



DATA-DRIVEN APPROACHES FOR ELECTRICAL IMPEDANCE TOMOGRAPHY IMAGE SEGMENTATION FROM PARTIAL BOUNDARY DATA

ALEXANDER DENKER^{✉*1}, ŽELJKO KERETA^{✉2}, IMRAJ SINGH^{✉2},
TOM FREUDENBERG^{✉1}, TOBIAS KLUTH^{✉1},
Peter Maass^{✉1} and Simon Arridge^{✉2}

¹Center of Industrial Mathematics, University of Bremen, Germany

²Department of Computer Science, University College London, England

ABSTRACT. Electrical impedance tomography (EIT) plays a crucial role in non-invasive imaging, with both medical and industrial applications. In this paper, we present three data-driven reconstruction methods for EIT imaging, that were submitted to the Kuopio tomography challenge 2023 (KTC2023). First, we introduce a post-processing method, which achieved first place at KTC2023. Further, we present a fully learned and a conditional diffusion approach. All three methods are based on a similar neural network backbone and were trained using a synthetically generated data set, providing a fair comparison of these different data-driven reconstruction methods.

1. Introduction. Electrical impedance tomography (EIT) is an imaging modality that uses electrical measurements taken on the boundary of an object that are used to recover electrical properties of its interior. In this paper we consider the reconstruction of conductivity, for which a series of currents are applied through electrodes attached to the object's boundary. The electrodes measure the resulting voltages, which are used to produce an image of the conductivity. EIT has numerous applications for example in medical diagnostics [5] or in non-destructive testing [18].

There are several mathematical models for the physics of the EIT measurement process. Let Ω be the domain, $\partial\Omega$ its boundary and $\cup_{l=1}^L e_l \subset \partial\Omega$ the set of $L \in \mathbb{N}$ electrodes attached to the boundary. The electric potential $u \in H_1(\Omega)$ is derived from Maxwell's equations and is governed by

$$-\nabla \cdot (\sigma \nabla u) = 0 \quad \text{in } \Omega, \quad (1a)$$

where $\sigma \in L^\infty(\Omega)$ is the conductivity distribution. The complete electrode model (CEM) [26] describes a realistic formulation of boundary conditions when a current is applied to the electrodes. First, the boundary is decomposed into two components: the electrodes e_l (identified with the part of the boundary they are attached to) and the remaining space between the electrodes, $\partial\Omega \setminus \cup_{l=1}^L e_l$. Second, the electrical conduction between the electrode and the corresponding part of the boundary

2020 *Mathematics Subject Classification.* Primary: 65M32, 68T07; Secondary: 92C55.

Key words and phrases. Inverse problems, deep learning, electrical impedance tomography.

*Corresponding author: Alexander Denker.

is accounted for. For a given current injection pattern $\mathbf{l} = (I_1, \dots, I_L)^\top \in \mathbb{R}^L$, the resulting model can be written as

$$\begin{cases} u + z_l \sigma \frac{\partial u}{\partial \nu} = U_l, & \text{on } e_l, \text{ for } l = 1, \dots, L, \\ \sigma \frac{\partial u}{\partial \nu} = 0, & \text{on } \partial\Omega \setminus \cup_{l=1}^L e_l, \\ \int_{e_l} \sigma \frac{\partial u}{\partial \nu} ds = I_l, & \text{on } e_l, \text{ for } l = 1, \dots, L, \end{cases} \quad (1b)$$

where $(z_1, \dots, z_L)^\top \in \mathbb{R}^L$ are the contact impedances, quantifying the effect of the resistive layer formed at the contact point of electrodes and the boundary, and $\mathbf{U} = (U_1, \dots, U_L)^\top \in \mathbb{R}^L$ is the voltage at the electrodes. CEM includes conservation of charge and a mean-free current constraint for the potentials, i.e.,

$$\sum_{l=1}^L I_l = 0 \text{ and } \sum_{l=1}^L U_l = 0, \quad (1c)$$

respectively. Equations (1a) to (1b) describe a single current injection pattern. In practice, several injection patterns are applied and corresponding electrode measurements are obtained. We denote the voltages and charges for the k -th injection pattern by $\mathbf{U}^{(k)}$ and $\mathbf{l}^{(k)}$, where $k = 1, \dots, K$. By $\mathbf{U} = (\mathbf{U}^{(1)}, \dots, \mathbf{U}^{(K)})$ and $\mathbf{I} = (\mathbf{l}^{(1)}, \dots, \mathbf{l}^{(K)})$ we denote stacked \mathbb{R}^{KL} vectors of voltages and charges at all electrodes and for all current patterns. Let further $\mathbf{F}(\sigma) = (\mathbf{F}^{(1)}(\sigma), \dots, \mathbf{F}^{(K)}(\sigma))^\top$ be the corresponding forward operator, applied to conductivity σ , for all electrodes and all current patterns. The resulting non-linear inverse problem can be written as

$$\mathbf{F}(\sigma)\mathbf{I} = \mathbf{U}, \quad (2)$$

where the goal is to reconstruct σ given electrode measurements \mathbf{U} .

1.1. KTC 2023 challenge. We outline three methods submitted to the Kuopio Tomography Challenge 2023 (KTC2023) [21]. The goal of KTC2023 was to reconstruct segmentation maps of resistive and conductive inclusions from partial boundary measurements. The measurements were acquired from a plastic circular tank with 32 equispaced stainless electrodes attached to the boundary. All 32 electrodes were used to collect the measurements and 16 electrodes (the odd numbered ones) were used for current injection patterns. For each current injection pattern voltages are taken between adjacent electrodes, resulting in 31 measurements per injection pattern. Five types of injection patterns are considered. An illustration of the measurement tank, electrodes, and injection patterns can be found in Figure 1.1.

The challenge was divided into 7 difficulty levels. The first level includes data from all 32 electrodes. Two additional electrodes are removed, proceeding counter-clockwise, in each subsequent challenge level, thus reducing the number of measurements and the number of applied current injection patterns decrease with each level. For example, in level 2 electrodes 1, 2 are removed and in the final level 7 electrodes 1 to 12 are removed. This means that all measurements from the upper left boundary are removed, cf. Figure 1.1, making the reconstruction of the conductivity map and segmentation map increasingly ill-posed for higher levels.

1.2. Our contribution. We propose three data-driven reconstruction methods to tackle the reconstruction of segmentation maps from partial EIT measurements:

- **FC U-Net:** a fully-learned approach that reconstructs directly from measurements, see Section 3.1.1 .

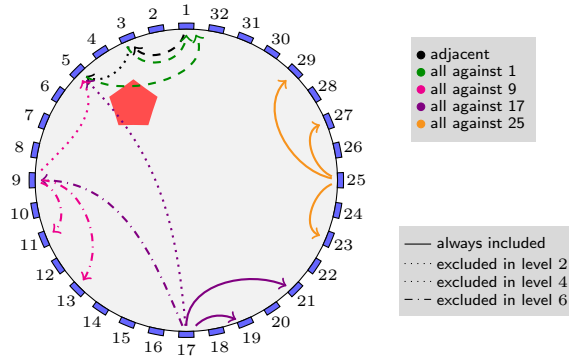


FIGURE 1. An illustration of the EIT measurement tank (Ω), the electrodes e_l , with a sample of the injection patterns. In black we show the *adjacent* injections; in green *all against* e_1 ; in pink *all against* e_9 ; in magenta *all against* e_{17} ; in orange *all against* e_{25} . Dashed injections are removed in the 2nd challenge level; dotted ones in the 4th; dash dotted in the 6th.

- **Post-Processing:** an approach that reconstructs from an initial reconstruction method, see Section 3.1.2.
- **Conditional-Diffusion:** a conditional diffusion approach that aims to directly model the posterior given initial reconstructions, see Section 3.2.1.

FC U-Net and Post-Processing are learned reconstruction approaches, whereas Conditional-Diffusion is an approach based on conditional generative modelling. The three proposed methods achieved the three highest scores at KTC2023, with Post-Processing performing the best overall. Additionally, the three approaches use a similar U-Net architecture with a comparable number of parameters and are trained using the same dataset of generated phantoms and simulated measurements.

1.3. Related work. Deep learning post-processing and fully learned reconstruction are two well-known data-driven approaches for medical image reconstruction [2]. Both of these frameworks have been applied to EIT image reconstruction. Our FC U-Net follows the model proposed by Chen et al. [6]. However, in Section 3.1.1 we propose a novel two-step training method for this fully learned model. Post-processing methods have been applied to EIT, e.g., by Martin et al. [19]. We extend this post-processing framework to deal with the different levels, corresponding to partial EIT measurements with increasing severity, of KTC2023. To the best of our knowledge, our method is the first application of a conditional diffusion model to real-world EIT data. Recently, Wang et al. [31] propose an unconditional diffusion model for EIT and make use of the sampling framework proposed by Chung et al. [8] to enable conditional sampling. However, they only evaluated their approach on simulated data with two or four circular inclusions, whereas we evaluate our approach on real measurements of complex objects.

2. Linearised reconstruction. EIT reconstruction deals with the recovery of the conductivity σ from a set of electrode measurements \mathbf{U} . A common technique is to linearise the non-linear forward operator \mathbf{F} around a homogeneous background

conductivity σ_{ref} and reconstruct a perturbation $\delta\sigma$ to this background [7, 10, 17]. The linearised forward operator is given as

$$\tilde{\mathbf{F}}(\sigma_{\text{ref}} + \delta\sigma; \sigma_{\text{ref}})\mathbf{I} := \mathbf{F}(\sigma_{\text{ref}})\mathbf{I} + \mathbf{J}_{\sigma_{\text{ref}}}\delta\sigma, \quad (3)$$

where $\mathbf{J}_{\sigma_{\text{ref}}} := \nabla\mathbf{F}(\sigma_{\text{ref}})$ is the Jacobian evaluated at the background conductivity. We further assume access to measurements \mathbf{U}_{ref} of the empty water tank, such that $\mathbf{F}(\sigma_{\text{ref}})\mathbf{I} = \mathbf{U}_{\text{ref}}$.

We then define the measurement perturbation as $\delta\mathbf{U} := \mathbf{U} - \mathbf{U}_{\text{ref}}$. The corresponding linearised problem is then to determine $\delta\sigma$ from

$$\mathbf{J}_{\sigma_{\text{ref}}}\delta\sigma = \delta\mathbf{U}. \quad (4)$$

The perturbation $\delta\sigma \in \mathbb{R}^M$ is discretised by M coefficients of a piecewise constant finite element expansion. The finite element approximations of the Jacobian, and forward operator, are discussed in Sections 2.1 and 2.2.

To solve Eqn. (4) we use the framework of variational regularisation as

$$\widehat{\delta\sigma} := \underset{\delta\sigma}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{J}_{\sigma_{\text{ref}}}\delta\sigma - \delta\mathbf{U}\|_{\Sigma^{-1}}^2 + \alpha\mathcal{J}(\delta\sigma), \quad (5)$$

where we assume a Gaussian noise model $\delta\mathbf{U} \sim \mathcal{N}(\mathbf{0}, \Sigma)$, and $\mathcal{J} : \mathbb{R}^M \rightarrow \mathbb{R}_{\geq 0}$ is a regulariser. We consider Tikhonov-type regularisers of the form $\mathcal{J}(\delta\sigma) = \frac{1}{2} \|\mathbf{L}\delta\sigma\|_2^2$. For this choice of a regulariser we can recover the solution to Eqn. (5) as

$$\widehat{\delta\sigma} = (\mathbf{J}_{\sigma_{\text{ref}}}^\top \Sigma^{-1} \mathbf{J}_{\sigma_{\text{ref}}} + \mathbf{L}^\top \mathbf{L})^{-1} \mathbf{J}_{\sigma_{\text{ref}}}^\top \Sigma^{-1} \delta\mathbf{U}. \quad (6)$$

The matrix $(\mathbf{J}_{\sigma_{\text{ref}}}^\top \Sigma^{-1} \mathbf{J}_{\sigma_{\text{ref}}} + \mathbf{L}^\top \mathbf{L})^{-1}$ can be computed offline, leading to a computationally cheap reconstruction method necessary for training the post-processing and conditional diffusion networks. In the following sections, we will discuss the implementation of the forward operator, the computation of the Jacobian $\mathbf{J}_{\sigma_{\text{ref}}}$ and the choice of regulariser \mathcal{J} .

2.1. Forward operator. The EIT forward operator \mathbf{F} defining CEM is non-linear. Evaluating \mathbf{F} for a given a conductivity σ requires solving the differential equations in (1b). We approximate the solution by applying the finite element method to the weak formulation of the CEM, see e.g. [16]. To incorporate the conservation of charge we introduce a Lagrange multiplier $\lambda \in \mathbb{R}$. The weak formulation of the CEM then reads: find $(u, \mathbf{U}, \lambda) \in H^1(\Omega) \times \mathbb{R}^L \times \mathbb{R}$ such that

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v \, dx + \sum_{l=1}^L \frac{1}{z_l} \int_{e_l} (u - U_l)(v - V_l) \, ds + \sum_{l=1}^L (\lambda V_l + \nu U_l) = \sum_{l=1}^L I_l V_l, \quad (7)$$

for all $(v, \mathbf{V}, \nu) \in H^1(\Omega) \times \mathbb{R}^L \times \mathbb{R}$, with $\mathbf{V} = (V_1, \dots, V_L)^\top$ and $\mathbf{U} = (U_1, \dots, U_L)^\top$.

To numerically approximate the forward model we use the Galerkin approximation to the CEM, see e.g. [17]. We give a short summary below. We represent the electric potential u using piecewise linear basis functions $\{\phi_i\}_{i=1}^N$, spanning a finite dimensional subspace V_N of $H^1(\Omega)$. The conductivity σ is represented using piecewise constant basis elements $\{\chi_i\}_{i=1}^M$, where each χ_i is the indicator function of exactly one simplex in the mesh. To simplify the notation we identify u and σ with the coefficients in their respective basis expansions. That is, $u \approx \sum_{i=1}^N u_i \phi_i \cong (u_1, \dots, u_N)^\top$ and analogously for $\sigma \approx \sum_{j=1}^M \sigma_j \chi_j \cong (\sigma_1, \dots, \sigma_M)^\top$.

Applying the above Galerkin approximation to (7) results in the linear system

$$\begin{pmatrix} A(\sigma) + B & C & \mathbf{0} \\ C^\top & D & \mathbf{1} \\ \mathbf{0}^\top & \mathbf{1}^\top & 0 \end{pmatrix} \begin{pmatrix} u \\ \mathbf{U} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{1} \\ 0 \end{pmatrix}, \quad (8)$$

with block matrices

$$\begin{aligned} A_{ij} &= \int_{\Omega} \sigma \nabla \phi_i \cdot \nabla \phi_j \, dx, & \text{for } i, j = 1, \dots, N \\ B_{ij} &= \sum_{l=1}^L \frac{1}{z_l} \int_{e_l} \phi_i \phi_j \, ds, & \text{for } i, j = 1, \dots, N \\ C_{ij} &= \frac{1}{z_j} \int_{e_j} \phi_i \, ds, & \text{for } i = 1, \dots, N \text{ and } j = 1, \dots, L \\ D_{ii} &= \frac{1}{z_i} \int_{e_i} 1 \, ds, & \text{for } i = 1, \dots, L, \end{aligned}$$

where $\mathbf{U} = (U_1, \dots, U_L)^\top \in \mathbb{R}^L$, and we omit injection pattern indices for brevity.

There are two properties of the linear system (8) that can be used to reduce the computational effort. First, for a fixed conductivity σ , the CEM is linear with respect to the injection patterns. This enables reusing intermediate steps of the procedure for solving Eqn. (8), e.g., the LU factorisation of the system matrix which is used to compute the numerical solution (u, \mathbf{U}) for each current pattern under consideration. Second, only the block matrix $A(\sigma)$ depends on σ , and needs to be recomputed. All other block matrices can be computed offline.

The resulting discrete forward operator is implemented with the finite element software FEniCSx [3], and is available online¹.

2.2. Jacobian. We compute the Jacobian $\mathbf{J}_{\sigma_{\text{ref}}}$ using the discrete function spaces for electric potential and conductivity. Alternative computational strategies using pixel grids or with the adjoint differentiation are demonstrated in [10, 17].

Given K injection patterns and L electrodes, the Jacobian $\mathbf{J}_{\sigma_{\text{ref}}}$ is an $LK \times M$ matrix. However, it is perhaps more intuitive to view the Jacobian as an $L \times K \times M$ tensor. Using [13, Appendix], the Jacobian can be expressed as

$$(\mathbf{J}_{\sigma_{\text{ref}}})_{\cdot, k, j} = \mathbf{W}_{k, j}, \text{ for } k = 1, \dots, K \text{ and } j = 1, \dots, M, \quad (9)$$

where $(w_{k, j}, \mathbf{W}_{k, j}) \in V_N \times \mathbb{R}^L$ is the solution to

$$\int_{\Omega} \sigma_{\text{ref}} \nabla w_{k, j} \cdot \nabla \phi_i \, dx + \sum_{l=1}^L \frac{1}{z_l} \int_{e_l} (w_{k, j} - (\mathbf{W}_{k, j})_l) (\phi_i - V_l) \, ds = - \int_{\Omega} \chi_j \nabla u^k \cdot \nabla \phi_i \, dx, \quad (10)$$

with $\sum_{l=1}^L (\mathbf{W}_{k, j})_l = 0$, where u^k is the potential corresponding to current pattern l^k . Similarly to Eqn. (7), we introduce a Lagrange multiplier to deal with the constraints, leading to the same system matrix as in Eqn. (8) but with a different right hand side,

$$\begin{pmatrix} A(\sigma_{\text{ref}}) + B & C & \mathbf{0} \\ C^\top & D & \mathbf{1} \\ \mathbf{0}^\top & \mathbf{1}^\top & 0 \end{pmatrix} \begin{pmatrix} w_{k, j} \\ \mathbf{W}_{k, j} \\ \lambda_{k, j} \end{pmatrix} = \begin{pmatrix} f_{k, j} \\ \mathbf{0} \\ 0 \end{pmatrix}, \quad (11)$$

¹https://github.com/alexdenker/eit_fenicsx

with

$$(f_{k,j})_i = - \int_{\Omega} \chi_j \nabla u^k \cdot \nabla \phi_i \, dx, \quad \text{for } i = 1, \dots, N. \quad (12)$$

Using the identity in Eqn. (9), $K \cdot M$ problems need to be solved to construct the Jacobian $\mathbf{J}_{\sigma_{\text{ref}}}$. However, since the dimensionality of the right hand side in Eqn. (11), i.e., the range of the forward operator, is at most N it suffices to compute the solutions $(w_r, \mathbf{W}_r, \lambda_r) \in V_N \times \mathbb{R}^L \times \mathbb{R}$ of

$$\begin{pmatrix} A(\sigma_{\text{ref}}) + B & C & \mathbf{0} \\ C^\top & D & \mathbf{1} \\ \mathbf{0}^\top & \mathbf{1}^\top & 0 \end{pmatrix} \begin{pmatrix} w_r \\ \mathbf{W}_r \\ \lambda_r \end{pmatrix} = \begin{pmatrix} \boldsymbol{\delta}_r \\ 0 \\ 0 \end{pmatrix} \quad (13)$$

where $\boldsymbol{\delta}_r = (\delta_{ir})_{i=1}^N \in \mathbb{R}^N$ is the r -th unit vector. Thus, we only need to solve N linear systems², instead of $K \cdot M$. As $f_{k,j}$ can be represented as a linear combination of $\{\boldsymbol{\delta}_r\}_{r=1}^N$, we can recover the Jacobian as

$$(\mathbf{J}_{\sigma_{\text{ref}}})_{\cdot, k, j} = \mathbf{W}_{k, j} = \sum_{r=1}^N (f_{k, j})_r \mathbf{W}_r. \quad (14)$$

Observe that the piecewise constant elements χ_j are non-zero on exactly one element of the mesh. Thus, only a few summands on the right hand side in Eqn. (14) remain, further reducing the computational complexity.

In higher challenge levels boundary electrodes are removed. This results both in fewer electrode measurements $\tilde{L} < L$ and fewer injection patterns $\tilde{K} < K$. As a result, the reduced Jacobian is of shape $\tilde{L}\tilde{K} \times M$ and it can be computed by removing the corresponding rows of the full Jacobian $\mathbf{J}_{\sigma_{\text{ref}}}$.

2.3. Regularisation. We consider Tikhonov-type regularisers $\mathcal{J}(\delta\sigma) = \frac{1}{2} \|\mathbf{L}\delta\sigma\|_2^2$. Note that for the reconstruction in Eqn. (6) we need access to $\mathbf{L}^\top \mathbf{L}$. Thus, we can define $\mathbf{P} := \mathbf{L}^\top \mathbf{L}$ instead of \mathbf{L} . We use three different regularisers:

- **First-order smoothness prior (FSM):** We define the mesh Laplacian $\mathbf{P}_{\text{FSM}} \in \mathbb{R}^{M \times M}$ with

$$(\mathbf{P}_{\text{FSM}})_{i, j} = \begin{cases} \text{deg}(i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } i \text{ is adjacent to } j \\ 0 & \text{else,} \end{cases} \quad (15)$$

where $\text{deg}(i)$ is the number of neighbours of mesh element i . Matrix \mathbf{P}_{FSM} can also be defined as $\mathbf{P}_{\text{FSM}} = \mathbf{L}_{\text{FMS}}^\top \mathbf{L}_{\text{FMS}}$, for \mathbf{L}_{FMS} constructed as in [5].

- **Smoothness prior (SM):** A smoothness distance matrix is constructed via $\mathbf{P}_{\text{SM}} := \boldsymbol{\Sigma}_{\text{SM}}^{-1}$ with $(\boldsymbol{\Sigma}_{\text{SM}})_{i, j} = a \exp(-\|x_i - x_j\|_2^2 / (2b^2))$ where x_i and x_j are the coordinates of mesh elements i and j . We choose $a = 0.025$ and $b = 0.4 \cdot 0.115$. This prior was used in the implementation provided by the organisers of KTC2023 [21].
- **Levenberg–Marquardt regulariser (LM) [12]:** The LM regulariser is used in the NOSER framework [7] and is defined as $\mathbf{P}_{\text{LM}} = \text{diag}(\mathbf{J}_{\sigma_{\text{ref}}}^\top \boldsymbol{\Sigma}^{-1} \mathbf{J}_{\sigma_{\text{ref}}})$. Note that $\boldsymbol{\Sigma}$ is the covariance matrix of the Gaussian noise model in Eqn.(5).

²Moreover, we have $N < M$, i.e., the dimension of piecewise linear elements is lower than the dimension of piecewise constant elements.

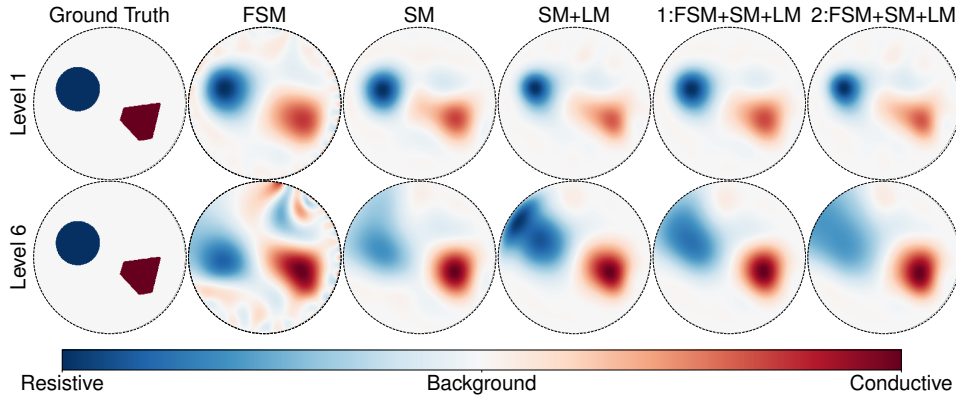


FIGURE 2. Example initial reconstructions on challenge levels 1 and 6. Level 6 was chosen as it best highlights differences in linearised reconstructions. We evaluate an independent FSM prior, independent SM prior, joint SM+LM prior and two joint priors FSM+SM+LM with different regularisation strengths. The chosen image is a sample of the validation data.

In summary, the regularised solution to Eqn. (5) is obtained by combining the three regularisers as

$$\mathbf{F}^\dagger(\delta\mathbf{U}) = (\mathbf{J}_{\sigma_{\text{ref}}}^\top \Sigma^{-1} \mathbf{J}_{\sigma_{\text{ref}}} + \alpha_{\text{FSM}} \mathbf{P}_{\text{FSM}} + \alpha_{\text{SM}} \mathbf{P}_{\text{SM}} + \alpha_{\text{LM}} \mathbf{P}_{\text{LM}})^{-1} \mathbf{J}_{\sigma_{\text{ref}}}^\top \Sigma^{-1} \delta\mathbf{U}, \quad (16)$$

where α_{FSM} , α_{SM} , and α_{LM} are the regularisation strengths, and \mathbf{F}^\dagger is the linearised reconstruction operator. The regularisation strengths are selected using a validation set of the four measurements provided by the organisers. The chosen regularisers promote different structures within the reconstructed images. Moreover, different regularisation choices produce clearly distinct reconstructions with corresponding artefacts, which is especially evident for higher challenge levels. For Post-Processing and Conditional-Diffusion approaches we use 5 different regularisation choices, illustrated in Figure 2. The guiding idea is that combining the information from the various reconstructions will improve the performance of the trained convolutional neural network.

The linearised reconstruction computed from Eqn. (16) resides on the piecewise constant mesh representation, whilst convolutional neural networks require inputs represented as a 256×256 pixel grid. Bilinear interpolation, denoted as $\mathcal{I} : \mathbb{R}^M \rightarrow \mathbb{R}^{256^2}$, was used to interpolate from mesh to image. We denote the resulting set of five interpolated linearised reconstructions as

$$\widehat{\delta\sigma} := \{\mathcal{I}(\mathbf{F}_j^\dagger(\delta\mathbf{U}))\}_{j=1}^5 \quad (17)$$

where the subscript denotes the j -th choice of regularisation strengths. The strength of regularisation varies between challenge level for all five linearised reconstructions. Namely, a weaker regularisation strength is required for the full view setting at level 1, meaning that a total of 35 variations of regularisation strengths are defined. For clarity and brevity we omit an index for the challenge level.

3. Deep learning approaches. We submitted three deep learning approaches to KTC2023. Two learned reconstructors, FC U-Net and Post-Processing, and a

generative approach, **Conditional-Diffusion**. Our approaches share the same U-Net backbone³ [22, 9]. This network includes a conditioning mechanism allowing the level (timestep) to more effectively influence the model’s output.

All models were trained on a simulated data set, which is described in Section 4.1. Let $\sigma \in \mathbb{R}^d$ with $d = 256^2$ denote the representation, discussed in Section 2.3, of the reconstruction on the square pixel grid, where the pixels outside of the circular water tank are always treated as the background class.

3.1. Learned reconstructors. The goal in learned reconstruction is to identify parameters $\hat{\theta}$ of a parametrised reconstruction operator $\mathcal{R}_\theta : \mathbb{R}^{KL} \rightarrow \mathbb{R}^d$, such that

$$\mathcal{R}_{\hat{\theta}}(\delta\mathbf{U}) \approx \delta\sigma. \quad (18)$$

Given a paired data set $\{(\delta\mathbf{U}^{(i)}, \delta\sigma^{(i)})\}_{i=1}^n$ of samples, we compute

$$\hat{\theta} = \operatorname{argmin}_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\delta\sigma^{(i)}, \mathcal{R}_\theta(\delta\mathbf{U}^{(i)})), \quad (19)$$

using a suitable loss function $\mathcal{L} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$. The mean-squared error loss function is commonly employed for reconstruction tasks [20]. However, the goal in the challenge was not to reconstruct the conductivity distribution, but rather to provide a segmentation into water (background), resistive and conductive inclusions. Therefore, we use categorical cross entropy (CCE) as a loss function. CCE is commonly used for image segmentation but has also been used for computed tomography segmentation [1]. Let $\mathcal{R}_{\hat{\theta}}$ denote the learned reconstructor. The model outputs logits, which are transformed to class probabilities by using a softmax function

$$\hat{p}_{i,c} := \operatorname{Softmax}(\mathcal{R}_{\hat{\theta}}(\delta\mathbf{U})_{i,\cdot}) := \frac{\exp(\mathcal{R}_{\hat{\theta}}(\delta\mathbf{U})_{i,c})}{\sum_{c'=1}^C \exp(\mathcal{R}_{\hat{\theta}}(\delta\mathbf{U})_{i,c'})}, \quad (20)$$

for all pixels $i = 1, \dots, d$ and all classes $c = 1, \dots, C$. Let further $p \in \{0, 1\}^{d \times C}$ be the one-hot encoding of the ground truth class. The CCE loss is defined as

$$\mathcal{L}_{\text{CCE}}(\hat{p}, p) = -\frac{1}{d} \sum_{i=1}^d \sum_{c=1}^C p_{i,c} \log(\hat{p}_{i,c}). \quad (21)$$

After training, the final segmentation is obtained by choosing the class with the highest probability as $(\operatorname{argmax}_c \hat{p}_{i,c})_{i=1, \dots, d}$. The network is directly trained for segmentation, thus avoiding the need for an additional segmentation step. For both learned reconstruction methods, FC U-Net and Post-Processing, we provide the challenge level as an additional input to the model and train a single model for all levels. They differ in the parametrisation of the reconstruction operator \mathcal{R}_θ . Where the FC U-Net implements a neural network directly acting on the measurements, the Post-Processing defines a two-step approach [20, 24].

3.1.1. FC U-Net. The design of the FC U-Net closely follows the work of Chen et al. [6]. The model consists of two components: an initial learned transformation that maps the measurements to a pixel grid and a subsequent segmentation, implemented as a convolutional neural network.

Instead of using the linear reconstruction method from Section 2.3, we will learn a linear mapping (represented as a single fully connected linear layer) that is applied to the measurements. However, learning a linear mapping from the measurements,

³ Accessible at <https://github.com/openai/guided-diffusion>

with dimension $KL = 2356$, to the pixel grid, would require more than 150M parameters and is computationally intractable. To reduce the number of parameters, we only learn a mapping to a 64×64 pixel grid and use a bilinear interpolation to the 256×256 pixel grid. The output of this initial transformation is used as an input to the second stage. Let $W \in \mathbb{R}^{64^2 \times KL}$ denote the initial linear layer and $\mathcal{S} : \mathbb{R}^{64^2} \rightarrow \mathbb{R}^{256^2}$ be the bilinear upsampling operator. The FC U-Net is given by

$$\mathcal{R}_\theta(\delta\mathbf{U}, k) := \tilde{\mathcal{R}}_\theta(\mathcal{S}(W\delta\mathbf{U}), k), \text{ for } k = 1, \dots, 7, \quad (22)$$

with k denoting the challenge level and $\tilde{\mathcal{R}}_\theta$ is implemented as the attention U-Net [9]. An overview of this approach is given in Figure 3. The missing measurements in $\delta\mathbf{U}$ for the higher challenge level are filled with zeros.

To learn the linear map W , for the initial reconstruction, and the weights θ for the segmentation, we propose a novel two phase training process. In the first phase only the initial linear layer is trained using a mean-squared-error loss

$$\min_W \sum_{k=1}^7 \sum_{i=1}^{n_k} \|\mathcal{S}(W\delta\mathbf{U}^{(k,i)}) - \delta\boldsymbol{\sigma}^{(k,i)}\|_2^2. \quad (23)$$

The aim of this phase is to provide a good initialisation of W . Afterwards, the full model is trained to provide a segmentation using the CCE loss

$$\min_{\theta, W} \sum_{k=1}^7 \sum_{i=1}^{n_k} \mathcal{L}_{\text{CCE}}(\hat{p}^{(k,i)}, p^{(k,i)}), \quad (24)$$

where $\hat{p}^{(k,i)} = \text{Softmax}(\tilde{\mathcal{R}}_\theta(\mathcal{S}(W\delta\mathbf{U}^{(k,i)}), k))$. In this joint optimisation of θ and W , we use a smaller learning rate for the linear layer W than for θ .

The dataset used for training the FC U-Net consisted only of random phantoms and simulated measurements. When evaluated on four challenge phantoms provided by the organisers, we noticed a deterioration in the final segmentation. To alleviate this generalisation problem, we introduce a finetuning phase, where the FC U-Net is trained for 1000 optimisation steps on these 4 challenge phantoms using a small learning rate of 1×10^{-6} , for both W and θ .

3.1.2. Post-Processing. Learned post-processing was one of the first applications of deep learning to inverse problems [2, 20]. In this approach an initial reconstruction (computed from a classical reconstruction method) is used as an input to a convolutional neural network. More precisely, the reconstruction operator is parametrised as $\mathcal{R}_\theta(\delta\mathbf{U}) = \tilde{\mathcal{R}}_\theta(\mathbf{F}^\dagger(\delta\mathbf{U}))$ where $\mathbf{F}^\dagger(\delta\mathbf{U})$ denotes the initial reconstruction. We adapt this approach in three ways. First, a bilinear interpolation step is used to map the mesh values to an image for the convolutional neural networks. Second, five linearised reconstructions are used as initial reconstructions, cf. Section 2.3. Last, the network is conditioned on the challenge level, as for the FC U-Net, cf. Section 3.1.1. These adaptations result in the formulation

$$\mathcal{R}_\theta(\delta\mathbf{U}, k) = \tilde{\mathcal{R}}_\theta(\widehat{\delta\boldsymbol{\sigma}}, k), \text{ for } k = 1, \dots, 7, \quad (25)$$

where $\widehat{\delta\boldsymbol{\sigma}}$ are the five interpolated linearised reconstructions and k is the challenge level. An overview of the Post-Processing network is given in Figure 4. The resulting network is trained for segmentation using the CCE loss function (24) over all challenge levels and training samples, with the predicted class probability given by

$$\hat{p}^{(k,i)} = \text{Softmax}(\tilde{\mathcal{R}}_\theta(\widehat{\delta\boldsymbol{\sigma}}^{(k,i)}, k)), \text{ for } k = 1, \dots, 7 \text{ and } i = 1, \dots, n_k. \quad (26)$$

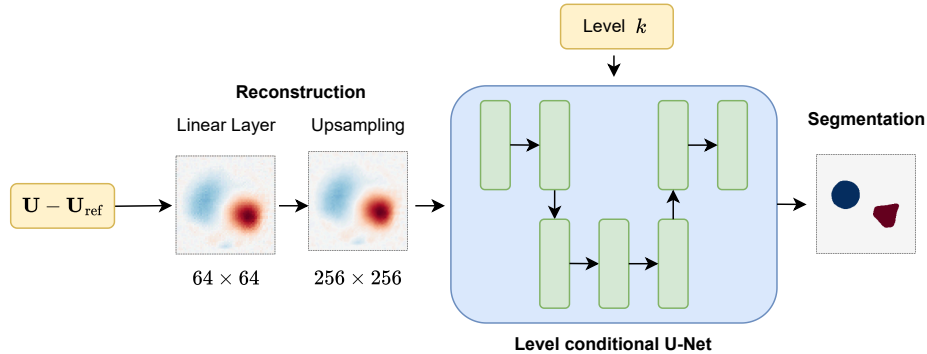


FIGURE 3. FC U-Net network. We first use a linear layer to map the measurements to a 64×64 pixel grid, this is then bilinearly interpolated to the 256×256 grid. The network is trained to output class probabilities using categorical cross-entropy loss. The class probabilities are converted to segmentation maps by assigning the class with highest probability.

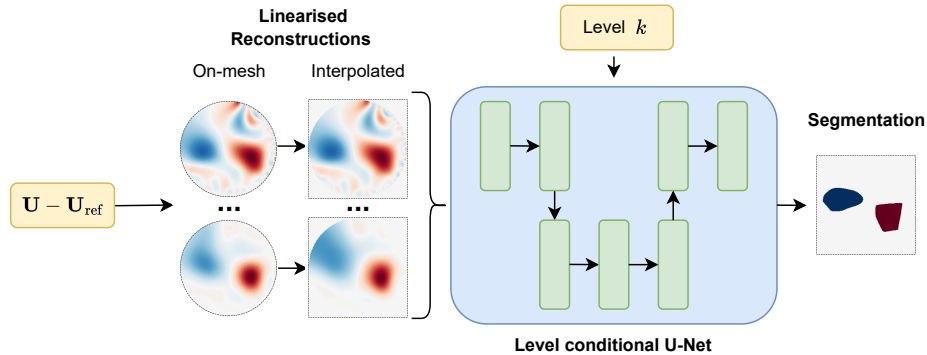


FIGURE 4. Post-Processing network. The five linearised reconstructions are interpolated to the pixel grid as described in Section 2.3. The network is trained to output class probabilities using categorical cross-entropy loss. The class probabilities are converted to segmentation maps by assigning the class with highest probability.

3.2. Conditional density estimation. From a statistical perspective of inverse problems, we are interested in recovering the posterior distribution $p^{\text{post}}(\boldsymbol{\sigma}|\mathbf{U})$, i.e., the conditional distribution of conductivity $\boldsymbol{\sigma}$ given the boundary measurements \mathbf{U} [28]. The goal in conditional density estimation is to approximate the true posterior $p(\boldsymbol{\sigma}|\mathbf{U})$ with a conditional probabilistic model $p_{\theta}(\boldsymbol{\sigma}|\mathbf{U})$ given a data set $\{(\boldsymbol{\sigma}^{(i)}, \mathbf{U}^{(i)})\}$, for $i = 1, \dots, n$ with $(\boldsymbol{\sigma}^{(i)}, \mathbf{U}^{(i)}) \sim p(\boldsymbol{\sigma}, \mathbf{U})$. In this work $p_{\theta}(\boldsymbol{\sigma}|\mathbf{U})$ is modelled using denoising diffusion probabilistic models (DDPM) [15, 25], which have shown promising results on many image generation tasks [9].

3.2.1. *Conditional diffusion models.* Conditional variants of diffusion models were proposed for various inverse problems, including super-resolution [23], time series imputation [29] and image inpainting [4]. Specifically, we build on ideas from [4].

We make use of the discrete time formulation of diffusion models [15]. DDPMs define a forward diffusion process, given by a Markov chain, which gradually adds noise to the data over $T = 1000$ timesteps as

$$\boldsymbol{\sigma}_t = \sqrt{1 - \beta_t} \boldsymbol{\sigma}_{t-1} + \sqrt{\beta_t} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (27)$$

with variances $\beta_1 \leq \dots \leq \beta_T$. The variances are chosen so that the terminal distribution approaches a standard Gaussian, $\boldsymbol{\sigma}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Given the noiseless sample $\boldsymbol{\sigma}_0$, the noisy sample at time t can be directly obtained as

$$\boldsymbol{\sigma}_t = \sqrt{\bar{\alpha}_t} \boldsymbol{\sigma}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \quad (28)$$

with $\bar{\alpha}_t = \prod_{i=1}^t (1 - \beta_i)$. The goal of DDPMs is to reverse this diffusion process by learning parametrised transition densities $p_\theta(\boldsymbol{\sigma}_{t-1} | \boldsymbol{\sigma}_t)$. Training a DDPM amounts to minimising the so-called $\boldsymbol{\epsilon}$ -matching loss [15]. This framework can be extended to conditional density estimation by including the measurements in the parametrised transition densities, i.e., $p_\theta(\boldsymbol{\sigma}_{t-1} | \boldsymbol{\sigma}_t, \delta \mathbf{U})$, using a conditional neural network $\boldsymbol{\epsilon}_\theta(\boldsymbol{\sigma}_t, \delta \mathbf{U}; t)$ and minimise a conditional $\boldsymbol{\epsilon}$ -matching loss

$$\min_{\theta} \mathbb{E}_{t \sim U(\{1, \dots, T\})} \mathbb{E}_{(\boldsymbol{\sigma}_0, \delta \mathbf{U}) \sim p(\boldsymbol{\sigma}_0, \delta \mathbf{U})} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})} [\|\boldsymbol{\epsilon}_\theta(\boldsymbol{\sigma}_t, \delta \mathbf{U}; t) - \boldsymbol{\epsilon}\|_2^2], \quad (29)$$

with $\boldsymbol{\sigma}_t$ as given in Eqn. (28) and the expectation over $(\boldsymbol{\sigma}_0, \delta \mathbf{U})$ is estimated using the simulated dataset and denoted by \mathbb{E} . An overview of the network is given in Figure 5, where the input to the network is a concatenation of the linearised reconstructions, interpolated to the pixel grid, and the noisy image $\boldsymbol{\sigma}_t$, together with the time step t .

In [15], the authors make use of ancestral sampling to sample from the learned distribution. However, this requires simulating the reverse process for all T timesteps, resulting in a computationally expensive sampling method. To increase the sampling speed, we make use of the accelerated sampling scheme proposed in the DDIM framework [27]. Let τ be a subsequence of $\{1, \dots, T\}$ of length S with $\tau_1 = 1$ and $\tau_S = T$. The DDIM sampling, starting with $\boldsymbol{\sigma}_{\tau_S} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, is given by

$$\boldsymbol{\sigma}_{\tau_{s-1}} = \sqrt{\bar{\alpha}_{\tau_{s-1}}} \hat{\boldsymbol{\sigma}}_0(\boldsymbol{\sigma}_{\tau_s}, \delta \mathbf{U}) + \sqrt{1 - \bar{\alpha}_{\tau_{s-1}} - \gamma_{\tau_s}^2} \boldsymbol{\epsilon}_\theta(\boldsymbol{\sigma}_{\tau_s}, \delta \mathbf{U}, \tau_s) + \gamma_{\tau_s} \boldsymbol{\epsilon}, \quad (30)$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and where $\hat{\boldsymbol{\sigma}}_0(\boldsymbol{\sigma}_{\tau_s}, \delta \mathbf{U})$ is the Tweedie estimate [11], defined by

$$\mathbb{E}[\boldsymbol{\sigma}_0 | \boldsymbol{\sigma}_t, \delta \mathbf{U}] \approx \hat{\boldsymbol{\sigma}}_0(\boldsymbol{\sigma}_t, \delta \mathbf{U}) = \frac{1}{\sqrt{\bar{\alpha}_t}} (\boldsymbol{\sigma}_t - \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_\theta(\boldsymbol{\sigma}_t, \delta \mathbf{U}; t)). \quad (31)$$

The stochasticity parameter γ_t in Eqn. (30) is chosen as

$$\gamma_{\tau_s} = \eta \sqrt{(1 - \bar{\alpha}_{\tau_{s-1}}) / (1 - \bar{\alpha}_{\tau_s})} \sqrt{1 - \bar{\alpha}_{\tau_s} / \bar{\alpha}_{\tau_{s-1}}}, \quad (32)$$

with a tunable hyperparameter η , see [27].

In our implementation, we do not directly feed $\delta \mathbf{U}$ into the epsilon model $\boldsymbol{\epsilon}_\theta$, but rather make use of the initial reconstructions introduced in Section 2.3. Thus, our model is of the form $\boldsymbol{\epsilon}_\theta(\boldsymbol{\sigma}_t, \widehat{\boldsymbol{\delta \sigma}}, t)$ where $\widehat{\boldsymbol{\delta \sigma}}$ denotes the set of five interpolated linearised reconstructions in Eqn. (16). In doing so we do not approximate the true posterior $p^{\text{post}}(\boldsymbol{\sigma} | \delta \mathbf{U})$, but rather a conditional distribution $p(\boldsymbol{\sigma} | \widehat{\boldsymbol{\delta \sigma}})$.

As the goal was to produce a segmentation and not a reconstruction, we do not represent $\boldsymbol{\sigma}$ using conductivity values, but rather as an image with values in $[0, 2]$.

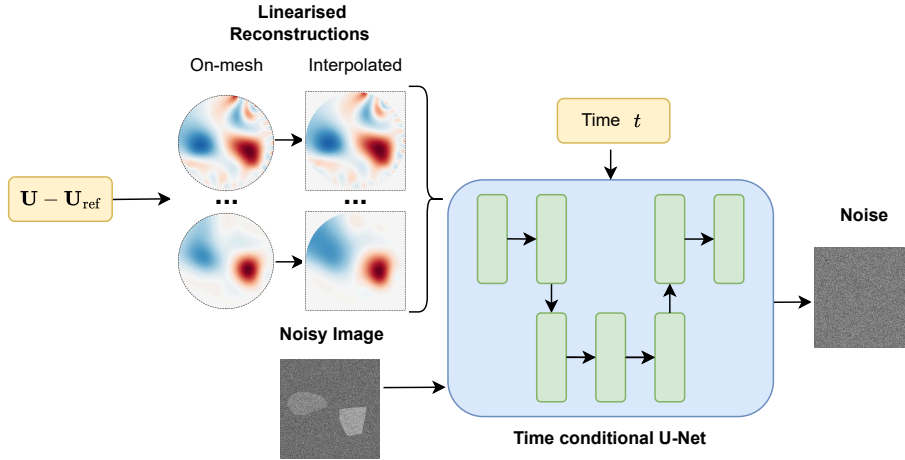


FIGURE 5. Conditional-Diffusion network. The five linearised reconstructions are interpolated to the pixel grid as described in Section 2.3. The noisy image and linearised reconstructions are input into the network. Using the ϵ -matching loss function, the network is trained to estimate the noise. Through sampling the network a segmentation map is obtained.

The segmentation is then obtained by rounding the reconstruction σ to the nearest $\{0, 1, 2\}$ integer, where 0 represents the background class, 1 the resistive and 2 the conductive inclusion class. For the final segmentation, we draw J samples $\sigma^{(j)}$ using DDIM Eqn. (30) and perform a pixel-wise majority voting

$$\hat{\sigma}_i = \operatorname{argmax}_{c=0,1,2} \#\{\sigma_i^{(j)} \mid \sigma_i^{(j)} = c, j = 1, \dots, J\}, \quad (33)$$

for all pixels $i = 1, \dots, d$, with $\#$ denoting the set cardinality.

4. Practical considerations. In this Section we report the practical considerations of our submission. First, we cover the generation of the training data, second, we give details about the computation of the linearised reconstruction and third, we discuss aspects of the neural network architecture.

4.1. Dataset. An important aspect of our submission is the creation of a simulated dataset suitable for training the different deep learning approaches. We start by generating random segmentation maps consisting of non-overlapping polygons, circles, rectangles and handdrawn objects on the 256×256 pixel grid. Example phantoms are presented in Figure 6, where we only visualise the circular water tank. Each object was assigned to be either resistive or conductive. The areas outside of an object, and outside the water tank, were assigned the background class. Given this segmentation map, we simulate conductivity values for the objects. The conductivity of resistive objects was randomly chosen in $[0.025 \text{ (Ohm} \cdot \text{m)}^{-1}, 0.125 \text{ (Ohm} \cdot \text{m)}^{-1}]$ and the conductivity of conductive objects in $[5.0 \text{ (Ohm} \cdot \text{m)}^{-1}, 6.0 \text{ (Ohm} \cdot \text{m)}^{-1}]$. The background was assigned a constant conductivity value of $0.745 \text{ (Ohm} \cdot \text{m)}^{-1}$, which was computed using the reference measurements of the empty water tank via least squares fitting [30]. In the next

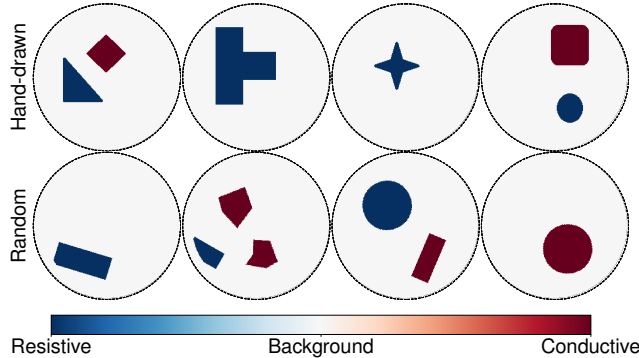


FIGURE 6. Top: Hand-drawn training phantoms. Bottom: Randomly generated training phantoms. For the visualisation, we only show the circular water tank. However, note that all models are trained using the square 256×256 pixel images.

step, the resulting phantoms were interpolated from the pixel grid to the piecewise constant finite element representation. The measurements were simulated using the forward operator specified in Section 2.1. Gaussian noise was added with zero mean and covariance $\Sigma = \text{diag}(0.05 \mathbf{U}_{\text{ref}} + 0.01 \max(\mathbf{U}_{\text{ref}}))$ according to the reference measurements of the empty water tank. For the simulation of the measurements a fixed contact impedance $z = 1 \times 10^{-6} \text{Ohm}$ was chosen for all 32 electrodes⁴. The number of training samples used per level is provided in Table 1. In total, we simulated more than 100K data pairs. The lower number of training samples for level 6 was due to technical problems in the simulation.

TABLE 1. Number of training samples used per level.

Level	1	2	3	4	5	6	7
Training samples	16527	16619	16591	16587	16604	12102	16298

4.2. Initial reconstruction. Both the Post-Processing approach in Section 3.1.2 and the Conditional-Diffusion in Section 3.2.1 require an initial reconstruction as the input. We experimented with different classical reconstruction methods. Iterative reconstruction methods, e.g., $L1$ -regularisation [13] or Gauß-Newton methods [5], resulted in higher quality reconstructions compared to the linearised approach in Section 2.3. However, as this initial reconstruction has to be computed for every example in the training set, i.e., for more than 100K examples in the dataset we used, the computational expense was a constraint. Thus, we decided against the computationally more expensive iterative methods and used the faster linearised reconstruction. However, even for the linearised reconstruction, simulation of the measurements and computation of the initial reconstruction took about a week.

The organisers provided a finite element implementation of the CEM. We decided to use our own implementation to more easily change the discrete function spaces and use a different mesh. A comparison of our mesh and the provided mesh is shown

⁴We also experimented with identifying the contact impedance using least squares fitting, but did not obtain good results.

in Figure 7. The provided mesh shows some small irregularities at the centre and top of the domain, which led to some differences in the forward solution and initial reconstructions. Instead, we use a uniform mesh with a mesh size of 0.005, which was created with the software Gmsh [14]. Further, we set the boundary elements to cover the electrodes.

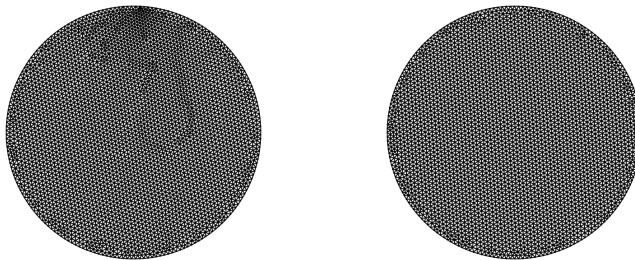


FIGURE 7. Left: The mesh provided by the organizers. Right: Our custom mesh for the forward operator.

4.3. Neural network architecture. We use a minimally adapted guided diffusion model proposed by [9]. The architecture consists of a U-Net [22] with attention blocks and time embedding. The time (scalar)DaTo embedding is adapted for Post-Processing and FC U-Net to allow the network to incorporate level information, meaning the training data across all levels can be used during training. The time or level information is introduced to the network by adaptive group normalisation layers [9]. Each group normalisation layer [33] in the U-Net is replaced with

$$\text{AdaGroupNorm}(h, z) = z_s \text{GroupNorm}(h) + z_t, \quad (34)$$

where h is the intermediate feature and $z = (z_s, z_t)$ is the output of a neural network taking the time or level information as an input. With our choice of hyperparameters, e.g., number of layers, channels, etc., the total number of trainable parameters is 31M.

The number of input and output channels of the U-Net vary between the approaches. Post-Processing and Conditional-Diffusion have five input channels corresponding to the five interpolated linearised reconstructions $\widehat{\delta\sigma}$, whereas FC U-Net has a single channel input for the interpolated learned reconstruction $\mathcal{S}(W\delta\mathbf{U})$. For the learned reconstructors a CCE loss is used that required the three class probabilities, thus three output channels are used. For Conditional-Diffusion the ϵ -matching loss is used, requiring a single channel output. Due to differences in input and output channels the backbone U-Nets do not have an equal number of parameters, albeit the differences are negligible. For FC U-Net the linear layer \mathcal{W} required 10M parameters; this is a significant increase in learnable parameters as compared to the other approaches.

5. Results and discussion. In this section we present the final challenge results for our three approaches. Quantitative scores are computed using structural similarity index measure (SSIM) [32] individually on binary maps of conductive and resistive inclusions. This was averaged to give a per-sample score. This per-sample score was summed across all samples of a level to give a level score, and then summed

over all levels to give the overall score of the methods⁵. For each challenge level, three different phantoms (A,B,C) were evaluated. Visual results are presented in Figure 8, Figure 9 and Figure 10. In most reconstructions, the number of objects, positions and rough shape are correctly identified. Exceptions are cases where a small conductive object was placed in the middle of the water tank and surrounded by other objects, see for example level 7 in Figure 8 or level 7 in Figure 9. Further, the reconstruction of objects on the upper left side of the water tank is often worse as the measurements of this part of the boundary are removed for higher challenge levels. See for example level 7 in Figure 9, where the shape of the rectangle at the top of the water tank can not be recovered.

Quantitative results are presented in Table 2. Besides our submission, we also present the results of the second and third best performing team. With a final score of 15.24 the Post-Processing approach was the best performing method in KTC2023. However, the FC U-Net was able to outperform this approach at levels 2, 4, 5 and 6. The second place with a score of 12.75 was achieved by a team from the federal University of ABC and the third place was achieved by a team from DTU with a score of 12.45. On level 4 the second place team achieved a higher score than the Post-Processing. However, both our FC U-Net approach, with a score of 15.13, and our Conditional-Diffusion, with a score of 14.60, would have won the challenge.

Post-Processing and the FC U-Net perform similarly, while worse performance can be observed with Conditional-Diffusion. For the Conditional-Diffusion approach, a separate neural network was trained for each challenge level. Thus, the network for each level was trained using only a subset of all available phantoms and measurements that were simulated. On the other hand, Post-Processing and FC U-Net utilise the training examples across all the levels. The learned reconstruction approaches utilise CCE loss specific to segmentation tasks, whereas Conditional-Diffusion uses a ϵ -matching which is required for DDPM. Rather than using a single sample, for Conditional-Diffusion we draw J conditional samples, and the segmentation was determined via majority voting. This could be extended to obtain a notion of uncertainty.

The Post-Processing and Conditional-Diffusion approaches both took a set of five linearised reconstructions as input. Through using a set of reconstructions with different regularisation strengths we attempt to obtain a more robust segmentation as the best regularisation strength is not known. In a similar sense, a set of reconstructions could be learned with the FC U-Net but would require significant increase in the number of learnable parameters.

6. Conclusion. The KTC2023 challenge provided an opportunity to evaluate state-of-the-art methods on the problem of reconstructing segmentation maps from EIT measurements. Our winning submissions utilised deep learning, with two learned reconstructor methods (FC U-Net and Post-Processing), as well as a Conditional-Diffusion generative method. The choice of network architecture and dataset are vitally important for deep learning approaches; requiring knowledge of the problem whilst being practical. In this work all submissions utilised the same training dataset and backbone network architecture; allowing for comparison between methods. Both FC U-Net and Post-Processing provided similar results, whereas Conditional-Diffusion performed less well. The learned reconstructors were trained

⁵Three phantoms were evaluated per level resulting in a maximum score of 21.

TABLE 2. Quantitative comparison of our three submissions via structural similarity index measure (SSIM). These are official challenge results, rounded to the nearest hundredth. The second place was achieved by Team ABC from the Federal University of ABC, Brasil. The third place was achieved by Team DTU from Technical University of Denmark. SSIM is averaged for a given sample between conductive and resistive inclusions. At each level the SSIM is summed across the three samples, and the overall sum for a method is summed across all samples and levels.

Level	1	2	3	4	5	6	7	Sum
FC U-Net	2.72	2.64	2.31	1.80	2.06	2.07	1.53	15.13
Post-Processing	2.76	2.56	2.54	1.71	2.06	1.92	1.69	15.24
Conditional-Diffusion	2.67	2.49	2.47	1.61	1.94	1.76	1.65	14.60
Team ABC	2.75	2.37	2.07	1.74	1.08	1.53	1.22	12.75
Team DTU	2.28	2.3	1.87	1.55	1.34	1.44	1.60	12.45

across all levels (utilising level-conditioning), whereas the individual Conditional-Diffusion networks were trained individually for each level, effectively reducing the training dataset size. The FC U-Net required an additional fine-tuning phase on the provided real measurements and phantoms. This was not needed for the Post-Processing network which only used simulated measurements and phantoms. The Post-Processing and Conditional-Diffusion methods took a set of five Tikhonov-regularised initial reconstructions as input, while the FC U-Net method used a single linear layer to map from measurements to images. Out of the three methods submitted the Post-Processing method gave the best performance. This suggests that a post-processing approach trained on a high-quality simulated data set can generalise to real data more easily than a fully learned method. Further work is necessary to fully evaluate the generalisation capabilities of different data-driven approaches.

Acknowledgments. Alexander Denker acknowledges the support by the Deutsche Forschungsgemeinschaft (DFG) within the framework of GRK 2224/1 “ π^3 : Parameter Identification – Analysis, Algorithms, Applications”. Tom Freudenberg acknowledges funding by the DFG – Project number 281474342/GRK2224/2. Imraj Singh was supported by the EPSRC-funded UCL Centre for Doctoral Training in Intelligent, Integrated Imaging in Healthcare (i4health) (EP/S021930/1) and the Department of Health’s NIHR-funded Biomedical Research Centre at University College London Hospitals. Željko Kereta is supported by the UK EPSRC grant EP/X010740/1. Simon Arridge is supported by the UK EPSRC EP/V026259/1. Peter Maass is supported by BMBF-project “#MOIN - MuKIDERM”. T. Kluth acknowledges support from the KIWi project funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF, project number 05M22LBA).

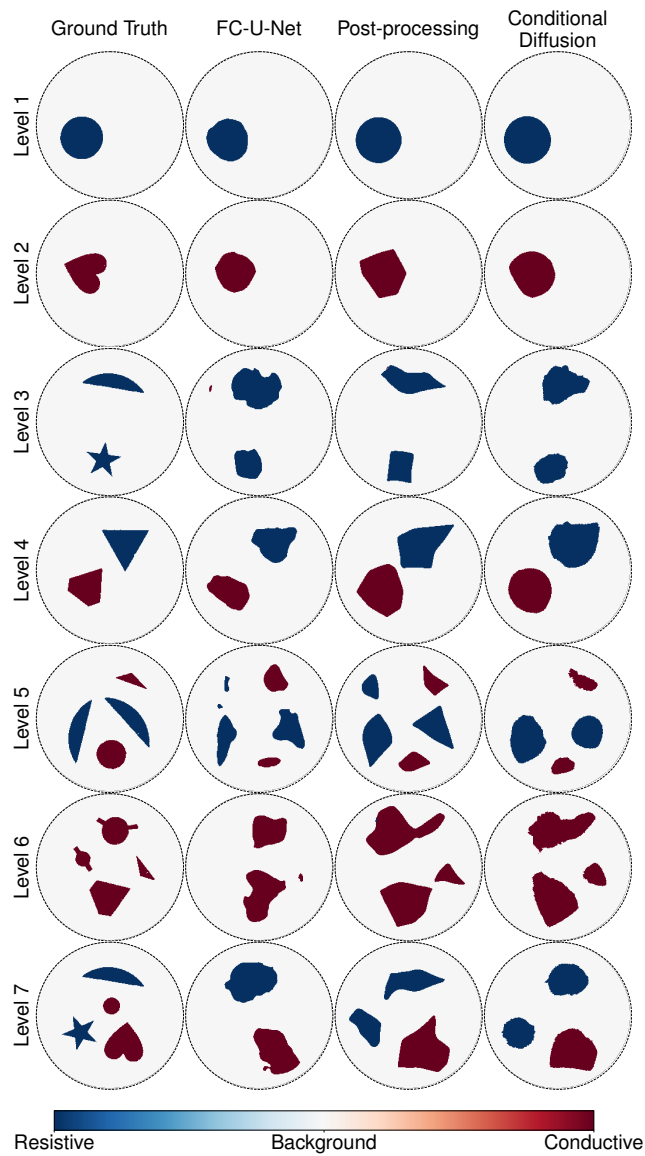


FIGURE 8. Segmentation of the three methods for sample *A* of level 1 to 7.

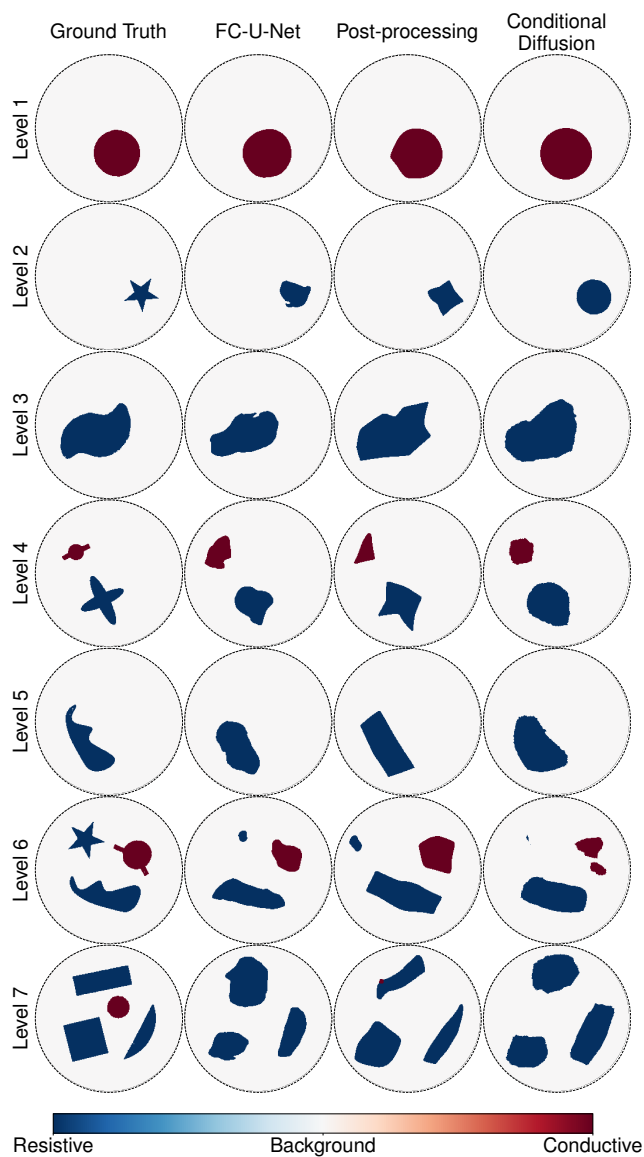


FIGURE 9. Segmentation of the three methods for sample *B* of level 1 to 7.

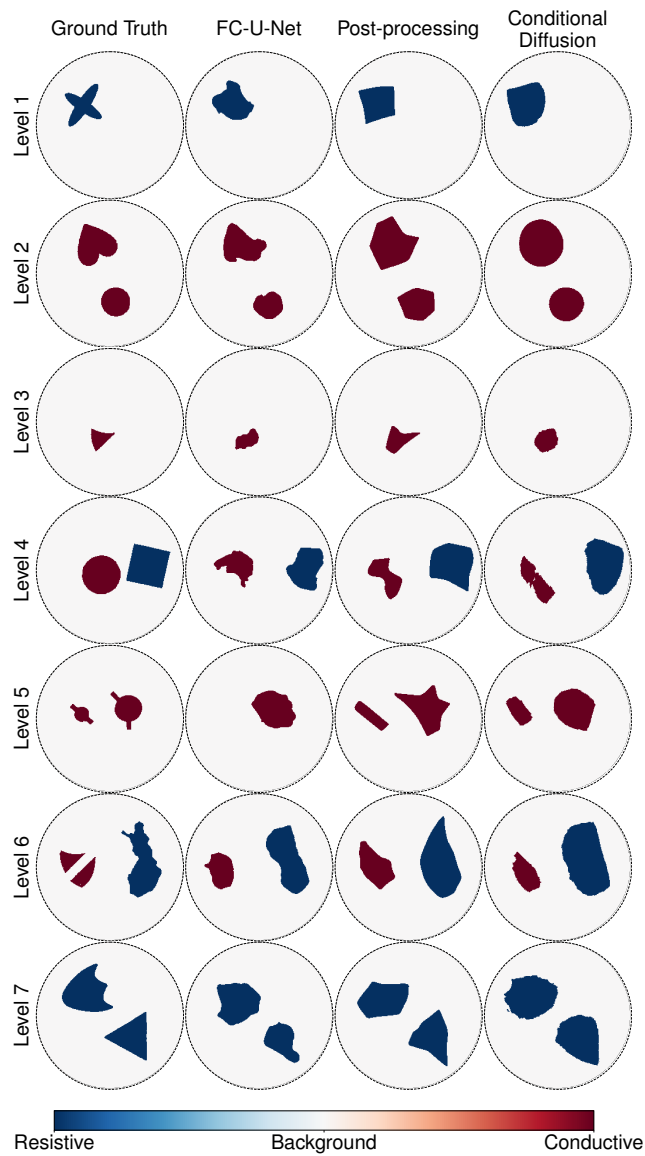


FIGURE 10. Segmentation of the three methods for sample C of level 1 to 7.

REFERENCES

- [1] C. Arndt, A. Denker, S. Dittmer, J. Leuschner, J. Nickel and M. Schmidt, Model-based deep learning approaches to the Helsinki Tomography Challenge 2022, *Applied Mathematics for Modern Challenges*, **1** (2023), 87-104.
- [2] S. Arridge, P. Maass, O. Öktem and C. Schönlieb, [Solving inverse problems using data-driven models](#), *Acta Numerica*, **28** (2019), 1-174.
- [3] I. Baratta, J. Dean, J. Dokken, M. Habera, J. Hale, C. Richardson, M. Rognes, M. Scroggs, N. Sime and G. Wells, DOLFINx, the next generation FEniCS problem solving environment, preprint, 2023.
- [4] G. Batzolis, J. Stanczuk, C. B. Schönlieb and C. Etmann, Conditional image generation with score-Based diffusion models, preprint, [arXiv:2111.13606](#), 2021.
- [5] A. Borsic, B. M. Graham, A. Adler and W. Lionheart, In vivo impedance imaging with total variation regularization, *IEEE Transactions on Medical Imaging*, **29** (2009), 44-54.
- [6] Z. Chen, Y. Yang, J. Jia AND P. Bagnaninchi, Deep learning based cell imaging with electrical impedance tomography, *2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, (2020), 1-6.
- [7] M. Cheney, D. Isaacson, J. C. Newell, S. Simske and J. Goble, NOSER, An algorithm for solving the inverse conductivity problem, *International Journal of Imaging Systems and Technology*, **2** (1990), 66-75.
- [8] H. Chung, B. Sim and J. Ye, Come-closer-diffuse-faster, Accelerating conditional diffusion models for inverse problems through stochastic contraction, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2022), 12413-12422.
- [9] P. Dhariwal and A. Nichol, Diffusion models beat gans on image synthesis, *Advances in Neural Information Processing Systems*, **34** (2021), 8780-8794.
- [10] D. C. Dobson and F. Santosa, An image-enhancement technique for electrical impedance tomography, *Inverse Problems* **10** (1994), 317-334.
- [11] B. Efron, [Tweedie's formula and selection bias](#), *Journal of the American Statistical Association*, **106** (2011), 1602-1614.
- [12] R. Fletcher, A modified Marquardt subroutine for non-linear least squares, *Theoretical Physics Division, Atomic Energy Research Establishment Harwell*, 1971.
- [13] M. Gehre, T. Kluth, A. Lipponen, B. Jin, A. Seppänen, J. Kaipio and P. Maass, [Sparsity reconstruction in electrical impedance tomography: An experimental evaluation](#), *Journal of Computational and Applied Mathematics*, **236** (2012), 2126-2136.
- [14] C. Geuzaine and J. Remacle, [Gmsh, A 3-D finite element mesh generator with built-in pre- and post-processing facilities](#), *Int. J. Numer. Methods. Eng.*, **79** (2009), 1309-1331.
- [15] J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, *Advances in Neural Information Processing Systems*, **33** (2020), 6840-6851.
- [16] N. Hyvönen, Complete electrode model of electrical impedance tomography, Approximation properties and characterization of inclusions, *SIAM Journal on Applied Mathematics*, **64** (2004), 902-931.
- [17] J. Kaipio, V. Kolehmainen, E. Somersalo and M. Vauhkohonen, [Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography](#), *Inverse Problems*, **16** (2000), 1487-1522.
- [18] K. Karhunen, A. Seppänen, A. Lehtikoinen, P. Monteiro and J. Kaipio, Electrical resistance tomography imaging of concrete, *Cement and Concrete Research*, **40** (2010), 137-145.
- [19] S. Martin and C. T. Choi, A post-processing method for three-dimensional electrical impedance tomography, *Scientific Reports*, **7** (2017), 7212.
- [20] G. Ongie, A. Jalal, C. Metzler, R. Baraniuk, A. Dimakis and R. Willett, Deep learning techniques for inverse problems in imaging, *IEEE Journal on Selected Areas in Information Theory*, **1** (2020), 39-56.
- [21] M. Räsänen, P. Kuusela, J. Jauhainen, M. Arif, K. Scheel, T. Savolainen and A. Seppänen, Kuopio Tomography Challenge 2023 (KTC2023), 2023.
- [22] O. Ronneberger, P. Fischer and T. Brox, U-net, Convolutional networks for biomedical image segmentation, *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2015*, (2015), 234-241.
- [23] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet and M. Norouzi, Image super-resolution via iterative refinement, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45** (2022), 4713-4726.

- [24] J. Schwab, S. Antholzer and M. Haltmeier, [Deep null space learning for inverse problems, convergence analysis and rates](#), *Inverse Problems*, **35** (2019), 025008.
- [25] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, *International Conference on Machine Learning (PMLR)*, (2015), 2256-2265.
- [26] E. Somersalo, M. Cheney and D. Isaacson, [Existence and uniqueness for electrode models for electric current computed tomography](#), *SIAM Journal on Applied Mathematics*, **52** (1992), 1023-1040.
- [27] J. Song, C. Meng and S. Ermon, Denoising diffusion implicit models, *International Conference on Learning Representations*, 2020.
- [28] A. Stuart, [Inverse problems: A bayesian perspective](#), *Acta Numerica*, **19** (2010), 451-559.
- [29] Y. Tashiro, J. Song, Y. Song and S. Ermon, CSDI, conditional score-based diffusion models for probabilistic time series imputation, *Advances in Neural Information Processing Systems* **34** (2021), 24804-24816.
- [30] T. Vilhunen, J. P. Kaipio, P. J. Vauhkonen, T. Savolainen and M. Vauhkonen, Simultaneous reconstruction of electrode contact impedances and internal electrical properties, I. Theory, *Measurement Science and Technology*, **13** (2002), 1848.
- [31] H. Wang, G. Xu and Q. Zhou, A comparative study of variational autoencoders, normalizing flows, and score-based diffusion models for electrical impedance tomography, *Journal of Inverse and Ill-posed Problems*, **0** (2024).
- [32] Z. Wang, A. Bovik, H. Sheikh and E. Simoncelli, Image quality assessment, from error visibility to structural similarity, *IEEE Transactions on Image Processing*, **13** (2004), 600-612.
- [33] Y. Wu and K. He, Group normalization, *Proceedings of the European conference on Computer Vision (ECCV)*, 2018.

Received March 2024; revised May 2024; early access May 2024.