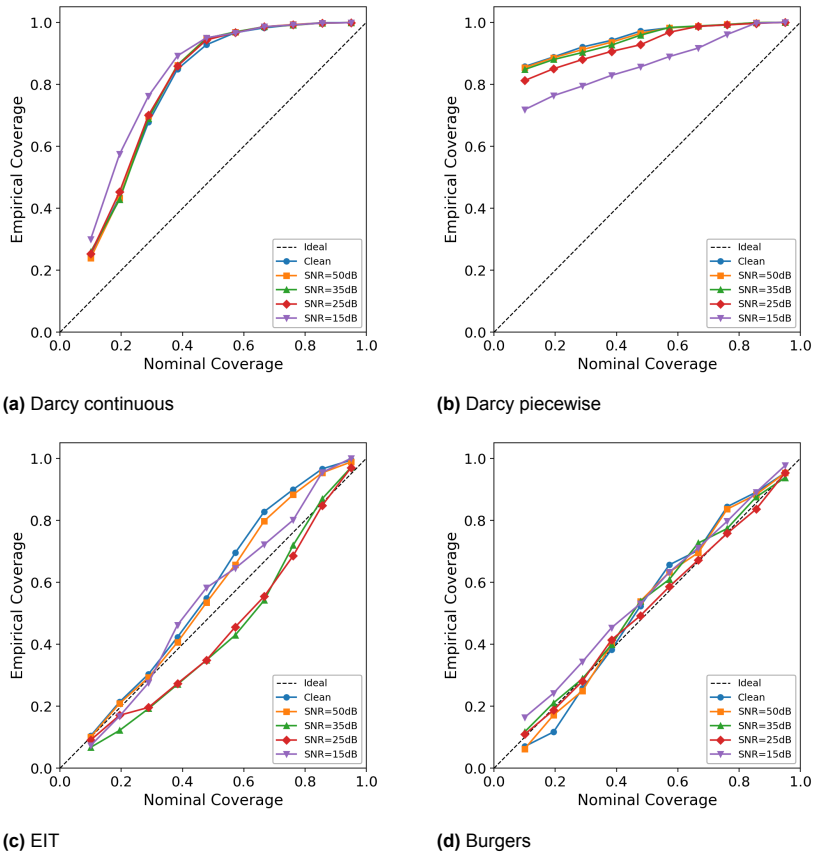
(d) Burgers

Figure 4.9: Calibration curves for the baseline posterior. The dashed line indicates ideal calibration. EIT and Burgers track the ideal line closely, while both Darcy variants overcover at all nominal levels.

4.4. Sensitivity analysis

If the posterior is well calibrated, it should respond to the quantity and quality of the observations. We test this by varying the SNR (Eq. 4.2) and the number of sensors M.

For the noise sensitivity analysis, we sweep $\mathrm{SNR} \in \{50, 35, 25, 15\}$ dB across all four problems. The noise sweep evaluates a single test instance with per-instance σ_{data} calibration rather than the per-seed calibration across three test instances used in Section 4.3, so the clean-condition metrics do not match the baseline in Table 4.3. Darcy piecewise, EIT, and Burgers converge reliably across all noise levels ($\hat{R} < 1.01$, ESS > 400 for every seed). Continuous Darcy exhibits intermittent convergence difficulties, with individual seeds exceeding these thresholds at several noise levels. The most severe case is at 50 dB, where one seed fails to converge ($\hat{R} = 1.2$, ESS < 10). Because the table reports three-seed means, these per-seed failures are partially masked by the averaging. Calibration curves are shown in Figure 4.10.



(a) Darcy continuous **(b)** Darcy piecewise

(c) EIT **(d)** Burgers

Figure 4.10: Noise sweep calibration curves. The dashed line indicates ideal calibration. Calibration remains stable across noise levels for all four benchmarks, confirming that the posterior responds appropriately to increasing observation noise.

For the continuous Darcy problem (Table 4.5), coverage remains near 100% across all noise levels while CI width grows only modestly from 0.33 to 0.37, indicating that the posterior overcovers uniformly across the sweep.

For the piecewise Darcy problem (Table 4.5), coverage remains at or above 99% at all noise levels, and the CI width is nearly constant (varying by less than 3%). At 15 dB the reconstruction accuracy (I_{corr}) degrades modestly from 0.84 to 0.78, but the posterior does not sharpen or widen. The insensitivity to noise is expected. The parameter σ_{data} controls the width of the Gaussian data likelihood and is set to the MAP residual for this problem (Section 4.2). Because the MAP residual ($\sigma_{\text{data}} \approx 0.012$) exceeds the observation noise at every sweep level (including 15 dB, where $\sigma_{\text{noise}} \approx 0.010$), σ_{data} does not change across conditions, and the data likelihood that enters the posterior is nearly identical at every noise level. For EIT and Burgers, by contrast, the observation noise exceeds σ_{data} at 25 dB, producing the noise sensitivity visible in Table 4.5.

Table 4.5: Noise sweep metrics (3-seed mean). For continuous problems, accuracy is rRMSE. For piecewise Darcy, it is I_{corr}. EIT and Burgers show clear sensitivity to noise, with CI widths widening to maintain coverage, while piecewise Darcy is insensitive because σ_{data} is dominated by model discrepancy.

Problem	SNR (dB)	Accuracy	CRPS	Cov. 95%	CI width
Darcy continuous	Clean	0.015	0.024	1.00	0.33
	50	0.016	0.024	1.00	0.34
	35	0.017	0.026	1.00	0.35
	25	0.017	0.026	1.00	0.35
	15	0.018	0.029	0.99	0.37
Darcy piecewise	Clean	0.84	0.71	1.00	4.6
	50	0.84	0.71	1.00	4.6
	35	0.84	0.71	1.00	4.6
	25	0.83	0.74	1.00	4.6
	15	0.78	0.86	0.99	4.5
EIT	Clean	0.0054	0.012	0.99	0.12
	50	0.0054	0.012	0.99	0.12
	35	0.0066	0.015	0.97	0.13
	25	0.015	0.034	0.96	0.27
	15	0.039	0.084	0.97	0.76
Burgers	Clean	0.080	0.0052	0.95	0.045
	50	0.081	0.0052	0.94	0.045
	35	0.088	0.0056	0.94	0.046
	25	0.14	0.0084	0.91	0.052
	15	0.20	0.012	0.92	0.076

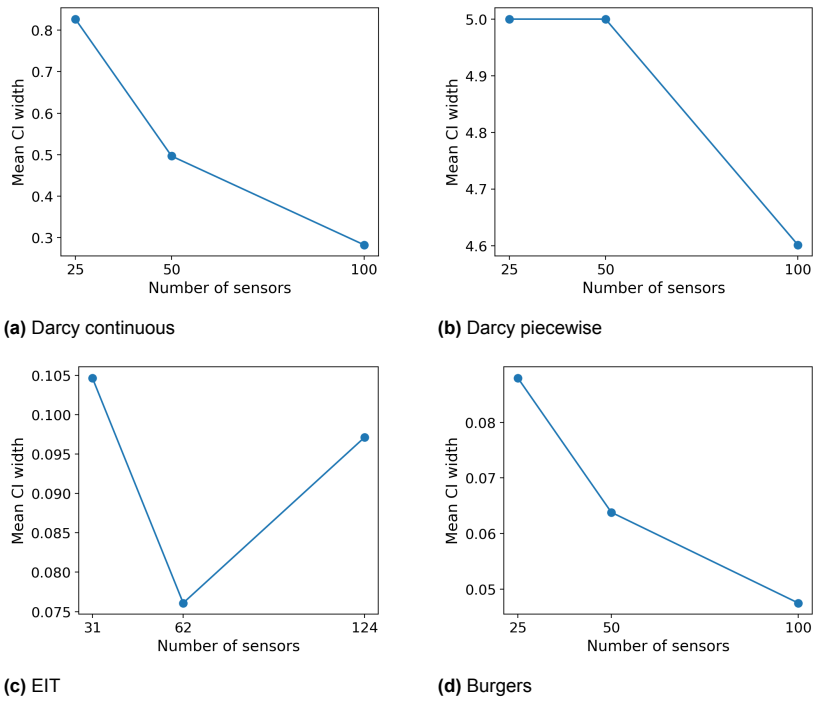
For the EIT problem (Table 4.5), reconstruction error grows by roughly $7\times$ across the sweep, but the posterior widens in response, maintaining coverage at or above 95% throughout. The posterior remains well calibrated even at high noise levels, with the flow prior regularising towards the learned latent distribution as the data likelihood becomes diffuse.

The Burgers problem (Table 4.5) shows a different pattern. At baseline the posterior achieves 95% coverage, consistent with the nominal level. As noise increases, reconstruction error grows from 0.080 to 0.20 while CI width widens from 0.045 to 0.076, and coverage remains between 91% and 95% across all conditions. The posterior thus responds to observation noise by widening appropriately, maintaining calibration throughout the sweep.

Table 4.6: Sensor sweep metrics. For continuous problems, accuracy is rRMSE. For piecewise Darcy, it is I_{corr}. Darcy piecewise, EIT, and Burgers converge across all sensor counts. Continuous Darcy shows per-seed convergence failures, particularly at $M = 25$ and $M = 50$. Continuous Darcy and Burgers show the expected trend of improving accuracy and narrowing CIs with more sensors, while piecewise Darcy is largely insensitive.

Problem	M	Accuracy	CRPS	Cov. 95%	CI width
Darcy continuous	25	0.046	0.063	1.00	0.72
	50	0.021	0.032	1.00	0.48
	100	0.0094	0.019	1.00	0.29
Darcy piecewise	25	0.78	0.84	1.00	5.0
	50	0.81	0.75	1.00	4.9
	100	0.83	0.68	1.00	4.5
EIT	31	0.0040	0.0096	1.00	0.10
	62	0.0057	0.012	0.96	0.076
	124	0.0049	0.011	0.99	0.11
Burgers	25	0.13	0.0085	0.99	0.085
	50	0.099	0.0065	0.99	0.061
	100	0.080	0.0054	0.97	0.049

We next vary the number of observation locations across all four problems (Table 4.6). The three solution-based problems (continuous Darcy, piecewise Darcy, and Burgers) use $M \in \{25, 50, 100\}$ interior sensors, while EIT uses $M \in \{31, 62, 124\}$ boundary sensors per excitation pattern.



(a) Darcy continuous **(b)** Darcy piecewise

(c) EIT **(d)** Burgers

Figure 4.11: Credible interval width as a function of sensor count (3-seed mean). CI width narrows consistently with sensor count for continuous Darcy and Burgers, while piecewise Darcy remains dominated by the flow prior.

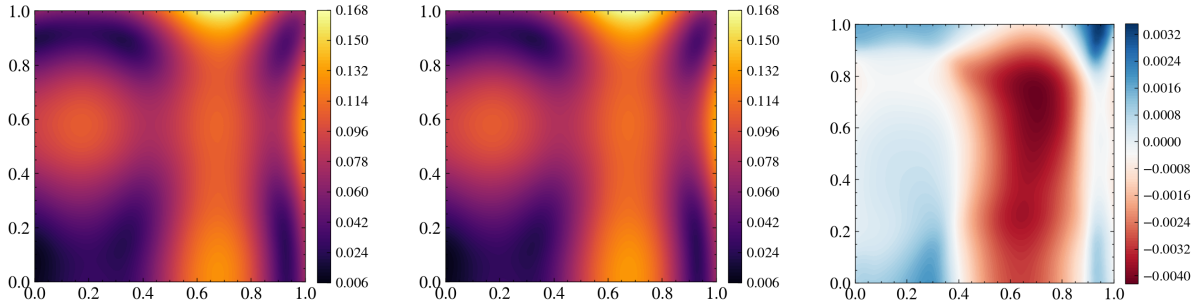
For continuous Darcy and Burgers, CI width narrows consistently with sensor count (Table 4.6, Figure 4.11), and both problems maintain coverage near the nominal 95% level across all sensor counts. For EIT, CI width drops sharply from $M = 31$ to $M = 62$ but widens at $M = 124$. Because σ_{data} is recalibrated via pilot chains at each sensor count (Section 4.2), the selected value roughly doubles from 0.1 at $M = 62$ to 0.2 at $M = 124$, widening the data likelihood enough to offset the additional observations. Piecewise Darcy shows only marginal narrowing across the sweep (CI width 5.0 to 4.5), with the posterior remaining close to the prior width of 5.0, consistent with the insensitivity observed in the noise sweep. As discussed in Section 4.3, CI width is a misleading summary for this binary field problem, since the underlying posterior uncertainty reflects disagreement over phase boundary locations rather than continuous variation in pixel values.

4.5. Role of the physics constraint

We compare sampling from a data-only posterior, which omits the PDE residual term, against the full posterior that incorporates the virtual likelihood (Table 4.7). To ensure a controlled comparison, σ_{data} is tuned once per test instance and held fixed across both conditions and all seeds, so that any difference in performance reflects the physics term alone. This tuning differs from the per-seed calibration in Section 4.3, and the data-only metrics in Table 4.7 therefore do not match the baseline NUTS results in Table 4.3.

For the continuous Darcy problem, the effect of including $\mathcal{F}_{\text{pde}}$ is inconsistent across test instances (Table 4.7). On average, the data-only posterior converges adequately ($\hat{R}_{\max} = 1.02 \pm 0.016$), while the physics-informed posterior has a higher mean $\hat{R}_{\max}$ (7.9 ± 12) due to one test instance where the sampler fails to converge ($\hat{R}_{\max} = 22$). ESS is comparable between conditions (900 ± 560 data-only versus 560 ± 560 physics). Divergent transitions are slightly lower with physics (15 ± 19 versus 24 ± 24). Coverage and reconstruction accuracy are similar between conditions, but the physics-informed posterior produces wider credible intervals (0.54 ± 0.21 versus 0.38 ± 0.36). The corresponding posterior standard deviation maps are shown in Figure 4.12. The high cross-instance variance makes it difficult to draw conclusions about the physics constraint on this problem.

For the piecewise-constant Darcy problem, including the physics constraint degrades sampler con-



(a) Data-only $\hat{\sigma}_a^{\text{post}}(\boldsymbol{x})$ **(b)** Physics-informed $\hat{\sigma}_a^{\text{post}}(\boldsymbol{x})$ **(c)** Difference

Figure 4.12: Posterior standard deviation maps for the continuous Darcy problem under data-only and physics-informed posteriors, and their difference.

vergence (Table 4.7). The data-only posterior converges reliably (ESS $15{,}000 \pm 4{,}400$, $\hat{R}_{\max} = 1.00$, zero divergent transitions), while the physics-informed posterior shows reduced ESS ($4{,}200 \pm 6{,}700$), elevated $\hat{R}_{\max}$ (1.08 ± 0.070), and 83 ± 105 divergent transitions. The large standard deviations under physics indicate that one test instance accounts for most of the degradation. Including physics narrows the CI width from 4.4 ± 0.51 to 3.4 ± 0.48, suggesting that $\mathcal{F}_{\text{pde}}$ does provide additional information for this problem, but the resulting posterior geometry is more difficult for the sampler to explore. Coverage remains near nominal under both conditions.

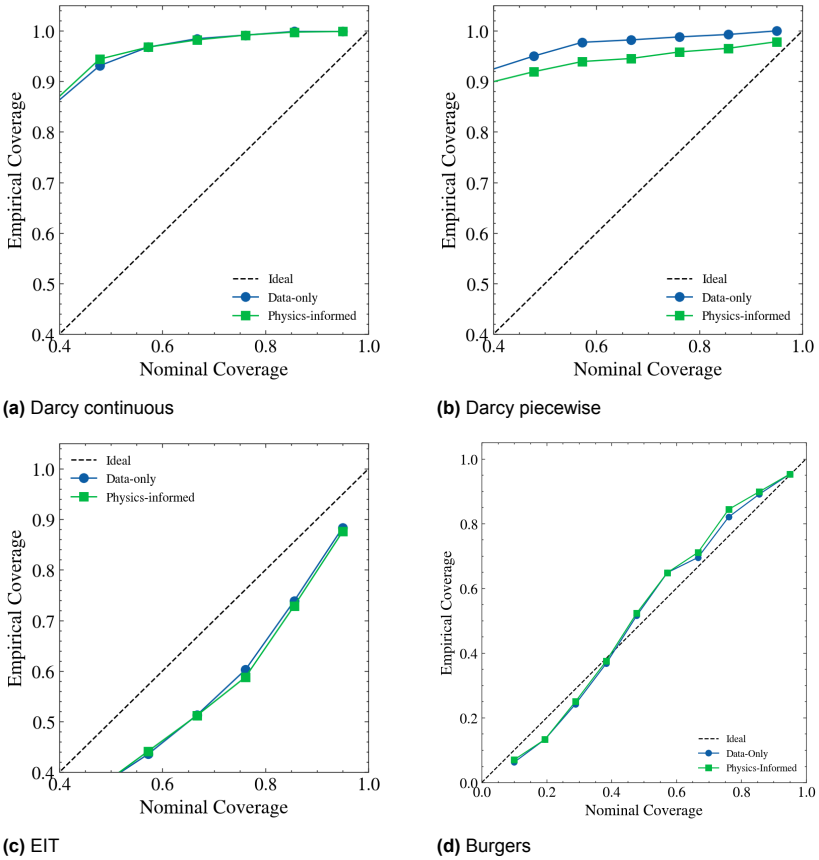
For EIT, the physics constraint provides the clearest benefit across all metrics (Table 4.7). The data-only posterior exhibits convergence failures on some test instances ($\hat{R}_{\max} = 3.9 \pm 2.6$, ESS $3{,}000 \pm 2{,}200$, divergent transitions 5.2 ± 6.3), while the physics-informed posterior converges reliably across all instances ($\hat{R}_{\max} = 1.00$, ESS $7{,}200 \pm 1{,}200$, zero divergent transitions). Including physics also improves reconstruction accuracy (0.0048 ± 0.00050 versus 0.011 ± 0.0062) and tightens credible intervals (0.077 ± 0.019 versus 0.17 ± 0.095). Coverage is similar between conditions (0.94 ± 0.056 versus 0.95 ± 0.060).

For the Burgers problem, the physics constraint has no measurable effect (Table 4.7). Both conditions produce nearly identical metrics: reconstruction accuracy (0.12 ± 0.068 data-only versus 0.12 ± 0.069 physics), coverage (0.93 ± 0.021 both), CI width (0.062 ± 0.014 both), and convergence diagnostics (ESS above $6{,}000$, $\hat{R}_{\max} = 1.00$, zero divergent transitions). This insensitivity is consistent with the observation from the noise and sensor sweeps that the Burgers posterior is driven almost entirely by the data likelihood at the baseline configuration. The calibration curves for all four problems are shown in Figure 4.13.

Table 4.7: Data-only versus physics-informed posterior metrics across all four benchmarks. Accuracy is rRMSE for continuous problems and I_{corr} for piecewise Darcy. Values are cross-instance means ± standard deviations across three test instances, each averaged over three seeds. Including physics improves all metrics for EIT but degrades convergence for piecewise Darcy and is neutral for Burgers.

Problem	Condition	Accuracy	Cov. 95%	CI width	$\text{ESS}_{\min}$	$\hat{R}_{\max}$
Darcy continuous	Data-only	0.033±0.042	0.97±0.043	0.38±0.36	900±560	1.02±0.016
	Physics	0.054±0.034	0.97±0.044	0.54±0.21	560±560	7.9±12
Darcy piecewise	Data-only	0.90±0.065	1.00±0.00	4.4±0.51	15,000±4,400	1.00±0.00
	Physics	0.86±0.10	0.99±0.015	3.4±0.48	4,200±6,700	1.08±0.070
EIT	Data-only	0.011±0.0062	0.95±0.060	0.17±0.095	3,000±2,200	3.9±2.6
	Physics	0.0048±0.00050	0.94±0.056	0.077±0.019	7,200±1,200	1.00±0.00
Burgers	Data-only	0.12±0.068	0.93±0.021	0.062±0.014	6,000±950	1.00±0.00
	Physics	0.12±0.069	0.93±0.021	0.062±0.014	6,200±670	1.00±0.00

These results indicate that the effect of including physics during inference is problem-dependent. For EIT, the physics constraint provides a clear and consistent improvement across all metrics, eliminating convergence failures, improving accuracy, and tightening credible intervals. For the continuous Darcy problem, the effect is inconsistent across test instances: physics reduces divergent transitions on average but can destabilise the sampler on individual instances, producing convergence failures that do not



(a) Darcy continuous

(b) Darcy piecewise

(c) EIT

(d) Burgers

Figure 4.13: Physics constraint calibration curves. Data-only and physics-informed posteriors are shown for each problem. The dashed line indicates ideal calibration. EIT shows the clearest improvement under physics, while the remaining problems show negligible or negative effects on calibration.

occur under the data-only posterior. For the piecewise-constant Darcy problem, the physics constraint degrades sampler convergence while narrowing credible intervals, suggesting that the additional information from $\mathcal{F}_{\text{pde}}$ creates a more complex posterior geometry. For Burgers, the physics constraint has no measurable effect, with both conditions producing identical metrics.

4.6. Out-of-distribution robustness

A method that produces uncertainty estimates should ideally recognise inputs that lie outside its training distribution. We test this by comparing in-domain and out-of-distribution test instances under the data-only posterior, using the OOD variants defined in Section 4.1.

For the continuous Darcy problem (Table 4.8), reconstruction error increases by roughly an order of magnitude under OOD inputs (rRMSE from 0.011 to 0.12). The posterior broadens in response, with mean standard deviation rising from 0.033 to 0.36 and CI width from 0.13 to 0.95, though not enough to maintain 98% coverage, which falls to 91%. The calibration curve for the OOD case falls below the line of perfect calibration at all nominal levels (Figure 4.14), indicating that the posterior partially detects the distributional shift by widening but underestimates the true increase in reconstruction uncertainty.

For the piecewise-constant Darcy problem (Table 4.8), coverage remains at 100% for both in-domain and OOD inputs, and all other metrics are essentially unchanged (I_{corr} 0.89 versus 0.90, CRPS 0.44 versus 0.42). The wide posterior associated with the piecewise-constant latent structure and the flow prior drives inference in both cases, so the distributional shift has no measurable effect.

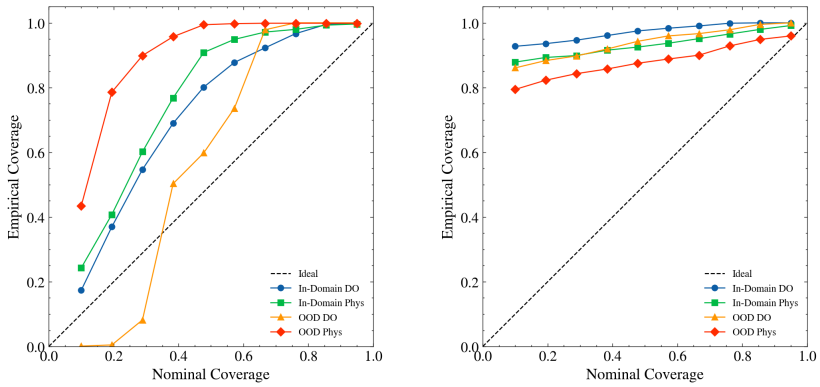
For EIT (Table 4.8, Figure 4.15), coverage drops from 97% in-domain to 23% under the OOD conductivity field. The posterior standard deviation for OOD inputs (0.041) is higher than in-domain (0.026), yet the posterior remains far too narrow to cover the greatly increased reconstruction error (rRMSE from 0.0050 to 0.076). The calibration curve falls well below the line of perfect calibration at all nominal levels (Figure 4.14).

For Burgers (Table 4.8, Figure 4.14), OOD inputs produce metrics indistinguishable from in-domain inputs (rRMSE 0.12 in both cases, coverage 0.97 versus 0.93, identical CI widths).

Table 4.8: In-domain vs out-of-distribution metrics (data-only posterior). Values are cross-instance means $\pm$ standard deviations across three test instances, each averaged over three seeds. For continuous problems, accuracy is rRMSE (lower is better). For piecewise Darcy, it is I_{corr} (higher is better).

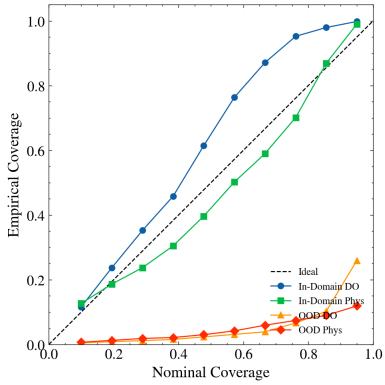
Problem	Setting	Accuracy	CRPS	Cov. 95%	CI width	$\bar{\sigma}$
Darcy continuous	In-domain	0.011 ± 0.0019	0.014 ± 0.0031	0.98 ± 0.019	0.13 ± 0.096	0.033 ± 0.024
	OOD	0.12 ± 0.037	0.21 ± 0.069	0.91 ± 0.076	0.95 ± 0.16	0.36 ± 0.12
Darcy piecewise	In-domain	0.90 ± 0.068	0.42 ± 0.25	1.00 ± 0.00051	4.4 ± 0.51	1.7 ± 0.24
	OOD	0.89 ± 0.048	0.44 ± 0.20	1.00 ± 0.0010	4.0 ± 1.1	1.6 ± 0.42
EIT	In-domain	0.0050 ± 0.00095	0.011 ± 0.0017	0.97 ± 0.025	0.10 ± 0.016	0.026 ± 0.0041
	OOD	0.076 ± 0.0086	0.14 ± 0.025	0.23 ± 0.22	0.15 ± 0.069	0.041 ± 0.020
Burgers	In-domain	0.12 ± 0.069	0.0084 ± 0.0030	0.93 ± 0.017	0.060 ± 0.015	0.015 ± 0.0038
	OOD	0.12 ± 0.053	0.0071 ± 0.0012	0.97 ± 0.023	0.060 ± 0.016	0.015 ± 0.0040

The OOD coverage results vary substantially across problems. Continuous Darcy shows partial detection of distributional shift, piecewise Darcy and Burgers are insensitive to it, and EIT fails to maintain coverage. For EIT, the coverage failure has two contributing factors. The flow prior penalises latent codes that lie outside the learned distribution, pulling posterior samples toward in-domain regions of latent space. Simultaneously, the coefficient decoder constrains what fields the posterior can represent at all. Without an ablation that removes the flow prior term, the relative contribution of each factor cannot be isolated from these experiments alone. The severity of the failure likely depends on the interaction between the two: if the OOD field lies far from anything the decoder can express, the prior concentrates density on the nearest representable field rather than broadening to reflect the model's limitations.

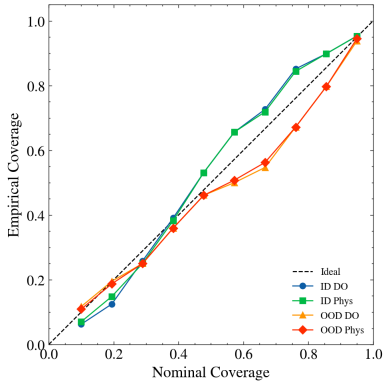


(a) Darcy continuous

(b) Darcy piecewise

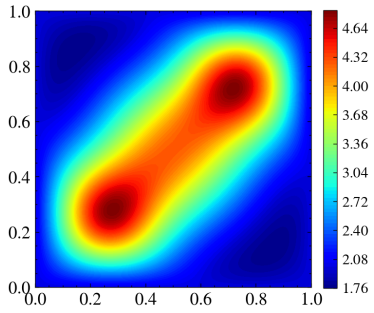


(c) EIT

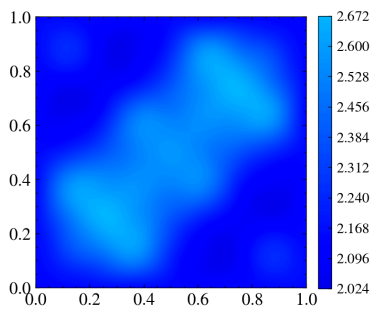


(d) Burgers

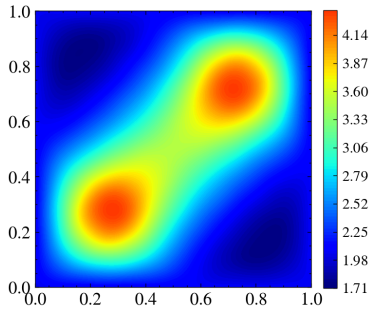
Figure 4.14: OOD calibration curves. The dashed line indicates ideal calibration.



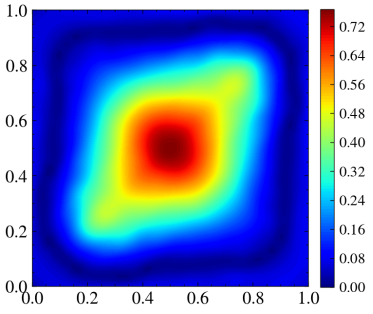
(a) a^{true}



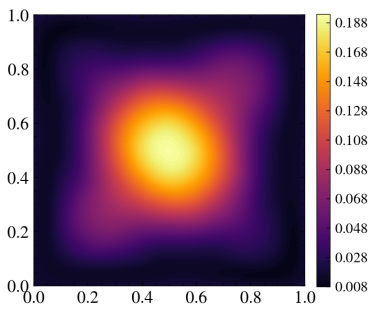
(b) $\tilde{a}^{\text{MAP}}$



(c) $\tilde{a}^{\text{post}}$



(d) $|\tilde{a}^{\text{post}} - a^{\text{true}}|$



(e) $\tilde{\sigma}_a^{\text{post}}$

Figure 4.15: EIT out-of-distribution results. From left to right: ground-truth OOD conductivity, Laplace MAP estimate, posterior mean, absolute error, and posterior standard deviation.

5

Discussion

5.1. Calibration and model discrepancy

Across the four benchmarks, point prediction quality is mixed: the Laplace posterior mean is closer to the truth on continuous Darcy, NUTS is substantially better on EIT, and the two are similar on Burgers (Table 4.3). The value of full MCMC therefore lies in calibrated uncertainty rather than improved point estimates. The latent-space potential energy $U(\beta_1) = \mathcal{F}_{\text{data}} + \mathcal{F}_{\text{pde}} + \mathcal{F}_{\text{prior}}$ is non-convex in β_1 because the decoders G_{θ_u} and G_{θ_a} are multi-layer neural networks. Although $\mathcal{F}_{\text{data}}$ measures a squared error, it does so after passing β_1 through G_{θ_u}, whose nonlinearity creates multiple local minima in the loss landscape. The same reasoning applies to $\mathcal{F}_{\text{pde}}$ and $\mathcal{F}_{\text{prior}}$. The Adam optimiser used for MAP estimation can therefore settle in a suboptimal local minimum, while MCMC explores the full posterior and the posterior mean draws on probability mass from regions that Adam never visits.

The calibration curves (Figure 4.9) split the four benchmarks into two groups. EIT and Burgers track the diagonal closely, indicating well-calibrated posteriors at the likelihood settings used. Both Darcy problems overcover at all nominal levels, but for different reasons. For continuous Darcy, the posterior is informative (CI width narrows $6\times$ relative to the prior) yet wider than necessary, consistent with σ_{data} exceeding the combined measurement and surrogate error. The approximation error framework of Kaipio and Somersalo [50] formalises this: the total observation error decomposes as $\varepsilon_{\text{total}} = \varepsilon_{\text{noise}} + (F(x) - f(x))$, where F is the true forward map and f the neural network surrogate, so σ_{data} should reflect the combined standard deviation rather than the measurement noise alone. We have not measured $F(x) - f(x)$ directly, but the overcoverage pattern and the near-constant CI width across the noise sweep (Table 4.5) are consistent with σ_{data} absorbing the surrogate error with margin to spare. Because MCMC does not introduce its own approximation bias, the overcoverage traces to the likelihood specification and can be corrected by tightening σ_{data}. For piecewise Darcy, overcoverage reflects an uninformative rather than a miscalibrated posterior: CI width is 4.3 out of a possible range of 5.0, the noise and sensor sweeps show near-complete insensitivity (Table 4.5, Table 4.6), and the posterior is dominated by the flow prior across all experimental conditions.

The Laplace approximation (Table 4.3) fails outright on piecewise Darcy and is unreliable on Burgers, while on EIT it produces oversmoothed posteriors with anti-correlated uncertainty. Only on continuous Darcy does Laplace converge reliably, where it achieves better CRPS and narrower credible intervals than NUTS, though NUTS still achieves a better NLL. The large cross-instance variance on this problem means neither method dominates conclusively. The computational cost of full MCMC is justified by the consistently well-calibrated posteriors it produces across all four problems.

5.2. Physics constraints during posterior sampling

The effect of including $\mathcal{F}_{\text{pde}}$ during sampling is problem-dependent (Table 4.7), and the pattern across benchmarks reveals that the physics constraint is not a universal benefit but rather one whose effectiveness depends on the interaction among observation model, posterior geometry, and hyperparameter calibration. For EIT, the physics constraint provides an unambiguous improvement: it eliminates con-

vergence failures, improves reconstruction accuracy, and tightens credible intervals. Because EIT recovers the interior conductivity from boundary-only measurements, the data likelihood leaves substantial ambiguity in the interior field, and the PDE residual provides gradient information in precisely those regions. For continuous Darcy, where dense interior sensors already constrain the posterior, the physics term is at best redundant and at worst destabilising, with one test instance exhibiting sampler failure under physics that does not occur under the data-only posterior. For piecewise Darcy, including physics narrows credible intervals but degrades sampler convergence, with divergent transitions and reduced ESS suggesting that $\mathcal{F}_{\text{pde}}$ reshapes the posterior geometry into a form that the sampler struggles to explore. For Burgers, the physics constraint has no measurable effect.

The physics constraint helps when data alone leaves the posterior under-determined, but the benefit is sensitive to how σ_{pde} interacts with the data likelihood. In our experiments, σ_{pde} was tuned per-problem and held fixed across test instances, yet the effect of including $\mathcal{F}_{\text{pde}}$ varied substantially even within a single benchmark, with one continuous Darcy instance exhibiting sampler failure under physics that did not occur under the data-only posterior. The practical value of including $\mathcal{F}_{\text{pde}}$ during sampling therefore depends less on whether physics is available than on whether the noise scale can be appropriately calibrated, motivating more systematic approaches to selecting σ_{pde} jointly with σ_{data}.

Across the four benchmarks, this pattern is consistent with the conditioning of each data-only inverse problem. Where observations are direct and dense, as in continuous Darcy with interior solution measurements and Burgers with spatiotemporal observations, the data likelihood already constrains the posterior tightly, and the PDE residual provides gradient information along directions that are already well-determined by the data. The solution decoder was itself trained with a PDE loss, so for in-distribution latent vectors it already produces physically consistent solution fields, and the explicit physics term adds little beyond what the decoder has learned. In this regime, the physics term does not reduce uncertainty but does reshape the posterior, and we hypothesise that this additional curvature is what destabilises the sampler in the continuous Darcy instance where convergence fails under physics. Where observations are indirect and leave large portions of the domain unconstrained, as in EIT with boundary-only measurements, the PDE residual provides gradient information in precisely the under-determined directions, producing the clearest improvement. Piecewise Darcy falls between these extremes, as the physics term narrows the posterior, indicating that it carries information the data likelihood does not, but the mixed-formulation PDE residual for this problem involves the coefficient values directly (Appendix A), so its sensitivity differs between the two phases, creating a posterior geometry that the sampler struggles to explore. The physics term therefore appears most useful when the data likelihood leaves the posterior under-determined, and counterproductive when it reshapes an already well-constrained posterior into a form that is harder to explore.

5.3. Learned prior limitations

A learned prior is valid only within its training distribution, so posterior estimates for OOD inputs are not guaranteed to reflect the true uncertainty. Our experiments confirm this (Table 4.8). The flow prior penalises any β_1 that deviates from the learned latent distribution, pulling samples towards regions of latent space that the encoder would produce for in-domain coefficient fields. When the true coefficient field lies outside the training distribution, the posterior concentrates on the nearest representable field rather than broadening to reflect the model's limitations. The severity depends on how far the true input is from anything the decoder can represent. When the OOD input is only mildly different from the training data, the posterior widens somewhat but not enough to maintain coverage. When the OOD input is qualitatively unlike the training data, the posterior collapses onto the wrong in-domain field with high confidence, producing uncertainty estimates that are both narrow and wrong (Table 4.8). This failure mode is distinct from the well-documented phenomenon of deep generative models assigning high likelihood to OOD data [51, 52]. In our setting, the prior correctly assigns low density to OOD latent vectors but uses that density to pull the posterior towards the learned manifold.

The OOD scenarios tested in Section 4.6, such as higher frequencies and larger amplitudes, represent mild distributional shift where the training and test distributions share the same functional form. In practice, the gap between synthetic training data and real-world observations is more severe. The coefficient fields used for training, smooth Gaussian random fields or simple binary geometries, are drastically simpler than real subsurface systems. Real geological formations exhibit multiscale hetero-

geneity and irregular structures far beyond what these synthetic distributions can represent [53]. When the training data cannot represent the complexity of the true coefficient field, the posterior will concentrate on the nearest in-domain field regardless of how different it is from reality, and the flow prior offers no indication that this has occurred.

The piecewise Darcy problem is the exception to this pattern. Coverage remains at 100% for both in-domain and OOD inputs, but this likely reflects the method's inability to produce informative posteriors for this problem rather than successful OOD detection. The posterior is dominated by the flow prior across all experimental conditions (Table 4.8, Table 4.5, Table 4.6), CI widths barely narrow from the unconditional prior, and the OOD and in-domain I_{corr} values show no statistically significant difference. The maintained coverage is coincidental rather than indicative of reliable uncertainty quantification.

This dependence on the training distribution limits the method's applicability as a black-box uncertainty quantifier. If the input may lie outside the training distribution, the posterior will underestimate uncertainty without any indication that it is doing so.

5.4. Limitations

The posterior is conditional on the pre-trained encoder, decoder, and normalising flow networks, so it quantifies uncertainty about the coefficient field but does not account for uncertainty in the network parameters themselves. The coefficient decoder G_{θ_a} constrains what the posterior can represent. If the decoder cannot express a particular field structure, no latent-space posterior will recover it. This limitation is inherent to latent-space Bayesian methods such as those of [35, 36], and differs from weight-space approaches [23] that integrate over network parameters. More generally, healthy convergence diagnostics do not guarantee that the posterior is informative, since they measure only whether the sampler has explored the target distribution, not whether that distribution is itself useful.

Most training-stage choices are inherited from the original IGNO implementation [7] without ablation. These include the per-problem loss weights w_{data} and w_{pde}, the encoder and decoder architectures, and the normalising flow configuration (Appendix F). The motivation behind these choices is not documented, and we do not know whether they were tuned or chosen by convention. The one change we made was reducing the latent dimension from 128 to between 6 and 16, after observing that most dimensions in the original configuration were effectively inactive, with standard deviations below 0.01. These dead dimensions caused the normalising flow to converge to degenerate solutions rather than capturing the structure of the active dimensions. This change was necessary for the normalising flow to function as a meaningful prior, but the reduced dimensions were chosen empirically rather than through a structured search.

The sampling procedure adds further hyperparameters on top of the inherited training configuration: the likelihood noise scale σ_{data}, the PDE noise scale σ_{pde}, the number of warmup and sampling steps, the mass matrix structure, and the target acceptance probability. Our experiments vary the problem conditions under which sampling is performed, such as noise level, sensor count, and whether physics constraints are included, but do not ablate the sampling hyperparameters themselves. σ_{data} is calibrated automatically per instance, and all other sampling settings are fixed across experiments for a given problem. Because training-stage and sampling-stage choices interact, a proper ablation would require retraining the IGNO model under each candidate setting and re-running the full downstream pipeline for each problem, test instance, and seed, which is prohibitively expensive. The effect of these choices on posterior quality therefore remains an open question.

The baseline and physics-constraint experiments evaluate on three test instances with three random seeds each, while the sensitivity sweeps fix a single test instance to isolate the effect of each varied quantity. Three test instances provide some coverage of instance-to-instance variability, but do not constitute a statistical evaluation across the test distribution. Evaluating across a larger test set would require running the full MCMC procedure for each instance, which is currently prohibitive. Wall-clock times per method and benchmark are reported in Appendix G. In particular, the noise and sensor-count sweep curves reflect the behaviour of a single test instance, so the variability of these trends across the test distribution is unknown. Repeating the full sweep grid across multiple test instances was not computationally feasible.

The piecewise-constant Darcy benchmark uses permeability values $k \in \{5, 10\}$, a 2:1 contrast ratio inherited from the DGenNO and IGNO experiments. This is at the mild end of the range found in computational studies of subsurface flow, where 12:1 contrasts appear in comparable Bayesian geometric inverse problems [5]. Real geological contrasts are far larger, with hydraulic conductivity varying over approximately thirteen orders of magnitude across common rock and soil types [53]. Our experiment is therefore a proof of concept for discrete-field posterior inference rather than a realistic subsurface scenario. The higher-contrast experiments confirm this limitation (Table 4.2). At 20:1 and 200:1 contrast, the solution decoder's test error rises from 0.025 to 0.59 and 0.96, meaning the forward model progressively fails to learn the PDE. The difficulty traces to the conditioning of the operator: for piecewise-constant coefficients, the conditioning of the discrete system scales with the contrast ratio $k_{\max}/k_{\min}$ [54]. The PDE residual (Appendix A) inherits this conditioning, so training gradients are dominated by errors in regions where k takes its larger value while remaining insensitive to errors elsewhere. This is visible in the training dynamics. The PDE loss fails to converge at 20:1 and 200:1 contrast while the data loss, which measures coefficient classification independent of the PDE operator, converges normally regardless of contrast. The loss weight w_{pde} was also held constant across contrast levels, which further exacerbated this problem. Scalar rescaling alone cannot resolve the internal ill-conditioning since it preserves the eigenvalue ratio within the PDE loss. Whether contrast-aware weighing could extend the method to realistic contrast ratios remains an open question.

In the classical numerical analysis sense, conditioning is a property of the mathematical problem, not the algorithm used to solve it, while stability is a property of the algorithm. If a problem is ill-conditioned, no algorithm, however stable, will produce an accurate answer. Several of the limitations above fall on the conditioning side: the contrast-ratio scaling of the PDE operator and the decoder's inability to represent certain field structures are intrinsic to the problem formulation and cannot be resolved by improving the inference algorithm. The sampler convergence difficulties fall on the stability side and could in principle be addressed with a better algorithm or better hyperparameter choices. This conditioning-versus-stability distinction directs future effort, since algorithmic improvements can only address stability-side limitations.

5.5. Future directions

The OOD experiments show that the posterior does not reliably detect distributional shift, and the gap between synthetic training distributions and real-world coefficient fields is likely far larger than anything tested here. Closing this gap is the main obstacle to using the method on real data. Fine-tuning the pretrained networks on a small set of real or high-fidelity observations is one way forward. The Bayesian framework is well suited to guide this process, since posterior diagnostics such as poor convergence or wide credible intervals can flag when the learned representations are inadequate for a given set of observations.

The posterior formulation used in this work decomposes into independent data, physics, and prior terms. The additive decomposition extends naturally to settings where the observation process involves more than sparse sampling of the solution field. The EIT benchmark already demonstrates this to some extent, as it recovers interior conductivity from boundary voltage measurements rather than direct field observations. In many practical inverse problems, the gap between the physical field and the observed data is wider, as it involves additional simulation stages. Replacing $\mathcal{F}_{\text{pde}}$ with a differentiable simulation of the full observation process requires only that the pipeline supports gradient computation with respect to β_1. Complex observation models also introduce additional uncertainty sources that a point estimate cannot capture but that the posterior can integrate over if they are encoded in the likelihood.

The computational overhead of MCMC relative to point estimation is substantial, ranging from approximately 16-fold for EIT to over 450-fold for piecewise Darcy (Appendix G), and limits the method to problems where the latent dimension is small enough for efficient mixing. Prior work reports MCMC failure at 512 dimensions [32]. A conditional normalising flow trained to approximate $p(\beta_1 \mid u_{\text{obs}})$ directly could provide fast initial posterior estimates at the cost of a single forward pass, with MCMC reserved for cases requiring higher accuracy or guaranteed convergence. Reducing the effective sampling dimension through automatic identification of the latent dimension [6] or hierarchical latent structures [7] would further extend the range of problems accessible to full posterior sampling.

6

Conclusion

PDE-constrained inverse problems are ill-posed, and responsible use of their solutions requires quantifying the uncertainty in recovered parameters. Neural operator methods such as IGNO achieve accurate deterministic reconstructions but produce only point estimates, leaving practitioners without a measure of confidence in the result. This thesis extended IGNO's inversion procedure to full Bayesian posterior sampling by adding a normalising flow prior term and replacing gradient-based optimisation with the No-U-Turn Sampler. The extension requires no retraining of any network component.

The experimental evaluation on four benchmark inverse problems addressed the three research questions posed in the introduction. For **RQ1**, Bayesian sampling in IGNO's latent space produces well-calibrated uncertainty estimates on in-domain problems, with empirical coverage between 93% and 100% at the nominal 95% level across all four benchmarks. The posterior mean matches or improves on the Laplace MAP point estimate in every case, and sensitivity analyses confirm that the posterior responds appropriately to changes in noise level and sensor count. The Laplace approximation, which is orders of magnitude cheaper, fails on two of the four problems and does not consistently outperform NUTS where it succeeds. For **RQ2**, the effect of including the PDE constraint during sampling is problem-dependent. The decoder was trained with a PDE loss and already produces physically consistent fields for in-distribution inputs, so the explicit physics term adds little when observations are dense. The physics constraint is most beneficial when observations alone leave the posterior underdetermined, as demonstrated by EIT, where boundary-only measurements leave the interior field unconstrained and including the PDE residual eliminates convergence failures, improves accuracy, and tightens credible intervals. For **RQ3**, the method does not reliably detect out-of-distribution coefficient fields. The flow prior pulls the posterior toward the learned latent manifold, producing narrow credible intervals that miss the true field. The severity varies across problems, from partial detection of distributional shift on continuous Darcy to the failure seen on EIT, where coverage drops from 97% to 23%.

The method's main limitation is its dependence on the training distribution. Because the flow prior actively concentrates the posterior on the learned manifold, the uncertainty estimates are reliable only when the true coefficient field is representable by the decoder. The most pressing open problem is the gap between synthetic training data and real-world observations. Fine-tuning the pre-trained components on limited real data, or replacing them with more expressive generative architectures, are two routes worth pursuing. More broadly, the fundamental components of the inference procedure, a low-dimensional latent space, a differentiable decoder, and a trainable density model on the latent representation, are not unique to IGNO. Investigating whether the same posterior sampling approach transfers to other encoder-decoder architectures is a natural direction for future work.

Declaration on the use of generative AI

This declaration is made in accordance with the EEMCS faculty guidelines on generative AI in end projects. The research direction, experimental design, mathematical derivations, and all technical contributions presented in this thesis are original work. Generative AI tools were used in a supporting capacity as described below, and all AI-generated output was checked and corrected where necessary before inclusion.

The experimental codebase required porting a PyTorch reference implementation to JAX in order to use NumPyro for MCMC sampling. Claude Opus 4.6 (Anthropic) was used to review the correctness of the resulting JAX code, to identify and fix bugs, and to debug numerical precision issues encountered on the TU Delft DAIC compute cluster, where floating-point behaviour differed across hardware environments. AI also assisted with routine development tasks such as extending experiment scripts written for one problem configuration to additional configurations, generating boilerplate infrastructure code, and adding documentation and notes to the experiments and core framework for future researchers. Claude also assisted in writing the Python scripts used to produce the plots and figures in this thesis; I reviewed and manually adjusted these scripts before use.

The mathematical derivations in this thesis are my own work, carried out by hand. Informal shorthand drafts of the equations were converted into properly formatted LaTeX with the help of Claude, which also helped enforce consistent notation across the document.

During the writing phase, Claude Opus 4.6 and Claude Sonnet 4.6 were used to improve the grammar, readability, and conciseness of prose I had written, to smooth transitions between sections, and to assist with LaTeX formatting. All arguments and technical analysis are original.

The cover illustration was generated using ChatGPT Images 2.0 (OpenAI) based on prompts I wrote. It serves as a conceptual visualisation and does not depict real experimental data.

Acknowledgements

Research reported in this work was facilitated by computational resources and support of the Delft AI Cluster (DAIC) at TU Delft [55] (RRID: SCR_025091), but remains the sole responsibility of the author, not the DAIC team.

References

[1] Robert H. Stolt and Arthur B. Weglein. *Seismic imaging and inversion: application of linear inverse theory*. Cambridge, UK ; New York: Cambridge University Press, 2012. 404 pp. isbn: 978-1-107-01490-9.

[2] Avinash C. Kak and Malcolm Slaney. *Principles of Computerized Tomographic Imaging*. Society for Industrial and Applied Mathematics, Jan. 2001. isbn: 978-0-89871-494-4 978-0-89871-927-7. doi: 10.1137/1.9780898719277. url: http://epubs.siam.org/doi/book/10.1137/1.9780898719277 (visited on 02/28/2026).

[3] Mikdam Jamal and Michael N. Morgan. "Characterization of Material Properties Based on Inverse Finite Element Modelling". In: *Inventions* 4.3 (Aug. 2, 2019), p. 40. issn: 2411-5134. doi: 10.3390/inventions4030040. url: https://www.mdpi.com/2411-5134/4/3/40 (visited on 02/28/2026).

[4] George S. Dulikravich et al. "Inverse determination of spatially varying material coefficients in solid objects". In: *Journal of Inverse and Ill-posed Problems* 24.2 (Apr. 1, 2016), pp. 181–194. issn: 0928-0219, 1569-3945. doi: 10.1515/jiip-2015-0057. url: https://www.degruyter.com/document/doi/10.1515/jiip-2015-0057/html (visited on 02/28/2026).

[5] Marco A Iglesias, Kui Lin, and Andrew M Stuart. "Well-posed Bayesian geometric inverse problems arising in subsurface flow". In: *Inverse Problems* 30.11 (Nov. 1, 2014), p. 114001. issn: 0266-5611, 1361-6420. doi: 10.1088/0266-5611/30/11/114001. url: https://iopscience.iop.org/article/10.1088/0266-5611/30/11/114001 (visited on 02/28/2026).

[6] Yaohua Zang and Phaedon-Stelios Koutsourelakis. "DGenNO: a novel physics-aware neural operator for solving forward and inverse PDE problems based on deep, generative probabilistic modeling". In: *Journal of Computational Physics* 538 (Oct. 2025), p. 114137. issn: 00219991. doi: 10.1016/j.jcp.2025.114137. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999125004206 (visited on 11/13/2025).

[7] Gang Bao and Yaohua Zang. *A unified physics-informed generative operator framework for general inverse problems*. Version Number: 1. 2025. doi: 10.48550/ARXIV.2511.03241. url: https://arxiv.org/abs/2511.03241 (visited on 11/16/2025).

[8] A. M. Stuart. "Inverse problems: A Bayesian perspective". In: *Acta Numerica* 19 (May 2010), pp. 451–559. issn: 0962-4929, 1474-0508. doi: 10.1017/S0962492910000061. url: https://www.cambridge.org/core/product/identifier/S0962492910000061/type/journal_article (visited on 02/08/2026).

[9] Sergej Igorevič Kabanihin. *Inverse and Ill-posed problems: theory and applications*. Inverse and ill-posed problems series Volume 55. Berlin: De Gruyter, 2012. isbn: 978-3-11-022400-9.

[10] Lorenz T. Biegler et al., eds. *Large-Scale PDE-Constrained Optimization*. Vol. 30. Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. isbn: 978-3-540-05045-2 978-3-642-55508-4. doi: 10.1007/978-3-642-55508-4. url: https://link.springer.com/10.1007/978-3-642-55508-4.

[11] Guy Chavent. *Nonlinear Least Squares for Inverse Problems: Theoretical Foundations and Step-by-Step Guide for Applications*. Scientific Computation. Dordrecht: Springer Netherlands, 2010. isbn: 978-90-481-2784-9 978-90-481-2785-6. doi: 10.1007/978-90-481-2785-6. url: http://link.springer.com/10.1007/978-90-481-2785-6 (visited on 12/01/2025).

[12] Tan Bui-Thanh et al. "A Computational Framework for Infinite-Dimensional Bayesian Inverse Problems Part I: The Linearized Case, with Application to Global Seismic Inversion". In: *SIAM Journal on Scientific Computing* 35.6 (Jan. 2013), A2494–A2523. issn: 1064-8275, 1095-7197. doi: 10.1137/12089586X. url: http://epubs.siam.org/doi/10.1137/12089586X (visited on 04/12/2026).

[13] Youssef M. Marzouk, Habib N. Najm, and Larry A. Rahn. "Stochastic spectral methods for efficient Bayesian solution of inverse problems". In: *Journal of Computational Physics* 224.2 (June 2007), pp. 560–586. issn: 00219991. doi: 10.1016/j.jcp.2006.10.010. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999106004839 (visited on 03/01/2026).

[14] Lu Lu et al. “Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators”. In: *Nature Machine Intelligence* 3.3 (Mar. 18, 2021), pp. 218–229. issn: 2522-5839. doi: 10.1038/s42256-021-00302-5. url: https://www.nature.com/articles/s42256-021-00302-5 (visited on 12/02/2025).

[15] Zongyi Li et al. “Fourier Neural Operator for Parametric Partial Differential Equations”. In: *International Conference on Learning Representations*. 2021. url: https://openreview.net/forum?id=c8P9NQVtmnO.

[16] Nikola Kovachki et al. “Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs”. In: *Journal of Machine Learning Research* 24.89 (2023), pp. 1–97. url: http://jmlr.org/papers/v24/21-1524.html.

[17] Arnaud Vadeboncoeur et al. “Fully probabilistic deep models for forward and inverse problems in parametric PDEs”. In: *Journal of Computational Physics* 491 (Oct. 2023), p. 112369. issn: 00219991. doi: 10.1016/j.jcp.2023.112369. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999123004643 (visited on 11/13/2025).

[18] Chuang Wang and Tian Wang. “Latent Neural Operator for Solving Forward and Inverse PDE Problems”. In: *Advances in Neural Information Processing Systems 37*. Advances in Neural Information Processing Systems 37. Vancouver, BC, Canada: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024, pp. 33085–33107. isbn: 979-8-3313-1438-5. doi: 10.52202/079017-1042. url: http://www.proceedings.com/079017-1042.html (visited on 11/13/2025).

[19] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM monographs on mathematical modeling and computation. Philadelphia: SIAM, 1998. 247 pp. isbn: 978-0-89871-403-6.

[20] D. Calvetti and E. Somersalo. “Inverse problems: From regularization to Bayesian inference”. In: *WIREs Computational Statistics* 10.3 (May 2018), e1427. issn: 1939-5108, 1939-0068. doi: 10.1002/wics.1427. url: https://wires.onlinelibrary.wiley.com/doi/10.1002/wics.1427 (visited on 02/08/2026).

[21] Tianping Chen and Hong Chen. “Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems”. In: *IEEE Transactions on Neural Networks* 6.4 (July 1995), pp. 911–917. issn: 10459227. doi: 10.1109/72.392253. url: http://ieeexplore.ieee.org/document/392253/ (visited on 12/02/2025).

[22] M. Raissi, P. Perdikaris, and G.E. Karniadakis. “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”. In: *Journal of Computational Physics* 378 (Feb. 2019), pp. 686–707. issn: 00219991. doi: 10.1016/j.jcp.2018.10.045. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999118307125 (visited on 12/03/2025).

[23] Liu Yang, Xuhui Meng, and George Em Karniadakis. “B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data”. In: *Journal of Computational Physics* 425 (Jan. 2021), p. 109913. issn: 00219991. doi: 10.1016/j.jcp.2020.109913. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999120306872 (visited on 02/28/2026).

[24] Christopher M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. New York: Springer, 2006. 738 pp. isbn: 978-0-387-31073-2.

[25] Guang Lin, Christian Moya, and Zecheng Zhang. “B-DeepONet: An enhanced Bayesian DeepONet for solving noisy parametric PDEs using accelerated replica exchange SGLD”. In: *Journal of Computational Physics* 473 (Jan. 2023), p. 111713. issn: 00219991. doi: 10.1016/j.jcp.2022.111713. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999122007768 (visited on 02/08/2026).

[26] Shailesh Garg and Souvik Chakraborty. “VB-DeepONet: A Bayesian operator learning framework for uncertainty quantification”. In: *Engineering Applications of Artificial Intelligence* 118 (Feb. 2023), p. 105685. issn: 09521976. doi: 10.1016/j.engappai.2022.105685. url: https://linkinghub.elsevier.com/retrieve/pii/S0952197622006753 (visited on 02/08/2026).

[27] W. K. Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (Apr. 1, 1970), pp. 97–109. issn: 1464-3510, 0006-3444. doi: 10.1093/biomet/57.1.97. url: https://academic.oup.com/biomet/article/57/1/97/284580 (visited on 03/01/2026).

[28] Radford M. Neal. “MCMC Using Hamiltonian Dynamics”. In: Steve Brooks et al. *Handbook of Markov Chain Monte Carlo*. 1st ed. New York: Chapman and Hall/CRC, May 10, 2011, pp. 113–162. isbn: 978-0-429-13850-8. doi: 10.1201/b10905-6. url: https://www.taylorfrancis.com/books/9780429138508/chapters/10.1201/b10905-6 (visited on 02/28/2026).

[29] Albert Tarantola. *Inverse problem theory and methods for model parameter estimation*. In collab. with Society for Industrial and Applied Mathematics. Other titles in applied mathematics 89. Philadelphia, Pa: Society for Industrial and Applied Mathematics, 2005. 1 p. isbn: 978-0-89871-572-9 978-0-89871-792-1. doi: 10.1137/1.9780898717921.

[30] Laurent Dinh, David Krueger, and Yoshua Bengio. “NICE: Non-linear Independent Components Estimation”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. url: http://arxiv.org/abs/1410.8516.

[31] George Papamakarios et al. “Normalizing Flows for Probabilistic Modeling and Inference”. In: *Journal of Machine Learning Research* 22.57 (2021), pp. 1–64. url: http://jmlr.org/papers/v22/19-1028.html.

[32] Agnimitra Dasgupta et al. “A dimension-reduced variational approach for solving physics-based inverse problems using generative adversarial network priors and normalizing flows”. In: *Computer Methods in Applied Mechanics and Engineering* 420 (Feb. 2024), p. 116682. issn: 00457825. doi: 10.1016/j.cma.2023.116682. url: https://linkinghub.elsevier.com/retrieve/pii/S0045782523008058 (visited on 02/08/2026).

[33] Sebastian Kaltenbach, Paris Perdikaris, and Phaedon-Stelios Koutsourelakis. “Semi-supervised invertible neural operators for Bayesian inverse problems”. In: *Computational Mechanics* 72.3 (Sept. 2023), pp. 451–470. issn: 0178-7675, 1432-0924. doi: 10.1007/s00466-023-02298-8. url: https://link.springer.com/10.1007/s00466-023-02298-8 (visited on 12/02/2025).

[34] Diederik P. Kingma and Max Welling. “Auto-Encoding Variational Bayes”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. url: http://arxiv.org/abs/1312.6114.

[35] Xuhui Meng et al. “Learning functional priors and posteriors from data and physics”. In: *Journal of Computational Physics* 457 (May 2022), p. 111073. issn: 00219991. doi: 10.1016/j.jcp.2022.111073. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999122001358 (visited on 02/08/2026).

[36] Xinchao Jiang et al. “Resolution-independent generative models based on operator learning for physics-constrained Bayesian inverse problems”. In: *Computer Methods in Applied Mechanics and Engineering* 420 (Feb. 2024), p. 116690. issn: 00457825. doi: 10.1016/j.cma.2023.116690. url: https://linkinghub.elsevier.com/retrieve/pii/S0045782523008137 (visited on 02/09/2026).

[37] Xuhui Meng. “Variational inference in neural functional prior using normalizing flows: application to differential equation and operator learning problems”. In: *Applied Mathematics and Mechanics* 44.7 (July 2023), pp. 1111–1124. issn: 0253-4827, 1573-2754. doi: 10.1007/s10483-023-2997-7. url: https://link.springer.com/10.1007/s10483-023-2997-7 (visited on 02/08/2026).

[38] Matthew James Beal. “Variational Algorithms for Approximate Bayesian Inference”. PhD thesis. Gatsby Computational Neuroscience Unit, University College London, 2003.

[39] Sebastian Kaltenbach and Phaedon-Stelios Koutsourelakis. “Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems”. In: *Journal of Computational Physics* 419 (Oct. 2020), p. 109673. issn: 00219991. doi: 10.1016/j.jcp.2020.109673. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999120304472 (visited on 12/09/2025).

[40] Maximilian Rixner and Phaedon-Stelios Koutsourelakis. “A probabilistic generative model for semi-supervised training of coarse-grained surrogates and enforcing physical constraints through virtual observables”. In: *Journal of Computational Physics* 434 (June 2021), p. 110218. issn: 00219991. doi: 10.1016/j.jcp.2021.110218. url: https://linkinghub.elsevier.com/retrieve/pii/S0021999121001133 (visited on 03/01/2026).

[41] Yaohua Zang and Gang Bao. *ParticleWNN: a Novel Neural Networks Framework for Solving Partial Differential Equations*. Nov. 12, 2023. doi: 10.48550/arXiv.2305.12433. arXiv: 2305.12433[cs]. url: http://arxiv.org/abs/2305.12433 (visited on 12/09/2025).

[42] Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. url: http://arxiv.org/abs/1412.6980.

[43] Matthew D. Hoffman and Andrew Gelman. “The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo”. In: *Journal of Machine Learning Research* 15.47 (2014), pp. 1593–1623. url: http://jmlr.org/papers/v15/hoffman14a.html.

[44] Tobin A Driscoll, Nicholas Hale, and Lloyd N Trefethen. *Chebfun guide*. 2014.

[45] Tilmann Gneiting and Adrian E Raftery. “Strictly Proper Scoring Rules, Prediction, and Estimation”. In: *Journal of the American Statistical Association* 102.477 (Mar. 2007), pp. 359–378. issn: 0162-1459, 1537-274X. doi: 10.1198/016214506000001437. url: http://www.tandfonline.com/doi/abs/10.1198/016214506000001437 (visited on 05/03/2026).

[46] Andrew Gelman and Donald B. Rubin. “Inference from Iterative Simulation Using Multiple Sequences”. In: *Statistical Science* 7.4 (Nov. 1, 1992). issn: 0883-4237. doi: 10.1214/ss/1177011136. url: https://projecteuclid.org/journals/statistical-science/volume-7/issue-4/Inference-from-Iterative-Simulation-Using-Multiple-Sequences/10.1214/ss/1177011136.full (visited on 02/28/2026).

[47] Andrew Gelman. *Bayesian data analysis*. Third edition. Chapman & Hall/CRC texts in statistical science. Boca Raton: CRC Press, 2014. 661 pp. isbn: 978-1-4398-4095-5.

[48] Aki Vehtari et al. “Rank-Normalization, Folding, and Localization: An Improved Rˆ for Assessing Convergence of MCMC (with Discussion)”. In: *Bayesian Analysis* 16.2 (June 1, 2021). issn: 1936-0975. doi: 10.1214/20-BA1221. url: https://projecteuclid.org/journals/bayesian-analysis/volume-16/issue-2/Rank-Normalization-Folding-and-Localization--An-Improved-R%cb%86-for/10.1214/20-BA1221.full (visited on 02/28/2026).

[49] Michael Betancourt. “A conceptual introduction to Hamiltonian Monte Carlo”. In: (Jan. 2017). arXiv: 1701.02434 [stat.ME].

[50] Jari Kaipio and E Somersalo. *Statistical and computational inverse problems*. en. 2005th ed. Applied Mathematical Sciences. New York, NY: Springer, Dec. 2004.

[51] Eric Nalisnick et al. “Do Deep Generative Models Know What They Don’t Know?” In: *International Conference on Learning Representations*. 2019. url: https://openreview.net/forum?id=H1xwNhCcYm.

[52] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. “Why Normalizing Flows Fail to Detect Out-of-Distribution Data”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 20578–20589. url: https://proceedings.neurips.cc/paper_files/paper/2020/file/ecb9fe2fbb99c31f567e9823e884dbec-Paper.pdf.

[53] R. Allan Freeze and John A. Cherry. *Groundwater*. Englewood Cliffs, N.J: Prentice-Hall, 1979. 604 pp. isbn: 978-0-13-365312-0.

[54] Daniel Peterseim and Robert Scheichl. “Robust Numerical Upscaling of Elliptic Multiscale Problems at High Contrast”. In: *Computational Methods in Applied Mathematics* 16.4 (2016), pp. 579–603. doi: 10.1515/cmam-2016-0022.

[55] Delft AI Cluster (DAIC). *The Delft AI Cluster (DAIC), RRID:SCR_025091*. 2024. doi: 10.4233/rrid:scr_025091. url: https://daic.tudelft.nl/.

[56] Holger Wendland. “Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree”. In: *Advances in Computational Mathematics* 4.1 (Dec. 1995), pp. 389–396. issn: 1019-7168, 1572-9044. doi: 10.1007/BF02123482. url: http://link.springer.com/10.1007/BF02123482 (visited on 02/28/2026).

[57] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. http://github.com/jax-ml/jax. Version 0.3.13. 2018.

[58] Conor Durkan et al. “Neural Spline Flows”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. url: https://proceedings.neurips.cc/paper_files/paper/2019/file/7ac71d433f282034e088473244df8c02-Paper.pdf.

[59] Du Phan, Neeraj Pradhan, and Martin Jankowiak. “Composable Effects for Flexible and Accelerated Probabilistic Programming in NumPyro”. In: *Program Transformations for ML Workshop at NeurIPS 2019*. 2019. url: https://openreview.net/forum?id=H1g1niFhIB.

A

Weak-form PDE residuals

This appendix derives the weak-form PDE residuals used in $\mathcal{F}_{\text{pde}}$ for each benchmark problem. All four benchmarks use the Particle Weak-form Neural Network (ParticleWNN) framework [41], which replaces strong-form collocation with integration against compactly supported test functions.

A.1. ParticleWNN framework

The ParticleWNN approach distributes N_c collocation points $\{\boldsymbol{x}_j\}_{j=1}^{N_c}$ within the domain Ω and associates each with a compactly supported radial basis function (CSRBF) test function w_j centred at $\boldsymbol{x}_j$ with support radius R_j. The test functions are Wendland CSRBFs, which are smooth, non-negative, and vanish outside $B(\boldsymbol{x}_j, R_j)$ [41, 56]. For a PDE of the form $\mathcal{N}_a[u] = f$, the weak-form residual against test function w_j is obtained by multiplying both sides by w_j and integrating over the support ball $B(\boldsymbol{x}_j, R_j)$. The weak residual for collocation point j is

$$r_{w_j}(a, u) = \frac{1}{|B(\boldsymbol{x}_j, R_j)|} \int_{B(\boldsymbol{x}_j, R_j)} \mathcal{I}_j(a, u; \boldsymbol{x})\, d\boldsymbol{x},$$

where $\mathcal{I}_j$ denotes the PDE-specific weak-form integrand obtained by multiplying the PDE by w_j and, where applicable, integrating by parts. The integrand depends on the specific PDE. The normalisation by $|B(\boldsymbol{x}_j, R_j)|$ ensures residuals are comparable across support balls of different sizes. Integration is approximated by quadrature on an $n_{\text{grid}} \times n_{\text{grid}}$ uniform grid within each support ball.

A.2. Continuous Darcy flow

The strong-form PDE is $-\nabla \cdot (k \nabla p) = f$ on $\Omega = [0, 1]^2$ with $p = 0$ on $\partial\Omega$. Multiplying by test function w_j and integrating by parts (using that w_j vanishes on the boundary of its support ball) yields

$$\int_{B_j} k\, \nabla p \cdot \nabla w_j\, d\boldsymbol{x} = \int_{B_j} f\, w_j\, d\boldsymbol{x},$$

where $B_j = B(\boldsymbol{x}_j, R_j)$. The weak-form residual is

$$r_{w_j}(k, p) = \frac{1}{|B_j|} \int_{B_j} (k\, \nabla p \cdot \nabla w_j - f\, w_j)\, d\boldsymbol{x}, \tag{A.1}$$

with $k = G_{\theta_a}(\beta_1)$ and $p(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$. This formulation requires only first-order derivatives of p, which the MultiONet architecture provides via automatic differentiation. Quadrature uses $n_{\text{grid}} = 9$.

A.3. Piecewise-constant Darcy flow

For piecewise-constant permeability, the strong-form PDE $-\nabla \cdot (k \nabla p) = f$ involves a discontinuous coefficient $k \in \{5, 10\}$. The IGNO implementation uses a mixed formulation with auxiliary stress variables

$\boldsymbol{s} = (s_1, s_2)$ to handle this:

$$\begin{aligned} \boldsymbol{s} &= k\,\nabla p, \\ -\nabla \cdot \boldsymbol{s} &= f. \end{aligned}$$

Two separate MultiONet stress decoders $G_{\theta_{s_1}}$ and $G_{\theta_{s_2}}$ predict the stress components. The weak-form PDE residual combines a constitutive residual and a divergence residual. The constitutive residual penalises disagreement between the predicted stress and $k\nabla p$:

$$r_{w_j}^{\text{const}}(k, p, \boldsymbol{s}) = \frac{1}{|B_j|} \int_{B_j} \|\boldsymbol{s} - k\,\nabla p\|^2 \, d\boldsymbol{x}.$$

The divergence residual enforces the PDE via integration by parts:

$$r_{w_j}^{\text{div}}(\boldsymbol{s}) = \frac{1}{|B_j|} \int_{B_j} (\boldsymbol{s} \cdot \nabla w_j - f\, w_j)\, d\boldsymbol{x}.$$

The total weak-form residual combines both: $r_{w_j} = r_{w_j}^{\text{const}} + \sqrt{N_c}\, r_{w_j}^{\text{div}}$. Quadrature uses $n_{\text{grid}} = 7$.

A.4. Electrical impedance tomography

The strong-form PDE is $-\nabla \cdot (\gamma \nabla u) = 0$ on $\Omega = [0, 1]^2$ with $u = g$ on $\partial\Omega$, where $\gamma > 0$ is the conductivity. Since the source term vanishes, integrating against w_j and applying integration by parts yields

$$\int_{B_j} \gamma\, \nabla u \cdot \nabla w_j \, d\boldsymbol{x} = 0.$$

The weak-form residual is

$$r_{w_j}(\gamma, u) = \frac{1}{|B_j|} \int_{B_j} \gamma\, \nabla u \cdot \nabla w_j \, d\boldsymbol{x}, \tag{A.2}$$

with $\gamma = G_{\theta_a}(\beta_1)$ and $u(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$. For EIT, the residual is evaluated for each of the $L = 20$ boundary conditions, with $\boldsymbol{\beta} = (\beta_1, \beta_2^{(l)})$ for the l-th boundary pattern. The total PDE residual averages over all boundary conditions. Quadrature uses $n_{\text{grid}} = 7$.

A.5. Burgers equation

The strong-form PDE is $\partial_t u + u\partial_x u = \lambda \partial_{xx} u$ on $x \in [-1, 1]$, $t \in [0, 1]$ with $u(\pm 1, t) = 0$ enforced by the mollifier $m(x) = \sin(\pi x/2 + \pi/2)$ applied to the decoder output. The solution decoder predicts $\tilde{u}(x, t; \boldsymbol{\beta})$ and the physical field is $u = \tilde{u} \cdot m(x)$. Distributing the PDE over the mollified field and multiplying by a 1D Wendland test function w_j centred at (x_j, t_j) with radius R_j, then integrating over the support interval $[x_j - R_j, x_j + R_j]$, yields the weak-form integrand

$$\mathcal{I}_j(u; x, t) = \partial_t u \cdot w_j + u\, \partial_x u \cdot w_j + \lambda\, \partial_x u \cdot \partial_x w_j,$$

where integration by parts has been applied to the viscosity term (using that w_j vanishes at the boundary of its support). The weak-form residual is

$$r_{w_j}(u) = \frac{1}{|[x_j - R_j, x_j + R_j]|} \int_{x_j - R_j}^{x_j + R_j} \mathcal{I}_j(u; x, t_j)\, dx.$$

Integration uses 1D Wendland test functions with $n_{\text{grid}} = 10$ quadrature points per support interval. The collocation points (x_j, t_j) are sampled uniformly over $[-1, 1] \times [0, 1]$ at each training iteration.

B

MCMC implementation details

This appendix describes the MCMC sampler used in Section 3.4 and reports convergence diagnostics for all experiments.

B.1. Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) achieves faster mixing than random-walk Metropolis-Hastings by incorporating gradient information [28]. HMC expresses the target density through a potential energy $U(q)$, with $p(q) \propto \exp(-U(q))$, and introduces auxiliary momentum variables p with kinetic energy $K(p) = p^\top M^{-1} p/2$, where M is a mass matrix. Each iteration draws fresh momentum from $\mathcal{N}(0, M)$ and simulates Hamiltonian dynamics on $H(q,p) = U(q) + K(p)$ using the leapfrog integrator, proposing a new state far from the current one. Because the dynamics approximately conserve energy, these distant proposals are accepted at high rates, avoiding the slow diffusive exploration of random-walk methods. A Metropolis correction step ensures the chain has the exact target distribution as its stationary distribution despite the finite step-size discretisation error. In practice, HMC requires tuning two parameters: the leapfrog step size ϵ and the number of integration steps L per proposal.

B.2. No-U-Turn Sampler

A critical challenge in HMC is choosing the trajectory length $L\epsilon$. NUTS adaptively sets trajectory length by simulating forward and backward in time until the trajectory begins to return towards its starting point (a U-turn condition). NUTS uses a binary tree structure to efficiently detect this. It also adapts the step size ϵ during a warmup phase via dual averaging, targeting a desired acceptance rate (typically 0.8). This automatic tuning makes HMC practical without extensive hyperparameter search.

Algorithm 1 summarises the full posterior sampling procedure built on top of NUTS. The algorithm takes as input the pre-trained IGNO networks and a set of observations, constructs the potential energy function from the terms defined in Section 3.4, and returns an ensemble of plausible coefficient and solution fields.

Algorithm 1 Posterior sampling via NUTS in IGNO latent space

Input: Pre-trained networks G_{θ_u}, G_{θ_a}, F_ϕ; observations u_{obs}; noise scale σ_{data}; PDE noise scale σ_{pde} (optional); warmup iterations N_w; sampling iterations N_s; number of chains C
Output: Posterior coefficient fields $\{\tilde{a}^{(i)}\}_{i=1}^{C\cdot N_s}$ and solution fields $\{\tilde{u}^{(i)}\}_{i=1}^{C\cdot N_s}$

Define potential energy:
$U(\beta_1) \leftarrow \mathcal{F}_{\text{data}}(\beta_1;\, G_{\theta_u},\, u_{\text{obs}},\, \sigma_{\text{data}}) + \mathcal{F}_{\text{prior}}(\beta_1;\, F_\phi)$
if PDE constraints included **then**
  $U(\beta_1) \leftarrow U(\beta_1) + \mathcal{F}_{\text{pde}}(\beta_1;\, G_{\theta_u},\, G_{\theta_a},\, \sigma_{\text{pde}})$
end if

Initialise:
$\beta_1^{(0)} \leftarrow F_\phi^{-1}(\mathbf{0})$ ▷ NF mode point

Warmup (N_w iterations): adapt step size ϵ and mass matrix M via NUTS [43]

Sample:
for $c = 1, \ldots, C$ **in parallel do**
  for $i = 1, \ldots, N_s$ **do**
    $\beta_1^{(i,c)} \leftarrow \text{NUTS}(U,\, \nabla_{\beta_1} U,\, \beta_1^{(i-1,c)},\, \epsilon,\, M)$
  end for
end for

Decode:
for each sample $\beta_1^{(i,c)}$ **do**
  $\tilde{a}^{(i,c)} \leftarrow G_{\theta_a}(\beta_1^{(i,c)})$
  $\tilde{u}^{(i,c)}(\cdot) \leftarrow G_{\theta_u}(\beta_1^{(i,c)},\, \beta_2,\, \cdot)$
end for

B.3. Implementation notes

Our implementation uses automatic differentiation (JAX framework) to compute gradients of the potential energy $U(\beta_1)$ through IGNO's fixed neural networks [57]. The NUTS step size, trajectory length, and mass matrix are all adapted during warmup. For problems with $d_1 \leq 10$ (continuous Darcy and EIT), a full dense mass matrix is adapted, while problems with $d_1 > 10$ (Burgers and piecewise-constant Darcy) use a diagonal mass matrix. The target acceptance probability is 0.85 for most configurations and 0.95 when the observation noise is small ($\sigma_{\text{data}} < 0.02$). For the high-dimensional piecewise-constant Darcy problem ($d_1 = 200$), the target acceptance probability is reduced to 0.65 and the maximum tree depth to 8.

Table B.1: MCMC sampling configuration per benchmark problem.

Problem	Warmup	Samples/chain	Chains	Total samples
Darcy continuous	5,000	2,000	4	8,000
Darcy piecewise	15,000	5,000	4	20,000
EIT	5,000	2,000	4	8,000
Burgers	5,000	2,000	4	8,000

For the physics-constrained Burgers posterior (Section 4.5), the sampling count is increased to 3,000 per chain (12,000 total) to improve effective sample size under the more complex posterior geometry introduced by the PDE residual term. The piecewise-constant Darcy problem uses sequential rather than parallel chain execution due to the memory requirements of NUTS at $d_1 = 200$.

The observation noise scale σ_{data} is selected per problem using one of two methods. Because the pre-trained neural operator acts as an approximate forward model, σ_{data} must account for model discrepancy in addition to measurement noise [50]. For piecewise-constant Darcy, the MAP prediction achieves low residual error at sensor locations, so σ_{data} is set to the RMSE of the MAP forward-model prediction at those locations, floored at the known measurement noise level. This captures the residual observation uncertainty after optimisation and adapts automatically to each test instance. For continuous Darcy, EIT, and Burgers, σ_{data} is selected from a problem-specific candidate grid via short pilot MCMC runs. Each pilot runs 2 chains for 2,000 warmup and 500 sampling iterations, initialised at the flow mode. Posterior samples are decoded to coefficient fields and the empirical 95% credible interval coverage is computed. The candidate whose coverage is closest to the nominal 95% level is selected, with an asymmetric loss that penalises overcoverage twice as heavily as undercoverage. Candidates with minimum effective sample size below 20 are excluded. The candidate grids are $\{0.005, 0.01, 0.02, 0.04, 0.08, 0.15\}$ for continuous Darcy and $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$ for EIT, and $\{0.001, 0.002, 0.003, 0.005, 0.007, 0.01\}$ for Burgers, reflecting the different observation scales. Table B.2 reports the selected values. For the pilot MCMC method, the selected σ_{data} can vary across seeds because each seed produces a different posterior geometry.

Table B.2: Selected observation noise scale σ_{data} per benchmark (seed 42 baseline). For problems using pilot MCMC selection, the value can differ across seeds.

Problem	Selection method	σ_{data}
Darcy continuous	Pilot MCMC	0.08
Darcy piecewise	MAP residual	0.012
EIT	Pilot MCMC	0.3
Burgers	Pilot MCMC	0.002

C

Laplace approximation baseline

The Laplace approximation constructs a Gaussian posterior centred at the MAP estimate, using the local curvature of the negative log-posterior as the precision matrix [24]. It provides a computationally cheap uncertainty baseline that assumes the posterior is approximately Gaussian.

The MAP estimate is obtained by running the deterministic inversion procedure in Algorithm 2 with the hyperparameters from Table F.1.

Given $\hat{\beta}_{1\mathrm{MAP}}$, the Hessian of the MAP inversion objective is computed via automatic differentiation, symmetrised, and regularised with λI ($\lambda = 10^{-4}$). If the regularised matrix is not positive definite, its eigenvalues are clipped to a small positive floor and the matrix is reconstructed. Posterior samples are drawn from $\mathcal{N}(\hat{\beta}_{1\mathrm{MAP}}, H^{-1})$ and clipped to $[-1, 1]$ to respect the tanh support. The sample count matches the NUTS baseline.

D

Additional experimental results

D.1. Darcy continuous

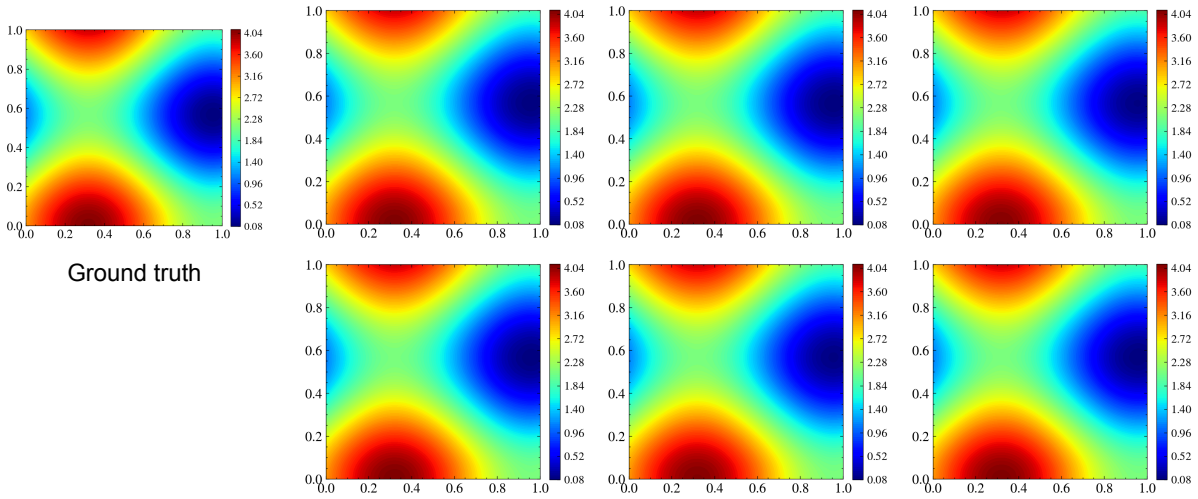


Figure D.1: Posterior sample gallery for the continuous Darcy baseline. The left panel shows the ground truth coefficient field. The six right panels show decoded coefficient fields $\tilde{a}^{(i)} = G_{\theta_a}(\beta_1^{(i)})$ from independent posterior samples.

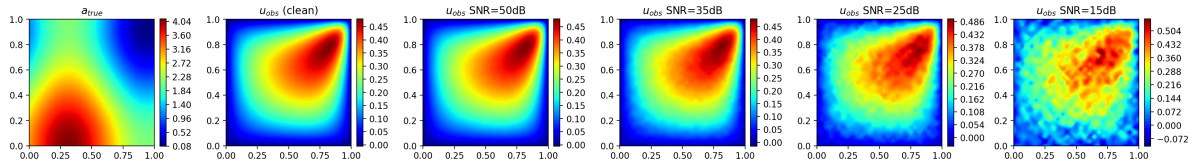
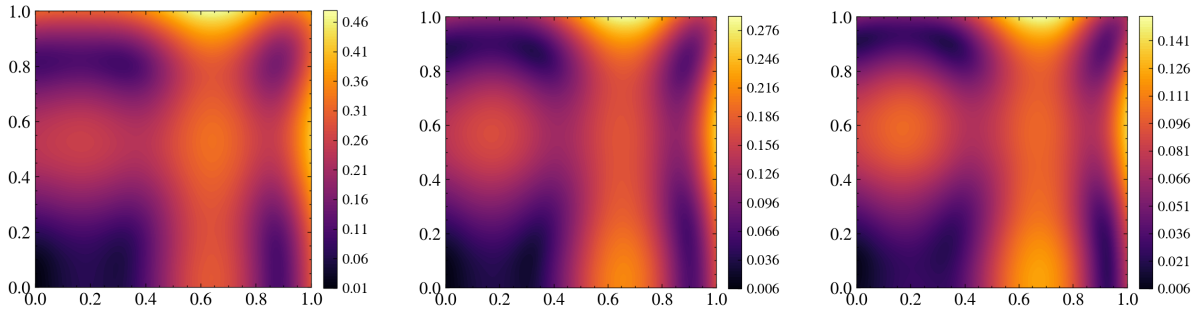


Figure D.2: Observation fields for the continuous Darcy noise sweep at each SNR level. Columns show the solution field u corrupted with increasing noise.



(a) 25 sensors **(b)** 50 sensors **(c)** 100 sensors

Figure D.3: Posterior standard deviation fields across sensor counts for the continuous Darcy sensor sweep.

D.2. Darcy piecewise

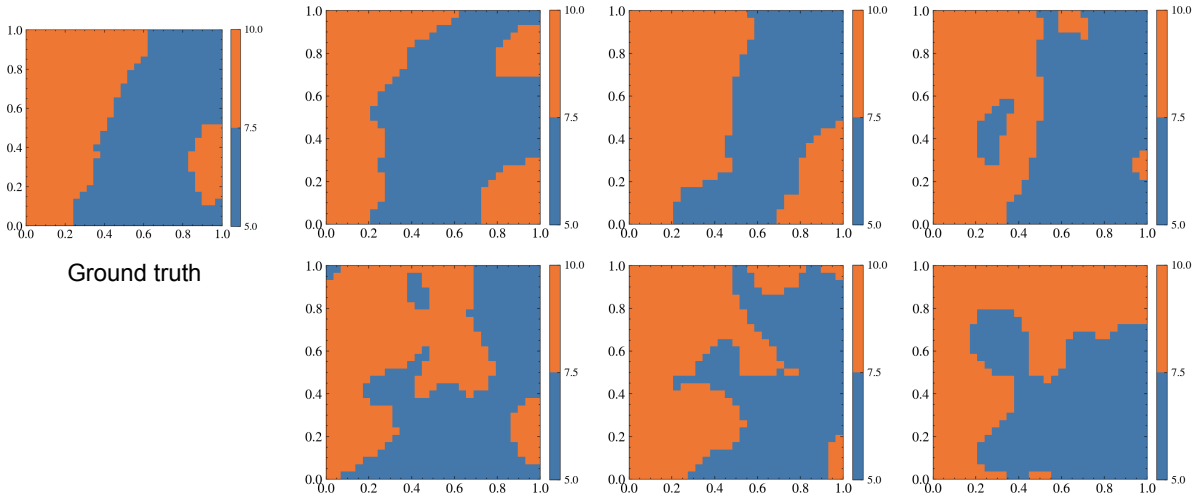


Figure D.4: Posterior sample gallery for the piecewise Darcy baseline. The left panel shows the ground truth coefficient field. The six right panels show decoded coefficient fields $\tilde{a}^{(i)} = G_{\theta_a}(\beta_1^{(i)})$ from independent posterior samples.

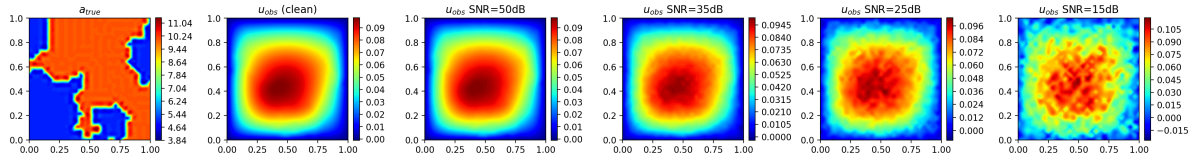
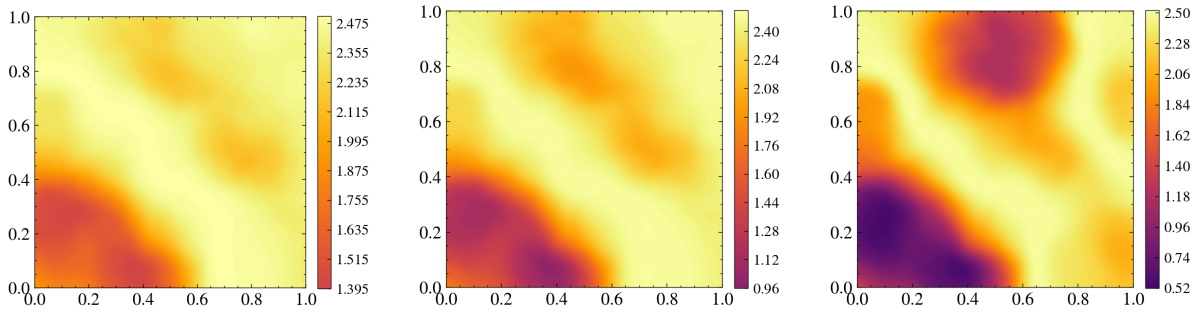


Figure D.5: Observation fields for the piecewise Darcy noise sweep at each SNR level. Columns show the solution field u corrupted with increasing noise.



(a) 25 sensors **(b)** 50 sensors **(c)** 100 sensors

Figure D.6: Posterior standard deviation fields across sensor counts for the piecewise Darcy sensor sweep.

D.3. EIT

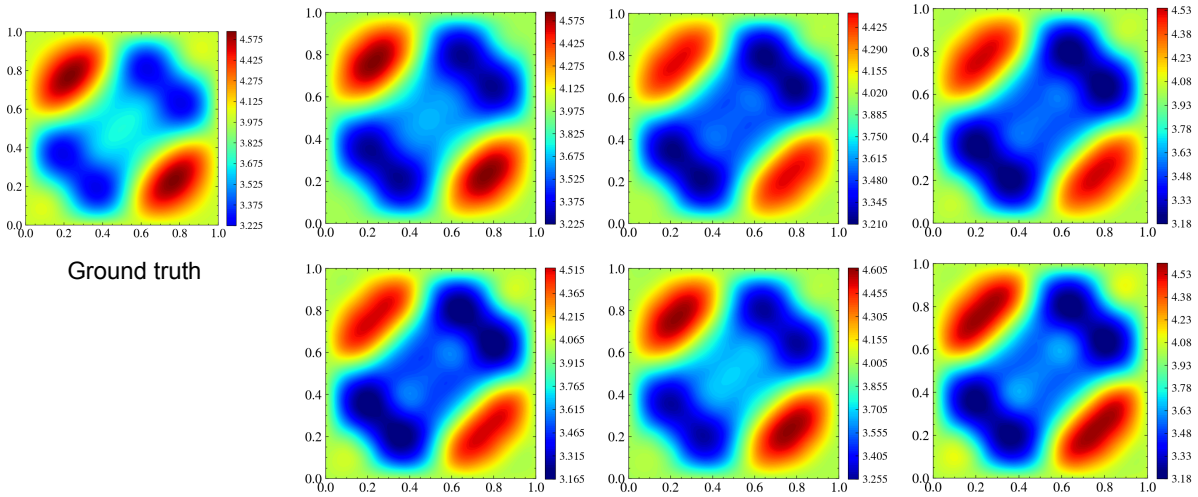
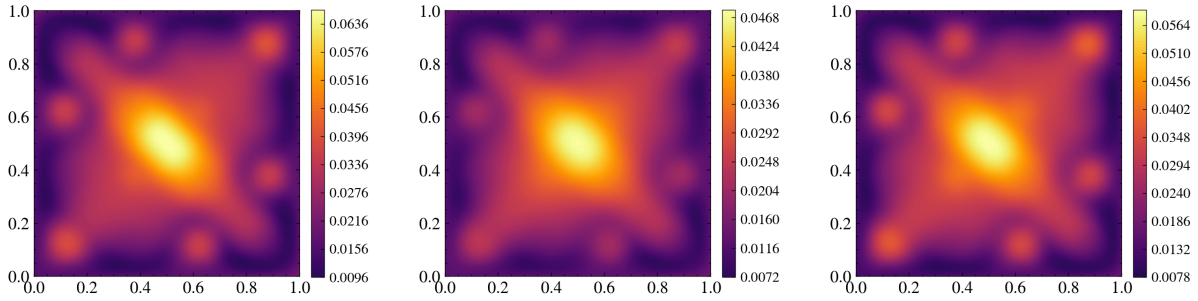


Figure D.7: Posterior sample gallery for the EIT baseline. The left panel shows the ground truth conductivity field. The six right panels show decoded conductivity fields $\tilde{a}^{(i)} = G_{\theta_a}(\beta_1^{(i)})$ from independent posterior samples.



(a) 31 sensors **(b)** 62 sensors **(c)** 124 sensors

Figure D.8: Posterior standard deviation fields across sensor counts for the EIT sensor sweep.

D.4. Burgers

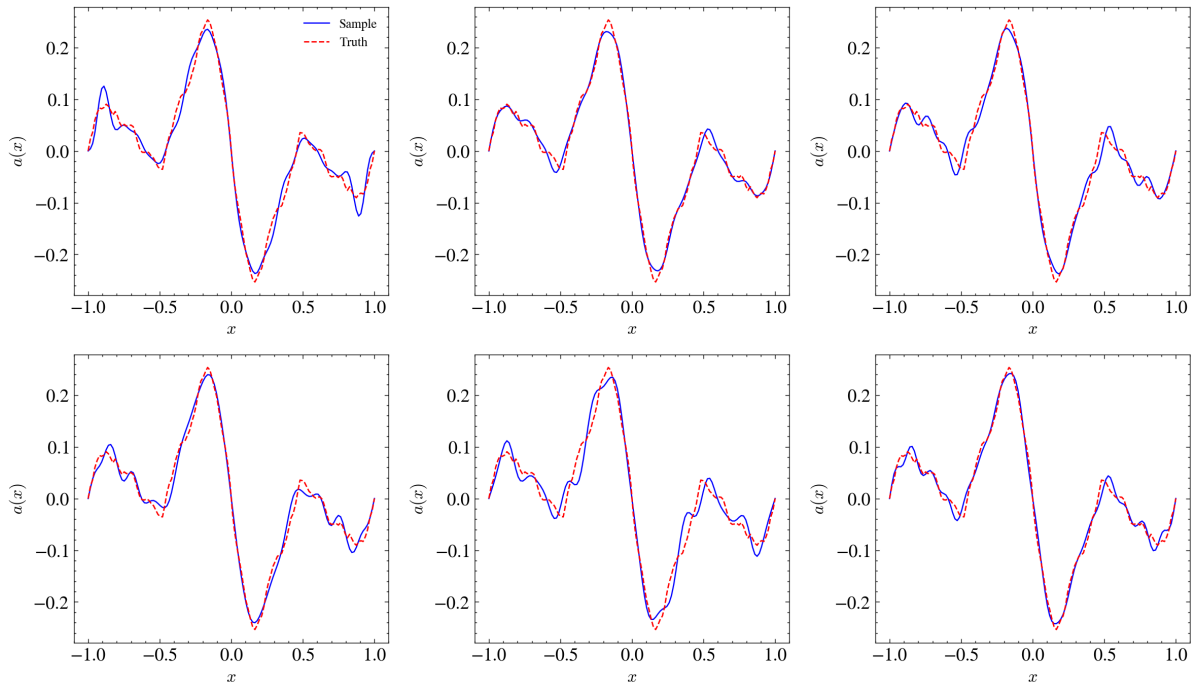


Figure D.9: Posterior sample gallery for the Burgers baseline. Each panel shows a sampled initial condition $\tilde{a}^{(i)}(x) = G_{\theta_u}(\beta_1^{(i)}, (x, 0)) \cdot m(x)$ overlaid on the true initial condition.

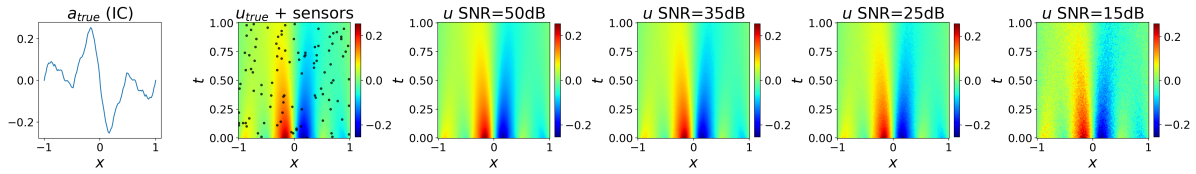
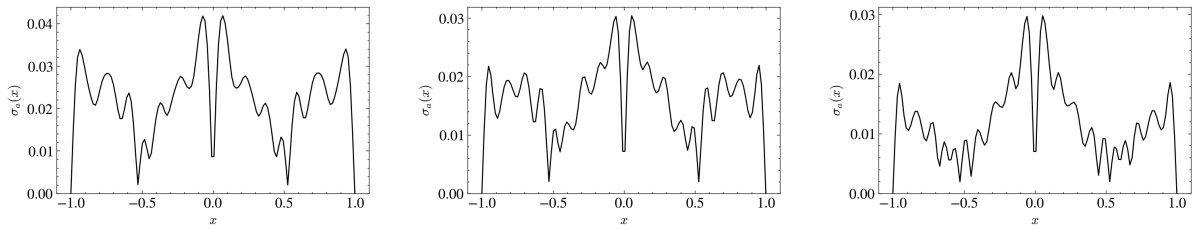


Figure D.10: Observation fields for the Burgers noise sweep at each SNR level. Panels show the solution $u(x, t)$ corrupted with increasing noise.



(a) 25 sensors **(b)** 50 sensors **(c)** 100 sensors

Figure D.11: Posterior standard deviation across sensor counts for the Burgers sensor sweep.

E

Normalising Flow Architecture Details

E.1. Base Distribution

The normalising flow $F_\phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_1}$ maps the learned latent distribution to a Beta(α, α) base distribution rescaled to $[-1, 1]^{d_1}$. The latent space β_1 is constrained to $[-1, 1]^{d_1}$ by the tanh activation in the coefficient encoder E_{θ_a}. Using a bounded base distribution provides better support matching than an unbounded distribution with infinite support, since the Beta density concentrates probability within the bounded region where valid latent representations exist, avoiding wasted probability mass outside the support and ensuring the flow learns the density structure within the physically meaningful region.

The symmetric Beta distribution on $[0, 1]$ has density

$$p_{\text{Beta}}(z; \alpha, \alpha) = \frac{z^{\alpha-1}(1-z)^{\alpha-1}}{B(\alpha, \alpha)},$$

where $B(\alpha, \alpha) = \frac{\Gamma(\alpha)^2}{\Gamma(2\alpha)}$ is the beta function. To rescale to $[-1, 1]$, we apply the affine transformation $z' = 2z - 1$, yielding the base distribution density on $[-1, 1]^{d_1}$. The multivariate base density is the product of marginals:

$$p_z(\boldsymbol{z}) = \prod_{j=1}^{d_1} \frac{1}{2}\, p_{\text{Beta}}\left(\frac{z_j + 1}{2};\, \alpha, \alpha\right).$$

The shape parameter α differs across problems. Continuous Darcy and EIT use $\alpha = 8$, piecewise-constant Darcy uses $\alpha = 5$, and Burgers uses $\alpha = 10$. All four benchmark problems use this Beta base distribution.

E.2. Neural Spline Flow Architecture

All four benchmark problems use neural spline flow coupling layers, which replace affine transformations with monotonic rational-quadratic splines [58]. This provides more expressive element-wise transformations while retaining tractable Jacobian computations.

In each coupling layer, the input $\boldsymbol{z} \in \mathbb{R}^d$ is split into two groups: $\boldsymbol{z} = (\boldsymbol{z}_A, \boldsymbol{z}_B)$. The layer applies two sequential transformations: first, each element of $\boldsymbol{z}_B$ is transformed by a monotonic rational-quadratic spline whose parameters are predicted by a conditioner network taking $\boldsymbol{z}_A$ as input; then $\boldsymbol{z}_A$ is transformed by a second spline whose parameters are predicted by a conditioner taking the updated $\boldsymbol{z}_B$ as input. This double-sided structure ensures all dimensions are transformed in every layer. Each spline is defined by $K = 5$ bins on a bounded interval $[-B, B]$, with $K + 1$ knot points and $K + 1$ derivative values at the knots. Monotonicity is enforced by requiring positive derivatives, implemented via a softplus transformation. Since each half-transformation has a triangular Jacobian, the log-determinant of the full layer is the sum of log-derivatives from both spline passes.

The number of coupling layers and conditioner hidden dimension are problem-dependent. Continuous Darcy uses 2 coupling layers with conditioner hidden dimension 32. Piecewise-constant Darcy uses

3 coupling layers with hidden dimension 128. EIT uses 3 coupling layers with hidden dimension 32. Burgers uses 3 coupling layers with hidden dimension 56. Each conditioner is a two-hidden-layer fully connected network with SiLU activations. The partition of dimensions alternates between coupling layers to ensure all dimensions are transformed.

F

Implementation Details

All experiments were run on NVIDIA A40 GPUs provided by DAIC [55]. The software stack uses JAX 0.4.35 with CUDA 12 for automatic differentiation and JIT compilation [57], NumPyro 0.19 for the NUTS sampler [59], Flax 0.8 for neural network modules, and Optax 0.2 for the Adam optimiser used in training and MAP estimation.

This appendix provides problem-specific architectural and optimisation details for the four benchmark inverse problems. The latent dimension d_1 differs across problems. Continuous Darcy and EIT use $d_1 = 6$, piecewise-constant Darcy uses $d_1 = 200$, and Burgers uses $d_1 = 16$.

Algorithm 2 summarises the deterministic inversion procedure. All network parameters are fixed after training; only the latent code β_1 is optimised. The learning rate follows a step decay schedule, reducing by a factor γ every S iterations. Table F.1 gives the per-problem hyperparameters. This procedure serves as both the MAP initialisation for NUTS sampling and the point estimate for the Laplace approximation.

Algorithm 2 Deterministic inversion via latent optimisation

Require: Pre-trained networks G_{θ_a}, G_{θ_u}, F_ϕ; observations u_{obs}; learning rate η_0; decay factor γ; step size S; total iterations T

Draw $N = 1000$ samples $\{\boldsymbol{z}^{(n)}\}_{n=1}^{N}$ from the base distribution and map through the inverse flow: $\beta_1^{(n)} = F_\phi^{-1}(\boldsymbol{z}^{(n)})$
Initialise $\beta_1 \leftarrow \frac{1}{N}\sum_{n=1}^{N} \beta_1^{(n)}$
Initialise Adam optimiser with learning rate η_0
for $t = 1, \ldots, T$ **do**
  Decode fields: $\tilde{a} = G_{\theta_a}(\beta_1)$, $\tilde{u}(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x})$
  Compute objective $\mathcal{F}(\beta_1) = \mathcal{F}_{\text{data}}(\beta_1) + \mathcal{F}_{\text{pde}}(\beta_1)$
  Update β_1 with Adam using $\nabla_{\beta_1}\mathcal{F}$
  if $t \bmod S = 0$ **then**
    $\eta \leftarrow \gamma \cdot \eta$
  end if
end for
return β_1^*, recovered coefficient $\tilde{a}^* = G_{\theta_a}(\beta_1^*)$, predicted solution $\tilde{u}^*(\boldsymbol{x}) = G_{\theta_u}(\boldsymbol{\beta}^*, \boldsymbol{x})$

Table F.1: Deterministic inversion hyperparameters for each benchmark. All problems use the Adam optimiser with step-decay learning rate scheduling. η_0: initial learning rate; S: decay step size; γ: decay factor; T: total iterations; M: number of observations.

Problem	η_0	S	γ	T	w_{data} / w_{pde}	M
Darcy continuous	0.01	125	0.8	1000	50 / 1	100
Darcy piecewise	0.1	200	0.1	1000	1 / 1	100
EIT	0.01	125	0.25	1000	100 / 1	124
Burgers	0.01	125	0.8	1000	50 / 1	100

F.1. Continuous Darcy flow

The coefficient encoder E_{θ_a} takes the permeability field evaluated on a 29×29 uniform grid as input. It consists of a convolutional neural network (CNN) with three hidden layers (64 output channels, kernel size 3×3, stride 2) followed by a fully connected network with hidden layers of 128 and 64 neurons. SiLU activations are used in all hidden layers, with a tanh activation on the output layer to constrain $\beta_1 \in [-1, 1]^{d_1}$. Since the boundary condition is fixed ($p = 0$ on $\partial\Omega$), the boundary encoder is omitted and $\boldsymbol{\beta} = \beta_1$.

Both the solution decoder G_{θ_u} and coefficient decoder G_{θ_a} use the MultiONet architecture with 6 hidden layers of 100 neurons each in both branch and trunk networks. The activation function is TanhSin$(x) = \tanh(\sin(\pi x + \pi)) + x$. A mollifier $f(\boldsymbol{x}) = \sin(\pi x_1)\sin(\pi x_2)$ is applied to the solution decoder output to enforce the homogeneous Dirichlet boundary condition automatically.

The weak-form PDE residual (Eq. A.1) uses N_c test functions on an $n_{\text{grid}} = 9$ quadrature grid per support ball.

IGNO training uses Adam with initial learning rate 5×10^{-4} and weight decay 10^{-4}, halved every 400 epochs, a batch size of 50, and a total of 2,000 epochs. The loss weights are $\lambda_{\text{pde}} = 1.0$ and $\lambda_{\text{data}} = 0.25$. The normalising flow is trained jointly with a loss weight of 0.0125. In addition to the weighted loss terms, training includes a top-k penalty that sums the largest squared residuals within each batch to improve robustness to outlier collocation points. The number of penalised residuals and relative weighting are problem-dependent.

F.2. Piecewise-constant Darcy flow

The coefficient encoder E_{θ_a} is a fully connected network. The 29×29 coefficient image is flattened into a vector and passed through two dense layers with 448 and 224 neurons respectively, with SiLU activations and a tanh output.

The solution decoder uses the MultiONet architecture with 5 hidden layers of 100 neurons each and TanhSin activation. The coefficient decoder outputs logits for per-pixel binary classification using the MultiONet architecture with 5 hidden layers of 256 neurons each. The trunk network uses the activation $\text{SiLU\_Sin}(x) = \text{SiLU}(\sin(\pi x + \pi)) + x$, and the branch network uses $\text{SiLU\_Id}(x) = \text{SiLU}(x) + x$. A sigmoid activation maps the logits to predicted probabilities $p_i \in [0, 1]$ for each grid point, and the reconstruction loss is the binary cross-entropy corresponding to the Bernoulli likelihood in Eq. 3.1. At inference time, a threshold of 0.5 is applied to obtain binary coefficient values $\tilde{a}(\boldsymbol{x}_i) \in \{k_{\text{low}}, k_{\text{high}}\}$.

Two additional stress decoder networks $G_{\theta_{s_1}}$ and $G_{\theta_{s_2}}$ predict the auxiliary stress variables used in the mixed weak-form formulation. Each stress decoder uses the MultiONet architecture with 5 hidden layers of 100 neurons each. The weak-form PDE residual uses $n_{\text{grid}} = 7$ quadrature points per support ball.

IGNO training uses Adam with initial learning rate 5×10^{-4} and weight decay 10^{-4}, halved every 200 epochs, a batch size of 25, and a total of 1,000 epochs. The loss weights are $\lambda_{\text{pde}} = 1.0$ and $\lambda_{\text{data}} = 1.0$. The normalising flow is trained separately for 2,000 epochs with Adam at learning rate 10^{-3}.

F.3. Electrical impedance tomography

The coefficient encoder E_{θ_a} takes the conductivity field on a 32×32 uniform grid. It consists of a CNN with four hidden layers (64 output channels, kernel size 3×3, stride 2) followed by a fully connected network with one hidden layer of 32 neurons, with SiLU activations and a tanh output.

The boundary encoder $E_{\theta_{bc}}$ uses one-hot encoding: for boundary condition g_l, the latent vector $\beta_2 = \boldsymbol{e}_l \in \mathbb{R}^{20}$, where $\boldsymbol{e}_l$ is the l-th standard basis vector. This enables IGNO to handle the $L = 20$ boundary voltage patterns efficiently.

Both the solution decoder and coefficient decoder use the MultiONet architecture with 5 hidden layers of 100 neurons each and TanhSin activation. A mollifier $f(\boldsymbol{x}) = \sin(\pi x_1)\sin(\pi x_2)$ enforces boundary conditions: $\tilde{u} = G_{\theta_u}(\boldsymbol{\beta}, \boldsymbol{x}) \cdot f(\boldsymbol{x}) + g_l(\boldsymbol{x})$.

The weak-form PDE residual (Eq. A.2) uses $n_{\text{grid}} = 7$ quadrature points per support ball, evaluated for each of the $L = 20$ boundary conditions.

IGNO training uses Adam with initial learning rate 5×10^{-4} and weight decay 10^{-4}, halved every 200 epochs, a batch size of 100, and a total of 2,000 epochs. The loss weights are $\lambda_{\text{pde}} = 1.0$ and $\lambda_{\text{data}} = 1.0$. The normalising flow is trained separately for 2,000 epochs with Adam at learning rate 10^{-3}.

F.4. Burgers equation

The coefficient encoder E_{θ_a} takes the initial condition field $a(x) = u(x, 0)$ evaluated on a uniform spatial grid as input. It is a fully connected network with layers $[n_{\text{mesh}}, 128, 64, 16]$, ELU activations in hidden layers, and a tanh output activation to constrain $\beta_1 \in [-1, 1]^{16}$. Since the boundary conditions are fixed ($u = 0$ at $x = \pm 1$), the boundary encoder is omitted and $\boldsymbol{\beta} = \beta_1$.

The solution decoder G_{θ_u} uses the MultiONet architecture with 6 hidden layers of 100 neurons each in both branch and trunk networks, with a TanhSin activation. It takes space-time coordinates $(x, t) \in [-1, 1] \times [0, 1]$ as input and the latent code β_1 as the branch input. The mollifier $m(x) = \sin(\pi x/2 + \pi/2)$ is applied to the decoder output to enforce zero boundary conditions at $x = \pm 1$. Unlike the Darcy and EIT problems, Burgers has no separate coefficient decoder; the initial condition is recovered as $a(x) = u(x, 0; \beta_1) = G_{\theta_u}(\beta_1, (x, 0)) \cdot m(x)$.

The normalising flow uses a neural spline flow with 3 coupling layers and conditioner hidden dimension 56, with a $\text{Beta}(10, 10)$ base distribution on $[-1, 1]^{16}$.

IGNO training uses Adam with initial learning rate 10^{-3} and weight decay 10^{-4}, decayed by a factor of $1/3$ every 1,000 epochs, a batch size of 50, and a total of 5,000 epochs. The loss weights are $\lambda_{\text{pde}} = 1.0$ and $\lambda_{\text{data}} = 10.0$. The normalising flow is trained separately for 300 epochs with Adam at learning rate 10^{-3}.

G

Wall-clock computational times

Table G.1: Wall-clock times (seconds) per test instance for each inference method and benchmark. MCMC columns report the time for MAP initialisation (shorter optimisation for NUTS starting point), NUTS warmup, and posterior sampling. Laplace columns report MAP estimation (Table F.1), Hessian computation, and sampling from the Gaussian approximation. Piecewise Darcy Laplace entries are absent because all instances had negative Hessian eigenvalues. Each cell reports the mean $\pm$ standard deviation across three seeds, where each seed's value is the average over three test instances. MCMC with physics is 2–4$\times$ slower than data-only MCMC, while Laplace is 30–100$\times$ faster than data-only MCMC overall.

	MCMC (data-only)			MCMC (with physics)			Laplace		
Problem	MAP	Warmup	Sampling	MAP	Warmup	Sampling	MAP	Hessian	Sampling
Darcy continuous	8.8 ± 0.064	400 ± 260	130 ± 72	8.2 ± 0.17	4400 ± 560	1700 ± 210	11 ± 0.048	4.4 ± 0.0020	0.67 ± 0.0068
Darcy piecewise	12 ± 1.2	3900 ± 260	1600 ± 310	4.3 ± 0.055	13000 ± 34	4400 ± 190	15 ± 1.4	n/a	
EIT	12 ± 0.89	140 ± 22	37 ± 2.8	10 ± 0.22	190 ± 3.7	40 ± 4.5	14 ± 0.43	5.6 ± 0.15	0.70 ± 0.015
Burgers	8.4 ± 0.55	440 ± 54	190 ± 29	7.2 ± 0.034	560 ± 71	350 ± 54	9.2 ± 0.032	3.3 ± 0.043	0.68 ± 0.011

H

Reproducibility statement

All source code for the methods and experiments presented in this thesis is publicly available at `https://github.com/1henrypage/b-igno`.

All randomness is managed through JAX's functional pseudorandom number generator [57], which carries no global mutable state. A single integer seed determines the full sequence of random operations in both training and posterior inference, and all experimental results are evaluated across three independent seeds.

TF32 arithmetic is disabled, 64-bit mode is enabled, and matrix multiplication precision is forced to full float32 to prevent numerical discrepancies in PDE gradient computations. Same-seed runs on identical hardware reproduce identical results.