

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

William Herzberg, Daniel B. Rowe, Andreas Hauptmann, *Member, IEEE*, 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-type Method (GCNM), which includes the forward model in 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 and meshes, out of distribution data as well as experimental data, from purely simulated training data and without transfer training.

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

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, where a data set of known ground truth images and measurement input pairs are used to train a network that is

W. Herzberg, D. Rowe, and S.J. Hamilton are with the Department of Mathematical and Statistical Sciences; Marquette University, Milwaukee, WI 53233 USA, (william.herzberg@mu.edu), (daniel.rowe@mu.edu), (sarah.hamilton@mu.edu)

A. Hauptmann is with the Research Unit of Mathematical Sciences; University of Oulu, Oulu, Finland and with the Department of Computer Science; University College London, London, United Kingdom, (e-mail: andreas.hauptmann@oulu.fi)

Manuscript submitted: July 2, 2021. This work was supported in part by National Institute Of Biomedical Imaging And Bioengineering of the National Institutes of Health under Award Number R21EB028064. The work of AH is supported by the Academy of Finland project numbers: 336796, 338408.

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 with parameters θ can be used, with the approximate solution as the input, to improve this initial reconstruction and obtain a cleaned version as, [3], [4],

$$\hat{x} = \Lambda_\theta(\mathcal{T}^\dagger y).$$

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 [5], [6]:

$$x_{k+1} = \Lambda_{\theta_k}(x_k, y, \mathcal{T}).$$

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.

Hence, we propose considering the data defined over a FEM mesh 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 [9]. Here, we apply GCNs to FEM mesh data in the context of learned image reconstruction, opening a new avenue to work directly on the problem specific meshes.

The proposed method, dubbed the Graph Convolutional Newton-type Method (GCNM), utilizes the model-based learning approach by incorporating GCNs into traditional iterative methods for solving nonlinear inverse problems, namely Newton-type methods, which use FEM meshes to solve 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 without the need for transfer training. 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 motivate and introduce the novel GCNM as well as the post-processing GResNet used for comparison. Next, Section III, reviews the mathematical problem of electrical impedance tomography (EIT), a highly ill-posed nonlinear inverse problem, which will serve as a challenging case study for the proposed GCNM. 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. GRAPH CONVOLUTIONAL NEWTON-TYPE METHOD

The proposed Graph Convolutional Newton-type Method (GCNM) is a new model-based image reconstruction technique that operates on FEM meshes and builds on iterative second order (Newton-type) methods but changes how the current iterate x_k and its update δx_k are combined. Classic methods obtain the next iterate as $x_{k+1} = x_k + \delta x_k$, where δx_k is computed according to a chosen optimization method such as Gauss-Newton (GN) or Levenberg-Marquardt (LM). The GCNM, by contrast, computes the next iterate using a trained network Λ_{θ_k} and the adjacency matrix \mathcal{A} , for the graph over which x_k and δx_k are defined, via

$$x_{k+1} = \Lambda_{\theta_k} ([x_k, \delta x_k], \mathcal{A}), \quad (1)$$

where δx is the traditional update (e.g. GN, LM). The two terms x_k and δx_k are concatenated and then used as input to a graph convolutional (GCN) block. The output of the GCN block provides the next iterate x_{k+1} , see Figure 1. In this work, the structure of the blocks Λ_{θ_k} are the same at each iteration but each block will have its own unique set of trainable parameters θ_k . In this study we chose the LM updates for the GCNM, to not enforce strong data specific priors and rather expect the networks to learn the features from the training set. We note that the proposed framework extends to other updates arising from GN methods and one could include stronger priors, such as total variation (TV), if desired.

As with other learned model-based methods [5], [6], [10]–[12] 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 back-propagation through the updates δx_k , computed by evaluating the model equations of the underlying problem 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 of training each block separately [10]. Given a training set of true $x_{true,(i)}$ and current iterate $x_{k,(i)}$ pairs for $i = 1, \dots, M$, this leads then to a loss function requiring iterative-wise optimality

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

In order to consider a quantity x and update δx , 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. Alternatively, if the solution is defined at the FEM nodes, those would be the natural graph nodes to use.

A. Graph Convolutions

Kipf and Welling [9] proposed a convolutional layer for neural networks that operates on graph data and is analogous to the convolutional layer used on pixel grid data, given input $H^{(i)}$ the layer is defined as:

$$H^{(i+1)} = g \left(\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}} H^{(i)} W^{(i)} \right). \quad (3)$$

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 with number

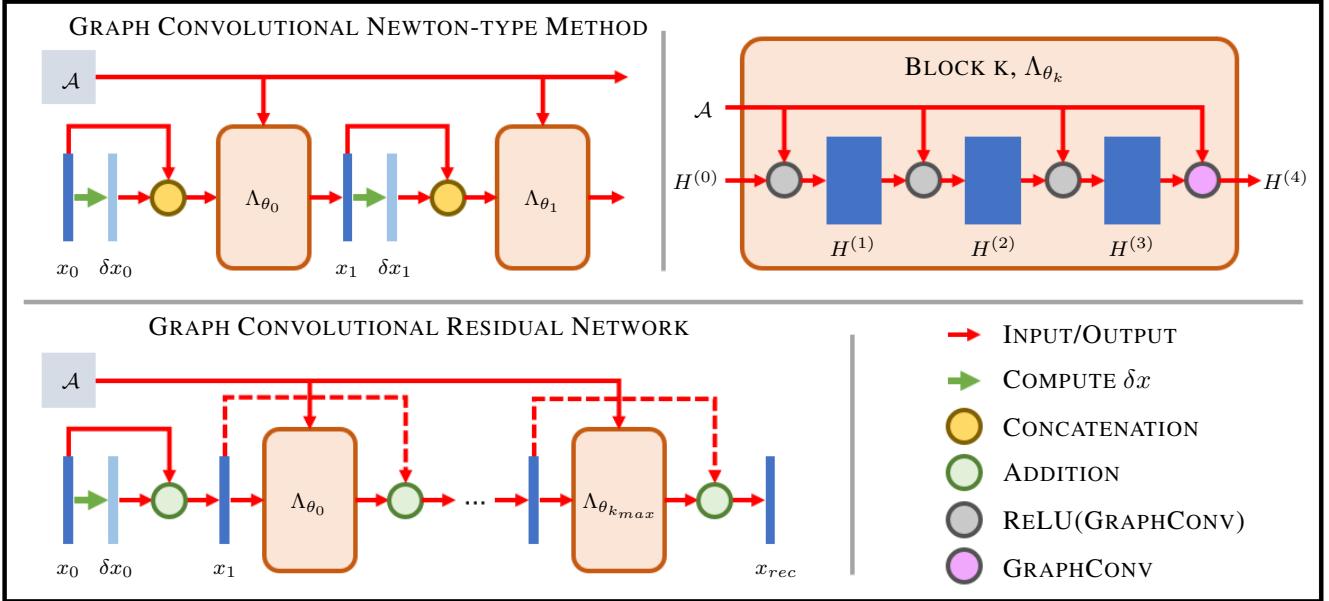


Fig. 1. (Top Left) The first two iterations of the GCNM are shown. (Bottom Left) The GResNet which takes the first iteration of a classical Newton method as input to the first block and a total of k_{max} blocks is shown. (Top Right) One block of a network is shown. The same block structure is used in both GCNM and the GResNet. In GCNM, the input $H^{(0)}$ is the concatenation of x_k and δx_k and the output $H^{(4)}$ is x_{k+1} . For the GResNet, $H^{(0)}$ is x_1 for the first block and the sum of the previous block's input and output for the remaining blocks.

of input $f^{(i)}$ and output $f^{(i+1)}$ features. Finally, $g(\cdot)$ denotes the nonlinear activation function. If biases are desired, an extra column of ones $\vec{1} \in \mathbb{R}^M$ is concatenated to the input 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

$$H^{(i+1)} = g \left(\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}} \left[H^{(i)}, \vec{1} \right] \left[W^{(i)}; b^{(i)T} \right] \right). \quad (4)$$

In (3) and (4), 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 $\tilde{D}^{-\frac{1}{2}} \tilde{\mathcal{A}} \tilde{D}^{-\frac{1}{2}}$) for aggregating information that is not learned, but instead defined by the mesh.

B. 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 $f^{(0)} = 2$ initial features such that $H^{(0)} = [x_k, \delta x_k]$. Then, three graph convolutional layers with ReLU activation functions expand the feature dimension to 250 features 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 x_{k+1} defined over the nodes. Similar to [6], [10], 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], [10], 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 could be considered for a particular problem but this is outside the scope of this current work.

One important aspect of the new GCNM is that it uses model information iteratively. That means, the output from one block x_{k+1} is used to compute the next update δx_{k+1} , and then both are used as input in the next block $\Lambda_{\theta_{k+1}}$. At each iteration, new information from the forward problem is being introduced by δx_k and a new GCN block acts on the inputs. This can be a strength when the original updates δx_{k+1} converge toward the true solution. Alternatively, 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 [13], [14]. We will see this correcting nature in the experiments (e.g. Fig. 5).

Finally, when regularization parameters are used in the computation of the updates δx_k that are used as input to the networks, the parameters can be tuned to new test data after training. This will be demonstrated in the following experiments where different noise distributions, parameter ranges, and experimental data are considered.

III. CASE STUDY: ELECTRICAL IMPEDANCE TOMOGRAPHY

We chose Electrical Impedance Tomography (EIT) as a case study for the proposed GCNM due to its nonlinear nature and severe ill-posedness. 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* [15]

$$\nabla \cdot \sigma \nabla u = 0 \quad \text{in } \Omega \subset \mathbb{R}^n, \quad (5)$$

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 [16], Tikhonov regularization [17], and Total Variation (TV) regularization [18]. 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. The most common applications of time-difference EIT focus on monitoring heart and lung function of hospitalized patients. *Absolute*, also called *static*, EIT imaging recovers the conductivity at the time of measurement from a single frame V of EIT data. While in time-difference imaging, some of the modeling errors can cancel out, in absolute imaging they do not and hence require a correction [19], [20]. 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 [21], [22] for a further literature review of applications.

Alternatively, direct (non-iterative) reconstruction methods such as the D-bar method [23]–[26], and Calderón’s method [15], [27] show promise for fast robust absolute and time-difference EIT imaging. However, 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 learned 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 [28], [29] and the dominant-current approach [30] where a CNN utilizing the popular U-net architecture [31] 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 [12] 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 [32], where the authors propose a Quasi Newton method by learning

updates of the Jacobians. The GCNM presented in Section II 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 $\Omega \subset \mathbb{R}^n$, 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 (5) are given by the complete electrode model [33],

$$\left\{ \begin{array}{l} \int_{e_\ell} \sigma \frac{\partial u}{\partial \hat{n}} dS = I_\ell, \quad \ell = 1, 2, \dots, L, \\ \sigma \frac{\partial u}{\partial \hat{n}} \Big|_{\partial\Omega \setminus \cup e_\ell} = 0, \\ (u + z_\ell \sigma \frac{\partial u}{\partial \hat{n}}) \Big|_{e_\ell} = U_\ell, \quad \ell = 1, 2, \dots, L, \\ \sum_{\ell=1}^L I_\ell = 0, \\ \sum_{\ell=1}^L U_\ell = 0, \end{array} \right. \quad (6)$$

where L is the number of electrodes and e_ℓ is the ℓ^{th} electrode; z_ℓ , I_ℓ , and U_ℓ , are the contact impedance, current injected, and electric potential on the ℓ^{th} electrode, respectively; and \hat{n} is the outward unit vector normal to the boundary. Following [34], [35], the forward problem (5) and (6) can be solved using FEM to determine the voltages $U(\sigma)$ on the electrodes for a given conductivity σ . Here we consider the 2D ($n = 2$) case.

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

$$F(\sigma) = \frac{1}{2} \|U(\sigma) - V\|^2 + R(\sigma), \quad (7)$$

where $V = (V_1^{(1)}, \dots, V_L^{(1)}, \dots, V_1^{(K)}, \dots, V_L^{(K)})^T$ in \mathbb{R}^{KL} represents a vector of the measured voltages on each of the L electrodes for K linearly independent current patterns and $U(\sigma) = (U_1^{(1)}(\sigma), \dots, U_L^{(1)}(\sigma), \dots, U_1^{(K)}(\sigma), \dots, U_L^{(K)}(\sigma))^T$ 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, also called the data-fidelity term.

For an initial guess, σ_0 , we use the best constant conductivity fit to the data [36]. Then, (7) can be rewritten as

$$F(\sigma_0 + \delta\sigma) = \frac{1}{2} \|U(\sigma_0 + \delta\sigma) - V\|^2 + R(\sigma_0 + \delta\sigma), \quad (8)$$

with the intent to minimize it with respect to $\delta\sigma$. Solving (8) iteratively leads to the following classic update rule,

$$\sigma_{k+1} = \sigma_k + \delta\sigma_k, \quad (9)$$

given an estimate σ_k . We then iterate until a satisfactory solution for (7) is found.

1) *The Levenberg-Marquardt Algorithm:* When there is no explicit regularization included in the objective function (7), then $R(\sigma) = 0$. The Taylor expansion of (8) 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 with respect to $\delta\sigma$ equal to 0. This yields the update

$$\delta\sigma = -F''(\sigma)^{-1}F'(\sigma), \quad (10)$$

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

$$\begin{aligned} F'(\sigma) &= J(\sigma)^T(U(\sigma) - V) \\ F''(\sigma) &= J(\sigma)^T J(\sigma) + \sum_i U_i''(\sigma)(U_i(\sigma) - V_i), \end{aligned} \quad (11)$$

where $J(\sigma)$ is the Jacobian of the simulated voltages $U(\sigma)$ (e.g., computed by [34], [35]). In Newton's method, the Hessian is computed exactly according to (11). 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 $\lambda_{\text{LM}} I$ where $\lambda_{\text{LM}} \in \mathbb{R}^+$, acting as regularizer of the ill-posed problem. This has the benefit of improving the condition number for the matrix to be inverted in (10) which is of particular importance when the Jacobian is rank deficient. The approximate solution to (8) is then

$$\delta\sigma_{\text{LM}} = -\left(J(\sigma)^T J(\sigma) + \lambda_{\text{LM}} I\right)^{-1} J(\sigma)^T(U(\sigma) - V). \quad (12)$$

We obtain an iterative reconstruction algorithm by using the LM updates in the update rule (9) and a suitable stopping criteria. In this work, we will use $\delta\sigma_{\text{LM}}$ for our learned GCNM.

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 $R(\sigma) = \lambda_{\text{TV}} \sum_i |\mathcal{L}_i \sigma|$ where \mathcal{L} is a sparse matrix representing the discrete gradient, see [18] for details. One often considers a smoothed approximation of Total Variation regularization using $R(\sigma) = \lambda_{\text{TV}} \sum_i \sqrt{(\mathcal{L}_i \sigma)^2 + \gamma}$. The approximate solution to minimize (8) is then

$$\begin{aligned} \delta\sigma_{\text{TV}} &= -(J(\sigma)^T J(\sigma) + \lambda_{\text{TV}} \mathcal{L}^T E^{-1} \mathcal{L})^{-1} \\ &\quad \left(J(\sigma)^T(U(\sigma) - V) + \lambda_{\text{TV}} \mathcal{L}^T E^{-1} \mathcal{L} \sigma \right) \end{aligned} \quad (13)$$

where $\gamma \in \mathbb{R}^+$ is the smoothing parameter that can be varied [18], and $E = \text{diag}\left(\sqrt{(\mathcal{L}\sigma)^2 + \gamma}\right)$ is a diagonal matrix. We will compare the GCNM reconstructions to TV reconstructions using (13).

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
$N_{\text{inclusions}}$	1-4	1-3	3	2-4
σ_{bkg}	[0.40, 0.43]	[0.40, 0.43]	0.424	0.135
σ_{low}	[0.15, 0.25]	[0.15, 0.25]	0.24	0.061
σ_{high}	[0.65, 0.95]	[0.65, 0.95]	0.75	0.323

IV. METHODS

To evaluate the performance of the novel GCNM we looked at a case study for 2D absolute EIT imaging. We trained a GCNM network using the LM update (III-B1) using 400 simulated training samples and 100 simulated testing samples. The simulated samples each contained 1-4 piece-wise constant elliptical inclusions all defined on the same circular mesh (radius 140mm, $L = 32$ equally spaced electrodes of width 20mm and height 20mm). For each simulated conductivity phantom, the forward EIT problem (5) and (6) was solved using FEM with approximately 5,000 triangular elements using adjacent current patterns with current amplitude 2mA. The inverse problem was solved using a FEM mesh with approximately 4,000 elements. A study of the effect of mesh granularity on reconstruction quality across methods was not explored. Figure 2 shows sample simulated phantoms used in the training with conductivity values given in Table I.

Prior to solving the inverse problem, noise was added to the simulated voltages of the training data using

$$v_p = v_p + \nu \text{mean}(|v_p|) \epsilon_p.$$

The parameter ν controls the level of noise, v_p is the vector of voltage measurements on the electrodes for the p^{th} current pattern, and $\epsilon_p \in \mathbb{R}^L$ is a vector of Gaussian random numbers. Unless otherwise stated, 0.5% noise ($\nu = 0.005$) was used in this study. This corresponds to an SNR of about 51dB. The KIT4 system has an SNR of approximately 65.52dB [19] which is well above the value used here.

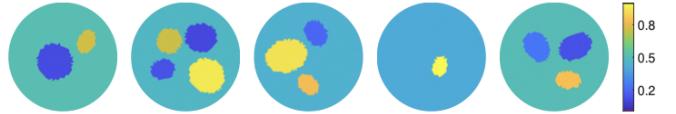


Fig. 2. Examples of "Truths" from the simulated training data.

A. Examples Considered

Several test cases will be used to assess the performance of the trained GCNM. First we consider samples within the distribution of the training data (Case 1). 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 out of distribution cases (Table II). Cases 2-6 explore samples from simulated data outside the distribution of training data. Namely, Cases 2-3 use a chest shaped domain

TABLE II

SUMMARY OF TEST CASES EXPLORED. THE DOMAIN SHAPE, WHETHER OR NOT THE MESH IS THE SAME AS THE ONE USED IN TRAINING, AND OTHER INFORMATION ARE NOTED.

Case	Mesh	Domain	Other info:
1	Old	Circle	Consistent with training data.
2	New	Chest	Perimeter 900mm.
3	New	Chest	Incorrect domain modeling (electrodes).
4	New	Oval	Incorrect domain modeling. True: Chest. Recon: Oval (110mm, 170mm).
5	Old	Circle	'L-shaped' targets with sharp corners.
6	Old	Circle	Varying levels of noise (0%, 1%, and 2%).
7	New	Circle	ACT3 data ($r=150\text{mm}$). Trig current patterns.
	New	Circle	KIT4 data ($r=140\text{mm}$). $L = 16$ electrodes.
	New	Chest	KIT4 data. $L = 16$ electrodes.

with perimeter 900mm. Case 2 assumes correctly known electrode locations but 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. Case 4 explores incorrect domain modeling by reconstructing on an ovular domain (semi-major/minor axes of 170mm and 110mm) while the measured voltages were simulated on the chest shaped domain used in Cases 2-3. This corresponds to a measurement scenario where the patient shape is not accurately known. Case 5 introduces sharp corners by using 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 reconstruct absolute EIT images from experimental data in Case 7 using data collected with the 32 electrode ACT3 system [37], [38] and 16 electrode KIT4 system [39], respectively. The conductivity values for all targets are listed in Table I. The archival ACT3 data used trigonometric current patterns with maximum amplitude 0.2mA and frequency 28.8kHz on a tank of radius 150mm, with 32 equally spaced electrodes of width 25mm and saline height 16mm. The KIT4 data used adjacent current patterns with amplitude 3mA at current frequency 10kHz on 16 approximately equally spaced electrodes. The circular tank had a radius of 140mm, with electrodes of width 25mm, and had two targets a large resistor (0.067 S/m) and small conductor (0.305 S/m) sitting in a saline bath 0.135 S/m of height 45mm. The chest shaped tank had perimeter 1020mm, electrodes of width 20mm, and contained conductive (pink) agar targets of conductivity 0.323 S/m and resistive (white) agar targets of conductivity 0.061 S/m in a saline bath of 0.135 S/m filled to a height of approximately 47mm. See [38] and [29] for additional experimental details for the ACT3 and KIT4 data, respectively.

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 $N(5\mu\Omega\text{m}, 0.5\mu\Omega\text{m})$ when computing the measured voltage. Then, for all reconstructions, including the experimental test

cases, the contact impedance was assumed to be the mean. Note that a device/experiment specific tuning of the contact impedance values for each case could yield improved results. Here we choose to forgo such tuning to emphasize the generalizability of the GCNM.

B. Comparison methods

We compare the learned GCNM results to the classical variational LM and TV approaches as well as a Graph Residual Network (GResNet). For the LM and TV algorithms, the regularization parameters, $\lambda_{\text{LM}} = 10$ for LM and $\lambda_{\text{TV}} = 0.005$ and $\gamma = 1e-8$ for TV, were chosen empirically as those which attained a minimum mean squared error (MSE) in conductivity for a subset of the training data. These remained constant except where specified in Case 7. Additionally, a line search was implemented in the classical methods to select the step length in the direction $\delta\sigma$ that minimizes the objective function (7). Iteration of the classical methods and the GCNM was stopped with $\sigma_{\text{rec}} = \sigma_k$ if the objective function failed to decrease on the following 3 iterations.

1) *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 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 to add the input and output of each block. See [40] for a discussion on skip connections and graph residual network size.

The first iteration of LM algorithm (without the use of a line search) with $\lambda_{\text{LM}} = 0.1$ 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.

C. Training Details for GCNM and GResNet

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 (III-B1) with $\lambda_{\text{LM}} = 0.1$. Each block was then trained minimising the iterate-wise loss function (2) using mini-batches of 10 samples and the Adam optimization method [41] 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 12 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

the CPU. Although a system of 10 blocks was trained, the reconstruction was chosen as $\sigma_{rec} = \sigma_k$ when the objective function (7) failed to decrease in the following three iterations or as the output of the final block $\sigma_{rec} = \sigma_{k_{max}}$ if the first criterion was not met.

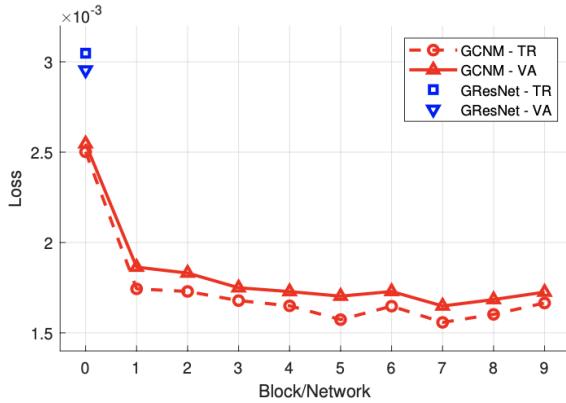


Fig. 3. The final loss values (training and validation) of each block of the GCNM and for the GResNet as a whole.

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 about 6 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 at Network 0 on the horizontal axis because only one Newton-type 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.

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_σ , dynamic range

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

ℓ_1 relative conductivity error

$$RE_{\sigma_k}^{\ell_1} = \frac{\|\sigma_k - \sigma\|_1}{\|\sigma\|_1},$$

as well as the ℓ_2 relative voltage error

$$RE_V^{\ell_2}(\sigma_k) = \frac{\|U(\sigma_k) - V\|_2}{\|V\|_2}. \quad (14)$$

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 of the ‘L-shaped’ targets. Case

6 uses an average over 100 samples at 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 also out of distribution as the GCNM network and GResNet network were not optimized for the specific EIT devices, target types, contact impedances, current patterns (ACT3), or number of electrodes (KIT4).

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 iterative method. Note that the GCNM outperforms the other methods in visual sharpness as the only method to clearly separate all four and all three targets for Samples A and B, respectively. Table III shows that the GCNM required the lowest number of iteration on average and achieved the lowest conductivity errors MSE_σ and $RE_\sigma^{\ell_1}$ of the iterative schemes. As expected, the classical methods LM and TV both achieved lower relative voltage errors $RE_V^{\ell_2}$ as they minimize over the voltage error but both required more than 15 iterations on average with the LM averaging the full 20 allowed. It is not unexpected that the GCNM and GResNet do not achieve the lowest relative voltage error as the networks were optimized to reduce the MSE in the conductivity rather than the voltage data. The TV method achieved the best dynamic range closely followed by the GCNM. The GResNet provided significantly improved reconstructions over the initialization in a single application but often overestimated the dynamic range as can be seen in sample B of Figure 4. 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 III. Beginning with Case 2, the chest shaped domain, again the GCNM and TV produce the sharpest reconstructions with TV outperforming the other methods in all but the relative voltage error (LM) but requiring an average of 15.8 iterations. In Sample B, only the GCNM and GResNet were able to clearly separate all four targets. By the metrics, the GCNM performed second-best in MSE_σ , $RE_\sigma^{\ell_1}$ and DR in an average of 4.4 iterations. Recall that the learned networks were 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. In Case 3, for which the electrode locations are incorrectly modeled, visually

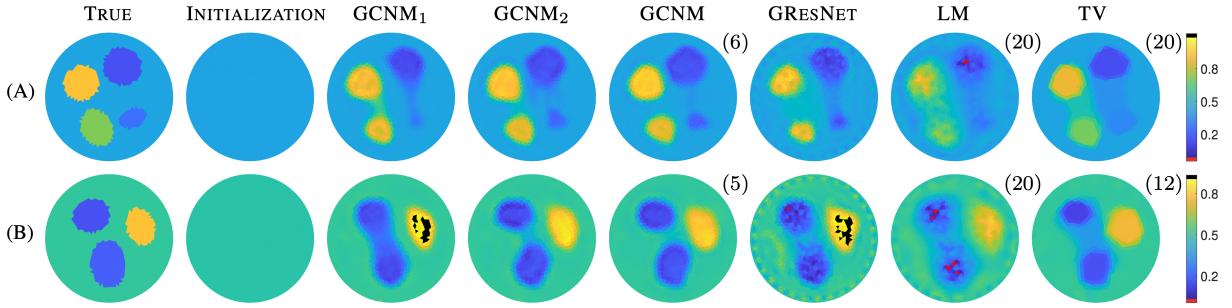


Fig. 4. Case 1: Results for two samples of testing data consistent with, but not included in, the training data. The initialization, first two iterations of the GCNM, and the GCNM reconstruction are compared to the reconstructions from GResNet, LM, and TV. Black and red colors represent going 20% over or under the color bar range, respectively. Iteration numbers chosen by the stopping criterion are shown at the top right for each iterative method.

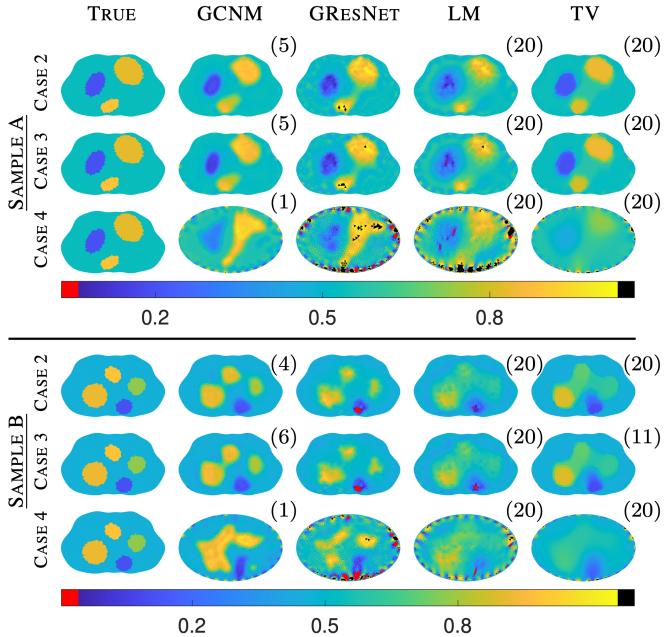


Fig. 5. Results for Cases 2-4: exploring a chest shaped domain, and incorrect domain modeling

the results are similar to that of Case 2 with the GCNM and GResNet again obtaining clearly identifiable separated targets requiring an average of 3.9 iterations and a single application, respectively, and the GCNM obtaining the best dynamic range and second-best $\text{RE}_{\sigma}^{\ell_1}$. The traditional LM and TV methods slightly outperformed the learned methods in MSE_{σ} and $\text{RE}_{\sigma}^{\ell_2}$ again requiring over 16 iterations on average.

In Case 4 we reconstruct assuming an ovular domain instead of the true chest shaped domain. Typically, mis-modelling of the domain for absolute EIT results in large artefacts near the boundary of the domain where the mis-modelling has occurred. 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. While Figure 5 shows that none of the methods are able to resolve all the targets for these samples, the GCNM appears the most stable followed by GResNet. The LM method resolves some but not all of the targets but suffers additional artefacts in

the center. The TV reconstructions are dominated by domain mismatch modeling errors at the electrodes resulting in very low contrast, non-separated, reconstructions in the center of the images. Large magnitude errors in pixels at the boundary of the domain led to large dynamic range values that are reflected in Table IV with an average of 444.7%. By the metrics, the GCNM outperformed all methods in an average of just one iteration, except for the relative voltage error. As the domain model is such a poor match, we cannot expect the iterates to minimize the objective function. To overcome this, one could consider different stopping criteria for GCNM. The LM method preformed second-best according to the metrics. We point out that 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.

Figure 6 explores Case 5 where the target inclusion is an ‘L’, a notoriously challenging target for EIT. 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 GResNet image contains significant fluctuations at the domain boundary, as does LM. The GCNM contains a low conductivity artefact near the inner corner of the ‘L’, as does the TV image, but still resulted in the sharpest corner reconstruction and best metrics (see Table III) aside from the voltage error.

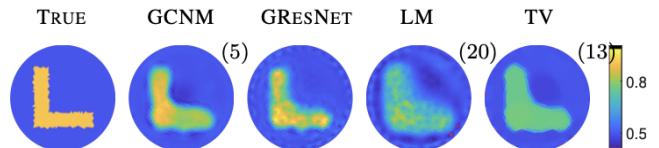


Fig. 6. Results for Case 5: An L-shaped out of distribution inclusion.

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 (51dB SNR). While all the methods performed admirably on this test, the GCNM was the only method able to separate the three targets clearly, up through 1% noise (45dB SNR), whereas the other methods blurred the two resistive objects.

TABLE III

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	$RE_\sigma^{\ell_1}$	$RE_V^{\ell_2}$	DR(%)
1	GCNM	4.8	1.71e-03	3.34e-02	9.44e-03	113.3
	GResNet	1.0	2.95e-03	5.75e-02	1.68e-02	131.2
	LM	20.0	3.96e-03	7.24e-02	2.68e-03	124.9
	TV	15.8	2.62e-03	4.40e-02	2.66e-03	89.8
2	GCNM	4.4	3.21e-03	5.21e-02	2.98e-03	126.9
	GResNet	1.0	5.56e-03	7.78e-02	1.19e+00	199.4
	LM	20.0	4.25e-03	8.08e-02	2.93e-03	154.8
	TV	15.8	2.46e-03	4.49e-02	4.78e-03	101.6
3	GCNM	3.9	5.22e-03	7.45e-02	5.47e-02	171.6
	GResNet	1.0	6.29e-03	8.82e-02	2.47e-01	219.1
	LM	20.0	5.11e-03	9.66e-02	2.79e-03	190.7
	TV	16.6	4.27e-03	6.72e-02	3.12e-03	182.4
4	GCNM	1.0	2.47e-02	2.30e-01	1.68e-01	200.8
	GResNet	1.0	6.94e-02	3.47e-01	1.38e+01	773.0
	LM	20.0	3.55e-02	2.80e-01	8.77e-03	361.5
	TV	20.0	5.98e-02	3.12e-01	8.01e-02	444.7
5	GCNM	5.0	1.59e-03	3.06e-02	1.26e-02	134.5
	GResNet	1.0	2.08e-03	4.78e-02	1.44e-02	169.9
	LM	20.0	6.87e-03	1.14e-01	2.42e-03	292.5
	TV	16.0	2.73e-03	4.87e-02	2.55e-03	102.2

It should be noted that 2% noise (39dB SNR) is an extreme amount of noise for EIT reconstruction with systems capable of SNR of 65dB, comparable to 0.1% noise [19]. Nevertheless, the GCNM images are not corrupted by errors, and as shown in Figure 8, adjusting the reconstruction parameter λ_{LM} in the update $\delta\sigma$ in (III-B1) resulted in even fewer artefacts and more clearly defined targets.

Table IV demonstrates that the GCNM and GCNM2 methods outperformed the other methods for all non-voltage metrics except the dynamic range in the 1% noise case where TV was superior and GCNM2 was second. The GResNet, in just one iteration, performed well across noise levels while the GCNM and GCNM2 used approximately 5 iterations and TV averaged more than 15 (for $\nu > 0$). The LM method ranked in the top two best for $RE_V^{\ell_2}$ but fell behind in the remaining metrics, as well as visual sharpness requiring, on average, the full 20 iterations.

Note that above 1.0% noise, using a larger regularization parameter $\lambda_{LM} = 5$ resulted in reconstructions with lower MSE and relative ℓ_1 error with GCNM2 vs GCNM. This is a key feature of the proposed GCNM as a model-based approach, that even after training a network with a fixed regularization parameter one can adjust and control the performance if noise levels and distributions change, offering greater flexibility and control.

B. Experimental Results

We considered five data sets from two different EIT machines using three tanks, 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 specific EIT machine hardware. Here, the forward model was not tuned to the

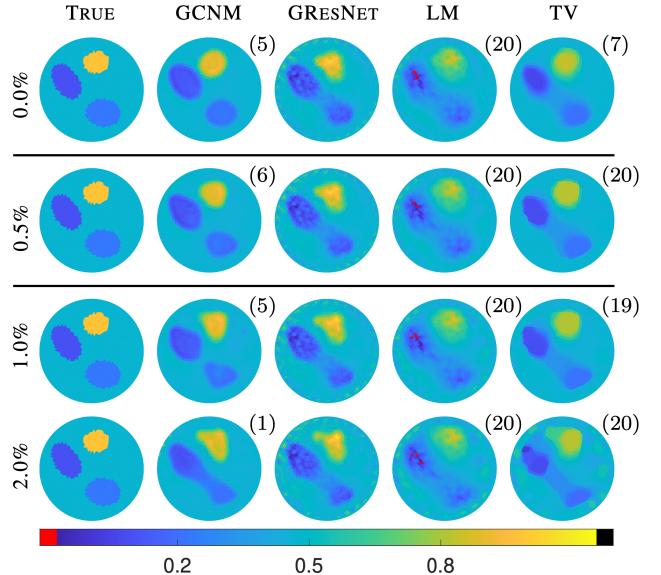


Fig. 7. Results for Case 6: Varying the noise level. Note that samples with 0.5% noise were used during training.

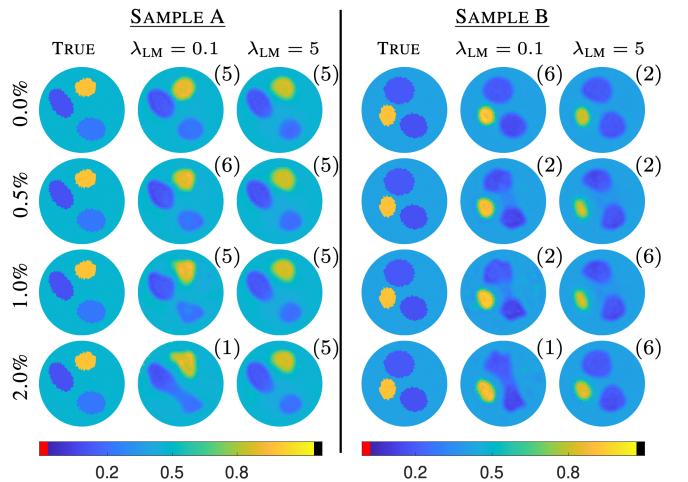


Fig. 8. Case 6: GCNM reconstructions for two representative samples with varying levels of added noise shown with two different values for the regularization parameter λ_{LM} in (III-B1). Note that the parameter was only changed when testing; all training was done with $\lambda = 0.1$.

respective EIT machines and the same contact impedance was used for all electrodes, the mean of those used in the training.

As the noise distributions and parameter ranges are different for the experimental data when compared to the training data, it is natural that the regularization parameters would need to be tuned for all methods. For the ACT3 reconstructions, $\lambda_{LM} = 10$ for the GCNM and GResNet, $\lambda_{LM} = 100$ for the LM algorithm, and $\lambda_{TV} = 0.02$ for the TV method. All of the KIT4 samples used $\lambda_{LM} = 100$ for the GCNM and GResNet, $\lambda_{LM} = 50$ for the LM algorithm, and $\lambda_{TV} = 0.02$ for the TV method. The TV methods for both experimental setups used $\gamma = 1e-8$, the same as was used for the simulated data. Here it is important to emphasize again that the networks of the learned methods were not retrained with these new regularization parameters. The ability to adjust the regularization parameters used in the

TABLE IV

METRICS FOR VARYING THE NOISE LEVEL AND ADJUSTING λ_{LM} . THE METHOD "GCNM" USED $\lambda = 0.1$ FOR COMPUTING $\delta\sigma$ (SAME AS TRAINING) WHILE "GCNM2" USED $\lambda = 5$.

Case	Method	Its	MSE_σ	$RE_\sigma^{\ell_1}$	$RE_V^{\ell_2}$	DR(%)
0.0%	GCNM	4.7	1.56e-03	3.04e-02	9.47e-03	112.7
	GCNM2	4.7	3.06e-03	4.20e-02	9.18e-03	94.7
	GResNet	1.0	2.88e-03	5.60e-02	1.62e-02	128.4
	LM	20.0	3.90e-03	7.00e-02	6.09e-04	122.3
	TV	11.1	2.53e-03	4.20e-02	3.47e-04	88.2
0.5%	GCNM	4.8	1.72e-03	3.34e-02	9.44e-03	113.3
	GCNM2	4.7	3.06e-03	4.21e-02	9.62e-03	95.0
	GResNet	1.0	2.96e-03	5.76e-02	1.68e-02	131.2
	LM	20.0	3.96e-03	7.24e-02	2.68e-03	124.9
	TV	15.8	2.62e-03	4.41e-02	2.66e-03	89.8
1.0%	GCNM	4.2	2.22e-03	4.00e-02	1.17e-02	117.7
	GCNM2	4.8	3.05e-03	4.24e-02	1.09e-02	95.1
	GResNet	1.0	3.21e-03	6.21e-02	1.83e-02	135.0
	LM	20.0	4.15e-03	7.79e-02	5.25e-03	130.5
	TV	18.1	2.99e-03	5.41e-02	5.26e-03	98.3
2.0%	GCNM	1.6	3.22e-03	4.90e-02	2.43e-02	118.2
	GCNM2	5.0	3.08e-03	4.30e-02	1.45e-02	96.0
	GResNet	1.0	4.22e-03	7.69e-02	2.82e-02	154.4
	LM	20.0	4.90e-03	9.27e-02	1.04e-02	147.4
	TV	19.7	4.84e-03	8.92e-02	1.04e-02	135.3

update terms after training adds to the flexibility of the GCNM and GResNet. To compute the MSE_σ , $RE_\sigma^{\ell_1}$ and $RE_V^{\ell_2}$ metrics, a simulated truth image was created using the photographs from the experiments and the measured conductivity values reported for the agar targets and saline. Results are presented in Table V. Note that the LM and TV methods required all 20 iterations whereas the GCNM ranged from 3 to 6 iterations across samples.

For the ACT3 data, each reconstruction method recovered the heart and lungs' shapes but failed to completely separate the large resistive lungs. The GCNM and TV methods produced the most distinct target boundaries and the lowest MSE_σ but overestimated the dynamic range. The GCNM obtained the lowest relative ℓ_1 error $RE_\sigma^{\ell_1}$ and required 3 iterations. The GResNet had the least distinct targets visually and significantly underestimating the conductivity of the heart, but the best DR. Similarly to the simulated data cases, the LM and TV methods outperformed the learned methods in relative voltage error RE_V requiring 20 iterations each.

The background conductivity in the KIT4 data was 0.135 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 input data into scale for the trained networks, the network inputs σ_k and $\delta\sigma_k$ were scaled up by a factor of 3 and the outputs σ_{k+1} were scaled back down by a factor of 3; see Figure 9.

Beginning with the circular tank, both targets were well located by the all methods for the KIT4-A data. The GResNet and GCNM obtained the best MSE_σ , $RE_\sigma^{\ell_1}$, and DR. There are clear artefacts present in bottom right corner of the tank for the LM and TV images. More broadly, for all of the KIT4 reconstructions, all methods contain an artefact that region suggesting a poor matching of the forward model and/or contact impedance in that region. A more device-specific tuning of the forward solver to the experimental KIT4 device and contact impedances may improve the results.

Next, for the KIT4-B data on the chest domain, the two resistive lungs and conductive heart are visible for all methods, each containing slightly different visual artefacts. The GResNet obtained the lowest MSE_σ and $RE_\sigma^{\ell_1}$ and the GCNM the best dynamic range. The LM reconstruction contained the least distinct targets. Sample KIT4-C contained a conductive portion in the bottom left (viewer's) lung. The LM and TV method significantly overestimated that portion as can be seen by the black pixels in Figure 9 but as usual achieved the lowest relative voltage error $RE_V^{\ell_2}$. The GResNet achieved the best MSE_σ and $RE_\sigma^{\ell_1}$ but the top of the viewer's left lung not clearly visible or defined. The GCNM overestimated the size of the conductive portion of the lung but obtained the second best MSE_σ , $RE_\sigma^{\ell_1}$ and the best DR. Lastly, reconstructions for sample KIT4-D from the GCNM, LM, and TV methods all show the correct targets with the GResNet underestimating the contrast of the viewer's left lung. The GCNM and GResNet outperformed the classic methods across metrics except for the relative voltage error. Summarizing across the KIT4 samples, the GCNM obtained the best DR for all KIT4 samples, and routinely performed in the top two for both MSE_σ and $RE_\sigma^{\ell_1}$, while only requiring 3-6 iterations compared to the 20 required by the classic methods.

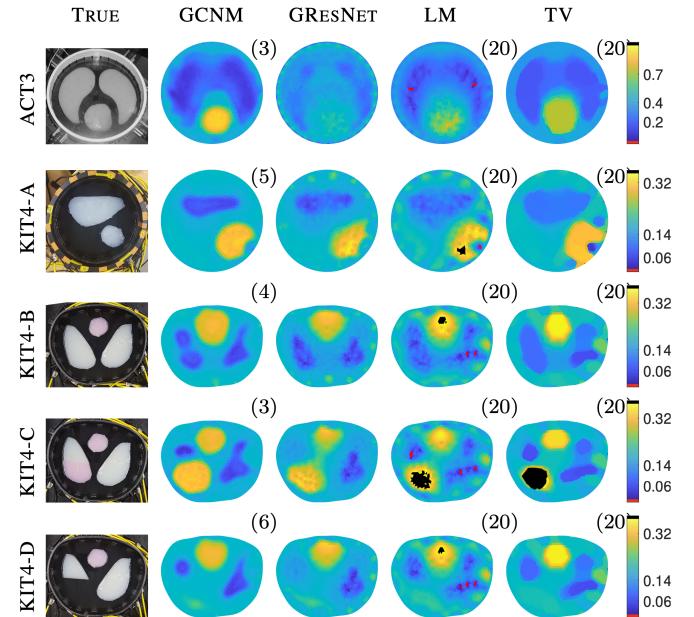


Fig. 9. Results for Case 7: Experimental Data. Column 1 contains the experimental setups. The ACT3 sample is shown with a colorbar range of [0, 1] while the KIT4 samples are shown with colorbar ranges of +/- 20% of the true range.

C. Further Discussion

All computations were performed on the hardware described in Section IV-C. Computational costs per sample were as follows: 2.4 seconds per iteration of GCNM, about 12.3 seconds per iteration of LM and TV, and 4.6 seconds per reconstruction using GResNet. Each forward solve took approximately 1.1 seconds and each Jacobian computation approximately 1.3 seconds. We emphasize that forward solver and the individual algorithms were not optimized for speed in this study. Nonetheless, the GCNM provided a significant speedup in

TABLE V

TABLE OF METRICS FOR EACH OF THE EXPERIMENTAL SAMPLES FROM ACT3 AND KIT4.

Case	Method	Its	MSE_σ	$RE_\sigma^{\ell_1}$	$RE_V^{\ell_2}$	DR(%)
ACT3	GCNM	3	1.11e-02	1.76e-01	6.58e-02	167.7
	GResNet	1	1.77e-02	2.60e-01	1.96e-01	104.8
	LM	20	1.37e-02	2.56e-03	4.84e-03	166.6
	TV	20	1.11e-02	2.11e-01	4.34e-03	112
KIT4-A	GCNM	5	2.40e-03	3.07e-01	3.38e-02	110.8
	GResNet	1	2.16e-03	2.95e-01	3.46e-02	124
	LM	20	2.87e-03	3.27e-01	3.13e-02	180.6
	TV	20	4.28e-03	3.93e-01	3.13e-02	223.8
KIT4-B	GCNM	4	3.53e-03	3.96e-01	3.10e-02	104.6
	GResNet	1	2.91e-03	3.56e-01	3.91e-02	120.5
	LM	20	3.35e-03	3.91e-01	2.66e-02	166.1
	TV	20	3.95e-03	4.26e-01	2.66e-02	156.1
KIT4-C	GCNM	3	3.11e-03	3.74e-01	3.27e-02	104.8
	GResNet	1	2.91e-03	3.55e-01	3.75e-02	124.9
	LM	20	3.34e-03	3.84e-01	2.72e-02	157.6
	TV	20	3.76e-03	4.10e-01	2.72e-02	150.7
KIT4-D	GCNM	6	3.11e-03	2.93e-01	2.94e-02	117.5
	GResNet	1	2.91e-03	3.07e-01	3.21e-02	126.1
	LM	20	3.23e-03	3.23e-01	2.58e-02	223.1
	TV	20	3.41e-03	3.27e-01	2.58e-02	169.9

reconstruction time by requiring approximately five iterations (at 2.4 sec/iter) instead of more than 15 for TV and the full 20 for LM, and without the need to perform a line search at each iteration. Therefore, the approximate computation time per image for GCNM was 12 seconds the same approximate cost of a single iteration of LM and TV, as implemented in this study. The GCNM reliably produced images with clearly identifiable targets, and the best or second best metrics across methods for the simulated and experimental data considered in this work, 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. CONCLUSION

We successfully introduced a novel 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 improved robustness with respect to noise and 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 can still be used later on different domain shapes, without the need for sub-optimal embeddings to rectangular domains. 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 and three tanks. 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-type 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.

ACKNOWLEDGEMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. We also thank the EIT groups at RPI and UEF for sharing the respective experimental data sets, in particular Ville Kolehmainen of UEF for his helpful discussions on the TV method.

REFERENCES

- [1] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb, “Solving inverse problems using data-driven models,” *Acta Numerica*, vol. 28, pp. 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*, vol. 44, no. 10, pp. 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*, vol. 26, no. 9, pp. 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*, vol. 33, no. 12, p. 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*, vol. 555, no. 7697, pp. 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*, vol. 36, no. 9, p. 094004, 2020.
- [9] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: <http://arxiv.org/abs/1609.02907>
- [10] 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*, vol. 37, no. 6, pp. 1382–1393, 2018.
- [11] A. Hauptmann, J. Adler, S. R. Arridge, and O. Öktem, “Multi-scale learned iterative reconstruction,” *IEEE Transactions on Computational Imaging*, 2020.
- [12] 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*, vol. 41, 06 2020.

- [13] S. Lunz, A. Hauptmann, T. Tarvainen, C.-B. Schönlieb, and S. Arridge, “On learned operator correction in inverse problems,” *SIAM Journal on Imaging Sciences*, vol. 14, no. 1, pp. 92–127, 2021.
- [14] D. Smyl, T. N. Tallman, J. A. Black, A. Hauptmann, and D. Liu, “Learning and correcting non-gaussian model errors,” *Journal of Computational Physics*, vol. 432, p. 110152, 2021.
- [15] A.-P. Calderón, “On an inverse boundary value problem,” in *Seminar on Numerical Analysis and its Applications to Continuum Physics (Rio de Janeiro, 1980)*. Rio de Janeiro: Soc. Brasil. Mat., 1980, pp. 65–73.
- [16] C. Li, M. Karamehmedovic, E. Sherina, and K. Knudsen, “Levenberg-marquardt algorithm for acousto-electric tomography based on the complete electrode model,” 2020.
- [17] W. R. B. Lionheart, “EIT reconstruction algorithms: pitfalls, challenges and recent developments,” *Physiological Measurement*, vol. 25, no. 1, pp. 125–142, feb 2004. [Online]. Available: <https://doi.org/10.1088/0967-3334/25/1/021>
- [18] A. Borsic, B. Graham, A. Adler, and W. Lionheart, “In vivo impedance imaging with total variation regularization,” *IEEE Transactions on Medical Imaging*, vol. 29, pp. 44–54, 2010.
- [19] 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*, vol. 30, no. 2, pp. 231–242, 2011.
- [20] 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*, vol. 9, no. 1, pp. 211–229, 2015.
- [21] L. Borcea, “Electrical impedance tomography,” *Inverse Problems*, vol. 18, pp. 99–136, 2002.
- [22] J. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical Applications*. SIAM, 2012.
- [23] A. I. Nachman, “Global uniqueness for a two-dimensional inverse boundary value problem,” *Annals of Mathematics*, vol. 143, pp. 71–96, 1996.
- [24] 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*, vol. 39, no. 6, p. 064005, 2018. [Online]. Available: <http://stacks.iop.org/0967-3334/39/i=6/a=064005>
- [25] J. L. Mueller and S. Siltanen, “The d-bar method for electrical impedance tomography—demystified,” *Inverse Problems*, vol. 36, no. 9, p. 093001, sep 2020. [Online]. Available: <https://doi.org/10.1088/1361-6420/aba2f5>
- [26] M. Dodd and J. L. Mueller, “A real-time D-bar algorithm for 2-D electrical impedance tomography data,” *Inverse Problems and Imaging*, vol. 8, no. 4, pp. 1013–1031, 2014. [Online]. Available: <http://aimsces.org/journals/displayArticlesnew.jsp?paperID=10528>
- [27] 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*, vol. 36, no. 9, pp. 1868–1875, 2017.
- [28] S. J. Hamilton and A. Hauptmann, “Deep D-Bar: Real-time electrical impedance tomography imaging with deep neural networks,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 10, p. 2367–2377, Oct 2018. [Online]. Available: <http://dx.doi.org/10.1109/TMI.2018.2828303>
- [29] 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*, vol. 40, no. 7, p. 074002, jul 2019. [Online]. Available: <https://doi.org/10.1088/1361-6579/ab21b2>
- [30] Z. Wei, D. Liu, and X. Chen, “Dominant-current deep learning scheme for electrical impedance tomography,” *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 9, pp. 2546–2555, 2019.
- [31] 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*. Springer, 2015, pp. 234–241.
- [32] 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*, pp. 1–1, 2021.
- [33] E. Somersalo, M. Cheney, and D. Isaacson, “Existence and uniqueness for electrode models for electric current computed tomography,” *SIAM Journal on Applied Mathematics*, vol. 52, no. 4, pp. 1023–1040, 1992. [Online]. Available: <http://www.jstor.org/stable/2102189>
- [34] J. Kaipio, V. Kolehmainen, E. Somersalo, and M. Vauhkonen, “Statistical inversion and monte carlo sampling methods in electrical impedance tomography,” *Inverse Problems - INVERSE PROBL*, vol. 16, 07 2000.
- [35] M. Vauhkonen, “Electrical impedance tomography and prior information,” 1997.
- [36] 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*, vol. 44, no. 11, pp. 1051 –1060, Nov. 1997.
- [37] 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*, vol. 41, no. 8, pp. 713–722, August 1994.
- [38] 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*, vol. 23, pp. 821–828, 2004.
- [39] 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.
- [40] J. Zhang and L. Meng, “Gresnet: Graph residual network for reviving deep gnns from suspended animation,” 2019.
- [41] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2015.