

---

# A unified physics-informed generative operator framework for general inverse problems

---

Gang Bao<sup>a</sup> and Yaohua Zang<sup>b\*</sup>

<sup>a</sup>Zhejiang University, Center for Interdisciplinary Applied Mathematics, Yuhangtang Road 886, 310058 Hangzhou, China

<sup>b</sup>Technical University of Munich, Professorship of Data-driven Materials Modeling, School of Engineering and Design, Boltzmannstr. 15, 85748 Garching, Germany  
\*yaohua.zang@tum.de

## Abstract

Solving inverse problems governed by partial differential equations (PDEs) is central to science and engineering, yet remains challenging when measurements are sparse, noisy, or when the underlying coefficients are high-dimensional or discontinuous. Existing deep learning approaches either require extensive labeled datasets or are limited to specific measurement types, often leading to failure in such regimes and restricting their practical applicability. Here, a novel generative neural operator framework, IGNO, is introduced to overcome these limitations. IGNO unifies the solution of inverse problems from both point measurements and operator-valued data without labeled training pairs. This framework encodes high-dimensional, potentially discontinuous coefficient fields into a low-dimensional latent space, which drives neural operator decoders to reconstruct both coefficients and PDE solutions. Training relies purely on physics constraints through PDE residuals, while inversion proceeds via efficient gradient-based optimization in latent space, accelerated by an a priori normalizing flow model. Across a diverse set of challenging inverse problems, including recovery of discontinuous coefficients from solution-based measurements and the EIT problem with operator-based measurements, IGNO consistently achieves accurate, stable, and scalable inversion even under severe noise. It consistently outperforms the state-of-the-art method under varying noise levels and demonstrates strong generalization to out-of-distribution targets. These results establish IGNO as a unified and powerful framework for tackling challenging inverse problems across computational science domains.

**Keywords:** Inverse Problems, Deep Neural Operator, Deep Generative Modeling, PDEs, EIT Problem

## 1 Introduction

Inverse problems arise whenever unknown properties of a system need to be inferred from indirect or incomplete observations of its physical states. They are fundamental to many areas of science and engineering, from reconstructing subsurface properties in seismic imaging [1] and determining tissue conductivity in medical diagnostics [2, 3] to discovering microstructural properties in materials design [4]. These problems are governed by partial differential equations (PDEs) encoding physical laws relating unknown parameters to observable quantities [5, 6, 7, 8]. However, PDE inverse problems are notoriously challenging: they are typically high-dimensional, ill-posed, and sensitive

to noise. Even small measurement errors or model uncertainties can lead to large deviations in reconstructed parameters, making robust and efficient inversion a long-standing challenge.

Because analytical solutions are rarely attainable except in oversimplified cases, traditional approaches formulate inverse problems as constrained optimization tasks that iteratively update parameters while satisfying PDE constraints through forward and adjoint solves [9, 10]. While mathematically rigorous, these methods demand hundreds of expensive PDE simulations per inversion, making them impractical for large-scale or high-dimensional problems. Surrogate and reduced-order models [11] reduce cost but typically sacrifice accuracy beyond their calibration regimes. Bayesian inference frameworks [12] provide uncertainty quantification but face the curse of dimensionality, as sampling high-dimensional posteriors is computationally prohibitive.

Physics-informed machine learning (PIML) has recently introduced a paradigm shift by embedding governing physical laws directly into neural network training [13, 14, 15, 16, 17, 18, 19]. For inverse problems, PIML treats the unknown coefficients as trainable parameters while enforcing physical laws, enabling simultaneous recovery of hidden parameters and solutions [16, 20, 21, 22, 23, 24, 25]. However, existing PIML-based inversion methods exhibit notable limitations. They are primarily restricted to **solution-based inverse problems** with pointwise measurements and struggle fundamentally with **operator-based inverse problems**, such as electrical impedance tomography (EIT) [2], where measurements consist of input–output operators like Dirichlet-to-Neumann (DtN) maps rather than scalar values. Inverting such operators requires solving entire families of PDEs under multiple boundary conditions, dramatically increasing computational costs. Moreover, PIML training remains expensive, and the lack of structured representations for high-dimensional parameter spaces leads to highly non-convex optimization landscapes and poor convergence.

Deep neural operators (DNOs) [26, 27, 28, 29] have emerged as a promising alternative, capable of learning mappings between infinite-dimensional function spaces and predicting PDE solutions across entire families of inputs after a single training phase. DNO-based inversion typically follows two strategies: (i) using trained DNOs as fast forward surrogates within iterative optimization [30, 31], which still requires costly searches in high-dimensional coefficient spaces; or (ii) training inverse DNOs that directly map observations to coefficients [32, 33, 34], which demand massive labeled datasets, are limited to fixed sensor configurations, and exhibit poor robustness to noise and distribution shifts.

Recent advances in physics-informed DNOs (PI-DNOs) have sought to overcome these challenges by embedding PDE constraints into training, thereby reducing data dependence [35, 36, 37, 38, 39]. For example, PI-DeepONet [38] applies Bayesian MCMC for inversion but struggles in high-dimensional regimes and cannot handle operator-based measurements. PI-DIONs [40] trains via strong-form residuals but is restricted to fixed sensors and requires extensive solution-based measurements. DGenNO [39] introduces latent input compression for Darcy flow but remains confined to solution-based settings and exhibits sensitivity to initialization.

To address these limitations, this study proposes the **Inverse Generative Neural Operator (IGNO)**, a unified, physics-informed generative framework designed to efficiently and robustly solve general PDE inverse problems (Fig. 1). The key innovation of IGNO lies in transforming the intractable task of optimizing directly within high-dimensional, irregular coefficient spaces into efficient gradient-based optimization within a smooth, low-dimensional latent manifold. This is achieved by the generative architecture of IGNO, which consists of dual encoders that compress coefficient fields and boundary conditions into compact latent representations,  $\beta = (\beta_1, \beta_2)$ , and two MultiONet-based decoders [39] that respectively reconstruct coefficients from  $\beta_1$  and predict PDE solutions from  $\beta$ . Training of IGNO proceeds in a purely physics-informed manner by minimizing PDE residuals formulated in weak or strong form, requiring no labeled input–output pairs or precomputed forward simulations. This self-supervised paradigm enables efficient learning even with minimal data, while ensuring consistency with governing physical laws. During inversion, IGNO performs gradient-based optimization directly in the latent space to minimize discrepancies between predictions and observations while maintaining PDE consistency. To enhance convergence and robustness, a normalizing flow is also integrated into IGNO that maps the latent space to a standard Gaussian distribution, providing statistically informed initializations that accelerate convergence and stabilize optimization in noisy or ill-posed settings. A critical distinguishing capability of IGNO is its unified treatment of diverse measurement modalities within a single framework. Through its dual-encoder architecture, IGNO naturally accommodates both solution-based observations consisting

of sparse point measurements and operator-based measurements such as DtN maps in EIT. For operator-valued data, IGNO handles multiple boundary conditions simultaneously without repeatedly solving PDEs during training, enabling efficient inversion from operator-based measurements where existing PI-DNOs struggle to operate.

The framework is validated through extensive numerical experiments encompassing both continuous and discontinuous inverse targets. For solution-based problems, IGNO reconstructs smooth and piecewise-constant permeability fields in Darcy flow from sparse pressure measurements. For operator-based problems, it solves the EIT problem by recovering conductivity from DtN boundary operators. Across varying noise levels, out-of-distribution targets, and high-dimensional coefficient spaces, IGNO achieves reconstruction errors 3 to 6 times lower than the state-of-the-art method, with superior recovery of fine-scale features and strong extrapolation performance. Notably, IGNO maintains stable and accurate inversion in challenging scenarios where the baseline method produces substantial distortions or fails entirely, including recovery of discontinuous coefficients where traditional gradient-based methods cannot operate due to undefined gradients. These results establish IGNO as a scalable and broadly applicable framework for PDE inverse problems. By synergistically combining generative latent modeling, neural operator architectures, and physics-informed learning, IGNO effectively addresses key challenges in data efficiency, noise robustness, and generalization that have long hindered inverse problem solvers. Beyond the canonical problems demonstrated here, the methodology can naturally extend to inverse problems in geophysical exploration, materials characterization, and climate modeling, illustrating how the principled fusion of generative modeling, operator learning, and physics constraints can advance the frontier of computational science.

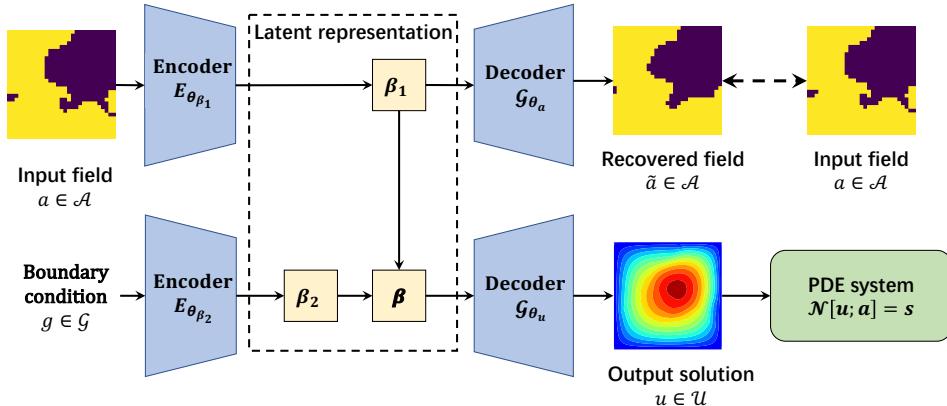


Figure 1: The IGNO framework for PDE inverse problems. The generative neural operator architecture encodes the coefficient field  $a$  and the boundary condition  $g$  into a structured latent variable  $\beta = (\beta_1, \beta_2)$  via encoders  $E_{\theta_{\beta_1}}$  and  $E_{\theta_{\beta_2}}$ , respectively. Two MultiONet-based decoders reconstruct the coefficient  $a$  from  $\beta_1$  (coefficient decoder  $G_{\theta_a}$ ) and predict the PDE solution  $u$  from  $\beta$  (solution decoder  $G_{\theta_u}$ ). The model is trained in a purely physics-informed manner, driven solely by governing PDE residuals without requiring paired input-output data. For inversion, gradient-based optimization in the low-dimensional latent space enables efficient recovery of unknown coefficients from sparse, noisy observations.

## 2 Results

### 2.1 Validation strategy and experimental setup

We validate IGNO across a suite of canonical PDE inverse problems that collectively test the framework’s ability to handle diverse coefficient structures, measurement modalities, and noise levels. The tested problems include the recovery of (i) continuous coefficients and (ii) piecewise-constant coefficients in Darcy flow, and (iii) the reconstruction of electrical conductivity in the EIT problem. These examples progressively increase in difficulty from low-noise, in-distribution inference to highly noisy, out-of-distribution recovery from operator-valued data, allowing systematic eval-

ation of IGNO’s accuracy, robustness, and generalization. In all experiments, IGNO is compared with the state-of-the-art Physics-Informed Deep Inverse Operator Networks (PI-DIONs) [40] under consistent training and optimization setups, except in the EIT case. Details of the PI-DIONs implementation are provided in Note 4 of the Supplementary Information (SI). We note that PI-DIONs cannot handle operator-based measurements, thus limiting comparisons to solution-based cases. For each problem, we evaluate performance under three noise levels: low ( $\text{SNR} = 50 \text{ dB}$ ), medium ( $\text{SNR} = 25 \text{ dB}$ ), and high ( $\text{SNR} = 15 \text{ dB}$ ), where  $\text{SNR}$  denotes the signal-to-noise ratio. Quantitative evaluations use the **root-mean-square error (RMSE)** for continuous targets and the **cross-correlation indicator** ( $I_{corr}$ ) for piecewise-constant targets, complemented by visual assessment of fine-scale reconstructions.

Training data consist of 1,000 coefficient samples for the Darcy flow problem with continuous permeabilities and 10,000 samples with piecewise-constant permeabilities. For the EIT problem, 1,000 conductivity fields and 20 boundary conditions are used, leading to a total of  $1000 \times 20$  samples. Crucially, training of IGNO requires only coefficient fields and boundary conditions, without any precomputed PDE solutions, whereas the PI-DIONs method relies on finite element method (FEM) solutions for all training samples. Finally, the normalizing flow model used for latent-space initialization is trained on the learned latent representations after the main model training. Architectural and optimization details are summarized in the **Methods** or **Supplementary Methods** in SI.

## 2.2 Continuous coefficient recovery from solution-based measurements

The first benchmark involves recovering a smooth permeability field in the Darcy flow equation from sparse, noisy pressure measurements. This problem serves as a controlled setting for evaluating IGNO’s data efficiency and noise robustness. The governing PDE is a second-order elliptic equation on the unit square domain:

$$\begin{aligned} -\nabla(k(x, y)\nabla p(x, y)) &= f(x, y), \quad \text{in } \Omega = [0, 1]^2, \\ p(x, y) &= 0, \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where  $k$  denotes the unknown permeability,  $p$  the pressure, and  $f = 10$  the source term. The permeability fields follow  $k(x, y) = 2.1 + \sin(\omega_1 x) + \cos(\omega_2 y)$ , with  $(\omega_1, \omega_2)$  sampled independently from the uniform distribution  $U(0, 7\pi/4)^2$ . Measurements consist of pressure values at  $m = 100$  randomly placed interior sensors, shown as black dots in the right of Fig. 2(a), and are corrupted by additive Gaussian noise at the specified SNR levels.

Fig.2(b)–(c) presents the reconstructed permeability fields and corresponding pointwise absolute errors for an in-distribution test case under three noise levels. Comparative performance results of both methods across different noise levels are presented in Supplementary Table 1 of the SI. As shown in Fig.2(b), IGNO accurately recovers both the global patterns and fine-scale variations of the permeability field, even under severe noise ( $\text{SNR} = 15 \text{ dB}$ ), yielding smooth reconstructions with errors primarily localized in regions of rapid spatial variation. Reconstruction quality degrades only slightly as noise increases, with RMSEs of 0.56%, 0.61%, and 1.15% for SNR values of 50, 25, and 15 dB, respectively. In contrast, PI-DIONs exhibit substantially higher errors across all noise levels, with RMSEs of 3.66%, 3.80%, and 4.15% for SNR values of 50, 25, and 15 dB, respectively. As shown in Fig. 2(c), the PI-DIONs method fails to resolve fine-scale structures and produces noticeable distortions, particularly in regions of low permeability. Pointwise absolute error maps further confirm that IGNO maintains uniformly small errors across the domain, whereas PI-DIONs exhibit larger, spatially correlated error patterns.

The superior performance of IGNO can be attributed to its latent-space optimization strategy. Instead of directly optimizing over the high-dimensional coefficient space, IGNO performs inversion within a compact, well-structured latent manifold where gradients are smooth and correlated with meaningful physical variations. This transformation substantially alleviates the ill-posedness inherent to direct inversion. The embedded PDE residual constraints further enhance stability by enforcing physical consistency between the reconstructed permeability and the corresponding solution. In addition, the a priori normalizing flow provides statistically informed initialization, which reduces the likelihood of convergence to poor local minima and improves overall optimization reliability. In contrast, the PI-DIONs method predicts coefficients directly from noisy observations in the original, unstructured parameter space. This direct mapping is highly sensitive to measurement noise and lacks an explicit mechanism to enforce physical consistency, often leading to reconstructions that deviate significantly from true fields.

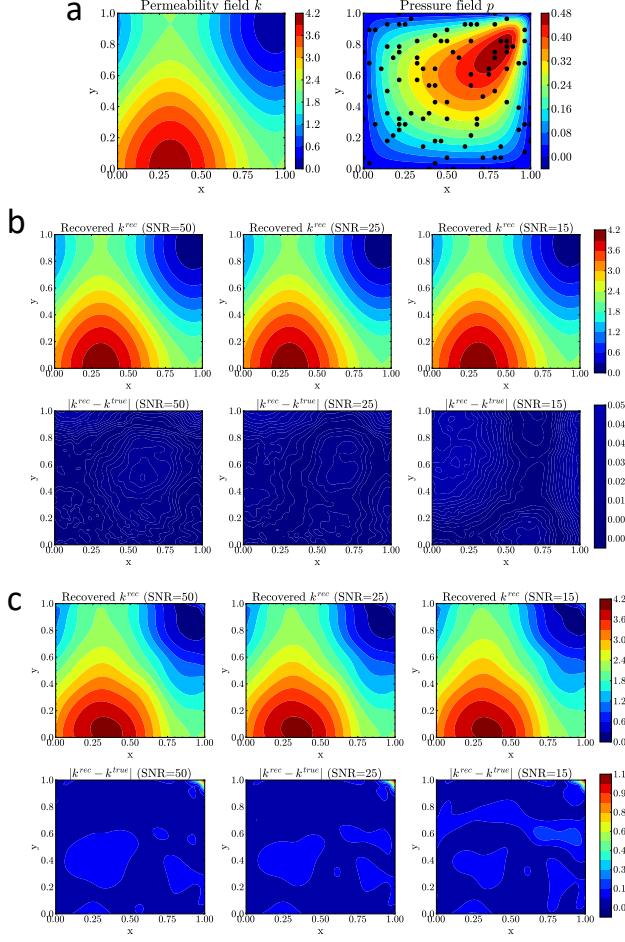


Figure 2: Recovery of **continuous coefficients** with solution-based measurements (**In-distribution case**): (a) the ground truth permeability field  $k$  and the corresponding pressure field  $p$  (black dots denote  $m = 100$  random sensors); (b) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by **IGNO** under different noise levels; (c) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by **PI-DIONs** under different noise levels.

### 2.3 Discontinuous coefficient recovery from solution-based measurements

The second benchmark tackles a more challenging scenario in which the coefficient field is piecewise-constant, representing heterogeneous media with sharp interfaces between two permeability phases ( $k = 10$  in phase 1 and  $k = 5$  in phase 2), as shown in Fig. 3(a). This configuration closely mimics realistic geological or material systems where discontinuities naturally occur. This problem is particularly challenging because the gradients of  $k$  are undefined at discontinuities, rendering traditional gradient-based optimization infeasible without special treatment. To overcome this challenge, IGNO employs a probabilistic phase decoder that estimates, for each spatial location, the likelihood of belonging to a given phase (phase 1). The decoder is trained using a cross-entropy loss instead of the mean-squared error (MSE) loss, as detailed in the SI. This probabilistic formulation prevents optimization breakdown at discontinuities and promotes naturally regularized reconstructions. During inference, a threshold of 0.5 is applied to the predicted probability map to obtain the final binary phase segmentation. For training, piecewise-constant permeability fields are generated from a cutoff Gaussian process  $\mathcal{GP}(0, (-\Delta + 9I)^{-2})$  [27, 39], and only  $m = 100$  sparse pressure measurements are used for inversion, shown as black dots in the right of Fig. 3(a).

Fig. 3(b)–(c) presents the reconstructed permeability fields and pointwise absolute error maps obtained by both methods under varying noise levels for an in-distribution test case. The performance of both methods under different noise levels is reported in Supplementary Table 3 of the SI. Again,

IGNO accurately reconstructs both the global phase topology and the sharp interfaces between phases across all noise levels, achieving cross-correlation indicators  $I_{corr}$  of 0.969, 0.960, and 0.951 for SNR values of 50, 25, and 15 dB, respectively. As shown in Fig. 3(b), IGNO maintains morphologically faithful reconstructions even when measurements are contaminated with severe 15 dB noise, with pointwise absolute errors primarily confined to phase boundaries where discontinuities pose inherent challenges. In contrast, the PI-DIONs method fails to recover the correct structures, yielding  $I_{corr}$  values of only 0.718, 0.715, and 0.726, indicative of largely random reconstructions with no resemblance to the ground truth. As seen in Fig. 3(c), the PI-DIONs method produces spatially incoherent patterns with widespread errors across the domain rather than localized near interfaces. This failure stems from two intrinsic limitations of PI-DIONs: (1) the reliance on strong-form PDE residuals, which require differentiability of  $k$  and thus break down at discontinuities; and (2) direct recovery of  $k$  in a high-dimensional, irregular function space, which makes the inversion process extremely challenging. IGNO overcomes these challenges through its generative latent-space formulation. By embedding discontinuous coefficient fields into a smooth latent manifold, IGNO converts a fundamentally discrete inversion task into a differentiable, well-conditioned optimization problem. The trained coefficient decoder then maps the optimized latent representation back to the discontinuous field, effectively reconstructing sharp interfaces without requiring explicit gradient information. Furthermore, the normalizing flow-based initialization accelerates convergence and enhances reconstruction fidelity by providing latent variables corresponding to statistically plausible phase configurations.

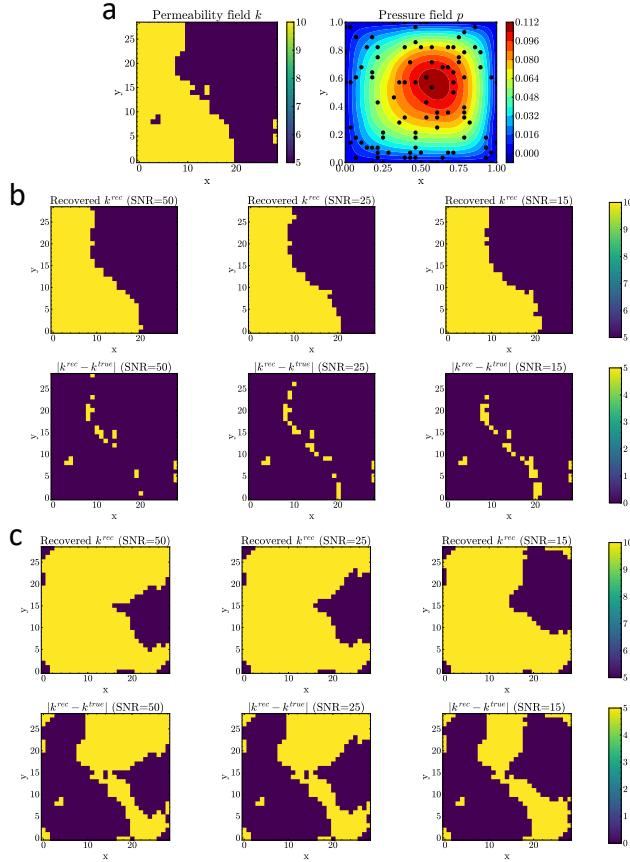


Figure 3: Recovery of **piecewise-constant coefficients** with solution-based measurements (**In-distribution case**): (a) the true permeability field  $k$  (Phase 1 shown in yellow) and the corresponding pressure field  $p$  (black dots denote  $m = 100$  random sensors); (b) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by IGNO under different noise levels; (c) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by PI-DIONs under different noise levels.

## 2.4 Recovery of conductivity from operator-based measurements: the EIT problem

The EIT problem represents a fundamentally different class of inverse problems, where measurements correspond not to pointwise solution values but to an operator that maps boundary inputs to boundary responses. Specifically, the goal is to recover the spatially varying conductivity field  $\gamma > 0$  in the elliptic PDE:

$$\begin{aligned} -\nabla \cdot (\gamma(x, y) \nabla u(x, y)) &= 0, \quad \text{in } \Omega = [0, 1]^2, \\ u(x, y) &= g(x, y), \quad \text{on } \partial\Omega, \end{aligned} \tag{2}$$

from the DtN operator  $\Lambda_\gamma[g] : g \mapsto \gamma \frac{\partial u}{\partial \vec{n}}|_{\partial\Omega}$ , which maps imposed boundary voltages  $g$  to induced boundary currents  $\gamma \frac{\partial u}{\partial \vec{n}}|_{\partial\Omega}$ , where  $\vec{n}$  denotes the unit outward normal. In this study, the DtN operator is approximated using  $L = 20$  distinct boundary voltage patterns defined by  $g_l = \cos(2\pi(x \cos(\theta_l) + y \sin(\theta_l)))$ , where  $\theta_l = 2\pi l / 20$ . Operator-based measurements are collected from  $m = 128$  uniformly distributed sensors along the four boundaries. This formulation is substantially more challenging than solution-based inversion, since each measurement corresponds to an entire PDE operator rather than discrete solution values at sensor locations. IGNO effectively addresses this complexity through its dual-encoder architecture. The coefficient encoder  $E_{\theta_{\beta_1}}$  compresses the high-dimensional conductivity field into a compact latent variable  $\beta_1$ , while the boundary encoder  $E_{\theta_{\beta_2}}$  represents each boundary condition as a one-hot latent vector  $\beta_2$ . Together, these latent representations enable the solution decoder to efficiently predict the full operator response within a unified generative framework. Training is conducted in a purely physics-informed manner by minimizing the residuals of the governing elliptic PDE, eliminating the need for any labeled conductivity-operator pairs. Further implementation details are provided in the Supplementary Methods of the SI, while the performance of IGNO in this problem is reported in Supplementary Table 5.

Fig. 4(b) shows the recovered conductivity fields and pointwise absolute errors obtained by IGNO under different noise levels for an in-distribution test case, where the conductivity follows trigonometric patterns representative of the training distribution (left panel of Fig. 4(a)). IGNO achieves RMSEs of 0.44%, 1.28%, and 2.77% for SNR values of 50, 25, and 15 dB, respectively, accurately capturing both smooth variations and fine-scale features. The reconstruction quality remains high even under severe noise, with errors distributed relatively uniformly across the domain. The framework naturally accommodates all  $L = 20$  boundary conditions via its dual-encoder structure, efficiently incorporating operator information without requiring repeated PDE solves during training. To evaluate generalization beyond the training distribution, we tested an out-of-distribution case in which the conductivity is drawn from a shifted parameter distribution, producing markedly different spatial patterns (right panel of Fig. 4(a)). Despite the substantial distribution shift, IGNO maintains robust performance, with RMSEs of 2.19%, 2.46%, and 2.93% for SNR values of 50, 25, and 15 dB, respectively. While errors are moderately higher than in the in-distribution case, Fig. 4(c) indicates that the framework successfully reconstructs the overall conductivity structure and fine-scale variations. The modest degradation in performance compared to the in-distribution case highlights IGNO’s strong extrapolation capabilities.

These results underscore a key advantage of IGNO: the ability to perform operator-based inversion without retraining or repeated PDE solves for each boundary condition. Once trained, the model can accommodate any combination of boundary inputs by adjusting the boundary latent representation, enabling efficient inversion from operator-valued measurements that would be intractable for existing physics-informed DNOs or PIML-based methods. The dual-encoder design encodes both coefficient and boundary condition information into structured latent representations, providing a unified treatment of diverse measurement modalities. Furthermore, IGNO can handle changes in sensor configurations without retraining, in contrast to data-driven DNOs that require extensive labeled pairs and retraining for new sensor arrangements. These capabilities demonstrate IGNO’s practicality for realistic applications where measurements are noisy, multimodal, and acquired from varying sensor setups, such as in medical imaging and non-destructive testing.

## 2.5 Generalization to out-of-distribution targets

Beyond the out-of-distribution EIT results presented above, we systematically assessed IGNO’s ability to extrapolate to coefficient distributions that differ substantially from the training data. For continuous Darcy flow problems, we tested on permeability fields with frequency parameters drawn from entirely non-overlapping ranges compared to training, i.e.,  $(\omega_1, \omega_2) \sim U(7\pi/4, 2\pi)$ . For

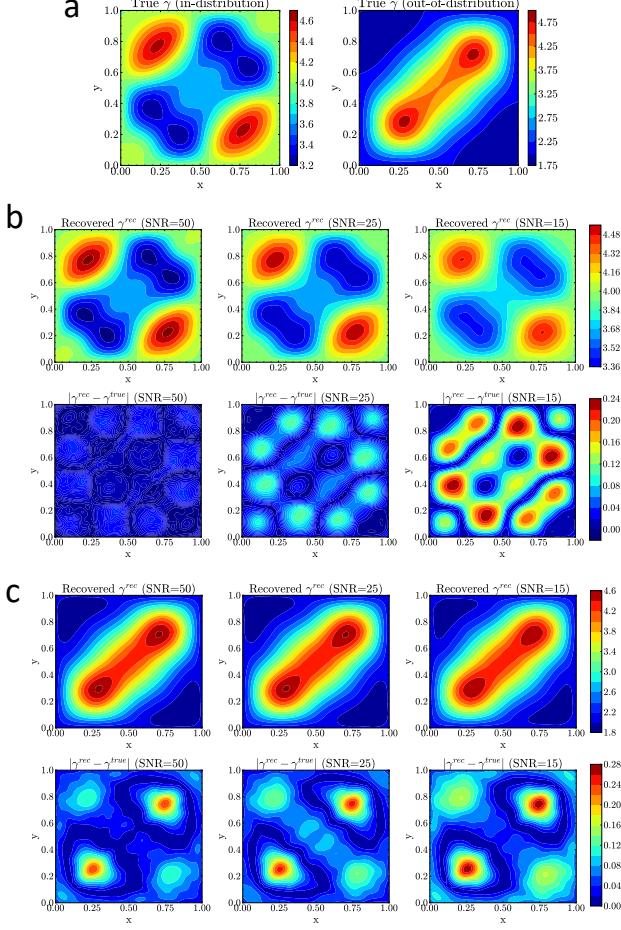


Figure 4: Performance of IGNO in solving the EIT problem: (a) the truth conductivity  $\gamma$  in In-distribution (left) and Out-of-distribution (right) cases; (b) recovered conductivity  $\gamma^{rec}$  and corresponding pointwise absolute errors under different noise levels in In-distribution case; (c) recovered conductivity  $\gamma^{rec}$  and corresponding pointwise absolute errors under different noise levels in Out-of-distribution case.

piecewise-constant problems, a different cutoff Gaussian process,  $\mathcal{GP}(0, (-\Delta + 16I)^{-2})$ , was used to generate more complex phase geometries. The performance of both methods in recovering continuous and discontinuous targets under the out-of-distribution case is reported in Supplementary Tables 2 and 4 of the SI, respectively.

For continuous coefficients, IGNO maintained RMSEs of 0.64%, 1.13%, and 1.94% across noise levels for the out-of-distribution target. This represents only a modest degradation from in-distribution performance and remains 3–5 times more accurate than the PI-DIONs method, which yields RMSEs of 4.76%, 4.84%, and 6.39% for SNR = 50, 25, and 15 dB. As shown in Fig. 5(b), IGNO successfully reconstructs both large-scale and fine-scale structures, with pointwise errors remaining low even under high noise levels. By contrast, Fig. 5(c) shows that the PI-DIONs method produces substantial distortions and higher error concentrations, particularly in regions of rapidly varying permeability. Moreover, the RMSE of PI-DIONs increases sharply with noise, highlighting its limited robustness to measurement errors. In contrast, IGNO maintains consistently low errors across all noise levels, demonstrating both robustness and strong generalization to out-of-distribution targets.

For piecewise-constant coefficients, IGNO achieves  $I_{corr}$  values of 0.960, 0.949, and 0.914 across SNR = 50, 25, and 15 dB, respectively, indicating strong morphological agreement despite the distribution shift. As shown in Fig. 6(b), the recovered permeability fields  $k^{rec}$  closely resemble the

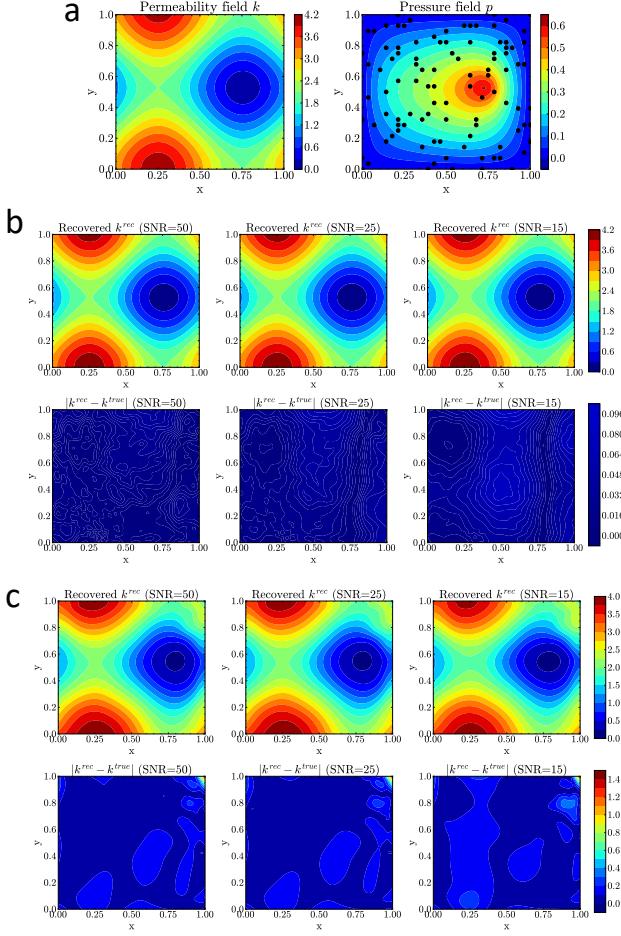


Figure 5: Recovery of **continuous coefficients** with solution-based measurements (**Out-of-distribution case**): (a) the ground truth permeability field  $k$  and the corresponding pressure field  $p$  (black dots denote  $m = 100$  random sensors); (b) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by IGNO under different noise levels; (c) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by PI-DIONs under different noise levels.

true fields, with pointwise errors concentrated primarily along phase boundaries. By contrast, Fig. 6(c) shows that the PI-DIONs method fails consistently across all noise levels, producing  $I_{corr}$  values around 0.78 and recovering a single phase almost everywhere, resulting in completely incorrect predictions. These results demonstrate that the latent space learned by IGNO captures fundamental physical structures rather than merely memorizing training data, enabling robust generalization to unseen coefficient patterns.

### 3 Discussion

The results presented above demonstrate that IGNO provides a robust, accurate, and computationally efficient framework for solving diverse PDE inverse problems. Across all benchmarks, including continuous and piecewise-constant coefficients in Darcy flow and operator-based EIT inversion, IGNO consistently outperforms the state-of-the-art method, achieving reconstruction errors 3 to 6 times lower while maintaining fidelity to fine-scale structures and sharp interfaces. These advances establish IGNO as a broadly applicable methodology that effectively addresses long-standing challenges in inverse problem solving, such as data efficiency, noise robustness, and generalization to unseen coefficient distributions.

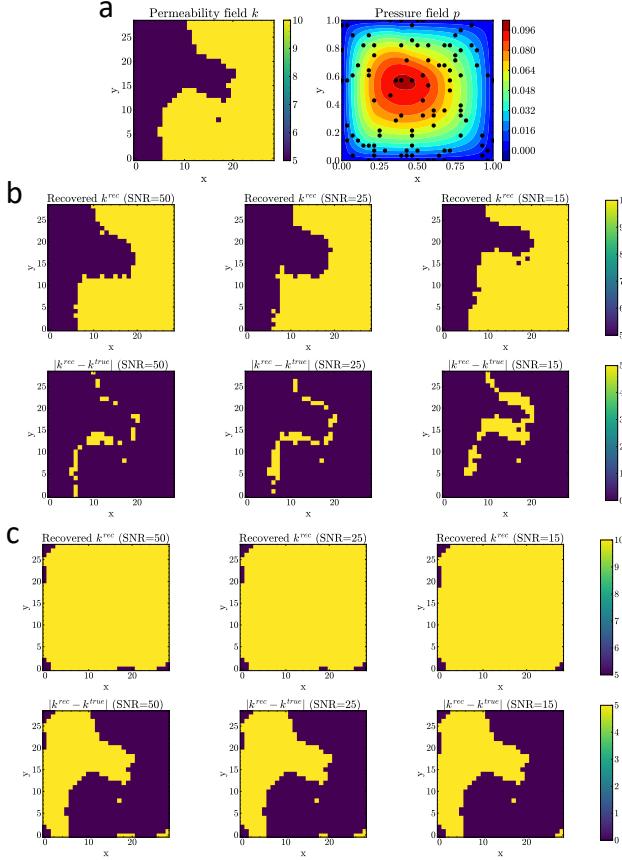


Figure 6: Recovery of **piecewise-constant coefficients** with solution-based measurements (**Out-of-distribution case**): (a) the true permeability field  $k$  (Phase 1 shown in yellow) and the corresponding pressure field  $p$  (black dots denote  $m = 100$  random sensors); (b) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by IGNO under different noise levels; (c) recovered permeability  $k^{rec}$  and corresponding pointwise absolute errors obtained by PI-DIONs under different noise levels.

The superior performance of IGNO stems from its **latent-space generative architecture**, which transforms high-dimensional, irregular coefficient recovery into smooth, low-dimensional optimization problems. By embedding both coefficients and boundary conditions into structured latent representations, IGNO produces well-behaved gradients during inversion that align with physically meaningful variations, substantially alleviating the ill-posedness that plagues direct inversion methods. This advantage is particularly evident in discontinuous coefficient recovery, where IGNO accurately reconstructs sharp phase interfaces, achieving cross-correlations above 0.95 even under severe noise, whereas the PI-DIONs method fails due to undefined gradients at discontinuities. The incorporation of PDE residual constraints further stabilizes inversion by enforcing consistency with governing laws, while the normalizing flow provides statistically informed initialization that accelerates convergence and mitigates the risk of poor local minima. This combination of latent modeling, physics-informed constraints, and probabilistic initialization enables IGNO to maintain robust performance even under high measurement noise or out-of-distribution conditions.

The **physics-informed training paradigm** underpins IGNO’s exceptional data efficiency by eliminating reliance on costly labeled datasets. IGNO requires only coefficient fields and boundary conditions for training, with PDE residuals, formulated in weak or strong form, serving as supervisory signals. This self-supervised approach stands in sharp contrast to purely data-driven DNOs, which typically demand tens of thousands of paired input–output samples generated via expensive forward simulations or physical experiments. Such data efficiency is particularly advantageous in domains like seismic imaging, where acquiring large labeled datasets can be ethically or practically challeng-

ing, or in engineering applications where high-fidelity simulations are computationally prohibitive. Moreover, the incorporation of PDE residual constraints not only reduces data requirements but also enforces physical consistency between reconstructed coefficients and corresponding solutions, further enhancing the stability and reliability of the inversion process.

Another key advantage of IGNO is its **unified treatment of diverse measurement modalities** within a single framework. Unlike existing physics-informed DNOs and PIML-based methods, which are restricted to solution-based observations at fixed sensor locations, IGNO naturally accommodates both solution-based measurements and operator-valued data, such as DtN maps in EIT. Its dual-encoder design separately compresses boundary conditions and coefficients into latent variables, enabling efficient inversion from operator data without repeatedly solving families of PDEs during training or inference. This flexibility not only reduces computational costs but also extends applicability to complex experimental settings where measurements are multimodal, sparse, or noisy. For the EIT problem, IGNO successfully reconstructs conductivity from 20 distinct boundary voltage patterns, achieving RMSEs below 3% even under high noise, while maintaining robust performance for out-of-distribution targets with spatial patterns markedly different from the training data. This operator-based inversion capability represents a fundamental advance, as no existing physics-informed DNO can handle such measurements.

The results further highlight IGNO’s strong generalization capabilities. For out-of-distribution targets with frequencies or phase geometries not encountered during training, IGNO maintains low reconstruction errors and accurately captures essential physical structures, including phase topology in discontinuous media and conductivity variations in EIT. This demonstrates that IGNO learns a physically meaningful latent space that encodes fundamental structural properties of the coefficient fields, rather than merely memorizing the training data. In contrast, the state-of-the-art method exhibits obvious performance degradation under the same conditions, underscoring the limitations of direct inversion in high-dimensional, unstructured coefficient spaces. In addition, IGNO demonstrates significant computational efficiency, enhancing its practical utility. Once trained, a single model can handle multiple inverse problems with varying observations, sensor configurations, or boundary inputs without retraining. Gradient-based optimization in the low-dimensional latent space converges within 200–500 steps, with each iteration requiring only forward passes through the decoders rather than costly PDE solves. For the largest problem tested, i.e., the EIT problem, individual inversions complete in under 1 minute, representing a substantial reduction in computational cost compared to traditional PDE-constrained optimization, which requires hundreds of forward and adjoint PDE solves per inversion.

Despite these advantages, several limitations and opportunities for future development merit consideration. First, IGNO requires a representative training set of coefficient fields to learn the latent space effectively, and its performance may degrade for coefficient distributions that are extremely far from the training manifold. Similarly, highly anisotropic or multiscale problems with features exceeding the capacity of the chosen latent dimension may pose challenges for accurate reconstruction. Future work could investigate adaptive latent dimensions and hierarchical latent representations to further enhance robustness in highly heterogeneous settings. Second, the current framework focuses on deterministic inversion, producing point estimates of unknown coefficients. Extending IGNO to probabilistic inversion that quantifies uncertainty in recovered parameters would provide valuable information for decision-making in high-stakes applications, such as medical diagnostics or infrastructure assessment. The existing latent space structure and normalizing flow components provide a natural foundation for such extensions via variational inference or sampling-based approaches.

Overall, IGNO establishes a scalable, versatile, and unified framework for PDE inverse problems, bridging physics-informed machine learning, neural operator learning, and generative modeling. By leveraging structured latent representations, enforcing PDE constraints, and performing efficient latent-space optimization, IGNO effectively addresses key challenges in data efficiency, noise robustness, and generalization. Beyond the canonical benchmarks presented here, this methodology is broadly applicable to real-world scientific and engineering problems, including geophysical exploration, medical imaging, nondestructive testing, and materials characterization, where measurements are often noisy, multimodal, and sparsely sampled. The combination of accuracy, efficiency, and flexibility positions IGNO as a powerful tool for advancing computational inverse problem solving in high-dimensional, real-world scenarios.

## 4 Methods

### 4.1 Problem formulation

We consider PDE inverse problems governed by the general PDE system:

$$\begin{aligned}\mathcal{N}(u, a) &= s, \quad \text{in } \Omega \\ \mathcal{B}(u) &= g, \quad \text{on } \partial\Omega\end{aligned}\tag{3}$$

where  $u \in \mathcal{U}(\Omega)$  denotes the PDE solution,  $a \in \mathcal{A}(\Omega)$  is the unknown coefficient function,  $s \in \mathcal{S}(\Omega)$  is the source term, and  $g \in \mathcal{G}(\partial\Omega)$  represents boundary conditions on domain  $\Omega \subset \mathbb{R}^d$  with the Lipschitz boundary  $\partial\Omega$ . The operators  $\mathcal{N}$  and  $\mathcal{B}$  denote the differential and boundary operators, respectively, and  $\mathcal{U}, \mathcal{A}, \mathcal{S}, \mathcal{G}, \mathcal{H}$  represent suitable function spaces over their respective domains.

The objective of the inverse problem is to determine unknown quantities, i.e., the coefficient  $a$ , in the PDE system from (possibly noisy) measurements of the solution  $u$ . Formally, let  $\mathcal{M}(u)$  denote the measurement operator, the form of which depends on the accessibility of sensors:

- **Solution-based measurements.** In this setting, observations consist of solution values at  $m$  sensor locations  $\mathbf{X}_m = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \Omega$ . Therefore, the measurement operator is defined as  $\mathcal{M}_{sol} = (u(x_1), \dots, u(x_m)) \in \mathbb{R}^m$ . The inverse problem then seeks a coefficient  $a$  such that the corresponding PDE solution  $u$  matches  $\mathcal{M}_{sol}$ .
- **Operator-based measurements.** When interior measurements are unavailable, one seeks to impose a set of boundary conditions  $\{g_1, g_2, \dots, g_L\}$  on  $\partial\Omega$  and record the corresponding measurements  $\{\Lambda_a[g_1], \dots, \Lambda_a[g_L]\}$  on boundary, where  $\Lambda_a$  denotes the boundary observation operator. For instance, in EIT,  $\Lambda_a$  corresponds to the DtN operator that maps an imposed voltage  $g$  to the induced current  $a\nabla u \cdot \vec{n}$ , with  $\vec{n}$  being the unit outward normal vector. The resulting measurement data can be expressed as:

$$\mathcal{M}_{op} : g_l(\mathbf{x}_i) \mapsto \Lambda_a[g_l](\mathbf{x}_i), \quad l = 1, \dots, L; i = 1, \dots, m. \tag{4}$$

where  $\mathbf{X}_m = \{\mathbf{x}_1, \dots, \mathbf{x}_m\} \subset \partial\Omega$  denotes boundary sensors. An illustration of the operator-based measurement  $\mathcal{M}_{op}$  in the EIT problem is provided in Supplementary Figure 2 of the SI. The inverse problem in this setting seeks to recover the coefficient  $a$  such that the induced operator  $\Lambda_a$  matches the measured operator  $\mathcal{M}_{op}$ .

### 4.2 The Inverse Generative Neural Operator (IGNO) framework

**Architecture overview.** The IGNO framework consists of four neural network components that together enable efficient, physics-informed inversion, as illustrated in Fig. 1. Two encoder networks  $E_{\theta_{\beta_1}}$  and  $E_{\theta_{\beta_2}}$  compress high-dimensional inputs into low-dimensional latent representations. Specifically, the coefficient encoder  $E_{\theta_{\beta_1}}$  maps a coefficient field  $a$  to a latent variable  $\beta_1 \in \mathbb{R}^{d_{\beta_1}}$ , while the boundary encoder  $E_{\theta_{\beta_2}}$  maps a boundary condition  $g$  to a latent variable  $\beta_2 \in \mathbb{R}^{d_{\beta_2}}$ . These are concatenated to form the joint latent representation  $\beta = (\beta_1, \beta_2)$ . Two decoder networks then map latent representations to outputs: the coefficient decoder  $\mathcal{G}_{\theta_a}$  reconstructs the coefficient field  $a$  from  $\beta_1$ , while the solution decoder  $\mathcal{G}_{\theta_u}$  predicts the PDE solution  $u$  from the combined latent representation  $\beta$ . Both decoders are parameterized using the MultiONet architecture [39], an enhanced variant of DeepONet that aggregates information across multiple network layers to improve representational capacity. The detailed architecture of the MultiONet is described in Supplementary Note 1 of SI, with a schematic illustration presented in Supplementary Figure 1. For solution-based inverse problems with fixed boundary conditions, the boundary encoder is omitted, and  $\beta = \beta_1$  serves as the sole latent representation. In contrast, for operator-based problems, both encoders are utilized to accommodate varying boundary conditions while maintaining a shared coefficient representation across all boundary inputs.

**Physics-informed training.** All encoder and decoder parameters,  $\theta = (\theta_u, \theta_a, \theta_{\beta_1}, \theta_{\beta_2})$ , are jointly optimized using a physics-informed loss function that embeds the governing PDEs directly into training, eliminating the need for labeled solution data:

$$\mathcal{L}(\theta) = \lambda_{pde}\mathcal{L}_{pde}(\theta) + \lambda_{bd}\mathcal{L}_{bd}(\theta_u, \theta_{\beta_1}, \theta_{\beta_2}) + \lambda_{rec}\mathcal{L}_{rec}(\theta_a, \theta_{\beta_1}, \theta_{\beta_2}), \tag{5}$$

where  $\lambda_{pde}$ ,  $\lambda_{bd}$ , and  $\lambda_{rec}$  are positive weights balancing each term. Given training data  $\mathcal{D} = \{g^{(n)}, a^{(n)}\}_{n=1}^N$ , which contains only coefficient fields and boundary conditions, the **PDE loss**  $\mathcal{L}_{pde}$  enforces that the predicted coefficients and solutions satisfy the governing equations through residual minimization:

$$\mathcal{L}_{pde} = \frac{1}{N} \sum_{n=1}^N \left\| \mathcal{R}^{(n)}(\mathcal{G}_{\theta_u}(\boldsymbol{\beta}^{(n)}), \mathcal{G}_{\theta_a}(\boldsymbol{\beta}_1^{(n)})) \right\|_2^2, \quad (6)$$

where  $\mathcal{R}^{(n)} = (r_{w_1}^{(n)}, \dots, r_{w_K}^{(n)}) \in \mathbb{R}^K$  is a vector of weighted residuals, with each component defined as:

$$r_{w_k}^{(n)} = \int_{\Omega} [\mathcal{N}(\mathcal{G}_{\theta_u}(\boldsymbol{\beta}^{(n)}), \mathcal{G}_{\theta_a}(\boldsymbol{\beta}_1^{(n)})) - s^{(n)}] w_k d\boldsymbol{x}. \quad (7)$$

Here,  $w_k$  denotes the weighting function. For strong-form residuals,  $w_k$  is chosen as a Dirac delta function  $\delta(\boldsymbol{x} - \boldsymbol{x}_k)$ , yielding pointwise residual evaluation at collocation points  $\boldsymbol{x}_k$ . For weak-form residuals, preferred when dealing with discontinuous coefficients or reduced solution regularity, compactly supported radial basis functions (CSRBFs) [19, 41] are employed as weighting functions. These localized functions significantly reduce computational cost while improving handling of singularities and adaptation to complex geometries.

The **boundary loss**,  $\mathcal{L}_{bd}$ , enforces boundary conditions:

$$\mathcal{L}_{bd} = \frac{1}{N} \sum_{n=1}^N \left\| \mathcal{B}^{(n)}(\mathcal{G}_{\theta_u}(\boldsymbol{\beta}^{(n)})(\Xi_{bd})) - \mathbf{g}^{(n)} \right\|_2^2, \quad (8)$$

where  $\mathbf{g}^{(n)} = g^{(n)}(\Xi_{bd}) \in \mathbb{R}^{N_{bd}}$  represents boundary values at the boundary sensor set  $\Xi_{bd} = (\xi_{bd,1}, \dots, \xi_{bd,N_{bd}}) \subset \partial\Omega$ .

The **reconstruction loss**,  $\mathcal{L}_{rec}$ , ensures that the coefficient decoder accurately reconstructs input coefficients from their latent representations:

$$\mathcal{L}_{rec} = \frac{1}{N} \sum_{n=1}^N \left\| \mathcal{G}_{\theta_a}(\boldsymbol{\beta}_1^{(n)})(\Xi_a) - \mathbf{a}^{(n)} \right\|_2^2, \quad (9)$$

where  $\mathbf{a}^{(n)} \in \mathbb{R}^{N_a}$  denotes coefficient values sampled at grid points  $\Xi_a = (\xi_{a,1}, \dots, \xi_{a,N_a}) \subset \Omega$ . For piecewise-constant coefficients, this loss is replaced by a cross-entropy formulation in which the decoder predicts phase membership probabilities rather than permeability values, preventing gradient instability at discontinuities. Further methodological details are provided in the Supplementary Methods of the SI.

The model is trained using stochastic gradient descent with the ADAM optimizer, an initial learning rate of  $1 \times 10^{-3}$  (halved every 2,500 epochs), a batch size of 50, and a total of 10,000 epochs until convergence. This purely physics-driven training requires no precomputed PDE solutions, in contrast to data-driven DNOs that depend on large labeled datasets. Once trained, the solution decoder  $\mathcal{G}_{\theta_u}$  serves as a fast, surrogate PDE solver, while the coefficient decoder  $\mathcal{G}_{\theta_a}$  enables efficient coefficient reconstruction from optimized latent representations. The algorithm for training IGNO is provided in Supplementary Note 3 of the SI.

### 4.3 Gradient-based inversion in latent space

With trained encoders and decoders, inverse problems are solved via gradient-based optimization directly in the low-dimensional, structured latent space, rather than in the original high-dimensional and often irregular coefficient space. This transformation substantially enhances both tractability and stability of the inversion process.

**Solution-based case.** For inverse problems with solution-based measurements  $\mathcal{M}_{sol}$ , where the boundary condition is fixed, the latent variable reduces to  $\boldsymbol{\beta} = \boldsymbol{\beta}_1$ . The inversion is formulated as the following optimization problem:

$$\begin{aligned} \boldsymbol{\beta}_1^* &= \arg \min_{\boldsymbol{\beta}_1} \mathcal{F}(\boldsymbol{\beta}_1) := \mathcal{F}_{data}(\boldsymbol{\beta}_1) + \mathcal{F}_{pde}(\boldsymbol{\beta}_1) \\ &= \frac{\rho_{data}}{m} \sum_{i=1}^m |\mathcal{G}_{\theta_u}(\boldsymbol{\beta}_1)(\boldsymbol{x}_i) - u(\boldsymbol{x}_i)|^2 + \rho_{pde} \left\| \mathcal{R}(\mathcal{G}_{\theta_u}^*(\boldsymbol{\beta}), \mathcal{G}_{\theta_a}^*(\boldsymbol{\beta}_1)) \right\|_2^2. \end{aligned} \quad (10)$$

Here, the term  $\mathcal{F}_{data}$  measures the data mismatch between predicted and observed solution at sensor locations  $\mathbf{X}_m$ , weighted by  $\rho_{data}$ . The term  $\mathcal{F}_{pde}$  serves as a physics-based regularization term, weighted by  $\rho_{pde}$ , enforcing consistency with the PDE by minimizing the residual norm.

**Operator-based case.** For inverse problems with operator-based measurements  $\mathcal{M}_{op}$  as in (4), the full latent variable  $\beta = (\beta_1, \beta_2)$  is employed, but optimization is performed only with respect to  $\beta_1$  to recover the unknown coefficient field:

$$\begin{aligned} \beta_1^* &= \arg \min_{\beta_1} \mathcal{F}(\beta_1) := \mathcal{F}_{data}(\beta_1) + \mathcal{F}_{pde}(\beta_1) \\ &= \frac{\rho_{data}}{mL} \sum_{l=1}^L \sum_{i=1}^m \left| \Lambda_{\mathcal{G}_{\theta_a^*}(\beta_1)}[g_l](\mathbf{x}_i) - \Lambda_a[g_l](\mathbf{x}_i) \right|^2 + \frac{\rho_{pde}}{L} \sum_{l=1}^L \left\| \mathcal{R}^{(l)} \left( \mathcal{G}_{\theta_a^*}(\beta^{(l)}), \mathcal{G}_{\theta_a^*}(\beta_1) \right) \right\|_2^2. \end{aligned} \quad (11)$$

In this case, the term  $\mathcal{F}_{data}$  quantifies the mismatch between predicted and true operator outputs across all  $L$  boundary conditions and  $m$  sensors, weighted by  $\rho_{data}$ . The term  $\mathcal{F}_{pde}$  enforces PDE consistency for each boundary condition, with  $\beta^{(l)} = (\beta_1, E_{\theta_{\beta_2}^*}(g_l))$  combining the coefficient latent representation and the encoded boundary condition.

The latent-space optimization is performed using the ADAM optimizer with an initial learning rate of 0.01, decayed by a factor of 2/3 every 250 steps for solution-based cases, or adaptively per problem for operator-based cases. Convergence typically occurs within 200–500 iterations. Gradients with respect to  $\beta_1$  are computed efficiently via automatic differentiation through the trained decoders. After convergence, the optimal latent vector  $\beta_1^*$  is passed through the coefficient decoder  $\mathcal{G}_{\theta_a^*}$  to reconstruct the recovered coefficient field  $a$ . The algorithm for latent-space optimization is provided in Supplementary Note 3 of the SI.

#### 4.4 Normalizing flow-based a priori initialization

To enhance the efficiency and robustness of latent-space optimization, we introduce a normalizing flow model  $F_{\theta_{NF}} : \mathbb{R}^{\beta_1} \rightarrow \mathbb{R}^{\beta_1}$  that maps the learned latent space of  $\beta_1$  to a standard multivariate normal distribution  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ . Specifically, we employ the Real-valued Non-Volume Preserving (RealNVP) architecture [42], an invertible neural network that enables tractable density estimation via the change-of-variables:

$$\log p_{\beta_1}(\beta_1) = \log p_{\mathbf{z}}(\mathbf{z}) + \log \left| \det \frac{\partial \mathbf{z}}{\partial \beta_1} \right|, \quad (12)$$

where  $\mathbf{z} = F_{\theta_{NF}}(\beta_1)$  follows the standard Gaussian density  $p_{\mathbf{z}}$ . A detailed description of the flow model is provided in Note 2 of the SI. The flow is trained by minimizing the negative log-likelihood over the training set of learned latent representations obtained from the trained IGNO model:

$$\mathcal{L}_{NF}(\theta_{NF}) = -\frac{1}{N} \sum_{n=1}^N \left[ \log p_{\mathbf{z}}(F_{\theta_{NF}}(\beta_1^{(n)})) + \log \left| \det \frac{\partial F_{\theta_{NF}}(\beta_1^{(n)})}{\partial \beta_1} \right| \right]. \quad (13)$$

Once trained, the flow model provides statistically informed initialization by sampling  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  from the standard Gaussian and mapping to the latent space via the inverse flow  $\beta_1^{init} = F_{\theta_{NF}}^{-1}(\mathbf{z})$ . This initialization strategy captures the distribution of physically plausible coefficients learned during training, substantially accelerating convergence and improving robustness compared to random initialization, particularly in noisy or ill-posed scenarios.

#### 4.5 Evaluation metrics

Two quantitative metrics are employed to assess reconstruction performance. For continuous coefficients, the **relative root-mean-square error (RMSE)** measures reconstruction accuracy as:

$$\text{RMSE} = \sqrt{\frac{\sum_i (a^{rec}(\mathbf{x}_i) - a^{true}(\mathbf{x}_i))^2}{\sum_i (a^{true}(\mathbf{x}_i))^2}}, \quad (14)$$

where  $a^{rec}$  and  $a^{true}$  denote recovered and ground-truth coefficients evaluated at grid points.

For discontinuous targets, morphological similarity between piecewise-constant fields is quantified using the **cross-correlation indicator** ( $I_{corr}$ ) [43]:

$$I_{corr} = \frac{\sum_i (\tilde{a}^{true}(\mathbf{x}_i))^2 (\tilde{a}^{rec}(\mathbf{x}_i))^2}{\sqrt{\sum_i (\tilde{a}^{ture}(\mathbf{x}_i))^2} \sqrt{\sum_i (\tilde{a}^{rec}(\mathbf{x}_i))^2}}, \quad (15)$$

where  $\tilde{a}$  denotes coefficients rescaled to [0,1]. The indicator  $I_{corr}$  ranges from 0 to 1, with values close to 1 indicating strong morphological agreement between the reconstructed and true phase topologies.

## 5 Code Availability Statement

All code necessary to reproduce the results presented in this paper will be made publicly available upon publication in our GitHub repository.

## 6 Data Availability Statement

Upon publication, the training datasets and pre-trained models will be made publicly available via Kaggle dataset. The raw data to create the figures of the main text can be made available from the corresponding author upon reasonable request.

## Acknowledgement

The work is partially supported by National Natural Science Foundation of China (U21A20425) and a Key Laboratory of Zhejiang Province.

## References

- [1] R. H. Stolt, A. B. Weglein, Seismic imaging and inversion: Volume 1: Application of linear inverse theory, Cambridge University Press, Cambridge, UK, 2012.
- [2] G. Uhlmann, Electrical impedance tomography and calderón's problem, *Inverse problems* 25 (12) (2009) 123011.
- [3] M. Bertero, P. Boccacci, C. De Mol, Introduction to inverse problems in imaging, CRC press, Boca Raton, FL, 2021.
- [4] Y. Zang, P.-S. Koutsourelakis, Psp-gen: Stochastic inversion of the process–structure–property chain in materials design through deep, generative probabilistic modeling, *Acta Materialia* 284 (2025) 120600.
- [5] D. W. Hahn, M. N. Özisik, Heat conduction, John Wiley & Sons, Hoboken, NJ, 2012.
- [6] F. M. White, J. Majdalani, Viscous fluid flow, Vol. 3, McGraw-Hill New York, New York, NY, 2006.
- [7] J. G. Van Bladel, Electromagnetic fields, John Wiley & Sons, Hoboken, NJ, 2007.
- [8] R. W. Ogden, Non-linear elastic deformations, Dover Publications, Mineola, NY, 1997.
- [9] G. Chavent, Nonlinear least squares for inverse problems: theoretical foundations and step-by-step guide for applications, Springer Science & Business Media, Dordrecht, 2010.
- [10] C. L. Byrne, Iterative optimization in inverse problems, CRC Press, Boca Raton, FL, 2014.
- [11] M. Frangos, Y. Marzouk, K. Willcox, B. van Bloemen Waanders, Surrogate and reduced-order modeling: a comparison of approaches for large-scale statistical inverse problems, *Large-Scale Inverse Problems and Quantification of Uncertainty* (2010) 123–149.
- [12] D. Calvetti, E. Somersalo, Inverse problems: From regularization to bayesian inference, *Wiley Interdisciplinary Reviews: Computational Statistics* 10 (3) (2018) e1427.
- [13] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, *Proceedings of the National Academy of Sciences* 115 (34) (2018) 8505–8510.
- [14] J. Sirignano, K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, *Journal of computational physics* 375 (2018) 1339–1364.
- [15] B. Yu, et al., The deep ritz method: a deep learning-based numerical algorithm for solving variational problems, *Communications in Mathematics and Statistics* 6 (1) (2018) 1–12.
- [16] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational physics* 378 (2019) 686–707.
- [17] E. Kharazmi, Z. Zhang, G. E. Karniadakis, Variational physics-informed neural networks for solving partial differential equations, *arXiv preprint arXiv:1912.00873* (2019).
- [18] Y. Zang, G. Bao, X. Ye, H. Zhou, Weak adversarial networks for high-dimensional partial differential equations, *Journal of Computational Physics* 411 (2020) 109409.
- [19] Y. Zang, G. Bao, Particlewnn: a novel neural networks framework for solving partial differential equations, *arXiv preprint arXiv:2305.12433* (2023).
- [20] G. Bao, X. Ye, Y. Zang, H. Zhou, Numerical solution of inverse problems by weak adversarial networks, *Inverse Problems* 36 (11) (2020) 115003.
- [21] Y. Chen, L. Lu, G. E. Karniadakis, L. Dal Negro, Physics-informed neural networks for inverse problems in nano-optics and metamaterials, *Optics express* 28 (8) (2020) 11618–11633.
- [22] M. Rasht-Behesht, C. Huber, K. Shukla, G. E. Karniadakis, Physics-informed neural networks (pinns) for wave propagation and full waveform inversions, *Journal of Geophysical Research: Solid Earth* 127 (5) (2022) e2021JB023120.
- [23] H. Gao, M. J. Zahr, J.-X. Wang, Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems, *Computer Methods in Applied Mechanics and Engineering* 390 (2022) 114502.
- [24] A. D. Jagtap, Z. Mao, N. Adams, G. E. Karniadakis, Physics-informed neural networks for inverse problems in supersonic flows, *Journal of Computational Physics* 466 (2022) 111402.

- [25] V. C. Scholz, Y. Zang, P.-S. Koutsourelakis, Weak neural variational inference for solving bayesian inverse problems without forward models: applications in elastography, *Computer Methods in Applied Mechanics and Engineering* 433 (2025) 117493.
- [26] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Graph kernel network for partial differential equations, arXiv preprint arXiv:2003.03485 (2020).
- [27] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, arXiv preprint arXiv:2010.08895 (2020).
- [28] L. Lu, P. Jin, G. Pang, Z. Zhang, G. E. Karniadakis, Learning nonlinear operators via deeponet based on the universal approximation theorem of operators, *Nature machine intelligence* 3 (3) (2021) 218–229.
- [29] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to pdes, *Journal of Machine Learning Research* 24 (89) (2023) 1–97.
- [30] Z. Gao, L. Yan, T. Zhou, Adaptive operator learning for infinite-dimensional bayesian inverse problems, *SIAM/ASA Journal on Uncertainty Quantification* 12 (4) (2024) 1389–1423.
- [31] Q. Guo, Y. He, M. Liu, Y. Zhao, Y. Liu, J. Luo, Reduced geostatistical approach with a fourier neural operator surrogate model for inverse modeling of hydraulic tomography, *Water Resources Research* 60 (6) (2024) e2023WR034939.
- [32] R. Molinaro, Y. Yang, B. Engquist, S. Mishra, Neural inverse operators for solving pde inverse problems, arXiv preprint arXiv:2301.11167 (2023).
- [33] T. Wang, C. Wang, Latent neural operator for solving forward and inverse pde problems, *Advances in Neural Information Processing Systems* 37 (2024) 33085–33107.
- [34] D. Long, Z. Xu, Q. Yuan, Y. Yang, S. Zhe, Invertible fourier neural operators for tackling both forward and inverse problems, arXiv preprint arXiv:2402.11722 (2024).
- [35] S. Wang, H. Wang, P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed deeponets, *Science advances* 7 (40) (2021) eabi8605.
- [36] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, *ACM/IMS Journal of Data Science* 1 (3) (2024) 1–27.
- [37] N. Navaneeth, T. Tripura, S. Chakraborty, Physics informed wno, *Computer Methods in Applied Mechanics and Engineering* 418 (2024) 116546.
- [38] A. Jiao, Q. Yan, J. Harlim, L. Lu, Solving forward and inverse pde problems on unknown manifolds via physics-informed neural operators, arXiv preprint arXiv:2407.05477 (2024).
- [39] Y. Zang, P.-S. Koutsourelakis, Dgenno: a novel physics-aware neural operator for solving forward and inverse pde problems based on deep, generative probabilistic modeling, *Journal of Computational Physics* 538 (2025) 114137.
- [40] S. W. Cho, H. Son, Physics-informed deep inverse operator networks for solving pde inverse problems, arXiv preprint arXiv:2412.03161 (2024).
- [41] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, *Advances in computational Mathematics* 4 (1995) 389–396.
- [42] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real nvp, arXiv preprint arXiv:1605.08803 (2016).
- [43] P. Bourke, Cross correlation, Cross Correlation”, Auto Correlation—2D Pattern Identification 596 (1996).

# Supplementary Material

## A Supplementary Note 1: The MultiONet Architecture

The MultiONet architecture extends the standard DeepONet by aggregating information from multiple intermediate layers of both branch and trunk networks, enriching the learned representation without increasing trainable parameters. The architecture, illustrated in Fig. 1, comprises two sub-networks:

- **Trunk network:** Encodes the spatial coordinates  $\mathbf{x} \in \Omega$  of the output field.
- **Branch network:** Encodes a latent vector  $\beta$  representing a compact descriptor of the input functions.

Unlike the original DeepONet, MultiONet computes a weighted average of inner products between layer-wise features from multiple network depths rather than relying solely on last-layer features. Formally, the MultiONet mapping  $\mathcal{G}(\beta)(\mathbf{x})$  is expressed as:

$$\mathcal{G}(\beta)(\mathbf{x}) = \frac{1}{l} \sum_{k=1}^l w^{(k)} \left( b^{(k)}(\beta) \odot t^{(k)}(\mathbf{x}) \right) + b_0, \quad (16)$$

where  $b^{(k)}(\beta)$  and  $t^{(k)}(\mathbf{x})$  denote the outputs from the  $k$ -th layers of the branch and trunk networks,  $l$  represents the total number of layers,  $w^{(k)}$  indicates trainable weights,  $b_0$  is the bias term, and  $\odot$  represents the inner product operation.

## B Supplementary Note 2: Normalizing Flow for Initialization

In gradient-based inversion, initial guess quality for the latent variable affects both convergence speed and optimization quality. Random initialization (e.g., from uniform or Gaussian distributions) ignores the trained latent space structure and may lead to inefficient optimization, particularly for highly nonlinear PDE inverse problems. We employ a normalizing flow (NF) model to learn a bijective mapping between the latent space and a standard multivariate normal distribution, enabling informed and probabilistically meaningful initialization. Specifically, we use the RealNVP architecture [42], which consists of invertible coupling layers. Each coupling layer splits the input  $\mathbf{x} \in \mathbb{R}^d$  into  $\mathbf{z}_1 \in \mathbb{R}^{d/2}$  and  $\mathbf{z}_2 \in \mathbb{R}^{d/2}$ , then applies:

- Forward:  $\mathbf{y}_1 = \mathbf{z}_1$ ,  $\mathbf{y}_2 = \mathbf{z}_2 \odot \exp(s(\mathbf{z}_1)) + t(\mathbf{z}_1)$
- Inverse:  $\mathbf{z}_1 = \mathbf{y}_1$ ,  $\mathbf{z}_2 = (\mathbf{y}_2 - t(\mathbf{y}_1)) \odot \exp(-s(\mathbf{y}_1))$

where  $s(\cdot)$  and  $t(\cdot)$  are scale and translation networks parameterized by fully connected neural networks (FCNNs).

## C Supplementary Note 3: Training and Inversion Algorithms

---

### Algorithm 1 Training the IGNO Framework

---

**Inputs:**

Training data  $\mathcal{D} = \{a^{(n)}, g^{(n)}\}_{n=1}^N$ ; loss weights  $\lambda_{pde}, \lambda_{bd}, \lambda_{rec}$ .

**Initialize:**

Model parameters  $\theta = (\theta_u, \theta_a, \theta_{\beta_1}, \theta_{\beta_2})$ ; learning rate  $lr$ .

**while** Convergence or maximum number of iterations not reached **do**

Obtain latent representations:

$$\beta_1^{(n)} \leftarrow E_{\beta_1}(\mathbf{a}^{(n)}), \quad \beta_2^{(n)} \leftarrow E_{\beta_2}(\mathbf{g}^{(n)}). \quad n = 1, \dots, N$$

Obtain the predicted PDE solutions and recovered coefficients:

$$u_{\text{pred}}^{(n)} \leftarrow \mathcal{G}_{\theta_u}(\beta_1^{(n)}), \quad a_{\text{pred}}^{(n)} \leftarrow \mathcal{G}_{\theta_a}(\beta_1^{(n)}). \quad n = 1, \dots, N$$

Compute the loss  $\mathcal{L}(\theta)$  and update model parameters with SGD:

$$\theta \leftarrow \theta - lr \odot \nabla_{\theta} \mathcal{L}(\theta).$$

**if** at every 2500th epoch **then**

    Update the learning rate to  $lr \leftarrow lr/2$ .

**end if**

**end while**

**Outputs:**

Trained encoders  $E_{\theta_{\beta_1}^*}, E_{\theta_{\beta_2}^*}$ , and decoders  $\mathcal{G}_{\theta_u^*}, \mathcal{G}_{\theta_a^*}$ .

---



---

### Algorithm 2 Inversion via Latent Optimization

---

1: **Inputs:**

Trained networks  $E_{\theta_{\beta_1}^*}, E_{\theta_{\beta_2}^*}, \mathcal{G}_{\theta_u^*}, \mathcal{G}_{\theta_a^*}$ ; Trained normalizing flow  $F_{\theta_{NF}^*}$ ;  
measurements  $\mathcal{M}$

2: **Initialize:**

Sample  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ; Set  $\beta_1 \leftarrow F_{\theta_{NF}^*}^{-1}(\mathbf{z})$ ; learning rate  $lr$

3: **while** Convergence or maximum number of iterations not reached **do**

4:     Compute optimization objective  $\mathcal{F}(\beta_1)$ .

5:     Update  $\beta_1$  with gradient descent:

$$\beta_1 \leftarrow \beta_1 - lr \cdot \nabla_{\beta_1} \mathcal{F}(\beta_1)$$

6:     **if** at every 250th epoch **then**

7:         Update the learning rate to  $lr \leftarrow lr/2$ .

8:     **end if**

9: **end while**

10: **Outputs:**

Recovered coefficient  $a^{rec} = \mathcal{G}_{\theta_a^*}(\beta_1)$ .

---

## D Supplementary Note 4: Baseline Method Details

We compare IGNO against physics-informed deep inverse operator networks (PI-DIONs) [40], a recent state-of-the-art method for PDE inverse problems with solution-based measurements. This method employs the DeepONet architecture to parameterize both the unknown coefficient  $a$  and the PDE solution  $u$ . Coordinates  $\mathbf{x}$  are treated as inputs to the trunk networks, while solution-based measurements  $\mathcal{M}_{sol}$  are input to the branch networks. The two resulting models are denoted  $\mathcal{G}_{\theta_u}$  and  $\mathcal{G}_{\theta_a}$ , parameterized by  $\theta_u$  and  $\theta_a$ , respectively. Predictions at location  $\mathbf{x}$  are given by  $\mathcal{G}_{\theta_u}(\mathcal{M}_{sol})(\mathbf{x})$  for the solution and  $\mathcal{G}_{\theta_a}(\mathcal{M}_{sol})(\mathbf{x})$  for the coefficient.

Training of PI-DIONs relies on a physics-informed composite loss that combines (i) a physics loss, consisting of strong-form PDE residuals ( $\mathcal{L}_{pde}$ ) and boundary mismatches ( $\mathcal{L}_{bd}$ ), and (ii) a data loss ( $\mathcal{L}_{data}$ ) that penalizes deviations between predicted and observed solutions at fixed sensors. For  $N$  training samples with solution-based measurements  $\{\mathcal{M}_{sol}^{(n)}\}_{n=1}^N$  collected on sensors  $\mathbf{X}_m = (\mathbf{x}_1, \dots, \mathbf{x}_m) \subset \Omega$ , the total loss is defined as:

$$\begin{aligned}\mathcal{L} &= \lambda_{physics}(\mathcal{L}_{pde} + \mathcal{L}_{bd}) + \lambda_{data}\mathcal{L}_{data} \\ &= \frac{\lambda_{physics}}{N} \sum_{n=1}^N \left( \|\mathcal{R}^{(n)}(\mathcal{G}_{\theta_u}(\mathcal{M}_{sol}^{(n)}), \mathcal{G}_{\theta_u}(\mathcal{M}_{sol}^{(n)}))\|_2^2 + \|\mathcal{B}^{(n)}(\mathcal{G}_{\theta_u}(\mathcal{M}_{sol}^{(n)})(\Xi_{bd})) - \mathbf{g}^{(n)}\|_2^2 \right) \\ &\quad + \frac{\lambda_{data}}{N} \sum_{n=1}^N \|\mathcal{G}_{\theta_u}(\mathcal{M}_{sol}^{(n)})(\mathbf{X}_m) - \mathcal{M}_{sol}^{(n)}\|_2^2,\end{aligned}\tag{17}$$

where  $\mathcal{R}^{(n)} = (r_{w_1}^{(n)}, \dots, r_{w_K}^{(n)}) \in \mathbb{R}^K$  denotes the strong-form residual vector. The terms  $\lambda_{physics}$  and  $\lambda_{data}$  are loss weights. Once trained, the recovered coefficient  $a$  and solution  $u$  for test measurements  $\mathcal{M}_{sol}^{test}$  are directly obtained via  $\mathcal{G}_{\theta_a^*}(\mathcal{M}_{sol}^{test})$  and  $\mathcal{G}_{\theta_u^*}(\mathcal{M}_{sol}^{test})$ , respectively. PI-DIONs is not applicable to operator-based inverse problems and serves as a baseline only for solution-based cases.

## E Supplementary Methods

### E.1 General Implementation Details

Unless otherwise specified, all models are trained using the ADAM optimizer with an initial learning rate of  $1 \times 10^{-3}$ , reduced by half every 2500 epochs. The batch size is set to 50, and training proceeds for 10,000 epochs to ensure convergence. For the proposed IGNO framework, loss weights are set as  $\lambda_{pde} = 0.25$ ,  $\lambda_{bd} = 0.5$ , and  $\lambda_{rec} = 2$ . For PI-DIONs, loss weights are set as  $\lambda_{physics} = 1$  and  $\lambda_{data} = 2$  for best performance. All experiments are conducted under identical hardware conditions on a 64-core AMD Ryzen CPU with an NVIDIA RTX 4090 GPU.

### E.2 Continuous Coefficient Recovery with Solution-based Measurements

#### E.2.1 Problem Setup

The governing PDE is the following Darcy's flow equation:

$$\begin{aligned}-\nabla(k(x, y)\nabla p(x, y)) &= f(x, y), \quad \text{in } \Omega = [0, 1]^2, \\ p(x, y) &= 0, \quad \text{in } \partial\Omega,\end{aligned}\tag{18}$$

where  $k$  is the permeability field,  $p$  is the pressure field, and  $f = 10$  is the source term. Measurements are collected at  $m = 100$  sensor locations  $\mathbf{X}_m = (\mathbf{x}_1, \dots, \mathbf{x}_m)$ , with additive Gaussian noise contamination:

$$\mathcal{M}_{sol} = (\hat{p}(\mathbf{x}_1), \dots, \hat{p}(\mathbf{x}_m)), \text{ where } \hat{p}(\mathbf{x}_i) = p(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2).\tag{19}$$

Noise levels are characterized by the signal-to-noise ratio (SNR), which is defined as:

$$\text{SNR} = 10 \log 10 \left( \frac{\frac{1}{m} \sum_{i=1}^m p^2(\mathbf{x}_i)}{\sigma^2} \right).\tag{20}$$

In particular, we examine three scenarios: low noise (SNR=50), medium noise (SNR=25), and high noise (SNR=15).

For training IGNO,  $N = 1000$  synthetic permeability fields are generated as  $k(x, y) = 2.1 + \sin(\omega_1 x) + \cos(\omega_2 y)$  with  $\omega_1, \omega_2$  sampled independently from the uniform distribution  $U(0, 7\pi/4)^2$ . Since the boundary condition is fixed (i.e.,  $g = 0$ ) across all samples, only the permeability encoder  $E_{\beta_1}$  is required. For training PI-DIONs, pressure fields  $p^{(n)}$  are computed by solving (1) via the finite element method (FEM) for each permeability sample  $k^{(n)}$ . Then, training measurements  $\mathcal{M}_{sol}^{(n)}$  are formulated by sampling the pressure values at the same sensor locations  $\mathbf{X}_m$  without added noise.

To evaluate the performance of both methods on this problem, we consider two classes of test targets:

- **In-distribution target:** permeability fields  $k$  generated with  $(\omega_1, \omega_2) \sim U(0, 7\pi/4)$ , consistent with the training distribution.
- **Out-of-distribution target:** permeability fields  $k$  generated with  $(\omega_1, \omega_2) \sim U(7\pi/4, 2\pi)$ , which lie entirely outside the training distribution.

In both cases, noisy test measurements  $\mathcal{M}_{sol}^{test}$  are generated by solving (1) with FEM and perturbing the pressure values with Gaussian noise at the specified SNR levels.

### E.2.2 Model Setup

To extract latent representations from the input coefficients with the encoder  $E_{\theta_{\beta_1}}$ , we set  $\Xi_a = (\xi_{a,1}, \dots, \xi_{a,N_a}) \subset \Omega$  as a  $29 \times 29$  uniform grid, so that the encoder input is  $\mathbf{a} = a(\Xi_a)$ . Since the boundary condition is fixed ( $g = 0$ ) across all samples, no boundary input is needed (i.e., the boundary encoder vanishes). Moreover, a mollifier  $f(x, y) = \sin(\pi x) \sin(\pi y)$  is applied to the output of the solution decoder, i.e.,

$$u_{pred}(x, y) = \mathcal{G}_{\theta_u}(\beta)(x, y) \cdot f(x, y), \quad (21)$$

which enforces the boundary condition automatically and removes the need for an explicit boundary loss during training. Below, we provide details of the network architectures for both methods:

#### The IGNO method

- **Encoder  $E_{\theta_{\beta_1}}$ .** It consists of a Convolutional Neural Network (CNN) followed by a Feed-Forward Fully Connected Network (FFCN). The CNN has three hidden layers with 64 output channels each, kernel size  $(3, 3)$ , and stride 2. The FFCN has two hidden layers of 128 neurons each. The SiLU activation is applied to all hidden layers, while the output layer uses Tanh to constrain the latent variables to a bounded cubic region.
- **Solution Decoder  $\mathcal{G}_{\theta_u}$ .** The solution decoder  $\mathcal{G}_{\theta_u}$  adopts the MultiONet architecture. Both branch and trunk networks are FFCNs with 6 hidden layers of 100 neurons each. A custom activation function, Tanh\_Sin, is used in all hidden layers:

$$\text{Tanh\_Sin}(x) = \tanh(\sin(\pi x + \pi)) + x. \quad (22)$$

- **Coefficient Decoder  $\mathcal{G}_{\theta_a}$ .** The coefficient decoder  $\mathcal{G}_{\theta_a}$  adopts the same architecture as  $\mathcal{G}_{\theta_u}$ .
- **NF model  $F_{\theta_{NF}}$ .** The NF model consists of three flow steps, each parameterized by a fully connected network with two hidden layers of 64 neurons per layer and SiLU activations.

#### The PI-DIONs method

- **Decoder  $\mathcal{G}_{\theta_u}$ .** For fair comparison, the PI-DIONs solution decoder adopts the same architecture as in IGNO. However, the input to the branch network is the solution-based measurement  $\mathcal{M}_{sol}$ , instead of the latent variable  $\beta$ .
- **Decoder  $\mathcal{G}_{\theta_a}$ .** Similarly, the coefficient decoder  $\mathcal{G}_{\theta_a}$  uses the same architecture as in IGNO, with the branch network taking  $\mathcal{M}_{sol}$  as input instead of  $\beta_1$ .

### E.2.3 Latent-space Optimization Details

Gradients of the objective  $\mathcal{F}$  with respect to  $\beta_1$  are computed via PyTorch’s automatic differentiation. The weights  $\rho_{data}$  and  $\rho_{pde}$  are set to 50 and 1, respectively. Optimization is performed using ADAM with an initial learning rate of 0.01, decayed by a factor of  $2/3$  every 250 steps, for a total of 500 updates.

## E.3 Piecewise-Constant Coefficient Recovery with Solution-based Measurements

### E.3.1 Problem Setup

The governing equation remains Darcy’s law (1) but with piecewise-constant coefficient:

$$k(x, y) = \begin{cases} 10, & (x, y) \in \Omega_1 \\ 5, & (x, y) \in \Omega_2 \end{cases} \quad (23)$$

where  $\Omega_1$  and  $\Omega_2$  represent two disjoint phases (i.e., phase 1 and phase 2) such that  $\Omega_1 \cup \Omega_2 = \Omega$ . Measurements  $\mathcal{M}_{sol}$  are collected on  $m = 100$  fixed sensors  $\mathbf{X}_m$  randomly sampled in  $\Omega$ . The measurements are contaminated with Gaussian noise, and three levels of noise are considered: low (SNR=50), medium (SNR=20), and high (SNR=15). The discontinuous nature of the coefficient field makes this problem particularly difficult for existing methods: (i) Gradient-based methods require differentiability of  $k$ , but in this case,  $k$  is discrete-valued, so gradients with respect to  $k$  are not well defined; (ii) Non-gradient methods such as evolutionary algorithms or MCMC may, in principle, handle discrete parameters, but they become computationally prohibitive when  $k$  lies in a high- or infinite-dimensional function space, restricting their applicability to only low-dimensional problems. The proposed IGNO framework naturally overcomes these challenges by introducing a low-dimensional and well-structured latent space representation of the original high-dimensional and discontinuous coefficient field. This transforms the otherwise intractable inverse problem into a tractable optimization problem in a continuous latent space, making it both efficient and robust.

To train IGNO, piecewise-constant permeability fields are generated by using a cutoff Gaussian Process  $\mathcal{GP}(0, (-\Delta + 9I)^{-2})$  [27, 39]. For each GP realization,  $k(x, y) = 10$  if the underlying GP-value is greater than 0 and  $k(x, y) = 5$  otherwise. Again, only the permeability encoder  $E_{\beta_1}$  is required as the boundary condition is fixed across all samples. For training PI-DIONs, the pressure field  $p^{(n)}$  corresponding to each permeability sample  $k^{(n)}$  is obtained using FEM. Noise-free pressure values on the fixed sensors  $\mathbf{X}_m$  are then used as training measurements  $\mathcal{M}_{sol}^{(n)}$ .

We evaluate both methods on two types of inverse targets:

- **In-distribution target:** The target permeability  $k$  is generated from the same distribution used in training, i.e., the cutoff  $\mathcal{GP}(0, (-\Delta + 9I)^{-2})$ .
- **Out-of-distribution target:** The target  $k$  is generated from a different distribution, namely the cutoff  $\mathcal{GP}(0, (-\Delta + 16I)^{-2})$ . The higher-order correlation structure of  $k$  differs significantly from the training distribution, resulting in more complex geometries of the permeability field.

In both cases, test measurements  $\mathcal{M}_{sol}^{test}$  are obtained by solving Darcy's equation with FEM, and then corrupted with Gaussian noise corresponding to the specified SNR levels.

### E.3.2 Model Setup

The coefficient  $k$  is represented by an image of size  $29 \times 29$ , denoted by  $\Xi_a = (\xi_{a,1}, \dots, \xi_{a,N_a})$ . Each pixel value corresponds to the permeability of the associated phase (either 10 or 5). To improve the performance of IGNO in this problem, the coefficient decoder is not trained to predict the coefficient values directly. Instead, it predicts the probability that a spatial location  $x$  belongs to phase 1. Accordingly, the recovery loss  $\mathcal{L}_{rec}$  is replaced by a cross-entropy loss:

$$\mathcal{L}_{rec} = \frac{1}{N} \sum_{n=1}^N \sum_{i=1}^{N_a} \left[ z_i^{(n)} \log \sigma(\mathcal{G}_{\theta_a}(\beta_1^{(n)})(\xi_{a,i})) + (1 - z_i^{(n)}) \log (1 - \sigma(\mathcal{G}_{\theta_a}(\beta_1^{(n)})(\xi_{a,i}))) \right], \quad (24)$$

where  $\sigma(\cdot)$  is the sigmoid function, and  $z_i$  is a binary label indicating the true phase at location  $\xi_{a,i}$  ( $z_i = 1$  for phase 1 and  $z_i = 0$  for phase 2). At inference time, the recovered coefficient can be obtained either by sampling from the predicted probability field or, as done in this work, by applying a cut-off rule: values greater than 0.5 are assigned to phase 1 and others to phase 2. This strategy is not applicable to PI-DIONs, as the method relies on strong-form PDE residuals that require differentiability with respect to the coefficient field. Consequently, the coefficient decoder in PI-DIONs is designed to predict permeability values directly. To obtain binary reconstructions, a threshold of 7.5 is then applied to convert the continuous predictions into discrete phase labels. For IGNO, the  $29 \times 29$  coefficient image is used as the coefficient encoder input. Since the boundary condition is fixed ( $g = 0$ ) across all samples, the boundary encoder is omitted. To enforce boundary conditions automatically, a mollifier  $f(x, y) = \sin(\pi x) \sin(\pi y)$  is applied to the solution decoder output:

$$u_{pred}(x, y) = \mathcal{G}_{\theta_u}(\beta)(x, y) \cdot f(x, y). \quad (25)$$

Below, we present the network architectures for both methods:

### The IGNO method

- **Encoder**  $E_{\theta_{\beta_1}}$ . The encoder  $E_{\theta_{\beta_1}}$  is selected as an FFCN. The input coefficient image is first flattened into a vector, then passed through two dense layers with 512 and 256 neurons, respectively. SiLU activations are applied in all hidden layers, and a Tanh activation is applied at the output.
- **Solution Decoder**  $\mathcal{G}_{\theta_u}$ . The solution decoder  $\mathcal{G}_{\theta_u}$  adopts the MultiONet architecture. Both the branch and trunk networks are FFCNs with five hidden layers of 100 neurons each. The custom activation function Tanh\_Sin (Eq. (22)) is applied to all hidden layers.
- **Coefficient Decoder**  $\mathcal{G}_{\theta_a}$ . The coefficient decoder  $\mathcal{G}_{\theta_a}$  outputs the probability that a given spatial point belongs to phase 1. The recovered permeability field is obtained by thresholding the probability field at 0.5 and mapping to 10 (phase 1) or 5 (phase 2). Both branch and trunk networks follow the MultiONet design, with five hidden layers of 256 neurons each. The trunk employs the custom activation

$$\text{SiLU\_Sin}(x) = \text{SiLU}(\sin(\pi x + \pi)) + x, \quad (26)$$

while the branch employs

$$\text{SiLU\_Id}(x) = \text{SiLU}(x) + x. \quad (27)$$

A Sigmoid activation is applied at the output to ensure predictions lie in  $[0, 1]$ .

- **NF model**  $F_{\theta_{NF}}$ . The NF model consists of three flow steps. Each step is parameterized by an FFCN with two hidden layers of 128 neurons and SiLU activations.

### The PI-DIONs method

- **Decoder**  $\mathcal{G}_{\theta_u}$ . The solution decoder of PI-DIONs adopts the same architecture as in IGNO. However, its branch network takes as input the solution-based measurements  $\mathcal{M}_{sol}$  rather than the latent variable  $\beta$ .
- **Decoder**  $\mathcal{G}_{\theta_a}$ . Unlike IGNO, PI-DIONs predict the coefficient field  $a$  directly, because probability-based cut-offs are non-differentiable and incompatible with strong-form residual training. For fairness, the decoder  $\mathcal{G}_{\theta_a}$  adopts the same architecture as in IGNO, except that the branch network input is  $\mathcal{M}_{sol}$ , and no activation is applied at the output layer.

#### E.3.3 Latent-space Optimization Details

Gradients are computed via PyTorch automatic differentiation. The weights  $\rho_{data}$  and  $\rho_{pde}$  are set to 1 and 1, respectively. The ADAM optimizer is used with an initial learning rate of 0.1, halved every 50 steps. Each inverse problem is solved with 500 gradient updates.

### E.4 The EIT Problem with Operator-based Measurements

#### E.4.1 Problem Setup

The EIT is governed by the elliptic PDE with Dirichlet boundary conditions:

$$\begin{aligned} -\nabla(\gamma(x, y)\nabla u(x, y)) &= 0, & \text{in } \Omega &= [0, 1]^2, \\ u(x, y) &= g(x, y), & \text{in } \partial\Omega, \end{aligned} \quad (28)$$

where  $\gamma > 0$  denotes the unknown conductivity field, assumed to be smooth, and  $g$  is the prescribed Dirichlet voltage. The goal is to recover  $\gamma$  from the DtN map  $\Lambda_\gamma$ , defined as:

$$\Lambda_\gamma[g] : g \longrightarrow \gamma \frac{\partial u}{\partial \vec{n}}|_{\partial\Omega}, \quad (29)$$

which maps the input voltage  $g$  into the current  $\gamma \frac{\partial u}{\partial \vec{n}} = \gamma \nabla u \cdot \vec{n}$  at boundary with  $\vec{n}$  being the unit outward normal vector. In practice, the DtN map is approximated using a finite set of pre-defined voltage-current pair, defined as  $\mathcal{M}_{op} = \left\{ (g_l(\mathbf{X}_m), \Lambda_\gamma[g_l](\mathbf{X}_m)) \right\}_{l=1}^L$ , where  $\mathbf{X}_m = (\mathbf{x}_1, \dots, \mathbf{x}_m)$  denotes sensors on the boundary. In our experiments, we use  $m = 128$  equally spaced sensors along

the four boundaries, as illustrated on the right of Fig. 2(a). To approximate the DtN map, we set  $L = 20$ , with the  $l$ -th input voltage defined as  $\cos(2\pi(x \cos(\theta_l) + y \sin(\theta_l)))$  with  $\theta_l = 2\pi l/20$ . Fig. 2(b) shows examples of three input voltages ( $l = 1, 10, 20$ ), their corresponding boundary currents, and the PDE solutions, with the conductivity field displayed on the left of Fig. 2(a).

Since the EIT problem involves operator-based measurements, the PI-DIONs method is not applicable. Therefore, we only evaluate the proposed IGNO in this problem. For training,  $N = 1000$  conductivity fields are generated from trigonometric functions of the form  $\gamma(x, y) = \sum_{k=1}^K \exp(c_k \sin(k\pi x) \sin(k\pi y))$  with  $K = \text{mod}(\bar{K})$ , where  $\bar{K} \sim U[1, 5]$  and  $c_k \sim U[-1, 1]$ . An example of a sampled conductivity is shown on the left of Fig. 2(a). For each conductivity, the  $L = 20$  input voltages  $g_l$  are used as Dirichlet conditions in (2), leading to  $NL = 20000$  total data samples  $\{\gamma^{(n)}, g^{(n)}\}_{n=1}^{NL}$ . Importantly, the training of IGNO does not require computing the corresponding currents, avoiding expensive PDE simulations.

For the inverse problem, we consider two target scenarios:

- **In-distribution target:**  $\gamma$  is drawn from the same distribution used for training.
- **Out-of-distribution target:**  $\gamma$  is sampled from a shifted distribution with  $\bar{K} \sim U[1, 5]$  and  $c_k \sim U[1, 1.5]$ .

In both cases, operator-based measurements  $\mathcal{M}_{op}$  are generated by solving (2) with FEM and adding Gaussian noise. We consider three noise levels: SNR= 50 (low), SNR= 25 (medium), and SNR= 15 (high).

#### E.4.2 Model Setup

To extract latent representations from input coefficients, a uniform grid  $\Xi_a$  with size  $32 \times 32$  is used, which leads to the coefficient encoder input being  $a = a(\Xi_a)$ . Since this problem involves multiple boundary conditions, a boundary encoder  $E_{\theta_{\beta_2}}$  is needed to encode boundary inputs. Given that only  $L = 20$  boundary conditions are used, this encoder is implemented as a one-hot encoding rather than a trainable network. To enforce boundary conditions automatically, we define a mollifier for the solution decoder output. Specifically, for solution prediction with boundary condition  $g_l$ :

$$u_{pred} = \mathcal{G}_{\theta_u}(\beta) \cdot f + g_l, \quad (30)$$

where  $f(x, y) = \sin(\pi x) \sin(\pi y)$ . This ensures that the predicted solution satisfies the boundary conditions without introducing an explicit boundary loss term.

Below, we provide details of the network architectures:

- **Coefficient Encoder  $E_{\theta_{\beta_1}}$ .** The coefficient encoder  $E_{\theta_{\beta_1}}$  consists of a CNN followed by an FFCN. The CNN has four hidden layers with 64 output channels per layer, kernel size  $(3, 3)$ , and a stride of 2. The subsequent FFCN has two hidden layers of 64 neurons each. SiLU activations are applied to all hidden layers, and Tanh is used at the output.
- **Boundary Encoder  $E_{\theta_{\beta_2}}$ .** To encode the boundary conditions  $\{g_l\}_{l=1}^L$ , we use a one-hot encoding: for boundary  $g_l$ , the latent vector  $\beta_2 = e_l \in \mathbb{R}^{20}$ , where  $e_l$  is the  $l$ -th unit vector. This enables IGNO to handle operator-based measurements with multiple boundary conditions efficiently.
- **Solution Decoder  $\mathcal{G}_{\theta_u}$ .** The solution decoder adopts the MultiONet architecture. Both branch and trunk networks are FFCNs with five hidden layers of 100 neurons per layer. The custom Tanh\_Sin activation is applied to all hidden layers.
- **Coefficient Decoder  $\mathcal{G}_{\theta_a}$ .** The coefficient decoder  $\mathcal{G}_{\theta_a}$  shares the same architecture as  $\mathcal{G}_{\theta_u}$ .
- **The NF model  $F_{\theta_{NF}}$ .** The NF model consists of three flow steps, each parameterized by an FFCN with two hidden layers of 128 neurons and SiLU activations.

#### E.4.3 Latent-space Optimization Details

For In-distribution targets, gradients of the objective  $\mathcal{F}$  with respect to  $\beta_1$  are computed via PyTorch automatic differentiation. Weights are set as  $\rho_{data} = 100$  and  $\rho_{pde} = 1$ . The ADAM optimizer is used with an initial learning rate of 0.01, reduced by half every 25 steps, for a total of 200 updates.

For Out-of-distribution targets, weights are set to  $\rho_{data} = 100$  and  $\rho_{pde} = 0.001$ . The ADAM optimizer is used with an initial learning rate of 0.01, decreased by a factor of 2/3 every 100 steps, with a total of 200 updates.

## F Supplementary Tables

### F.1 Continuous Coefficient Recovery with Solution-based Measurements

#### F.1.1 Table 1: In-distribution case

IGNO achieves 3 to 7 times lower reconstruction errors than the PI-DIONs method across all noise levels for in-distribution continuous coefficient recovery. Even under severe noise (SNR=15 dB), IGNO maintains RMSE below 1.2%, while the PI-DIONs method exceeds 4%.

Table 1: RMSEs obtained by different methods under different noise levels in recovery of **continuous coefficients** with solution-based measurements. (**In-distribution case**)

	SNR = 50	SNR = 25	SNR = 15
IGNO	0.0056	0.0061	0.0115
PI-DIONs	0.0366	0.0380	0.0415

#### F.1.2 Table 2: Out-of-distribution case

For out-of-distribution targets with frequency parameters entirely outside training range, IGNO maintains strong generalization with only modest performance degradation compared to in-distribution cases. The PI-DIONs method shows substantial error increases (4.8-6.4%), while IGNO remains 3 to 5 times more accurate.

Table 2: RMSEs obtained by different methods under different noise levels in recovery of **continuous coefficients** with solution-based measurements. (**Out-of-distribution case**)

	SNR = 50	SNR = 25	SNR = 15
Inv-GenNO	0.0064	0.0113	0.0194
PI-DIONs	0.0476	0.0484	0.0639

### F.2 Piecewise-Constant Coefficient Recovery with Solution-based Measurements

#### F.2.1 Table 3: In-distribution case

The cross-correlation indicator  $I_{corr}$  measures morphological similarity, with values near 1 indicating strong agreement between reconstructed and true phase topologies. IGNO achieves  $I_{corr} > 0.95$  across all noise levels, indicating accurate phase topology recovery. The PI-DIONs method fails with  $I_{corr} \sim 0.72$ , indicating essentially random reconstructions due to the inability to handle undefined gradients at discontinuities.

Table 3: Cross-correlations  $I_{corr}$  obtained by different methods under different noise levels in recovery of **piecewise-constant coefficients** with solution-based measurements. (**In-distribution case**)

	SNR = 50	SNR = 25	SNR = 15
IGNO	0.969	0.960	0.951
PI-DIONs	0.718	0.715	0.726

#### F.2.2 Table 4: Out-of-distribution case

For out-of-distribution piecewise-constant targets with different Gaussian process correlation structures producing more complex phase geometries, IGNO maintains  $I_{corr} > 0.91$  even under high

noise. The PI-DIONs method continues to fail ( $I_{corr} \sim 0.78$ ), demonstrating an inability to generalize for discontinuous inverse problems.

Table 4: Cross-correlations  $I_{corr}$  obtained by different methods under different noise levels in recovery of **piecewise-constant coefficients** with solution-based measurements. (**Out-of-distribution case**)

	SNR = 50	SNR = 25	SNR = 15
IGNO	0.960	0.949	0.914
PI-DIONs	0.778	0.779	0.777

### F.3 The EIT Problem with Operator-based Measurements

#### F.4 Table 5: the EIT problem

For the operator-based EIT problem, IGNO demonstrates robust performance across both distribution types. In-distribution reconstruction achieves sub-0.5% error even under low noise, degrading gracefully to 2.8% at SNR=15 dB. Out-of-distribution targets show only modest error increases (2.2–2.9%), indicating strong extrapolation capabilities. The PI-DIONs method cannot handle operator-based measurements and does not apply to this problem.

Table 5: RMSEs obtained by the proposed IGNO in solving the **EIT** problem in both In-distribution and Out-of-distribution cases under different noise levels.

	SNR = 50	SNR = 25	SNR = 15
In-distribution	0.0044	0.0128	0.0277
Out-of-distribution	0.0219	0.0246	0.0293

## G Supplementary Figures

### G.1 Figure 1: The MultiONet architecture

The MultiONet architecture in Fig. 1 shows branch and trunk networks with multi-layer feature aggregation.

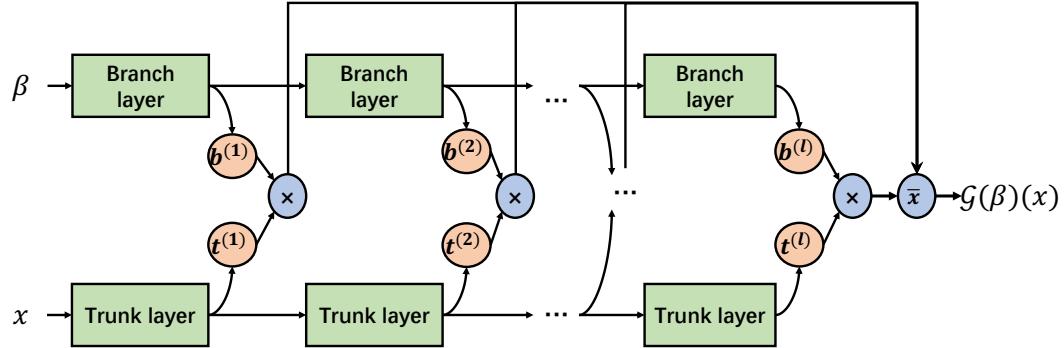


Figure 1: The MultiONet architecture.

### G.2 Figure 2: Illustration of operator-based measurements in the EIT problem

In the EIT problem, the unknown conductivity field  $\gamma$  is inferred from the DtN operator  $\Lambda_\gamma$ , which maps imposed boundary voltages  $g$  to the corresponding boundary currents  $\gamma \frac{\partial u}{\partial n}|_{\partial\Omega}$ . This operator is approximated using  $L = 20$  distinct boundary voltage patterns and measurements collected from

$m = 128$  equally spaced boundary sensors. Figure 2(a) illustrates an example conductivity field  $\gamma$  (left) and the corresponding boundary sensor configuration (right). Figure 2(b) shows three representative boundary conditions  $g_l$ , the resulting current responses  $\Lambda_\gamma[g_l]$ , and the corresponding PDE solutions  $u$  for  $l = 1, 10, 20$ .

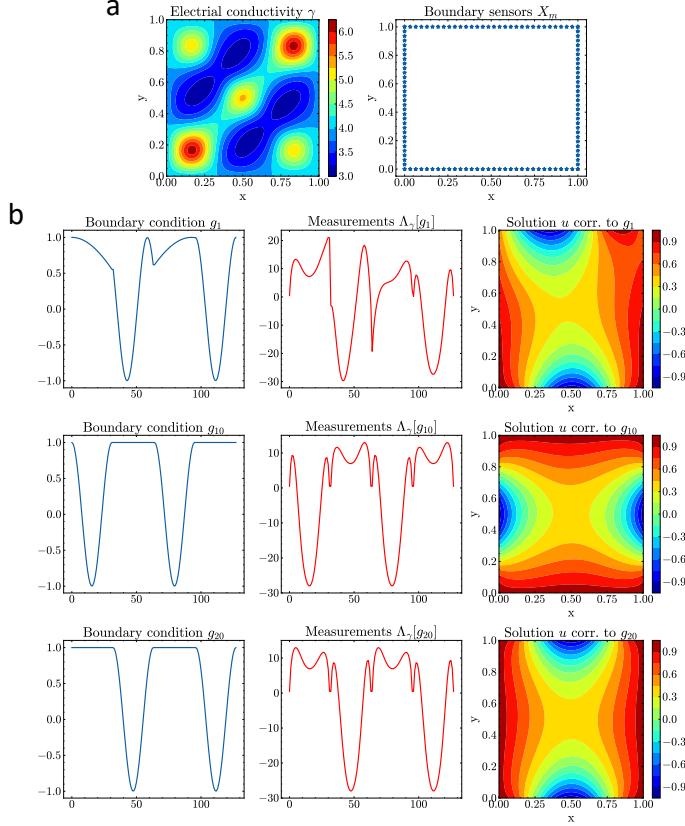


Figure 2: Illustration of operator-based measurements in the EIT problem: (a) An example of conductivity  $\gamma$  (left) and the boundary sensors  $X_m$  (right); (b) Examples of boundary conditions  $g_l$  (left), corresponding current measurements  $\Lambda_\gamma[g_l]$  (middle), and PDE solutions  $u$  (right), when  $l = 1, 10, 20$ .