ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

DEEP-PLUG-AND-PLAY PROXIMAL GAUSS-NEWTON METHOD WITH APPLICATIONS TO NONLINEAR, ILL-POSED INVERSE PROBLEMS

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Colibazzi, F., Lazzaro, D., Morigi, S., Samoré, A. (2023). DEEP-PLUG-AND-PLAY PROXIMAL GAUSS-NEWTON METHOD WITH APPLICATIONS TO NONLINEAR, ILL-POSED INVERSE PROBLEMS. INVERSE PROBLEMS AND IMAGING, 17(6), 1226-1248 [10.3934/ipi.2023014].

Availability:
This version is available at: https://hdl.handle.net/11585/946154 since: 2026-05-20

Published:
DOI: http://doi.org/10.3934/ipi.2023014

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

03 June 2026

DEEP-PLUG-AND-PLAY PROXIMAL GAUSS-NEWTON METHOD WITH APPLICATIONS TO NONLINEAR, ILL-POSED INVERSE PROBLEMS

Francesco Colibazzi[1], Damiana Lazzaro[1], Serena Morigi[*1] and Andrea Samoré[2]

[1]Dept. of Mathematics, University of Bologna, Bologna, Italy

[2]School of Biomedical Engineering, University of Sydney, Sydney, Australia

(Communicated by Sung Ha Kang)

Abstract. In this paper we propose a proximal Gauss-Newton method for the penalized nonlinear least squares optimization problem arising from regularization of ill-posed nonlinear inverse problems. By exploiting the modular structure that characterizes the proximal-type methods, we plug in a pre-trained graph neural net denoiser in place of the standard proximal map. This allows to mould the prior on the data. An encoder-decoder Graph U-Net architecture is proposed as denoiser, which works on unstructured data; its mathematical formulation is derived to analyse the Liptschitz condition. With the intent of showing the benefits of applying deep Plug-and-Play reconstructions, we consider as an exemplar application, the nonlinear Electrical Impedance Tomography, a promising non-invasive imaging technique mathematically formulated as a highly nonlinear ill-posed inverse problem.

1. **Introduction.** This paper is devoted to the solution of nonlinear ill-posed inverse problems formulated as an operator equation as follows

$$\text{Given} \quad y \in R(F), \quad \text{find} \quad x \quad \text{so that} \quad F(x) = y, \tag{1}$$

where $F : D(F) \subset \mathcal{X} \to \mathcal{Y}$ is a Fréchet differentiable nonlinear operator, mapping the Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$ with domain $D(F)$ and range $R(F)$. The inverse problem (1) is ill-posed in the sense that, due to the nonlinearity of F, the solution need not being unique and can be also very sensitive to small perturbations in the observed data. In many applications, the perturbations are from measurement errors caused by noise, which result in large deviations in the solution. Therefore, in practice, the available data in (1) is only an approximation y^δ to y satisfying

$$\left\| y^\delta - y \right\|_{\mathcal{Y}} \leq \delta$$

where $\delta > 0$ is a given small noise level, and $\|\cdot\|_{\mathcal{Y}}$ is the norm induced by the inner product on $\mathcal{Y}$. Hence, given the noisy data, problem (1) can be formulated as a nonlinear least squares problem which needs to be stabilized. Widely used regularization methods range from the nonlinear Tikhonov regularization [41], the

2020 *Mathematics Subject Classification.* Primary: 65K10; Secondary: 68T20.

Key words and phrases. Ill-posed inverse problems, proximal Gauss-Newton, nonlinear optimization, deep-PnP, graph neural networks.

*Corresponding author: Serena Morigi.

regularized Gauss-Newton (RGN) method, to the Levenberg-Marquardt algorithm, which aims to regularize the ill-conditioned linear system derived from computing the Gauss-Newton search direction.

In this paper, we propose a proximal Gauss-Newton method for the optimization problem arising from regularization of ill-posed nonlinear 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 [32]. The difficulties that we want to face 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, the proximity step with an efficient denoiser algorithm. The term *denoiser* derived from its wide usage in the imaging scientific community, motivated by the connection with the basic proximal minimization algorithm that can be employed for the reconstruction from noisy data. The resulting algorithm is generically called Plug-and-Play (PnP) and the name of the parent proximal optimization method (i.e. PnP- forward-backward (FB) [9], PnP- Alternating Direction Method of Multiplier (ADMM) [37], PnP- fast iterative shrinkage thresholding algorithm (FISTA) [29, 50], PnP- Primal-Dual Hybrid Gradient method (PDHG) [36]). These Plug-and-Play strategies, introduced in [56], have shown competitive results with state-of-the-art methods in image recovery [48, 50, 29]. They take advantages of successful denoiser algorithms such as BM3D [10], WNNM [15], BLS-GSM [40], class-specific Gaussian mixture model (GMM) denoiser [61], and nonlocal means [4], as well as deep learning-based denoisers [60, 59].

Motivated by these recent successes we propose a deep-PnP proximal Gauss-Newton method which incorporates a deep learning denoiser tailored on the data. Unlike traditional model-based methods which need to specify the explicit and handcrafted image priors, our method allows to implicitly define the prior according to the data. Moreover, the proposed deep-PnP denoiser works on unstructured data, thus extending the standard case of data defined on regular grids, which is commonly considered in the imaging contexts. It is based on a graph convolutional network, proposed in [24], 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 nonlinear and ill-posed problem, we focus on the Electrical Impedance Tomography (EIT) inverse problem for the reconstruction of conductivity distributions. EIT is a promising imaging modality by which a body can be monitored using electrodes distributed on its boundary surface. Starting from the measured voltages it is possible to reconstruct the distribution of internal electrical conductivity [6]. As a radiation-free non-invasive monitoring tool, EIT has attracted much attention in the last two decades in a variety of biomedical applications. The forward data generation process is described by a Finite Element Method (FEM) differential model, while the reconstruction process (backward) responds to an inverse, highly ill-posed, nonlinear least squares formulation. Due to the intrinsic ill-posedness of the problem, the EIT represents an open mathematical challenge to new appropriate numerical techniques of optimization and differential

modelling. In addition to the EIT problem analysed, other interesting applications, formulated as non-linear ill-posed inverse problems, can be solved by applying the proposed deep-PnP approach. We mention for this purpose some promising medical imaging modalities such as the Optical Tomography (OT) [52], the Electromagnetic Tomography (EMT) [57], and the Photoacustic Tomography (PAT) [20].
The main contributions of the work are summarized below:

- On the algorithmic side we introduce a regularized proximal Gauss-Newton method with a scaled proximal mapping, which is addressed to the solution of nonlinear 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 nonlinear EIT problems.

The paper is organized as follows. The remaining of this section is devoted to a brief overview of related works on the main ingredients of the proposal: proximal Newton-type algorithms, PnP methods and optimization methods for the EIT problem. In Section 2 we present regularization models to stabilize the solutions of nonlinear ill-posed inverse problems, and in Section 3 proximal methods are proposed for their solution. The deep-PnP denoiser is introduced in Section 4 and formalized and analysed in Section 5. The proposed deep-PnP proximal Gauss-Newton method is described in Section 5.2 and applied in Section 6 to the solution of the inverse EIT problem. Numerical experiments validate the effectiveness and feasibility of the proposed approach in Section 7. Conclusions are drawn in Section 8.

1.1. **Related works.** *Proximal Newton-type methods for ill-posed problems.* The classical Gauss–Newton method, originally proposed for solving nonlinear least squares, can also be used to minimize the sum of an additional term to combine such a least squares fitting with a nonsmooth but convex regularization term. However, such a basic approach suffers from stability problems, especially in ill-posed inverse problems. Other methods have been proposed as extensions of the Gauss–Newton method to the general composite minimization problem by introducing a proximal operator of a nonsmooth penalty function [32]. The basic proposal in [44] is hardly adaptable for managing ill-posed problems since requires inverting potentially ill-conditioned operators. To address ill-posedness, in [12] the authors proposed a variable metric proximal regularized Gauss-Newton which incorporates two regularization terms. Towards an efficient computation, in [38] the authors proposed variable metric proximal method with a diagonal Barzilai-Borwein metric, and in [1] a *diagonal* $\pm$ *rank* r metric is presented which allows for an efficient proximal map evaluation. In [28] the authors proposed a relaxed inexact proximal Gauss–Newton method.

Plug-and-Play strategies using CNN Denoisers. With the development of deep learning techniques, variational networks, such as plug-and-play methods and unrolled network, have become increasingly popular in a broad range of imaging problems. In unrolling (or unfolding) technique each iteration of an optimization algorithm is a layer of the network. Unrolled algorithms have been proposed for the

solution of inverse image processing starting from the most popular optimization algorithms, such as [18] based on gradient descent, [49] based on ADMM, [58] based on ISTA. However, integrating neural networks into overall training process when dealing with severely ill-conditioned inverse problems, does not lead in general to optimal results as the weight updating is extremely sensitive to the inevitable small perturbations on the data in the training phase. A more robust response to ill-conditioning is given by the plug-and-play methods, which use deep learning-based denoisers as regularization priors. In [53, 34], the authors proposed IRCNN denoisers for image inpainting and deblurring. In [54], the IRCNN denoiser is applied for plug-and-play deblurring and Single Image Superresolution. The same imaging problems have been addressed in [42], with the TNRD denoiser, and in [36] with the DnCNN-S denoiser.

Driven by the great empirical success, some authors have begun to explore theoretical aspects of the convergence of these PnP algorithms [51]. In [43], Ryu et al. provided a fixed-point convergence analysis of PnP-ADMM, and proposed the spectral normalization to stabilize training, and in [14] the equivalence of the denoiser to a proximal map of a convex function has been proved but only for a class of linear denoisers [13].

Numerical solutions for the inverse EIT problem. Variational approaches to the solution of the EIT problem are currently the most successful and widely used both for linearized and non-linearized EIT models. Classical variational formulations for linear EIT problems relies on the minimization of the Tikhonov functional when smooth conductivity solutions are expected, [35, 55], or on Total Variation (TV) functional when discontinuities in the reconstructed conductivities have to be preserved [3]. To overcome the inaccuracy of the EIT linearized approach, in [26] the regularization is combined with an optical imaging prior. Precursor of the numerical optimization methods for the direct solution of the nonlinear EIT inverse formulation is the Newton One-Step Algorithm (NOSER) [7]. In the wake of this, nonlinear EIT solutions have been derived by a first-order proximal gradient method [25], and Regularized Gauss-Newton (RGN) method coupled with both smooth and nonsmooth (TV) regularizers [3]. With the rapid development of deep learning, a few neural networks-based EIT reconstructions have been proposed both for structured data in [17], and for unstructerd meshes (Graph CNN) in [23]. In [8] the authors introduced a hybrid model-based and data-based method which exploits a fully connected neural network to learn the regularizer operator of the RGN algorithm from data.

2. Regularizing nonlinear ill-posed inverse problems.

The most popular regularization method for nonlinear least squares problems is Tikhonov regularization, i.e., to seek an approximation x_α^δ to the solution of (1) as a minimizer of the quadratic functional

$$\mathcal{J}(x;\alpha) := \frac{1}{2}\left\|F(x) - y^\delta\right\|_{\mathcal{Y}}^2 + \frac{\alpha}{2}\left\|L(x - x_0)\right\|_{\mathcal{X}}^2, \tag{2}$$

where $\alpha > 0$ is the regularization parameter and L is a linear differential operator. The second term in the functional (2), so-called regularizer, favours minimal norm and smoothness properties of the solution, and the initial guess x_0 in (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 (2) is a global convex function. But this property is lost as soon as the operator is nonlinear. As a consequence, the nonconvex functional (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 (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 [31, 46], we propose to extend the function in (2) by incorporating an eventually non-convex and not necessarily differentiable penalty function $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$, as follows

$$\mathcal{J}(x) := \underbrace{\frac{1}{2}\left\|F(x) - y^{\delta}\right\|_{\mathcal{Y}}^2 + \frac{\alpha}{2}\left\|L(x - x_0)\right\|_{\mathcal{X}}^2}_{f_\alpha(x)} + g(x), \tag{3}$$

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

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

$$\nabla f_\alpha(x) = J^T(x)(F(x) - y^{\delta}) + \alpha L^T L(x - x_0), \tag{4}$$

$$\nabla^2 f_\alpha(x) = \nabla^2 F(x)(F(x) - y^{\delta}) + J^T(x)J(x) + \alpha L^T L, \tag{5}$$

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

3. **Solving nonlinear ill-posed inverse problems by Proximal Newton-type methods.** Our task can be formulated as the following minimization problem

$$\min_{x\in\mathbb{R}^n} \left\{\mathcal{J}(x) := f_\alpha(x) + g(x)\right\}, \tag{6}$$

where $f_\alpha : \mathbb{R}^n \to \mathbb{R}$, defined in (3), is continuously differentiable, and $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a proper, lower semicontinuous, convex but not necessarily differentiable penalty function. 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, $x_2, \ldots$, 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 (6), then it is a stationary point of (6), i.e., one has

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

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

Unlike the Newton's method which approximates $\mathcal{J}$ in (6) with a local quadratic model, proximal methods approximate near a generic iterate x_k only the smooth part f_α with a local quadratic model, thus considering as $\mathcal{J}$ approximant at a given y, the following model

$$\bar{\mathcal{J}}(y;) = f_\alpha(x_k) + \nabla f_\alpha(x_k)^T(y - x_k) + \frac{1}{2}(y - x_k)^T H_k(x_k)(y - x_k) + g(y), \tag{7}$$

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

Proximal Newton-type methods are second-order methods which rely on accurate approximations H_k of the Hessian $\nabla^2 f_\alpha(x_k)$. In particular, the *proximal Newton method* deals directly with $H_k := \nabla^2 f_\alpha(x_k)$, and approximations to $\nabla^2 f_\alpha(x_k)$, 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 3.1. Let $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a proper, lower semicontinuous, convex function and Q be a positive definite matrix. Then the scaled proximal mapping of g at x is

$$\operatorname{prox}_g^Q(x) := \arg\min_{y \in \mathbb{R}^n} \left\{ g(y) + \frac{1}{2}\|y - x\|_Q^2 \right\}, \tag{8}$$

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

As proved in [1], the mapping $\operatorname{prox}_g^Q(x)$ exists and is unique; moreover it is nonexpansive, i.e., for all $x, y \in \mathbb{R}^n$

$$\|\operatorname{prox}_g^Q(y) - \operatorname{prox}_g^Q(x)\|_Q^2 \leq \|y - x\|_Q^2, \tag{9}$$

and can be also implicitly defined as

$$\operatorname{prox}_g^Q(x) = (I + Q^{-1}\partial g)^{-1}(x). \tag{10}$$

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

$$z_{k+1} = x_k - H_k^{-1}\nabla f_\alpha(x_k) \tag{11}$$

$$x_{k+1} = \operatorname{prox}_g^{H_k}(z_{k+1}). \tag{12}$$

In fact, by minimizing the local quadratic model (7) we get

$$x_{k+1} = \arg\min_y \left\{ \nabla f_\alpha(x_k)^T (y - x_k) + \frac{1}{2}(y - x_k)^T H_k (y - x_k) + g(y) \right\} \tag{13}$$

$$= \arg\min_y \left\{ \frac{1}{2}\|\bar{Q}(x_k - H_k^{-1}\nabla f_\alpha(x_k) - y)\|_2^2 + g(y) \right\}, \tag{14}$$

$$= \arg\min_y \left\{ \frac{1}{2}\|x_k - H_k^{-1}\nabla f_\alpha(x_k) - y\|_{H_k}^2 + g(y) \right\}, \tag{15}$$

where in (14) 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 (14) to (15) we apply

$$\|y - x\|_{H_k}^2 = < H_k(y - x), (y - x) > = < \bar{Q}(y - x), \bar{Q}(y - x) > = \|\bar{Q}(y - x)\|_2^2. \tag{16}$$

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.

In the following result, which proof is in Appendix, in order to face the ill-posedness, we characterize the proximal Newton-type method for a specific choice of the matrix H_k, $f_\alpha(x)$ defined as in (3) and $g(x)$ Moreover, we will easily show that the proximal Newton-type method approximates the solution of the nonlinear ill-posed problem (6) by iteratively solving optimization sub-problems where $\tilde{f}_\alpha(x)$ contains a good linear approximation of the nonlinear operator $F(x)$.

Proposition 3.2. *Let f_α be defined in (3), with F be a nonlinear, Frechét differentiable operator, and using for the Hessian in (5) the following approximation*

$$\nabla^2 f_\alpha(x) \approx H = J^T(x)J(x) + \alpha L^T L. \tag{17}$$

The iterative step kth, with $k = 1, 2, \ldots$, *of the proximal Gauss-Newton method* (11)-(12) *reads as*

$$z_{k+1} = x_k - H_k^{-1}(J^T(x_k)(F(x_k) - y^\delta) + \alpha L^T L(x_k - x_0)) \quad (18)$$

$$x_{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|J(x_k)(x - z_{k+1})\|_2^2 + \frac{\alpha}{2}\|L(x - z_{k+1})\|_2^2 + g(x) \right\} \quad (19)$$

which is equivalent to

$$x_{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ \tilde{f}_\alpha(x) + g(x) \right\}, \quad (20)$$

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

A standard choice for the regularization term $g(x)$ in (6), is $g(x) := \|\nabla x\|_1$ which leads to the minimization sub-problem:

$$x_{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ \|\nabla x\|_1 + \frac{1}{2}\|J(x - z_{k+1})\|_2^2 + \frac{\alpha}{2}\|L(x - z_{k+1})\|_2^2 \right\}. \quad (21)$$

An approximate solution of (21) can be obtained by a first-order proximal gradient method, which, starting from $\tilde{x}_0 = z_{k+1}$, iterates for $i = 0, ...,$ until convergence to $\tilde{x}^*$, as

$$\tilde{z}_{i+1} = \tilde{x}_i - \tilde{\beta} H_k(\tilde{x}_i - z_{k+1}) \quad (22)$$

$$\tilde{x}_{i+1} = prox_g(\tilde{z}_{i+1}) \quad (23)$$

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 (17). Finally $x_{k+1} = \tilde{x}^*$.

The convex optimization problem (21) is a typical TV-L2 model which can be efficiently solved for structured data. However, in this work the domain is discretized by an unstructured triangular mesh. In this case, an efficient way to solve (21) is to apply the anisotropic TV algorithm proposed in [8] as a generalization of the original proposal in [33] for structured data.

The specific choice of the Hessian approximation given in (17) characterizes the proximal Newton-type algorithm for the solution of the optimization problem (6) as *proximal Gauss-Newton algorithm*. The overall sketch is provided in Algorithm 1 for the ℓ_1-norm penalty $g(x)$.

Algorithm 1 Proximal Gauss-Newton-TV (PGN-TV)

Input: x_0, $\alpha > 0$,
Output: x^* % *conductivity distribution*
  For $k = 0, 1, \ldots$, **do until convergence:**
    Update metric H_k from (17)
    $z_{k+1} = x_k - H_k^{-1}\nabla f_\alpha(x_k)$
    $x_{k+1} \leftarrow \text{prox}_g^{H_k}(z_{k+1})$ solved by (22)-(23)
  end

4. **Scaled plug-and-play algorithm.** An important feature of the proximal Gauss-Newton method (18)-(19) is its modular structure. In particular, (18) can be regarded as an updating step as it involves the forward model $f_\alpha(x)$, whereas (19)

can be regarded as a variable metric denoising step as it involves the prior $g(x)$. In particular, (19) can be rewritten as

$$x_{k+1} = \arg\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{2}\|x - z_{k+1}\|^2_{H_k} + g(x) \right\}. \tag{24}$$

In case of image restoration, the variational model (24) is a typical Maximum A Posteriori (MAP) denoiser where z_{k+1} represents the noisy observation. Differently, in our context z_{k+1} in (24) is an intermediate reconstruction of x, since we are dealing with the more challenging task of image reconstruction.

The mathematical equivalence of the proximal operator to the MAP denoising has inspired Venkatakrishnan [56] 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 1 and replacing the proximal operator with a off-the-shelf denoiser $\mathcal{D}_H$, we get

$$x_{k+1} \leftarrow \mathcal{D}_H(z_{k+1}). \tag{25}$$

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. **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 [27]. Here, we consider instead a graph 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-Euclidean manifold domain and relies on the encoder-decoder Graph-U-Net architecture [24]. 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 [24].

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

$$\mathcal{G} = T_M \circ \cdots \circ T_1, \tag{26}$$

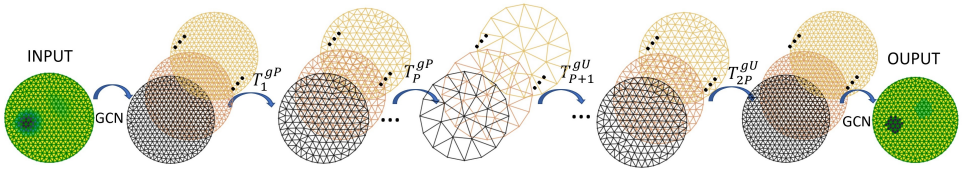

FIGURE 1. GU-Net denoiser structure.

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 [30], 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

$$T_\ell : X \mapsto \sigma(GCN(p(X); \Theta_\ell)), \tag{27}$$

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 $\bar{A} := D^{-1/2}AD^{-1/2}$, 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 $\ell = 1, \ldots, M$ applied to the input feature array of layer ℓ, X_ℓ, is defined as follows

$$X_{\ell+1} = T_\ell(X_\ell) := \sigma((D^{-1/2}AD^{-1/2})p(X_\ell)\Theta_\ell), \tag{28}$$

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

$$GCN(p(X); \Theta_\ell) := \bar{A}p(X_\ell)\Theta_\ell. \tag{29}$$

Hence, the linear convolutional operator GCN, maps a feature matrix Y into a convolved data matrix $Z = \bar{A}Y\Theta$. Note that $D^{-1/2}AD^{-1/2}$ 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 $2(n_c \times n_f) + Mn_f^2$. 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 [24] as mathematical operators. At this aim, we incorporate these operators into (26), so that the encoder-decoder network reads as follows:

$$\mathcal{G} = \underbrace{T_{2P}^{gU} \circ \cdots \circ T_{P+1}^{gU}}_{decoder} \circ \underbrace{T_P^{gP} \circ \cdots \circ T_1^{gP}}_{encoder}, \tag{30}$$

where in T_ℓ^{gP}, $p : X \mapsto gPool(X)$, for $\ell = 1, \ldots, P$, and T_ℓ^{gU}, $p : X \mapsto gUnpool(X)$, for $\ell = P+1, \ldots, 2P$, 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. 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.

At each gPool layer ℓ, (encoder of level ℓ) with $\ell = 1, \ldots, P$, 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

$$X_{\ell+1} = (SX_\ell) \odot s(S(X_\ell \nu^\ell)) \mathbf{1}_{n_c}^T \tag{31}$$

  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

$$X_{\ell_u+1} = S^T X_{\ell_u}. \tag{32}$$

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

The overall structure of the GU-Net-denoiser is illustrated in Fig.1.

5.1. **On the nonexpansiveness of the GU-Net-denoiser.** Building nonexapansive 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 [43]. 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 $T : \mathcal{X} \to \mathcal{X}$ is Lipschitzian with constant contraction factor $\delta \in]0, +\infty[$ if $(\forall x \in \mathcal{X})(\forall y \in \mathcal{X})$

$$\|Tx - Ty\| \leq \delta \|x - y\|. \tag{33}$$

If $\delta < 1$ above, then T is a strict contraction. If $\delta = 1$ then T is nonexpansive.
The following result easily derives from [16].

Lemma 5.1. *The upper bound on the Lipschitz constant for the entire $\mathcal{G}$ network defined in (30), assuming that each layer $T_\ell, \ell = 1, .., M$, defined in (27) is represented by the same composition of functions, is given by:*

$$\mathcal{L}(\mathcal{G}) \leq (\mathcal{L}(T_\ell))^M, \tag{34}$$

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 (28) has a Lipschitz constant $\mathcal{L}(T_\ell)$ lower than 1, under mild assumptions on Pool and Unpool operators.

Proposition 5.2. *Let $\mathcal{G}$ be the GU-Net defined in (30), with M layers $T_\ell : X \mapsto \sigma(GCN(p(X); \Theta))$, defined in (27), $\ell = 1, \ldots, M$, where $GCN(X; \Theta)$ 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 5.1 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 (33) and (29), we have

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

Here we neglected the activation function σ as the ReLU function used has Lipschitz constant one, [16], 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. □

5.2. **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:

$$z_{k+1} = x_k - H_k^{-1}\nabla f_\alpha(x_k) \tag{35}$$

$$x_{k+1} = \mathcal{D}_H(z_{k+1}). \tag{36}$$

Regarding the forward step (35), the gradient is computed as in (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 (17), with matrix L corresponding to either the identity matrix or the discretization of the first-order derivatives. We will named the proposed method, shortly, *PnP-PGN*.

6. **Application to the inverse EIT problem.** In this section we consider the solution of the inverse Electrical Impedance Tomography problem, a challenging ill-posed problem which aims to estimate the internal electrical conductivity distribution of an object Ω by injecting current and measuring voltages, using electrodes on the boundary $\partial\Omega$ of the object.

The *forward EIT problem* aims at determining the boundary voltages for a given object with known conductivity x. Following the accurate Complete Electrode Model (CEM), introduced in [47], the forward EIT problem can be formulated as follows:

$$\nabla \cdot (x\nabla u) = 0 \quad \text{in} \quad \Omega, \tag{37}$$

with boundary conditions introduced by modelling electrodes

$$\begin{cases} u + z_l x\frac{\partial u}{\partial n} = V_l & \text{on} \quad E_l,\ l = 1,..,L, \\ \int_{E_l} x\frac{\partial u}{\partial n} ds = I_l & \text{on} \quad \Gamma, \\ x\frac{\partial u}{\partial n} = 0 & \text{on} \quad \tilde{\Gamma}, \end{cases} \tag{38}$$

where Γ ($\tilde{\Gamma}$) is the boundary $\partial\Omega$ with (without) electrodes, V_l is the unknown voltage to be measured by l-th electrode E_l when the currents I_l are applied, z_l are the contact impedances. The solution of forward EIT problem amounts to solving the boundary problem (37)-(38), using, in general, the Finite Element Method [6].

From now on we restrict the conductivities x to a finite-dimensional space of piecewise polynomials defined on a domain Ω discretized into n_T triangles $\{\tau_j\}_{j=1}^{n_T}$, with x being constant over each of them. Given a FEM of an EIT medium, the solution of the *forward EIT problem* is the vector of voltages 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 $y^\delta \in \mathbb{R}^{n_M}$ induced on another pair of electrodes.

The goal of the *inverse EIT problem*, given the measured voltages y^δ on the electrodes, is to reconstruct electrical conductivity distribution x of the internal part of the object Ω.

The underlying nonlinear least squares 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. The computational difficulties of the EIT inverse problem can be partially alleviated by applying classical regularization methods which rely on the linearized model of the nonlinear forward operator [3, 26]. However, the linearization of the EIT formulation is only a rough approximation of the real measurement system, and this significantly affects the reconstruction results, see [25]. To reduce the ill-posedness while keeping high accuracy reconstructions, the nonlinear least squares EIT problem is usually regularized by adding a penalty term, and reads as

$$x^* \in \arg\min_{x\in\mathbb{R}^{n_T}} f_\alpha(x), \qquad f_\alpha(x) = \|F(x) - y^\delta\|_2^2 + \frac{\alpha}{2}\|L(x - x_o)\|_2^2, \tag{39}$$

with f_α defined as in (3), $F : \mathbb{R}^{n_T} \to \mathbb{R}^{n_M}$ represents the discrete version of the Forward Operator defined by (37)-(38) which is a nonlinear 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 (39) is the classical generalized nonlinear Tikhonov model, a standard choice in EIT [3]. A popular approximated solution of the optimization problem (39) can be obtained by applying the iterative Regularized Gauss-Newton (RGN) method which iterates as follows

$$x_{k+1} = x_k + p_k \tag{40}$$

where p_k is the search direction from the current iterate determined by solving the linear system

$$(J(x_k)^T J(x_k) + \alpha L^T L)p_k^{\text{RGN-Tik}} = J(x_k)^T(y^\delta - F(x_k)) - \alpha L^T L(x_k - x_0). \tag{RGN-Tik}$$

The regularizing Levenberg-Marquardt (LM) algorithm [19] simplifies the RGN algorithm, by computing the search direction as

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

7. **Applying PnP-PGN to the inverse nonlinear 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 the convergence for all the experiments.
Measurements y^δ are simulated through *adjacent injection - adjacent measurement* protocol via pyEIT, a python-based framework for Electrical Impedance Tomography [2].

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). The Mean Squared Error (MSE) defined as

$$MSE(x^{GT}, x^*) := \frac{\|x^{GT} - x^*\|_2^2}{n_T}, \tag{41}$$

measures how well the original conductivity distribution is reconstructed in case a ground truth conductivity distribution x^{GT} is known. The structural similarity measure SSIM between a reconstructed conductivity x^* and the corresponding ground truth x^{GT} is defined as follows

$$SSIM(x^{GT}, x^*) := \frac{1}{n_T} \sum_{i=1}^{n_T} SSIM_{\tau_i}(x_i^{GT}, x_i^*), \tag{42}$$

where $SSIM_{\tau_i}(x_i^{GT}, x_i^*)$ is the similarity measure on a single triangle τ_i; we refer the reader to [8] for details on the implementation. Similarly to the SSIM metric used in image processing, $SSIM(X, Y)$ in (42) quantifies how much X and Y are different, with $SSIM = 1$ if X and Y are identical and $SSIM$ which tends toward 0 when X and Y are very different.

7.1. **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 $\mathcal{D}_H$ in (36) of PnP-PGN 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 (30), 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 (28). 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.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 $\hat{x}^j, j = 1, \ldots, 5$, which characterize a decreasing level of accuracy, or, better, an increasing level of noise. The training sample $\hat{x}^j, j = 1, \ldots, 5$ 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

$$\theta_k^* \leftarrow \text{argmin}_{\theta_k} \tfrac{1}{n_S} \mathcal{L}(\theta_k)$$

$$\mathcal{L}(\theta_k) := (1-\gamma) \textstyle\sum_{i=1}^{n_S} \sum_{j=0}^{5} \|\mathcal{G}(\hat{x}_i^j; \theta_k) - x_i^{GT}\|_2^2 + \gamma \sum_{i=1}^{n_S} \sum_{j=0}^{5} \|\mathcal{G}(\hat{x}_i^j; \theta_k) - x_i^{GT}\|_1, \tag{43}$$

with $\gamma \in (0, 1)$, and $\mathcal{G}(\cdot; \theta_k)$ the deep GU-Net denoiser defined in (30). 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 (43). 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

$$y^\delta = y + \eta, \tag{44}$$

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.

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 [39] and PyTorch Geometric [11], 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.

7.2. **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 1, labelled as PGN-TV), as well as the variational network proposed in [8], named EITGN-NET, and the iterative Graph Convolutional Newton-type Method, named GCNM [23].

The parameter α in the function f_α defined in (3) is set to 10^{-8} for the methods PnP-PGN, PGN-TV and RGN-Tik. This choice corresponds to a sufficiently low α 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 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:

$$k > 200\,, \quad err_k := \left\|x_{k+1} - x_k\right\|_2 / \left\|x_k\right\|_2 < 5 \cdot 10^{-4}. \tag{45}$$

In Table 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 (17). 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.3 we show the conductivity reconstructions obtained by applying PnP-PGN (with $L = D$), RGN-Tik, PGN-TV and GCNM, to six representative test cases; together with the associated (MSE, SSIM) values. In Fig.3 (first column) is reported the ground truth (GT) image x^{GT} for a qualitative inspection. In the last row of Fig. 3 we validated the generalizability to different domains of the PnP-PGN method, that uses the GU-Net-denoiser pre-trained only on circular anomalies. In particular the case labelled 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-know drawback of the learning-based approaches when they are applied to inference tests slightly different from what they have been trained with.

	PnP-PGN (L= I)	PnP-PGN (L=D)	EITGN-NET	RGN-Tik	PGN-TV	GCNM
MSE	0.0042	0.0036	0.0049	0.0067	0.0055	0.0040
SSIM	0.91	0.91	0.87	0.80	0.86	0.87

TABLE 1. Performance on averaged MSE, SSIM metrics on the entire set of test cases.

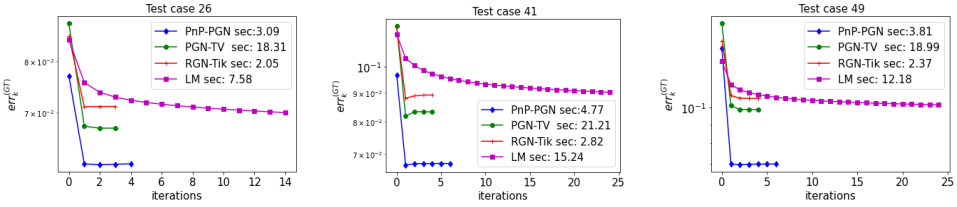

FIGURE 2. Relative error (46) between x^{GT} and the approximated solutions obtained by the algorithms PnP-PGN, PGN-TV, RGN-Tik, and LM, for three different test cases.

In Fig. 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 three representative test cases. In particular, the plots report the relative errors $err_k^{(GT)}$, in terms of the number k of iterations

$$err_k^{(GT)} := \left\|x_k^* - x^{GT}\right\|_2 / \left\|x^{GT}\right\|_2 \ . \tag{46}$$

The solution obtained by the proposed PnP-PGM, with a prior learned by the data, outperforms over 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.

7.3. **Robustness to noise.** We validated our PnP-PGN algorithm starting by acquisitions y^δ corrupted as in (44), 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 (43) with $\bar{x}^{\delta j}$ obtained starting from the noisy training data set corresponding to $\delta = 5 \times 10^{-3}$.

In Fig.4 and Fig.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.

To summarize the performance in the noisy experiments, in Table 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

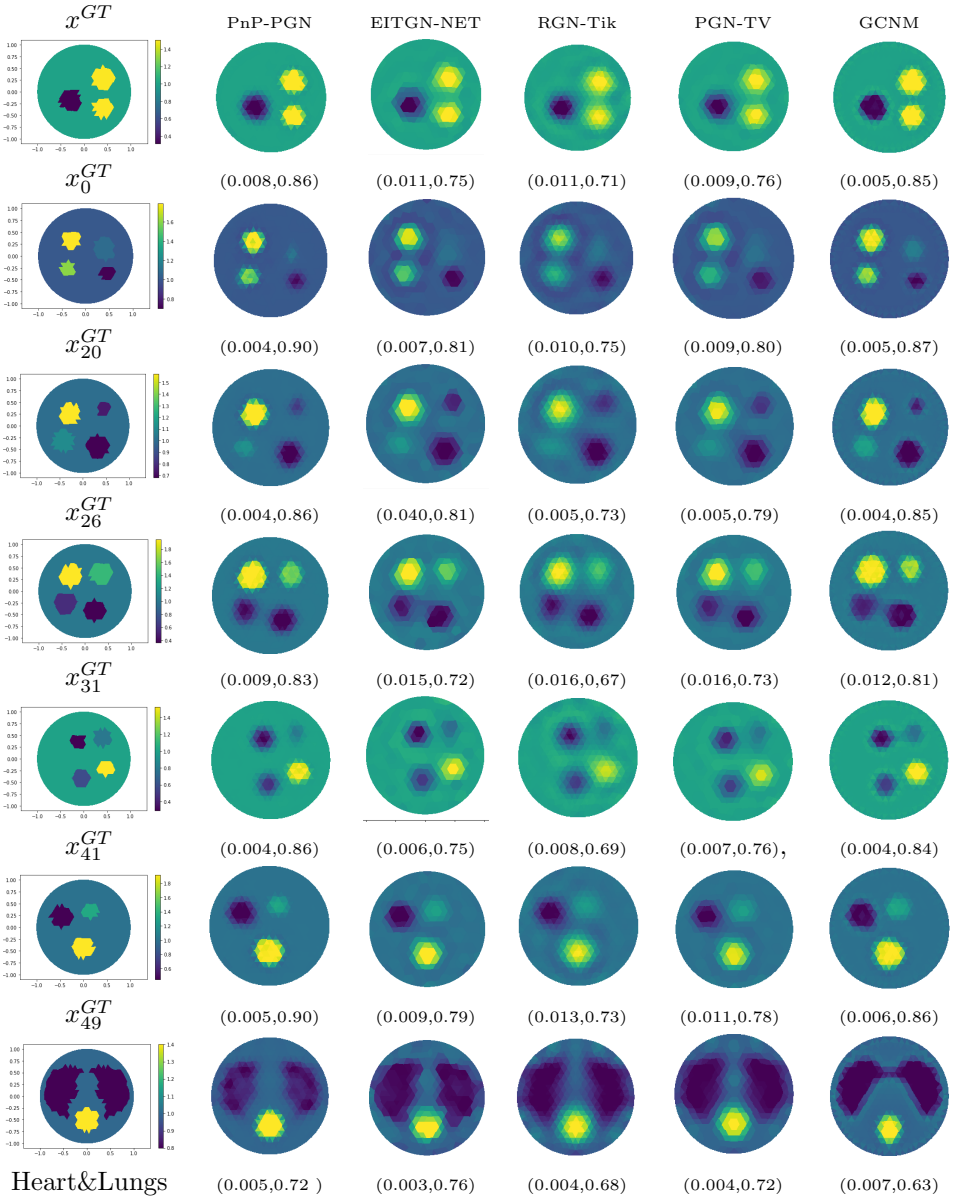

FIGURE 3. Reconstructions obtained by (column-wise from 2-6) PnP-PGN, RGN-Tik, EITGN-NET, PGN-TV and GCNM; x^{GT} (first column) is the ground truth conductivity; (MSE,SSIM) values are reported for each test case.

and SSIM the benefits of a data-driven regularization approach in a variational well performing optimization method.

7.4. **Performance with experimental data.** We tested our PnP-PGN algorithm on the open 2D electrical impedance tomography dataset [21] described in detail in

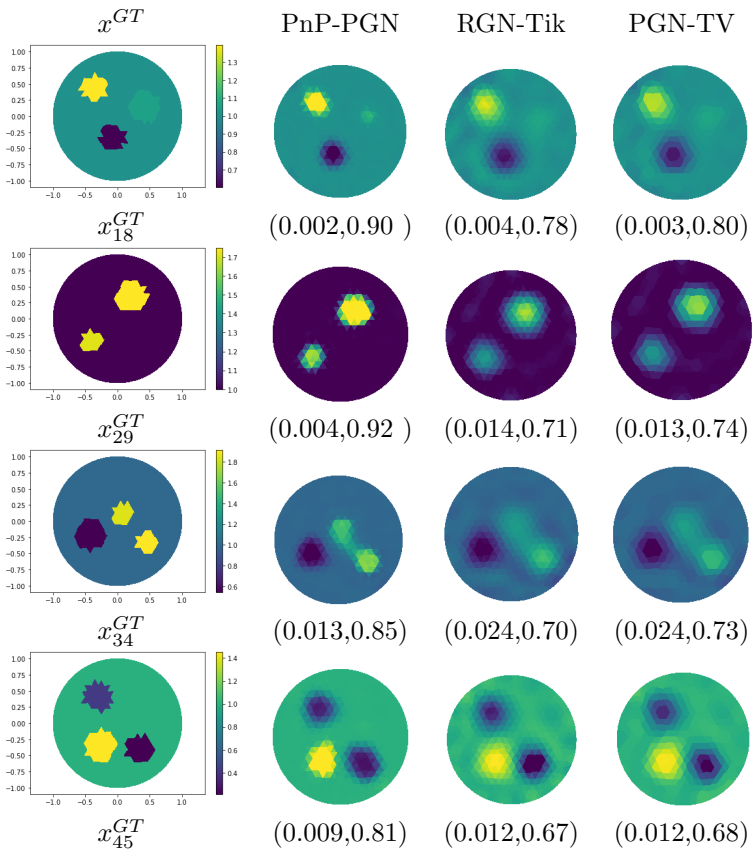

FIGURE 4. Robustness to the noise: comparison between x^{GT} and reconstructed conductivities from noisy measurements (SNR=55 db) obtained by PnP-PGN (first column), RGN-Tik (second column) and PGN-TV (third column).

	PnP-PGN	RGN-Tik	PGN-TV	GCNM
noise 55dB				
MSE	0.004	0.009	0.008	0.004
SSIM	0.90	0.76	0.78	0.87
noise 48dB				
MSE	0.005	0.009	0.009	0.004
SSIM	0.89	0.71	0.73	0.87

TABLE 2. Noisy measurements: performance on averaged MSE and SSIM metrics on the entire set of test cases.

[22]. The setup consists of a circular tank containing 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 kind 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

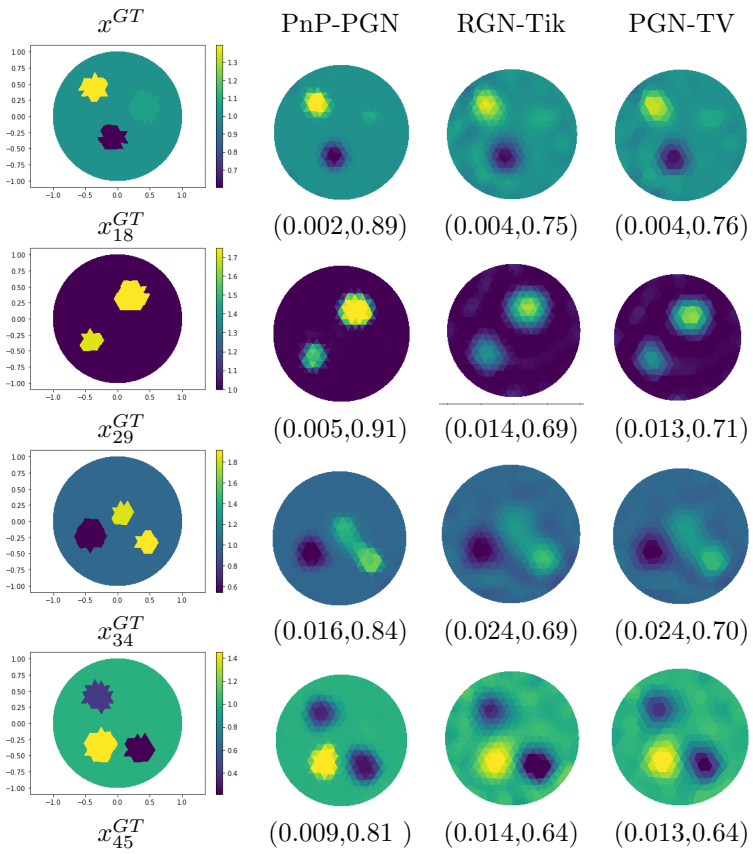

FIGURE 5. Robustness to the noise: comparison between x^{GT} and reconstructed conductivities from noisy measurements (SNR=48db) obtained by PnP-PGN (first column), RGN-Tik (second column) and PGN-TV (third column)

were performed for each example, and for each current injection 13 adjacent 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. 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 challenging case of hollow metal objects. In fact, with PnP-PGN the hole (shielded region) is visible in the reconstruction while it is completely (incorrectly) filled in the RGN-Tik solution. Note that the learned PnP denoiser has been trained with anomalies without holes.

Shapes cannot be recovered by either algorithms. 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.

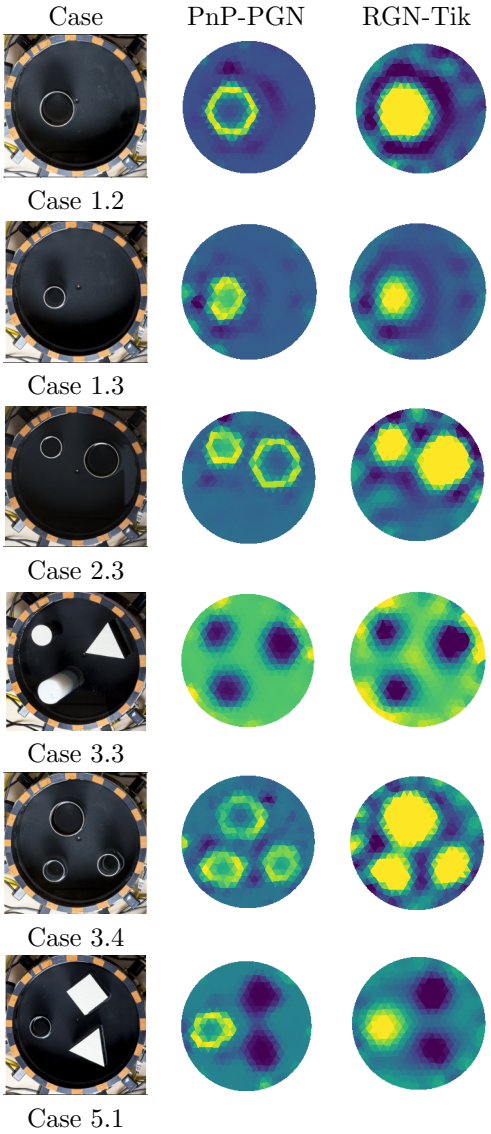

FIGURE 6. Reconstructions with experimental data from the 2D electrical impedance tomography dataset [21]. Picture of the setup (left), reconstruction with the proposed PnP-PGN (center), reconstruction with RGN-Tik for comparison (right).

8. **Conclusions.** The PnP regularization methods belong to the class of variational network, 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 nonlinear problems on unstructured data, which generalize the uniform grids (images) typically used with state-of-the-art PnP denoisers. We show the performance of our proposal

in the reconstruction of conductivity distributions, solutions of inverse EIT problems. The results outperform those of classical state-of-the-art variational methods.

The proposed PnP-PGN benefits of a powerful pre-trained graph network which 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, as proposed in [45]. We will further investigate the theoretical analysis on the convergence of the PnP framework, and we will explore the possibility to apply the consensus equilibrium strategy [5] to obtain an optimization-free EIT reconstruction.

Acknowledgments. This work was supported in part by the National Group for Scientific Computation (GNCS-INDAM), Research Projects 2022, and in part by MIUR RFO projects.

Appendix A.

Proof of Prop. 3.2. By simply replacing ∇f_α defined as in (4), in (11), we get (18). As concerns (19), starting from (12) and applying (16) and (17), we get

$$\begin{aligned}\|y-x\|_H^2 &=< H(y-x),(y-x) >\\ &=< (J^TJ+\alpha L^TL)(y-x),(y-x) >\\ &= \|J(y-x)\|_2^2+\alpha\|L(y-x)\|_2^2.\end{aligned}$$

Equations (18)-(19) are equivalent to the scaled proximal map

$$x_{k+1}=\mathrm{prox}_g^{H_k}(x_k-H_k^{-1}(J^T(x_k)(F(x_k)-y^\delta)+\alpha L^TL(x_k-x_0))). \tag{47}$$

In order to prove the equivalence between (18)-(19) and (20), we prove the equivalence between (47) and (20). At this aim, we consider the first-order necessary conditions for (20), satisfied for x^*, which read as

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

with

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

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

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

$$x^*\in(I+H^{-1}\partial g)^{-1}(I-H^{-1}\nabla\tilde{f}_\alpha)(x^*). \tag{48}$$

According to (10), equation (48) can be interpreted as fixed point scheme that generates a sequence $\{x_{k+1}\}$ by iterating:

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

namely,

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

which can be rewritten as the proximal map (47). □

REFERENCES

[1] S. Becker, J. Fadili and P. Ochs, On Quasi-Newton forward-backward splitting: Proximal calculus and convergence, *SIAM Journal on Optimization*, **29** (2019), 2445–2481.

[2] B. Liu, B. Yang, C. Xu, J. Xia, M. Dai, Z. Ji, F. You, X. Dong, X. Shi and F. Fu, pyEIT: A python based framework for electrical impedance tomography, *SoftwareX*, **7** (2018), 304-308.

[3] A. Borsic, B. M. Graham, A. Adler and W. R. B. Lionheart, In vivo impedance imaging with total variation regularization, in *IEEE Transactions on Medical Imaging*, **29** (2010), 44-54.

[4] A. Buades, B. Coll and J.-M. Morel, A non-local algorithm for image denoising, in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, (2005), 60-65.

[5] G. T. Buzzard, S. H. Chan, S. Sreehari and C. A. Bouman, Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium, *SIAM Journal on Imaging Sciences*, **11** (2018), 2001-2020.

[6] A.-P. Calderón, On an inverse boundary value problem, In: *Seminar on Numerical Analysis and its Applications to Continuum Physics*, (1980), 65-73.

[7] M. Cheney, D. Isaacson, J. C. Newell, S. Simske and J. Goble, NOSER: An algorithm for solving the inverse conductivity problem, *Int. J. Imaging Syst. Technol.*, **2** (1990), 66-75.

[8] F. Colibazzi, D. Lazzaro, S. Morigi and A. Samoré, Learning nonlinear electrical impedance tomography, *Journal of Scientific Computing*, **90** (2022), Paper No. 58, 23 pp.

[9] P. L. Combettes and V. R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Modeling & Simulation*, **4** (2005), 1168-1200.

[10] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, *IEEE Trans. Image Process.*, **16** (2007), 2080-2095.

[11] M. Fey and J. E. Lenssen, Fast graph representation learning with PyTorch Geometric, In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[12] H. Fu, H. Liu, B. Han, Bo, Y. Yang and Y. Hu, A proximal iteratively regularized Gauss-Newton method for nonlinear inverse problems, *Journal of Inverse and Ill-posed Problems*, **25** (2017), 341-356.

[13] R. G. Gavaskar, C. D. Athalye and K. N. Chaudhury, On plug-and-play regularization using linear denoisers, in *IEEE Transactions on Image Processing*, **30** (2021), 4802-4813.

[14] R. G. Gavaskar and K. N. Chaudhury, Plug-and-play ista converges with kernel denoisers, *IEEE Signal Processing Letters*, **27** (2020), 610-614.

[15] D. Geman and C. Yang, Nonlinear image recovery with half-quadratic regularization, *IEEE Transactions on Image Processing*, **4** (1995), 932-946.

[16] H. Gouk, E. Frank, B. Pfahringer and M. J. Cree, Regularisation of neural networks by enforcing Lipschitz continuity, *Machine Learning*, **110** (2021), 393-416.

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

[18] K. Hammernik, et al., Learning a variational network for reconstruction of accelerated MRI data, *Magnetic Resonance in Medicine*, **79** (2018), 3055-3071.

[19] M. Hanke, A regularizing Levenberg - Marquardt scheme, with applications to inverse groundwater filtration problems, *Inverse Problems*, **13** (1997), 79-95.

[20] N. Hänninen, A. Pulkkinen, S. Arridge, T. Tarvainen, Adaptive stochastic Gauss-Newton method with optical Monte Carlo for quantitative photoacoustic tomography, *J. Biomed Opt.*, **27** (2022), 083013.

[21] A. Hauptmann, V. Kolehmainen, N.M. Mach, T. Savolainen, A. Seppänen, S. Siltanen, 2D electrical impedance tomography dataset, https://doi.org/10.5281/zenodo.1203914, 2017.

[22] A. Hauptmann, V. Kolehmainen, N.M. Mach, T. Savolainen, A. Seppänen, S. Siltanen, Open 2D Electrical Impedance Tomography data archive, https://doi.org/10.48550/arXiv.1704.011782017, 2017.

[23] W. Herzberg, D. B. Rowe, A. Hauptmann and S. J. Hamilton, Graph convolutional networks for model-based learning in nonlinear inverse problems, *IEEE Transactions on Compuational Imaging*, **7** (2021), 1341-1353.

[24] G. Hongyang and S. Ji Graph U-Nets, Proceedings of the 36th International Conference on Machine Learning, 2019, **97**, 2083–2092.

[25] M. Huska, D. Lazzaro and S. Morigi, A forward-backward strategy for handling non-linearity in Electrical Impedence Tomography, *Lecture Notes LNCS 12951 Part III*, Computational

Science and Its Applications, Eds: Gervasi, O., Murgante, B., Misra, S., Garau, C., Blečić, I., et al., ISBN 978-3-030-86970-0, 2021, 635-651.

[26] M. Huska, D. Lazzaro, S. Morigi, A. Samore and G. Scrivanti, Spatially-adaptive variational reconstructions for linear inverse electrical impedance tomography, *J. Sci. Comput.*, **84/3** (2020), 29 pp.

[27] F. Jia, W. H. Wong and T. Zeng, DDUNet: Dense dense U-net with applications in image denoising, *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, 354-364.

[28] J. Jauhiainen, P. Kuusela, A. Seppïen and T. Valkonen, Relaxed Gauss–Newton methods with applications to electrical impedance tomography, *SIAM Journal on Imaging Sciences*, **13** (2020), 1415-1445.

[29] U. S. Kamilov, H. Mansour and B. Wohlberg, A Plug-and-play priors approach for solving nonlinear imaging inverse problems, in *IEEE Signal Processing Letters*, **24** (2017), 1872-1876.

[30] T. N. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, International Conference on Learning Representations, 2017.

[31] A. Lanza, S. Morigi and F. Sgallari, Convex image denoising via non-convex regularization, In J.-F. Aujol, M. Nikolova, and N. Papadakis, editors, *Proc. Scale Space and Variational Methods in Computer Vision (SSVM)*, Springer International Publishing, 2015, 666-677.

[32] J. D. Lee, Y. Sun and M. A. Saunders, Proximal Newton-type methods for minimizing composite functions, *SIAM Journal on Optimization*, **24** (2014), 1420-1443.

[33] Y. Li and S. Osher, A new median formula with applications to PDE based denoising, *Communications in Mathematical Sciences*, **7** (2009), 741-753.

[34] Z. Li and J. Wu, Learning deep cnn denoiser priors for depth image inpainting, *Applied Sciences*, **9** (2019), 1103.

[35] W. R. B. Lionheart, EIT Reconstruction algorithms: Pitfalls, challenges and recent developments, *Physiological Measurement*, **25** (2004), 125.

[36] T. Meinhardt, et al., Learning proximal operators: Using denoising networks for regularizing inverse imaging problems, *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[37] N. Parikh and S. Boyd, Proximal algorithms, *Found. Trends Optim.*, **3** (2014), 127-239.

[38] Y. Park, S. Dhar, S. Boyd and S. Mohak, Variable metric proximal gradient method with diagonal Barzilai-Borwein stepsize, *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (2020), 3597-3601.

[39] A. Paszke, S. Gross, F. Massa, et al., Pytorch: An imperative style, high-performance deep learning library, In *Advances in Neural Information Processing Systems (NIPS)*, (2019), 8024-8035.

[40] J. Portilla, V. Strela, M. J. Wainwright and E. P. Simoncelli, Image denoising using scale mixtures of gaussians in the wavelet domain, *IEEE Transactions on Image Processing*, **12** (2003), 1338-1351.

[41] R. Ramlau and G. Teschke, Tikhonov replacement functionals for iteratively solving nonlinear operator equations, *Inverse Problems*, **21** (2005), 1571-1592.

[42] Y. Romano, M. Elad and P. Milanfar, The little engine that could: Regularization by denoising (RED), *SIAM J. Imaging Sci.*, **10** (2017), 1804-1844.

[43] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang and W. Yin, Plug-and-play methods provably converge with properly trained denoisers, *Proc. Int. Conf. Mach. Learn.*, **97** (2019), 5546-5557.

[44] S. Salzo and S. Villa, Convergence analysis of a proximal Gauss-Newton method, *Computational Optimization and Applications*, **53** (2012), 557-589.

[45] A. Samoré, M. Rusci, D. Lazzaro, P. Melpignano, L. Benini and S. Morigi, BrightNet: A deep CNN for OLED-Based point of care immunofluorescent diagnostic systems, *IEEE Transactions on Instrumentation and Measurement*, **69** (2020), 6766-6775.

[46] I. Selesnick, A. Lanza, S. Morigi and F. Sgallari, Non-convex total variation regularization for convex denoising of signals, *J. Math. Imaging and Vision*, **62** (2020), 825-841.

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

[48] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons and C. A. Bouman, Plug-and-play priors for bright field electron tomography and sparse interpolation, *IEEE Trans. Comput. Imaging*, **2** (2016), 408-423.

[49] J. Sun, H. Li, Z. Xu, Deep ADMM-net for compressive sensing MRI, *Advances in Neural Information Processing Systems*, **29** (2016).
[50] Y. Sun, B. Wohlberg and U. S. Kamilov, An online plug-and-play algorithm for regularized image reconstruction, *IEEE Trans. Comput. Imaging*, 5 (2019), 395-408.
[51] Y. Sun, Z. Wu, X. Xu, B. Wohlberg and U. S. Kamilov, Scalable plug-and-play ADMM with convergence guarantees, in *IEEE Transactions on Computational Imaging*, **7** (2021), 849-863.
[52] T. Tarvainen, M. Vauhkonen and S. R. Arridge, Gauss-Newton reconstruction method for optical tomography using the finite element solution of the radiative transfer equation, *Journal of Quantitative Spectroscopy and Radiative Transfer*, **109** (2008), 2767-2778.
[53] T. Tirer and R. Giryes, Image restoration by iterative denoising and backward projections, *IEEE Trans. Image Process.*, **28** (2019), 1220-1234.
[54] T. Tirer and R. Giryes, Super-resolution via image-adapted denoising cnns: Incorporating external and internal learning, *IEEE Signal Processing Letters*, **26** (2019), 1080-1084.
[55] M. Vauhkonen, D. Vadasz, P. A. Karjalainen, E. Somersalo and J. P. Kaipio, Tikhonov regularization and prior information in electrical impedance tomography, in *IEEE Transactions on Medical Imaging*, **17** (1998), 285-293.
[56] S. V. Venkatakrishnan, C. A. Bouman and B. Wohlberg, Plug-and-Play priors for model based reconstruction, *2013 IEEE Global Conference on Signal and Information Processing*, 2013, 945-948.
[57] J. Xiang, Y. Dong, M. Zhang and Y. Li, Design of a magnetic induction tomography system by gradiometer coils for conductive fluid imaging, *IEEE Access*, **7** (2019), 56733-56744.
[58] J. Zhang and B. Ghanem, ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018).
[59] K. Zhang, W. Zuo, S. Gu and L. Zhang, Learning deep CNN denoiser prior for image restoration, *Proc. IEEE Comput. Vis. Pattern Recognit.*, (2017), 3929-3938.
[60] K. Zhang, W. Zuo and L. Zhang, Deep plug-and-play super-resolution for arbitrary blur kernels, *Proc. IEEE Comput. Vis. Pattern Recognit.*, (2019), 1671-1681.
[61] D. Zoran and Y. Weiss, From learning models of natural image patches to whole image restoration, in *IEEE International Conference on Computer Vision*, (2011), 479-486.