Graph Convolutional Networks for Model-Based Learning in Nonlinear Inverse Problems

William Herzberg, Daniel B. Rowe, Andreas Hauptmann, and Sarah J. Hamilton

Abstract—The majority of model-based learned image reconstruction methods in medical imaging have been limited to uniform domains, such as pixelated images. If the underlying model is solved on nonuniform meshes, arising from a finite element method typical for nonlinear inverse problems, interpolation and embeddings are needed. To overcome this, we present a flexible framework to extend model-based learning directly to nonuniform meshes, by interpreting the mesh as a graph and formulating our network architectures using graph convolutional neural networks. This gives rise to the proposed iterative Graph Convolutional Newton's Method (GCNM), which directly includes the forward model into the solution of the inverse problem, while all updates are directly computed by the network on the problem specific mesh. We present results for Electrical Impedance Tomography, a severely ill-posed nonlinear inverse problem that is frequently solved via optimization-based methods, where the forward problem is solved by finite element methods. Results for absolute EIT imaging are compared to standard iterative methods as well as a graph residual network. We show that the GCNM has strong generalizability to different domain shapes, out of distribution data as well as experimental data, from purely simulated training data.

Index Terms—conductivity, electrical impedance tomography, finite element method, graph convolutional networks, model-based deep learning,

I. INTRODUCTION

MANY tomographic image reconstruction tasks fit under the umbrella of *inverse problems* as they seek to recover an image x from indirect measurements y. Further, these are typically ill-posed which means that finding a unique solution in a stable manner ranges from difficult to impossible without the use of prior knowledge about the problem. With the rise of deep learning, classical reconstruction techniques have been paired with, or replaced by, learned methods that can increase stability and promote uniqueness by enforcing strong data-driven priors. Specifically, supervised learning is still the most common approach for medical image reconstruction tasks. A data set of known ground truth images and measurement input pairs are used to train a network that is later used to compute

predictions from new inputs. When the forward operator $\mathcal{T}$ for the imaging task is known, a set of training data can be simulated, which is especially useful when a large number of experimental training samples are not available.

Learned image reconstruction can be broken down further into three main categories [1], [2]. The first, and most common, uses an analytically known inversion operator $\mathcal{T}^{\dagger}$ to obtain an approximate reconstruction that can suffer from noise or other artefacts such as distortions. Then, a neural network Λ_θ can be used, with the approximate solution as the input, to improve this initial reconstruction and obtain a cleaned version as $\hat{x} = \Lambda_\theta\left(\mathcal{T}^{\dagger}y\right)$, [3], [4]. The second category is commonly referred to as iterative *model-based* techniques (or unrolled methods) because information about the known forward model $\mathcal{T}$ is intertwined with learned components in an iterative manner. At each such iterative step, a neural network Λ_{θ_k} computes an updated reconstruction that may use a combination of the current iterate, measurements, and forward model, as inputs: $x_{k+1} = \Lambda_{\theta_k}\left(x_k, y, \mathcal{T}\right)$ [5], [6]. A third category might use a neural network to compute the image directly from the measurements [7]. This approach is far less common in imaging tasks, due to training instabilities, the need for large data, and limited generalization capabilities with respect to changes in the measurement setup [8].

The majority of learned image reconstruction tasks use convolutional neural networks (CNNs), which are especially favorable for imaging applications due to their translational invariance and capability to leverage local dependencies and structures. Applications in medical imaging have been limited to pixel/voxel grids, but for many nonlinear inverse problems, the forward model is solved using the finite element method (FEM) which requires discretizing the domain onto special meshes. These meshes often have triangular elements and are very irregular, unlike the pixel grids needed for the application of a CNN. To incorporate CNNs into imaging tasks where the image is initially defined over such meshes, one needs to perform an interpolation step, or equivalent, to convert mesh data to pixel-grid data and embed it into a rectangular domain. Consequently, it would be more natural to perform the learned image reconstruction directly on the domain and geometry defined by the FEM mesh.

To achieve this, the data defined over a FEM mesh is considered as graph data so that graph convolutional networks (GCNs) can be used as an alternative to the traditional CNN in a model-based learned approach for solving nonlinear inverse problems. In the simplest form, graph data is composed of nodes connected by edges. In this work, non-directional, unweighted, homogeneous graphs are considered. GCNs were

designed specifically for graph data and are intended to leverage many of the same benefits as traditional CNNs; shared weights, translational invariance, and localization. Here, we apply GCNs to FEM mesh data in the context of learned image reconstruction, opening a new avenue to directly work on the problem specific meshes.

The proposed method, dubbed the Graph Convolutional Newton's Method (GCNM), utilizes the model-based approach by incorporating GCNs into traditional iterative methods for solving nonlinear inverse problems, namely Newton methods, which specifically use FEM meshes while solving the forward problem. This enables us to leverage the advantages offered by convolutional networks, while being able to operate directly on the FEM meshes used in the forward problem. This eliminates the need to convert between meshes and pixel grids, and work more naturally on the underlying domain geometry. Furthermore, using the forward model at each iteration reaps many of the advantages seen in other model-based methods. In this work we show that the GCNM has strong fitting abilities, requires fewer iterations than classical optimization-based methods, and can generalize well to new domain shapes, FEM meshes, and noise patterns. We will compare our approach to classical optimization based methods, as well as a post-processing ResNet with graph convolutional layers (GResNet), which will showcase the importance of repeated incorporation of model information at each iteration.

In the following, we first review electrical impedance tomography (EIT), a highly ill-posed nonlinear inverse problem, which will serve as test case in this study for the proposed GCNM. Section III introduces the GCNM as well as the post-processing GResNet used here. Section IV describes the examples considered, training data used, and evaluation metrics that will be used to assess the results. Results for simulated and experimental data are presented in Section V, and conclusions are drawn in Section VI.

II. ELECTRICAL IMPEDANCE TOMOGRAPHY

EIT is an imaging modality that uses electrodes attached to the surface of a domain to inject current and measure the resulting electrical potential. The electrical measurements are used to recover the conductivity distribution σ inside the domain. The physical problem in EIT can be modeled via the *conductivity equation*

$$\nabla \cdot \sigma \nabla u = 0, \tag{1}$$

where u denotes the potential and $0 < \sigma < \infty$. The recovery of the internal conductivity from surface electrical measurements is a severely ill-posed, nonlinear inverse problem which requires carefully designed numerical solvers robust to noise and modeling errors.

Most commonly, the reconstruction problem in EIT is formulated in a variational setting and solved by iterative methods that minimize the error between measured voltages V and simulated voltages $U(\sigma)$ corresponding to a guess conductivity σ. Additional regularization is needed due to the ill-posedness and instability of the reconstruction problem; popular approaches include the Levenberg-Marquardt (LM) algorithm, Tikhonov regularization [9], and Total Variation (TV)

regularization [10]. These methods frequently suffer from high sensitivity to modeling errors and, if not compensated for, are limited in practice to *time-difference* EIT imaging which recovers the change in conductivity $\Delta\sigma$ relative to a reference data set/frame. Applications of time-difference EIT include monitoring heart and lung function of hospitalized patients, and kidney-reflux. *Absolute*, also called *static*, EIT imaging recovers the conductivity at the time of measurement from a single frame of EIT data V. Whereas in time-difference imaging the modeling errors can cancel out, in absolute imaging they do not and hence require a correction [11], [12]. Developing fast robust image reconstruction algorithms for absolute EIT imaging is important for applications such as breast cancer imaging, stroke classification, and nondestructive evaluation where a pre-injury set of measurements is unavailable; see [13], [14] for a literature review of applications.

Alternatively, direct (non-iterative) reconstruction methods such as the D-bar method [15]–[18], and Calderón's method [19], [20] show promise for fast robust absolute, in addition to time-difference, EIT imaging. Nevertheless, these methods often suffer from blurred reconstructions as a result of a low-pass filtering of the associated (non)linear Fourier data required by the reconstruction algorithms and offer limited applicability in an iterative model-based reconstruction framework.

As with other computational imaging tasks, deep learning has been leveraged in many ways to improve EIT reconstruction quality while maintaining or reducing inference time. Several direct approaches in the category of post-processing based learned image reconstruction have been proposed, such as Deep D-bar methods [21], [22] and the dominant-current approach [23] where a CNN utilizing the popular U-net architecture [24] is trained and used to improve an initial reconstruction in the image space on a pixel grid. Alternatively, a neural network-based supervised descent method (SDM) described in [25] falls into the *model-based* category. The inputs are residuals in the measurement space while the outputs are updates in the image space defined over a FEM mesh. Another model-based approach was presented in [26], where the authors propose a Quasi Newton method by learning updates of the Jacobians. The GCNM presented in Section III also falls in the *model-based* category, using GCNs with inputs and outputs in the image space defined over the problem specific FEM meshes used in the optimization method.

A. Solving the Forward Problem in EIT

Given a domain $\Omega \subset \mathbb{R}^2$, the EIT forward problem is to determine the electrical potential u at the boundary of the domain $\partial\Omega$ when current is applied and the conductivity distribution of the interior is known. The boundary conditions for the conductivity equation (1) are given by the complete electrode model [27],

$$\begin{cases} \int_{e_\ell} \sigma \frac{\partial u}{\partial \hat{n}} dS = I_\ell, & \ell = 1, 2, ..., L, \\ \sigma \frac{\partial u}{\partial \hat{n}}\big|_{\partial\Omega/\cup e_\ell} = 0, \\ (u + z_\ell \sigma \frac{\partial u}{\partial \hat{n}})\big|_{e_\ell} = U_\ell, & \ell = 1, 2, ..., L, \\ \sum_{\ell=1}^{L} I_\ell = 0, \\ \sum_{\ell=1}^{L} U_\ell = 0, \end{cases} \quad (2)$$

where L is the number of electrodes and e_ℓ is the ℓth electrode; z_ℓ, I_ℓ, and U_ℓ, are the contact impedance, current injected, and electric potential on the ℓth electrode, respectively; and $\hat{n}$ is the outward unit vector normal to the boundary. Following [28], [29], the forward problem (1) and (2) can be solved using FEM to determine the voltages $U(\sigma)$ on the electrodes for a given conductivity σ.

B. The Inverse Problem for EIT

Many iterative methods for EIT reconstruction have been proposed that begin as a minimization problem with the objective function

$$F(\sigma) = \frac{1}{2}\|U(\sigma) - V\|^2 + R(\sigma) \quad (3)$$

where $V = \left(V_1^{(1)}, \ldots, V_L^{(1)}, \ldots, V_1^{(K)}, \ldots, V_L^{(K)}\right)^T$ in $\mathbb{R}^{KL}$ represents a vector of the measured voltages on each of the L electrodes for K linearly independent current patterns and $U(\sigma) = \left(U_1^{(1)}(\sigma), \ldots, U_L^{(1)}(\sigma), \ldots, U_1^{(K)}(\sigma), \ldots, U_L^{(K)}(\sigma)\right)^T$ in $\mathbb{R}^{KL}$ is a vector of the simulated voltages at the L electrodes produced by the same K current patterns using the conductivity σ. The term $R(\sigma)$ represents a possible regularization term added to the norm of the residuals, commonly called the data-fidelity term.

The problem in (3) is nonlinear and thus a good initial guess of the solution is required. Here, σ_0 is the best constant conductivity fit to the data [30]. With this initial guess, (3) can be rewritten as

$$F(\sigma_0 + \delta\sigma) = \frac{1}{2}\|U(\sigma_0 + \delta\sigma) - V\|^2 + R(\sigma_0 + \delta\sigma) \quad (4)$$

with the intent to minimize it with respect to $\delta\sigma$. Solving (4) iteratively leads to the following update rule,

$$\sigma_{k+1} = \sigma_k + \delta\sigma_k, \quad (5)$$

given a current estimate σ_k. We then iterate until a good solution for (3) is found.

1) The Levenberg-Marquardt Algorithm: When there is no explicit regularization included in the objective function (3), then $R(\sigma) = 0$. The Taylor expansion of (4) to the quadratic term is then given by

$$F(\sigma + \delta\sigma) = F(\sigma) + F'(\sigma)(\delta\sigma) + \frac{1}{2}F''(\sigma)(\delta\sigma)^2,$$

and a minimum can be found by setting the gradient w.r.t. $\delta\sigma$ equal to 0. This yields the update

$$\delta\sigma = -F''(\sigma)^{-1}F'(\sigma) \quad (6)$$

where $F'(\sigma)$ and $F''(\sigma)$ are the gradient and Hessian of the objective function F. When $R(\sigma) = 0$, these are defined as

$$F'(\sigma) = J(\sigma)^T (U(\sigma) - V)$$
$$F''(\sigma) = J(\sigma)^T J(\sigma) + \sum_i U_i''(\sigma)(U_i(\sigma) - V_i) \quad (7)$$

where $J(\sigma)$ is the Jacobian of the simulated voltages $U(\sigma)$ (e.g., computed by [28], [29]). In Newton's method, the Hessian is computed exactly according to (7). In the Gauss-Newton (GN) method, the second term is ignored due to the costly computation of the second order derivative $U_i''(\sigma)$. Alternatively, the Levenberg-Marquardt (LM) algorithm proposes replacing the second term with a scaled identity matrix λI where $\lambda \in \mathbb{R}^+$. This has the benefit of improving the condition number for the matrix to be inverted in (6) which is of particular importance when the Jacobian is rank deficient. The approximate solution to (4) is then

$$\delta\sigma_{\text{LM}} = -\left(J(\sigma)^T J(\sigma) + \lambda I\right)^{-1} J(\sigma)^T (U(\sigma) - V). \quad (8)$$

We obtain an iterative reconstruction algorithm by using the above LM updates in the update rule (5) and a suitable stopping criteria. In particular, we will design learned iterative reconstruction using the above updates.

2) Regularized Gauss-Newton: Alternatively, one can enforce certain priors, such as piecewise constant conductivity reconstructions, that are desirable in EIT, as opposed to smooth reconstructions. This is typically approached via Total Variation (TV) with $R(\sigma) = \lambda \sum_i |\mathcal{L}_i \sigma|$ where $\mathcal{L}$ is a sparse matrix representing the discrete gradient, see [10] for details. Considering the difficulties in minimizing a sum of absolute values, one typically considers a smoothed approximation of Total Variation regularization using $R(\sigma) = \lambda \sum_i \sqrt{(\mathcal{L}_i\sigma)^2 + \gamma}$, the approximate solution to (4) is then

$$\delta\sigma_{TV} = -(J(\sigma)^T J(\sigma) + \lambda\mathcal{L}^T E^{-1}\mathcal{L})^{-1}$$
$$\left(J(\sigma)^T (U(\sigma) - V) + \lambda\mathcal{L}^T E^{-1}\mathcal{L}\sigma\right) \quad (9)$$

where $\gamma \in \mathbb{R}^+$ is the smoothing parameter that can be varied [10], and $E = \text{diag}\left(\sqrt{(\mathcal{L}\sigma)^2 + \gamma}\right)$ is a diagonal matrix.

III. GRAPH CONVOLUTIONAL NEWTON'S METHOD

The proposed Graph Convolutional Newton's Method (GCNM) is a new model-based image reconstruction technique that builds on any iterative Newton method but changes how the current iterate σ_k and its update $\delta\sigma_k$ are combined. As illustrated in Fig. 1 the classic updating rule (5) is now performed by a network Λ_{θ_k} via

$$\sigma_{k+1} = \Lambda_{\theta_k}([\sigma_k, \delta\sigma_k], \mathcal{A}), \quad (10)$$

where $\delta\sigma$ is the LM update (8) and the two terms $[\sigma_k, \delta\sigma_k]$ are concatenated and then used as input $H^{(0)}$ to a graph convolutional (GCN) block. Here, $\mathcal{A}$ denotes the adjacency matrix for the graph over which σ_k and $\delta\sigma_k$ are defined, as will be discussed in Section III-A. The output of the GCN block provides the next iterate σ_{k+1}. The structure of the blocks Λ
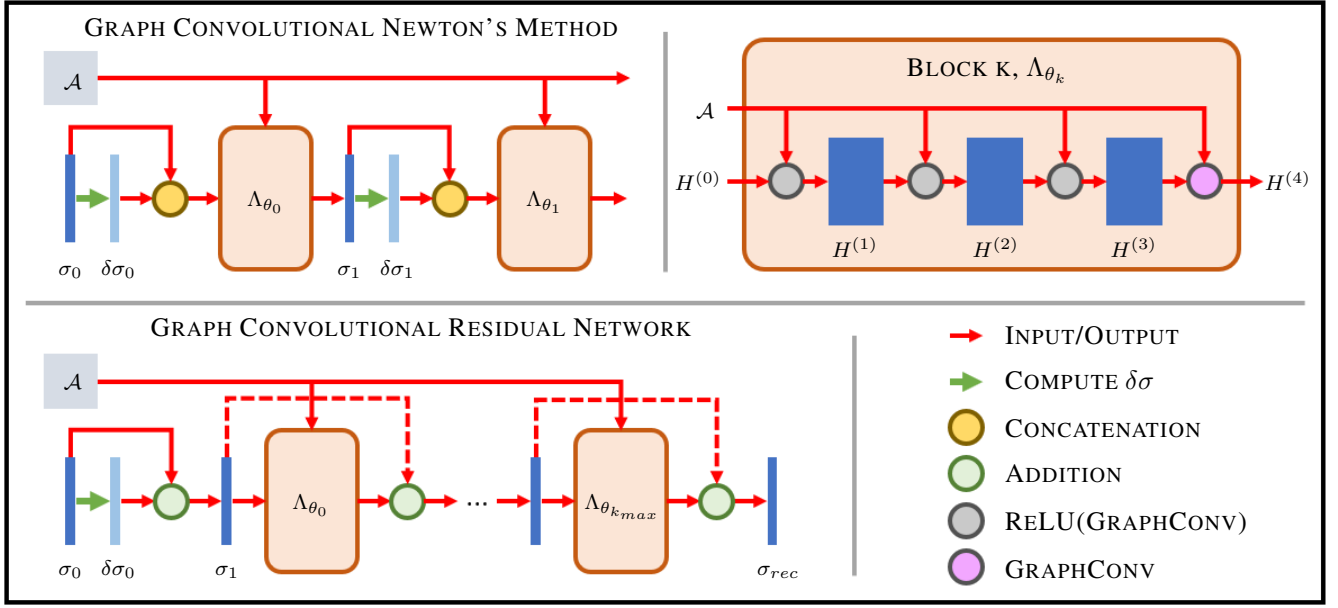
Fig. 1. (Top Left) The first two iterations of the GCNM are shown. (Bottom Left) The GResNet which takes the first iteration of a classical Newton method as input to the first block and a total of k_{max} blocks is shown. (Top Right) One block of a network is shown. The same block structure is used in both GCNM and the GResNet. In GCNM, the input $H^{(0)}$ is the concatenation of σ_k and $\delta\sigma_k$ and the output $H^{(0)}$ is σ_{k+1}. For the GResNet, $H^{(0)}$ is σ_1 for the first block and the sum of the previous block's input and output for the remaining blocks.

are the same at each iteration but each block will have its own unique set of trainable parameters θ_k. We chose the LM updates for GCNM, as we expect the networks to learn a data specific prior from the training set, rather than enforcing a specific one.

Similar to other learned model-based methods [5], [6], [25], [31], [32] there are two options for training the network: train the entire system of k_{max} blocks end-to-end, or train each block sequentially. In our case, an end-to-end training is not practical for two main reasons. First, to update the network parameters we would need to perform backprop-agation through the updates given by (8), where $J(\sigma)$ is computed by evaluating the model equations (2) given by a FEM solver, which is not practical. This leads directly to the second problem, that evaluating the model equations is time consuming and would lead to extensive training times. Thus, we follow here the sequential approach [31] of training each block separately. Given a training set of true conductivity $\sigma_{true,(i)}$ and current iterate $\sigma_{k,(i)}$ pairs for $i = 1, \ldots, M$, this leads then to a loss function requiring iterative-wise optimality

$$\text{Loss}(\theta_k) = \frac{1}{M} \sum_{i=1}^{M} (\Lambda_{\theta_k} ([\sigma_k, \delta\sigma_k], \mathcal{A})_{(i)} - \sigma_{true,(i)})^2. \quad (11)$$

A. Interpreting FEM Meshes as Graph Data

In order to consider the conductivity σ and update $\delta\sigma$, defined on a FEM mesh with M elements, as graph data, two pieces are needed: a feature matrix and an adjacency matrix. The feature matrix $H \in \mathbb{R}^{M \times f}$ has one row for each graph node and one column for each feature defined over the nodes. The adjacency matrix $\mathcal{A} \in \mathbb{R}^{M \times M}$ is sparse and describes how the graph nodes are connected. Only the entries $\mathcal{A}_{ij} = 1$

are nonzero where graph nodes i and j are connected. Here, we consider each mesh element as a graph node and two elements are connected if they share at least one mesh node. The same definition was used when forming $\mathcal{L}$ in the total variation regularization term. With the mesh data now defined over a graph, it follows that graph convolutions are a natural choice for a propagation rule.

B. Graph Convolutions

Kipf and Welling [33] proposed a convolutional layer for neural networks that operates on graph data and is analogous to the convolutional layer used on pixel grid data:

$$H^{(i+1)} = g\left(\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}} H^{(i)} W^{(i)}\right). \quad (12)$$

Self loops, or connections from graph node i to itself, are included in the adjacency matrix by $\tilde{\mathcal{A}} = \mathcal{A} + I$ and $\tilde{D}^{-\frac{1}{2}}$ denotes the inverse of the square root of each element from $\tilde{D} = \text{diag}(\sum_j \tilde{\mathcal{A}}_{ij})$. The weight matrix $W^{(i)} \in \mathbb{R}^{f^{(i)} \times f^{(i+1)}}$ contains the trainable parameters for the layer and $g(\cdot)$ denotes the nonlinear activation function. If biases are desired, an extra column of ones $\vec{1} \in \mathbb{R}^M$ is concatenated to the feature matrix and an extra row of trainable parameters $b^i \in \mathbb{R}^{f^{(i+1)}}$ is concatenated to the weight matrix resulting in

$$H^{(i+1)} = g\left(\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}} \left[H^{(i)}, \vec{1}\right] \left[W^{(i)}; b^{(i)T}\right]\right). \quad (13)$$

In (12) and (13), one can think of $\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}} H^{(i)}$ as an aggregation of features within the neighborhood of each graph node. Then, multiplication with $W^{(i)}$ computes a node's output features as linear combinations of its aggregated input features. Stacking X of these layers increases a node's receptive field to all of its X neighborhood in the graph. Note

that standard 2D convolutions learn linear combinations of neighbors' feature values for aggregating within a neighborhood while GCN layers only use a specific weighted average (described by $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$) for aggregating information.

C. Network Structure

At each iteration of the GCNM, a GCN block with the structure shown in Fig. 1 is used. The input graph uses the M elements from the mesh as nodes in the graph with an adjacency matrix also formed from the mesh. Each graph node has $f = 2$ initial features such that $H^{(0)} = [\sigma_k, \delta\sigma_k]$. Then, several graph convolutional layers with nonlinear activation functions expand the feature dimension before a final graph convolutional layer with linear activation is used to produce the output of the block. The output is a graph with the same adjacency matrix as the input, but with only one feature σ_{k+1} defined over the nodes. Similar to [6], [31], each block's network structure was chosen to be small and quite simple as compared to typically larger post-processing networks. Note, that in contrast to [6], [31], we do not use a residual update. The above architecture has been found to work well for all test problems considered here; more specialised architectures can be considered for a particular problem but this is outside the scope of this current work.

Finally, one important aspect of the new GCNM is that it uses model information iteratively. That means, the output from one block σ_{k+1} is used to compute the next Newton update $\delta\sigma_{k+1}$, and then both are used as input the next block $\Lambda_{\theta_{k+1}}$. At each iteration, new information from the forward problem is being introduced by $\delta\sigma_k$ and a new GCN block acts on the inputs. This can be a strength when the Newton updates converge toward the true solution. On the other hand, if the forward model is not accurate, the GCN can compensate and correct for the wrong components and extract useful information for the updates, acting as a learned model correction [34]. We will see this correcting nature in the experiments (e.g. Fig. 5).

D. A Graph Convolutional Residual Network

The most computationally expensive step in each iteration of the GCNM for EIT is computing the Jacobian of the forward problem $J(\sigma)$ used in the Newton update $\delta\sigma$. To show the importance of including this model information, the GCNM will also be compared to a graph convolutional residual network (GResNet) of about the same size. As is common in other residual networks, the GResNet used in this work and shown in Fig. 1 used skip connections made between the input and output of each block. See [35] for a discussion on skip connections and graph residual network size.

The first iteration of LM algorithm with $\lambda = 200$ was used as input to the GResNet with similarly structured blocks as the GCNM. In total, five blocks with the skip connections were used in series to compute the final reconstruction, σ_{rec}. To be clear, the main difference between the GResNet and the GCNM, is the lack of new information being introduced at each block and can be understood as an ablation study for the model information. The GCNM is a model-based iterative

method while the GResNet is a post-processing network. Without the need to compute the updates at each iteration, the GResNet is trained end-to-end.

IV. METHODS

Here we describe the specific examples considered, training data used, and metrics to be used for evaluation of the results.

A. Training Data and Examples Considered

The training data set consisted of 400 samples on a circular domain of radius 140mm with $L = 32$ electrodes, and 100 samples used for validation. Each sample has 1-4 elliptical inclusions of either higher or lower conductivity compared to the background (see Table I and Fig. 2).

Fig. 2. Examples of "Truths" from the simulated training data.

The first example case we consider is samples within the distribution of the training data. With an eye on the larger question of how well the trained network generalizes to data it has not seen, we focus the remainder of our attention on the following examples. Cases 2-6 explore samples from simulated data outside the distribution of training data. Namely, Case 2 looks at a chest shaped domain, Case 3 has samples with error added to the electrode locations when simulating the measured voltages, Case 4 explores incorrect domain modeling by reconstructing on a circular domain while the measured voltages were simulated on an ovular domain, and Case 5 has samples with large 'L'-shaped targets as opposed to the smaller ovular targets used in training. Case 6 considers samples with varying levels of added noise. Lastly, we turn to experimental EIT data in Case 7, using data collected using the ACT3 [36], [37] and KIT4 systems [38].

The same circular domain shape was used in Cases 1, 4, 5, and 6. Note that in Case 4, the reconstruction took place on the circular domain, despite the measurements being simulated on an elliptical domain with semi-major axes 130mm and 150mm. Cases 2 and 3 used a chest shaped domain for both simulating voltages and reconstruction. For all simulated samples, the simulated voltage data came from a mesh with approximately 5,000 elements, while reconstruction for all simulated and experimental data took place on different meshes with about 4,000 elements. The experimental data used circular domains with radii of 150mm and 140mm, respectively.

TABLE I

INCLUSION NUMBERS AND CONDUCTIVITY VALUES FOR THE SAMPLES IN EACH TEST CASE ARE SHOWN BELOW. THE CONDUCTIVITY VALUES ARE PRESENTED IN S/M AND WHERE A RANGE IS SHOWN, VALUES WERE DRAWN FROM A UNIFORM DISTRIBUTION WITHIN THE RANGE.

	Training	Cases 1-6	ACT3	KIT4
$N_{inclusions}$	1-4	1-3	3	1-3
σ_{bkg}	[0.40, 0.43]	[0.40, 0.43]	0.424	0.03
σ_{low}	[0.15, 0.25]	[0.15, 0.25]	0.24	0
σ_{high}	[0.65, 0.95]	[0.65, 0.95]	0.75	inf

The simulated test samples for Cases 1-4 and 6 have a slightly restricted distribution of inclusions. Each sample has only 1-3 inclusions, the upper limit for inclusion eccentricity is lowered, and more spacing between inclusions is enforced. Samples from Case 5 share the same conductivity ranges as the other test cases but have "L"-shaped inclusions.

Noise was added to the simulated voltages of the training data and Cases 1, 2, 5, and 6 using $v_p = v_p + \nu \text{mean}\left(|v_p|\right) \epsilon_p$. The percentage shown is used as ν, v_p is the vector of voltage measurements on the electrodes for the pth current pattern, and $\epsilon_p \in \mathbb{R}^L$ is a vector of standard normal random numbers. The training data and Cases 1, 2 and 6 used 0.5% noise ($\nu = 0.005$) while Case 5 compares various levels.

Case 3 assumes that errors are made when placing the electrodes in an experiment. The locations along the boundary of the electrodes are shifted by $N(0\text{mm}, 1\text{mm})$ when simulating the measured voltages as compared to where they are assumed to be during reconstruction (evenly spaced). The same shift was used for all samples in Case 3.

Finally, an important piece of modeling the forward problem is estimating the contact impedance between the electrodes and the surface of the domain. For all of the simulated cases, the contact impedance at each electrode was selected from $N(2.4\mu\Omega\text{m}, 0.1\mu\Omega\text{m})$ when computing the measured voltage. Then, for all reconstructions, including the experimental test cases, the contact impedance was assumed to be the mean.

B. Training

The same set of data was used to train both learned methods. For the GCNM, we computed the updates $\delta\sigma$ following the LM algorithm (8) with $\lambda = 200$. Each block was then trained minimising the iterate-wise loss function (11) using mini-batches of 10 samples and the Adam optimization method [39] as it is implemented in PyTorch (version 1.7.1). A learning rate of 0.002 was used. Lastly, training was only stopped when the validation loss failed to decrease for 200 epochs, and the trainable parameters that resulted in the minimum validation loss were saved. No constraints on the trainable parameters were used. Training the GCNM with 10 blocks took approximately 10 hours. Training of the blocks was done on a NVIDIA Titan V GPU while simulating the forward problem and computing $\delta\sigma$ were performed on a CPU cluster. Although a system of 10 blocks was trained, the reconstruction was chosen as $\sigma_{rec} = \sigma_k$ when $\|\delta\sigma_{k+1}\|_2 > 0.99 \times \|\delta\sigma_k\|_2$ or as the output of the final block $\sigma_{rec} = \sigma_{k_{max}}$ if the first criterion was not met.

The training for GResNet used the same sized mini-batches, optimizer, and stopping criteria as for training the GCNM networks, and was trained end-to-end in under 4 hours, including the time to solve the forward problem and compute $\delta\sigma_0$, on the same hardware. Fig. 3 displays the training and validation losses at the epoch with the minimum validation loss for each of the 10 blocks in the GCNM and the GResNet as a whole. The losses for the GResNet are shown for Iteration 0 on the horizontal axis because only one Newton update is used, the same as block 0 of the GCNM. The GResNet is not able to achieve a smaller validation loss than the first iteration

of GCNM which indicates that a deeper residual network structure with GCN blocks did not perform well and may have difficulties to capture the underlying structures for the EIT reconstruction problem.
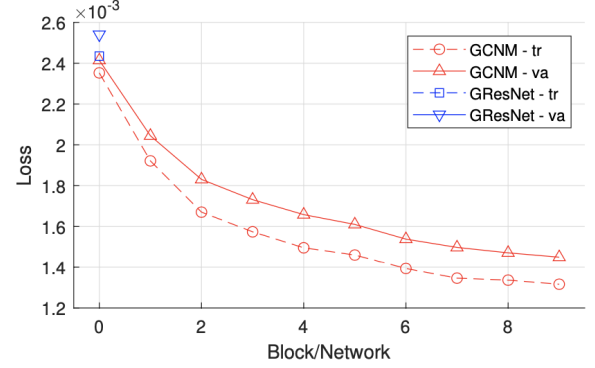

Fig. 3. The final loss values (training and validation) of each block of the the GCNM and for the GResNet as a whole.

C. Comparison methods

We compare the learned results to the classical variational approaches. For the LM algorithms, we chose the same regularization parameter $\lambda = 200$ as in the GCNM, as it was found empirically to perform best in attaining a minimum mean squared error (MSE) in conductivity for a subset of the training data. We used a relative error in the voltage measurements

$$RE_V(\sigma_k) = \frac{\|U(\sigma_k) - V\|_2}{\|V\|_2} \tag{14}$$

as stopping criteria and the iteration was stopped when $RE_V(\sigma_{k+1}) > 0.99 \times RE_V(\sigma_k)$ with $\sigma_{rec} = \sigma_k$.

For the total variation regularized Gauss-Newton, we used $\lambda = 0.1$ and the smoothing parameter $\gamma = 1\text{e}{-8}$, which were also empirically found from a subset of the training data. The stopping criteria used for LM algorithms was also used for the regularized Gauss-Newton TV method.

D. Metrics

As there is no gold-standard metric for EIT images, we include the following metrics to give information on the quality of the reconstructed conductivity: MSE, dynamic range

$$DR = \frac{\max(\sigma_k) - \min(\sigma_k)}{\max(\sigma_{true}) - \min(\sigma_{true})} \times 100\%,$$

an ℓ_1 relative error, and the relative voltage error $RE_V(\sigma_k)$ (14). To compute the metrics, for the test Cases 1-4, averages over 100 simulated samples will be reported. Case 5 uses an average over only 10 samples due to the lack of deviation in inclusion location and size. Case 6 uses an average over 100 samples for each noise level. For the experimental data, the metrics each correspond to a single sample of data.

V. RESULTS

We now present the results for simulated in and out of distribution data, as well as experimental data that is out of distribution.
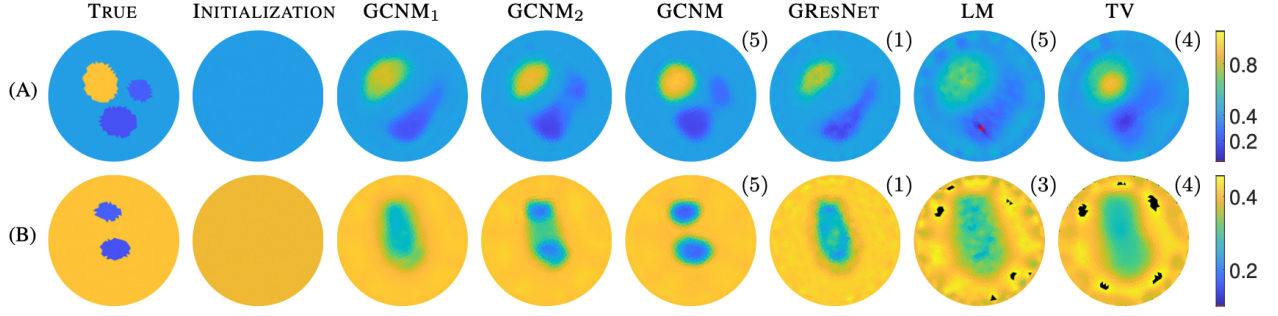
Fig. 4. Case 1: Results for two samples of testing data consistent with, but not included in, the training data. The initialization, first two iterations of the GCNM, and the GCNM reconstruction are compared to the reconstructions from GResNet, LM, and TV. Black and red colors represent going over or under the color bar range, respectively.

A. Simulated Results

Figure 4 compares the results of the new GCNM against GResNet, LM, and TV for two samples consistent with, but not used in, the training or validation data. The initialization as well as the next two iterations of GCNM are shown in addition to the output GCNM image chosen by the stopping criterion. The number in the top right of each image denotes the iterate chosen by the stopping criterion for each method. Note that the GCNM outperforms the other methods not only in visual sharpness but in each of the metrics (see Table II) aside from the voltage error RE_V, which is not unexpected as the network was optimized to reduce the MSE in the conductivity rather than the voltage data. Furthermore, GCNM is the only method to separate the three targets in (A) and two targets in (B). The output from the GResNet looks similar to the first iteration of the GCNM except with more artefacts, especially near the boundary.

Next, Figure 5 presents results of the methods on *out of distribution* data for Cases 2-4; metrics are presented in Table II. Beginning with Case 2, the chest shaped domain, again the GCNM produces the sharpest reconstructions and outperforms the other methods in MSE and relative ℓ_1 error, and had the second best dynamic range across the 100 test samples, even though the network was optimized for only data coming from a circular domain. This generalizability stems directly from the translation invariance of convolutions and nicely illustrates the capabilities of the network to handle a mesh that it was not trained on. This holds promise for clinical imaging settings where domain shapes would be inconsistent from patient to patient. Similarly, in Case 3, for which the electrode locations are incorrectly modeled, the GCNM outperformed the traditional LM and TV methods, visually and by the metrics, except for the voltage error.

Typically, mis-modelling of the domain results in large artefacts near the boundary of the domain where the mis-modeling has occurred. Errors are present in the GCNM and GResNet reconstructions at the boundary near the electrodes with the largest errors, however the interior reconstruction remains quite stable. In contrast, the errors are larger in magnitude (note the red and black pixels near the edges which correspond to conductivity values outside the range of $\pm20\%$ of the true conductivity values) and more prominent in the LM and TV images. This effect becomes more pronounced when considering Case 4 where the data comes from an

ovular domain but is assumed to have come from a circular domain. Here we see major artefacts around the boundary in the GResNet and LM reconstructions. The GCNM contains boundary artefacts as well, but the effect is less pronounced. The TV method outperforms GCNM, GResNet, and LM in all metrics considered with the GCNM following in second for each. Though the metrics may indicate TV resulted in the best reconstruction, visually it did not move very far from the initialization, and no inclusion boundaries can be interpreted from the image. Cases 3 and 4 are particularly important points of study for EIT where it is unlikely to precisely know the electrode locations and a patient breathing or moving changes their location as well as the domain shape.
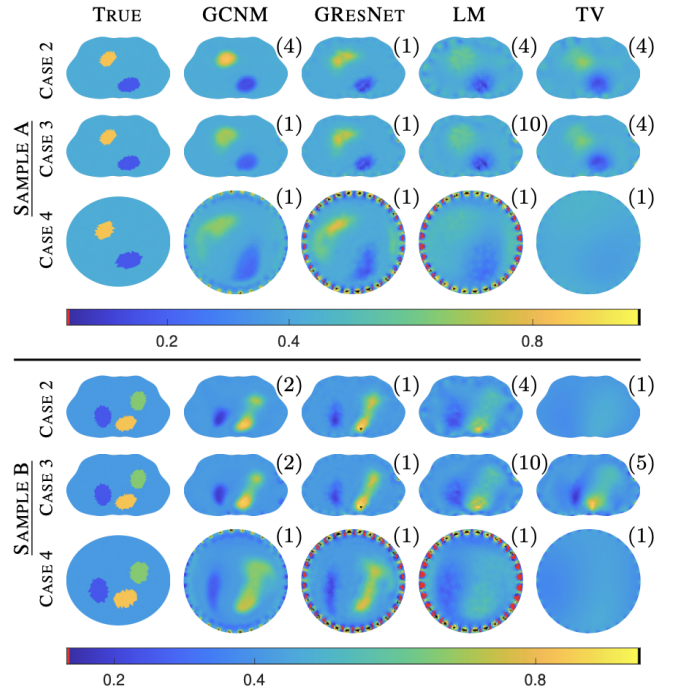

Fig. 5. Results for Cases 2-4: exploring a chest shaped domain, and incorrect domain modeling

Figure 6 explores Case 5 where the target inclusion is an 'L'. Recall that the training data was only elliptical inclusions and thus this target was quite different than those to which the network was exposed during training. Nonetheless, each method produces visually recognizable 'L' shaped targets. The LM and TV methods contain more pixels out of the $\pm20\%$

TABLE II
ERROR METRICS FOR CASES 1-5. REPORTED METRICS ARE AVERAGES OVER 100 SAMPLES EXCEPT FOR CASE 5, WHICH USED 10 SAMPLES. BOLD TABLE ENTRIES REPRESENT BEST IN CASE SCORES.

Case	Method	Its	MSE_σ $(\times 10^3)$	Rel. ℓ_1 Error	RE_V	DR(%)
1	GCNM	2.7	**1.804**	**0.034**	0.0108	**101.5**
	GResNet	1.0	2.618	0.053	0.0196	102.4
	LM	3.3	3.294	0.071	**0.0039**	81.0
	TV	3.8	3.215	0.074	0.0041	136.9
2	GCNM	2.7	**1.668**	**0.034**	0.0118	113.3
	GResNet	1.0	2.382	0.052	0.0196	126.2
	LM	3.4	3.092	0.072	**0.0038**	108.2
	TV	3.7	3.420	0.075	0.0047	185.8
3	GCNM	1.3	**2.669**	**0.050**	0.0488	**156.0**
	GResNet	1.0	4.188	0.069	0.0242	342.0
	LM	9.8	4.240	0.086	**0.0025**	267.8
	TV	4.2	7.277	0.106	0.0026	520.1
4	GCNM	1.1	26.507	0.224	0.6108	250.5
	GResNet	1.0	60.997	0.362	2.9135	714.9
	LM	1.0	63.198	0.388	5.0291	713.5
	TV	1.0	**6.225**	**0.106**	**0.0700**	**143.0**
5	GCNM	3.2	**1.801**	**0.040**	0.0150	**142.9**
	GResNet	1.0	2.759	0.066	0.0195	172.9
	LM	3.9	4.098	0.101	**0.0039**	152.9
	TV	3.9	4.929	0.111	0.0043	290.2

Fig. 7. Results for Case 6: Varying the noise level. Note that samples with 0.5% noise were used during training.

Fig. 8. Case 6: GCNM reconstructions for two samples with varying levels of added noise shown with two different values for the regularization parameter λ in (8). Note that the parameter was only changed when testing; all training was done with $\lambda = 200$.

threshold. The GResNet image contains significant fluctuations at the domain boundary, as does LM. The GCNM contains a low conductivity artefact near inner corner of the 'L' but still resulted in the sharpest reconstruction and best metrics (see Table II) aside from the voltage error.
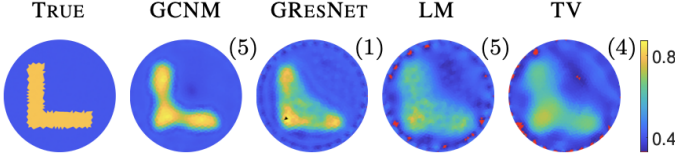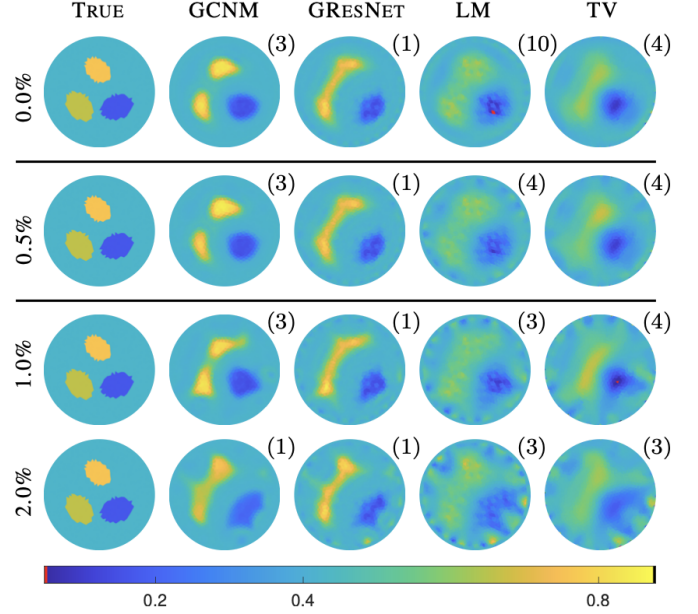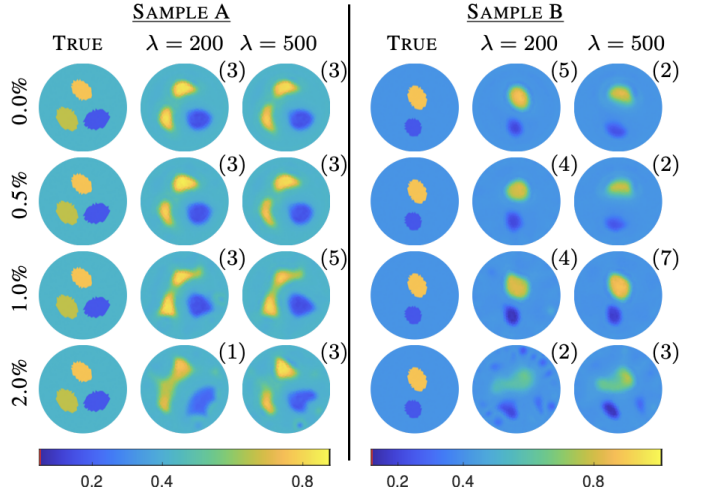

Fig. 6. Results for Case 5: An L-shaped out of distribution inclusion.

1) Noise: Figure 7 compares conductivity reconstructions for varying levels of noise in the simulated voltage data corresponding to Case 6. Recall that the network was trained with 0.5% noise. As the level of noise in the data increases, the LM and TV methods suffer from the classic boundary errors common in absolute EIT imaging, namely over and underestimating the conductivity near the boundary. For the TV reconstructions, large magnitude errors in a few pixels at the edge of the domain that are difficult to see visually led to large dynamic range values that are reflected in Table III. It should be noted that 2% noise is an extreme amount of noise for EIT reconstruction with most systems capable of SNR comparable to 0.1% noise. Nevertheless, the GCNM images are not corrupted by boundary errors, and as shown in Figure 8 and Table III, adjusting the reconstruction parameter λ in the update $\delta\sigma$ in (8) results in even fewer artefacts. At around 1.0% noise or more, using a larger regularization parameter $\lambda = 500$ resulted in reconstructions with lower MSE and relative ℓ_1 error. This is a key feature of the proposed GCNM as a model-based approach, that even after training with a fixed regularization parameter we can adjust and control the performance if noise levels change, offering greater flexibility and control.

B. Experimental Results

We considered three data sets from two different EIT machines, the 32-electrode ACT3 system from Rensselaer Polytechnic Institute and the 16-electrode KIT4 system from the University of Eastern Finland. Results are shown in Figure 9. Recall that for all reconstruction methods these are *absolute* EIT images, not difference images, and thus they are notoriously challenging to obtain without a careful tuning of the forward model to the EIT machine hardware. Here, the forward model was not tuned to the respective EIT machines and the same contact impedance was used for all electrodes, the mean of those used in the training. The regularization parameter for (8) was lowered to $\lambda = 10$ for the ACT3 data and increased to $\lambda = 10000$ for the KIT4 data. This was done

TABLE III
METRICS FOR VARYING THE NOISE LEVEL AND ADJUSTING λ. THE METHOD "GCNM" USED $\lambda = 200$ FOR COMPUTING $\delta\sigma$ (SAME AS TRAINING) WHILE "GCNM2" USED $\lambda = 500$.

Noise Level	Method	Its	MSE_σ ($\times 10^3$)	Rel. ℓ_1 Error	RE_V	DR(%)
0.0%	GCNM	2.5	**1.769**	**0.033**	0.0098	99.3
	GResNet	1.0	2.548	0.052	0.0189	**99.6**
	LM	10.0	2.789	0.061	**0.0003**	83.6
	TV	4.2	2.869	0.065	0.0007	105.8
	GCNM2	2.3	2.174	0.037	0.0110	88.1
0.5%	GCNM	2.7	**1.804**	0.034	0.0108	101.5
	GResNet	1.0	2.618	0.053	0.0196	102.4
	LM	3.3	3.294	0.071	**0.0039**	81.0
	TV	3.8	3.215	0.074	0.0041	136.9
	GCNM2	2.6	2.204	0.037	0.0119	89.9
1.0%	GCNM	2.7	**2.110**	0.039	0.0137	106.4
	GResNet	1.0	2.791	0.056	0.0215	108.6
	LM	2.6	3.809	0.083	**0.0076**	94.6
	TV	3.2	3.838	0.084	0.0082	169.7
	GCNM2	2.9	2.201	**0.038**	0.0139	**95.1**
2.0%	GCNM	1.3	3.033	0.051	0.0242	101.3
	GResNet	1.0	3.546	0.069	0.0297	128.5
	LM	2.6	5.778	0.116	**0.0155**	157.9
	TV	2.0	4.432	0.088	0.0175	147.4
	GCNM2	2.3	**2.887**	**0.048**	0.0214	**99.7**

so that the condition numbers of $\left(J^T J + \lambda I\right)$ were on the same order of magnitude as the training data. Increasing λ to 500 for the higher noise cases in Fig. 8 had the same effect on the condition number. These adjustments resulted in GCNM, GResNet, and LM reconstructions with more clearly defined targets for the experimental data as well.

For the ACT3 data, each reconstruction method recovered the heart and lungs' shapes but failed to completely separate the lungs. The two learned methods produce the most distinct target boundaries with the GCNM reconstruction having the most recognizable shapes. The dynamic range of the GCNM, GResNet, and LM method images were 169%, 153%, and 159%, respectively, while the TV image had a DR of 342% with the greatest errors occurring at the boundary. Similarly to the simulated data cases, the LM and TV methods outperformed the learned methods in relative voltage error RE_V.

The background conductivity in the KIT4 data was 0.03 S/m whereas the training, testing, and ACT3 data was in the range [0.40, 0.43] S/m. For the learned methods, to bring the data into scale for the trained networks, the network inputs σ_k and $\delta\sigma_k$ were scaled up by a factor of 14 and the outputs σ_{k+1} were scaled back down by a factor of 14; see Figure 9. Additionally, we point out that the KIT4 data contains targets that violate the mathematical model; the metal targets are infinite conductors and the plastic triangle is a perfect resistor. A more careful tuning of the forward solver to the experimental KIT4 device would likely improve the results for all methods.

For both KIT4 samples, all targets were located by all methods, but the GCNM achieved the clearest target boundaries with the fewest artefacts. We highlight that sharp corners of the triangle were recovered by the GCNM even though the network was trained solely on elliptical targets. As the KIT4 data contained metal targets, only relative voltage error is reported. For the KIT4-A case, the LM method achieved the

lowest RE_V at 0.007 compared to the TV method at 0.047, GCNM at 0.059, and GResNet at 0.071. For the KIT4-B case, the LM method stopped after just one iteration, likely due in part to negative conductivity values that break the assumptions of the forward model. As such, LM method resulted in the highest RE_V at 1.304. GCNM achieved the lowest RE_V at 0.055 compared to GResNet at 0.076 and TV method at 0.199.
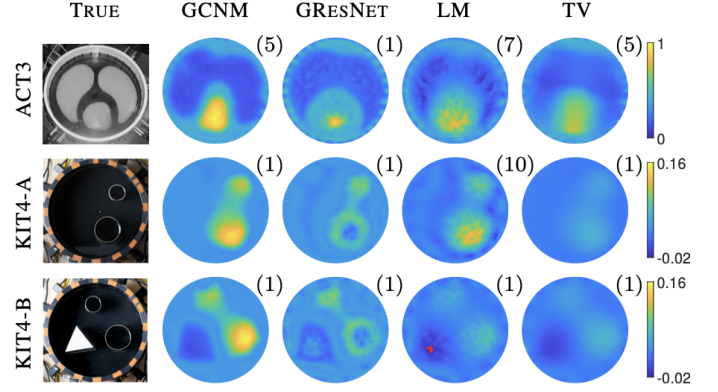

Fig. 9. Results for Case 7: Experimental Data. Column 1 contains the experimental setups. Each row is shown on its own colorbar with the range chosen to highlight the majority of the variation.

C. Further Discussion

All computations were performed on the hardware described in Section IV-B. Computational costs per sample were as follows: 2.5 seconds per iteration of GCNM and LM, 3.7 seconds per iteration of TV, and 3.4 seconds per reconstruction using GResNet. We emphasize that the individual algorithms were not optimized for speed in this study. Furthermore, a better tuning of the regularization parameters and forward model for the TV approach would likely result in improved reconstructions. Nonetheless, the GCNM provided significant improvements in MSE, relative ℓ_1 error, dynamic range, and visual reconstruction for the simulated and experimental data considered here, in particular for the out of distribution data, with only a small, general, training set used. The flexibility of the approach to later allow the user to adjust the regularization parameter(s) of the Newton-type method used (LM here) without retraining the network adds generalizability. Of particular importance to absolute EIT imaging were the incorrect domain modeling cases (3 and 4). The apparent robustness of GCNM to such errors holds promise for clinical absolute EIT imaging.

VI. CONCLUSIONS

We successfully introduced an approach to combine optimization-based solution methods with deep learning directly on nonuniform problem-specific solution meshes. The proposed GCNM provides a simple, yet highly flexible, network architecture that combines the current iterate σ_k and its Newton-type update $\delta\sigma_k$ to produce a new improved estimate. This enables the network to leverage the information from both inputs, learn a task-specific prior from training data, as well as a robust correction of potential model mismatch. As the domain modeling is done outside the network, a network can be trained on a simplified domain such as a circle, and learned

weights are still applicable to different domain shapes. This holds promise for several medical imaging applications where training data could be simulated on a fixed/average domain and then applied to patient-specific models.

While we used a simple prototypical network architecture to demonstrate the effectiveness of the model-based approach, extensions to different architectures can be considered as well as additional information supplied to the network. Finally, the approach extends naturally into 3D with no major change, as the graph structure is not limited by spatial dimensions. In terms of the specific application, EIT is a highly ill-posed nonlinear inverse problem and as such represents an especially challenging case study. The presented results demonstrate a promising improvement for absolute EIT imaging given that we did not tune the forward solver to the experimental data taken from the two separate machines. Additionally, we trained only on general ellipse inclusions and were able reconstruct targets of different shapes. This can be attributed to the additional information supplied by the Newton update. We expect that the presented approach extends directly to, and bears great promise for, other tomographic reconstruction problems, where data and image are closely tied to problem specific finite element meshes, instead of pixel/voxel grids.

REFERENCES

[1] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[2] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging, 2020.

[3] E. Kang, J. Min, and J. C. Ye. A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Medical physics*, 44(10):e360–e375, 2017.

[4] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[5] P. Putzky and M. Welling. Recurrent inference machines for solving inverse problems. *arXiv preprint arXiv:1706.04008*, 2017.

[6] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[7] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.

[8] D. O. Baguer, J. Leuschner, and M. Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *Inverse Problems*, 36(9):094004, 2020.

[9] W. R. B. Lionheart. EIT reconstruction algorithms: pitfalls, challenges and recent developments. *Physiological Measurement*, 25(1):125–142, feb 2004.

[10] A. Borsic, B. Graham, A. Adler, and W. Lionheart. In vivo impedance imaging with total variation regularization. *IEEE Transactions on Medical Imaging*, 29:44–54, 2010.

[11] A. Nissinen, V. P. Kolehmainen, and J. P Kaipio. Compensation of modelling errors due to unknown domain boundary in electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 30(2):231–242, 2010.

[12] L. Dong, V. Kolehmainen, S. Siltanen, A. Laukkanen, and A. Seppänen. Estimation of conductivity changes in a region of interest with electrical impedance tomography. *Inverse Problems and Imaging*, 9(1):211–229, 2015.

[13] L. Borcea. Electrical impedance tomography. *Inverse Problems*, 18:99–136, 2002.

[14] J.L. Mueller and S. Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.

[15] A. I. Nachman. Global uniqueness for a two-dimensional inverse boundary value problem. *Annals of Mathematics*, 143:71–96, 1996.

[16] S. J. Hamilton, J. L. Mueller, and T. R. Santos. Robust computation in 2d absolute eit (a-eit) using d-bar methods with the exp approximation. *Physiological Measurement*, 39(6):064005, 2018.

[17] J. L. Mueller and S. Siltanen. The d-bar method for electrical impedance tomography—demystified. *Inverse Problems*, 36(9):093001, sep 2020.

[18] M. Dodd and J. L. Mueller. A real-time D-bar algorithm for 2-D electrical impedance tomography data. *Inverse Problems and Imaging*, 8(4):1013–1031, 2014.

[19] A.-P. Calderón. On an inverse boundary value problem. In *Seminar on Numerical Analysis and its Applications to Continuum Physics (Rio de Janeiro, 1980)*, pages 65–73. Soc. Brasil. Mat., Rio de Janeiro, 1980.

[20] P. A. Muller, J. L. Mueller, and M. M. Mellenthin. Real-time implementation of calderón's method on subject-specific domains. *IEEE Transactions on Medical Imaging*, 36(9):1868–1875, 2017.

[21] S. J. Hamilton and A. Hauptmann. Deep D-Bar: Real-time electrical impedance tomography imaging with deep neural networks. *IEEE Transactions on Medical Imaging*, 37(10):2367–2377, Oct 2018.

[22] S. J. Hamilton, A. Hänninen, A. Hauptmann, and V. Kolehmainen. Beltrami-net: domain-independent deep d-bar learning for absolute imaging with electrical impedance tomography (a-EIT). *Physiological Measurement*, 40(7):074002, jul 2019.

[23] Z. Wei, D. Liu, and X. Chen. Dominant-current deep learning scheme for electrical impedance tomography. *IEEE Transactions on Biomedical Engineering*, 66(9):2546–2555, 2019.

[24] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[25] Z. Lin, R. Guo, K. Zhang, M. Li, S. Xu, and A. Abubakar. Neural network based supervised descent method for 2d electrical impedance tomography. *Physiological Measurement*, 41, 06 2020.

[26] D. Smyl, T. Tallman, D. Liu, and A. Hauptmann. An efficient quasi-newton method for nonlinear inverse problems via learned singular values. *IEEE Signal Processing Letters*, pages 1–1, 2021.

[27] 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, 1992.

[28] J. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen. Statistical inversion and monte carlo sampling methods in electrical impedance tomography. *Inverse Problems - INVERSE PROBL*, 16, 07 2000.

[29] M. Vauhkonen. Electrical impedance tomography and prior information, 1997.

[30] H. Jain, D. Isaacson, P. M. Edic, and J. C. Newell. Electrical impedance tomography of complex conductivity distributions with noncircular boundary. *IEEE Transactions on Biomedical Engineering*, 44(11):1051–1060, Nov. 1997.

[31] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge. Model-based learning for accelerated, limited-view 3-d photoacoustic tomography. *IEEE transactions on medical imaging*, 37(6):1382–1393, 2018.

[32] A. Hauptmann, J. Adler, S. R. Arridge, and O. Öktem. Multi-scale learned iterative reconstruction. *IEEE Transactions on Computational Imaging*, 2020.

[33] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016.

[34] S. Lunz, A. Hauptmann, T. Tarvainen, C.-B. Schönlieb, and S. Arridge. On learned operator correction in inverse problems. *SIAM Journal on Imaging Sciences*, 14(1):92–127, 2021.

[35] J. Zhang and L. Meng. Gresnet: Graph residual network for reviving deep gnns from suspended animation, 2019.

[36] R. D. Cook, G. J. Saulnier, D. G. Gisser, J. C. Goble, J. C. Newell, and D. Isaacson. ACT3: A high-speed, high-precision electrical impedance tomograph. *IEEE Transactions on Biomedical Engineering*, 41(8):713–722, August 1994.

[37] D. Isaacson, J. L. Mueller, J. C. Newell, and S. Siltanen. Reconstructions of chest phantoms by the D-bar method for electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 23:821–828, 2004.

[38] A. Hauptmann, V. Kolehmainen, N. M. Mach, T. Savolainen, A. Seppänen, and S. Siltanen. Open 2d electrical impedance tomography data archive. *arXiv preprint arXiv:1704.01178*, 2017.

[39] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2015.