

ALMA MATER STUDIORUM—UNIVERSITÀ DI BOLOGNA

**DOTTORATO DI RICERCA IN
MONITORAGGIO E GESTIONE DELLE STRUTTURE E
DELL'AMBIENTE - SEHM2
Ciclo XXXV**

Settore Concorsuale: 01/A5

Settore Scientifico Disciplinare: MAT/08

**DATA DRIVEN REGULARIZATION MODELS
OF NON-LINEAR ILL-POSED
INVERSE PROBLEMS IN IMAGING**

Presentata da: Francesco Colibazzi

Coordinatore Dottorato:

Prof. Luca De Marchi

Supervisore:

Prof.ssa Serena Morigi

Co-supervisore:

Prof.ssa Damiana Lazzaro

Esame finale anno 2023

ABSTRACT

Imaging technologies are widely used in application fields such as natural sciences, engineering, medicine, and life sciences. A broad class of imaging problems reduces to solve ill-posed inverse problems (IPs). Traditional strategies to solve these ill-posed IPs rely on variational regularization methods, which are based on minimization of suitable energies, and make use of knowledge about the image formation model (forward operator) and prior knowledge on the solution, but lack in incorporating knowledge directly from data. On the other hand, the more recent learned approaches can easily learn the intricate statistics of images depending on a large set of data, but do not have a systematic method for incorporating prior knowledge about the image formation model. The main purpose of this thesis is to discuss data-driven image reconstruction methods which combine the benefits of these two different reconstruction strategies for the solution of highly nonlinear ill-posed inverse problems. Mathematical formulation and numerical approaches for image IPs, including linear as well as strongly nonlinear problems are described. More specifically we address the Electrical impedance Tomography (EIT) reconstruction problem by unrolling the regularized Gauss-Newton method and integrating the regularization learned by a data-adaptive neural network. Furthermore we investigate the solution of non-linear ill-posed IPs introducing a deep-PnP framework that integrates the graph convolutional denoiser into the proximal Gauss-Newton method with a practical application to the EIT, a recently introduced promising imaging technique. Efficient algorithms are then applied to the solution of the limited electrodes problem in EIT, combining compressive sensing techniques and deep learning strategies. Finally, a transformer-based neural network architecture is adapted to restore the noisy solution of the Computed Tomography problem recovered using the filtered back-projection method.

Keywords: nonlinear inverse problems, variational models, learning models, MLP neural network, graph neural network, transformer neural network, learning regularizer, image reconstruction, anisotropic total variation, compressive sensing, Electrical Impedance Tomography, Computed Tomography

Contents

Introduction	1
1 Ill-posed inverse problems and numerical optimization strategies	5
1.1 Ill-posed inverse problems	5
1.1.1 Analysis of non-linear ill-posedness	6
1.1.2 Variational Regularization method	7
1.1.3 The discrepancy principle	7
1.2 Variational regularization models for linear problems	8
1.2.1 Truncated Singular Value Decomposition (TSVD) Regularization	9
1.2.2 Tikhonov Regularization	10
1.2.3 Total variation regularization	11
1.2.4 ADMM method for TV-L2	12
1.3 Variational regularization models for non-linear problems	13
1.3.1 Tikhonov regularization for non-linear inverse problems	13
1.3.2 Morozov's discrepancy for Tikhonov regularization	14
1.4 Numerical Optimization Algorithms	16
1.4.1 First-order Methods	16
1.4.2 Second-Order Methods	17
2 Data driven techniques	21
2.1 Bayesian formulation	22
2.2 Data-driven methods	22
2.2.1 End-to-end Models	23
2.3 Hybrid methods	25
2.3.1 Learned Iterative Scheme	25
2.3.2 Learning the Regularizer	27
3 Electrical Impedance Tomography problem	29
3.1 EIT forward problem	29
3.1.1 Forward Operator	30
3.2 EIT inverse problem	31
3.3 EIT stimulation pattern	31

4	Learning the regularizer for EIT inverse problem	33
4.1	Regularized Gauss-Newton for EITNL	34
4.1.1	Learning the regularization matrix	35
4.2	Unrolled Iterative Regularized Gauss-Newton	35
4.2.1	Loss function design	36
4.2.2	Regularizer network	37
4.2.3	2D-Mesh anisotropic TV attention-like mechanism	38
4.3	Simulations and Numerical Experiments	40
4.3.1	Training and Testing Data Sets	41
4.3.2	Figures of merit	41
4.3.3	Effect of the number of blocks and benefit of the ATV mechanism	43
4.3.4	Example 1 - Performance comparisons	44
4.3.5	Example 2 - robustness to the noise	46
4.3.6	Example 3 - Regularizer efficacy	48
5	PnP strategy for EIT inverse problem	51
5.1	Regularizing non-linear ill-posed inverse problems	52
5.2	Characterisation of proximal Newton-type methods	53
5.3	Scaled Plug-and-Play algorithm	55
5.4	The learned GU-Net-denoiser	55
5.4.1	On the nonexpansiveness of the GU-Net-denoiser	57
5.4.2	Deep-PnP proximal Gauss-Newton method	58
5.5	Application to the Inverse EIT Problem	58
5.6	Applying PnP-PGN to the inverse non-linear EIT problem	59
5.6.1	Training Data for the GU-Net-denoiser	59
5.6.2	Performance comparisons	60
5.6.3	Robustness to noise	62
5.6.4	Performance with experimental data	64
6	Limited electrodes EIT reconstruction	69
6.1	Preliminaries on EIT reconstructions	70
6.2	EIT-Limited Electrodes Problem Setup	71
6.3	A Compressed Sensing approach to EIT-LE problem	71
6.4	A Learned Approach to EIT-LE problem	73
6.4.1	VNet architecture	73
6.5	Numerical Experiments	76
7	Post-Processing Transformer Network for CT inverse problem	81
7.1	Background on Transformers	82
7.1.1	Self-Attention	82
7.1.2	Masked Self-Attention	82
7.1.3	Multi-Head Attention	83

CONTENTS

7.2	The Computed Tomography problem	83
7.2.1	Motivation	83
7.3	Implementation details and Results	84
7.3.1	Vformer Architecture	84
7.3.2	Training	85
7.3.3	Numerical Results	86
	Conclusions	89

Introduction

Inverse problems (IPs) in imaging are prevalent across a wide range of application fields in signal and image processing, and computer vision, such as image denoising, deblurring, segmentation, or including imaging reconstruction tasks. Computed tomography (CT), Electrical Impedance tomography (EIT), Positron Emission Tomography (PET), and Magnetic Resonance Imaging (MRI) are some examples of reconstruction imaging problems. Most of the IPs in imaging are ill-posed and dealing with this issue is crucial in applications where decisions are made based on the recovered model parameters, such as in medical imaging for diagnosis. Traditional studies on IPs have mainly concentrated on identifying conditions under which solutions to these ill-posed issues can be found, as well as techniques for approximating solutions in a stable manner when noise is given. The most popular imaging models and reconstruction methods are reformulated in analytical manner using the variational framework which includes regularization term. Regularized solutions are obtained through solving optimization methods for linear and non-linear functional. Additionally, the presence of not differentiability, non-convexity, and constraints are considered. This variational formulation is focused on the study of regularization techniques in order to promote properties of the solutions such as sparsity or its representation in appropriate spaces. The goal of the regularization step is to overcome the ill-posedness of the problem since such problems are extremely sensitive to perturbations (e.g. noise) in the data: small perturbation in the input can cause an arbitrarily large perturbation of the solution. In the Bayesian framework, the image prior serves as a regularizer in the computation of a maximum-a-posterior (MAP) estimation of the reconstructed image. Developing efficient variational models is challenging, as it necessitates a thorough understanding of the underlying physics of the acquisition process, and requires intricate modeling of the statistics of the unknowns.

In the field of inverse problems in imaging, deep learning (DL) approaches that utilize neural networks with multiple internal layers have gained popularity in the past decade. DL methods have demonstrated the ability to effectively use prior information from training data in order to address the uncertainty of solutions to ill-posed inverse problems. However, there is a significant gap in understanding the mathematical principles behind the exceptional performance of DL methods in solving IPs. The main purpose of this thesis is to discuss data-driven approaches to regularize and then solve inverse problems of image reconstruction. We introduce several new methods merging the advantages of both variational and deep learning frameworks. Exploiting the capacity of DL models to characterize the input data and we propose different approaches as effective ways to incorporate domain knowledge into DL models.

Contribution and Outline

This thesis is organized into seven chapters. In the first Chapter, we illustrate some preliminaries on inverse problems, and numerical strategies to solve them in a model-based mathematical framework considering linear and non-linear forward operators. In Chapter 2 data-driven approaches related to deep-learning techniques are described with a particular focus on end-to-end models, hybrid models, and learning regularization strategies. In Chapter 3 we describe the mathematical formulation of the Electrical Impedance Tomography problem as the main use case purposed in the numerical experiments. The last four chapters rely on the learning models applied to the image reconstruction inverse problems:

- In Chapter 4 we investigate the solution to EIT inverse problem. The non-linear approach to this challenging problem commonly relies on the iterative regularized Gauss-Newton method, which, however, has several drawbacks: the critical choice of the regularization matrix and parameter and the difficulty in reconstructing step changes, as smooth solutions are favored. We address these problems by learning a data-adaptive neural network as the regularization functional and integrating a local anisotropic total variation layer as an attention-like function into an unrolled Gauss-Newton network. We finally show that the proposed learned non-linear EIT approach strengthens the Gauss-Newton approach providing robust and qualitatively superior reconstructions.
- In Chapter 5 we propose a proximal Gauss-Newton method for the penalized nonlinear least squares optimization problem arising from the regularization of ill-posed nonlinear inverse problems. By exploiting the modular structure that characterizes the proximal-type methods, we plug in a pre-trained neural net denoiser in place of the standard proximal map. This allows to mould the prior on the data. An encoder-decoder deep convolutional network is proposed as a denoiser, which works on unstructured data; its mathematical formulation is derived to analyse the Lipschitz condition. With the intent of showing the benefits of applying deep-PnP reconstruction, we consider as an exemplar application, the nonlinear EIT, a promising non-invasive imaging technique mathematically formulated as a highly nonlinear ill-posed inverse problem.
- In Chapter 6 we state the Limited Electrode problem in EIT, and propose solutions inspired by the application of compressive sensing techniques and deep learning strategies on the raw boundary impedance data, in order to recover the target reconstruction quality while using a relatively low number of nonlinear measurements.
- In Chapter 7 we investigate the potential of Transformer neural networks (Tnets), initially proposed for sequence to sequence learning, by adapting them to image data. This question sparked immense interest in the computer vision community and it is based on the scheme for replacing convolution with self-attention. Given an image, without specific constraints on patch size, Tnets extract patches from the input data and feed them into a Transformer encoder to obtain a global representation, which will finally be transformed for a specific task. In our case, we built Vformes, a Tnet-based architecture, as a deep learning-based

denoiser applied and post-processing in the Computed Tomography inverse problem. This contribution will be further developed in future work.

Related Publications

This thesis mainly refers in chapters 4, 5 and 6 to the following published papers:

- Colibazzi, F., Lazzaro, D., Morigi, S., & Samoré, A. (2022). *Learning nonlinear electrical impedance tomography*. Journal of Scientific Computing, 90(1), 58.
- Colibazzi, F., Lazzaro, D., Morigi, S., & Samoré, A. (2022). *Deep-Plug-and-Play proximal Gauss-Newton methods with applications to the nonlinear, inverse Electrical Impedance Tomography problem*. Inverse Problems and Imaging. doi: 10.3934/ipi.2023014
- Colibazzi, F., Lazzaro, D., Morigi, S., & Samoré, A. (2023). *Limited Electrodes Models in Electrical Impedance Tomography reconstruction*. SSVM Proceedings, Springer, In: Calatroni L., Donatelli M., Morigi S., Prato M., Santacesaria M. (eds) Scale Space and Variational Methods in Computer Vision. SSVM 2023. Lecture Notes in Computer Science, vol 14009. Springer Cham. 2023, pp.66-78.

Chapter 1

Ill-posed inverse problems and numerical optimization strategies

In this chapter, we give the main definitions of ill-posed inverse problems (IPs). We will consider various related examples and then we report some optimization techniques for the solution of IPs. A practical introduction to linear and non-linear IPs from both a theoretical and computational perspective is given by [MS12]. In [SW12] non-linear IPs and their applications like electrical impedance tomography (EIT), X-ray computed tomography (CT), magnetic resonance imaging (MRI), and diffuse optical tomography (DOT) are dealt with by understanding the underlying physical phenomena as well as the implementation details of image reconstruction algorithms.

1.1 Ill-posed inverse problems

Given two Hilbert spaces X and Y called *data space* and *measurement space* respectively, let $\mathcal{D}(F) \subset X$ the domain of interest, we consider a physical system where the variables and parameters interact with each other. A mathematical formulation of the direct problem or forward problem of this model is given by:

$$y = F(x) \tag{1.1}$$

where $F : \mathcal{D}(F) \rightarrow Y$ is the linear or non-linear *forward map* and $x \in X$ is the input data, $y \in Y$ the measurements data. We observe that if F is linear we have $F(x) = Fx$. In practice, the measurements data y in (1.1) are almost never available precisely then these are only an approximation y^δ to y satisfying

$$\|y^\delta - y\|_Y \leq \delta \tag{1.2}$$

where $\delta > 0$ is the noise level and $\|\cdot\|_Y$ is the norm induced by the inner product on Y . Informally we can see a direct problem as the search for an effect from a cause. An example is knowing how X-rays waves attenuate as they pass through a body in response to the size, shape, and material composition of the body. Knowing the cause, we can determine the attenuation of the radiation. Basically, in terms of input/output values, solve a forward problem means to find the output given input. The inverse problem instead can be viewed as to recover the original input x in the data space X from a measured output $y^\delta \in Y$. Considering the previous example, the inverse problem

is to reconstruct the inner structure of an unknown physical body from the knowledge of X-ray images collected from different directions. Most inverse problems are complicated to solve: we may not be able to control the input precisely because we have technical limitations and the measured output will always be contaminated by noise. In this thesis we will concentrate on *ill-posed inverse problems*. First of all, we introduce the concept of *well-posed problem* proposed by Hadamard in [Had02] according to which a well-posed problem has the three following properties:

- **Existence:** there should be at least one solution.
- **Uniqueness:** there should be at most one solution.
- **Stability:** the solution must depend continuously on data.

If at least one of these requirements is not met, the problem is *ill-posed*. The definition of well-posedness above is equivalent to the request that F is surjective and injective and that the inverse mapping F^{-1} is continuous. The non-linearity of the problem implies the failure of uniqueness, continuous dependence on data, and the existence properties of the solution. The reason is the fact that the inverse problems are highly sensitive to noisy measurements. More generally, every non-degenerate compact operator between infinite-dimensional Hilbert spaces whose range is infinite naturally leads to ill-posed inverse problems. Research on inverse problems has aimed to identify the conditions under which solutions to these types of ill-posed problems exist and to develop methods for approximating solutions in a stable manner when noise is present as in [EHN96], [BB18], [K⁺11].

1.1.1 Analysis of non-linear ill-posedness

Uniqueness

In non-linear inverse problem if the uniqueness does not hold, the problem can be regarded by studying the quotient space $\{[x] | x \in X\}$ related to the equivalence classes of objects yielding the same data

$$[x] = \{z \in X | F(z) = F(x)\}.$$

Then the representative satisfies the minimal norm solution $\|\cdot\|_X$ of an equivalence class can be considered as good approximation of the ideal solution. Otherwise, a good possibility is to measure more data or enrich your knowledge of measurement using a-priori information. These strategies are called *regularization*.

Continuous dependence on data

Given the uniqueness property, the problem of the continuous dependence on data depends on the fact that the operator F might not have a continuous inverse on $F(D(F))$. We can enforce the uniqueness analysis by adding conditional stability results given by:

$$\|x - z\|_X \leq \Phi(\|F(x) - F(z)\|_Y) \tag{1.3}$$

where x and z are two solutions belonging to the same equivalence class, and $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ is a non-negative continuous scalar function satisfying $\Phi(0) = 0$. The inequality (1.3) estimates bound the difference $\|x - z\|_X$ by the difference $\|F(x) - F(z)\|_Y$ in infinite-precision data. Basically the conditional stability are rarely applicable because the measured data $y \notin F(D(F))$.

Existence

The problem of existence relies on the fact that often $y \notin F(D(F))$ where $F(D(F)) \subset Y$, Y infinite dimensional space. Then we have a problem with the existence of a preimage $F^{-1}(y)$. To overcome this problem, a classical strategy is to model the measurement function $y = F(x)$ using, in the discrete formulation, a projection operator $P_k : Y \rightarrow Y_k \subset Y$ where Y_k satisfies the following isometry $Y_k \simeq \mathbb{R}^k$. Then we can replace the non-linear problem with the following $y = P_k(F(x)) + \epsilon$.

1.1.2 Variational Regularization method

The first approach to tackle an inverse problem consists of applying the *variational regularization methods* which impose certain regularity conditions on the desired solution to single out a useful and stable result. Regularization methods give us a function from the data space to the model space close to the inverse of the forward operator. In the discrete setting we consider as data space $X = \mathbb{R}^n$ and $Y = \mathbb{R}^m$ then $x \in \mathbb{R}^n$ and $y^\delta \in \mathbb{R}^m$ and the forward operator F is a matrix of size $m \times n$. The regularization approach consists to reformulate the ill-posed problem with a well-posed, consisting of the minimization of a cost functional $\mathcal{J} : \mathbb{R}^n \rightarrow \mathbb{R}_+$. This functional is the sum of two terms which are determined by some statistical assumptions on the noise and the type of image we are considering. In formula:

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \{\mathcal{J}(x; \lambda) = \mathcal{L}(x; y^\delta, F) + \lambda R(x)\} \quad (1.4)$$

where:

- $\mathcal{L}(x; y^\delta, F) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the *fidelity term*. It measures the prior information on the residue.
- $R(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is called *regularization term* which contains the prior information on the image imposing e.g., regularity or sparsity.
- $\lambda \in \mathbb{R}^+$ is the *regularization parameter*. It allows setting the trade-off between fidelity and the regularization terms. Its setting is usually a quite delicate issue.

1.1.3 The discrepancy principle

The choice of the regularization parameter in the iterative methods is crucial to obtain a good solution of the problem. Substantial literature related to inverse problems in imaging has been devoted to the derivation of methods for automatically tuning the regularization parameter λ . In general, we do not know the exact solution of the problem then is more difficult to set the regularization parameter. In [Han98] the authors propose the *discrepancy Morozov principle*. At

the core of the discrepancy principle there is the idea that, typically, no benefit is expected when the problem is solved more accurately than the accuracy of the data. Assuming the noisy data as in (1.2), the discrepancy principle thus consists in looking for a solution x belonging to the discrepancy set, defined as

$$D_{mor} := \{x \in \mathbb{R}^n \quad \text{s.t.} \quad \tau_1 \delta \leq \|F(x) - y^\delta\|_2 \leq \tau_2 \delta\}, \quad (1.5)$$

where $\tau_2 \approx 1$ is a parameter avoiding the under-estimation for $\tau_2 > 1$ and the over-estimation for $\tau_2 < 1$ of the noise. Theoretically, δ is the upper bound for the noise level, hence one should only consider $1 < \tau_1 \leq \tau_2$.

1.2 Variational regularization models for linear problems

Given the discrete formulation of (1.1) we can determine the degree of ill-conditioning of the linear inverse problems by studying the singular value decomposition (SVD).

Theorem 1.1. *Let $F \in \mathbb{R}^{m \times n}$ be a real matrix. Then the SVD factorization of F is given by*

$$F = U \Sigma V^T = \sum_{i=1}^n u_i \sigma_i v_i^T, \quad (\Sigma = U^T F V)$$

where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix in case F being square, and as close as possible to a diagonal matrix in case F being wide, with diagonal entries satisfying

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_{\min(m,n)} = 0,$$

$r = \text{rank}(F)$, σ_i are the singular values of F and $U = [u_1, u_2, \dots, u_m] \in \mathbb{R}^{m \times m}$ and $V = [v_1, v_2, \dots, v_n] \in \mathbb{R}^{n \times n}$ columns u_j and v_j are left and right singular vectors of F respectively s. t. $U^T U = V^T V = I$.

When F is square and non-singular the inverse F^{-1} can be determined easily from whose SVD is inherited from the SVD of F , namely:

$$F^{-1} = V \Sigma^{-1} U^T,$$

where Σ^{-1} is the diagonal matrix with entries: $1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r, 1/\sigma_{r+1} = \dots = 1/\sigma_{\min(m,n)} = 0$. If any of the singular values $\sigma_i = 0$, then the Σ^{-1} does not exist. In practical situations, a matrix may have singular values that are not exactly equal to zero but are so close to zero that it is not possible to accurately compute them. In such cases, the matrix is what we call ill-conditioned because dividing by the singular values ($1/\sigma_i$) for singular values σ_i that are arbitrarily close to zero will result in numerical errors. The degree to which ill-conditioning prevents a matrix from being inverted accurately depends on the ratio of its largest to smallest singular value, a quantity known as the *condition number* defined as:

$$\text{Cond}(F) = \frac{\sigma_1}{\sigma_r}.$$

When the condition number may be large we have a problem related to the inversion of F since the diagonal inverse matrix Σ^{-1} contains floating point numbers of very different sizes. Therefore

the condition number is assumed to be large for ill-posed problems and it tends to infinity for rank-deficient matrices F such that $r = \text{rank}(F) < \min(m, n)$.

The concept of least-squares inversion is the most used method for solving inverse problems. We want to find the minimum-norm solution defined as:

Definition 1.1. A vector $x^* \in \mathbb{R}^n$ is called a least-squares solution of the linear system $Fx = y^\delta$ if

$$\|Fx^* - y^\delta\| = \min_{x \in \mathbb{R}^n} \|Fx - y^\delta\|.$$

Further, x^* is called the minimum-norm solution if

$$\|x^*\| = \inf\{\|x\| : x \text{ is a least-squares solution of } Fx = y^\delta\}.$$

More in general, when F is not invertible, we can still consider *the Moore-Penrose* or *pseudoinverse* matrix F^\dagger and introduce its SVD.

Theorem 1.2. Let $F \in \mathbb{R}^{m \times n}$ and denote by $F = U\Sigma V^T$ the SVD of F . The minimum-norm solution of the discretized problem $Fx = y^\delta$ is given by $F^\dagger y^\delta$ where

$$F^\dagger y^\delta = V\Sigma^+ U^T y^\delta$$

and $\Sigma^\dagger = \text{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r, 0, \dots, 0)$.

Then the theorem shows that applying SVD the solution of the linear inverse problem is given by:

$$x^* = \sum_{i=1}^n \frac{\langle u_i^T, y^\delta \rangle}{\sigma_i} v_i.$$

Typically the matrix F presents very small singular values which can lead to the amplification of small perturbations in the data. We thus expect the condition number $\text{Cond}(F)$ to be very large and F to be very ill-conditioned. As consequence, its inversion is not feasible and different strategies must be proposed to address the solution of the far from harmless linear system:

- Truncated SVD (TSVD) Regularization
- Tikhonov Regularization

1.2.1 Truncated Singular Value Decomposition (TSVD) Regularization

Assume that the matrix is either square or overdetermined i.e. $F \in \mathbb{R}^{m \times n}$ with $m \gg n \geq r = \text{rank}(F)$. To reduce noise amplification and obtain an acceptable solution, we have to discard the smallest singular values of matrix F thus using a low-rank but better-conditioned matrix. We thus consider a threshold τ and set to zero all the singular values of Σ smaller than τ . τ is called *regularization parameter* and it corresponds to considering only the first k singular values s.t. $\sigma_1 \geq \dots \geq \sigma_k > \tau$, $k < r$. The minimum-norm solutions using the truncated singular values decomposition (TSVD) [Han87] method is given by:

$$x_{TSVD}^* = F^\dagger(y^\delta) \approx V_k \Sigma_k^\dagger U_k^T(y^\delta) = \sum_{i=1}^k \frac{\langle u_i^T, y^\delta \rangle}{\sigma_i} v_i$$

where the TSVD matrix is defined as:

$$\Sigma_k^\dagger = \text{diag}(1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_k, 0, \dots, 0) \in \mathbb{R}^{n \times m}.$$

We can equivalently write:

$$x_{TSVD}^* = \sum_{i=1}^n w(\sigma_i) \frac{\langle u_i^T, y^\delta \rangle}{\sigma_i} v_i$$

where

$$w(\sigma_i) = \begin{cases} 1 & \text{if } \sigma_i \leq \tau \\ 0 & \text{if } \sigma_i > \tau \end{cases}$$

$w(\sigma_i)$ represents the *filter function* that eliminates the smallest singular values, whereas the singular values above the truncation threshold are not modified. The truncation threshold τ determines the number of SVD components be kept in the regularized solution and, hence, the choice of its value is crucial for the TSVD method.

1.2.2 Tikhonov Regularization

Another common regularization method used to incorporate the regularity requirement in the problem formulation is Tikhonov Regularization. The regularized solution is determined as the solution x_{tik}^* of the unconstrained minimization problem:

$$x_{tik}^* = \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Fx - y^\delta\|_2^2 + \frac{\lambda}{2} \|x\|_2^2 \right\}.$$

The first term in the functional measures the data consistency, i.e. measure how well solution x_{tik}^* predicts data y^δ ; the second term instead measures the regularity of the solution. $\lambda > 0$ is the regularization parameter which balances the contribution of the regularization term. In order to penalize different features of the final solution we can consider the generalized Tikhonov regularization. In formula:

$$x_{tik}^* = \underset{x}{\operatorname{argmin}} \left\{ \frac{1}{2} \|Fx - y^\delta\|_2^2 + \frac{\lambda}{2} \|Lx\|_2^2 \right\}.$$

where $L \in \mathbb{R}^{p \times n}$, $m \geq n \geq p$ represents the discretization matrix of some differential linear operator. If the fidelity term is too large, the solution x_{tik}^* cannot be considered a good solution because it differs too much from the perturbed input. On the other hand, if the residue is too small, the solution will be contaminated by noise. The regularization term is based on the knowledge that the solution is dominated by the high-frequency components which are large in modulus. Hopefully, by checking the x norm, most components with noise are suppressed. The regularization parameter balances the two terms. The bigger it is, the more weight it gives after the regularization. Conversely, if it is too small, the solution will be too irregular. The aim is to find a good balance between the two terms through the regularization parameter. Under certain assumptions, the Tikhonov formulation admits a unique solution as explained in [Tik43], [Tik63] [TA77].

Theorem 1.3. *Let*

$$x_{tik}^* = \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Fx - y^\delta\|_2^2 + \frac{\lambda}{2} \|Lx\|_2^2 \right\}.$$

the Tikhonov formulation with $\lambda > 0$. If $\text{Ker}(F) \cap \text{Ker}(L) = \{0\}$, then the Tikhonov solution is unique and is given by:

$$x_{tik}^* = (F^T F + \lambda L^T L)^{-1} F^T y^\delta. \quad (1.6)$$

Given the SVD of $F = U \Sigma X^{-1}$ and $L = V M X^{-1}$, then the generalized Tikhonov's solution of (1.6) is given by:

$$x_{tik}^* = \sum_{i=1}^{r_L} \frac{\gamma_i}{\gamma_i^2 + \lambda} \langle u_i^T, y^\delta \rangle x_i + \sum_{i=r_L+1}^n \langle u_i^T, y^\delta \rangle x_i$$

where $r_L = \text{rank}(L) \geq p$ and $\gamma_i = \frac{\sigma_i}{m_i}$ for $i = 1, \dots, r_L$. To better note the analogy between the solution just found by Tikhonov regularization and the solution by TSVD, we can write

$$x_{tik}^* = \sum_{i=1}^n \frac{\sigma_i}{\sigma_i^2 + \lambda} \langle u_i^T, y^\delta \rangle v_i = \sum_{i=1}^n \frac{\sigma_i^2}{\sigma_i^2 + \lambda} \frac{\langle u_i^T, y^\delta \rangle}{\sigma_i} v_i.$$

As in TSVD, we can introduce the *filter factors* $w(\sigma_i) = \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$ which attenuate the components $\sigma_i < \lambda$. For singular values $\sigma_i \gg \lambda$, the filter factors are close to 1 and the corresponding components contribute to the solution x_{tik}^* with almost all their magnitude. If instead $\sigma_i \ll \lambda$ the solution components are filtered.

1.2.3 Total variation regularization

The most common regularization strategies impose (explicitly or implicitly) a penalty on nonsmooth regions in a reconstructed image. Such methods confer stability to the reconstruction process but limit the capability of describing sharp variations in the sought solution. One technique to permit image regularization without imposing smoothing is the *Total Variation* (TV) formulation of regularization introduced in [ROF92].

Definition 1.2. Given a 2-dimensional matrix $u \in \mathbb{R}^{m \times n}$, with $N = mn$, we define the discrete isotropic total variation regularizer as:

$$TVI(u) = \sum_{j=1}^N \|(Lu)_j\|_2 = \sum_{j=1}^N \sqrt{(L_h u)_j^2 + (L_v u)_j^2} \quad (1.7)$$

where $(Lu)_j = ((L_h u)_j, (L_v u)_j)$ is the discrete gradient operator.

Definition (1.7) can be slightly modified by considering the l_1 -norm instead of the euclidean norm, thus getting to the anisotropic total variation

$$TVA(u) = \sum_{j=1}^N \|(Lu)_j\|_1 = \sum_{j=1}^N |(L_h u)_j| + |(L_v u)_j|. \quad (1.8)$$

TVA is a more suitable choice in presence of images presenting edges oriented only along the x and the y axes. While the diffusion produced by TVI is the same in every direction of the xy cartesian plane, the choice of TVA produces a more concentrated diffusion along the axes. This is mainly due to the properties of the level curves of the two regularization terms, reported in Figure 1.1.

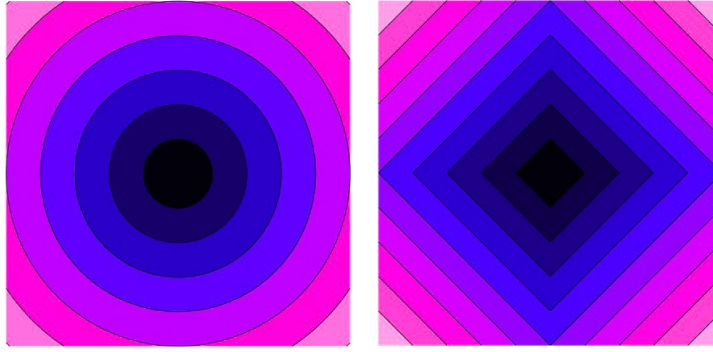


Figure 1.1: Level curve of TVI (left) and TVA (right) regularization penalty.

Given the TVA penalty, the variational formulation of our optimization problem takes the form of:

$$x_{tik,tv}^* = \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|Fx - y^\delta\|_2^2 + \frac{\lambda}{2} \sum_{i=1}^N \|(Lx)_i\|_2 \right\} \quad (\text{TV-L2})$$

Since the TV regularizer is not a differentiable term, we don't compute the gradient-based minimization algorithms. One possible choice to solve this kind of non-differentiable optimization problem is the Alternating Direction Method of Multipliers (ADMM) method [GM76]. At the core of ADMM there is the idea of splitting the original problem into smaller ones that are easier to handle. The solution of the smaller sub-problems thus coordinates in order to find a solution for the large problem.

1.2.4 ADMM method for TV-L2

Introducing an auxiliary variable $t \in \mathbb{R}^2$, we can reformulate the problem (TV-L2) as constrained minimum problem in two variables:

$$\{\hat{x}, \hat{t}\} = \arg \min_{x,t} \left\{ \sum_{i=1}^N \|t_i\|_2 + \frac{\lambda}{2} \|Fx - y^\delta\|_2^2; \text{ subject to } t_i = (Lx)_i \quad \forall \quad i = 1, \dots, N \right\}.$$

The associated Lagrangian function is:

$$L(x, t, \rho; \beta) = \sum_i \left(\|t_i\| - \rho_i^T (t_i - (Lx)_i) + \frac{\beta}{2} \|t_i - Lx_i\|^2 \right) + \frac{\lambda}{2} \|Fx - y^\delta\|_2^2,$$

where we refer to β as the penalty parameter. The ADMM iterative method give us the sequences $\{t^k\}, \{x^k\}, \{\rho^k\}$ defined via the following iterative scheme:

$$\begin{aligned} (\text{primal}) \quad & \begin{cases} t^{k+1} &= \arg \min_y L(x^k, t, \rho^k; \beta) \\ x^{k+1} &= \arg \min_x L(x, t^{k+1}, \rho^k; \beta) \end{cases} \\ (\text{dual}) \quad & \begin{cases} \rho^{k+1} &= \rho^k - \beta(t^{k+1} - Lx^{k+1}). \end{cases} \end{aligned}$$

The minimization over t is splitted in N sub-problems as:

$$t_i^{k+1} = \arg \min_{t_i \in \mathbb{R}^2} \left\{ \|t_i\| + \frac{\beta}{2} \left\| t_i - \left((Lx^k)_i + \frac{1}{\beta} (\rho^k)_i \right) \right\|^2 \right\}, \quad i = 1, \dots, N,$$

which are as unique solution:

$$t_i^{k+1} = \max \left\{ \left\| (Lx^k)_i + \frac{1}{\beta}(\rho^k)_i \right\| - \frac{1}{\beta}, 0 \right\} \frac{(Lx^k)_i + \frac{1}{\beta}(\rho^k)_i}{\left\| (Lx^k)_i + \frac{1}{\beta}(\rho^k)_i \right\|}, \quad i = 1, \dots, d.$$

The x minimization is a least square problem that has as solution the solution of a the following linear system:

$$\left(L^T L + \frac{\mu}{\beta} F^T F \right) x = L^T \left(t^{k+1} - \frac{1}{\beta} \rho^k \right) + \frac{\mu}{\beta} F^T b.$$

As stopping criterium for the ADMM iterative algorithm we can use the relative error between two solution steps:

$$\frac{\|x^{k+1} - x^k\|}{\max\{\|x^k\|, 1\}} < \epsilon.$$

1.3 Variational regularization models for non-linear problems

In this section, we will focus on non-linear forward operator.

1.3.1 Tikhonov regularization for non-linear inverse problems

In [EKN89] the authors consider non-linear ill-posed problems in a Hilbert space setting, they show that Tikhonov regularization is a stable method for solving non-linear ill-posed problems and give us conditions that guarantee the convergence for the regularised solutions.

Definition 1.3. A function $F : X \rightarrow Y$ is *weakly closed* if for any sequence $\{x_n\} \subset X$, weak convergence of x_n to x in X and weak convergence of $F(x_n)$ to $y \in Y$ implies $x \in X$ and $F(x) = y$.

Similarly to the linear case, we define the concept of x^* minimum-norm solution for the non-linear forward operator:

$$\|F(x^*) - y^\delta\| = \min_{x \in \mathbb{R}^n} \{ \|F(x) - y^\delta\| \quad \text{s.t.} \quad x \in D(F) \}$$

and

$$\|x^* - x_0\| = \min_{x \in \mathbb{R}^n} \{ \|x - x_0\| / \|F(x) - y^\delta\| = \|F(x^*) - y^\delta\| \}.$$

Here x_0 plays the role of a selection criterion, then its choice influences which solution we want to approximate. Let F continuous and weakly closed function and recalling the fact that y in (1.1) is an approximated measurement data which satisfies (1.2). For convenience here we will make this fact explicit, then we consider the following Tikhonov regularized optimization problem:

$$\min_{x \in \mathbb{R}^n} \|F(x) - y^\delta\|^2 + \lambda \|x - x_0\|^2, \quad (1.9)$$

$x \in D(F)$. Another required assumption for the following results is the attainability property which means that exists an element $x \in D(F)$ such that:

$$F(x) = y.$$

Since F is non-linear, the solution will not be unique, in general. Concerning the study of the stability in the sense of continuous dependence of the solution it is exhibited in the following theorems:

Theorem 1.4. *Let $\lambda > 0$ and let $\{y_k\}$ and $\{x_k\}$ be sequences where $y_k \rightarrow y^\delta$ and x_k is a minimizer of (1.9) with y^δ replaced by y_k . Then there exists a convergent subsequence of $\{x_k\}$ and the limit of every convergent subsequence is a minimizer of (1.9).*

Another question is if the solution of (1.9) converges towards a solution of (1.1) as $\lambda \rightarrow 0$ and $y^\delta \rightarrow y$. To solve this question the authors consider a slightly generalization of (1.9) and examine the problem of finding an element $x_{\lambda}^{\delta,\eta} \in \Omega$ such that:

$$\|F(x_{\lambda}^{\delta,\eta}) - y^\delta\|^2 + \lambda \|x_{\lambda}^{\delta,\eta} - x_0\|^2 \leq \|F(x) - y^\delta\|^2 + \lambda \|x - x_0\|^2 + \eta \quad (1.10)$$

where $\eta \geq 0$ reflects computational reality, since the problem (1.9) can only be solved approximately. Given the attainability assumption and since F is weakly closed the following theorem shows that under certain conditions, solution of (1.10) converges towards an x^* minimum-norm solution. We observe that x^* need not be unique.

Theorem 1.5. *Let $F(x) = y$, let $y^\delta \in Y$ with $\|y - y^\delta\| \leq \delta$ and let $\lambda(\delta, \eta)$ be such that $\lambda(\delta, \eta) \rightarrow 0$, $\delta^2/\lambda(\delta, \eta) \rightarrow 0$ and $\eta/\lambda(\delta, \eta) \rightarrow 0$ as $\delta \rightarrow 0$, $\eta \rightarrow 0$. Then every sequence $\{x_{\lambda_k}^{\delta_k, \eta_k}\}$, where $\delta_k \rightarrow 0$, $\eta_k \rightarrow 0$, $\lambda_k := \lambda(\delta_k, \eta_k)$ and $x_{\lambda_k}^{\delta_k, \eta_k}$ is a solution of (1.10), has a convergent subsequence. The limit of every convergent subsequence is a x^* minimum-norm solution. If, in addition, the x^* minimum-norm solution is unique, then*

$$\lim_{\delta \rightarrow 0, \eta \rightarrow 0} x_{\lambda(\delta, \eta)}^{\delta, \eta} = x^*.$$

1.3.2 Morozov's discrepancy for Tikhonov regularization

In this section we report some results regardless the Morozov's discrepancy principle (1.5) as an a-posteriori parameter choice rule for Tikhonov regularization functional using non-linear operators. Given the problem (1.1) under the assumption (1.2), we consider a general Tikhonov regularization functional i.e.

$$\mathcal{J}_{\lambda, q}(x) = \|F(x) - y^\delta\|^q + \lambda R(x),$$

$\lambda > 0$, $q > 0$, R a general convex penalty term. In [AR09] the authors analyze that under what circumstances the discrepancy principle can be applied to the non-linear inverse problem (1.1) namely they are interested in choosing $\lambda = \lambda(\delta, y)$ such that:

$$\tau_1 \delta \leq \|F(x_{\lambda}^{\delta}) - y^\delta\| \leq \tau_2 \delta, \quad (1.11)$$

where $\tau_1, \tau_2 > 0$ and x_{λ}^{δ} denotes the regularized solution. Define now the following condition:

Condition 1. Let $R : D(R) \subset X \rightarrow \mathbb{R}^+$, with $0 \in D(R)$ be a convex functional such that

1. $R(x) = 0$ if and only if $x = 0$.
2. R is weakly lower semi-continuous.
3. R is weakly coercive, i.e. $\|x_n\| \rightarrow \infty$ implies $R(x_n) \rightarrow \infty$.

Definition 1.4. The regularized solution will be a minimizers x_λ^δ of the Tikhonov functional

$$J_\lambda(x) = \begin{cases} \|F(x) - y^\delta\|^q + \lambda R(x) & \text{if } x \in D(R) \cap D(F) \\ +\infty & \text{otherwise} \end{cases} \quad (1.12)$$

Since the minimizer of (1.12) will in general not be unique for non linear operator given a fixed y^δ , we denote the set of all minimizers by M_λ :

$$M_\lambda = \{x_\lambda^\delta \in X : J_\lambda(x_\lambda^\delta) \leq J_\lambda(x), \forall x \in X\}. \quad (1.13)$$

A solution x^* of (1.1) is called R -minimizing solution if

$$R(x^*) = \min\{R(x) : F(x) = y\}$$

and denote the set of all R -minimizing solution by \mathcal{S} . We assume that $\mathcal{S} \neq \emptyset$.

At this level, we show that the following conditions suffice to guarantee the existence of a positive regularization parameter fulfilling the discrepancy principle.

Condition 2. Assume that y^δ satisfies

$$\|y - y^\delta\| \leq \delta \leq \tau_2 \delta < \|F(0) - \tau^\delta\|,$$

and that there is no $\lambda > 0$ with minimizers $x_1, x_2 \in M_\lambda$ s.t.

$$\|F(x_1) - y^\delta\| < \tau_1 \delta \leq \tau_2 \delta < \|F(x_2) - y^\delta\|.$$

Theorem 1.6. *If Condition 2 is fulfilled, then there are $\lambda = \lambda(\delta, y^\delta) > 0$ and $x_\lambda^\delta \in M_{\lambda(\delta, y^\delta)}$ such that (1.11) holds.*

The next paragraph contains a result that shows that for a suitable regularizer, the penalized solution converges with respect to the penalty term as the noise level goes to zero.

Condition 3. Let $\{x_n\} \subset X$ be such that $x_n \rightarrow \bar{x} \in X$ and $R(x_n) \rightarrow R(\bar{x}) < \infty$; then x_n converges to \bar{x} with respect to R , i.e.

$$R(x_n - \bar{x}) \rightarrow 0.$$

In [GHS08] is shown that the choice of a penalty term defined as weighted l_p -norm of the coefficient with respect to some frame $\{\phi_\pi\}_{\pi \in \Pi} \subset X$

$$R_{p,w} := \|x\|_{w,p} = \left(\sum_{\pi \in \Pi} w_\pi |\langle x, \phi_\pi \rangle|^p \right)^{1/p}, \quad 1 \leq p \leq 2$$

with $0 < w_{\min} \leq w_\pi$, satisfies the Conditions 3 and 1.

Theorem 1.7. *Let $\delta_n \rightarrow 0$ and F, R satisfy Conditions 3 and 1. Assume that y^{δ_n} fulfills Condition 2 and choose $\lambda_n = \lambda(\delta_n, y^{\delta_n})$, $x_n \in M_{\lambda_n}$ such that (1.11) holds; then x_n converges to \mathcal{S} with respect to R .*

1.4 Numerical Optimization Algorithms

Consider the following minimization problem:

$$\min_{x \in \mathbb{R}^n} \{ \mathcal{J}(x; \lambda) := f(x) + \lambda g(x) \}, \quad (1.14)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a convex, continuously differentiable fidelity function, and $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ is a convex but not necessarily differentiable penalty function, $\lambda > 0$. All numerical methods for the solution of non-linear and thus, in general, non-convex optimization problems are iterative: from a starting point x_0 the method produces a sequence of vectors x_1, x_2, \dots , which converges to a stationary point x^* , which is (hopefully) a local minimizer for the given cost function. If $x^* \in \mathbb{R}^n$ is a local minimum of (1.14), then it is a stationary point of (1.14), i.e., one has

$$0 \in \nabla f(x^*) + \partial g(x^*),$$

where $\partial g(\cdot)$ is the subdifferential of g .

1.4.1 First-order Methods

The most popular methods for minimizing composite functions are first-order methods that use proximal mappings to handle the nonsmooth part g .

Proximal Gradient Method

Proximal methods approximate near a generic iterate x_k the smooth part f of (1.14) with a local quadratic model, thus considering as \mathcal{J} approximant at a given y , the following model

$$\bar{\mathcal{J}}(y;) = f(x_k) + \nabla f(x_k)^T (y - x_k) + \frac{1}{2} (y - x_k)^T H_k(x_k) (y - x_k) + \lambda g(y), \quad (1.15)$$

where H_k is a symmetric positive definite matrix which approximates the Hessian $\nabla^2 f(x)$.

The most popular methods for minimizing a composite function, formulated as in (1.14), are the *proximal gradient methods* which are first-order methods that use proximal mappings to handle the nonsmooth part g and consider an Hessian approximation in (1.15) equal to (a scaled version of) the identity, i.e. $H_k := \frac{1}{\beta} I$, $\beta > 0$.

$$\begin{aligned} z_{k+1} &= x_k - \beta \nabla f(x_k) \\ x_{k+1} &= \text{prox}_g^\beta(z_{k+1}) \end{aligned}$$

where the proximal mapping of a convex function g at z with parameter β is defined as

$$\text{prox}_g^\beta(z) := \arg \min_x \left\{ g(x) + \frac{1}{2\beta} \|x - z\|_2^2 \right\}.$$

FISTA [BT09] corresponds to an optimized approach to accelerating the proximal gradient method. Despite its simplicity, proximal gradient methods perform poorly, as they use only first-order information, and hence each iteration is less expensive but typically exhibits much slower convergence than a Newton-like method.

1.4.2 Second-Order Methods

Gauss-Newton Method

The Gauss-Newton (GN) method is the most commonly used one for minimizing non-linear least squares problems and performs a line search strategy with a specific choice of a descent direction. Consider the second-order Taylor's expansion approximation of the objective function at point x_k :

$$f(x_k + p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p + O(p^3). \quad (1.16)$$

Given an initial guess x_0 , Newton's method performs the next iteration

$$x_{k+1} = x_k + p_k$$

where the search direction p is computed by minimizing the function:

$$f_*(p_k) = f(x_k) + p_k^T \nabla f(x_k) + \frac{1}{2} p_k^T \nabla^2 f(x_k) p_k.$$

The search direction p_k is a minimum of f_* if the gradient of f_* vanishes on p i.e.

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

The Gauss-Newton method essentially approximates the Hessian matrix in (1.16) by ignoring all the second order terms from $\nabla^2 f(x_k)$, so that the search direction p is obtained by solving the following linear system:

$$J(x_k)^T J(x_k) p_k^{GN} = -J(x_k)^T r(x_k). \quad (GN)$$

Regularized Gauss-Newton method

The regularized Gauss-Newton method (RGN), using a generalized Tikhonov regularizer, consists of applying GN to the minimization problem (1.14)

$$x^* = \operatorname{argmin}_x \{ \mathcal{J}(x; \lambda) = f(x) + \lambda g(x) \}, \quad g(x) = \|Lx\|_2^2 \quad (1.17)$$

where L is a matrix representing either a discrete first/second order differential operator, or a positive diagonal matrix, or simply the identity matrix. The gradient and the approximated Hessian matrix of the objective function in (1.17) are respectively given as follows:

$$\nabla \mathcal{J}(x; \lambda) = J(x)^T r(x) + \lambda L^T L x, \quad (1.18)$$

$$\nabla^2 \mathcal{J}(x; \lambda) = \nabla^2 F(x) r(x) + J(x)^T J(x) + \lambda L^T L.$$

The search direction p_k from the current iterate satisfies the linear system

$$(J(x_k)^T J(x_k) + \lambda L^T L) p_k^{\text{RGN-Tik}} = J(x_k)^T r(x_k) + \lambda L^T L x_k. \quad (\text{RGN-Tik})$$

The Tikhonov-type regularization $g(x)$ in (1.17), regardless of the choice of L , favors smooth solutions and fails in reconstructing step conductivity changes. As we have previously seen, a

popular alternative to the Tikhonov regularizer, is the l_1 -norm Total Variation regularizer defined in (1.8). Replacing (1.8) in (1.17) enforces piecewise-constant conductivity reconstructions, and well preserves sharp conductivity changes, at the cost of a non-smooth optimization problem to solve. Following [BGAL09], the approximate solution of the problem (1.17) with a smoothed TV regularizer reads as

$$(J(x_k)^T J(x_k) + \lambda L^T E^{-1} L) p_k^{\text{RGN-TV}} = J(x_k)^T r(x_k) + \lambda L^T E^{-1} L x_k, \quad (\text{RGN-TV})$$

where E is a diagonal matrix defined as $E := \text{diag}(\sqrt{(Lx)^2 + \gamma})$, and $\gamma > 0$ is the smoothing parameter.

Levenberg-Marquardt Method

The Levenberg-Marquardt method [Han97] is another algorithm directly regularizes the ill-conditioned linear system (GN) adding a scaled identity matrix λI , or even λL , to the coefficient matrix, thus improving its condition number. Then the approximate solution of (1.17) is then given by

$$(J(x_k)^T J(x_k) + \lambda L^T L) p_k^{\text{LM}} = J(x_k)^T r(x_k). \quad (\text{LM})$$

Proximal Newton-type methods

Proximal Newton-type methods are second-order methods which rely instead on more accurate approximations H_k of the Hessian $\nabla^2 f(x_k)$ in (1.15). In particular, the *proximal Newton method* deals directly with $H_k := \nabla^2 f(x_k)$. Approximations to $\nabla^2 f(x_k)$, according to quasi-Newton strategies, lead to *proximal quasi-Newton methods*.

In general we can express the proximal Newton-type method using scaled proximal mapping, thus revealing the connection with the proximal gradient method.

Definition 1.5. Let $g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, convex function and Q be a positive definite matrix. Then the scaled proximal mapping of g at x is

$$\text{prox}_g^Q(x) := \arg \min_{y \in \mathbb{R}^n} \left\{ g(y) + \frac{1}{2} \|y - x\|_Q^2 \right\},$$

where $\|z\|_Q = \sqrt{z^T Q z}$ is the Q -norm.

As proved in [BFO19], the mapping $\text{prox}_g^Q(x)$ exists and is unique; moreover it is nonexpansive, i.e., for all $x, y \in \mathbb{R}^n$

$$\| \text{prox}_g^Q(y) - \text{prox}_g^Q(x) \|_Q^2 \leq \|y - x\|_Q^2,$$

and can be also implicitly defined as

$$\text{prox}_g^Q(x) = (I + Q^{-1} \partial g)^{-1}(x), \quad (1.19)$$

where ∂ denotes the subdifferential of g .

The proximal Newton-type method can be written as composite steps using scaled proximal mappings, where H_k plays the role of Q :

$$z_{k+1} = x_k - H_k^{-1} \nabla f(x_k) \quad (1.20)$$

$$x_{k+1} = \text{prox}_g^{H_k}(z_{k+1}). \quad (1.21)$$

In fact, by minimizing the local quadratic model (1.15) we get

$$x_{k+1} = \arg \min_y \left\{ \nabla f(x_k)^T (y - x_k) + \frac{1}{2} (y - x_k)^T H_k (y - x_k) + g(y) \right\} \quad (1.22)$$

$$= \arg \min_y \left\{ \frac{1}{2} \|\bar{Q}(x_k - H_k^{-1} \nabla f(x_k) - y)\|_2^2 + g(y) \right\}, \quad (1.23)$$

$$= \arg \min_y \left\{ \frac{1}{2} \|x_k - H_k^{-1} \nabla f(x_k) - y\|_{H_k}^2 + g(y) \right\}, \quad (1.24)$$

where in (1.23) we resort on the assumption that H_k is symmetric, positive definite hence is factorized as $H_k = \bar{Q}^T \bar{Q}$, and from (1.23) to (1.24) we apply

$$\|y - x\|_{H_k}^2 = \langle H_k(y - x), (y - x) \rangle = \langle \bar{Q}(y - x), \bar{Q}(y - x) \rangle = \|\bar{Q}(y - x)\|_2^2. \quad (1.25)$$

Due to the ill-posedness of the continuous problem, the approximation of the Hessian matrix H_k is ill-conditioned and, in general, even not positive definite.

Solution of RGN methods

The linear systems (RGN-Tik), (RGN-TV), and (LM) can be described by the following unified formulation

$$(J(x_k)^T J(x_k) + \lambda \mathcal{R}) p_k = b_k, \quad (\text{RGN})$$

where b_k is the associated right hand side and \mathcal{R} is a generic regularizer operator. Then the linear systems (RGN-Tik), (RGN-TV), and (LM) can be directly solved by

$$p_k = V_R \Sigma_R^{-1} U_R^T b_k, \quad (1.26)$$

where U_R , Σ_R , and V_R are given by the SVD of the coefficient matrix in (RGN-Tik), (RGN-TV) or (LM).

Algorithm 1 RGN

Input: $x_0, \lambda > 0$,

Output: x^* % optimal solution

for $k = 0, 1, \dots$, **do**

compute direction p_k by (1.26)

$x_{k+1} = x_k + p_k$

end for

$\sigma^* = \sigma_K$

end

In Algorithm 1 we summarize the iterative steps of the regularized Gauss-Newton algorithm where the convergence is satisfied when two successive iterates differ for a given threshold.

The success of all these approaches is based on the critical choice of the λ regularization parameter, that is, on a long and painful exhaustive research that in practical applications is not in general sustainable.

Chapter 2

Data driven techniques

Analyzing inverse problems through analytical methods (e.g. model-based) has been a longstanding topic of research. The analytical approach involves clearly defining the forward model, establishing criteria for a solution, and selecting a method to find the solution. With the advent of neural networks, many computer vision tasks have motivated attempts at using deep-learning (e.g. data-driven model) to achieve better performance in terms of efficacy and efficiency in many applications. Hybrid models that combine analytical and deep learning techniques have been proposed to address generalization issues while still maintaining the effectiveness of deep learning models. Fig. 2.1 shows an illustration of the connections between model-based methods, data-driven methods and hybrid method. In [OJM⁺20] the authors propose a taxonomy used to categorize different problems and reconstruction methods based on the knowledge or not of the forward operator and the use of this information in training and testing time. In [MHÖ⁺23] are studied the relevant notions of convergence for data-driven image reconstruction, which forms the basis of a survey of learned methods with mathematically rigorous reconstruction guarantees. A deep learning and hybrid methods review for solving imaging inverse problems, focusing on image and video super-resolution and image restoration is described in [LTMK21].

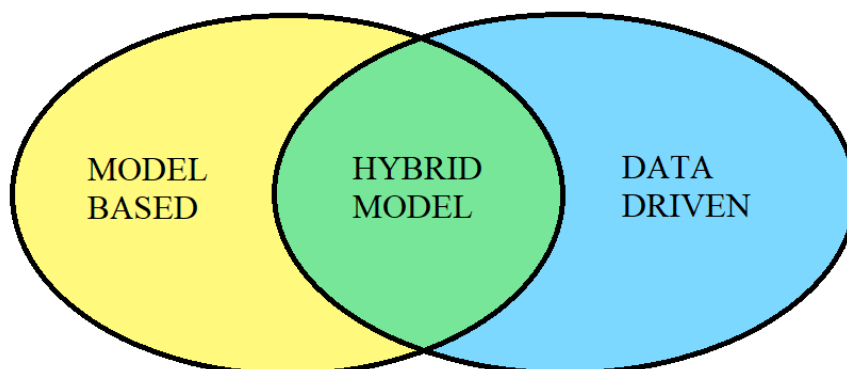


Figure 2.1: Illustration of the connections between model-based methods, data-driven methods and hybrid method.

2.1 Bayesian formulation

In the statistical formulation of inverse problems domain, knowledge is introduced into the problem through the probabilistic models used to describe the unknown quantities. Reconstructions arising from minimizing a variational functional can alternatively be interpreted as Bayes estimators. As reported in [KS06], the measured data is a realization y of the Y -valued random variable

$$Y = F(X) + \delta,$$

where δ is a random variable for the measurement noise and aims to estimate the posterior distribution of X conditioned on $Y = y$, denoted as $\pi_{post}(X = x|Y = y)$. We can reformulate the posterior distribution in terms of the data-likelihood and the prior using the Bayes rule:

$$\pi_{post}(X = x|Y = y) = \frac{\pi_{data}(Y = y|X = x)\pi_X(X = x)}{Z(y)}$$

where $Z(y)$ is a normalizing quantity independent of x . The data likelihood $\pi_{data}(Y = y|X = x)$ gives the probability of measuring certain data given a reconstruction and this is typically well known in terms of the physics of the problem.

For extracting a point-estimate from the posterior, we can compute its mode, leading to the so-called maximum a-posteriori probability (MAP) estimate:

$$\min_{x \in \mathcal{X}} -\log \pi_{data}(Y = y|X = x) - \log \pi_X(X = x). \quad (2.1)$$

If $\pi_{data}(Y = y|X = x)$ and $\pi_X(X = x)$ are convex the problem can be reduced to a convex optimization problem. Convex problems are nice since every local optimum is a global optimum and there is a wealth of algorithms for solving them reasonably fast. The MAP estimation has problem when:

- the statistics of the noise are not known;
- the distribution of the signal is not known or the log-likelihood does not have a closed form;
- the forward operator is not known or only partially known.

One way to overcome ill-posedness is to incorporate regularization in order to stabilize the reconstruction. In Bayesian inversion, regularization is commonly achieved by choosing a prior distribution on images, which assigns a low probability to images with undesired characteristics. Regularization involves using a manually created model that includes prior knowledge and expected properties of the solution.

2.2 Data-driven methods

Within artificial intelligence and machine learning, there are two basic approaches: *supervised* learning and *unsupervised* learning. The main difference is one uses labeled data to help predict outcomes, while the other does not. In supervised learning we have pairs of ground-truth images x with their corresponding measurements y . This match is done by using the forward operator on

clean data. Training a network in a supervised fashion in an inverse problem task means learning a function that, given a sample y of measurement and desired outputs, best approximates the inverse mapping between input and output observable in the data. Unsupervised learning instead works on an unmatched dataset of images x and measurements y so its goal is to infer the natural structure present within a set of data points.

2.2.1 End-to-end Models

The goal of end-to-end models is to estimate the function $f_\theta(\cdot)$ which maps the measurements y directly to reconstructed images x . The universal approximation theorem in [HSW90] states that a fully connected neural network with a large number of neurons in its hidden layer may represent any function we want to learn, as long as our activation functions meet some minor constraints. If the forward operator F is known, a common practice is to use a neural network as post-processing after the reconstruction. To map the measurements back to the image domain you use an approximate inverse of F , denoted as \tilde{F}^{-1} . The selection of \tilde{F}^{-1} is specific to the inverse problem, but common options include using the adjoint of F^\top or the pseudo-inverse F^\dagger . Several inverse problems benefit of residual connection in the reconstruction networks: the residual strategy first applies identity mapping to x , then it performs element-wise addition $f_\theta(x)+x$. Mathematically the overall pipeline is defined as:

$$f_\theta(y) = h_\theta(\tilde{F}^{-1}y) + \tilde{F}^{-1}y. \quad (2.2)$$

For this approach the training is computed in the supervised fashion. Fig.2.2 shows this learning strategy. In [JMFU17] the authors suggest utilizing a U-net to clean up the estimation obtained

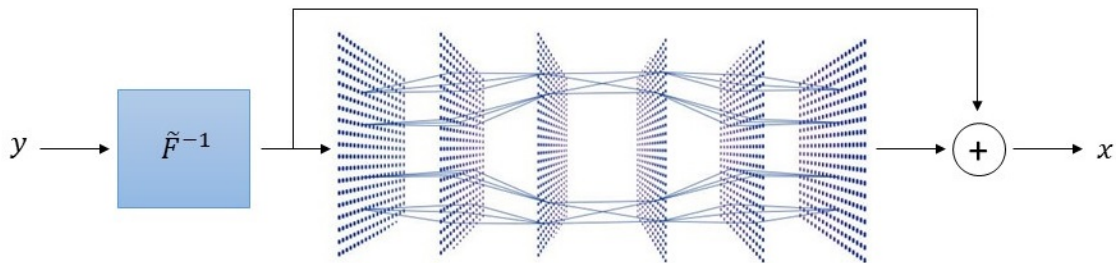


Figure 2.2: Given the reconstruction obtained by applying \tilde{F}^{-1} , the network is used as image post-processing step. Skip connection is implemented.

through filtered back projection in medical imaging issues.

When the forward operator is unknown or the forward model may be completely known but numerically complex to evaluate, the most straightforward approach is to consider the reconstruction map as a "black box" that can be closely approximated by a selected neural network by matching directly the input y and output x . In other words, this task can be viewed as learning an inversion model. An example of this approach is reported in [ZLC⁺18] where the authors present a comprehensive framework for image reconstruction that transforms it into a supervised learning task driven by data, enabling the mapping from sensor data to image domain to be learned from a suitable training dataset. In the super-resolution tasks, the same approach is investigated in [ZLL⁺18].

A different way in the class of end-to-end methods related to unsupervised learning task is the use of generative adversarial networks (GANs) as reported in [GPAM⁺20]. In contrast to the previous approach which model the MAP formulation, GANs work on the data distribution $\pi_X(X = x)$. The training process of GANs enables the model to learn the intricate patterns present in natural data distributions without explicitly defining them. Given the distribution over x and y , GANs consists of two networks: a generator g_{θ_1} and a discriminator d_{θ_2} ; samples are generated by g_{θ_1} by mapping a random vector $z \sim \mathcal{N}(0, 1)$ to the data distribution. Under the unsupervised assumption, considering g_{θ_1} and d_{θ_2} the inverse approximation of each other, realize the joint distribution between x and y modeled by the nets g_{θ_1} and d_{θ_2} . The training minimizes the two loss functions defined as:

$$\begin{cases} \mathcal{L}_d(\theta_2; \theta_1) &= -\mathbb{E}_{x \sim X}[\log d_{\theta_2}(x)] - \mathbb{E}_{y \sim Y}[\log(1 - d_{\theta_2}(g_{\theta_1}(y)))] \\ \mathcal{L}_g(\theta_1; \theta_2) &= -\mathbb{E}_{y \sim Y} \left[\log \frac{d_{\theta_2}(g_{\theta_1}(y))}{1 - d_{\theta_2}(g_{\theta_1}(y))} \right] \end{cases}$$

The purpose of the discriminator in a GANs is to differentiate between real data and fake data generated by the generator, and the generator's goal is to produce data that can trick the discriminator into thinking it is real. After the training, the discriminator is discarded and only the generator is used. The GANs strategy is depicted in Fig.2.3. Given unpaired samples of images

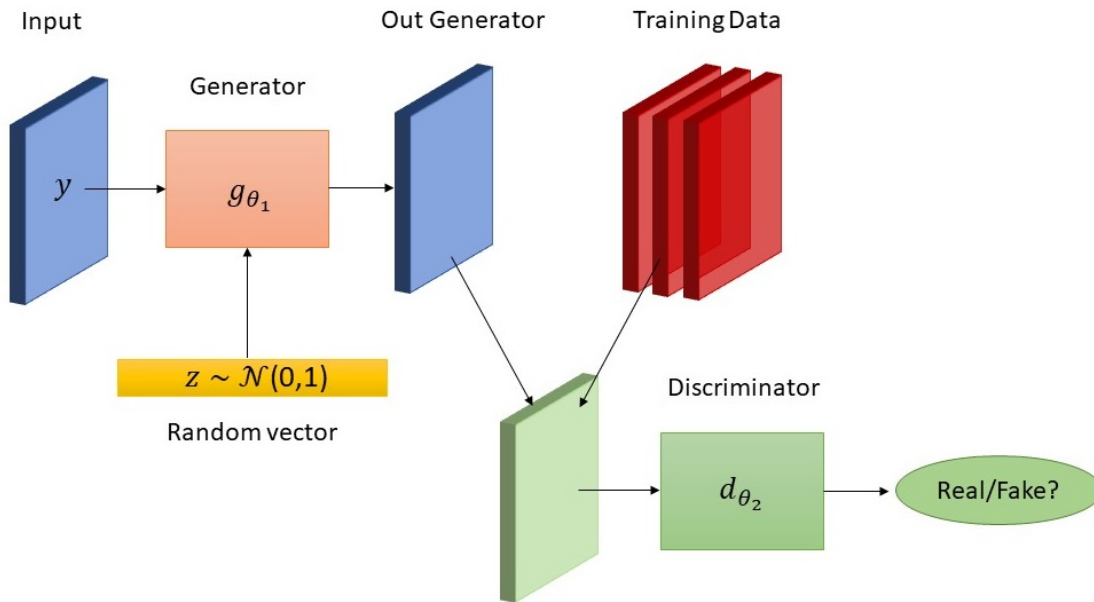


Figure 2.3: IP GANs framework. Given an observed image, the generator produces a prediction for the output image, and the discriminator decides if its input was generated by the generator or came from the training data.

and measurements the approach of cycleGANs (cGANs) is investigated in [ZPIE17] where the nets learn the forward and backward mappings between the image and the measurement domain. To promote the approximation of the data distribution, the cGANs approach for the generator takes into account the random vector z and the observation y . The cGANs used in imaging inverse problems optimize an additional loss besides the traditional adversarial loss. This is due to the fact that cGANs exhibit a tendency to converge towards specific patterns that appear frequently

as artifacts. An examples of cGANs in a super-resolution task is treated in [LLTMK19].

A similar approach to GANs is called AmbientGAN [BPD18] where the adversarial training procedure is investigated. The difference is that in conventional GANs, the discriminator d_{θ_2} is trained to differentiate between the distribution π_X and the generated distribution $\pi_{g_{\theta_1}(z)}$, while in AmbientGAN, the discriminator is trained to distinguish between the distribution π_Y and the ambient generated distribution $\pi_{F(g_{\theta_1}(z))}$. This procedure is applied to learn an implicit generative model given only lossy measurements of samples from the distribution of interest. AmbientGAN learns the distribution π_Y by optimizing the following functional:

$$\min_{g_{\theta_1}} \max_{d_{\theta_2}} \mathbb{E}_y[\log(d_{\theta_2}(y))] - \mathbb{E}_{z,F}[\log(1 - d_{\theta_2}(F(g_{\theta_1}(z))))],$$

where g_{θ_1} and $d_{\theta_2}(y)$ are the generator and discriminator networks respectively and z the random latent variable. After the training, for inference the reconstruction $x^* = G(z^*)$ is computed by solving the least squares problems:

$$z^* := \arg \min_{z \in \mathbb{R}^k} \|F(d_{\theta_2}(z)) - y\|^2. \quad (2.3)$$

Therefore, DL methods have an advantage over traditional methods as they can learn and optimize the entire mapping process, including feature extraction, resulting in an increase in performance. However, this advantage comes at the cost of requiring large amounts of data for training. Despite this requirement, DL methods have been successfully applied to solving inverse problems due to the availability of large simulated datasets generated by applying the forward model (when it is given).

2.3 Hybrid methods

We examined in the previous section the use of end-to-end neural networks to solve image inverse problems. Incorporating the domain knowledge into these class models is harder and they lack robustness when it comes to handling unseen samples of degradation during training. For this reason, the class of hybrid models began to be studied which means combining both deep-learning and model-based methods to solve imaging inverse problems. Exist several different approaches to combine both methods.

2.3.1 Learned Iterative Scheme

The first class of hybrid method is the unfolding or unrolling algorithms. Iterative methods are typically ended based on a stopping criterion that guarantees certain convergence properties. The unrolling algorithm, consists of executing a fixed number of iterations K , or unrolling the optimization algorithm. The layers of the network are used to implement the steps of an iterative algorithm: the network parameters take the place of the respective algorithm's parameters. Fig.2.4 shows an illustration of the unrolling framework.

This approach is supervised and utilizes the forward operator within the learning model to improve data efficiency. The selection of the optimization algorithm to unroll is crucial but not

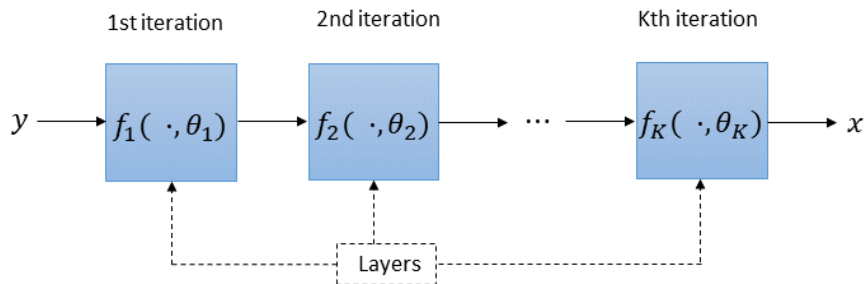


Figure 2.4: Illustration of unrolling algorithm: a deep network can be created by cascading the iterations of a corresponding iterative algorithm $f(\cdot, \theta_i)$ for a fixed number of iterations K . Some components of the algorithm are learned at each step during the training phase.

well understood, in terms of its impact on the performance of unrolled optimization networks. The only requirement is that each iteration of the unrolled algorithm is almost everywhere differentiable. Learned iterative methods were proposed for the first time in [GL10] to develop network approximations for sparse coding task via unfolding the iterative soft-thresholding algorithm. In [AÖ17] the authors derive a partially learned gradient decent scheme for solving an inverse problem in the functional analytic setting. The method results in a gradient-like iterative scheme, where the "gradient" component is learned using a convolutional network that includes the gradients of the data discrepancy and regularizer as input in each iteration. In [AÖ18] is presented a learned primal-dual scheme that also uses the dual space. This unrolled architecture is then trained using data pairs from measurements and ground truth reconstructions. In [LTG⁺20] is proposed an unrolling net which gives us a generalization of the traditional iterative total-variation regularization method in the gradient domain.

Expressing and acquiring knowledge about more parts of the update process increases the potential solutions, but without limitations, it may lead to a loss of certainty in comparison to traditional methods. A different group of methods use nets as data priors and integrates them directly into the optimization process of the proximal analytical methods. In [DSHW17] and [LCM⁺19] the authors propose an unrolled version of the proximal gradient descent scheme. Replace the proximal operator of the regularization used in many convex energy minimization algorithms with a denoising neural network is a strategy used in [MMHC17]. More generally the variable splitting technique is used to separate the original problem into two subproblems: a regularized recovery method (subproblem 1) is used which penalizes using the squared Euclidean distance to an image. This image is then estimated using a suitable denoising technique (subproblem 2), which in these methods is a neural network. Some popular iterative splitting techniques are the alternating direction method of multipliers (ADMM) shown in [SLX⁺16] and the half-quadratic splitting (HQS) investigate in [GY95], or in [ZG18] if based on ISTA.

The general class of reconstruction methods that allow us to replace the proximal operator with a pre-trained neural network is called a Plug-and-Play (PnP) strategy. Originally the PnP is proposed in [VBW13] where the proxy is replaced with an off-the-shelf denoiser. The idea to replace the proxy with a denoising neural network implies that the regularization functional $R(x)$ in (1.4) is not explicitly, as defined [GJN⁺18]. A close-related approach is given in [REM17] where, instead,

the regularization functional is explicitly defined as

$$R(d) := \langle x, x - \Lambda(x) \rangle$$

where $\Lambda : X \rightarrow X$ is a trained denoising net operator.

2.3.2 Learning the Regularizer

In this subsection we will focus on the different approaches to use deep learning to train a regularizer $R(x) : X \rightarrow \mathbb{R}$. A first idea is proposed in [LSAH20] where the regularizer is learned by a CNN which is trained in a supervised setting using an encoder-decoder setup. Here the solved variational formulation is quite different from (1.4) in fact the reconstruction operator is given by:

$$x^* \in \arg \min_{x \in \mathbb{R}^n} \{ \mathcal{J}(x : \lambda) = \mathcal{L}(x; y, F) + \lambda R(\Lambda_\theta(x)) \},$$

where $\Lambda : X \rightarrow \Theta$ is a pre-trained neural network and $R : \Theta \rightarrow [0, \infty]$ is the l_p -norm regularization functional. In order to apply the functional analytic regularization theory of Grasmair et al. (2008), the authors enforce the properties of weakly lower semicontinuous and coercive of the regularization term. One extension of NETT is presented in [ONSH21] where the required coercitivity properties of the network are overcome with an additional term without losing the coercive regularizer property. With the same idea of replacing the typically hand-crafted regularization functional with a neural network in the variational formulation, in [LÖS18] the framework is inspired by how discriminative networks are trained using the GAN architectures. To overcome the problem that the classical learning methods for regularization functionals do not scale to the high dimensional parameter spaces the proposed framework consists to compute a new training algorithm based on the ideas in Wasserstein generative adversarial models, i.e training the network as a critic to tell apart ground truth images from unregularized reconstructions.

The regularizer R is parametrized by the network architecture of Λ_θ for some choice of parameters θ . Mathematical formalization of the regularization functionals is: denote $x \in X$ independent samples from the distribution of ground truth images \mathbb{P}_r , $y \in Y$ independent samples from the distribution of measurements \mathbb{P}_Y . Given \tilde{F}^{-1} the pseudo-inverse, it is used to map the measurement distribution \mathbb{P}_Y to a distribution on image space. Let $\mathbb{P}_n = (\tilde{F}^{-1})_\# \mathbb{P}_Y$ the distribution generated by the pseudo-inverse reconstruction ($\#$ is the push-forward of measures); then the samples of \mathbb{P}_n is corrupted with noise that both depend on the noise model e as well as on the operator F .

The goal of the regularization functional Λ_θ is to tell apart the distribution \mathbb{P}_r and \mathbb{P}_n -taking high values on typical samples of \mathbb{P}_n and low values typical samples of \mathbb{P}_r . The loss function, inspired by the Wasserstein- flavored loss functional, used for learning the regularizer is:

$$\mathbb{E}_{X \sim \mathbb{P}_r} [\Lambda_\theta(X)] - \mathbb{E}_{X \sim \mathbb{P}_n} [\Lambda_\theta(X)] + \lambda \cdot \mathbb{E} [(\|\nabla_x \Lambda_\theta(tX)\| - 1)_+^2].$$

The last term in the loss functional serves to enforce the trained network Λ_θ to be Lipschitz continuous with constant one. Then apply the learned regularization functional to the variational problem and solve it via the (sub) gradient descent method.

An alternative approach that has gained attention in learning the regularizer, is to create a prior

distribution by utilizing recent advancements in deep generative modeling, such as Variational Autoencoders (VAEs), GANs, and normalizing flows. Given a neural network $g_{\theta_1} : \mathbb{R}^k \rightarrow X$, the following generative model $x = g_{\theta_1}(z)$ where $z \sim \mathcal{N}(0, \mathbb{I}_k)$ is the latent random variable, is trained by solving the optimization problem (2.3). In [HPZ22] the authors apply the previous strategy using VAEs networks: $\forall k \in \mathbb{N}$, starting from a random initialization $z_0 \in \mathbb{R}^k$, we define $z^* \sim \mathcal{N}(\sqrt{1 - \delta^2}z_k, \mathbb{I}_k)$ and performing the preconditioned Crank-Nicolson algorithm to define the next step by setting $z_{k+1} = z_k$.

Hence since each layer performs a more complex operation with respect to the end-to-end layers, the number of layers used is fewer. This translates into models with far fewer parameters. Those results in models that have fewer parameters and require less data for training, and also have better generalization capabilities. In addition, each layer performs a known step, as a result, the model can be easily interpreted.

Chapter 3

Electrical Impedance Tomography problem

Electrical Impedance Tomography (EIT) is a nondestructive imaging technique that aims to reconstruct the inner conductivity distribution of a medium starting from a set of measured voltages registered by a series of electrodes that are positioned on the surface of the medium.

This technique was developed in the early 1900s in geological studies but since then it has been adopted in a variety of fields, from medical imaging and industrial process monitoring to structural health monitoring and tissue engineering.

EIT can be divided into two parts: the forward problem, and the inverse problem. The forward problem is the process of solving the potential distribution or boundary voltage distribution with known excitation current and conductivity distribution. The model for the forward problem, known as Complete Electrode Model introduced in [CING89], is based on an elliptic partial differential equation subject to a set of constraints and Neumann boundary conditions that account for the setting with real data, discreteness of the electrodes, and the extra parameters designed by the measuring devices. The traditional methods for solving forward problem are the finite element method [ÖM19], the finite element difference method [AAL⁺07] and the boundary element method [dMFH00]. The inverse problem of EIT is to solve the conductivity distribution in the field by using the voltage value obtained from the forward problem and the boundary voltage measurement. The problem itself is ill-posed.

3.1 EIT forward problem

Given a bounded simply connected C^∞ domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, the corresponding forward EIT problem one wants to find the electric potential u in the interior of the object Ω and at the electrodes $\mathcal{E}_L = \{E_1, \dots, E_L\}$, given some applied current and inner conductivity $\sigma : \Omega \rightarrow \mathbb{R}$. The

forward EIT problem can be formulated as follows:

$$\begin{cases} \nabla \cdot (\sigma(x)\nabla u(x)) = 0 & \text{in } \Omega, \\ u + z_i \sigma \frac{\partial u}{\partial n} = V_i & \text{on } E_i, i = 1, \dots, L, \\ \int_{E_i} \sigma \frac{\partial u}{\partial n} ds = I_i & \text{on } \Gamma, \\ \sigma \frac{\partial u}{\partial n} = 0 & \text{on } \tilde{\Gamma}, \end{cases} \quad (3.1)$$

where $\Gamma = \bigcup_{i=1}^L E_i \subset \Omega$ is the boundary with electrodes and $\tilde{\Gamma} = \partial\Omega \setminus \bigcup_{i=1}^L E_i$ is the boundary without electrodes. V_i is the unknown voltage to be measured by i -th electrode E_i when the currents I_i are applied. z_i are the effective contact impedances for the i -th electrode which justifies that there is an extra resistance between the electrode and the tank. In [SCI92] the authors proved that the CEM model (3.1) for the EIT forward problem has a unique solution and the solution depends continuously on the input current I .

3.1.1 Forward Operator

The *Forward Operator* \tilde{F} , which operates between the Hilbert spaces X and Y , maps the conductivity σ to the solution of the forward problem:

$$\tilde{F} = F_2 \circ F_1: \mathfrak{S} \subset X \rightarrow Y \quad (3.2)$$

$$\sigma \mapsto (u, V) \quad (3.3)$$

where $\mathfrak{S} = \{\sigma \in L^\infty \mid \sigma \nabla u = 0\}$ denotes the domain of definition of F . F_1 maps the conductivity σ to the potential u

$$F_1: X \rightarrow Y \quad (3.4)$$

$$\sigma \mapsto u \quad (3.5)$$

F_2 maps the potential u to the voltage differences V

$$F_2: Y \rightarrow Y$$

$$u \mapsto V.$$

From now on we restrict the conductivities σ to a finite dimensional space of piecewise polynomials. We consider the object domain Ω discretized into n_T subdomains $\{\tau_j\}_{j=1}^{n_T}$ and σ constant over each of them. In particular we discretize Ω with a triangular mesh. Given a Finite Elements Model (FEM) of an EIT medium, we calculate the vector of voltages, V , for each FEM degree of freedom. For a given stimulation pattern a vector of n_M measurements is acquired, obtained by injecting current through an electrodes pair and then measuring the corresponding voltage V induced on another pair of electrodes. Then $F: \mathbb{R}^{n_T} \rightarrow \mathbb{R}^{n_M}$ represents the discrete version of the Forward Operator (3.3) as a nonlinear vector map. Since \tilde{F} is Fréchet differentiable, F' is a matrix, called the Jacobian of F and denoted by J ; each element of $J \in \mathbb{R}^{n_M \times n_T}$ is defined as

$$\{J(u_d, u_m)\}_{i,j} = \int_{\tau_j} \nabla u_d \cdot \nabla u_m d\Omega, \quad (3.6)$$

where the row index i corresponds to the i th measurement, associated with the d th driving potential u_d and m th measurement potential u_m , while the column index j corresponds to the subdomain τ_j .

Considering measured data corrupted by additive noise, we can assume the following noisy non-linear observation model

$$V = F(\sigma) + \eta, \quad (3.7)$$

where $V \in \mathbb{R}^{n_M}$ represents the vector of all the measured electrode potentials whose dimension n_M depends on the choice of a measurement protocol, and $\eta \in \mathbb{R}^{n_M}$ is a zero-mean Gaussian distributed measurement noise vector.

3.2 EIT inverse problem

The nverse problem refers to the process of calculating the information of the object or the system according to the measurement results. The inverse problem is a highly ill-conditioned non-linear problem. Given the measured value V of finite edge voltage and the operator F of the forward problem, we can solve the case which is closest to the real conductivity distribution. The simplest way to fit the data is to consider the so called least squares approach:

$$\sigma^* = \arg \min_{\sigma} f(\sigma), \quad f(\sigma) = \int_{\Omega} (F(\sigma) - V)^2 d\Omega. \quad (3.8)$$

3.3 EIT stimulation pattern

The data acquisition method is directly related to the quantity of information collected and used for reconstruction, it is critical to obtain a good accuracy. The quality of the reconstruction will depend on the quantity and quality of the acquired measurements (injection and measurement strategies, the quality of the measurements, etc). The stimulation pattern is the strategy that chooses which electrodes current injection and voltage measurements are performed in order to define the boundary conditions of (3.1). Different current injection and measurement methods exist in multi-pole measurement. This involves defining certain electrode configurations that will be used for current injection, and other configurations that will be used for measurement. The most popular is called the pair-wise injection strategy, whereby only two injection electrodes are used. The pairs of electrodes in which current is injected are defined as driving or injection pairs and the ones on which the voltages are measured are defined as measurement pairs. When a pair of electrodes is chosen as the driving pair, it is usually excluded from the available measuring pairs. Aimed to provide as much information as possible for the retrieval conductivity distribution, among these methods, the most commonly used are:

- *adjacent injection* when current is injected among two neighbouring electrodes;
- *opposite injection* when current is injected among two opposite electrodes.

The same choices of electrode pairs can be applied in the process of acquisition of voltage measurements, yielding the so-called *adjacent* and *opposite* measurement protocols. In [HCW01] the

authors show that the current density is higher in the center of the domain using opposite injection otherwise adjacent injection favors detection near the electrodes. Fig. 3.1 shows an illustration of adjacent injection - adjacent measurement protocol. In our works the collected measurements are

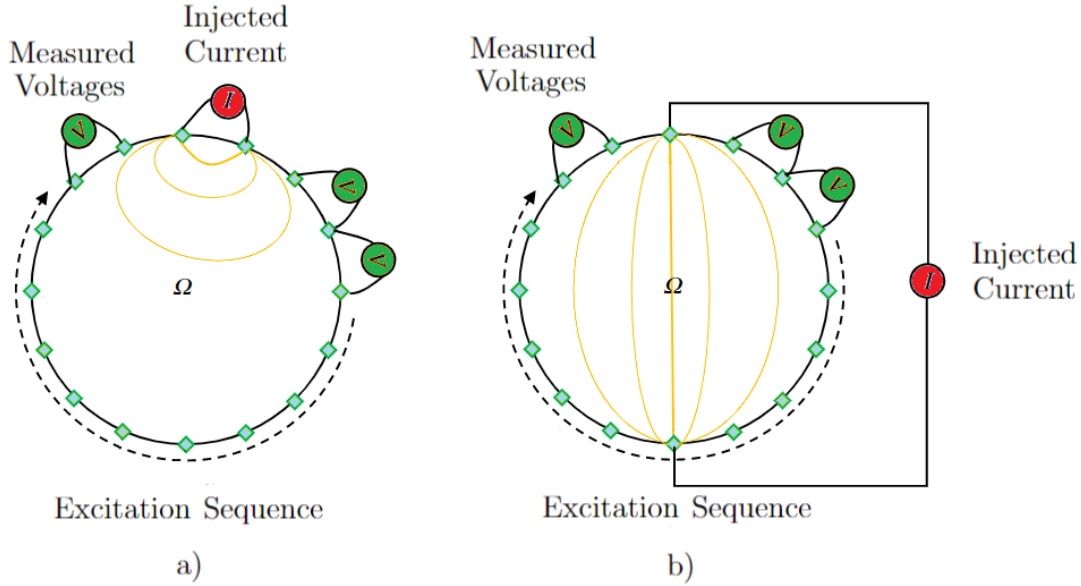


Figure 3.1: Illustration of classic data collection strategies in EIT. (a) Adjacent pair drive. (b) Opposite pair drive.

given using adjacent acquisition protocol: we use adjacent electrodes for the injections and adjacent electrodes for the measurements: given a set of electrodes $\mathcal{E}_L = \{E_0, E_1, \dots, E_L\}$, for each $i = 1, \dots, L$ we drive a current of $I_i = 1$ and $I_{i+1} = -1$ through the i -th pair of electrodes $(E_i, E_{i+1}) \in \mathcal{E}_L$ while all other electrodes are kept insulated. Given this configuration, we consider the electrode index always modulo L , namely the index 0 refers to the L -th electrode and the index $L + 1$ refers to the first electrode. The resulting electric potential $u^{(i)}$ solves (3.5) with:

$$I_l := \delta_{i,l} - \delta_{i+1,l} \quad \text{for } l = 1, \dots, L.$$

Assuming the electric current between the i -th electrode pair, we measure the voltage between the j -th pair of electrodes, i.e.

$$V_{ji}(\sigma) := u^{(i)}|_{E_j} - u^{(i)}|_{E_{j+1}}.$$

The measurement matrix is given by iterating this algorithm for all $j, i = 1, \dots, L$.

Chapter 4

Learning the regularizer for EIT inverse problem

The solution to the inverse EIT problem can be obtained by inverting the forward operator, the non-linear mapping that characterizes the forward problem. Despite the improvements that have followed over the years, the ill-posed nature of the EIT inverse problem still posed many challenges, and meaningful reconstructions could often not be obtained, especially in the most common case of limited and noisy data. Regularization strategies proved to be essential to address this issue. Regularization methods constrain the space of possible reconstructions by embedding prior information about the expected structure of the domain. Among them, Total Variation (TV) promotes sparsity in the reconstructed distribution, thus favor piecewise constant (step changes) conductivities, while, Tikhonov regularization favours smooth solutions. Generalized Tikhonov methods make use of a smoothing penalty term that can include approximations of differential operators [Lio04] or structural priors [VVK⁺98] to further improve reconstruction quality.

In addition to these model-based approaches, deep learning has recently become a new frontier of EIT and in general of inverse problems in imaging. Recent works for solving EIT inverse problems with deep learning focused on fully learned (end-to-end) and post-processing learned approaches. The former utilizes conventional neural networks as 'black-box' by feeding sufficiently large amount of training samples to learn the linear/non-linear mapping from the measured data to the target image without explicitly modeling the domain knowledge, i.e. the forward operator

In this chapter we present a *learning regularizer strategy* which aims at combining the advantages of model-based and data-driven paradigms. The key idea of the model-based data-learning proposal, inspired by many works on inverse imaging, consists in embedding a variational model in an unrolled iterative optimization algorithm which learns all the parameters for the prior model during the training procedure.

We unroll the iterative Gauss-Newton method, well assessed for optimizing a general non-linear smooth model, to solve the inverse EIT non-linear problem. This allows us to incorporate knowledge on the forward model into the neural network as well as directly optimize conductivity reconstruction quality based on training examples.

In addition, we introduce an attention-like strategy that can both sparsify the output of each itera-

tion, driven by the structural anisotropic properties of the input. This feature has been inspired by Total-Variation Sparse Attention (TVMAX), a recently introduced attention mechanism to select visual attention over features generated by a CNN which further encourages the joint selection of adjacent spatial locations [MNMM20]. We named the proposed mechanism Anisotropic Total Variation (ATV). Unlike TVMAX, ATV does not map all the probability mass onto a probability simplex, and does not rely on iterative forward computations. Although non-linear, ATV can be solved with a non-iterative algorithm, and we provide a closed-form expression for its Jacobian. This leads to an efficient implementation of its forward and backward steps. ATV is then incorporated in each block of our unrolled Gauss-Newton EIT network as to transform non-linearly the output of the block and encourage structured and sparse attention over contiguous regions.

4.1 Regularized Gauss-Newton for EITNL

Given the EIT problem explained in chapter 3, we consider the measured data corrupted by additive noise. Assuming the non-linear degradation model (3.7) and the given measurements V , we aimed to solve the non-linear least square problem (3.8). The underlying optimization problem is hard to solve, as the boundary currents depend non-linearly on the conductivity. This means that the optimization problem is nonconvex. We consider a two-dimensional convex domain Ω , where some inclusions with conductivity σ are embedded in a background material with conductivity σ_0 . In difference imaging the unknown conductivity to be reconstructed in the inversion model is the contrast $\sigma - \sigma_0$, i.e., conductivity change when the measurement data before the change is available; here σ_0 is the background conductivity. The proposed approach allows to reconstruct the absolute conductivity distribution when only the data after the change is available (σ_0 is not known).

We can reformulate the Gauss-Newton method 1.4.2 under the EIT description. Given the the second-order Taylor's expansion approximation of the function $f(\sigma)$ as in (1.16), the gradient and the Hessian of $f(\sigma)$ are given respectively by

$$\nabla f(\sigma) = J(\sigma)^T(F(\sigma) - V), \quad \nabla^2 f(\sigma) = J(\sigma)^T J(\sigma) + \sum_k r_k(\sigma) \nabla^2 r_k(\sigma), \quad (4.1)$$

with $J(\sigma)$ the Jacobian matrix of $r(\sigma) := F(\sigma) - V$. Then according to (GN), the search direction p is obtained by solving the following linear system

$$J^T(\sigma)J(\sigma)p^{GN} = -J^T(\sigma)(F(\sigma) - V). \quad (\text{GN-EIT})$$

Gauss-Newton method starts from an initial guess σ_0 and performs a line search along the direction p_k^{GN} to obtain the new conductivity iterate σ_{k+1} as

$$\sigma_{k+1} = \sigma_k + p_k^{GN}. \quad (4.2)$$

In (GN-EIT), V is the measurement vector, and the coefficient matrix, which involves the Jacobian matrix, is a linear operator. However, due to its compact nature, the operator has an unbounded (discontinuous) inverse. This causes the solution to be unstable against variations in the data, hence violating Hadamard's third criterion for well posedness. Therefore, applying the Gauss-Newton method to problem (3.8) yields inaccurate solutions; one could instead employ some form

of regularization on the sought solution σ . This issue can be addressed by applying the regularized Gauss-Newton method by adding an additional regularizer term in the variational formulation as reported in (1.17). Given different type of regularizer as reported in Section 1.4.2, the linear systems (RGN-Tik), (RGN-TV), and (LM) is solved directly using the SVD decomposition as described in (1.26).

4.1.1 Learning the regularization matrix

The a priori choice of the regularization operator $g(\sigma)$ in (RGN-Tik), (RGN-TV), and (LM), and the crucial selection of the regularization parameter λ can be avoided by learning automatically the regularizer from data, and simply setting $\lambda = 1$.

Let $J(\sigma_k) = U\Sigma V^T$ denote the SVD of $J(\sigma_k)$, where $U \in \mathbb{R}^{n_M \times n_M}$, and $V \in \mathbb{R}^{n_T \times n_M}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{n_M \times n_M}$ is the diagonal matrix which contains non-increasing singular values on the main diagonal. We construct the regularizer operator \mathcal{R}_θ as:

$$\mathcal{R}_\theta := V\Sigma_\theta V^T, \quad (4.3)$$

where Σ_θ , the diagonal matrix of singular values, will be learned by the network, as will be described in Section 4.2.2. Considering \mathcal{R}_θ in (4.3), and the decomposition $J^T J = V\Sigma^2 V^T$, then the linear system (RGN) can be rewritten as follows

$$(V\Sigma^2 V^T + V\Sigma_\theta V^T)p_k = J(\sigma_k)^T(V - F(\sigma_k)) + \mathcal{R}_\theta \sigma_k. \quad (4.4)$$

This choice for \mathcal{R}_θ allowed us to exploit the low-dimensional SVD for $J \in \mathbb{R}^{n_M \times n_T}$ instead of a more expensive SVD decomposition for the coefficient matrix in (RGN) of dimension $n_T \times n_T$ ($n_T \gg n_M$). Moreover, the solution p_k of the linear system reduces to the following matrix-vector product:

$$p_k = V(\Sigma^2 + \Sigma_\theta)^{-1} V^T (J(\sigma_k)^T (V - F(\sigma_k)) + \mathcal{R}_\theta \sigma_k). \quad (4.5)$$

4.2 Unrolled Iterative Regularized Gauss-Newton

Our goal is to iteratively solve (4.2)-(4.5) by executing a pre-determined number of iterations K of the regularized Gauss-Newton Algorithm 1, where the regularization operator $g(\sigma)$ is learned across a fully connected network. Moreover, the regularization parameter λ is avoided, so that to provide a parameter-free algorithmic approach to the non-linear EIT problem. The resulting unrolled iterative architecture, named EITGN-NET, is depicted in Figure 4.1, and can be interpreted as a neural network which generates a sequence of approximated solutions σ_k , for $k = 1, \dots, K$, with $\sigma^* = \sigma_K$.

Starting with an initial guess σ_0 , the update mapping at a given iteration $k \in \{0, \dots, K-1\}$ reads as $\sigma_{k+1} \leftarrow \mathcal{G}(\sigma_k; J, V, \Sigma_\theta)$, where

$$\mathcal{G}(\sigma_k; J, V, \Sigma_\theta) := \sigma_k - (J(\sigma_k)^T J(\sigma_k) + \mathcal{R}_\theta)^{-1} r(\sigma_k) = \sigma_k - p_k \quad (4.6)$$

where $r(\sigma_k) = J(\sigma_k)^T (V - F(\sigma_k)) + \mathcal{R}_\theta \sigma_k$, and p_k is computed by solving (4.5).

The regularization operator \mathcal{R}_θ is a prior defined in (4.3) and computed by a Fully Connected

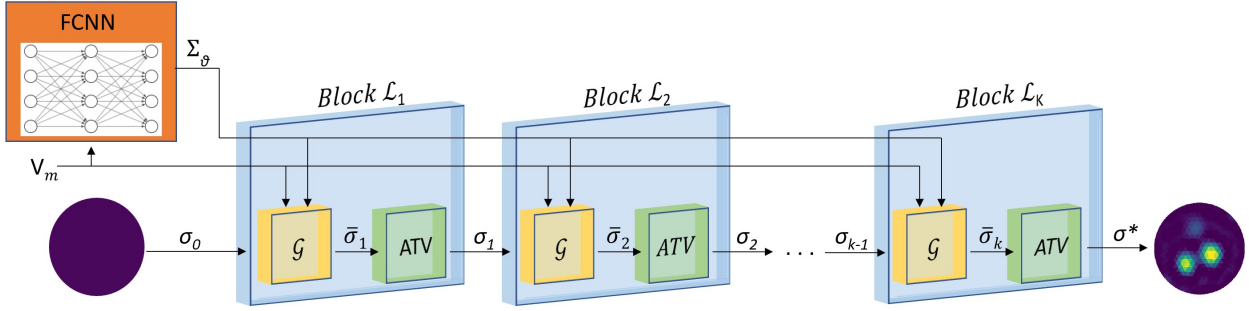


Figure 4.1: EITGN-NET architecture

Neural Network (FCNN). To encompass a large family of priors, we advocate a fully connected-based estimator \mathcal{R}_θ with weights θ , described in Section 4.2.2, that can be learned from historical (training) data.

The k -th iterative block \mathcal{L}_k corresponds to one iteration in algorithm RGN: first updates \mathcal{G} according to (4.6), and then combines the result with the ATV attention-like function to encourage piece-wise structures over contiguous regions on the intermediate reconstructions. In formulas, given an initial σ_0 , for $k = 1, \dots, K$ each block performs the following steps

$$\begin{cases} \bar{\sigma}_k = \mathcal{G}(\sigma_{k-1}; J, V, \Sigma_\theta) \\ \sigma_k = ATV(\bar{\sigma}_k) \end{cases} \quad (4.7)$$

The unrolled data flow is sketched in Algorithm 2. In the experiments reported in Section 4.3 we used $K = 10$, which provided a sufficiently good compromise between efficiency and accuracy in the reconstruction.

Algorithm 2 EITGN-NET

Input: $\sigma^0, V,$
Output: σ^* % conductivity distribution
 $\Sigma_\theta \leftarrow FCNN(V)$
for $k = 0, \dots, K$ **do**
 $\bar{\sigma}_k \leftarrow \mathcal{G}(\sigma_{k-1}; J, V, \Sigma_\theta)$
 $\sigma_k \leftarrow ATV(\bar{\sigma}_k)$
end for
 $\sigma^* = \sigma_K$
end

The loss function used during the network training is defined in Section 4.2.1, while the regularization network to construct \mathcal{R}_θ , and the ATV layer will be described in Section 4.2.2 and Section 4.2.3, respectively.

4.2.1 Loss function design

In the network training process, EITGN-NET is applied on N_s training samples $\{V^{(i)}, \sigma_i^{GT}\}_{i=1}^{N_s}$ to obtain N_s reconstruction results σ_i^* . This allows for tuning the unknown parameters Θ of EITGN-

NET defined in (4.11), by minimizing a loss function. Specifically, the loss function encodes both reconstruction and regularization costs and can be formulated as follows:

$$\begin{aligned} L_{tot}(\Theta) &= L_{rec} + L_{reg} \\ &= \frac{1}{N_s} \sum_{i=1}^{N_s} \|\sigma_i^* - \sigma_i^{GT}\|_2^2 + \frac{1}{N_s} \sum_{i=1}^{N_s} \|\mathcal{R}_\theta \sigma_i^*\|_2^2. \end{aligned}$$

The loss function L_{rec} measures how well the estimated conductivity distributions σ_i^* by the EITGN-NET matches the ground-truth ones σ_i^{GT} . The second term L_{reg} penalizes the network when the regularization contribution is too large, which could deviate from the original linear system to be solved. Given the training samples, the network EITGN-NET learns a parameter set Θ minimizing the loss function $L_{tot}(\Theta)$ by applying a gradient descent algorithm. Then the derivatives $\frac{\partial L_{tot}}{\partial \Theta^k}$ can be calculated by back-propagation technique using chain-rule

$$\frac{\partial L_{tot}}{\partial \Theta^k} = \frac{\partial L_{tot}}{\partial \bar{\sigma}_k} \cdot \frac{\partial \mathcal{G}_{\Theta^k}}{\partial \Theta^k} = \frac{\partial L_{tot}}{\partial \bar{\sigma}_k} \cdot \frac{\partial \bar{\sigma}_k}{\partial \Theta^k}. \quad (4.8)$$

Considering the functional operator on a generic block k given by (4.7), we have

$$\frac{\partial L_{tot}}{\partial \bar{\sigma}_k} = \sum_{k=1}^K \frac{\partial L_{tot}}{\partial ATV_k} \cdot \frac{\partial ATV_k}{\partial \bar{\sigma}_k}, \quad (4.9)$$

where $\frac{\partial L_{tot}}{\partial ATV_k}$ is in the backward propagation stage, and $\frac{\partial ATV_k}{\partial \bar{\sigma}_k}$ will be defined in (4.17).

4.2.2 Regularizer network

The regularization matrix \mathcal{R}_θ is constructed starting from its singular values Σ_θ following (4.3), by learning Σ_θ across a fully connected regression network $FCNN(V)$ consisting of four hidden layers each composed of n_M neurons. The input layer is initialized with the n_M measurement values V , and the output layer has n_M nodes that represent the sorted singular value vector Σ_θ .

In a fully connected network each neuron of a given layer has a connection (or synapse) towards each neuron of the next layer. For a given artificial neural node i , let there be m inputs, x_1 through x_m , and weights $\theta_{i,1}$ through $\theta_{i,m}$. The output of the i th neuron is:

$$y_i = \phi_1\left(\sum_{j=1}^m \theta_{i,j} x_j + b_i\right)$$

where ϕ_1 is the LeakyReLU activation function, defined as $LeakyReLU(z) = \max(\alpha_1 z, z)$, with a negative region slope of $\alpha_1 = 0.5$. For the output layer the function ϕ_1 is replaced by ϕ_2 which is the Rectified Linear Unit (ReLU) activation function

$$ReLU(z) = \max(\alpha_2, z + \alpha_2), \quad (4.10)$$

with translation factor $\alpha_2 = 1 \times 10^{-9}$. The output vector of each layer ℓ can be represented as

$$y^{[\ell]} = \phi_1(W^{[\ell]}x + b^{[\ell]}),$$

where the matrix $W^{[\ell]} \in \mathbb{R}^{n_M \times n_M}$ and the vector $b^{[\ell]} \in \mathbb{R}^{n_M}$, are the weights and the bias of the ℓ th layer. Hence the FCNN output is a vector

$$\begin{aligned} \Sigma_\theta(\theta; V) = & \phi_2(\phi_1(W^{[5]}(\phi_1(W^{[4]}\phi_1(W^{[3]}\phi_1(W^{[2]}\phi_1(W^{[1]}V + b^{[1]} \\ & + b^{[2]} + b^{[3]} + b^{[4]} + b^{[5]}), \end{aligned}$$

where the set of weights and bias is denoted by

$$\Theta = \{W^{[1]}, W^{[2]}, W^{[3]}, W^{[4]}, W^{[5]}, b^{[1]}, b^{[2]}, b^{[3]}, b^{[4]}, b^{[5]}\}. \quad (4.11)$$

The choice of ϕ_2 as defined in (4.10) guarantees the non-singularity of Σ_θ and of the associated coefficient matrix in (4.4).

4.2.3 2D-Mesh anisotropic TV attention-like mechanism

The proposed attention-like mechanism, named ATV, acts inbetween the blocks of the EITGN-NET. It is derived by extending the total variation image denoising algorithm introduced in [LO09] to an arbitrary 2D triangular mesh domain.

The ATV mechanism operates on a 2D domain represented by a triangular mesh. Before discussing the proposed ATV operator, we introduce some notations and we formulate the total variation operator on a 2D mesh. Let us assume a planar domain $\Omega \subset \mathbb{R}^2$ which is approximated by a triangulated mesh (V_{tr}, T, E) , where $V_{tr} \in \mathbb{R}^{n_V \times 2}$, $V_{tr} = \{v_i\}_{i=1}^{n_V}$ represents the set of vertices, $T \in \mathbb{R}^{n_T \times 3}$, $T = \{\tau_m\}_{m=1}^{n_T}$ is the set of triangles and $E \in \mathbb{R}^{n_E \times 2}$, $E = \{e_j\}_{j=1}^{n_E}$ is the set of edges.

We denote by $\mathcal{N}(\tau_i) = \{\tau_j \in T : \tau_j \cap \tau_i \in V_{tr}\}$ the set of the triangles $\tau_j \in T$ which share one of their vertices or edges with the triangle τ_i . A basic $\mathcal{N}(\tau_i)$ could simply consider the edge intersection, thus reducing the cardinality of $\mathcal{N}(\tau_i)$ to 3 for inner triangles and 2 for boundary triangles.

We assume a piece-wise constant strictly bounded measurable function $\sigma : \Omega \rightarrow \mathbb{R}$ over the mesh triangles, thus the gradient operator vanishes to zero everywhere but the mesh edges along which it is constant. In particular, the discrete anisotropic gradient magnitude $\|(\nabla\sigma)_i\|_1$ on a triangle τ_i is defined as a weighted sum over the neighborhood triangles in $\mathcal{N}(\tau_i)$ as follows

$$\|(\nabla\sigma)_i\|_1 := \sum_{k \in \mathcal{N}(\tau_i)} w_k |\sigma_i - \sigma_k|, \quad i = 1, \dots, n_T, \quad (4.12)$$

where each weight w_k is associated to the triangle neighbor $\tau_k \in \mathcal{N}(\tau_i)$, and is defined as:

$$w_k = \frac{1}{\|barycentre(\tau_k) - barycentre(\tau_i)\|_2}. \quad (4.13)$$

We notice that (4.12) approximates the local variation of σ as sum of the weighted variations in each neighbors direction, thus discretizing an anisotropic total variation operator.

For a given triangle τ_i , we assume to know the values $\bar{\sigma}_k$ for all the neighbors in $\mathcal{N}(\tau_i)$, then the unknown value σ at τ_i is obtained as solution of the following local minimization problem:

$$\sigma^* = ATV(\sigma) := \arg \min_{\sigma \in \mathbb{R}} \mathcal{F}(\sigma), \quad (4.14)$$

$$\mathcal{F}(\sigma) := \sum_{k \in \mathcal{N}(\tau_i)} w_k |\sigma - \bar{\sigma}_k| + \mu |\sigma - \bar{\sigma}_i|^2, \quad i = 1, \dots, n_T.$$

The proposed ATV carries out the triangle-wise anisotropic TV, by solving the optimization problem (4.14) for any triangle τ_i . In Proposition 4.1, following [LO09], we report the explicit solution of the local optimization problem (4.14).

Proposition 4.1. *Let τ be a generic triangle with n the cardinality of $\mathcal{N}(\tau)$. Assuming $w_k > 0$ be the weight defined in (4.13) and associated to $\tau_k \in \mathcal{N}(\tau)$, the values $\bar{\sigma}_k$ on $\tau_k \in \mathcal{N}(\tau)$ be sorted as $\bar{\sigma}_1 \leq \bar{\sigma}_2 \leq \dots \leq \bar{\sigma}_n$. Then the minimizer of problem (4.14) is given by $\sigma^* \in \mathbb{R}$ which is the unique minimizer of the anisotropic total variation problem*

$$\sigma^* = \arg \min_{\sigma \in \mathbb{R}} \mathcal{F}(\sigma) = \text{median}\{\bar{\sigma}_1, \dots, \bar{\sigma}_n, \bar{\sigma}_i + \frac{1}{2\mu}W_0, \bar{\sigma}_i + \frac{1}{2\mu}W_1, \dots, \bar{\sigma}_i + \frac{1}{2\mu}W_n\}, \quad (4.15)$$

where

$$W_j = - \sum_{k=1}^j w_k + \sum_{k=j+1}^n w_k, \quad j = 0, \dots, n. \quad (4.16)$$

The ATV optimization problem (4.14) is solved by applying the closed formula (4.15) on the 2D-mesh, as outlined in Algorithm 3. The median value computation in the algorithm could represent a computational bottleneck. However, this computation becomes cheap by using the smart strategy proposed in [LO09].

Algorithm 3 ATV

Input: $\bar{\sigma} \in \mathbb{R}^{n_T}$, $\mu > 0$,

Output: $\sigma^* \in \mathbb{R}^{n_T}$

Initialize $\sigma = \bar{\sigma}$

for each $\tau_i \in T$ **do**

 compute $W \in \mathbb{R}^{n+1}$ as in (4.16)

 set $u = (\sigma_0, \dots, \sigma_{i-1}, \sigma_{i+1}, \dots, \sigma_n) \in \mathbb{R}^n$

$u \leftarrow \text{sort}(u)$ in ascending order

 compute $p \in \mathbb{R}^{n+1}$, with $p_k = \sigma_i + \frac{1}{\mu}W_k$, $k = 0, \dots, n$

$\sigma_i^* \leftarrow \text{median}(u_1, u_2, \dots, u_n, p_0, p_1, p_2, \dots, p_n)$

end for

end

Note that Algorithm 3 realizes a local anisotropic total variation. The result in Proposition 4.2 guarantees instead that, by iterating Algorithm 3, the convergence to the global anisotropic total variation is guaranteed. The attention module here proposed applies local ATV using only Algorithm 3, and thus does not need any iterative procedure to converge.

Proposition 4.2. *The algorithm ATV repeated until convergence, defines the vector $\sigma^* = (\sigma_1^*, \dots, \sigma_{n_T}^*) \in \mathbb{R}^{n_T}$ which is the global minimizer of the anisotropic total variation problem*

$$\sigma^* = \arg \min_{\sigma \in \mathbb{R}^{n_T}} \|\nabla \sigma\|_1 + \mu \|\sigma - \bar{\sigma}\|_2^2.$$

Proof. By iterating (4.14) j times for the ℓ th component $\sigma_\ell^{(j)}$, we get $\sigma_\ell^{(j+1)} = \arg \min_{\sigma_\ell \in \mathbb{R}} \mathcal{F}^{(j)}(\sigma_\ell)$. Hence

$$\mathcal{F}^{(j)}(\sigma_\ell^{(j+1)}) \leq \mathcal{F}^{(j)}(\sigma_\ell^{(j)}).$$

This implies that the energy over all triangles decreases, that is

$$\mathcal{F}(\sigma^{(j+1)}) \leq \mathcal{F}(\sigma^{(j)}),$$

and since it is also bounded from below, the sequence $\mathcal{F}(\sigma^{(j)})$ converges.

From Lemma 3.1 in [LO09], we have

$$|\sigma^{(j)} - \sigma^{(j+1)}| \leq \sqrt{\frac{\mathcal{F}(\sigma^{(j)}) - \mathcal{F}(\sigma^{(j+1)})}{\mu}}.$$

Therefore $\sigma^{(j)}$ converges to the vector $\sigma^* \in \mathbb{R}^{n_T}$. \square

In order to apply the $ATV(\sigma)$ attention-like mechanism in a neural network trained by back-propagation, two problems must be addressed. The first is the forward computation: how to evaluate $ATV(\sigma)$, i.e., how to solve the optimization problem in (4.14). The second is the backward computation: how to evaluate the Jacobian of $ATV(\sigma)$. The forward computation is solved by the closed-form (4.15). For the backward computation the main challenge is how to compute the derivatives of $ATV(\sigma)$. Note that ATV is smooth everywhere except on the (zero-measure) set of non-differentiable points. In particular, on a triangle τ_i , the partial derivative of ATV are defined as

$$\frac{\partial ATV}{\partial \bar{\sigma}_i} = \sum_{k \in \mathcal{N}(\tau_i)} w_k \text{sign}(\sigma - \bar{\sigma}_k) + 2\mu(\sigma - \bar{\sigma}_i), \quad (4.17)$$

which simplifies as follows

$$\begin{aligned} \frac{\partial ATV}{\partial \bar{\sigma}_i} &= \\ &= \begin{cases} 2k - n + 2\mu(\sigma^* - \bar{\sigma}_k) & \text{if } \bar{\sigma}_k < \sigma^* < \bar{\sigma}_{k+1} \\ \ell + (m - \ell)\epsilon - (n - m) + 2\mu(\sigma^* - \bar{\sigma}_i) & \text{if } \bar{\sigma}_\ell < \bar{\sigma}_{\ell+1} = \dots = \bar{\sigma}_m = \sigma^* < \bar{\sigma}_{m+1} \end{cases}. \end{aligned} \quad (4.18)$$

The bottom condition in (4.18) corresponds to the case σ^* coincides with one of the $\bar{\sigma}_k$. In this case ϵ satisfies

$$\epsilon = \begin{cases} 1 & \text{if } i \leq \ell, \\ p \in [-1, 1] & \text{if } \ell + 1 \leq i \leq m, \\ -1 & \text{if } i > m. \end{cases}$$

The backpropagation task in (4.8) is then completed by using (4.18).

4.3 Simulations and Numerical Experiments

To evaluate the performance of the proposed network, a series of numerical and experimental tests was conducted on a set of synthetic 2D experiments. All examples simulate a circular tank slice of unitary radius represented by a mesh grid of 660 triangles. In the circular boundary ring 16

electrodes are equally spaced located. The conductivity of the background liquid is set to be $\sigma_0 = 1.0 \Omega m^{-1}$. Measurements are simulated through *opposite injection - adjacent measurement* protocol via pyEIT [LYX⁺18] a python based framework for Electrical Impedance Tomography. In all the examples the setup is considered blind, that is no a priori information about the sizes or locations of the inclusions is considered.

In the examples illustrated in Section 4.3.4-4.3.6 we compare the performance of the proposed EITGN-NET with the iterative Regularized Gauss Newton method (RGN), and the Levenberg Marquardt (LM) implemented using pyEIT library, and the D-bar method [MS12], with code kindly provided by the authors. The D-bar output is provided and visualized in image form, suitably handled for comparisons. The comparisons are conducted qualitatively by visually inspecting the artifacts and quantitatively by calculating the metrics described in Section 4.3.2 such as mean-square error (MSE), structural similarity (SSIM), dynamic range (DR), and the evaluation index for EIT images (EIEI). In all the experiments with RGN and LM algorithms, we hand-tuned the regularization parameter λ so as to fairly achieve the best performance in terms of MSE values.

The learning rate is set to 10^{-3} . The learning is performed for 20 epochs. The network is implemented on a PC with Intel i7 CPU and 32-GB RAM with Pytorch and AdamW has been used for optimization. The training process took nearly 30min each epoch of the training.

4.3.1 Training and Testing Data Sets

In the training process, we employed 250 randomly generated test cases, where 200 of them were used as training dataset and the other 50 test cases were used as testing dataset. Experimentally, we found that 20 epochs were enough to sufficiently decrease the total loss function. The collected weights Θ at epoch 20 have been used in EITGN-NET in the testing phase.

A training dataset contains a total number of 200 pairs of the ground-truth conductivity σ^{GT} and their corresponding collected voltages, V . Each test case consists of a random number from 1 to 4 of anomalies inside a circular tank, localized randomly and characterized by random radius in the range $[0.15 - 0.25]$ and magnitude in the range $[0.2, 2]$. Here it is assumed that each actual anomaly approximately consists of the same material, while the background consists of another homogeneous material. Consequently, a preferable EIT reconstruction meets the following conditions: C1) Anomalies of homogeneous material must have the same intensity. C2) Anomalies and backgrounds have higher and lower intensities than those of artifacts, respectively. C3) The number of triangles partitioned into the cluster of artifacts should be as small as possible.

4.3.2 Figures of merit

The performance is assessed both qualitatively and quantitatively. The quantitative analysis is performed on the mesh elements, via the Mean Squared Error (MSE) defined as

$$MSE_{\sigma} = \frac{\|\sigma^{GT} - \sigma^*\|_2^2}{n_T},$$

which measures how well the original conductivity distribution is reconstructed in case a ground truth conductivity distribution σ^{GT} is known.

Another metric commonly used for measuring the similarity between two images is the Structural SIMilarity (SSIM) Index, which has been suitably modified to act directly on a mesh. In particular, local $SSIM_{\tau_i}$ measures are computed on local neighborhoods $\mathcal{N}(\tau_i)$, which move triangle by triangle on the entire mesh Ω . In other words, for each triangle τ_i , the $SSIM_{\tau_i}$ is calculated on its neighborhood. The global structural similarity measure SSIM between a reconstructed conductivity σ^* and the corresponding ground truth σ^{GT} is defined as follows

$$SSIM(\sigma^{GT}, \sigma^*) := \frac{1}{n_T} \sum_{i=1}^{n_T} SSIM_{\tau_i}(\sigma_i^{GT}, \sigma_i^*), \quad (4.19)$$

where for each triangle i th,

$$SSIM_{\tau_i}(\sigma_i^{GT}, \sigma_i^*) = \frac{(2\mu_i^{GT} \mu_i^* + c_1)(2s_i^{GT*} + c_2)}{((\mu_i^{GT})^2 + (\mu_i^*)^2 + c_1)((s_i^{GT})^2 + (s_i^*)^2)},$$

with μ_i denotes the mean of the σ values in the neighbors $\mathcal{N}(\tau_i)$ of the triangle τ_i , s_i represents its associated standard deviation

$$s_i^{GT*} := \frac{1}{|\mathcal{N}(\tau_i)|} \sum_{k \in \mathcal{N}(\tau_i)} (\sigma_k^{GT} - \mu_k^{GT})(\sigma_k^* - \mu_k^*)$$

and finally $c_1 = \max(\sigma^{GT}) * 0.01^2$ and $c_2 = \max(\sigma^{GT}) * 0.03^2$.

Analogously to the use in image processing, $SSIM(X, Y)$ in (4.19) quantifies how much X and Y are different, with $SSIM = 1$ if X and Y are identical and $SSIM$ tends toward 0 when X and Y are very different.

According to conditions C1)-C2)-C3) the quality of any EIT reconstruction will increase as the number of artifact values decreases. In [WYS⁺18] an ad hoc evaluation index for EIT (Evaluation Index Electrical Impedance - EIEI-) has been introduced, which measures the homogeneity of the anomalies. Let n_1 be the number of triangles of backgrounds, with associated values denoted by $\sigma^{(1)}$, n_2 the number of triangles classified as artifacts, with associated values $\sigma^{(2)}$, and n_3 the number of triangles containing anomalies, with associated values $\sigma^{(3)}$. Unlike in [WYS⁺18], in our work the anomalies can have different values, thus we adapted the clusterization phase, accordingly. The variances of these clusters are given by

$$\delta_j = \sum_{i=1}^{n_j} |\sigma_i^{(j)} - \bar{\sigma}^{(j)}| / n_j, \quad j = 1, 2, 3.$$

The values of variances reflect the homogeneity of backgrounds, anomalies and artifacts. Then the EIEI measure is defined as follows

$$EIEI := w_1 T_1 + w_2 T_2 \quad (4.20)$$

with $T_1 := 1 - n_2/n_T$ the quantity of artifact triangles, $T_2 := 1 - (\delta_1 n_1/n_T + \delta_3 n_3/n_T)$, the quantity of background and anomalies, the weighting values w_1 and w_2 represent the certainty of T_1 and T_2 , which is evaluated as

$$w_1 := \sum_{i=1}^{n_2} \sigma_i^{(2)} / n_2, \quad w_2 := \left(\sum_{i=1}^{n_1} \sigma_i^{(1)} + \sum_{i=1}^{n_3} \sigma_i^{(3)} \right) / (n_1 + n_3).$$

Together with the quantitative EIEI measure defined in (4.20), this figure of merit allows for a qualitative insight provided by a structure maps (EIEI map) where yellow color identifies the artefacts, red color the anomalies and blue color the background. Finally, another significant metric to assess the quality of the reconstructed conductivity is the dynamic range defined as

$$DR = \frac{\max \sigma^* - \min \sigma^*}{\max \sigma^{GT} - \min \sigma^{GT}} \times 100\%.$$

A DR value which widely differs from 100, indicates that the conductivity contrast was not well preserved in the numerical optimization process.

4.3.3 Effect of the number of blocks and benefit of the ATV mechanism

To evaluate the effect of the number of blocks K used by the proposed EITGN-NET, we evaluated the performance for varying K values. In particular, we report in 4.1 the quantitative measures MSE and SSIM, averaging the results on the testing set, with $K = \{8, 10, 12\}$. We noticed that the performance is gradually improved for increasing K values and tends to remain stable when K is larger than 10. Therefore the choice $K = 10$ configuration is a preferable setting which represents a good compromise between the result quality and the computational cost. The conductivity recon-

Blocks (K)	8	10	12
MSE	0.0039	0.0037	0.0038
SSIM	0.87	0.89	0.89

Table 4.1: Effect of a different numbers of blocks K in the unrolled iterative EITGN-NET.

struction quality obtained by EITGN-NET is affected by the new ATV mechanism introduced. To highlight how much this mechanism influences the results, in Fig.4.2 we illustrate the reconstructions produced by the EITGN-NET with ATV (top row) and without activating the ATV stage (bottom row). From a visual inspection we can observe sharper and more artifact-free results when ATV is applied.

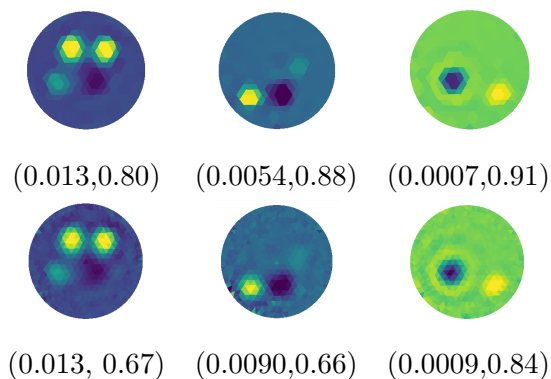


Figure 4.2: First row: EITGN-NET with ATV. Second row: EITGN-NET without ATV. (MSE, SSIM) values are reported for each test cases GT 225, GT 226, GT 232.

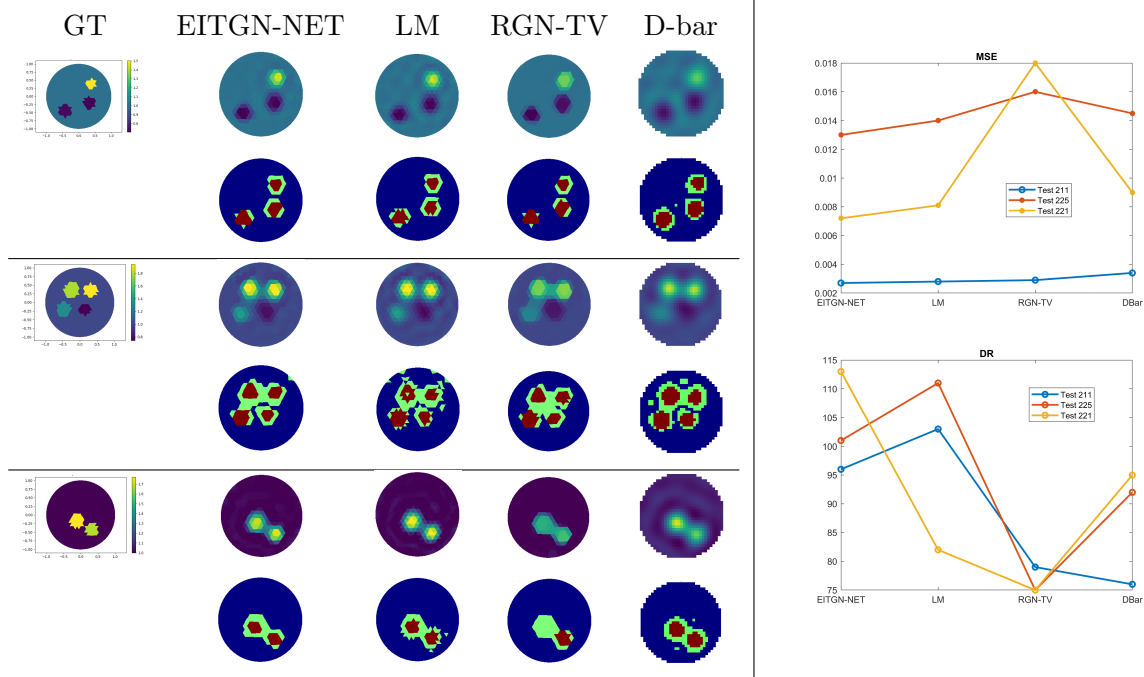


Figure 4.3: Performance in terms of MSE, DR, EIEI metrics of the reconstructions by EITGN-NET, LM, RGN-TV, and D-bar for test case GT 211, GT 225 and GT 221 (from top to bottom).

4.3.4 Example 1 - Performance comparisons

In Fig.4.3 we illustrate the performance of EITGN-NET with respect to RGN-TV, LM and D-bar when applied to the reconstruction of three test cases. In Fig.4.3 (right panel) we plot the graphs corresponding to MSE and DR, while the maps on the left panel visualize the GT, EITGN-NET, RGN-TV, LM and D-bar reconstructions with their associated EIEI maps (bottom row). We recall that, for MSE the lower, the better, while for DR the values closest to 100 are preferred. A qualitative inspection in Fig. 4.3 (left panel) highlights the superior quality of EITGN-NET which better recovers sharper structures, presents a lower number of artifacts, and in general improves the separation between the anomalies. By a visual inspection of Fig.5.3, we can compare σ^{GT} with the reconstructed results obtained by EITGN-NET (σ^* , first column), RGN-TV (second column) and D-bar (third column). The associated EIEI structure maps are illustrated in the bottom rows, yellow for artefacts, red for anomalies and blue for background; the EIEI values are reported below each structure map, for each test case. The test case GT 237 contains 4 anomalies, two of which present intensity values very close to the background value. Nevertheless all the methods are able to detect them properly. The well known contrast reduction effect of the TV regularizer is slightly visible everywhere. In particular, in the test case GT 208, the intensity of the anomalies is significantly reduced by the other methods, while accurately recovered by our network. In general, the EITGN-NET is able to automatically detect all the anomalies with the minimum artifacts, as confirmed by the EIEI values reported below each structure map, while preserving faithfully the anomalies shape and piecewise-constant amplitudes. Performance on averaged MSE, SSIM, and DR metrics on the entire set of test cases are reported in Table 4.2. The proposed EITGN-NET has been trained on datasets with circular anomalies. To verify the robustness to other shaped anomalies, we tested different shaped

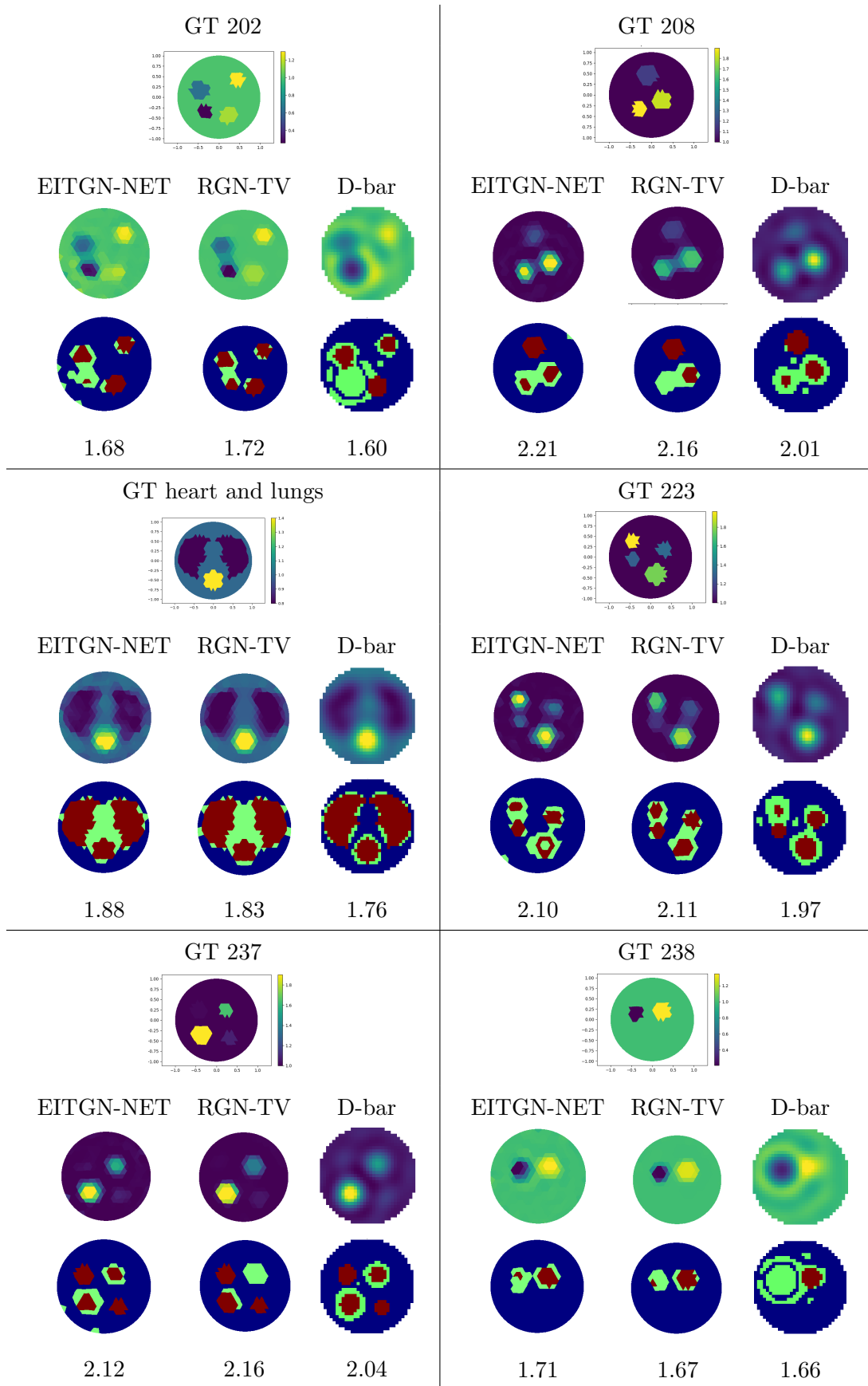


Figure 4.4: Comparisons between σ^{GT} (top rows) and conductivity results reconstructed by EITGN-NET (first column), RGN-TV (second column) and D-bar (third column) with the associated structure maps (EIEI metric) on the bottom. EIEI values are reported for each test case.

conductivities. By the way of illustration in Fig.5.3 (case labelled by *heart and lungs*) we report the results of a phantom image representing heart and lungs anomalies which are effectively detected by the proposed network. As expected, the RGN-TV behaves slightly better in this case since the network was trained with different shaped anomalies. This confirms the well-know drawback of the learning-based approaches when the testing dataset is different from what they have been trained with. We trust that, with a suitable training, EITGN-NET will provide more accurate reconstructions even in case of similar anomalies.

	EITGN-NET	LM	DBar	RGN-TV
MSE	0.0037	0.0072	0.0060	0.0038
SSIM	0.89	0.77	0.82	0.93
DR	110	119	107	89

Table 4.2: Performance on averaged MSE, SSIM, DR metrics on the entire set of test cases.

4.3.5 Example 2 - robustness to the noise

The measured voltages are easily contaminated in practical EIT measurements. In order to evaluate the robustness of the proposed method to noisy measurements we corrupted the voltage measurements generated from the solution of the forward model by additive white Gaussian noise. In particular, we generated the noisy measurement vector V by adding a vector $\bar{n} \sim \mathcal{N}(0, s^2)$ of Gaussian noise characterized by zero-mean and standard deviation $\eta\bar{V}$, with \bar{V} the average value. The corresponding degradation model is the following:

$$V = V + \eta \bar{V} \text{rand}(n_M).$$

We trained the network using 50% samples corresponding to corrupted V and 50% samples corresponding to noise-free V measurements. During the training phase we used $\eta = 5 \times 10^{-3}$ which corresponds to a quality $\text{SNR} = 48\text{dB}$, while we validated the network with two different noise levels $\eta = 2.5 \times 10^{-3}$ ($\text{SNR}=54\text{dB}$) and the more severe $\eta = 5 \times 10^{-3}$ ($\text{SNR}=48\text{dB}$).

The conductivities obtained by the EITGN-NET with noisy measurements ($\text{SNR}=48\text{dB}$) are illustrated in Fig.5.5 for the test cases reported in Fig.4.3. The results in Fig.5.5 show the robustness of the proposed neural network to noise levels up to $\text{SNR}=48\text{dB}$. The results with degradations less severe than $\text{SNR}=48\text{dB}$ are not reported. From a visual comparison of the associated EIEI results we can observe how the noise affects the quality of the reconstructions, slightly corrupting the identification of the different anomalies. We recall that for EIEI metric, the higher, the better. Hence, more severe noise levels demand for new training of the proposed EITGN-NET with a tuning of the α_2 parameter in (4.10).

To summarize the performance, in Table 5.2 we report the averaged values on 50 test samples, both for the noise-free test cases, using the network trained without noisy samples, and for the two different degraded input, corresponding to a SNR of 48dB and 54dB. We do not report the results from noisy measures for RGN-TV and LM since for each test case we should have tuned manually

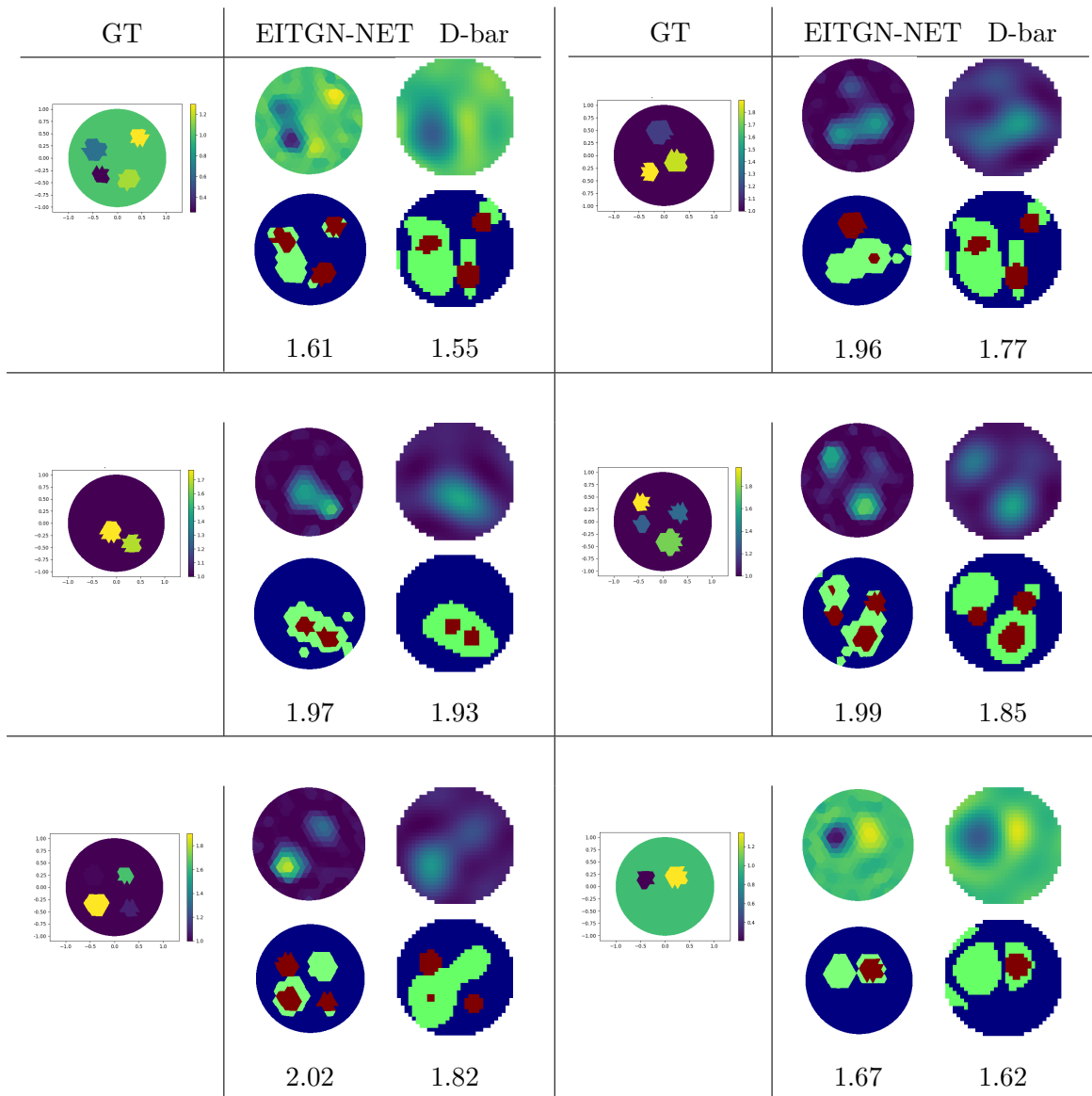


Figure 4.5: Robustness to the noise: comparison between σ^{GT} and reconstructions from noisy measurements obtained by EITGN-NET (first column) and D-bar (second column): the reconstructed conductivities σ^* (first row) and associated structure maps for EIEI metric (second row). EIEI values are reported for each test case.

	EITGN-NET	D-bar
noise 54dB		
MSE	0.0078	0.0107
SSIM	0.75	0.75
DR	102	88
noise 48dB		
MSE	0.0085	0.0111
SSIM	0.67	0.62
DR	110	84

Table 4.3: Noisy measurements: performance on averaged MSE, SSIM, and DR metrics on the entire set of test cases.

the regularization parameter. The reported metrics confirm the overall better performance of the proposed network.

4.3.6 Example 3 - Regularizer efficacy

We illustrate the effectiveness of the learned regularizer by checking the behaviours of different popular regularizers with respect to the proposed \mathcal{R}_θ obtained by the network described in Section 4.2.2. Specifically, we denote by $R_1 = L^T L$ the Laplace regularizer, by $R_2 = \sqrt{\text{diag}(J^T J)}$ the popular regularizer proposed in NOSER [CIN⁺90], and by R_3 the TV regularizer defined in (1.8). The averaged condition number of the linear systems (4.4) during the testing phase is reduced through the network blocks from 10^{21} to 10^7 , thus confirming the benefit of the network regularizer \mathcal{R}_θ . By the way of illustration we report in Figs.4.6, 4.7 four test cases of conductivity reconstructions produced by the Gauss Newton method using as regularizer R_1 , R_2 and R_3 for different regularization parameters λ , together with the EITGN-NET results obtained with the learned regularizer \mathcal{R}_θ . The fully variational RGN method with R_1 , R_2 and R_3 requires the critical selection of the optimal regularization parameter λ to get the best result, which is obtained by manually tuning λ in order to improve the accuracy of the results. We report only three λ values for each regularizer, between them the optimal ones. Unlike, the learned regularizer \mathcal{R}_θ automatically varies itself since it is driven by the data, and thus it is completely parameter free. The learned regularizers, starting from different settings of initial weights in the FCNN, for the same data set V_m , lead to conductivity reconstructions of the same quality and to a significant improvement of the condition number of the linear system (4.4) preserving the same order of magnitude. This allows to produce a quasi optimal reconstruction.

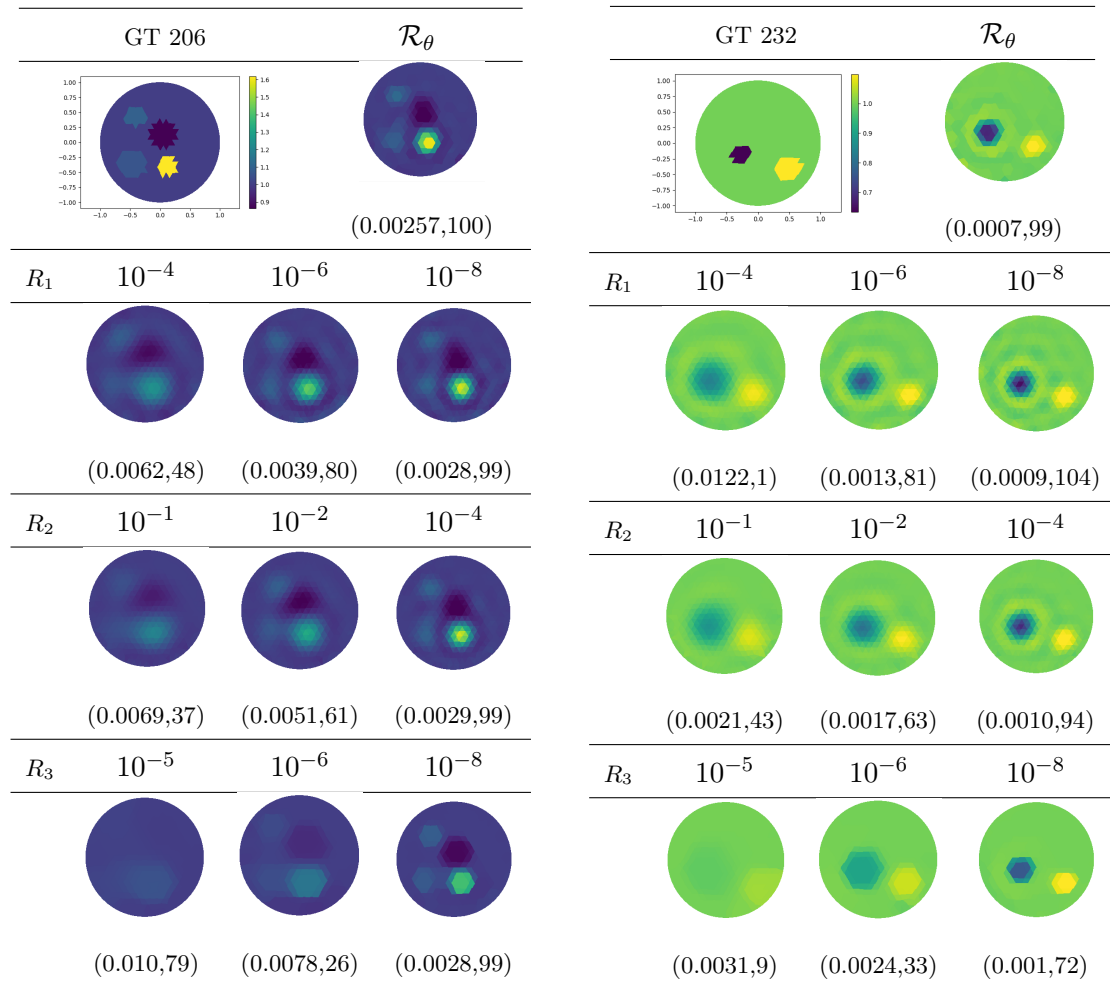


Figure 4.6: Regularized Gauss-Newton reconstruction results for samples 206 and 232 with associated (MSE,DR) values for different regularizers and different λ values together with our learned regularizer \mathcal{R}_θ .

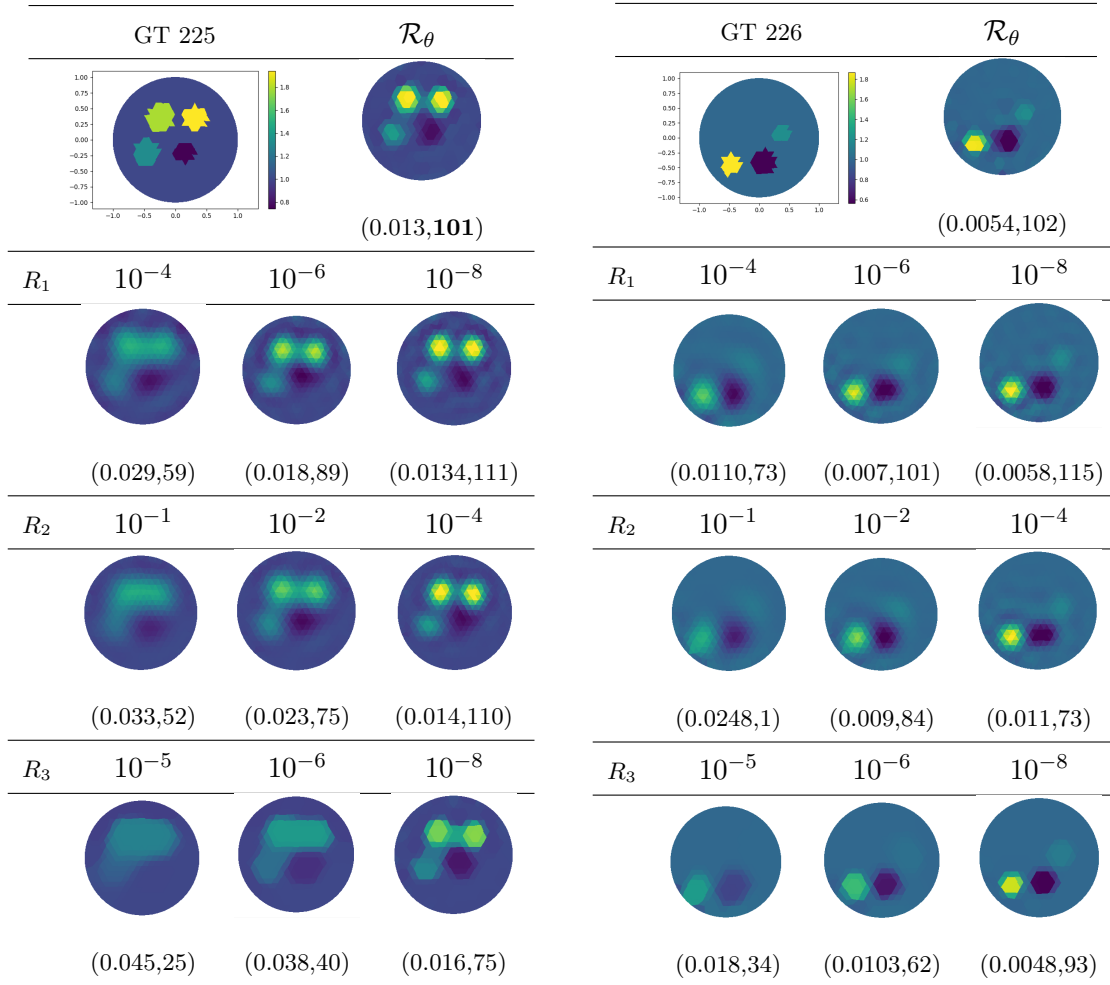


Figure 4.7: Regularized Gauss-Newton reconstruction results for samples 225 and 226 with associated (MSE,DR) values for different regularizers and different λ values together with our learned regularizer \mathcal{R}_θ .

Chapter 5

PnP strategy for EIT inverse problem

This section is devoted to the solution of non-linear ill-posed inverse problems formulated as an operator equation. For convenience we rewrite the problem:

$$\text{given } y \in R(F), \text{ find } x \text{ so that } F(x) = y, \quad (5.1)$$

where $F : D(F) \subset X \rightarrow Y$ is a Fréchet differentiable non-linear operator, mapping the Hilbert spaces X and Y with domain $D(F)$ and range $R(F)$. We propose a proximal Gauss-Newton method for the optimization problem (5.1) arising from regularization of ill-posed non-linear inverse problems. Proximal Newton-type methods, generalizing Newton-type methods, minimize the sum of a Lipschitz differentiable function and a non-smooth function handled using a proximal step [LSS14]. The difficulties that we want to face in this way are twofold. First, the ill-posedness of the problem which leads to the necessary adoption of proper strategies for handling indefinite Hessian approximations. The simplest strategy consists in adding a multiple of the identity to the Hessian approximation. In our formulation, this is overcome by incorporating a Tikhonov-like regularization term. Moreover, the proximity operator might not be computable, either efficiently, or at all. To overcome this difficulty, following the successful Plug-and-Play strategy, we propose to replace at each iteration $k \in \mathbb{N}$, the proximity step with an efficient denoiser algorithm. We propose a deep-PnP proximal Gauss-Newton method which incorporates a deep learning denoiser tailored to the data. The suggested denoiser for unstructured data, extends the standard case of data defined on regular grids, which is commonly considered in the imaging contexts. The proposed deep-PnP denoiser is based on a graph convolutional network, proposed in [GJ19], instead of the more popular Convolutional Neural Networks (CNN), thus widening the fields of applications.

With the intent of showing the benefits of applying the deep-PnP proximal Gauss-Newton method to a highly non-linear and ill-posed problem, we focus on the EIT inverse problem for the reconstruction of conductivity distributions. The main contributions of the section are summarized below:

- On the algorithmic side, we introduce a proximal Gauss-Newton method with a scaled proximal mapping, which is addressed to the solution of non-linear ill-posed inverse problems.
- We propose, and mathematically formalize, a deep denoiser based on a graph convolutional network which, under mild assumptions, satisfies the Lipschitz condition.

- We introduce a deep-PnP framework which integrates the graph convolutional denoiser into the proximal Gauss-Newton method, thus avoiding the crucial parameter tuning phase and naturally learning the best fitting prior to the data. This is shown to perform greatly for the non-linear EIT problems.

5.1 Regularizing non-linear ill-posed inverse problems

Given the Tikhonov regularization, i.e., to seek an approximation x_λ^δ to the solution of (5.1) as a minimizer of the quadratic functional

$$\mathcal{J}(x; \lambda) := \frac{1}{2} \left\| F(x) - y^\delta \right\|_Y^2 + \frac{\lambda}{2} \|L(x - x_0)\|_X^2, \quad (5.2)$$

where $\lambda > 0$ is the regularization parameter and L is a linear differential operator. The second term in the functional (5.2), so-called regularizer, favours minimal norm and smoothness properties of the solution, and the initial guess x_0 in (5.2) plays the role of a selection criterion. In case available a priori information this can be used in the selection, or, if possible, even a rough reconstruction $x_0 = F^{-1}(y^\delta)$ can be applied.

For a linear operator F , the Tikhonov functional (5.2) is a global convex function. But this property is lost as soon as the operator is non-linear. As a consequence, the nonconvex functional (5.2) might have several local minima, and it is not clear to which minimum a minimization process will converge. Moreover, a different choice of x_0 can lead to a different local minimum.

The differentiability of the cost function in (5.2) allows us to apply Gauss-Newton method to determine a minimizer. The downside, however, is that the l_2 -norm regularizer can cause considerable over-smoothing on the solution and therefore do not favor sparse or spatially inhomogeneous (containing discontinuities) solutions, which are instead sought by many applications.

Motivated by the success of the wide class of sparsity-inducing penalty functions in the regularisation of ill-posed linear problems [LMS15, SLMS20], we propose to extend the function in (5.2) by incorporating an eventually non-convex and not necessarily differentiable penalty function $h : X \rightarrow \mathbb{R} \cup \{+\infty\}$, as follows

$$\mathcal{J}(x) := \underbrace{\frac{1}{2} \left\| F(x) - y^\delta \right\|_Y^2 + \frac{\lambda}{2} \|L(x - x_0)\|_X^2}_{f_\lambda(x)} + h(x), \quad (5.3)$$

where $f_\lambda(x)$ is continuous and differentiable.

For the purpose of developing a numerical solution of the optimization problem, we assume the vector function $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $m \leq n$ to be Fréchet differentiable, and we define the gradient and Hessian of $f_\lambda : \mathbb{R}^n \rightarrow \mathbb{R}$, as follows

$$\nabla f_\lambda(x) = J^T(x)(F(x) - y^\delta) + \lambda L^T L(x - x_0) \quad (5.4)$$

$$\nabla^2 f_\lambda(x) = \nabla^2 F(x)(F(x) - y^\delta) + J^T(x)J(x) + \lambda L^T L, \quad (5.5)$$

with $J \in \mathbb{R}^{m \times n}$, $J = F'(x)$, the Jacobian matrix with components $(J(x))_{ij} = \frac{\partial F_i}{\partial x_j}(x)$.

Our task can be formulated as the minimization problem (1.14) where we consider $f(x) = f_\lambda(x)$ and $g(x) = h(x)$. The minimization is computed by apply the proximal Newton-type method as explained in Section 1.4.2.

5.2 Characterisation of proximal Newton-type methods

In the following result, in order to face the ill-posedness, we characterize the proximal Newton-type method for a specific choice of the matrix H_k , $f_\lambda(x)$ and $h(x)$ a lower semi-continuous, convex, eventually non-smooth function. Moreover, we will easily show that the proximal Newton-type method approximates the solution of the non-linear ill-posed problem by iteratively solving optimization sub-problems where $\tilde{f}_\lambda(x)$ contains a good linear approximation of the non-linear operator $F(x)$.

Proposition 5.1. *Let f_λ be defined in (5.3), with F be a non-linear, Frechét differentiable operator, and using for the Hessian in (5.5) the following approximation*

$$\nabla^2 f_\lambda(x) \approx H = J^T(x)J(x) + \lambda L^T L. \quad (5.6)$$

The iterative step k th, with $k = 1, 2, \dots$, of the proximal Gauss-Newton method (1.20)-(1.21) reads as

$$z_{k+1} = x_k - H_k^{-1}(J^T(x_k)(F(x_k) - y^\delta) + \lambda L^T L(x_k - x_0)) \quad (5.7)$$

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|J(x_k)(x - z_{k+1})\|_2^2 + \frac{\lambda}{2} \|L(x - z_{k+1})\|_2^2 + h(x) \right\} \quad (5.8)$$

which is equivalent to

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \tilde{f}_\lambda(x) + h(x) \right\}, \quad (5.9)$$

where $\tilde{f}_\lambda(x) = \frac{1}{2} \|F(x_k) - y^\delta + J(x_k)(x - x_k)\|_2^2 + \frac{1}{2} \lambda \|L(x - x_0)\|_2^2$.

Proof By simply replacing ∇f_λ defined as in (5.4), in (1.20), we get (5.7). As concerns (5.8), starting from (1.21) and applying (1.25) and (5.6), we get

$$\|y - x\|_H^2 = \langle H(y - x), (y - x) \rangle = \langle (J^T J + \lambda L^T L)(y - x), (y - x) \rangle = \|J(y - x)\|_2^2 + \lambda \|L(y - x)\|_2^2.$$

Equations (5.7)-(5.8) are equivalent to the scaled proximal map

$$x_{k+1} = \text{prox}_h^{H_k}(x_k - H_k^{-1}(J^T(x_k)(F(x_k) - y^\delta) + \lambda L^T L(x_k - x_0))). \quad (5.10)$$

In order to prove the equivalence between (5.7)-(5.8) and (5.9), we prove the equivalence between (5.10) and (5.9). At this aim, we consider the first-order necessary conditions for (5.9), satisfied for x^* , which read as

$$0 \in \nabla \tilde{f}_\lambda(x^*) + \partial h(x^*),$$

with

$$\nabla \tilde{f}_\lambda(x) = J^T(x_k) \left(F(x_k) - y^\delta + J(x_k)(x - x_k) \right) + \lambda L^T L(x_k - x_0).$$

For any fixed positive definite matrix H the followings are equivalent:

$$\begin{aligned} 0 &\in H^{-1} \nabla \tilde{f}_\lambda(x^*) + H^{-1} \partial h(x^*) \\ 0 &\in H^{-1} \nabla \tilde{f}_\lambda(x^*) - x^* + x^* + H^{-1} \partial h(x^*) \end{aligned}$$

$$(I + H^{-1}\partial h)(x^*) \in (I - H^{-1}\nabla\tilde{f}_\lambda)(x^*)$$

$$x^* \in (I + H^{-1}\partial g)^{-1}(I - H^{-1}\nabla\tilde{f}_\lambda)(x^*). \quad (5.11)$$

According to (1.19), equation (5.11) can be interpreted as fixed point scheme that generates a sequence $\{x_{k+1}\}$ by iterating:

$$x_{k+1} \in (I + H_k^{-1}\partial h)^{-1}(I - H_k^{-1}\nabla\tilde{f}_\lambda)(x_k)$$

namely,

$$x_{k+1} \in (I + H_k^{-1}\partial h)^{-1} \left(x_k - H_k^{-1}(J^T(x_k)(F(x_k) - y^\delta) + \lambda L^T L(x_k - x_0)) \right)$$

which can be rewritten as the proximal map (5.10).

A standard choice for the regularization term is $h(x) := \|\nabla x\|_1$ which leads to the minimization sub-problem:

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \|\nabla x\|_1 + \frac{1}{2} \|J(x - z_{k+1})\|_2^2 + \frac{\lambda}{2} \|L(x - z_{k+1})\|_2^2 \right\}. \quad (5.12)$$

An approximate solution of (5.12) can be obtained by a simple proximal gradient method, which, starting from $\tilde{x}_0 = z_{k+1}$, iterates for $i = 0, \dots$, until convergence to \tilde{x}^* , as

$$\tilde{z}_{i+1} = \tilde{x}_i - \tilde{\beta} H_k(\tilde{x}_i - z_{k+1}) \quad (5.13)$$

$$\tilde{x}_{i+1} = \text{prox}_h(\tilde{z}_{i+1}) \quad (5.14)$$

where the step-size is required to satisfy $\tilde{\beta} \in (0, \frac{1}{\rho(H_k^T H_k)})$, and H_k is defined in (5.6). Finally $x_{k+1} = \tilde{x}^*$.

The convex optimization problem (5.12) is a typical TV-L2 model which can be efficiently solved for structured data. However, in this work we consider a general 2-dimensional spatial domain discretized by an unstructured triangular mesh, so dealing with unstructured data. In this case, an efficient way to solve (5.12) is to apply the anisotropic TV algorithm proposed in [CLMS22] as a generalization of the original proposal in [LO09] for structured data.

The specific choice of the Hessian approximation given in (5.6) characterizes the proximal Newton-type algorithm for the solution of the optimization problem (1.14) as *proximal Gauss-Newton algorithm*. The overall sketch is provided in Algorithm 4 for the ℓ_1 -norm penalty $h(x)$.

Algorithm 4 Proximal Gauss-Newton-TV (PGN-TV)

Input: $x_0, \lambda > 0$,

Output: x^* % conductivity distribution

for $k = 0, \dots$ **do**

Update metric H_k from (5.6)

$z_{k+1} = x_k - H_k^{-1}\nabla f_\lambda(x_k)$

$x_{k+1} \leftarrow \text{prox}_h^{H_k}(z_{k+1})$ solved by (5.13)-(5.14)

end for

end

5.3 Scaled Plug-and-Play algorithm

An important feature of the proximal Gauss-Newton method (5.7)-(5.8) is its modular structure. In particular, (5.7) can be regarded as an updating step as it involves the forward model $f_\lambda(x)$, whereas (5.8) can be regarded as a variable metric denoising step as it involves the prior $h(x)$. In particular, (5.8) can be rewritten as

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \|x - z_{k+1}\|_{H_k}^2 + h(x) \right\}. \quad (5.15)$$

In case of image restoration, the variational model (5.15) is a typical Maximum A Posteriori (MAP) denoiser where z_{k+1} represents the noisy observation. With a little abuse of notation, since we are dealing with the more challenging task of image reconstruction, z_{k+1} in (5.15) is the gradient step result, i.e. an intermediate reconstruction of x .

The mathematical equivalence of the proximal operator to the MAP denoising has inspired Venkatakrishnan [VBW13] to introduce the plug-and-play priors framework for image reconstruction. This method exhibited great empirical success, and encouraged much follow-up work. Thus, taking the iterative proximal Gauss-Newton method sketched in Algorithm 4 and replacing the proximal operator with a off-the-shelf denoiser \mathcal{D}_H , we get

$$x_{k+1} \leftarrow \mathcal{D}_H(z_{k+1}). \quad (5.16)$$

The overall method can thus be called *proximal PnP- Gauss-Newton algorithm*. PnP strategies leverage both successful denoiser algorithms and deep learning-based denoiser. In the following section we introduce our learned PnP denoiser \mathcal{D}_H based on graph convolutional neural network.

5.4 The learned GU-Net-denoiser

With the development of deep learning techniques in image processing, CNN-based denoisers have shown promising performance in terms of effectiveness and efficiency. However, they can be applied only to data defined on regular grid domains, typically derived from imaging problems [JWZ21]. Here, we consider instead a more general neural network which works on unstructured data, as for all the problems derived from FEM models. This allows us to overcome the inaccuracy that inevitably arises if you work on a grid-based approximation of a domain originally represented by a triangular mesh. The denoising problem on meshes is here conceived for a function defined over a 2-manifold discretized by a mesh.

In the following we describe the proposed GU-Net-denoiser which extends the CNN-based denoiser to non-Euclidian manifold domain and relies on the encoder-decoder Graph-U-Net architecture [GJ19]. Graph-U-Net is a U-Net-like architecture for graph data which allows high-level feature encoding and decoding for network embedding. It is based on a convolution graph kernel and gPool and gUnpool layers. The pool (gPool) operation samples some nodes to form a smaller graph based on their scalar projection values on a trainable projection vector. As an inverse operation of gPool, the unpooling (gUnpool) operation restores the graph to its original structure with the help of locations of nodes selected in the corresponding gPool layer [GJ19].

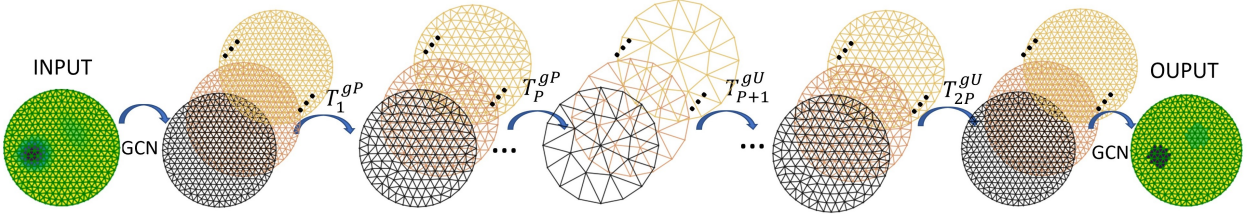


Figure 5.1: GU-Net denoiser structure.

The GU-Net-denoiser as well as the Graph-U-Net can be formalized as a composition of functions. At this aim, let us define an input mesh with n vertices characterized by an adjacent matrix $A \in \mathbb{R}^{n \times n}$, with non-zero entries $a_{ij} = 1$ for directly connected mesh nodes i and j and $a_{ii} = 1$ to incorporate connections from vertex i to itself. A Graph-U-Net with M layers is defined as the following composite function

$$\mathcal{G} = T_M \circ \dots \circ T_1, \quad (5.17)$$

with $(T_\ell)_{1 \leq \ell \leq M}$ being the layers of the network. Each layer is characterized by the composition of a graph convolution, named GCN [WK16], a ReLU activation function σ and a gPool/gUnpool operator, here denoted by a generic p , and is applied to the n_c -dimensional input feature array $X \in \mathbb{R}^{n \times n_c}$, namely

$$T_\ell : X \mapsto \sigma(\text{GCN}(p(X); \Theta_\ell)), \quad (5.18)$$

where $\Theta_\ell \in \mathbb{R}^{n_c \times n_f}$ denotes the trainable weight matrix of layer ℓ , and n_f is the number of output features of the layer ℓ .

Let $\bar{A} := D^{-1/2} A D^{-1/2}$, where D is the diagonal node degree matrix to normalise A with entries $D_{ii} = \sum_{j=1}^n a_{ij}$, and $Y_\ell = p(X_\ell)$.

The layer-wise forward-propagation operation T_ℓ , for each $\ell = 1, \dots, M$ applied to the input feature array of layer ℓ , X_ℓ , is defined as follows

$$X_{\ell+1} = T_\ell(X_\ell) := \sigma((D^{-1/2} A D^{-1/2}) p(X_\ell) \Theta_\ell),$$

where $\sigma(\cdot)$ is applied point-wise, and we used

$$\text{GCN}(p(X); \Theta_\ell) := \bar{A} p(X_\ell) \Theta_\ell. \quad (5.19)$$

Hence, the linear convolutional operator GCN, maps a feature matrix Y into a convolved data matrix $Z = \bar{A} Y \Theta$. Note that $D^{-1/2} A D^{-1/2}$ has eigenvalues in the range $[0, 1]$.

We denote by $\theta = \{\Theta_\ell\}_{\ell=1}^M$ the set of all the trainable weights of the GU-Net-denoiser. If we consider as the complexity of the network the number of trainable weights, then it is given by $2(n_c \times n_f) + M n_f^2$. Since the weights do not depend on the number of vertices of the mesh n , the trainable weights can be easily used for any dimension of the input mesh. This is an important aspect of our GU-Net denoiser proposal.

We now rewrite the gPool and gUnpool operators described in [GJ19] as mathematical operators. At this aim, we incorporate these operators into (5.17), so that the encoder-decoder network

reads as follows:

$$\mathcal{G} = \underbrace{T_{2P}^{gU} \circ \dots \circ T_{P+1}^{gU}}_{\text{decoder}} \circ \underbrace{T_P^{gP} \circ \dots \circ T_1^{gP}}_{\text{encoder}}, \quad (5.20)$$

where in T_ℓ^{gP} , $p : X \mapsto gPool(X)$, for $\ell = 1, \dots, P$, and T_ℓ^{gU} , $p : X \mapsto gUnpool(X)$, for $\ell = P+1, \dots, 2P$, with $M = 2P$. Let $\nu \in \mathbb{R}^N$ be a normalized trainable vector, and $S \in \mathbb{R}^{k \times N}$ be the matrix obtained by eliminating from the identity matrix I_N the rows corresponding to the nodes selected by a vector. At each gPool layer ℓ , (encoder of level ℓ) with $\ell = 1, \dots, P$, corresponds a gUnpool layer $\ell_u = \ell + P$ (decoder of level ℓ_u) which shares the same sampling matrix S defined by the vector of the indices of the first k maximum values of the vector $X_\ell \nu^\ell$. Specifically, we have

- $gPool$, defined as

$$X_{\ell+1} = (SX_\ell) \odot s(S(X_\ell \nu^\ell)) \mathbf{1}_{n_c}^T \quad (5.21)$$

where s is a sigmoid function applied point-wise, the symbol \odot denotes the Hadamard product, $\mathbf{1}_{n_c} \in \mathbb{R}^{n_c}$ is a vector with all components being 1.

Update of the Adjacent matrix: $A_{\ell+1} = SA_\ell S^T$.

- $gUnpool$ defined as

$$X_{\ell_u+1} = S^T X_{\ell_u}. \quad (5.22)$$

Update of the Adjacent matrix: $A_{\ell_u+1} = S^T A_{\ell_u} S$.

Note that the value of k for the $gPool$ operator is predefined and fixed for all the levels; it defines the percentage of down-sampling (and corresponding up-sampling in $gUnpool$) on graph data.

The overall structure of the GU-Net-denoiser is illustrated in Fig.5.1.

5.4.1 On the nonexpansiveness of the GU-Net-denoiser

Building nonexpansive network appears to be the key to obtain convergent Plug-and-Play algorithms. It is indeed shown that a sufficient condition to get the convergence of PnP iterates is to ensure the firm nonexpansiveness of the denoiser [RLW⁺19].

A method to build such nonexpansive network relies on an accurate control of the Lipschitz constant of the convolutional layers. In particular, by tightly constraining the Lipschitz constant of feedforward graph convolutional network to be smaller than 1.

We recall that an operator $T : X \rightarrow X$ is Lipschitzian with constant contraction factor $\delta \in]0, +\infty[$ if $(\forall x \in X)(\forall y \in X)$

$$\|Tx - Ty\| \leq \delta \|x - y\|. \quad (5.23)$$

If $\delta < 1$ above, then T is a strict contraction. If $\delta = 1$ then T is nonexpansive.

The following result easily derives from [GFPC21].

Lemma 5.2. *The upper bound on the Lipschitz constant for the entire \mathcal{G} network defined in (5.17), assuming that each layer $T_\ell, \ell = 1, \dots, M$, defined in (5.18) is represented by the same composition of functions, is given by:*

$$\mathcal{L}(\mathcal{G}) \leq (\mathcal{L}(T_\ell))^M, \quad (5.24)$$

where $\mathcal{L}(T_\ell)$ is the Lipschitz constant of the generic layer ℓ .

Next we aim to prove a sufficient condition to ensure the nonexpansiveness of the GU-Net convolutional network, that is to ensure that, for every layer ℓ , the operator T_ℓ defined in (5.4) has a Lipschitz constant $\mathcal{L}(T_\ell)$ lower than 1, under mild assumptions on Pool and Unpool operators.

Proposition 5.3. *Let \mathcal{G} be the GU-Net defined in (5.20), with M layers $T_\ell : X \mapsto \sigma(\text{GCN}(p(X); \Theta))$, defined in (5.18), $\ell = 1, \dots, M$, where $\text{GCN}(X; \Theta)$ is the graph convolutional operator with $n_c \in \mathbb{N}$ input channels and $n_f \in \mathbb{N}$ output channels, σ is a ReLU function, and $p(X_\ell)$ is a Pool or Unpool operator defined as SX_ℓ and $S^T X_\ell$, respectively. Assuming each layer-wise weight matrix Θ_ℓ has $\|\Theta_\ell\| \leq 1$, then, \mathcal{G} is nonexpansive.*

Proof Since for Lemma 5.2 the Lipschitz constant of the full network \mathcal{G} is bounded by Lipschitz constant $\mathcal{L}(T_\ell)$ of the generic layer T_ℓ , we now determine $\mathcal{L}(T_\ell)$. Let $X^{(1)}, X^{(2)}$ be two different input feature arrays, then from definition (5.23), and (5.19), we have

$$\begin{aligned} \|\text{GCN}(p(X^{(1)}); \Theta_\ell) - \text{GCN}(p(X^{(2)}); \Theta_\ell)\| &= \|(\bar{A}SX^{(1)}\Theta_\ell) - (\bar{A}SX^{(2)}\Theta_\ell)\| \\ &= \|\bar{A}S(X^{(1)} - X^{(2)})\Theta_\ell\| \\ &\leq \|\bar{A}\| \|S\| \|X^{(1)} - X^{(2)}\| \|\Theta_\ell\| \\ &\leq \|X^{(1)} - X^{(2)}\|. \end{aligned}$$

Here we neglected the activation function σ as the ReLU function used has Lipschitz constant of 1 [GFPC21], then we considered that the spectral norm of \bar{A} is 1, and S for pool - and analogously S^T for unpool - derived from eliminating some rows of the identity matrix. Finally, under the hypothesis on $\|\Theta_\ell\|$, the Lipschitz constant is 1, and thus \mathcal{G} is nonexpansive.

5.4.2 Deep-PnP proximal Gauss-Newton method

In our proposal the pre-trained GU-Net denoiser \mathcal{D}_H is incorporated into a proximal Gauss-Newton method giving raise to the *deep-PnP proximal Gauss-Newton method* which iterates over the following steps:

$$z_{k+1} = x_k - H_k^{-1} \nabla f_\lambda(x_k) \quad (5.25)$$

$$x_{k+1} = \mathcal{D}_H(z_{k+1}). \quad (5.26)$$

Regarding the forward step (5.25), the gradient is computed as in (5.4), where the operator F and its Fréchet derivative J are given by the forward generation model, and the metric H_k is given in (5.6), with matrix L corresponding to either the identity matrix or the discretization of the first-order derivatives. We will name the proposed method, shortly, *PnP-PGN*.

5.5 Application to the Inverse EIT Problem

In this section we consider the solution of the inverse EIT problem formulated in Chapter 3. To reduce the ill-posedness while keeping high accuracy reconstructions, the non-linear least squares

EIT problem is usually regularized by adding a penalty term, and reads as

$$x^* \in \arg \min_{x \in \mathbb{R}^{n_T}} f_\lambda(x), \quad f_\lambda(x) = \|F(x) - y^\delta\|_2^2 + \frac{\lambda}{2} \|L(x - x_o)\|_2^2, \quad (5.27)$$

with f_λ defined as in (5.3), $F: \mathbb{R}^{n_T} \rightarrow \mathbb{R}^{n_M}$ represents the discrete version of the Forward Operator defined by (3.1) which is a non-linear vector map. Since the continuous forward operator is Fréchet differentiable, F' is a matrix, called the Jacobian of F and here denoted by J . The functional in (5.27) is the classical generalized non-linear Tikhonov model, a standard choice in EIT [BGAL09]. With this regularizer choice, the approximated solution of the optimization problem (5.27) applying the iterative RGN is given by select the search direction p_k by solving the linear system

$$(J(x_k)^T J(x_k) + \lambda L^T L) p_k^{\text{RGN-Tik}} = J(x_k)^T (y^\delta - F(x_k)) - \lambda L^T L(x_k - x_o). \quad (\text{RGN-Tik})$$

The regularizing Levenberg-Marquardt (LM) algorithm simplifies the RGN algorithm, by computing the search direction as

$$(J(x_k)^T J(x_k) + \lambda L^T L) p_k^{\text{LM}} = J(x_k)^T (y^\delta - F(x_k)). \quad (\text{LM})$$

5.6 Applying PnP-PGN to the inverse non-linear EIT problem

The experiments in this section aim to evaluate the proposed method PnP-PGN on a set of synthetic 2D data related to inverse EIT problems. Each sample represents the synthetic ground-truth conductivity x^{GT} and consists of a random number from 1 to 4 of anomalies inside a circular tank of unitary radius, localized randomly and characterized by random radius in the range $[0.15 - 0.25]$ and magnitude in the range $[0.2, 2]$. The circular tank is tessellated by a mesh grid of 660 triangles. In the circular boundary ring, 16 electrodes are equally-spaced located. The homogeneous conductivity of the background liquid is set to be $x_0 = 1.0 \Omega m^{-1}$, and it is assumed that each actual anomaly approximately consists of the same material. This choice for x_0 can be far from the solution. Nevertheless, it has guaranteed convergence for all the experiments.

Measurements y^δ are simulated through *opposite injection - adjacent measurement* protocol via pyEIT, a python-based framework for Electrical Impedance Tomography [LYX⁺18]. The comparisons are conducted qualitatively by visually inspecting the artifacts and quantitatively on the mesh elements, by calculating the metrics mean-square error (MSE), and structural similarity (SSIM).

5.6.1 Training Data for the GU-Net-denoiser

When the GU-Net-denoiser is trained on the training dataset, it is initialised with a set of random weights. These weights are then optimised during the training period and the optimum set of weights θ are produced and used in \mathcal{D}_H , the step (5.26) of the deep-PnP proximal Gauss-Newton method.

The pre-trained GU-Net-denoiser has a network depth $M = 12$ (number of layers) for encoder and decoder, which corresponds to $P = 6$ in (5.20), and 0.6 for the down-sampling (up-sampling) factor. A ReLU activation function σ and a GCN convolution are applied to all layers in (5.4).

The GU-Net denoiser is characterized by an initial $n_c = 1$ which is then transformed to $n_c = 64$ by the first GCN (see Fig.5.1) and $n_f = 64$. Therefore, the complexity is $O(10^4)$.

The training data set consists of 18000 samples, (x^{GT}, \hat{x}^j) , composed by 3000 different conductivity configurations, each provided with 6 different reconstructions: the x^{GT} itself ($j = 0$) and $\hat{x}^j, j = 1, \dots, 5$, which characterize a decreasing level of accuracy, or, better, an increasing level of noise. The training sample $\hat{x}^j, j = 1, \dots, 5$ is an approximated solution of the inverse EIT problem obtained by applying the LM algorithm for 1,2,3,4,5 iterations, respectively, starting from corresponding collected voltages y .

The training is carried out by minimizing at each epoch k , over the training set of cardinality n_S , the following loss function

$$\theta_k^* \leftarrow \underset{\theta_k}{\operatorname{argmin}} \frac{1}{n_S} \left\{ \mathcal{L}(\theta_k) := (1 - \gamma) \sum_{i=1}^{n_S} \sum_{j=0}^5 \|\mathcal{G}(\hat{x}_i^j; \theta_k) - x_i^{GT}\|_2^2 + \gamma \sum_{i=1}^{n_S} \sum_{j=0}^5 \|\mathcal{G}(\hat{x}_i^j; \theta_k) - x_i^{GT}\|_1 \right\} \quad (5.28)$$

with $\gamma \in (0, 1)$, and $\mathcal{G}(\cdot; \theta_k)$ the deep GU-Net denoiser defined in (5.20). In particular, the training phase for the reported examples used $\gamma = 0.25$. Through empirical observations we found that 150 epochs were enough to sufficiently decrease the total loss function (5.28). Hence the weights θ_{150} at epoch 150 feed the GU-Net-denoiser at the inference phase.

In a similar way, a noisy training data set of 18000 noisy samples $(x^{GT}, \hat{x}^{\delta j})$ has been created where the approximated solutions $\hat{x}^{\delta j}$ have been obtained starting from corrupted voltage measurements y^δ , according to the degradation model

$$y^\delta = y + \eta, \quad (5.29)$$

with $\eta \sim \mathcal{N}(0, s^2)$ Gaussian distributed noise characterized by zero-mean and standard deviation $s = \delta \bar{y}$, with \bar{y} the average value. (6.2) acts as equation (3.7) in the EIT degradation model.

In the training, the learning rate is set to 2.5×10^{-3} and the standard Adam optimizer has been used. The network is implemented using PyTorch [PGM⁺19] and PyTorch Geometric [FL19], and it is tested on a PC with Intel i7 CPU and 32-GB RAM with an NVIDIA Quadro P620 (2 GB graphics memory). The training process took nearly 7 hours for both data sets.

5.6.2 Performance comparisons

We show the performance of the proposed framework PnP-PGN, for the reconstruction of 50 test cases, and we compare it with iterative methods such as RGN, and the proximal Gauss-Newton method (Algorithm 4, labelled as PGN-TV), as well as the variational network proposed in [CLMS22], named EITGN-NET.

The parameter λ in the function f_λ defined in (5.3) is set to 10^{-8} for the methods PnP-PGN, PGN-TV and RGN-Tik. This choice corresponds to a sufficiently low λ value that allows to handle well-conditioned matrices without any undesired over-regularization effect.

In our implementation, PGN-TV incorporates a total variation penalty, and the RGN method a regularization matrix L which is a second-order high-pass filter (Laplace prior) with homogeneous Neumann boundary conditions. Due to the triangular discretization of the domain Ω , each row of

	PnP-PGN (L= I)	PnP-PGN (L=D)	EITGN-NET	RGN-Tik	PGN-TV
MSE	0.0042	0.0036	0.0049	0.0067	0.0055
SSIM	0.91	0.91	0.87	0.80	0.86

Table 5.1: Performance on averaged MSE, SSIM metrics on the entire set of test cases.

the graph Laplacian L has only 4 non-zero elements: it has value -1 for each adjacent triangle, and 3 for the triangle itself.

For all the tests, the iterations of the algorithms were stopped as soon as either of the two following conditions is fulfilled:

$$k > 200, \quad err_k := \|x_{k+1} - x_k\|_2 / \|x_k\|_2 < 5 \cdot 10^{-4}. \quad (5.30)$$

In Table 5.1 we show the compared performance based on averaged MSE and SSIM metrics, on a set of 50 test cases. The first two columns allow to compare the proposed PnP-PGN using $L = I$ and $L = D$ in (5.6). We note the significant improvement of PnP-PGN with $L = D$ under the same computational framework. Compared with the other optimization algorithms, tuned with the optimal regularization parameters, the proposed PnP-PGN leads to more accurate reconstructions without any parameter set-up.

In Fig.5.3 we show the conductivity reconstructions obtained by applying PnP-PGN (with $L = D$), RGN-Tik and PGN-TV, to six representative test cases; together with the associated (MSE,SSIM) values. In Fig.5.3 (first column) is reported the ground truth (GT) image x^{GT} for a qualitative inspection. In the last row of Fig.5.3 we validated the behavior of the PnP-PGN method when applied to test cases with anomalies having a different shape with respect to the training data set used to pre-train the GU-Net-denoiser which included only circular anomalies. In particular, the case labeled by *heart and lungs* is a phantom image representing heart and lungs anomalies which, according to the MSE measure, are slightly better detected by the other methods in the comparison. This confirms the well-known drawback of the learning-based approaches when the testing dataset is different from what they have been trained with. We trust that, with suitable training, PnP-PGN will provide more accurate reconstructions even in case of similar anomalies.

In Fig. 5.2 we give empirical evidence on the convergence of the sequence $\{x_k\}$ generated by the different iterative optimization methods to the optimal solution x^{GT} , for each of the three representative test cases. In particular, the plots report the relative errors $err_k^{(GT)}$, in terms of the number k of iterations

$$err_k^{(GT)} := \|x_k^* - x^{GT}\|_2 / \|x^{GT}\|_2. \quad (5.31)$$

The solution obtained by the proposed PnP-PGM, with a prior learned by the data, outperforms the other results. In particular, it is more accurate with respect to its variational counterpart (PGN-TV) where the prior is pre-determined as total variation and the regularization parameter is specifically tuned. Moreover, PnP-PGN is much more efficient in terms of computational time, as reported in the right corner of each plot. The RGN-Tik is the fastest but the results show low accuracy.

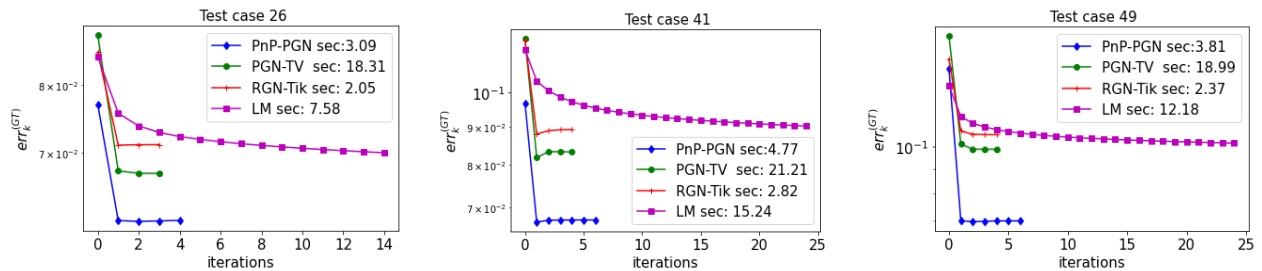


Figure 5.2: Relative error with respect to x^{GT} , obtained by the compared algorithms PnP-PGN, PGN-TV, RGN-Tik, LM for three different test cases.

5.6.3 Robustness to noise

We validated our PnP-PGN algorithm starting with acquisitions y^δ corrupted as in (6.2), using two different noise levels $\delta = 2.5 \times 10^{-3}$ (SNR=54dB) and the more severe $\delta = 5 \times 10^{-3}$ (SNR=48dB). In both cases, the collected weights θ were obtained by minimizing the loss function (5.28) with $\bar{x}^{\delta j}$ obtained starting from the noisy training data set corresponding to $\delta = 5 \times 10^{-3}$.

In Fig.5.4 and Fig.5.5 we show the conductivities obtained by the methods PnP-PGN, RGN-Tik and PGN-TV with both type of noisy measurements for four test cases. From a visual inspection the proposed PnP-PGN method tends to better preserve sharper structures, presents a lower number of artifacts, and in general improves the separation between the anomalies, even in case of strong noise degradations, see Fig.5.5.

To summarize the performance in the noisy experiments, in Table 5.2 we report the averaged values on 50 test samples, for the two different degraded inputs, corresponding to a SNR of 48dB and 55dB, which confirm for both the metrics MSE and SSIM the benefits of a data-driven regularization approach in a variational well performing optimization method.

	PnP-PGN	RGN-Tik	PGN-TV
noise 55dB			
MSE	0.0043	0.0088	0.0083
SSIM	0.90	0.76	0.78
noise 48dB			
MSE	0.0049	0.0092	0.0088
SSIM	0.89	0.71	0.73

Table 5.2: Noisy measurements: performance on averaged MSE and SSIM metrics on the entire set of test cases.

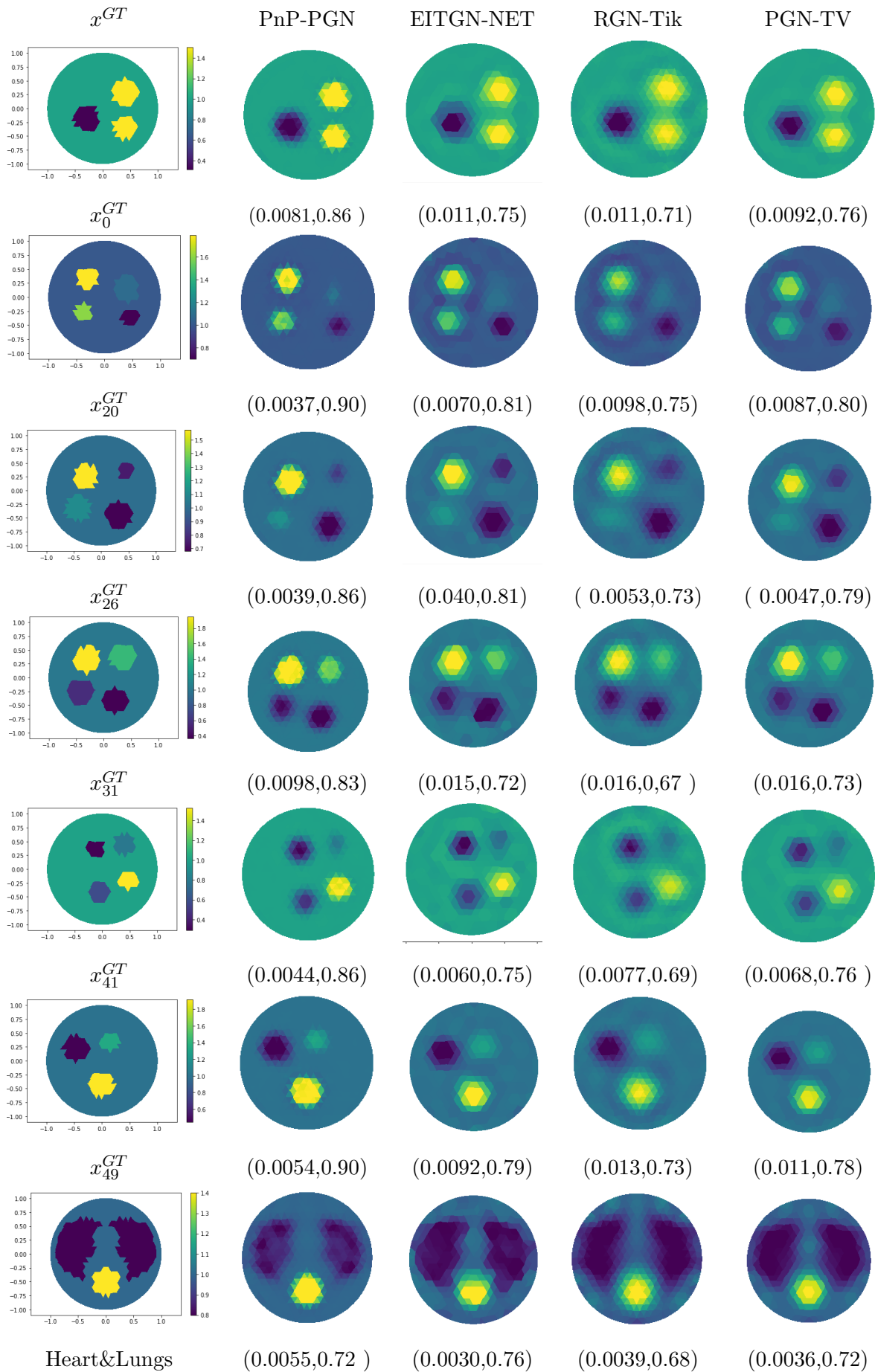


Figure 5.3: Reconstructions obtained by (column-wise from 2-5) PnP-PGN, RGN-Tik, EITGN-NET and PGN-TV; x^{GT} (first column) is the ground truth conductivity; (MSE,SSIM) values are reported for each test case.

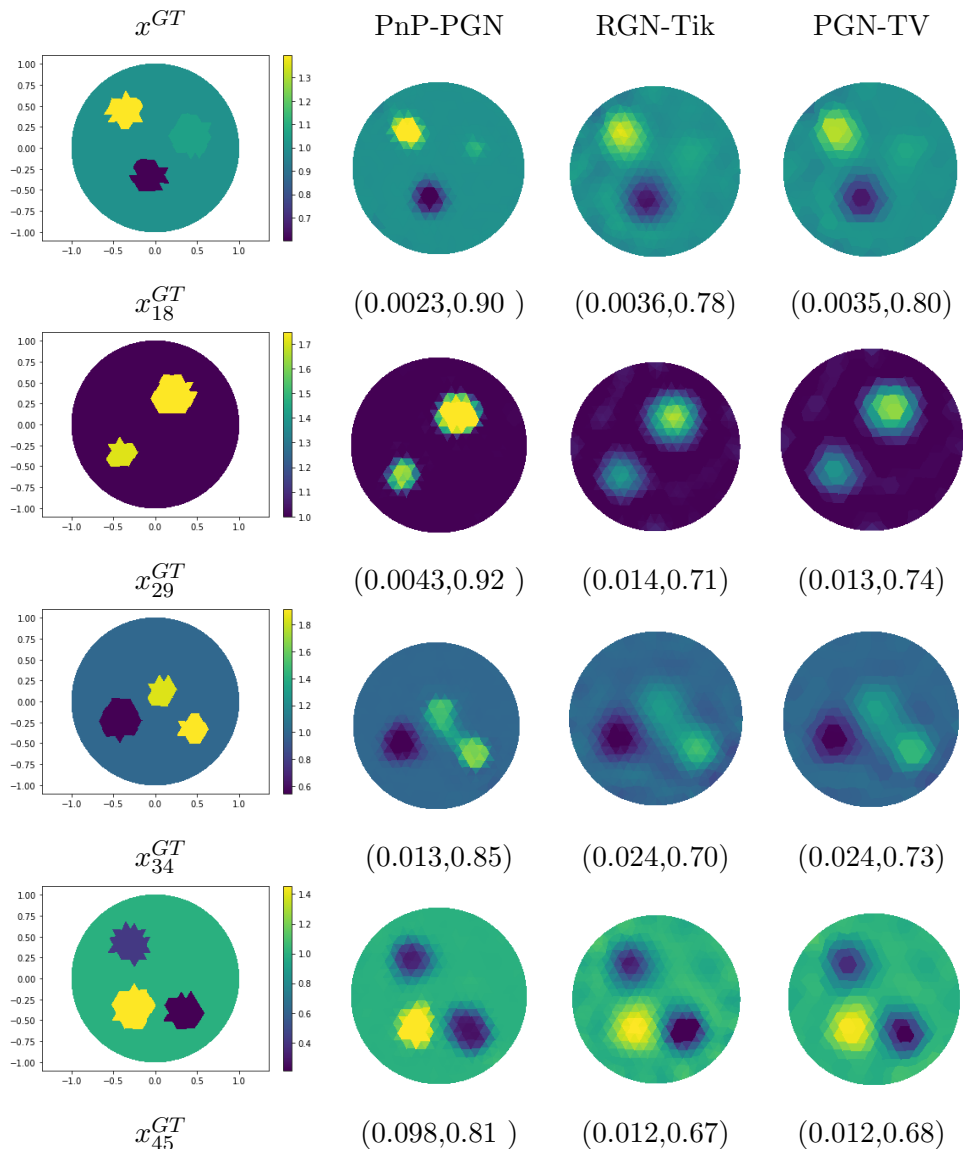


Figure 5.4: Robustness to the noise: comparison between x^{GT} and reconstructed conductivities from noisy measurements (SNR=55 db) obtained by PnP-PGN (first column), RGN-Tik (second column) and PGN-TV (third column).

5.6.4 Performance with experimental data

We tested our PnP-PGN algorithm on the open 2D electrical impedance tomography dataset [HKM⁺18] described in detail in [HKM⁺17].

The setup consists of a circular tank containing an electrically conductive solution and 16 uniformly spaced electrodes along the boundary. Electrodes span the whole depth of the tank, thus resulting in a 2D EIT configuration. Either plastic (low conductivity), metal (high conductivity) or, both kinds of objects are placed inside the tank, then measurements are acquired following an adjacent current injection - adjacent voltage measurement protocol. A total of 16 adjacent current injections were performed for each example, and for each current injection 13 adjacent

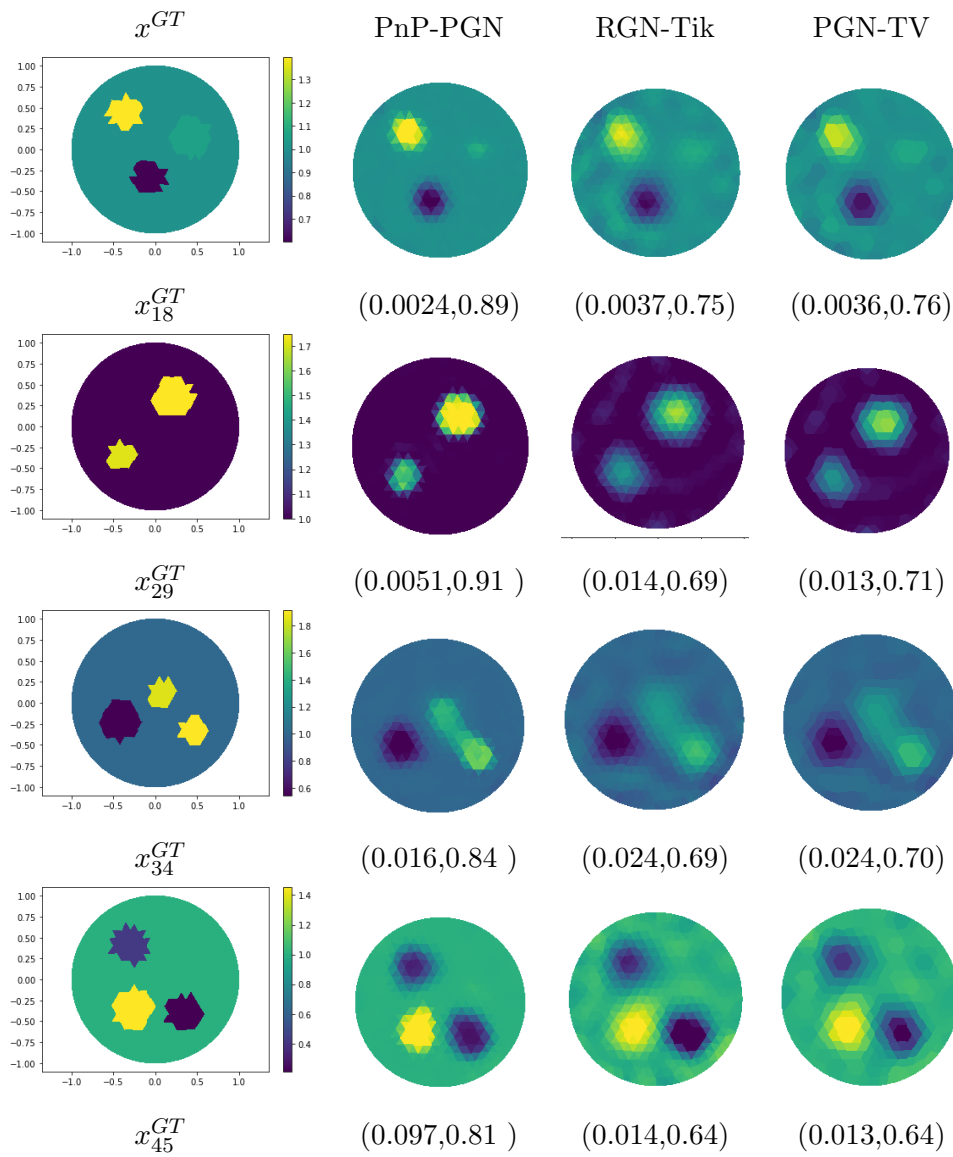


Figure 5.5: Robustness to the noise: comparison between x^{GT} and reconstructed conductivities from noisy measurements (SNR=48db) obtained by PnP-PGN (first column), RGN-Tik (second column) and PGN-TV (third column).

voltage measurements were recorded, excluding current-injecting electrodes. This results in 208 measured voltages which were then used for EIT reconstruction with the PnP-PGN and RGN-Tik algorithms for comparison. Reconstruction results are illustrated in Fig. 5.6. It is clear that the reconstructions obtained with the proposed PnP-PGN are of higher quality with respect to the RGN-Tik approach. PnP-PGN reconstructions contain very few background artifacts and also perform remarkably well even in the difficult case of hollow metal objects. In fact, with PnP-PGN the hole is visible in the reconstruction while it is completely filled in the RGN-Tik solution.

Shapes cannot be recovered by either algorithm. However, it should be noted that the training of the neural network part of PnP-PGN involved only circular anomalies. It is therefore expected that a training dataset containing other geometrically or irregularly shaped objects would further improve reconstruction performance.

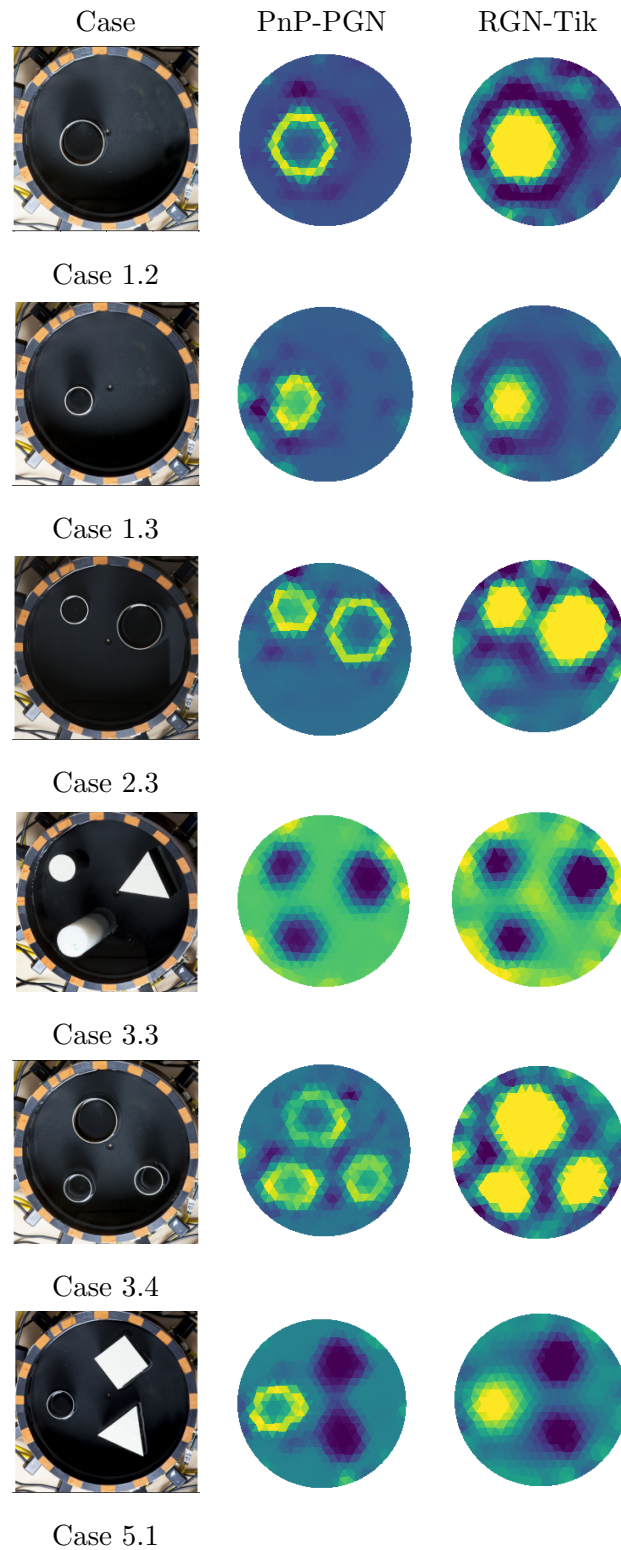


Figure 5.6: Reconstructions with experimental data from the 2D electrical impedance tomography dataset [HKM⁺18]. Picture of the setup (left), reconstruction with the proposed PnP-PGN (centre), reconstruction with RGN-Tik for comparison (right).

Chapter 6

Limited electrodes EIT reconstruction

In Electrical Impedance Tomography (EIT), the electrical conductivity σ inside a domain Ω is estimated from current and voltage measurements (I, V) at its boundary $\partial\Omega$. The voltage and current values are measured through a finite number n of electrodes attached to the boundary of the object. Having more electrodes should result in higher reconstruction quality. The number of electrodes used is a trade-off between accuracy, measurement time, and processing time. We expect, in general, that having more electrodes and thus more measurements, results in a higher reconstruction quality. However, increasing their number is not straightforward due to the added complexities and costs involved both in the instrumentation and in the time required to set up the system. Also, increasing the number of electrodes can lead to reduce their size, and thus increase contact impedance. Moreover, the more components are present in a system, the easier it is to have a few of them malfunctioning, especially electrodes, thus reducing reliability. It is thus clear that obtaining high quality EIT reconstructions with a reduced number of electrodes (and thus of measurements) would be of great help to reduce the costs and increase the reliability of EIT in practical applications. It would be even more useful to know the sufficient number of measurements to recover an optimal σ , under sparsity conditions on the unknown conductivity.

In this chapter, we formulate this problem by naming it Limited Electrodes (LE) problem and denoting with LE Model (LEM) the numerical methods proposed for its solution. There seems to be a natural connection with the basic ideas of Compressed Sensing (CS) theory which deals with the recovery of a sparse signal from a small number of linear measurements. In a well-posed context, the CS theory, under appropriate hypotheses, identifies a relationship between the minimum number of measurements necessary to obtain a "good reconstruction" of the signal, its size and its sparsity in some domains. In this section, we instead address the compressed sensing recovery problem in a setting where the observations are non-linear and we are concerned with sparse recovery principles for non-linear ill-posed inverse problems. Suppose we know the minimum number of measurements required to obtain an optimal σ^* reconstruction, but we are under the more realistic assumption of having fewer measurements (electrodes) available. It would be desirable to apply a method (LEM) which nevertheless allows us to obtain a reconstruction of quality comparable to σ^* .

In this work, we investigate two different LEMs to the EIT-LE problem. The first approach involves the applicability of CS techniques by exploiting the gradient sparsity of the conductivity distribution, which leads to a non-linear variational problem. The second proposal is a learned

residual approach on raw boundary impedance data. Both strategies will recover a reconstruction quality obtainable with a high number of electrodes (and thus measurements) while using for acquisitions a relatively low number of electrodes.

6.1 Preliminaries on EIT reconstructions

Following the formulation in Section 3.1, one can fully describe the behavior of the voltages in the electrodes E_i at the object/domain boundary $\partial\Omega_{E_i}$, given the conductivity of the medium σ and a set of injected currents $I_F = [I_1, \dots, I_n]$. This process constitutes the EIT Forward model and is described by a non-linear forward operator denoted by $F(\sigma; I_F)$. Since the continuous forward operator is Fréchet differentiable, F' is a matrix, called the Jacobian of F and in the following denoted by J . The numerical computation of the EIT forward model can be obtained by employing Finite Element (FE) techniques using a discrete approximation Ω_h of the domain Ω , represented by a triangulated mesh composed of n_T triangles.

Given a set of voltage and current measurements, the goal of the EIT inverse problem is to compute the approximate conductivity σ^* such that the voltage potentials predicted through the forward problem are as close to the actual measurements V as possible, which reads as

$$\sigma^* \in \arg \min_{\sigma \in \mathbb{R}^{n_T}} \|r(\sigma)\|_2^2, \quad r(\sigma) := F(\sigma; I_F) - V. \quad (6.1)$$

The underlying optimization problem is hard to solve, not only as the boundary currents depend non-linearly on the conductivity which makes the optimization problem non-convex, but also for the well-known sensitivity of the solutions to small voltage perturbations. To reduce the ill-posedness while keeping high accuracy reconstructions, the non-linear least squares EIT problem (6.1) is usually regularized by adding a penalty term and solving the regularized non-linear minimization problem by the Gauss-Newton iterative method, reported in 1.4.2 to the intended purpose of use in the proposed numerical solution of the EIT-LE problem. We investigate how to suit these regularization methods to deal with the challenge EIT-LE problem with a low number of data/measurements. The quality of the conductivity reconstruction σ^* is strictly related to the quantity and the quality of the acquired measurements V . The quality of the conductivity reconstruction σ^* is strictly related to the quantity and the quality of the acquired measurements V . In this simplified context we are aware of neglecting many other factors which could affect the reconstruction accuracy, i.e., the mismodeling of the domain and misplacement of electrodes. The data acquisition depends on the stimulation pattern. In our work, the collected measurements V are obtained using the adjacent acquisition protocol: adjacent electrodes for the injections and adjacent electrodes for the measurements. When a pair of electrodes is chosen as driving pair, it is usually excluded from the available measuring pairs. Each measurement of $V \in \mathbb{R}^{n_M}$, with $n_M = n \times (n - 3)$, is characterized by the pair (I_k, V_ℓ) , $k = 1, \dots, n, \ell = 1, \dots, n - 3$, where I_k indicates the current injection at a couple of electrodes (E_k, E_{k+1}) , and $V_\ell \in V$ represents the potential measure between the electrodes $(E_\ell, E_{\ell+1})$. As the measured data is unavoidably noisy, we assume the following noisy forward observation model

$$F(\sigma; I_F) = V + \eta, \quad (6.2)$$

where $\eta \in \mathbb{R}^{n_M}$ is a vector $\eta \sim \mathcal{N}(0, \bar{n}\bar{V})$ of Gaussian distributed measurements characterized by zero-mean, with \bar{V} the voltage average value.

6.2 EIT-Limited Electrodes Problem Setup

Let $\sigma_L \in \mathbb{R}^{n_T}$ be the conductivity obtained by an EIT setup with n_L electrodes and stimulation pattern (I_F^L, V^L) . A more accurate reconstruction $\sigma_H \in \mathbb{R}^{n_T}$ is expected to be obtained with n_H electrodes, and stimulation pattern (I_F^H, V^H) , with $n_H > n_L$, and, consequently, a larger number of measurements $|V^H| \gg |V^L|$.

Limited Electrodes Model (LEM)

Under the assumption of gradient-sparsity of $\sigma \in \mathbb{R}^{n_T}$, given a limited number of measurements n_M^L collected from a low-resource EIT configuration (I_F^L, V^L) with n_L electrodes, the objective of a Limited Electrode Model is to recover a reconstruction $\sigma^ \in \mathbb{R}^{n_T}$ from $V^L \in \mathbb{R}^{n_M^L}$ measures even if $n_M^L \ll n_T$, with an error bounded by:*

$$\|\sigma^* - \sigma\| \leq C(\|\eta\|, \|\nabla\sigma\|_0, n_T, n_M^L), \quad (6.3)$$

where $C > 0$ is a small constant, and the ℓ_0 pseudonorm $\|x\|_0$ measures the gradient sparsity counting the nonzero value in x .

The LEM will be considered successful if it reconstructs σ^* of better quality with respect to σ_L obtained by solving the inverse optimization problem (6.1), from measurements $V = V^L$ and injections $I_F = I_F^L$. Furthermore, the LEM will be considered optimal, in case it reconstructs a distribution σ^* close to σ_H .

We finally remark that, due to the ill-posedness of the EIT inverse problem, the straightforward interpolation of n_M^L on the boundary $\partial\Omega$ to obtain interpolated n_M^H measurements for solving (6.1), produces unacceptable reconstructions.

6.3 A Compressed Sensing approach to EIT-LE problem

Compressed sensing allow signals to be sampled far below the rate traditionally prescribed [CRT06]. Most of the theory developed for compressed sensing signal recovery assumes that samples are taken using linear measurements in a well-posed context. In this work, we instead address the compressed sensing recovery problem in a setting where the observations are non-linear and we are concerned with sparse recovery principles for non-linear ill-posed inverse problems.

In the linear case, CS theory relies on the sufficient restricted isometry property (RIP) of the linear measurement operator, that can be interpreted in terms of the Lipschitz property of the operator itself and its inverse. The classical CS theory has been extended in [AS21] to arbitrary linear operators with bounded inverse considering in particular the linearized EIT problem. In [AS21] the authors show that the electrical conductivity maybe stably recovered from a number of linearized EIT measurements proportional to the sparsity of the signal with respect to a wavelet

basis, up to a log factor. Preliminary theoretical results in [Har19, AS19, Blu13] paved the way for addressing EIT using CS strategies even in the more complex non-linear setup.

In this section we address a preliminary proposal in this direction. In addition we show experimentally that the number of measurements n_M needed for an optimal recovery are proportional to the sparsity of the unknown, with respect to the gradient of the conductivity distribution.

Let $\sigma \in \mathbb{R}^{n_T}$ be defined in $\Omega_h \subset \mathbb{R}^2$. The data of the reconstruction problem are represented in spatial domain by the potential measurements V^L obtained by a limited EIT configuration with n_L electrodes. Let n_M^L be the number of measurements using an n_L -electrodes configuration, and n_M^H be the ones obtained by the n_H -electrodes configuration, with $n_M^L \ll n_M^H$. Considering $\mathcal{S} \in \mathbb{R}^{n_M^L \times n_M^H}$ to be an acquisition-dependent projection matrix which allows to select a restricted number of potential measurements, then the compressed measurement model, named in this context *LE acquisition model* $F_{\mathcal{S}} : \mathbb{R}^{n_M^H} \rightarrow \mathbb{R}^{n_M^L}$, reads as

$$F_{\mathcal{S}}(\sigma; I_F^H) = V^L, \quad (6.4)$$

with $F_{\mathcal{S}}(\sigma; I_F^H) = \mathcal{S}(F(\sigma; I_F^H))$. In a more realistic scenario in which the measurements $V^L \in \mathbb{R}^{n_M^L}$ are corrupted by noise according to (6.2), the CS reconstruction problem can be stated in this case as the following constrained minimization:

$$\sigma^* \in \arg \min_{\sigma \in \mathbb{R}^{n_T}} R(\sigma) \quad \text{subject to} \quad \|\mathcal{S}(F(\sigma; I_F^H)) - V^L\|_2^2 \leq \eta^2, \quad (6.5)$$

where $R(\sigma)$ is a sparsifying function. In the present context, we have assumed that the conductivity distribution, reconstructed from each set of measurements, varies only in spatially localized small subregions of Ω , which consequently leads to imposing the sparsity of the gradient of the distribution. Therefore our reconstruction method chooses the convex, sparsity inducing function as $R(\sigma) := \|\nabla\sigma\|_1$, namely, it minimizes the total variation of the conductivity distribution. Sparse reconstruction can be performed by converting problem (6.5) into an unconstrained minimization problem and then solving it by applying the Regularized Gauss-Newton (RGN) method illustrated in Section 1.4.2 with residual $r(\sigma) = \mathcal{S}(F(\sigma; I_F^H)) - V^L$. The matrix \mathcal{S} has a block rectangular structure with $n_L \times n_H$ blocks, each block is either an Identity matrix of dimension $n_L - 3$, in case the injection is part of the acquired measurements, either a zero block $\mathbf{0}_{n_L-3}$, otherwise. The position of the nonzero blocks in a row depends on the selected acquisition protocol.

The relation between the sparsity factor $\|\nabla\sigma\|_0$ and the number of non-linear measurements n_M , a well-known result for linear CS, will be the subject of future investigation. By the way of illustration, we plot in Fig. 6.1, for four conductivity samples, the relative reconstruction errors $err := \|\sigma - \sigma^*\|/\|\sigma\|$, as the number of electrodes n (and thus the number of measurements n_M) varies. Each sample is representative of a different gradient-sparsity level, characterized by $\|\nabla\sigma\|_0 = \{10, 21, 29, 44\}$. Using $n = \{4, 6, 8, 12, 24\}$ electrodes, then the corresponding measurements are $n_M = \{4, 18, 40, 108, 504\}$, respectively. The horizontal blue line indicates the error threshold below which the reconstruction can be considered exact (corresponding to a PSNR=70). As expected, as the gradient-sparsity decreases, which corresponds to increasing values of $\|\nabla\sigma\|_0$, the number of measurements necessary to obtain a perfect reconstruction increases. It is also important to note that, after a certain threshold, the reconstruction improvement in the face of an increase the number of measurements is no longer significant.

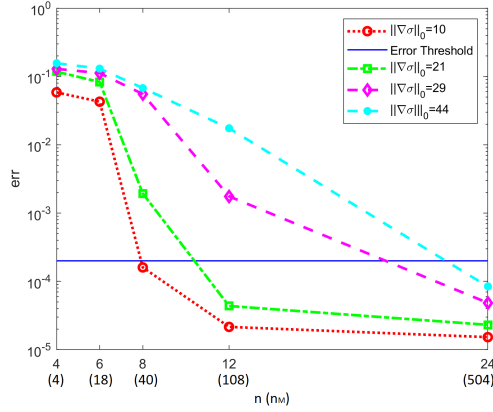


Figure 6.1: Plot of the error reconstruction for an increasing number of electrodes n (and measurements n_M) for four samples σ with different gradient-sparsity.

6.4 A Learned Approach to EIT-LE problem

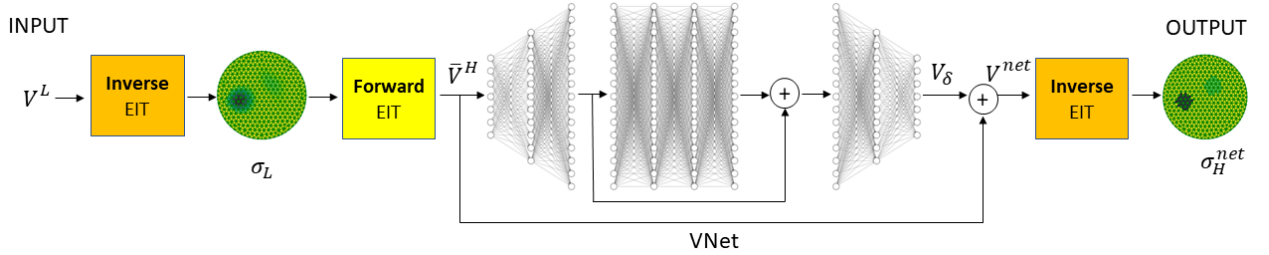


Figure 6.2: Learned LEM

A schematic of the EIT-LE framework is shown in Fig. 6.2. Given the voltage measurements V^L collected using n_L electrodes according to the stimulation pattern (I_F^L, V^L) , the reconstructed σ_L is computed by solving (6.1) using RGN-TV or RGN-TIK algorithm. Then by applying the forward EIT process $F(\sigma_L; I_F^H)$, with I_F^H virtual current injections, we get n_M^H virtual measurements \bar{V}^H . These measurements differ from V^H obtained by an EIT configuration (I_F^H, V^H) with n_H actual electrodes settings. We propose a residual network named VNet which learns a correction factor V_δ to recover a set of measurements V^{net} :

$$V^{net} = V_\delta + \bar{V}^H. \quad (6.6)$$

This residual strategy estimates the quantization noise caused by the acquisition using a limited number of measurements. The improved conductivity reconstruction σ_H^{net} is finally computed by solving the inverse problem (6.1).

6.4.1 VNet architecture

VNet is a deep fully-connected neural network (FCNN) built to provide an ad hoc efficient solution to the EIT-LE problem. In particular, VNet is an autoencoder network with an encoder stack to

learn high-level features, a latent space which is composed of the higher dimensional hidden layers, and a decoder module to reconstruct high-resolution voltage measurements.

Following a supervised learning-based approach, we denominate f_{Θ} , with parameters Θ , a predefined function that, given the input measurements \bar{V}^H , computes the updated measurement vector V^{net} , according to the proposed architecture VNet. We aim at learning the optimal vector Θ by minimizing a loss function \mathcal{L} on a training data set containing N measures-reconstructions pairs (V_i^H, σ_H^i) , $i = 1, \dots, N$, where V_i^H are the potential differences obtained by a configuration with n_H electrodes. In particular, the learning-based method is modeled as the following optimization problem

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\sigma_i^*, \sigma_H^i) \quad \text{subject to} \quad \sigma_i^* = f_{\Theta}(V_i^H),$$

$$\mathcal{L}(\sigma_i^*, \sigma_H^i) = \|V_i^{net}(\Theta) - V_i^H\|_2^2 + \|\sigma_i^* - \sigma_H^i\|_2^2.$$

The considered autoencoder is composed of fully connected layers, dropout layers to improve generalization capabilities, and residuals to both avoid the disappearing gradients effect and allow the learning of the correction term. In a standard encoder-decoder structure, the number of nodes per layer decreases with each subsequent layer of the encoder, and increases back in the decoder. As a result of extensive numerical experimentations, we decided instead to increase the number of neurons in each hidden layer in the encoder, and compress it in the decoder. The final hyperparameters (neurons and layers) configuration is tuned by optimizing the model capacity via the pruning technique: the neurons which have no impact on the performance of the network are trimmed during the training. The encoder module consists of three layers, initialized with n_M^H values, which learn an array of upsampling weights by increasing them up to a certain number m . Then the latent space module has four layers each of the same size and finally, the decoder has three layers with the last of dimension down to n_M^H . Each layer i is a standard linear operator, with weight coefficients $\theta_{i,j}$, which transforms the input features x_j into output features y_i , and reads as

$$y_i = \phi\left(\sum_{j=1}^m \theta_{i,j} x_j + b_i\right), \quad \phi(z) = PReLU(z) := \begin{cases} z, & \text{if } z \geq 0 \\ \beta z, & \text{otherwise} \end{cases}$$

where ϕ is the Parametric Rectified Liner Unit (PReLU) activation function for each layer, with learnable parameter β , except for the last layer where ϕ is the Hyperbolic Tangent (Tanh) function. Since PReLU has a learnable coefficient for the negative part of features, it can avoid the “dead features” caused by zero gradients in ReLU. The Tanh activation function is used to limit the values in $V_{\delta} \in [-1, 1]$, and improve the stability of training. Since the correction factor for each voltage is expected to be very small, it is important to have an activation function that is continuous and differentiable near the origin.

A dropout operator with probability $p = 0.2$ is inserted in the latent block to randomly zero some of the elements of the input layer using samples from a Bernoulli distribution. The idea is that instead of letting layers learn the underlying mapping, we let the network fit the residual mapping.







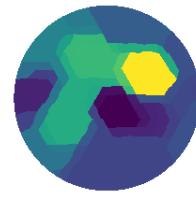

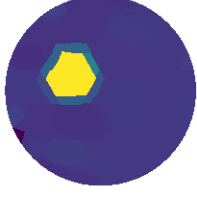
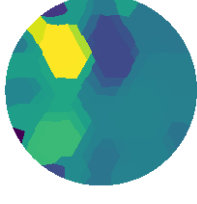
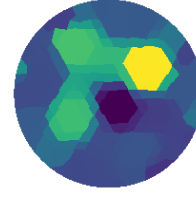

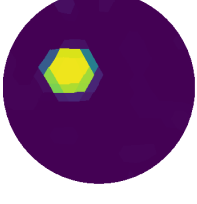
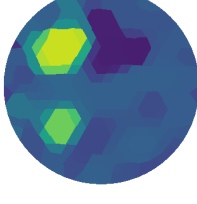
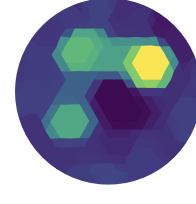

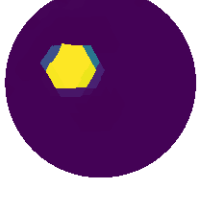



σ_{GT}	 GT 591	 GT 596	 GT 598	 GT 599
σ_8	 PSNR = 27.70	 PSNR = 18.28	 PSNR = 20.16	 PSNR = 26.57
σ_V	 PSNR = 30.39	 PSNR = 19.27	 PSNR = 21.00	 PSNR = 27.57
σ_{16}^{net}	 PSNR = 30.91	 PSNR = 22.35	 PSNR = 21.91	 PSNR = 28.12
σ_{16}	 PSNR = 31.75	 PSNR = 26.81	 PSNR = 24.59	 PSNR = 29.55

Figure 6.3: Example 1: (first row) σ_{GT} , samples 591, 596, 598, 599 of the Data-PC dataset; (second row) reconstructions using RGN-TV from V^8 measurements; (third row) results of CS-based LEM; (fourth row) results of learned LEM; (fifth row) reconstruction using RGN-TV from V^{16} measurements.

	PSNR	SSIM		PSNR	SSIM
σ_6	20.30	0.694	σ_8	22.07	0.741
σ_{12}^{net}	21.05	0.676	σ_{16}^{net}	23.16	0.765
σ_{24}	26.10	0.836	σ_{16}	25.25	0.848
(a)			(b)		
	PSNR	SSIM		PSNR	SSIM
σ_6	24.35	0.668	σ_{12}	27.81	0.763
σ_{12}^{net}	25.16	0.697	σ_{24}^{net}	28.90	0.777
σ_{12}	27.81	0.763	σ_{24}	32.27	0.890
(c)			(d)		

Table 6.1: Average results on the entire testing set: (a)-(b) Data-PC; (c)-(d) Data-S.

6.5 Numerical Experiments

The following numerical experiments show the performance of the proposed EIT-LE frameworks for computing gradient-sparse reconstructions of synthetically built 2D EIT samples. All the examples simulate a circular tank slice of unitary radius with a boundary ring with n equally-spaced electrodes. We built two different datasets, the first, named Data-PC, with a few separated anomalies characterized by constant conductivity, and a second dataset, named Data-S, with much smooth anomalies, occasionally overlapped.

In Data-PC each ground-truth sample σ_{GT} has a random number from 1 to 4 of anomalies localized randomly inside the domain with random radius in $[0.15, 0.25]$ and magnitude in $[0.2, 2]\Omega m^{-1}$. This dataset, being characterized by a strong gradient sparsity, lends itself to being almost perfectly reconstructed by a few measurements. In Data-S each sample σ_{GT} has a random number between 10 and 20 anomalies localized randomly inside the domain, eventually overlapped, radius in $[0.10, 0.15]$ and magnitude in $[1, 2]$. This dataset presents a lower but still well-pronounced level of sparsity with respect to Data-PC, hence more measurements will be required to reach an acceptable reconstruction accuracy.

For both datasets, the homogeneous conductivity of the background liquid is set to be $\sigma_0 = 1.0\Omega m^{-1}$. The regularization parameter λ in (RGN-Tik) and (RGN-TV) has been hand-tuned to obtain the best approximation to the solution of the ill-posed inverse reconstruction problem.

Example 1: CS-based Variational LEM vs Learned LEM Network

Given a set of $n_M^L = 40$ voltage measurements (V^L, I_F^L) , with $n_L = 8$ electrodes, we compare the reconstructions obtained by solving the CS-based model (6.5), with those computed by the learned LEM, using RGN-TV for the inverse EIT, on the Data-PC training set of 500 samples, a validation set with 100 samples, and a test set with 10 samples. Fig. 6.3 illustrates the reconstructions starting from noise-free measurements (in the first two columns) and two reconstructions obtained from noisy measurements according to the degradation model (6.2) with $\eta = 5 \times 10^{-3} \bar{V} rand(n_M)$, (in the third and fourth columns), corresponding to SNR=46dB of the measurements. The higher the gradient sparsity in the conductivity distribution, the more significant the gain in reconstruction accuracy, measured with the Peak SNR metric. Results from the learned LEM show a better

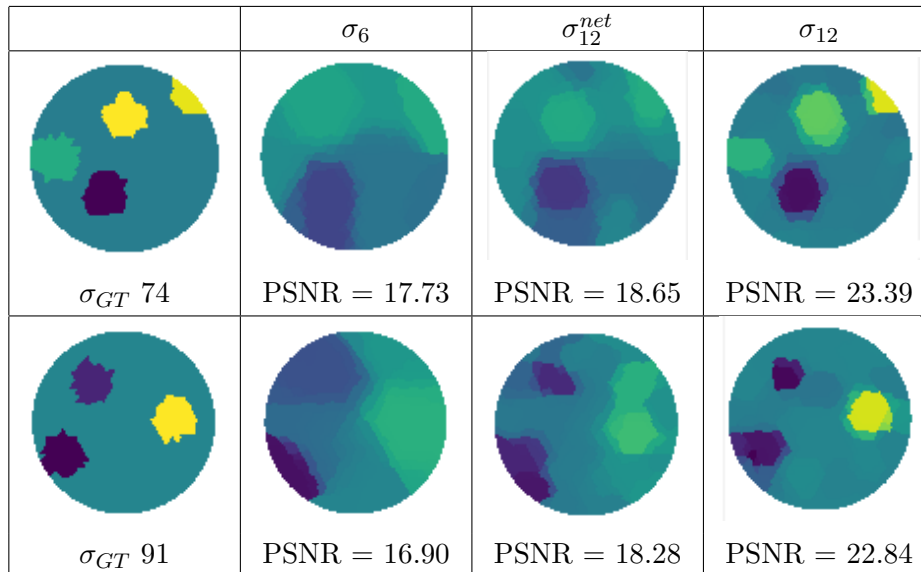


Figure 6.4: Example 2: Reconstructions of samples 74 and 91 from Data-PC dataset, obtained using n_M^6 measurements (second column), the Learned LEM framework (third column), and n_M^{12} measurements (fourth column)

separability of the anomalies and less artifacts with respect to the ones from the CS-based LEM. Despite the ill-posedness of the involved EIT inverse problems, both approaches allow to reconstruct the conductivity distributions from very few measurements under gradient-sparsity conditions, even in presence of noise.

Example 2: Learned LEM applied to Data-PC

In this example we consider the application of the learned LEM to the Data-PC dataset, using RGN-TV algorithm to solve (6.1). We tested this approach in two different settings: using $n_L = 6$ and $n_L = 8$ electrodes to obtain results qualitatively comparable with those obtained by $n_H = 12$ and $n_H = 16$ electrodes, respectively. In both cases the initial measurements were perturbed to obtain an average SNR of 46 dB. Fig. 6.4 illustrates, for two different samples of the dataset, the reconstructions σ_{12}^{net} , results of the learned LEM framework, compared with σ_6 (second column), obtained by a voltage measurement (I_F^6, V^6), and σ_{12} (fourth column), obtained by (I_F^{12}, V^{12}). The corresponding average metrics over the testing dataset are summarized in Table 6.1 (a). A similar comparison has been performed for reconstructions σ_{16}^{net} compared to those obtained by (I_F^8, V^8) and by (I_F^{16}, V^{16}), and the corresponding average results are summarized in Table 6.1 (b).

Example 3: Learned EIT-LE applied to Data-S

We applied the learned LEM to the Data-S dataset, using RGN-Tik algorithm to solve (6.1). The complexity of the dataset Data-S demands more measurements n_M to obtain acceptable results. We tested two settings: using $n_L = 6$ electrodes (corresponding to $n_M^6 = 18$ measurements) and $n_L = 12$ electrodes ($n_M^{12} = 108$ measurements) to obtain results qualitatively comparable with those obtained by $n_H = 12$ and $n_H = 24$ electrodes, respectively. In both cases the initial measurements were perturbed to obtain an average SNR of 46dB.

In Fig. 6.5 the reconstructions for two different samples of the dataset Data-S are shown: the reconstructions σ_{12}^{net} , results of the learned EIT-LE framework, compared with σ_6 , obtained by a

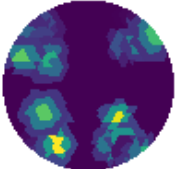
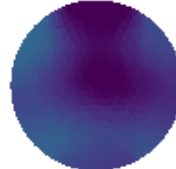
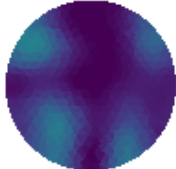
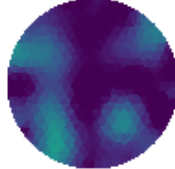
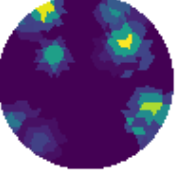

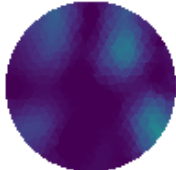
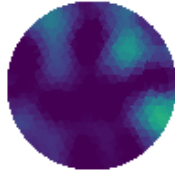
σ_{GT}	σ_6	σ_{12}^{net}	σ_{12}
 GT 54	 PSNR = 22.21	 PSNR = 23.27	 PSNR = 26.11
 GT 89	 PSNR = 22.50	 PSNR = 24.12	 PSNR = 31.92

Figure 6.5: Example 3: Reconstructions of samples 54 and 89 in the Data-S dataset obtained using n_M^6 measurements (second column), the Learned LEM (third column), and n_M^{12} measurements (fourth column).

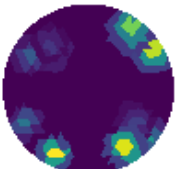
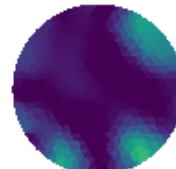
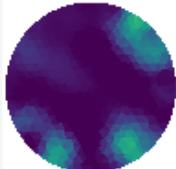
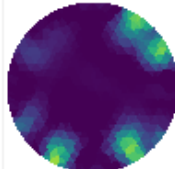
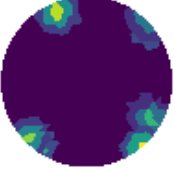
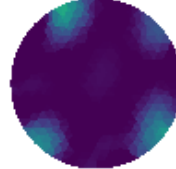
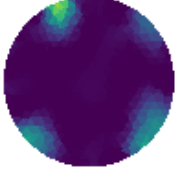
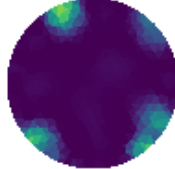
σ_{GT}	σ_{12}	σ_{24}^{net}	σ_{24}
 GT 57	 PSNR = 26.75	 PSNR = 27.40	 PSNR = 32.51
 GT 78	 PSNR = 30.23	 PSNR = 31.61	 PSNR = 34.69

Figure 6.6: Example 3: Reconstructions of samples 57 and 78 in Data-S dataset obtained using n_M^{12} measurements (second column), the Learned LEM framework (third column), and n_M^{24} measurements (fourth column).

voltage measurement (I_M^6, V^6) and σ_{12} , obtained by (I_M^{12}, V^{12}) , in the second and fourth columns, respectively. The corresponding average metrics PSNR and SSIM over the testing dataset are summarized in Table 6.1 (c). In a similar settings, Fig. 6.6 compares the reconstructions for σ_{12} , σ_{24}^{net} , and σ_{24} , and Table 6.1 (d) summarizes the associated average PSNR and SSIM values.

Chapter 7

Post-Processing Transformer Network for CT inverse problem

Transformers have achieved great success in many artificial intelligence fields, such as natural language processing, computer vision, audio and video processing. Scientists proposed a great variety of Transformer variants, in [KNH⁺22] and [TDBM22] there is a systematic and comprehensive literature review on these architectures. The Transformer was initially introduced as a sequence-to-sequence model [SVL14] for machine translation. The success of Transformers in the NLP domain also inspires computer vision researchers which have shown better model performance than traditional Convolutional Neural Networks (CNNs) in vision applications while requiring significantly fewer parameters and training time. Vision Transformers (ViTs) [DBK⁺20] have been successfully used for image classification, object detection [CMS⁺20], image recognition [TCD⁺21], semantic segmentation [SGLS21] and image super-resolution [YYF⁺20]. Different from convolutional networks, transformers require minimal inductive biases for their design and are naturally suited as set functions. An important feature of these models is their scalability to very high-complexity models and large-scale datasets. As compared to their convolutional and recurrent counterparts in deep learning, Transformers assume minimal prior knowledge about the structure of the problem, Therefore, they are typically pre-trained using pretext tasks on large-scale unlabelled datasets.

Transformer-based network structures are naturally good at capturing long-range dependencies in the data by global self-attention. For this reason, we investigate the performance obtained using a transformer architecture as a post-processing network after the image reconstruction in the 2D Computed Tomography (CT) problem. One of the main problems of CT is in fact the insufficient quality of reconstructed images. Typically, many systems use limitations during the scanning phase in order to speed up the process. Therefore, some errors and artifacts may reduce the quality of reconstructed images. The aim of this research is to develop methods in order to reduce the noise in tomographic images, which are obtained through limited-range angles, exploiting transformer network technology.

7.1 Background on Transformers

Transformer architectures are built on a self-attention mechanism that learns the connections between elements of a sequence.

7.1.1 Self-Attention

The self-attention mechanism models the interactions between all items of a sequence for structured prediction tasks. Let us denote a sequence of n entities (x_1, x_2, \dots, x_n) by $\mathbf{X} \in \mathbb{R}^{n \times d}$, where d is the embedding dimension to represent each entity. The goal of self-attention is to capture the interaction amongst all n entities by encoding each entity in terms of global contextual information. The main components used by the Transformer attention are the following learnable weight matrices: queries $\mathbf{W}^Q \in \mathbb{R}^{d \times d_q}$, keys $\mathbf{W}^K \in \mathbb{R}^{d \times d_k}$ and values $\mathbf{W}^V \in \mathbb{R}^{d \times d_v}$, where $d_q = d_k$. The input sequence \mathbf{X} is first projected onto these weight matrices used in generating different subspace representations $\mathbf{Q} = \mathbf{XW}^Q$, $\mathbf{K} = \mathbf{XW}^K$ and $\mathbf{V} = \mathbf{XW}^V$. The output $\mathbf{Z} \in \mathbb{R}^{n \times d_v}$ of the self attention layer is then given by,

$$\mathbf{Z} = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_q}}\right)\mathbf{V}.$$

For a given entity in the sequence, the self-attention implements the dot-product of the query with all keys. It subsequently divides each result by $\sqrt{d_q}$ and proceeds to normalize it using the softmax operator to get the attention scores. Each entity then becomes the weighted sum of all entities in the sequence, where weights are given by the attention scores. The standard self-attention layer attends to all entities. Self-attention differs from convolution in that the filters used are determined by the input, whereas in convolution they are fixed. Additionally, self-attention is not affected by the order of elements or changes in the number of elements, allowing it to handle irregular inputs easily, unlike convolution which is limited by its requirement for a grid structure.

7.1.2 Masked Self-Attention

The Transformer model, which is designed to predict the next element in a sequence, employs self-attention blocks in its decoder that are masked to prevent the model from taking into account future elements. The masked attention operator extracts localized features by constraining self attention to within the foreground region of the predicted mask for each query, instead of attending to the full feature map. This is done by an element-wise multiplication operation with a mask $\mathbf{M} \in \mathbb{R}^{n \times n}$, where \mathbf{M} is an upper-triangular matrix. The masked self-attention is defined by,

$$\text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_q}} \circ \mathbf{M}\right),$$

where \circ denotes Hadamard product. In summary, when predicting an element in a sequence, the attention scores of future elements are set to zero in the masked self-attention mechanism of the transformer model.

7.1.3 Multi-Head Attention

Having multiple attention heads enables the model to focus on different parts of the sequence in a more intuitive way. Therefore multi-head attention is a module for attention mechanisms that runs through an attention mechanism several times in parallel. In order to this the multi-head attention comprises multiple self-attention blocks h . Each block in the model has its own set of weight matrices that are learned during training: $\{\mathbf{W}^{Q_i}, \mathbf{W}^{K_i}, \mathbf{W}^{V_i}\}$, where $i = 0, \dots, (h-1)$. Given an input \mathbf{X} , the output of the h self-attention blocks in multi-head attention is concatenated into a single matrix $[\mathbf{Z}_0, \mathbf{Z}_1, \dots, \mathbf{Z}_{h-1}] \in \mathbb{R}^{n \times h \cdot d_v}$ and projected onto a weight matrix $\mathbf{X} \in \mathbb{R}^{h \cdot d_v \times d}$.

7.2 The Computed Tomography problem

The objective of computed tomography is to reconstruct two-dimensional (2D) or three-dimensional (3D) images of internal structures from collected signals through an object. In x -ray CT, a reconstructed image represents a distribution of radiation-ray linear attenuation coefficients. Data recorded on an x -ray detector array are actually x -ray intensity values after an x -ray beam penetrates an object. The attenuation of the x -ray intensity follows the Lambert–Beer’s law. After applying the negative logarithmic operation of the ratio between the output x -ray intensity and the input x -ray intensity, we obtain the line integral of the attenuation coefficient distribution along an x -ray path. The presentation of line integrals is typically associated with x -ray projections. Projection data are the immediate input to an image reconstruction algorithm. If we denote the attenuation (units: reciprocal length) by x , the measured data by y and we let l denote a line from the source to the detector then they are related through

$$y(l) = \int_l x ds \quad (7.1)$$

Mathematically, CT image reconstruction is a linear inverse problem where (7.1) is typically simplified by gathering the physics into a forward operator $F : X \rightarrow Y$, allowing us to rewrite the above equation as:

$$y = Fx \quad (7.2)$$

where the forward operator in (7.1) is called the Radon transform in 2D and the ray transform in 3D. In some applications, CT data cannot be acquired over the full angular range which is known as limited-angle CT. As consequence, the projection data is incomplete which results in intensity inhomogeneities as well as streaking artifacts in the image domain. Further sources for streaking artifacts are the non-linear attenuation of polychromatic x -rays or inelastic scattering of photons. One of the widely used methods in tomographic image reconstruction is Filtered Back Projection (FBP) algorithm. [FDK84].

7.2.1 Motivation

FBP type of method does not handle the noise statistics of measured data optimally. They regularise by recovering the bandlimited part of the image, an approach that is not sufficient to suppress noise and artifacts that degrade image quality in low-dose CT. Hence, the FBP-type of methods

renders images in low-dose CT that are sub-optimal for diagnostic interpretation. To overcome this issue, we investigated the use of transformer neural networks, called Vformer, as a post-processing image denoising technique. The goal of post-processing methods is to improve a pre-computed reconstruction. The framework consists to incorporate knowledge of the forward operator F into the reconstruction network by applying a pseudo-inverse of F , denoted by \tilde{F}^{-1} , to first map the measurements back to the image domain and then apply the Vformer net to remove artifacts from the resulting images. In our case $\tilde{F}^{-1} : Y \rightarrow X$ is given by the FBP algorithm. This approach structures the reconstruction map $\tilde{F}_\theta^{-1} : Y \rightarrow X$ as:

$$\tilde{F}_\theta^{-1}(y) = g_\theta(\tilde{F}^{-1}y)$$

where $g_\theta : X \rightarrow X$ is the trained Vformer net depending on parameters θ and it is interpreted as predicting the residual between the approximate inverse and the reconstructed image. This strategy is relatively easy to implement, given that the pseudo-inverse can be applied off-line, before the learning is performed, which reduces the learning to inferring an $X \rightarrow X$ transformation. The overall pipeline is depicted in Fig. 7.1.

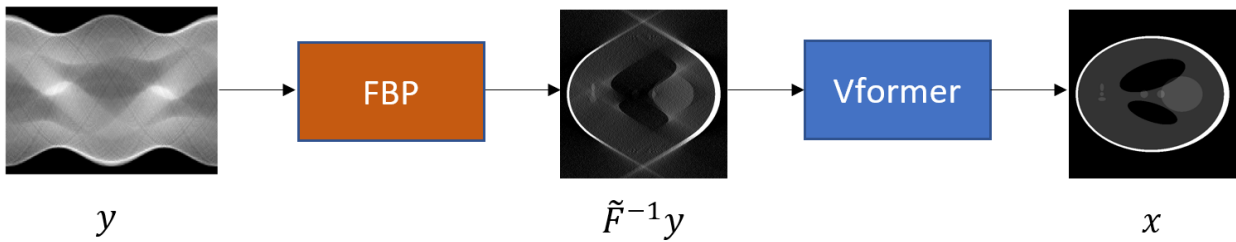


Figure 7.1: Vformer framework.

7.3 Implementation details and Results

7.3.1 Vformer Architecture

Inspired by the state of the art for image classification task performed by Vit net in [DBK⁺20] and a novel denoising transformer network Uformer explained in [WCB⁺22], the proposed Vformer net is a combination of the two previous networks. Given the 2D input image $x \in \mathbb{R}^{H \times W \times C}$, where (H, W) is the image resolution and C is the number of channels, we reshape it into a sequence of flattened 2D patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, (P, P) is the resolution of each image patch, and $N = HW = P^2$ is the resulting number of patches. Defined as D the latent vector size as the input transformer, each patch is flattened and mapped to the D dimensions with a trainable linear projection. Then position embeddings are included with the patch embeddings to maintain information about the position of the patches. The transformer block, depicted in Fig.7.2, consists of an alternating layer of multiheaded self attention and LeFF module. Layer normalization is applied before each block and residual connections are added after every block. In order to overcome the problem connected to the limited capability to leverage local context, we replace the multi-layer-perceptron of Vit with the locally-enhanced feed forward (LeFF) block inspired by [WCB⁺22]. This module is composed

by a linear projection layer on each token to expand its feature dimension. The tokens are then reshaped into 2D feature maps and a 3×3 depth-wise convolution is applied to capture local information. The features are then converted back into tokens through flattening and the number of channels is reduced through another linear layer to match the input channels. After each linear or convolution layer is applied GeLU activation function. After the transformer block, we reshape the flattened feature to the original 2D input image size using einops [Rog22] rearrange function for reordering multidimensional tensors alternating by linear sequential layer. A depth stage of 3×3 2D convolutional layer is added to smooth the image on the boundary of each patch. Fig. 7.3 shows the effects of the convolutional layer on the sheep-logan sample at the same iteration during the training. At these level the post-processing net is ViT without the classification block.

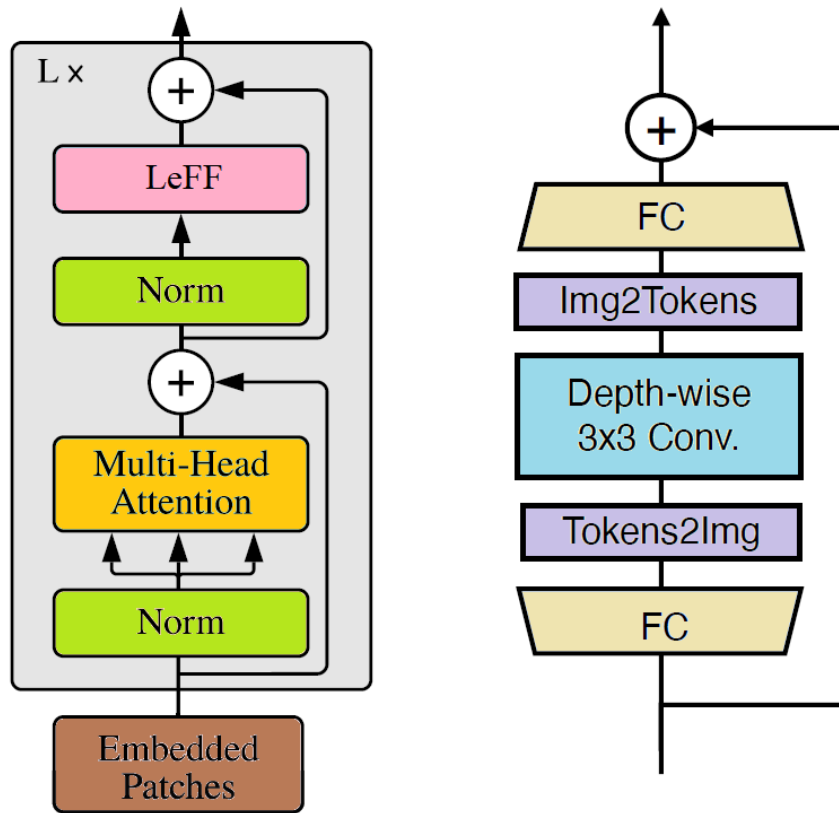


Figure 7.2: Transformer block (left). LeFF module (right).

7.3.2 Training

The data generation is done using the ODL [AKÖ17] framework with ASTRA [vAPC⁺16] as backend for computing the ray-transform and back-projections. The training and validation sets are generated "on the fly", giving an effectively infinite dataset. The labeled dataset is randomly generated ellipses on a 512×512 pixel domain. The projection geometry is selected as a sparse 30 view parallel beam geometry adding to the projection 0.1% additive Gaussian noise. Each iteration of the training scans 96000 images with a batch size of 10 samples. The training is performed by setting the patch size 16 in Vformer net. The evaluation is computed after 2400 iterations on 1280

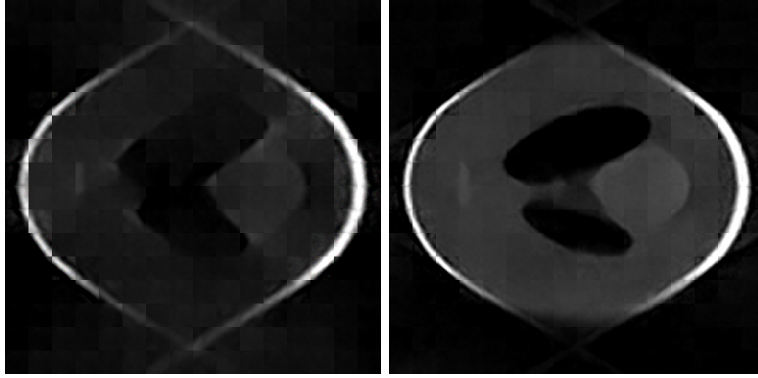


Figure 7.3: Influence of the last convolutional layer: Vit without convolutional tail block (left), Vit with convolutional tail block (right). At this level the Leff block is not implemented.

random ellipses samples. In addition, we tested the model on the Sheep Logan phantom. Vformer net tuning the unknown parameter Θ by minimizing the Charbonnier loss defined as:

$$l(I', \bar{I}) = \sqrt{\|I' - \bar{I}\|^2 + \epsilon^2},$$

where \bar{I} is the ground-truth image and $\epsilon = 10^{-3}$. The selected optimizer is AdamW with learning rate $\gamma = 1e - 4$ and wight decay $\lambda = 0.02$. The method described above is implemented in Python using Pytorch.

7.3.3 Numerical Results

We evaluate the performance on the validation set measuring the PSNR value. We compare our results with state of the art of denoising network Unet [JMFU17], Uformer [WCB⁺22] and ResVformer which is the proposed Vformer with the learning of the residual image R , then the restored image is obtained by $I' = I + R$. Fig. 7.4 shows the results of the Sheep Logan test image. We note that the FBP algorithm performs very poorly under noise assumption while the post-processing learned methods give comparable results. Vformer outperforms the other competitive networks: it removes streak artifacts that typically arise from limited angle tomography data and provides a good shape reconstruction of the ellipses especially around the edges of the phantom and structures spuriously created from noise. From a visual inspection, we can see that Vformer is capable to remove the cross in top/bottom center of FBP image with respect to CNN network. The main reason is that the transformer is a useful tool for capturing global information by utilizing the long range dependencies of pixels.

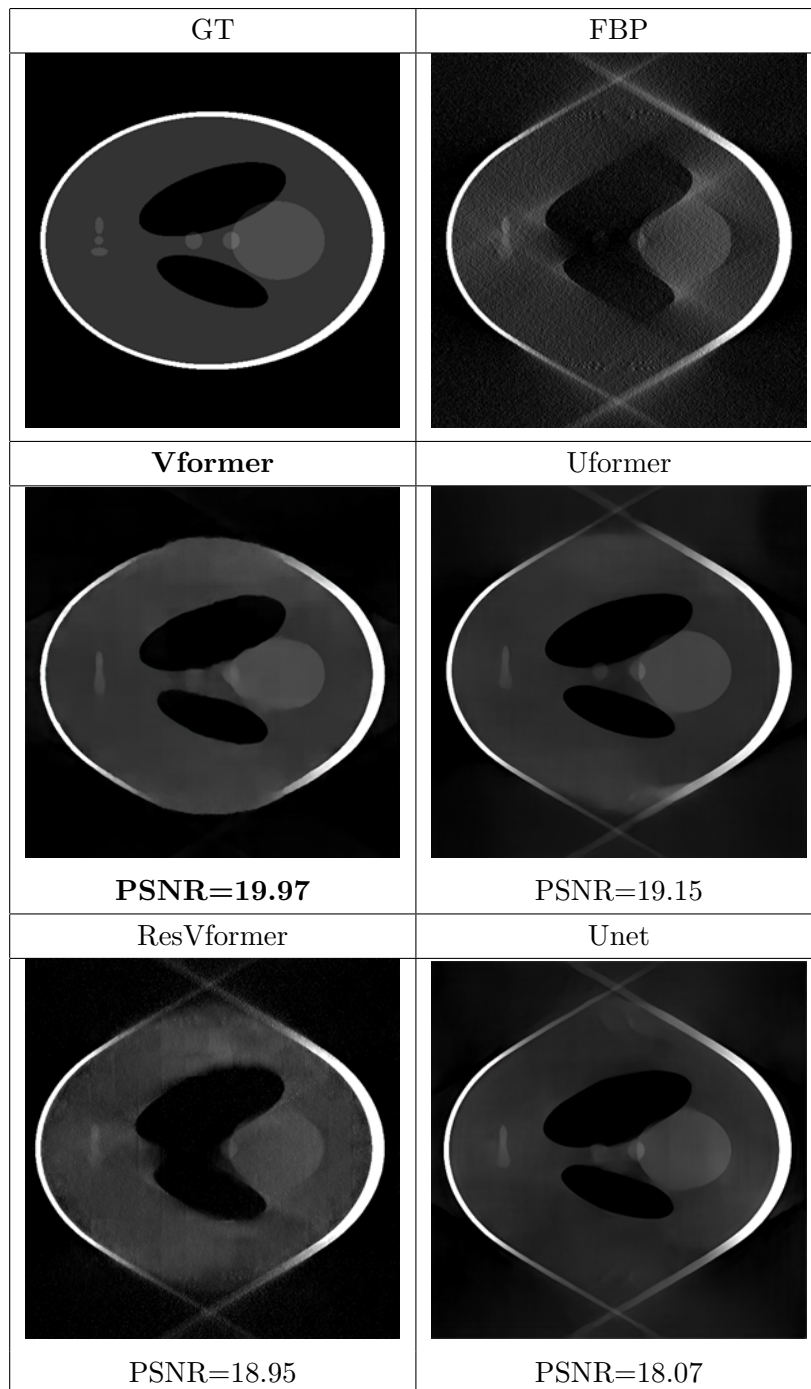


Figure 7.4: Sheep Logan Reconstruction. Comparison between Vformer, Uformer, ResVformer and Unet denoising network.

Conclusions

This Ph.D. thesis has addressed the image ill-posed inverse problem reconstruction by using hybrid iterative methods. In the first part of the thesis, a general overview of inverse problems and numerical and learning approaches has been shown: optimization methods with particular focus on regularization formulation for linear and non-linear IP are described. A data driven classification and survey are reported for the solution of IPs. The second part of the thesis is devoted to investigating the advantages of merging both variational and deep learning frameworks with practical applications in EIT and CT.

In Chapter 4 according to the learning regularizer strategy, we have proposed a hybrid model-based and data-based method for solving non-linear EIT inverse problems which exploits a fully connected neural network to learn the regularizer of the Gauss-Newton method from data. Moreover, we proposed an attention-like mechanism, based on anisotropic total variation, which efficacy in piece-wise conductivity reconstructions is demonstrated in the numerical section. This approach was shown to yield higher accuracy with respect to other state-of-the-art approaches for absolute EIT reconstruction (i.e. RGN, D-bar). The extensive testing presented in the dedicated sections supports this conclusion, which was verified in a number of different configurations: (i) different number of anomalies, (ii) noisy conditions, (iii) heart and lung model, potentially useful for respiration monitoring. While manual hyperparameter tuning and regularizer operator choice are often critical to obtain good reconstruction, in EITGN-NET the regularizer operator is intrinsically tuned, as the training procedure automatically optimizes it for the chosen dataset. Overall, EITGN-NET holds great potential for the integration of model and data-based methods for non-linear EIT reconstruction, providing a framework for the automation of lengthy, error-prone steps while maintaining explicit modeling constraints that allow for greater interpretability of the reconstruction with respect to end-to-end approaches, a crucial feature for mission-critical fields such as medical imaging.

In Chapter 5 we addressed the PnP regularization methods. They belong to the class of variational networks, which exploit the knowledge of the forward model combined with a powerful denoiser to obtain data-driven solutions to challenging optimization problems, thus offering implicit regularization and parameter-free solutions. The proposed deep-PnP proximal Gauss-Newton method solves ill-posed inverse non-linear problems on unstructured data, which generalize the uniform grids (images) typically used with state-of-the-art PnP denoisers. We showed the performance of our proposal in the reconstruction of conductivity distributions and solutions to inverse EIT problems. The results outperform those of classical state-of-the-art variational methods. The proposed PnP-PGN benefits from a powerful pre-trained graph network that performs the entire

denoising process on the mesh, thus paving the way for the usage of convolutional network solutions on FEM problems. Future work will address the optimization of the deep denoiser network model for deployment on a resource-constrained EIT embedded system and will further investigate the theoretical analysis of the convergence of the PnP framework.

In Chapter 6 we focused on obtaining accurate EIT reconstruction. It is generally dependent on the amount of data available and therefore on the number of electrodes present in the acquisition system. The increase in the number of these stimulation/acquisition elements is however associated with non-negligible technical difficulties and high costs. This work introduces the EIT-Limited Electrodes Model for the recovery of high-quality reconstructions from a limited number of measurements. Under the assumption of gradient-sparsity, we propose a variational LEM and a neural network-based LEM. Numerical results show a significant improvement in the quality of the reconstructed conductivity distribution both for the simulated datasets Data-PC , and Data-S, which present a different level of gradient-sparsity. This paves the way to LEM approaches which approximate reconstructions of high quality even if starting from a low profile setup with a few electrodes; thus reducing costs and widening the applicability of this imaging technique. The obtained results can be easily applied to any stimulation pattern. Future developments will include a theoretical investigation of the CS non-linear measurements.

In Chapter 7 we proposed Vformer, a neural network for a learned Post-Processing reconstruction algorithm in CT. Vformer is a transformer-based architecture that combines two different neural networks: Vit which is an image classification network and Uformer used for the image restoration task. Given Experimental results show that our proposed Vformer net can outperform the CNN-based method on the sheep-logan test image.

Bibliography

- [AAL⁺07] Juan-Felipe PJ Abascal, Simon R Arridge, William RB Lionheart, Richard H Bayford, and David S Holder. Validation of a finite-element solution for electrical impedance tomography in an anisotropic medium. *Physiological measurement*, 28(7):S129, 2007.
- [AKÖ17] Jonas Adler, Holger Kohr, and Ozan Öktem. Operator discretization library (odl). 2017.
- [AÖ17] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.
- [AÖ18] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
- [AR09] Stephan W Anzengruber and Ronny Ramlau. Morozov’s discrepancy principle for tikhonov-type functionals with nonlinear operators. *Inverse Problems*, 26(2):025001, 2009.
- [AS19] G. S. Alberti and M. Santacesaria. Calderón’s inverse problem with a finite number of measurements. *Forum of Mathematics, Sigma*, 7:e35, 2019.
- [AS21] G. S. Alberti and M. Santacesaria. Infinite dimensional compressed sensing from anisotropic measurements and applications to inverse problems in pde. *Applied and Computational Harmonic Analysis*, 50:105–146, 2021.
- [BB18] Martin Benning and Martin Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.
- [BFO19] Stephen Becker, Jalal Fadili, and Peter Ochs. On quasi-newton forward-backward splitting: Proximal calculus and convergence. *SIAM Journal on Optimization*, 29(4):2445–2481, 2019.
- [BGAL09] Andrea Borsic, Brad M Graham, Andy Adler, and William RB Lionheart. In vivo impedance imaging with total variation regularization. *IEEE transactions on medical imaging*, 29(1):44–54, 2009.
- [Blu13] Thomas Blumensath. Compressed sensing with nonlinear observations and related nonlinear optimization problems. *IEEE Transactions on Information Theory*, 59(6):3466–3474, 2013.

- [BPD18] Ashish Bora, Eric Price, and Alexandros G Dimakis. Ambientgan: Generative models from lossy measurements. In *International conference on learning representations*, 2018.
- [BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.
- [CIN⁺90] M. Cheney, D. Isaacson, J.C. Newell, S. Simske, and J. Goble. Noser: An algorithm for solving the inverse conductivity problem. *Int. J. Imag. Syst. Technol.*, 2:65–75, 1990.
- [CING89] K.-S. Cheng, D. Isaacson, J. C. Newell, and D. G. Gisser. Electrode models for electric current computed tomography. *IEEE Transactions on Biomedical Engineering*, 36(9):918–924, Sep. 1989.
- [CLMS22] Francesco Colibazzi, Damiana Lazzaro, Serena Morigi, and Andrea Samoré. Learning nonlinear electrical impedance tomography. *Journal of Scientific Computing*, 90(1):1–23, 2022.
- [CMS⁺20] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.
- [CRT06] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
- [DBK⁺20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [dMFH00] Jan C de Munck, Theo JC Faes, and Rob M Heethaar. The boundary element method in the forward and inverse problem of electrical impedance tomography. *IEEE transactions on Biomedical Engineering*, 47(6):792–800, 2000.
- [DSHW17] Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.
- [EHN96] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- [EKN89] Heinz W. Engl, Karl Kunisch, and Andreas Neubauer. Convergence rates for tikhonov regularisation of non-linear ill-posed problems. *Inverse Problems*, 5:523–540, 1989.
- [FDK84] Lee A Feldkamp, Lloyd C Davis, and James W Kress. Practical cone-beam algorithm. *Josa a*, 1(6):612–619, 1984.

- [FL19] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*, 2019.
- [GFPC21] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.
- [GHS08] Markus Grasmair, Markus Haltmeier, and Otmar Scherzer. Sparse regularization with lq penalty term. *Inverse Problems*, 24(5):055020, 2008.
- [GJ19] Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pages 2083–2092. PMLR, 2019.
- [GJN⁺18] Harshit Gupta, Kyong Hwan Jin, Ha Q. Nguyen, Michael T. McCann, and Michael Unser. Cnn-based projected gradient descent for consistent ct image reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1440–1453, 2018.
- [GL10] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.
- [GM76] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & mathematics with applications*, 2(1):17–40, 1976.
- [GPAM⁺20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [GY95] Donald Geman and Chengda Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE transactions on Image Processing*, 4(7):932–946, 1995.
- [Had02] Jacques Hadamard. Sur les problèmes aux dérivées partielles et leur signification physique. *Princeton University Bulletin*, 13:49–52, 1902.
- [Han87] Per Christian Hansen. The truncatedsvd as a method for regularization. *BIT Numerical Mathematics*, 27(4):534–553, 1987.
- [Han97] Martin Hanke. A regularizing levenberg-marquardt scheme, with applications to inverse groundwater filtration problems. *Inverse problems*, 13(1):79, 1997.
- [Han98] Per Christian Hansen. *Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion*. SIAM, 1998.
- [Har19] Bastian Harrach. Uniqueness and lipschitz stability in electrical impedance tomography with finitely many electrodes. *Inverse Problems*, 35(2):024005, jan 2019.

- [HCW01] Wang Huaxiang, Wang Chao, and Yin Wuliang. Optimum design of the structure of the electrode for a medical eit system. *Measurement Science and Technology*, 12(8):1020, 2001.
- [HKM⁺17] Andreas Hauptmann, Ville Kolehmainen, Nguyet Minh Mach, Tuomo Savolainen, Aku Seppänen, and Samuli Siltanen. Open 2d electrical impedance tomography data archive. *arXiv preprint arXiv:1704.01178*, 2017.
- [HKM⁺18] Andreas Hauptmann, Ville Kolehmainen, Nguyet Minh Mach, Tuomo Savolainen, Aku Seppänen, and Samuli Siltanen. 2d electrical impedance tomography dataset. 2018.
- [HPZ22] Matthew Holden, Marcelo Pereyra, and Konstantinos C Zygalakis. Bayesian imaging with data-driven priors encoded by neural networks. *SIAM Journal on Imaging Sciences*, 15(2):892–924, 2022.
- [HSW90] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural networks*, 3(5):551–560, 1990.
- [JMFU17] Kyong Hwan Jin, Michael T McCann, Emmanuel Froustey, and Michael Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.
- [JWZ21] Fan Jia, Wing Hong Wong, and Tiejiong Zeng. Ddunet: Dense dense u-net with applications in image denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 354–364, 2021.
- [K⁺11] Andreas Kirsch et al. *An introduction to the mathematical theory of inverse problems*, volume 120. Springer, 2011.
- [KNH⁺22] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.
- [KS06] Jari Kaipio and Erkki Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006.
- [LCM⁺19] Risheng Liu, Shichao Cheng, Long Ma, Xin Fan, and Zhongxuan Luo. Deep proximal unrolling: Algorithmic framework, convergence analysis and applications. *IEEE Transactions on Image Processing*, 28(10):5013–5026, 2019.
- [Lio04] William R.B. Lionheart. EIT reconstruction algorithms: Pitfalls, challenges and recent developments. In *Physiological Measurement*, 2004.
- [LLTMK19] Alice Lucas, Santiago Lopez-Tapia, Rafael Molina, and Aggelos K Katsaggelos. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing*, 28(7):3312–3327, 2019.

- [LMS15] Alessandro Lanza, Serena Morigi, and Fiorella Sgallari. Convex image denoising via non-convex regularization. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 666–677. Springer, 2015.
- [LO09] Yingying Li and Stanley Osher. A new median formula with applications to PDE based denoising. *Communications in Mathematical Sciences*, 7(3):741 – 753, 2009.
- [LÖS18] Sebastian Lunz, Ozan Öktem, and Carola-Bibiane Schönlieb. Adversarial regularizers in inverse problems. *Advances in neural information processing systems*, 31, 2018.
- [LSAH20] Housen Li, Johannes Schwab, Stephan Antholzer, and Markus Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.
- [LSS14] Jason D Lee, Yuekai Sun, and Michael A Saunders. Proximal newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.
- [LTG⁺20] Yuelong Li, Mohammad Tofighi, Junyi Geng, Vishal Monga, and Yonina C Eldar. Efficient and interpretable deep blind image deblurring via algorithm unrolling. *IEEE Transactions on Computational Imaging*, 6:666–681, 2020.
- [LTMK21] Santiago López-Tapia, Rafael Molina, and Aggelos K Katsaggelos. Deep learning approaches to inverse problems in imaging: Past, present and future. *Digital Signal Processing*, 119:103285, 2021.
- [LYX⁺18] Benyuan Liu, Bin Yang, Canhua Xu, Junying Xia, Meng Dai, Zhenyu Ji, Fusheng You, Xiuzhen Dong, Xuetao Shi, and Feng Fu. pyeit: A python based framework for electrical impedance tomography. *SoftwareX*, 7:304–308, 2018.
- [MHÖ⁺23] Subhadip Mukherjee, Andreas Hauptmann, Ozan Öktem, Marcelo Pereyra, and Carola-Bibiane Schönlieb. Learned reconstruction methods with convergence guarantees: A survey of concepts and applications. *IEEE Signal Processing Magazine*, 40(1):164–182, 2023.
- [MMHC17] Tim Meinhardt, Michael Moller, Caner Hazirbas, and Daniel Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [MNMM20] Pedro Henrique Martins, Vlad Niculae, Zita Marinho, and André F. T. Martins. Sparse and structured visual attention. *ArXiv*, abs/2002.05556, 2020.
- [MS12] Jennifer L. Mueller and Samuli Siltanen. *Linear and Nonlinear Inverse Problems with Practical Applications*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2012.

- [OJM⁺20] Gregory Ongie, Ajil Jalal, Christopher A Metzler, Richard G Baraniuk, Alexandros G Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.
- [ÖM19] Andreas Öchsner and Resam Makvandi. *Introduction*, pages 1–2. Springer International Publishing, Cham, 2019.
- [ONSH21] Daniel Obmann, Linh Nguyen, Johannes Schwab, and Markus Haltmeier. Augmented nett regularization of inverse problems. *Journal of Physics Communications*, 5(10):105002, 2021.
- [PGM⁺19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [REM17] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.
- [RLW⁺19] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR, 2019.
- [ROF92] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.
- [Rog22] Alex Rogozhnikov. Einops: Clear and reliable tensor manipulations with einstein-like notation. In *International Conference on Learning Representations*, 2022.
- [SCI92] Erkki Somersalo, Margaret Cheney, and David Isaacson. Existence and uniqueness for electrode models for electric current computed tomography. *SIAM Journal on Applied Mathematics*, 52(4):1023–1040, 1992.
- [SGLS21] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.
- [SLMS20] Ivan Selesnick, Alessandro Lanza, Serena Morigi, and Fiorella Sgallari. Non-convex total variation regularization for convex denoising of signals. *Journal of Mathematical Imaging and Vision*, 62(6):825–841, 2020.
- [SLX⁺16] Jian Sun, Huibin Li, Zongben Xu, et al. Deep admm-net for compressive sensing mri. *Advances in neural information processing systems*, 29, 2016.
- [SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

- [SW12] Jin Keun Seo and Eung Je Woo. *Nonlinear inverse problems in imaging*. John Wiley & Sons, 2012.
- [TA77] A. N. Tikhonov and V. IA. Arsenin. *Solutions of ill-posed problems / Andrey N. Tikhonov and Vasiliy Y. Arsenin ; translation editor, Fritz John*. Winston ; distributed solely by Halsted Press Washington : New York, 1977.
- [TCD⁺21] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.
- [TDBM22] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
- [Tik43] A. N. Tikhonov. On the stability of inverse problems. *Proceedings of the USSR Academy of Sciences*, 39:195–198, 1943.
- [Tik63] Andrei Nikolaevich Tikhonov. On the solution of ill-posed problems and the method of regularization. In *Doklady akademii nauk*, volume 151, pages 501–504. Russian Academy of Sciences, 1963.
- [vAPC⁺16] Wim van Aarle, Willem Jan Palenstijn, Jeroen Cant, Eline Janssens, Folkert Bleichrodt, Andrei Dabrovolski, Jan De Beenhouwer, K. Joost Batenburg, and Jan Sijbers. Fast and flexible x-ray tomography using the astra toolbox. *Opt. Express*, 24(22):25129–25147, Oct 2016.
- [VBW13] Singanallur V Venkatakrisnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.
- [VVK⁺98] M. Vauhkonen, D. Vadasz, P. A. Karjalainen, E. Somersalo, and J. P. Kaipio. Tikhonov regularization and prior information in electrical impedance tomography. *IEEE Transactions on Medical Imaging*, 17(2):285–293, April 1998.
- [WCB⁺22] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022.
- [WK16] Max Welling and Thomas N Kipf. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*, 2016.
- [WYS⁺18] Zeying Wang, Shihong Yue, Kai Song, Xiaoyuan Liu, and Huaxiang Wang. An unsupervised method for evaluating electrical impedance tomography images. *IEEE Transactions on Instrumentation and Measurement*, 67(12):2796–2803, 2018.

- [YYF⁺20] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5791–5800, 2020.
- [ZG18] Jian Zhang and Bernard Ghanem. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1828–1837, 2018.
- [ZLC⁺18] Bo Zhu, Jeremiah Z Liu, Stephen F Cauley, Bruce R Rosen, and Matthew S Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487–492, 2018.
- [ZLL⁺18] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.
- [ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.