Graph Neural Regularizers for PDE Inverse Problems

William Lauga
Department of Applied Mathematics
University of Cambridge
wlauga@gmail.com

James Rowbottom
Department of Applied Mathematics
University of Cambridge
jr908@cam.ac.uk

Alexander Denker
Department of Computer Science
University College London
a.denker@ucl.ac.uk

Željko Kereta
Department of Computer Science
University College London
z.kereta@ucl.ac.uk

Moshe Eliasof
Department of Applied Mathematics
University of Cambridge
me532@cam.ac.uk

Carola-Bibiane Schönlieb
Department of Applied Mathematics
University of Cambridge
cbs31@cam.ac.uk

Abstract

We present a framework for solving a broad class of ill-posed inverse problems governed by partial differential equations (PDEs), where the target coefficients of the forward operator are recovered through an iterative regularization scheme that alternates between FEM-based inversion and learned graph neural regularization. The forward problem is numerically solved using the finite element method (FEM), enabling applicability to a wide range of geometries and PDEs. By leveraging the graph structure inherent to FEM discretizations, we employ physics-inspired graph neural networks as learned regularizers, providing a robust, interpretable, and generalizable alternative to standard approaches. Numerical experiments demonstrate that our framework outperforms classical regularization techniques and achieves accurate reconstructions even in highly ill-posed scenarios.

1 Introduction

Inverse problems governed by partial differential equations (PDEs) arise in many scientific fields, including optical tomography [21], seismic imaging [32], and combinatorial problems [7]. In such inverse problems one seeks to recover unknown parameters of a physical system from incomplete or noisy measurements given by the PDE solution. These problems are typically ill-posed, meaning that solutions may not be unique, and can be highly unstable with respect to perturbations in the data.

In contrast, solving the given forward problem (i.e., the PDE) is typically well-posed and can be accurately discretized using finite element methods (FEM). This produces a mesh-based representation of the PDE domain, which can be interpreted as a graph. This graph structure makes it possible to exploit recent advances in graph neural networks (GNNs) to design learning methods that incorporate the physical structure of the PDE, with applications ranging from medical imaging [34] to geophysics [31]. In this work, we present a framework that leverages physics-inspired GNNs [4, 9] to learn a graph neural regularizer [11] (as a data-driven functional) applied to PDE inverse problems. Our approach directly operates on the FEM mesh, aligning with the numerical discretization of the forward operator. In contrast, black-box methods [29, 19] learn mappings directly from data; while powerful,

1st Workshop on Differentiable Systems and Scientific Machine Learning @ EurIPS 2025

they often lack interpretability and are usually limited to pixel-grid scenarios. We discuss related work in Appendix A. The main contributions of this work are:

- **Graph-based regularization for PDE inverse problems.** We propose a framework that learns *graph neural regularizers* directly on FEM meshes, exploiting their natural graph structure without grid interpolation, and allowing for transfer to irregular domains.
- **Physics-inspired variational formulation.** We embed physics-based GNNs as *learned regularizers* within a classical variational inversion scheme, ensuring interpretability and physical consistency.
- **Comprehensive empirical validation.** We demonstrate the effectiveness of our framework by applying it to three PDE inverse problems, and show that it can significantly outperform classical methods and is competitive with state-of-the-art black-box methods, even with far fewer parameters.

2 Solving PDE Inverse Problems with Graph Neural Regularizers

We focus on inverse problems governed by a PDE in a domain D, for a source term h, written as

$$\mathcal{L}(u; x) = h \quad \text{in } D, \tag{1}$$

with appropriate boundary conditions. The operator $\mathcal{L}$ depends on system parameters x. The goal is to recover parameters x from noisy and incomplete observations of the solution u.

Let B be some observation operator. Then, given observations $y = B(u)$ the inverse problem can be formulated as: recover x such that $B(\mathcal{G}^\dagger(x)) \approx y$, where $\mathcal{G}^\dagger : x \mapsto u$ is the forward operator solving the PDE for u given parameters x. A standard approach for solving this inverse problem is to consider the following variational problem

$$\hat{x} = \arg\min_x \frac{1}{2}\|B(\mathcal{G}^\dagger(x)) - y\|^2 + \alpha\mathcal{R}(x), \tag{2}$$

which balances a data-fidelity term with a regularization function $\mathcal{R}(x)$ [1]. A common classical approach that encourages smoothness is Laplacian regularization, given by $\mathcal{R}(x) = \frac{1}{2}x^T Lx$, where L is the unnormalized graph Laplacian matrix [26]. Instead of using a hand-crafted model, we aim to implement the regularizer as a neural network and learn its parameters. One approach to training the regularizer is bilevel learning [12, 20], which adapts the parameters of the network such that the reconstruction in Equation (2) minimizes some loss.

We follow the GRIP framework [8, 11] and consider a lifted version of Equation (2), i.e.,

$$\arg\min_{x_1,\dots,x_L} \frac{1}{2}\|B(\mathcal{G}^\dagger(x_L)) - y\|^2 + \underbrace{\sum_{l=1}^{L-1} \frac{1}{2}\|x_{l+1} - x_l\|^2}_{\text{kinetic}} + \underbrace{\sum_{l=1}^{L-1} \Phi(x_l, f_M, \mathcal{G}, \theta_l)}_{\text{potential}}, \tag{3}$$

with $\hat{x} = x_L$ as the reconstruction and the regularizer given by the kinetic and potential terms above. The potential is given by Φ and depends on the graph $\mathcal{G}$, meta-data f_M and trainable parameters θ_l for $l = 1, \dots, L-1$. Given a paired dataset $\{x^{(i)}, y^{(i)}\}_{i=1}^n$ we consider the mean-squared-error loss

$$\min_\theta \sum_{i=1}^n \|\hat{x}(y^{(i)}) - x^{(i)}\|^2, \tag{4}$$

where $\hat{x}(y^{(i)})$ is the reconstruction given by the lifted formulation in Equation (3). For optimization of the parameters, we make use of *unrolling* [24]. Here, we use an iterative method to optimize Equation (3) for a fixed number of steps and rely on backpropagation to obtain the gradients. In particular, differentiating the regularizer with respect to $x = (x_1, ..., x_L)$ gives the update rule

$$x_{l+1} = 2x_l - x_{l-1} + \nabla_{x_l}\Phi(x_l, f_M, \mathcal{G}, \theta_l), \tag{5}$$

where we directly parametrize the gradient $\nabla_{x_l}\Phi$ as a GNN layer [1]. We then solve Equation (3) by alternating between regularization steps obtained via a GNN, as in Equation (5), and minimizing the data-fidelity term, using Conjugate Gradient Least-Squares (CGLS). For the meta-data f_M we include a positional encodings, see Appendix C.1 and Algorithms 1, 2.

[1] In practice, we get better results by using the simpler residual update $x_{l+1} = x_l + \nabla_{x_l}\Phi(x_l, f_M, \mathcal{G}, \theta_l)$.

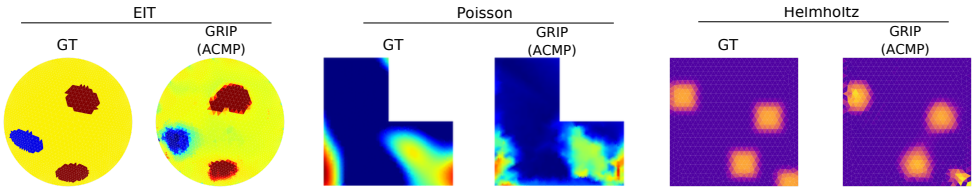

Figure 1: Reconstruction examples for: EIT, Poisson (Sparse), Inverse Wave Scattering (Helmholtz)

2.1 GNN Architectures

The connection between GNNs and differential equations is well understood [14]. Using physics-inspired GNN architectures ensures alignment between the model and the problem, providing a strong rationale for the approach, as we are applying this framework to physical scenarios.

Graph Neural Diffusion (GRAND). GRAND [4] is an approach that treats GNNs as discretizations of a PDE; it can be seen as a learnable graph-diffusion process. The transformer attention allows it to learn an anisotropic diffusion equation so node features evolve into community patterns, similar to many of the underlying distributions we are trying to recover, thus serving as a useful learnable prior

Allen-Cahn Message Passing (ACMP). The ACMP architecture [33] models the GNN as a particle system with interactive forces and the Allen-Cahn double-well potential. ACMP is an example of a more general class of reaction-diffusion GNNs [5, 10]. ACMP was chosen as a GNN as the double-well encourages node clustering, which is desirable if the target data is known to be, for example, piecewise constant. Our experiments confirm this useful property as ACMP strongly outperforms GRAND in piecewise-constant scenarios, but performance is more similar in the smoother scenarios. More details on the architectures of GRAND and ACMP can be found in Appendix C.

3 Experiments and Results

We apply our framework to three representative PDE inverse problems with different geometries to assess its effectiveness and generality. The goal is to evaluate whether our approach can (i) outperform standard GNN baselines, (ii) remain competitive with or even outperform state-of-the-art deep learning methods while using orders of magnitude fewer parameters and less training data, and (iii) provide better interpretability. Moreover, we seek to demonstrate that the framework naturally handles PDEs on irregular meshes without requiring interpolation, making it well-suited to practical inverse problem settings.

We compare against two standard fully-learned baselines: (i) a Graph Convolutional Network (GCN) [18] and (ii) a U-Net [29]. Note, that the U-Net requires an final interpolation step from a structured regular grid to the mesh. Some image comparisons are shown in 3, with more images in the appendix.

Poisson. We consider steady-state diffusion modeled by Poisson's equation $\nabla^2 u = a$ with boundary condition $u|_{\partial D} = 0$. The corresponding inverse problem is to reconstruct the source a from partial measurements of the concentration u. We consider two setups 'dense' and 'sparse', depending on the number of observations of u. Results are summarized in Table 1 and comparison plots in AppendixB.

Inverse Wave Scattering. We seek to infer the material properties of a medium by wave propagation in the frequency domain as described in [23]. The governing PDE is the Helmholtz equation

$$-\Delta u - \omega^2 a u = 0, \quad \text{in } D, \text{ and } u = g, \quad \text{on } \partial D, \tag{6}$$

where ω is a frequency and we have the Dirichlet boundary condition $g \in H^{\frac{1}{2}}(\partial D)$. We observe data on the boundary ∂D with a Dirichlet-to-Neumann map, and we try to solve the linearized version of the problem. Details of the linearization and of the boundary conditions are given in Appendix B. We study this problem, whose results are summarized in Table 2, on the square domain $D = [0, 1]^2$. The coefficient a is sampled from a distribution containing 1-4 square-shaped inclusions that are randomly distributed across the domain.

Electrical Impedance Tomography. EIT is a noninvasive imaging technique that uses electrical measurements taken on the boundary ∂D of an object D to reconstruct the conductivity of its interior [2]. Let $U_{l=1}^{L} e_l$ be the set of L electrodes attached to the boundary ∂D. The potential u is given by

$$-\nabla \cdot (\sigma \nabla u) = 0 \quad \text{in} \quad D, \tag{7}$$

Table 1: Results for the Poisson PDE ($\pm$ absolute standard deviation). We considered a dense sampling setup with 60% of vertices observed, and a sparse setup with 10% of vertices observed. We report the MSE between reconstructed and ground-truth coefficients, and the data-fit measuring agreement with observed data.

	Poisson (Dense)		Poisson (Sparse)	
Method	MSE (↓)	Data-fit (↓)	MSE (↓)	Data-fit (↓)
Laplacian	0.1100 ($\pm$0.035)	0.0023 ($\pm$0.001)	0.2201 ($\pm$0.074)	0.0067 ($\pm$0.007)
U-Net	0.1220 ($\pm$0.110)	0.0111 ($\pm$0.013)	0.1587 ($\pm$0.100)	0.0905 ($\pm$0.032)
GCN	0.0646 ($\pm$0.031)	0.0155 ($\pm$0.024)	0.1879 ($\pm$0.130)	0.1207 ($\pm$0.110)
Our framework				
GRIP (GRAND)	0.0647 ($\pm$0.025)	0.0259 ($\pm$0.050)	0.1441 ($\pm$0.066)	0.0835 ($\pm$0.020)
GRIP (ACMP)	0.0586 ($\pm$0.023)	0.0108 ($\pm$0.013)	0.1443 ($\pm$0.063)	0.0826 ($\pm$0.036)

where σ is the conductivity distribution. Realistic boundary conditions are given by the Complete Electrode Model (CEM), as described in [30]. The goal is to reconstruct σ given electrode measurements. We linearize the forward operator around a constant background conductivity. Results are reported in Table 2. Here, we also compare to a state-of-the-art U-Net from [6], which has about 10^7 parameters, whereas the GNNs in our framework have < 1,500 parameters.

Table 2: Results for Inverse Wave Scattering and EIT ($\pm$ absolute standard deviation). We report the MSE between reconstructed and ground-truth coefficients/conductivity, and the data-fit measuring agreement with observed boundary or electrode data.

	Inverse Wave Scattering		EIT	
Method	MSE (↓)	Data-fit (↓)	MSE (↓)	Data-fit (↓)
Laplacian	0.3700 ($\pm$0.083)	0.0047 ($\pm$0.027)	0.6967 ($\pm$0.160)	0.0675 ($\pm$0.091)
U-Net	0.0730 ($\pm$0.091)	0.0506 ($\pm$0.023)	0.3009 ($\pm$0.120)	0.0974 ($\pm$0.076)
GCN	0.2402 ($\pm$0.065)	0.0601 ($\pm$0.023)	0.3510 ($\pm$0.065)	0.3584 ($\pm$0.100)
Our framework				
GRIP (GRAND)	0.0953 ($\pm$0.036)	0.0022 ($\pm$0.001)	0.3887 ($\pm$0.16)	0.2229 ($\pm$0.51)
GRIP (ACMP)	0.0769 ($\pm$0.047)	0.0020 ($\pm$0.001)	0.3471 ($\pm$0.16)	0.2199 ($\pm$0.53)

4 Conclusion

We introduced a general framework for solving PDE-based inverse problems by learning graph-based regularizers with physics-inspired GNN architectures that exploit the natural graph structure of FEM discretizations. Across three representative PDE inverse problems we demonstrated that our framework consistently outperforms classical variational regularization methods and achieves performance competitive with state-of-the-art black-box deep learning approaches, occasionally even outperforming them as in the Poisson PDE example.

An important advantage of our method is that it preserves interpretability by incorporating the regularizer into a variational framework. The forward operator and data-fidelity term remain explicit, unlike purely supervised mappings such as U-Net. In contrast to most convolutional neural networks, such as the U-Net, our GNN-based regularizer operates natively on meshes and can be flexibly combined with iterative solvers such as CGLS, without intermediate interpolation steps. Further work will include combining our approach with Gauss-Newton iterations, to mitigate linearization errors, and extending it to more complex geometries.

Acknowledgements - We gratefully acknowledge support from the EPSRC Programme Grant The Mathematics of Deep Learning (EP/V026259/1), which supported the research activities and travel associated with this work, and was instrumental in facilitating the collaboration among the authors. ZK acknowledges support from the EPSRC (EP/X010740/1).

References

[1] *Variational Methods in Imaging*, volume 167 of *Applied Mathematical Sciences*. Springer New York, New York, NY, 2009. ISSN: 0066-5452.

[2] L. Borcea. Electrical impedance tomography. *Inverse Problems*, 18(6):R99–R136, Dec. 2002.

[3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer New York, New York, NY, 2008.

[4] B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, and E. Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pages 1407–1418. PMLR, 2021.

[5] J. Choi, S. Hong, N. Park, and S.-B. Cho. Gread: Graph neural reaction-diffusion networks. In *International conference on machine learning*, pages 5722–5747. PMLR, 2023.

[6] A. Denker, F. Margotti, J. Ning, K. Knudsen, D. N. Tanyu, B. Jin, A. Hauptmann, and P. Maass. Deep Learning Based Reconstruction Methods for Electrical Impedance Tomography, Aug. 2025. arXiv:2508.06281 [eess].

[7] M. Eliasof and E. Haber. Graph neural networks for binary programming. In *28th European Conference on Artificial Intelligence, ECAI 2025*. IOS Press BV, 2025.

[8] M. Eliasof, E. Haber, and E. Treister. DRIP: Deep regularizers for inverse problems.

[9] M. Eliasof, E. Haber, and E. Treister. Pde-gcn: Novel architectures for graph neural networks motivated by partial differential equations. *Advances in neural information processing systems*, 34:3836–3849, 2021.

[10] M. Eliasof, E. Haber, and E. Treister. Graph Neural Reaction Diffusion Models. *SIAM Journal on Scientific Computing*, 46(4):C399–C420, Aug. 2024.

[11] M. Eliasof, M. S. R. Siddiqui, C.-B. Schönlieb, and E. Haber. Learning Regularization for Graph Inverse Problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16):16471–16479, Apr. 2025.

[12] E. Haber and L. Tenorio. Learning regularization functionals a supervised training approach. *Inverse Problems*, 19(3):611–626, June 2003.

[13] A. Habring and M. Holler. Neural-network-based regularization methods for inverse problems in imaging. *GAMM-Mitteilungen*, 47(4):e202470004, Nov. 2024.

[14] A. Han, D. Shi, L. Lin, and J. Gao. From continuous dynamics to graph neural networks: Neural diffusion and beyond.

[15] J. Hertrich, H. S. Wong, A. Denker, S. Ducotterd, Z. Fang, M. Haltmeier, Z. Kereta, E. Kobler, O. Leong, M. S. Salehi, C.-B. Schönlieb, J. Schwab, Z. Shumaylov, J. Sulam, G. S. Wache, M. Zach, Y. Zhang, M. J. Ehrhardt, and S. Neumayer. Learning Regularization Functionals for Inverse Problems: A Comparative Study, Oct. 2025. arXiv:2510.01755 [cs].

[16] W. Herzberg, A. Hauptmann, and S. J. Hamilton. Domain independent post-processing with graph U-nets: applications to electrical impedance tomographic imaging. *Physiological Measurement*, 44(12):125008, Dec. 2023. Publisher: IOP Publishing.

[17] W. Herzberg, D. B. Rowe, A. Hauptmann, and S. J. Hamilton. Graph Convolutional Networks for Model-Based Learning in Nonlinear Inverse Problems. *IEEE Transactions on Computational Imaging*, 7:1341–1353, 2021.

[18] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[20] K. Kunisch and T. Pock. A Bilevel Optimization Approach for Parameter Learning in Variational Models. *SIAM Journal on Imaging Sciences*, 6(2):938–983, Jan. 2013.

[21] R.-Y. Lai, Q. Li, and G. Uhlmann. Inverse problems for the stationary transport equation in the diffusion scaling.

[22] M. Mardani, H. Monajemi, V. Papyan, S. Vasanawala, D. Donoho, and J. Pauly. Recurrent generative adversarial networks for proximal learning and automated compressive image recovery.

[23] R. Molinaro, Y. Yang, B. Engquist, and S. Mishra. Neural Inverse Operators for Solving PDE Inverse Problems.

[24] V. Monga, Y. Li, and Y. C. Eldar. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. *IEEE Signal Processing Magazine*, 38(2):18–44, Mar. 2021.

[25] S. Mukherjee, A. Hauptmann, O. Oktem, M. Pereyra, and C.-B. Schonlieb. Learned Reconstruction Methods With Convergence Guarantees: A survey of concepts and applications. *IEEE Signal Processing Magazine*, 40(1):164–182, Jan. 2023.

[26] B. Nadler, N. Srebro, and X. Zhou. Statistical analysis of semi-supervised learning: The limit of infinite unlabelled data. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc.

[27] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural Operator: Learning Maps Between Function Spaces With Applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[28] N. Polydorides and W. R. B. Lionheart. A Matlab toolkit for three-dimensional electrical impedance tomography: a contribution to the Electrical Impedance and Diffuse Optical Reconstruction Software project. *Measurement Science and Technology*, 13(12):1871–1883, Dec. 2002.

[29] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing.

[30] E. Somersalo, M. Cheney, and D. Isaacson. Existence and Uniqueness for Electrode Models for Electric Current Computed Tomography. *SIAM Journal on Applied Mathematics*, 52(4):1023–1040, Aug. 1992.

[31] M. P. van den Ende and J.-P. Ampuero. Automated seismic source characterization using deep graph neural networks. *Geophysical Research Letters*, 47(17):e2020GL088690, 2020.

[32] D. J. Verschuur and A. J. Berkhout. Seismic data analysis in the forward and inverse data space. Publisher: Society of Exploration Geophysicists.

[33] Y. Wang, K. Yi, X. Liu, Y. G. Wang, and S. Jin. ACMP: Allen-cahn message passing with attractive and repulsive forces for graph neural networks.

[34] L. Zhang, Y. Zhao, T. Che, S. Li, and X. Wang. Graph neural networks for image-guided disease diagnosis: A review. *Iradiology*, 1(2):151–166, 2023.

[35] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. *International Conference on Learning Representations*, 2021.

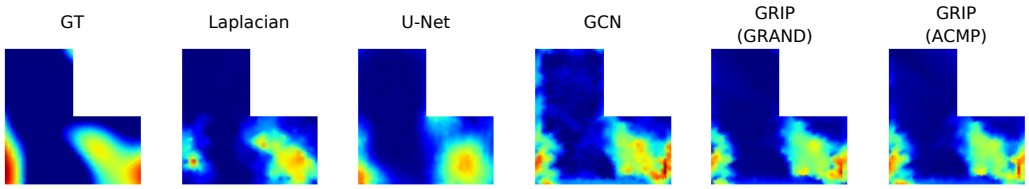

Figure 2: Poisson (sparse) reconstruction examples

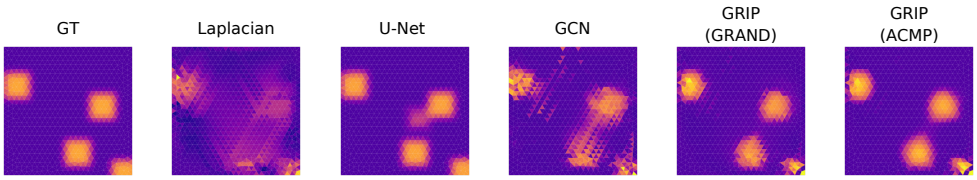

Figure 3: Inverse Wave Scattering reconstruction examples

A Related Work

On Learned Regularizers. Recent studies have explored learning the regularizer $\mathcal{R}$ from data, see e.g. [13, 25, 22, 15]. These data-driven regularizers often demonstrate improved performance compared to traditional handcrafted penalties. More recently, [11] introduced a graph-based learned regularization framework, demonstrating that incorporating structural relationships enables the method to generalize effectively to more complex and irregular data topologies.

On Neural Operators. Neural operators aim to learn mappings between function spaces [27]. There are two main design philosophies: (i) transforming functions into a different basis, such as in the Fourier Neural Operator [35], and learning the mapping there, or (ii) operating directly on irregular domains through graph-based representations. We make use of the latter approach and implement the mapping using GNNs. While GNNs have been used in EIT in the literature [16, 17], these works rely on supervised learning, directly learning a mapping from measurements to reconstructions. In contrast, we learn a regularizer to be used in the variational framework.

B PDE Descriptions and Details

This section provides additional details on the PDE inverse problems considered in the experiments. We solve the PDEs using FEM, see [3] for a mathematical overview. In particular, we employ a triangulation with N nodes of the domain D. This triangulation defines a graph structure, where each node defines a vertex and we add an edge whenever two nodes are connected. Further, this triangulation defines a finite-dimensional subspace as the set of piecewise linear functions restricted to the triangulation.

Linearization. To simplify training, we solve the linearized inverse problems by assuming a known reference coefficient a_{ref} and attempt to reconstruct a perturbation ∂a, yielding

$$\tilde{F}(a_{\text{ref}} + \partial a; a_{\text{ref}}) \approx F(a_{\text{ref}}) + J_{a_{\text{ref}}}(\partial a), \tag{8}$$

where $J_{a_{\text{ref}}} = \nabla F(a_{\text{ref}})$ is the Jacobian of the forward map.

B.1 Poisson Equation

Data generation. Ground-truth source fields $a(x)$ were randomly sampled at mesh vertices and smoothed using a Gaussian kernel. Forward solutions were obtained via finite element discretization on an L-shaped domain. The L-shaped mesh used had 1990 elements. Two setups were used:

- **Dense:** 60% of vertices observed.
- **Sparse:** 10% of vertices observed.

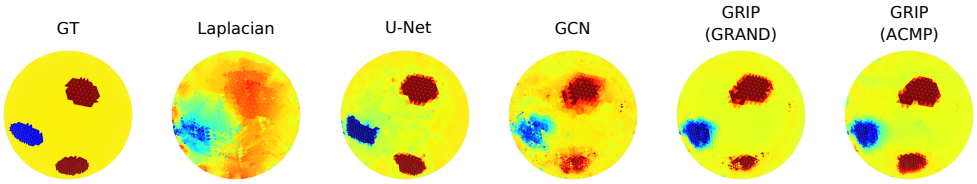

Figure 4: EIT reconstruction examples

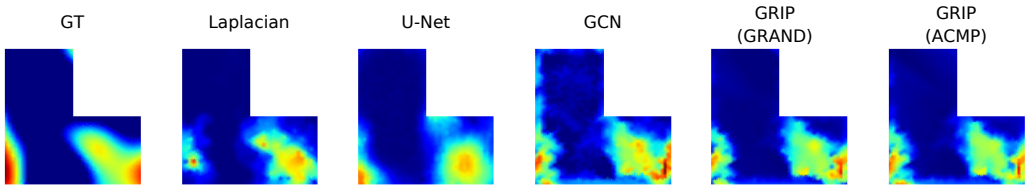

Figure 5: Poisson (sparse) reconstruction examples

FEM and weak formulation. For the Poisson PDE the weak formulation reads

$$\text{Find } u \in H_0^1(D) \text{ such that } \int_D \nabla u \cdot \nabla v dx = -\int_D a v dx, \quad \forall v \in H_0^1(D).$$

Let $\{\phi_j\}_{j=1}^N$ be the FEM basis of piecewise linear hat functions. If we approximate

$$u(x) \approx \sum_{j=1}^N u_j \phi_j(x), \quad a(x) \approx \sum_{j=1}^N a_j \phi_j(x),$$

we obtain a linear system for the coefficients $\mathbf{u} = (u_1, \ldots, u_N)$ as

$$K\mathbf{u} = -M\mathbf{a},$$

with $\mathbf{a} = (a_1, \ldots, a_N)$, stiffness matrix $K \in \mathbb{R}^{N \times N}$ and mass matrix $M \in \mathbb{R}^{N \times N}$ defined by

$$K_{i,j} = \int_D \nabla \phi_j \cdot \nabla \phi_j dx, \quad M_{i,j} = \int_D \phi_i \phi_j dx.$$

Note that the linear system in linear in $\mathbf{a}$, so we are able to directly use CGLS to approximate the solution.

B.2 Inverse Wave Scattering (Helmholtz Equation)

Boundary conditions We observe data on the boundary ∂D with the Dirichlet-to-Neumann map,

$$\Lambda_a : H^{1/2}(\partial D) \longrightarrow H^{-1/2}(\partial D), \tag{9}$$

$$\Lambda_a[g] = \left.\frac{\partial u}{\partial \nu}\right|_{\partial D}, \quad \forall g \in H^{1/2}(\partial D). \tag{10}$$

Our goal is to reconstruct the wave coefficient a from the map Λ_a.

Coefficient distribution. The coefficients are defined as mixtures of fourth-order Gaussian-like inclusions

$$a(x, y) = \sum_{k=1}^m \exp\big(- c(x - c_{1,k})^4 - c(y - c_{2,k})^4 \big), \tag{11}$$

with $m \in [1, 4]$ and $c = 2 \times 10^4/3$.

The distribution models a homogeneous medium featuring 1–4 square inclusions randomly positioned within the domain. For each sampled coefficient field, 20 Dirichlet boundary conditions are imposed with trigonometric boundary data. The mesh used here had 1578 elements.

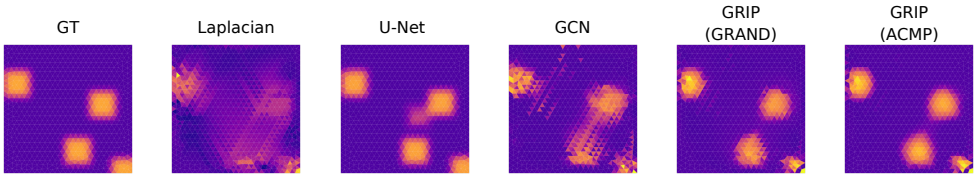

Figure 6: Inverse Wave Scattering reconstruction examples

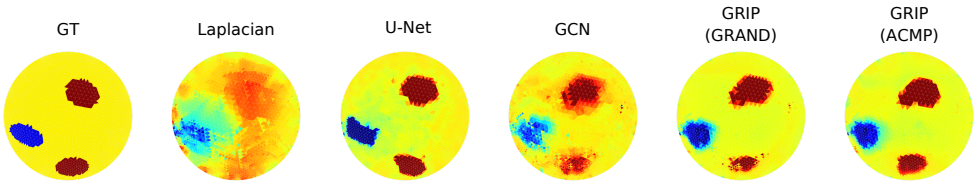

Figure 7: EIT reconstruction examples

FEM and weak formulation. The weak formulation for the Helmholtz equation reads

$$\text{Find } u \in H^1(D) \text{ such that } \int_D \nabla u \cdot \nabla v dx - \omega^2 \int_D auv dx = 0, \quad \forall v \in H_0^1(D),$$

with $u_{|\partial D} = g$. Further, we assume that both ω and a are real valued.

B.3 Electrical Impedance Tomography

Dataset generation We used synthetically-generated data on a circular domain, where each datapoint has 1-3 ellipses with a random constant conductivity. The mesh used for the EIT discretization had 4984 elements.

FEM and weak formulation. We employ the FEM method described in [6], which augments the standard FEM system to enforce conservation of charge and grounding condition, i.e., the sum of the voltages measured at the electrodes e_l has to sum to zero. We compute the Jacobian used in the linearization by the method described in [28] to reduce the computational cost.

C Model Architecture, Hyperparameters, and Training

C.1 On Meta-Data for PDE Inverse Problems.

In [11] it is shown that including meta-data, denoted f_M, can drastically improve model performance. Hence, we concatenate the graph positional-encodings and then apply a 2-layer multi-layer perceptron (MLP) before the regularization steps.

C.2 GNN Architectures

GRAND (Graph Neural Diffusion). GRAND treats message passing as a diffusion process. If X are some embedded node features, GRAND tries to solve:

$$X(T) = X(0) + \int_0^T \frac{\partial X(t)}{\partial t} dt, \tag{12}$$

where

$$\frac{\partial X_i(t)}{\partial t} = \sum_{j:(i,j)\in E_0} a\big(X_i(t), X_j(t)\big)\big(X_j(t) - X_i(t)\big), \tag{13}$$

where $a(\cdot,\cdot)$ is a learnable attention function. We use the linear variant (GRAND-l) for computational efficiency. Implicit Euler time discretization is applied.

Algorithm 1 Training the Graph Neural Regularizer

Require: Forward operator $\mathbf{G}^\dagger$, dataset $\{(x, y)\}$, number of iterations N_{max}

```
while not converged do
    for (x,y) in dataset do
        z = 0
        for k = 1 to N_max do
            Run CGLS steps on z
            Concatenate z with positional encodings
            Apply 2-layer MLP
            Run GNN_θ on z
        end for
        unembed z
        Compute loss of z vs. x as in Equation (4)
        Take gradient step w.r.t. (θ)
    end for
end while
return Fitted parameters (θ)
```

Algorithm 2 Reconstruction

Require: Forward operator $\mathbf{G}^\dagger$, observation y, number of iterations N_{max}

```
z = 0
for k = 1 to N_max do
    Run CGLS steps on z
    Concatenate z with positional encodings
    Apply 2-layer MLP
    Run GNN_θ on z
end for
unembed z
return z (the reconstruction)
```

ACMP (Allen–Cahn Message Passing). ACMP models message passing as a reaction–diffusion system:

$$\frac{\partial}{\partial t} x_i(t) = \alpha \odot \sum_{j \in N_i} \big(a(x_i(t), x_j(t)) - \beta\big)\big(x_j(t) - x_i(t)\big) + \delta \odot x_i(t) \odot \big(1 - x_i(t) \odot x_i(t)\big), \quad (14)$$

where β models attractive/repulsive forces between particles, and $x_i(t) \odot \big(1 - x_i(t) \odot x_i(t)\big)$ is the Allen-Cahn double-well potential. Setting β and δ to 0 reduces ACMP to GRAND. In our experiments we set β to 0 as we are only after the power of the double-well, and α and δ are learnable parameters.

For both GRAND and ACMP we use 32 layers.

C.3 Training Procedure

The training and reconstruction algorithms are given by Algorithms 1 and 2. We use 15 unrolling steps, with $15 - 30$ CGLS iterations per step. Models were trained for 100 to 400 epochs with early stopping. For training, we used 100 to 200 data points for the GNNs and 500 for the U-Net. Adam optimizer was used with a learning rate of 10^{-3}. We evaluated the model on a held-out set of 100 test data points. The U-Nets had approximately 10^7 parameters, whereas the GNNs had fewer than 1500.