TO BE OR NOT TO BE STABLE, THAT IS THE QUESTION:
UNDERSTANDING NEURAL NETWORKS FOR INVERSE PROBLEMS

A PREPRINT

Davide Evangelista  
Department of Mathematics, University of Bologna, Italy  
davide.evangelista5@unibo.it.

Elena Loli Piccolomini  
Department of Computer Science and Engineering, University of Bologna, Italy.

Elena Morotti  
Department of Political and Social Sciences, University of Bologna, Italy.

James Nagy  
Department of Mathematics, Emory University, Atlanta.

ABSTRACT

The solution of linear inverse problems arising, for example, in signal and image processing is a challenging problem since the ill-conditioning amplifies, in the solution, the noise present in the data. Recently introduced algorithms based on deep learning overwhelm the more traditional model-based approaches in performance, but they typically suffer from instability with respect to data perturbation. In this paper, we theoretically analyze the trade-off between stability and accuracy of neural networks, when used to solve linear imaging inverse problems for not under-determined cases. Moreover, we propose different supervised and unsupervised solutions to increase the network stability and maintain a good accuracy, by means of regularization properties inherited from a model-based iterative scheme during the network training and pre-processing stabilizing operator in the neural networks. Extensive numerical experiments on image deblurring confirm the theoretical results and the effectiveness of the proposed deep learning-based approaches to handle noise on the data.

Keywords  Neural Networks Stability · Linear Inverse Problems · Deep Learning Algorithms · Image Deblurring · trade-off accuracy stability

1 Introduction

Linear inverse problems of the form:

\[ y = Ax, \]  \hspace{1cm} (1)

where \( A \in \mathbb{R}^{m \times n} \) is a full-rank matrix discretizing a linear operator, \( x \in \mathbb{R}^n \) and \( y \in \mathbb{R}^m \) with \( m \geq n \), arise in various image processing tasks, such as deblurring or tomographic reconstruction \([28, 29, 40]\). It is well-known that in these applications, equation (1) represents the discretization of an ill-posed problem. Following the well-known Hadamard definition, a problem is ill-posed if either a solution does not exist, the solution is not unique or it does not continuously depend on the data \( y \). In the case considered in (1) the third condition holds, thereby the computation of the solution becomes very challenging when noise affects the data. In this work, we consider data corrupted by Gaussian noise, i.e.:

\[ y^\delta = Ax^\text{gt} + e, \quad e \sim \mathcal{N}(0, \delta^2 I); \]  \hspace{1cm} (2)

where \( \delta \) denotes the standard deviation of the white additive Gaussian noise, \( I \) is the identity matrix, and \( x^\text{gt} \) is the ground truth, clean image.

Traditional regularization approaches tackle problem (2) as the minimization of an objective function containing a data-fit term and a regularization prior, with possible further constraints on the solution \([9, 18]\). These terms theoretically
grant stability, but, in general, the computational time required by solvers is high and it may be necessary to choose many parameters, tuning them by trial and error on the data.

In the last few years, neural networks have been introduced with great success for the solution of problem (2), since they are capable of achieving greater accuracy than iterative regularized methods [2,13,33]. However, noise-related issues still persist, as their high accuracy is obtained at the expense of robustness against noise in the input data. Specifically, these networks frequently yield suboptimal results when applied to data contaminated with noise that differs from that encountered during the training phase. This tendency is commonly referred to as network instability. Some authors have already studied the behavior of neural networks in the presence of noise on the data, focusing on the solution of under-determined imaging inverse problems (i.e. when \( m < n \) in equation (1)). [22, 53, 44, 9, 15, 32, 34, 38, 39, 43, 48, 55, 56, 24, 15]. We note that the paper [24] offers a comprehensive bibliography on this topic, with the authors remarking that “stability implies a universal barrier on performance”.

However, to the best of our knowledge, a mathematically grounded understanding is still lacking and no works address the case of \( m \geq n \).

**Contributions** In this work, we look at neural networks as solvers of discrete ill-posed problems, and we contribute to the state-of-art as follows.

Firstly, we adapt the regularization theory presented by Engl et al. in [17] for solving discrete inverse problems through neural networks. It is noteworthy that Engl et al. examined regularization in Hilbert spaces, while our focus is on discrete inverse problems. Prior to introducing neural networks as solvers, we present a more general theory encompassing a broader class of functions, termed *reconstructors*, designed for addressing problem (2). Within this framework, we first formalize the two fundamental concepts of \( \eta \)-accuracy and \( \epsilon \)-stability, and then we present significant findings for a class of functions called *stabilizers*. We establish a mathematical relationship that quantifies the trade-off between stability and accuracy, demonstrating that enhancing a solver’s stability is impossible without compromising its accuracy. In this theoretical approach, neural networks have been analyzed as formal mathematical operators, shedding light on their wildly discussed ‘black-box’ nature/misinterpretation.

Secondly, we propose a new ground truth-free scheme for *reconstructors* based on neural networks. We refer to this approach as the *REgularized Neural Network* (ReNN), as the target images used in the training procedure are solutions of (2) computed through a regularization method. Beyond being more stable than commonly used neural networks as *reconstructors*, it is applicable in scenarios where collecting a set of ground-truth solutions is challenging or impossible, such as in medical imaging.

Finally, we present a novel stabilization strategy tailored for deep learning-based solvers, which incorporates a stabilizer within a pre-processing operator plugged into a neural network *reconstructor*. This approach demonstrates substantial efficacy in handling elevated noise levels in data. We have termed this methodology STabilized Neural Network (StNN). Furthermore, when integrated with the ReNN scheme, it evolves into the StReNN framework.

**Structure of the paper** The paper is structured as follows. In Section 2, we introduce theoretical concepts related to *reconstructors* for solving an inverse problem of the form presented in (2). In Section 3, we present stabilizers and elucidate their effectiveness by stating their properties, then Section 4 is dedicated to *reconstructors* based on neural networks and presents our proposals. Following that, in Section 5, we describe our experimental setup, whereas Section 6 showcases numerical results pertaining to deblurring and denoising. Finally, Section 7 comprises conclusions and outlines potential directions for future work.

### 2 Reconstructors for the solution of linear inverse problems

This section establishes the theoretical background of the manuscript, providing essential definitions and preliminary results. To improve the readability of the work, however, we start by introducing the notation we will use throughout the paper. We always consider \( x^g \) to lie in a subset \( X \) of \( \mathbb{R}^n \), the set of admissible data. We denote as \( Y = \text{Rg}(A, X) \) the range of \( A \) over \( X \), where \( A \) is a continuous linear operator. We assume \( Y \) to be dense-in-itself (i.e. without isolated point) so that, for any admissible \( x^g \in X \) and any neighborhood \( V \) of \( y = Ax^g \), there is at least an \( x^e \in X \), \( x^e \neq x^g \), such that \( y^e = Ay \in V \). When \( x \in X \subseteq \mathbb{R}^n \) or \( y \in Y \subseteq \mathbb{R}^m \), then \( |x| \) and \( |y| \) will be Euclidian norms. For any \( \epsilon > 0 \), we also define \( Y^\epsilon = \{ y + e : y \in Y, |e| \leq \epsilon \} \). With the following definitions, we can formalize the concept of *reconstructor* to solve problem (1) accurately.

**Definition 2.1.** Any continuous function \( \Psi : \mathbb{R}^m \rightarrow \mathbb{R}^n \), mapping \( y \) to \( x = \Psi(y) \), is called a *reconstructor*. 
Definition 2.2. A reconstructor $\Psi : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is said to be $\eta^{-1}$-accurate, with $\eta > 0$, if:

$$\eta = \sup_{x^g \in X} ||\Psi(Ax^g) - x^g||.$$ 

We define the set $\mathcal{R}_\eta = \{ \Psi : \mathbb{R}^m \rightarrow \mathbb{R}^n ; \Psi$ is a reconstructor with accuracy $\eta^{-1} \}$. We observe that without any other restriction, $\eta$ could be infinite. To avoid any issue, we will always consider reconstructors with finite $\eta$ in the following.

Example 2.1. An $\infty$-accurate reconstructor of problem (1) is given by:

$$\Psi^\dagger(y) = A^\dagger y = (A^*A)^{-1}A^*y,$$

where $A^\dagger$ is the pseudo-inverse matrix, $A^*$ is the transpose of $A$, and the last equality holds since $A$ is assumed to be full-rank. In this case $\Psi^\dagger : \mathbb{R}^m \rightarrow \mathbb{R}^n$ is $\infty$-accurate, as:

$$||\Psi^\dagger(Ax^g) - x^g|| = ||(A^*A)^{-1}A^*(Ax^g) - x^g|| = ||(A^*A)^{-1}(A^*A)x^g - x^g|| = 0.$$

However, reconstructors are rarely applied to noise-free data, hence a focus on the robustness of reconstructors with respect to noise is necessary.

Definition 2.3. Let $\epsilon > 0$ and $\Psi$ be an $\eta^{-1}$-accurate reconstructor applied to problem (2). We define the $\epsilon$-stability constant $C^\epsilon_\Psi$ of $\Psi$ as:

$$C^\epsilon_\Psi = \sup_{x^g \in X} \sup_{||e|| \leq \epsilon} \frac{||\Psi(Ax^g + e) - x^g|| - \eta}{||e||}.$$

We will consider in the following the realistic case of $C^\epsilon_\Psi < \infty$.

Definition 2.4. The reconstructor $\Psi$ is said to be $\epsilon$-stable for a given $\epsilon > 0$ if $C^\epsilon_\Psi \in [0, 1)$. Otherwise, $\Psi$ is said to be $\epsilon$-unstable.

An $\epsilon$-stable reconstructor $\Psi$ does not amplify corruptions having norm less than $\epsilon$ (as graphically represented in Figure 1), since (2.3) implies:

$$||\Psi(Ax^g + e) - x^g|| \leq \eta + C^\epsilon_\Psi ||e|| \quad \forall x^g \in X, \forall e \in \mathbb{R}^m, ||e|| \leq \epsilon.$$

Definition 2.5. We define the stability radius $\rho$ of $\Psi$ as:

$$\rho = \sup\{ \epsilon > 0 ; C^\epsilon_\Psi \in [0, 1) \}.$$

Example 2.2. A reconstructor with an infinite stability radius is the following. Given $\epsilon > 0$, if $\mu$ is a probability distribution over $X$ (for example, $\mu$ is the normalized Lebesgue distribution over $X$), the reconstructor defined as:

$$\Psi^X,\epsilon(y^\delta) = \int_X x^g \mu(dx), \quad \forall y^\delta \in \mathcal{Y}^\epsilon$$
is $\epsilon$-stable independently from the value of $\epsilon > 0$. Indeed:

$$\|\Psi^\epsilon(Ax^{gt} + e) - x^{gt}\| = \left\| \int_{X} x\mu(dx) - x^{gt} \right\| \leq \rho(\mathcal{X}),$$

where $\rho(\mathcal{X})$ is the radius of $\mathcal{X}$, defined as $\rho(\mathcal{X}) = \inf \{ r > 0 : \mathcal{X} \subseteq B(\int_{X} x\mu(dx); r) \}$. As a consequence the stability constant is infinite regardless $\epsilon$, and $\Psi^\epsilon$ has accuracy $\rho(\mathcal{X})^{-1}$.

**Example 2.3.** The pseudo-inverse reconstructor $\Psi^\dagger(y)$ in (2.1) is unstable for any $\epsilon > 0$ when $A$ is ill-conditioned. Indeed:

$$\|\Psi^\dagger(Ax^{gt} + e) - x^{gt}\| = \|(A^*A)^{-1}(A^*A)x^{gt} + (A^*A)^{-1}A^*e - x^{gt}\|$$

$$= \|(A^*A)^{-1}A^*e\|.$$ 

If $A = USV^*$ is the Singular Value Decomposition (SVD) of $A$, then:

$$(A^*A)^{-1}A^*e = (VS^2V^*)^{-1}V\Sigma U^*e = \sum_{i=1}^{n} \frac{u_i^T e}{\sigma_i} v_i,$$

which implies that $\|(A^*A)^{-1}A^*e\| \gg \|e\|$ when $A$ has singular values close to zero.

These examples shed light on a possible conflict between accuracy and stability for a given reconstructor $\Psi$. In the next paragraphs, we study this relationship.

### 2.1 Accuracy vs. Stability Trade-off

We can derive a relation between accuracy and stability, which becomes particularly interesting when $A$ is ill-conditioned.

**Lemma 2.1.** Let $\Psi : \mathbb{R}^m \rightarrow \mathbb{R}^n$ be an $\eta^{-1}$-accurate reconstructor. Then, for any $x^{gt} \in \mathcal{X}$ and for any $\epsilon > 0$, there exists $\tilde{e} \in \mathbb{R}^m$ with $\|\tilde{e}\| \leq \epsilon$ such that:

$$\|\Psi(Ax^{gt} + \tilde{e}) - x^{gt}\| \geq \|A^\dagger\tilde{e}\| - \eta .$$

**Proof.** Since $Ax^{gt} \in \mathcal{Y}$ for any $x^{gt} \in \mathcal{X}$, and since $\mathcal{Y}$ has no isolated points, then for any $\epsilon > 0$ there is an $\tilde{e} \in \mathbb{R}^m$ with $\|\tilde{e}\| \leq \epsilon$ such that $Ax^{gt} + \tilde{e} \in \mathcal{Y}$. Thus, $\exists x' \in \mathcal{X}$ such that $Ax^{gt} + \tilde{e} = Ax'$. Consequently:

$$\|\Psi(Ax^{gt} + \tilde{e}) - x^{gt}\| = \|\Psi(Ax') - x^{gt}\| \geq \|x' - x^{gt}\| - \|\Psi(Ax') - x'\|$$

$$\geq \|x' - x^{gt}\| - \eta .$$

Since $Ax^{gt} + \tilde{e} = Ax'$ by construction, it holds that $\tilde{e} = A(x^{gt} - x')$, which implies that $x^{gt} - x' = A^\dagger\tilde{e}$. To conclude:

$$\|\Psi(Ax^{gt} + \tilde{e}) - x^{gt}\| \geq \|x' - x^{gt}\| - \eta = \|A^\dagger\tilde{e}\| - \eta .$$

Since the corruption $\tilde{e}$ such that the relationship (3) holds for some $\epsilon > 0$ depends on $x^{gt}$, for any $x^{gt} \in \mathcal{X}$, we will consider the set:

$$E(x^{gt}) = \{ e \in \mathbb{R}^m : \text{Equation (3) holds for } e, \text{ for some } \epsilon > 0 \}. \quad (4)$$

**Theorem 2.2** (Trade-off Theorem). Under the assumptions of Lemma 2.1 it holds that, for any $x^{gt} \in \mathcal{X}$ and for any $\tilde{e} \in E(x^{gt})$ with $\|\tilde{e}\| \leq \epsilon$,

$$C_\Psi \geq \frac{\|A^\dagger\tilde{e}\| - 2\eta}{\|\tilde{e}\|}. \quad (5)$$

**Proof.** For any $x^{gt} \in \mathcal{X}$,

$$C_\Psi = \sup_{x \in \mathcal{X}} \sup_{\|e\| \leq \epsilon} \frac{\|\Psi(Ax + e) - x\| - \eta}{\|e\|} \geq \sup_{\|e\| \leq \epsilon} \frac{\|\Psi(Ax^{gt} + e) - x^{gt}\| - \eta}{\|e\|}.$$
If \( \tilde{e} \in E(x^{gt}) \), \( ||\tilde{e}|| \leq \epsilon \), is a perturbation defined on Lemma 2.1, we have:

\[
C_\Psi^e \geq \sup_{||e|| \leq \epsilon} \frac{||\Psi(Ax^{gt} + e) - x^{gt}|| - \eta}{||e||} \\
\geq \frac{||\Psi(Ax^{gt} + \tilde{e}) - x^{gt}|| - \eta}{||\tilde{e}||} \\
\geq \frac{||A^\dagger \tilde{e}|| - 2\eta}{||\tilde{e}||},
\]

which concludes the proof.

**Corollary 2.2.1.** Given the assumptions of Theorem 2.2 if \( \mathcal{X} = \mathbb{R}^n \), there is a constant \( C(A) > 0 \) which depends only on \( A \), such that:

\[
\rho \leq 2 \frac{\eta^{-1} C(A)}{\epsilon}.
\]

**Proof.** Consider a reconstructor \( \Psi \). By Theorem 2.2, for any \( \epsilon > 0 \), any \( x^{gt} \in \mathcal{X} \), and any \( \tilde{e} \in E(x^{gt}) \) with \( ||\tilde{e}|| \leq \epsilon \),

\[
C_\Psi^e \geq \frac{||A^\dagger \tilde{e}|| - 2\eta}{||\tilde{e}||}.
\]

We first observe that, if \( \mathcal{X} = \mathbb{R}^n \), then \( E(x^{gt}) = \mathcal{Y} : = \Psi(B(A, \mathcal{X})) \) for any \( x^{gt} \in \mathcal{X} \). Indeed, \( \hat{e} \in E(x^{gt}) \) if and only if there exists \( x' \in \mathcal{X} \) such that \( \hat{e} = A(x^{gt} - x') \). Since \( \mathcal{X} = \mathbb{R}^n \) is closed under addition, then \( x^{gt} - x' \in \mathcal{X} \), which implies that \( \hat{e} \in \mathcal{Y} \), thus \( E(x^{gt}) \subseteq \mathcal{Y} \). Conversely, if \( y \in \mathcal{Y} \), then by definition there exists \( x \in \mathcal{X} \) such that \( y = Ax \). By defining \( x' = x^{gt} - x \), then \( y = A(x^{gt} - x') \), which implies that \( y \in E(x^{gt}) \) and consequently \( E(x^{gt}) = \mathcal{Y} \).

Now, let \( A = U \Sigma V^* \) be the SVD of \( A \) and define \( \hat{e} = A \left( \frac{\epsilon}{\sigma_n} v_n \right) \), where \( \sigma_n \) and \( v_n \) are the smallest singular value of \( A \) and its associated right-singular vector, respectively. Note that \( \hat{e} \in \mathcal{Y} = E(x^{gt}) \) by definition. Moreover:

\[
\hat{e} = A \left( \frac{\epsilon}{\sigma_n} v_n \right) = U \Sigma V^* \left( \frac{\epsilon}{\sigma_n} v_n \right) = \frac{\epsilon}{\sigma_n} \sum_{i=1}^n \sigma_i u_i (v_i^T v_n).
\]

from which \( ||\hat{e}|| = \epsilon ||v_n|| = \epsilon \leq \epsilon \). Consequently, (5) holds for \( \hat{e} \). Additionally:

\[
A^\dagger \hat{e} = A^\dagger A \left( \frac{\epsilon}{\sigma_n} v_n \right) = \frac{\epsilon}{\sigma_n} v_n,
\]

hence \( ||A^\dagger \hat{e}|| = \frac{\epsilon}{\sigma_n} \). Given that, (5) reads:

\[
C_\Psi^e \geq \frac{||A^\dagger \hat{e}|| - 2\eta}{||\hat{e}||} = \frac{\frac{\epsilon}{\sigma_n} - 2\eta}{\epsilon}.
\]

As a consequence of the above relationship, if \( \frac{\frac{\epsilon}{\sigma_n} - 2\eta}{\epsilon} > 1 \), then \( C_\Psi^e > 1 \), i.e. \( \rho \leq \epsilon \). A simple computation shows that this holds if:

\[
\epsilon < \frac{2}{\eta^{-1} \left( \frac{1-\sigma_n}{\sigma_n} \right)} = \frac{2}{\eta^{-1} C(A)},
\]

concluding the proof by calling \( C(A) = \frac{1-\sigma_n}{\sigma_n} \).

The relation in Corollary 2.2.1 between the stability radius \( \rho \) and the accuracy \( \eta^{-1} \) suggests that there exists a trade-off between accuracy and stability, showing that a very accurate reconstructor is unstable for noise corruption larger than \( \frac{2}{\eta^{-1} C(A)} \). We remark that for ill-conditioned problems \( C(A) = \frac{1-\sigma_n}{\sigma_n} \) can be very large, making the radius potentially very small.

Similarly, Theorem 2.2 shows that a reconstructor \( \Psi \) can be \( \epsilon \)-stable only if its accuracy is bounded.
Corollary 2.2.2. Given the assumptions of Theorem 2.2, there exists $\hat{\eta}(A, \epsilon, \mathcal{X}) \in \mathbb{R} \cup \{+\infty\}$, such that any reconstructor $\Psi$ with accuracy $\eta^{-1} \geq \hat{\eta}(A, \epsilon, \mathcal{X})^{-1}$ is $\epsilon$-unstable, i.e. $C_{\Psi} \geq 1$.

Moreover, if $\mathcal{X} = \mathbb{R}^m$ and $\eta^{-1} \geq \frac{1}{C(A)\sigma_n}$, where $C(A) = \frac{1}{\sigma_n}$, then $\Psi$ is $\epsilon$-unstable.

Proof. From Theorem 2.2, $\Psi$ is $\epsilon$-unstable for a given $\epsilon > 0$ if $\frac{\|A^1 \hat{e}\| - 2n}{\|\hat{e}\|} \geq 1$. Such condition holds if and only if:

$$\eta \leq \frac{\|A^1 \hat{e}\| - \|\hat{e}\|}{2}.$$

Thus, if $\exists \, \hat{e} \in E(x^{gt})$ with $\|\hat{e}\| \leq \epsilon$ such that $\eta \leq \frac{\|A^1 \hat{e}\| - \|\hat{e}\|}{2}$, then $\Psi$ is $\epsilon$-unstable. In particular, if we define:

$$\hat{\eta}(A, \epsilon, \mathcal{X}) = \sup_{x^{gt} \in \mathcal{X}, \hat{e} \in E(x^{gt}), \|\hat{e}\| \leq \epsilon} \frac{\|A^1 \hat{e}\| - \|\hat{e}\|}{2},$$

we get the result. Note that, in general, $\hat{\eta}(A, \epsilon, \mathcal{X})$ could be infinite. In the assumption of $\mathcal{X} = \mathbb{R}^m$, we proved in Corollary 2.2.1 that for any $\epsilon > 0$ and any $x^{gt} \in \mathcal{X}$, we can always choose $\hat{e} \in E(x^{gt})$ with $\|\hat{e}\| \leq \epsilon$ such that $\|A^1 \hat{e}\| - \|\hat{e}\| = \frac{1}{\sigma_n} \epsilon = C(A) \epsilon$. Thus, $\Psi$ is $\epsilon$-unstable if:

$$\eta \leq C(A) \epsilon \frac{\epsilon}{2},$$

which proves the corollary. \qed

2.2 A sufficient condition for stability

Whenever a reconstructor is (locally) Lipschitz continuous, we can also derive conditions assessing its stability. First of all, we recall the definition of locally Lipschitz continuous reconstructors.

Definition 2.6. Given $\mathcal{Y} \subseteq \mathbb{R}^m$ and $\epsilon > 0$, we define the $\epsilon$-Lipschitz (also called local Lipschitz) constant of $\Psi$ over $\mathcal{Y}$ as:

$$L^\epsilon(\Psi, \mathcal{Y}) = \sup_{y \in \mathcal{Y}, z \in \mathbb{R}^m, \|z - y\| \leq \epsilon} \frac{\|\Psi(z) - \Psi(y)\|}{\|z - y\|}.$$

If $L^\epsilon(\Psi, \mathcal{Y}) < \infty$ for some $\epsilon > 0$, then $\Psi$ is said to be locally Lipschitz continuous.

Focusing on our problem (2), we remark we are interested in the cases where $\mathcal{Y} = \mathcal{R} \mathcal{g}(A, \mathcal{X})$. In this case, $y \in \mathcal{Y}$ implies that $\exists x^{gt} \in \mathcal{X}$ such that $y = Ax^{gt}$ and each $z \in \mathbb{R}^m$ with $\|z - y\| \leq \epsilon$ can be characterized by $z = Ax^{gt} + e$ for some $e \in \mathbb{R}^m$ with $\|e\| \leq \epsilon$. Thus, the definition of $L^\epsilon(\Psi, \mathcal{Y})$ can be rewritten as:

$$L^\epsilon(\Psi, \mathcal{Y}) = \sup_{x^{gt} \in \mathcal{X}, \|e\| \leq \epsilon} \frac{\|\Psi(Ax^{gt} + e) - \Psi(Ax^{gt})\|}{\|e\|}.$$

The importance of the local Lipschitz constant $L^\epsilon(\Psi, \mathcal{Y})$ lies in its strong relationship to the stability constant $C_{\Psi}$ of the reconstructor. Indeed, if $y = Ax^{gt} \in \mathcal{Y}$ is corrupted by additional noise $e$ with $\|e\| \leq \epsilon$, then $L^\epsilon(\Psi, \mathcal{Y})$ represents the maximum possible variation of the reconstruction obtained by $\Psi$ around the corrupted $y$, as stated by the following proposition.

Proposition 2.3. If $\Psi \in \mathcal{R}_\eta$ has local Lipschitz constant $L^\epsilon(\Psi, \mathcal{Y})$, then, for any $\|e\| \leq \epsilon$, it holds:

$$\|\Psi(Ax^{gt} + e) - x^{gt}\| \leq \eta + L^\epsilon(\Psi, \mathcal{Y})\|e\|.$$

Proof. By the triangle inequality, it follows that:

$$\|\Psi(Ax^{gt} + e) - x^{gt}\| \leq \|\Psi(Ax^{gt} + e) - \Psi(Ax^{gt})\| + \|\Psi(Ax^{gt}) - x^{gt}\|.$$
Since \( ||e|| \leq \epsilon \), the definition of local Lipschitz constant implies that:
\[
||\Psi(Ax^{gt} + e) - \Psi(Ax^{gt})|| \leq L'(\Psi, \mathcal{Y})||e||
\]
whereas the accuracy of \( \Psi \) gives:
\[
||\Psi(Ax^{gt}) - x^{gt}|| \leq \eta.
\]
Thus, we can conclude:
\[
||\Psi(Ax^{gt} + e) - x^{gt}|| \leq L'(\Psi, \mathcal{Y})||e|| + \eta.
\]

**Corollary 2.3.1.** Under the assumptions of Theorem 2.3, it holds:
\[
C^\epsilon \Psi \leq L'(\Psi, \mathcal{Y}).
\]

**Proof.** From the inequality in Theorem 2.3, we have:
\[
||\Psi(Ax^{gt} + e) - x^{gt}|| \leq \eta + L'(\Psi, \mathcal{Y})||e|| \iff L'(\Psi, \mathcal{Y}) \geq \frac{||\Psi(Ax^{gt} + e) - x^{gt}|| - \eta}{||e||}
\]
for any \( x^{gt} \in \mathcal{X} \) and any \( e \in \mathbb{R}^m \) with \( ||e|| \leq \epsilon \). Consequently, \( L'(\Psi, \mathcal{Y}) \) is a majorant of the set:
\[
\left\{ \frac{||\Psi(Ax^{gt} + e) - x^{gt}|| - \eta}{||e||}; x^{gt} \in \mathcal{X}, ||e|| \leq \epsilon \right\}.
\]
Since \( C^\epsilon \Psi \) is defined as the supremum of this set, by the minimality of the supremum we have \( C^\epsilon \Psi \leq L'(\Psi, \mathcal{Y}) \).

We remark that Corollary 2.3.1 proves that \( \Psi \) is \( \epsilon \)-stable if \( L'(\Psi, \mathcal{Y}) < 1 \), yielding a useful sufficient condition to the assessment of stability.

**Example 2.4.** Under suitable parameter choices, the Tikhonov reconstructor is \( \epsilon \)-stable for any \( \epsilon > 0 \). The Tikhonov reconstructor is built on the Tikhonov method [27, 50] and defined as:
\[
\Psi^{\lambda,L}(y^\delta) = \arg \min_{x \in \mathbb{R}^n} \frac{1}{2} ||Ax - y^\delta||^2 + \frac{\lambda}{2} ||Lx||^2,
\]
where \( \lambda > 0 \) is the regularization parameter and \( L \in \mathbb{R}^{d \times n} \) is a matrix such that \( \ker(A) \cap \ker(L) = \{0\} \). \( L \) is usually chosen as the identity or the forward-difference operator. We can prove the following proposition regarding Tikhonov stability.

**Proposition 2.4.** Let \( \epsilon > 0 \) and \( L \in \mathbb{R}^{d \times n} \). Then \( \exists \lambda > 0 \) such that:
\[
L'(\Psi^{\lambda,L}, \mathcal{Y}) < 1.
\]

**Proof.** For any \( \lambda > 0 \) and any \( y^\delta \in \mathcal{Y}^\delta \), it can be shown, by considering the normal equations of (8), that:
\[
\Psi^{\lambda,L}(y^\delta) = (A^*A + \lambda L^*L)^{-1}A^*y^\delta = \frac{1}{\lambda} (A^*A + L^*L)^{-1}A^*y^\delta.
\]
Consequently, for any \( y^\delta \in \mathcal{Y}^\delta \), it holds that \( \Psi^{\lambda,L}(y^\delta) \to 0 \) for \( \lambda \to \infty \). Then:
\[
L'(\Psi^{\lambda,L}, \mathcal{Y}) = \sup_{x^{gt} \in \mathcal{X}, ||e|| \leq \epsilon} \frac{||\Psi^{\lambda,L}(Ax^{gt} + e) - \Psi^{\lambda,L}(Ax^{gt})||}{||e||} \to 0 \quad \text{for} \quad \lambda \to \infty,
\]
which implies that, for all \( \alpha > 0 \), there exists \( \lambda > 0 \) such that for any \( \lambda > \bar{\lambda} \), \( L'(\Psi^{\lambda,L}, \mathcal{Y}) < \alpha \). Choosing \( \alpha = 1 \) we obtain the required result.

Corollary 2.3.1 and Proposition 2.4 demonstrate that it is always possible to build a stable Tikhonov reconstructor. Such property will play a crucial role in Subsection 4.3, where we will explain our proposed ReNN approach.

## 3 Stabilizers in the solution of linear inverse problems

In this section, we delve into additional properties pertinent to stable reconstructors, by introducing the novel concept of stabilizer which will be exploited in Subsection 4.4 to define our StNN and StReNN approaches.
3.1 Stabilizers and properties

**Definition 3.1.** A continuous functions \( \phi : \mathbb{R}^m \to \mathbb{R}^l \) is an \( \epsilon \)-stabilizer of a reconstructor \( \Psi : \mathbb{R}^m \to \mathbb{R}^n \) if:

1. \( \forall e \in \mathbb{R}^m \) with \( ||e|| \leq \epsilon \), \( \exists C^*_\phi \in [0,1) \) and \( \exists e' \in \mathbb{R}^n \) with \( ||e'|| = C^*_\phi ||e|| \) such that:
   \[
   \phi(Ax + e) = \phi(Ax) + e'.
   \]

2. \( \exists \gamma : \mathbb{R}^t \to \mathbb{R}^n \) such that \( \Psi = \gamma \circ \phi \).

The smallest constant \( C^*_\phi \) for which the definition holds is defined as the stability constant of the stabilizer \( \phi \).

We also define the set:

\[
S^*_\psi = \{ \Psi \in \Psi : \exists \gamma : \mathbb{R}^t \to \mathbb{R}^n, \exists \phi \epsilon-stabilizer \text{ such that } \Psi = \gamma \circ \phi \}.
\]

Whenever \( t = m \) and \( \gamma : \mathbb{R}^m \to \mathbb{R}^n \) is a reconstructor, the reconstructor \( \Psi \) is said to be \( \epsilon \)-stabilized with respect to \( \gamma \).

Note that, in the definition of \( \epsilon \)-stabilizer, we only require a stability condition for \( \phi \) in the first item. Interestingly, given a reconstructor \( \Psi = \gamma \circ \phi \), we can estimate its \( \epsilon \)-stability constant \( C^*_\phi \) by means of the constant \( C^*_\phi \) and the local Lipschitz constant of \( \gamma \), as proved in the following proposition.

**Proposition 3.1.** Let \( \Psi : \mathbb{R}^m \to \mathbb{R}^n, \Psi = \gamma \circ \phi \), with \( \phi \) being an \( \epsilon \)-stabilizer. If \( C^*_\phi \) is the constant mentioned in **Definition 3.1**, \( L^*(\gamma, T) \) is the local Lipschitz constant of \( \gamma \) with \( T = \phi(Y) \), it holds:

\[
C^*_\psi \leq L^*(\gamma, T)\ C^*_\phi.
\]

**Proof.** Let \( x^{gt} \in X \) and \( ||e|| \leq \epsilon \). Then:

\[
||\Psi(Ax^{gt} + e) - x^{gt}|| = ||\gamma(\phi(Ax^{gt} + e)) - x^{gt}||.
\]

Since \( \phi \) is a stabilizer, \( \phi(Ax^{gt} + e) = \phi(Ax^{gt}) + e' \) with \( ||e'|| \leq C^*_\phi ||e|| \). Thus:

\[
||(\gamma(\phi(Ax^{gt} + e)) - x^{gt})|| = ||\gamma(\phi(Ax^{gt}) + e') - x^{gt}||
\]

\[
\leq ||\gamma(\phi(Ax^{gt}) + e') - \gamma(\phi(Ax^{gt}))|| + ||\gamma(\phi(Ax^{gt})) - x^{gt}||
\]

\[
\leq \eta + L^*(\gamma, T)||e'|| = \eta + L^*(\gamma, T)C^*_\phi ||e||,
\]

which implies that \( L^*(\gamma, T)C^*_\phi \) is a majorant of the set:

\[
\left\{ \frac{||\Psi(Ax^{gt} + e) - x^{gt}|| - \eta}{||e||} : x^{gt} \in X, ||e|| \leq \epsilon \right\}.
\]

Since \( C^*_\phi \) is defined as the supremum of the same set, by the minimality of the supremum we have \( C^*_\psi \leq L^*(\gamma, T)\ C^*_\phi \).

**Theorem 3.2.** For any \( \epsilon > 0, \eta_1, \eta_2 > 0 \), let \( \Psi_1 = \gamma_1 \circ \phi_1 \in S^*_\eta_1 \), and \( \Psi_2 \in R_{\eta_2} \). If:

\[
C^*_\phi_1 \in \left[ 0, \frac{C^*_\psi_2}{L^*(\gamma_1, T)} \right]
\]

then:

\[
C^*_\psi_1 \leq C^*_\psi_2.
\]

**Proof.** Since (9) holds by hypothesis and \( C^*_\psi_1 \leq L^*(\gamma_1, T)C^*_\phi_1 \), for \( C^*_\phi_1 \in S^*_\eta_1 \) by **Theorem 3.1** we get:

\[
C^*_\psi_1 \leq L^*(\gamma_1, T)C^*_\phi_1 \leq L^*(\gamma_1, T) \frac{C^*_\psi_2}{L^*(\gamma_1, T)} = C^*_\psi_2,
\]

which concludes the proof.

The theorem yields interesting consequences for the special case where \( \Psi_1 \) and \( \Psi_2 \) share the same accuracy. For instance, when \( \Psi_1 = \gamma_1 \circ \phi_1 \in S^*_\eta_1 \) and \( \Psi_2 \in R_{\eta_2} \), if (9) holds, the theorem suggests that \( \Psi_1 \) is preferable to \( \Psi_2 \), as \( \Psi_1 \) is more stable than \( \Psi_2 \). In addition, we can state the following result, whose proof is trivial.
Corollary 3.2.1. Let $\Psi_1 = \gamma \circ \phi_1 \in S_\eta$ and $\Psi_2 = \gamma \circ \phi_2 \in S_\eta$. If $[\square]$ holds, then $C_{\phi_1} \leq C_{\phi_2}$.

In the next proposition, we show a result linking the accuracy of a reconstructor $\Psi \in S_\eta$ to a characterization of its $\epsilon$-stabilizer.

Proposition 3.3. Let $\Psi = \gamma \circ \phi \in S_\eta$. Let:

$$\sigma(\phi) := \sup \{ ||x_1 - x_2|| : x_1, x_2 \in \mathcal{X}, \phi(Ax_1) = \phi(Ax_2) \}.$$  

Then:

$$\eta^{-1} \leq \frac{2}{\sigma(\phi)}.$$ 

Proof. Let $x_1, x_2 \in \mathcal{X}$ such that $\phi(Ax_1) = \phi(Ax_2)$. Then:

$$||x_1 - x_2|| \leq ||\phi(Ax_1) - x_1|| + ||\phi(Ax_1) - x_2|| = ||\phi(Ax_1) - x_1|| + ||\phi(Ax_2) - x_2|| \leq 2\eta,$$

which implies that:

$$\eta \geq \frac{||x_1 - x_2||}{2}.$$ 

Since the estimation above holds for any $x_1, x_2$ with $\phi(Ax_1) = \phi(Ax_2)$, it holds for $\sigma(\phi)$, thus concluding the proof.

As a consequence of Proposition 3.3 if $\phi$ is the constant operator (having $C_{\phi} = 0$ as observed in Example 2.2), it gets $\sigma(\phi) = \infty$, which implies that for any $\gamma$, the accuracy of $\Psi = \gamma \circ \phi$ will be zero, whenever $\mathcal{X}$ is unbounded.

Now, in the following proposition, we show that a sequence of functions $\{\phi_k\}_{k \in \mathbb{N}}$ approximating $\Psi \in \mathcal{R}_\eta$, i.e.:

$$\lim_{k \to \infty} \sup_{y^c \in \mathcal{Y}} ||\phi_k(y^c) - \Psi(y^c)|| = 0.$$ 

can be exploited to construct a good stabilizer.

Proposition 3.4. Given a reconstructor $\Psi : \mathbb{R}^m \to \mathbb{R}^n$ with local Lipschitz constant $L^* (\Psi, \mathcal{Y}) < 1$ and a sequence of functions $\{\phi_k\}_{k \in \mathbb{N}}$ approximating $\Psi$, there exists $K \in \mathbb{N}$ such that for any $k \geq K$, $C_{\phi_k} < 1$.

Proof. Consider $x^g \in \mathcal{X}$ and $e \in \mathbb{R}^m$ with $||e|| \leq \epsilon$. To prove the result, we need to show that:

$$\phi_k(Ax^g + e) = \phi_k(Ax^g) + e' \text{ for } k \geq K,$$

with $||e'|| = C_{\phi}||e||$ and $C_{\phi} \in [0, 1)$. Let $e' := \phi_k(Ax^g + e) - \phi_k(Ax^g)$, then:

$$||e'|| = ||\phi_k(Ax^g + e) - \phi_k(Ax^g)|| \leq L^*(\phi_k, \mathcal{Y})||Ax^g + e - Ax^g|| = L^*(\phi_k, \mathcal{Y})||e||,$$

which implies that $C_{\phi_k} \leq L^*(\phi_k, \mathcal{Y})$. Since $\{\phi_k\}_{k \in \mathbb{N}}$ is a sequence of approximators of $\Psi$, for any $k \in \mathbb{N}$ there is a constant $c_k$ such that $||\phi_k(y^c) - \Psi(y^c)|| \leq c_k$ and $c_k \to 0$ as $k \to \infty$. Consequently, it holds:

$$L^*(\phi_k, \mathcal{Y}) = \sup_{x^g \in \mathcal{X}, ||e|| \leq \epsilon} \frac{||\phi_k(Ax^g + e) - \phi_k(Ax^g)||}{||e||} \leq \sup_{x^g \in \mathcal{X}, ||e|| \leq \epsilon} \frac{||\phi_k(Ax^g + e) - \Psi(Ax^g + e)|| + ||\phi_k(Ax^g) - \Psi(Ax^g)|| + ||\Psi(Ax^g + e) - \Psi(Ax^g)||}{||e||}$$

$$\leq \sup_{x^g \in \mathcal{X}, ||e|| \leq \epsilon} \frac{||\Psi(Ax^g + e) - \Psi(Ax^g)||}{||e||} + 2c_k,$$

which implies that $L^*(\phi_k, \mathcal{Y}) \to L^*(\Psi, \mathcal{Y})$ as $k \to \infty$. Since $L^*(\Psi, \mathcal{Y}) < 1$, $\exists K \in \mathbb{N}$ such that for any $k \geq K$, $L^*(\phi_k, \mathcal{Y}) < 1$. For those values of $k$, $C_{\phi_k} \leq L^*(\phi_k, \mathcal{Y}) < 1$. 


3.2 Tikhonov stabilizers

If we now consider the Tikhonov reconstructor $\Psi = \Psi^{\lambda, L}$ introduced in Example 2.4, it is possible to construct a sequence $\{\phi_k\}_{k \in \mathbb{N}}$ of $\epsilon$-stabilizers. In fact, recalling that $L^*(\Psi, Y) < 1$ for suitable $\lambda > 0$ as stated in Proposition 2.4, a simple way to generate the sequence $\{\phi_k\}_{k \in \mathbb{N}}$ is the following. Consider a convergent iterative algorithm for the solution of (8):

$$
\begin{align*}
    & x^0 \in \mathbb{R}^n, \\
    & x^{k+1} = T_k(x^k, y^\delta),
\end{align*}
$$

where $T_k(x^k, y^\delta)$ models the application of the $k$-th iterate of the algorithm, starting from $x^k$ and with datum $y^\delta$. To set an example, the Conjugate Gradient for Least Squares (CGLS) algorithm is an iterative method solving the normal equations associated with (9). Now, for any $k \in \mathbb{N}$ we can define the Tikhonov stabilizers $\phi_k$ to be the composition of the first $k$ iterations of the algorithm, i.e.:

$$
\phi_k(\cdot) = \bigcap_{i=1}^{k} T_i(\cdot, y^\delta).
$$

For the convergence property of the algorithm, $\{\phi_k\}_{k \in \mathbb{N}}$ is a sequence of functions approximating $\Psi^{\lambda, L}$ and with $C^\epsilon_{\phi_k} < 1$ for suitable $k \geq K$. Such property will be fundamental for the stabilization technique we propose in Subsection 4.4.

4 Neural networks for the solution of linear inverse problems

In this section, the theoretical results previously outlined are applied to scenarios where reconstructoris are operational-ized through neural networks. Concurrently, we delineate our methodologies aimed at advancing current state-of-the-art approaches. Figure 2 offers a detailed schematic that encapsulates all the approaches considered within this study. The 'Tik' label refers to the Tikhonov reconstructor $\Psi^{\lambda, L}$, defined in Example 2.4.

4.1 Parameter-dependent families of reconstructoris

We now consider a family of reconstructoris $\{\Psi_\Theta\}_{\Theta \in \mathbb{R}^s}$, depending on a vector of parameters $\Theta$, approximating a reconstructor $\Psi$ to solve problem (1). We prove in the following theorem that the stability of $\Psi_\Theta$ is strongly related to the stability of $\Psi$. 
**Theorem 4.1** (Approximation Theorem for Reconstructors). Let $\Psi$ be an $\eta^{-1}$-accurate reconstructor and let $\{\Psi_\Theta\}_{\Theta \in \mathbb{R}^s}$ be a set of reconstructors with accuracy $\eta_\Theta$ for any $\Theta$. We define, for any $\Theta \in \mathbb{R}^s$:
\[
\Delta(\Theta) := \sup_{x^g \in \mathcal{X}} \| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \|
\]
and:
\[
\Delta_\epsilon(\Theta) := \sup_{x^g \in \mathcal{X}, \|e\| < \epsilon} \| \Psi_\Theta(Ax^g + e) - \Psi(Ax^g + e) \|.
\]

If $\Delta(\Theta) \to 0$ when $\Theta \to \Theta^*$, then:
\[
\lim_{\Delta(\Theta) \to 0} \eta_\Theta = \eta.
\]  

Moreover, if $\Delta_\epsilon(\Theta) \to 0$ when $\Theta \to \Theta^*_\epsilon$, then:
\[
\lim_{\Delta_\epsilon(\Theta) \to 0} C^\epsilon_{\Psi_\Theta} = C^\epsilon_\Psi.
\]

**Proof.** Consider $x^g \in X$. Since:
\[
\| \Psi_\Theta(Ax^g) - x^g \| \leq \| \Psi(Ax^g) - x^g \| + \| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \|
\]
and:
\[
\| \Psi_\Theta(Ax^g) - x^g \| \geq \| \Psi(Ax^g) - x^g \| - \| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \|,
\]
it holds that:
\[
\| \| \Psi_\Theta(Ax^g) - x^g \| - \| \Psi(Ax^g) - x^g \| \| \leq \| \| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \| \|
\]
which implies that $\| \Psi_\Theta(Ax^g) - x^g \| \to \| \Psi(Ax^g) - x^g \|$ as $\Theta(\Theta) \to 0$ and consequently, $\eta_\Theta \to \eta$ as $\Delta(\Theta) \to 0$.

Now, consider $\epsilon > 0$ and $e \in \mathbb{R}^m$ with $\|e\| \leq \epsilon$. A similar computation shows that:
\[
\| \| \Psi_\Theta(Ax^g + e) - x^g \| - \| \Psi(Ax^g + e) - x^g \| \| \leq \Delta_\epsilon(\Theta),
\]
which implies that $\| \Psi_\Theta(Ax^g + e) - x^g \| \to \| \Psi(Ax^g + e) - x^g \|$ for $\Delta_\epsilon(\Theta) \to 0$. Consequently, for $\Delta_\epsilon(\Theta) \to 0$,
\[
C^\epsilon_{\Psi_\Theta} = \sup_{x^g \in \mathcal{X}, \|e\| \leq \epsilon} \| \Psi_\Theta(Ax^g + e) - x^g - \eta \| \to \sup_{x^g \in \mathcal{X}, \|e\| \leq \epsilon} \| \Psi(Ax^g + e) - x^g - \eta \| = C^\epsilon_\Psi,
\]
which concludes the proof. \qed

**Corollary 4.1.1.** For any $\Theta \in \mathbb{R}^s$, it holds:
\[
\eta_\Theta \leq \eta + \Delta(\Theta).
\]

**Proof.** Consider $x^g \in \mathcal{X}$. Then:
\[
\| \Psi_\Theta(Ax^g) - x^g \| \leq \| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \| + \| \Psi(Ax^g) - x^g \|.
\]
Since $\| \Psi_\Theta(Ax^g) - \Psi(Ax^g) \| \leq \Delta(\Theta)$ by hypothesis and $\| \Psi(Ax^g) - x^g \| \leq \eta$ since $\Psi$ is $\eta^{-1}$-accurate, then:
\[
\| \Psi_\Theta(Ax^g) - x^g \| \leq \eta + \Delta(\Theta),
\]
which shows that $\eta_\Theta \leq \eta + \Delta(\Theta)$. \qed

Note that $\Delta(\Theta)$ and $\Delta_\epsilon(\Theta)$ are, in general, not independent, as proved in the following proposition.

**Proposition 4.2.** For any $\epsilon > 0$, let $\Delta(\Theta)$ and $\Delta_\epsilon(\Theta)$ be the quantities defined in Theorem 4.1. Then:
\[
\Delta(\Theta) \leq \Delta_\epsilon(\Theta).
\]

**Proof.** Observe that, by definition of $\mathcal{Y}$ and $\mathcal{Y}^\epsilon$, $\Delta(\Theta)$ and $\Delta_\epsilon(\Theta)$ can be rewritten as:
\[
\Delta(\Theta) = \sup_{y \in \mathcal{Y}} \| \Psi_\Theta(y) - \Psi(y) \|,
\]
\[
\Delta_\epsilon(\Theta) = \sup_{y \in \mathcal{Y}^\epsilon} \| \Psi_\Theta(y) - \Psi(y) \|,
\]
where $\mathcal{Y}^\epsilon = \{ y + e : y \in \mathcal{Y}, \|e\| \leq \epsilon \} \supseteq \mathcal{Y}$. The result follows from the property that the supremum of a set must be larger than the supremum of its subsets. \qed
An insight on the stability properties of $\Psi_\Theta$ can be obtained by the following proposition.

**Proposition 4.3.** Let $\Psi_\Theta$ be a reconstructor parameterized by $\Theta \in \mathbb{R}^s$, approximating a reconstructor $\Psi$ with error $\Delta(\Theta) > 0$. Let $\eta_0^{-1}$ and $\eta^{-1}$ be the accuracy of $\Psi_\Theta$ and $\Psi$, respectively. If:

$$\Delta(\Theta) \leq \bar{\eta}(A, \epsilon, \lambda) - \eta$$

for a fixed $\epsilon > 0$, where $\bar{\eta}(A, \epsilon, \lambda)$ is the constant defined in Corollary 2.2.2, then $C_{\Psi_\Theta} \geq 1$.

**Proof.** Let $\epsilon > 0$ be fixed. By Corollary 4.1, the accuracy of $\Psi_\Theta$ can be estimated as $\eta_0 \leq \eta + \Delta(\Theta)$. Consequently, by Corollary 2.2.2 if $\eta + \Delta(\Theta) \leq \bar{\eta}(A, \epsilon, \lambda)$, then $\eta_0 \leq \bar{\eta}(A, \epsilon, \lambda)$, which implies that $C_{\Psi_\Theta} \geq 1$.

In the following paragraphs, we will analyze two particular families of reconstructors $\{\Psi_\Theta\}_{\Theta \in \mathbb{R}^s}$.

### 4.2 Neural Networks as reconstructors: the NN approach

Now we consider the set of neural networks defined by a fixed architecture as the family $\{\Psi_\Theta\}_{\Theta \in \mathbb{R}^s}$.

**Definition 4.1.** Given a neural network architecture $\mathcal{A} = (\nu, S)$ where $\nu = (\nu_0, \nu_1, \ldots, \nu_L) \in \mathbb{N}^{L+1}$, $\nu_0 = m$, $\nu_L = n$, defines the width of each layer and $S = (S_{1,1}, \ldots, S_{L,L})$, $S_{j,k} \in \mathbb{R}^{\nu_j \times \nu_k}$ is the set of matrices representing the skip connections, we define the parametric family of neural network reconstructors with architecture $\mathcal{A}$, parameterized by $\Theta \in \mathbb{R}^s$, as

$$\mathcal{F}_\mathcal{A}^\mathcal{A} = \{\Psi_\Theta : \mathbb{R}^m \rightarrow \mathbb{R}^n; \Theta \in \mathbb{R}^s\},$$

where $\Psi_\Theta(y^\delta) = z^L$ is given by:

$$\begin{cases}
  z^0 = y \\
  z^{l+1} = \rho(W^l z^l + b^l + \sum_{k=1}^{L} S_{l,k} z^k) \quad \forall l = 0, \ldots, L - 1
\end{cases}$$

(15)

and $W^l \in \mathbb{R}^{\nu_{l+1} \times \nu_l}$ is the weights matrix, $b^l \in \mathbb{R}^{\nu_{l+1}}$ is the bias vector.

Given $\mathcal{D} \subseteq \mathcal{X}$, consider the dataset $\mathcal{D} = \{(y_i^\delta, x_i^\delta); x_i^\delta \in \mathcal{D}\}_{i=1}^{N_0}$ of images according to (12). Training a neural network to solve the inverse problem (2) results in finding the parameters $\Theta^*$ such that the associated reconstructor $\Psi_{\Theta^*} \in \mathcal{F}_\mathcal{A}^\mathcal{A}$ satisfies:

$$\Psi_{\Theta^*} \in \arg\min_{\Psi_\Theta \in \mathcal{F}_\mathcal{A}^\mathcal{A}} \frac{1}{N_0} \sum_{i=1}^{N_0} \ell(\Psi_\Theta(y_i^\delta), x_i^\delta),$$

(16)

where $\delta \geq 0$ and $\ell : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_+$ is the loss function.

In this work, we consider as reconstructors $\Psi_\Theta$ the neural networks trained with the Mean Squared Error (MSE) loss. We will name this family as NN, in the following. We first apply NN onto noiseless data ($\delta = 0$), thereby (16) corresponds to:

$$\min_{\Psi_\Theta \in \mathcal{F}_\mathcal{A}^\mathcal{A}} \sum_{i=1}^{N_0} \|\Psi_\Theta(y_i^\delta) - x_i^{\delta}\|_2^2 = \min_{\Psi_\Theta \in \mathcal{F}_\mathcal{A}^\mathcal{A}} \sum_{i=1}^{N_0} \|\Psi_\Theta(Ax_i^{\delta}) - \Psi^\dagger(Ax_i^{\delta})\|_2^2,$$

(17)

which results in the minimization of $\Delta(\Theta)$ as introduced in Theorem 4.1 with $\Psi = \Psi^\dagger$.

We observe that when $A$ is ill-conditioned, $\bar{\eta}(A, \epsilon, \lambda)$ is large. This becomes particularly apparent when $\mathcal{X} = \mathbb{R}^n$, as under these circumstances, $\bar{\eta}(A, \epsilon, \lambda)$ is bounded below by a quantity depending on $C(A) = \frac{1}{\sigma_n}$. Additionally, the value of $\Delta(\Theta^*)$ derived from NN training likely meets the established inequality in Proposition 4.3, which leads to instability. This confirms that effective neural network training can produce a very accurate but unstable reconstructor $\Psi_\Theta$.

A widely adopted strategy to bolster robustness in neural networks is known as noise injection. This technique involves adding noise to the input of the network during its training phase. In this context, the set of reconstructors $\Psi_\Theta$, referred to as iNN, is defined by a neural network trained through the following equation:

$$\min_{\Psi_\Theta \in \mathcal{F}_\mathcal{A}^\mathcal{A}} \sum_{i=1}^{N_0} \|\Psi_\Theta(y_i^\delta) - x_i^{\delta}\|_2^2,$$

(18)

where $\delta > 0$. Research detailed in (10) has demonstrated that this approach effectively introduces a Tikhonov regularization term into the loss function. Although this technique, as described in (4), enhances the stability of the resultant network, the impact of noise injection on the accuracy of the model remains somewhat ambiguous. Furthermore, the optimal amount of noise to be added to each input to optimize the balance between stability and accuracy is still a subject of investigation.
4.3 Regularized NN-based reconstructors: the ReNN approach

To develop a reconstructor with improved stability compared to standard neural networks (NN), we harness the properties of Tikhonov regularization. It is important to note that a Tikhonov regularized reconstructor $\Psi^{\lambda,L}$ achieves stability for an appropriately chosen regularization parameter, as delineated in Proposition 2.4. This methodology will be referred to as the Regularized Neural Network (ReNN), denoted as $\Psi^{\lambda,L}_{\Theta}$. ReNN is defined by training a neural network with a new loss $\Psi^{\lambda,L}_\Theta$ as:

$$\Psi^{\lambda,L}_\Theta = \arg\min_{\Psi_\Theta \in \mathcal{F}_{\Theta}} \sum_{i=1}^{N_0} ||\Psi_\Theta (y_i^\delta) - \Psi^{\lambda,L}_\Theta (y_i^\delta)||_2^2, \quad (19)$$

with $\delta > 0$. We underline that ReNN does not require any ground-truth solutions $x^\delta_{gt}$ since the target is computed from the corrupted datum $y^\delta$ via the Tikhonov-regularized reconstructor. Furthermore, in the training of ReNN, noise is present not solely to the input of the neural network model, as is the case with iNN, but also to the input of the Tikhonov-regularized reconstructor, which is responsible for generating the target. In the following, we consider for simplicity the case $X = \mathbb{R}^n$, but similar results hold for a general $X \subset \mathbb{R}^n$.

Starting from inequality (14) it is easy to notice that (19) corresponds to the minimization of $\Delta_\epsilon(\Theta)$ in Theorem 4.1. Moreover, by Theorem 4.2, if $\Delta_\epsilon(\Theta)$ is small, as it is common when $\Psi_\Theta$ is a neural network, then $\Delta(\Theta) \in [0, \Delta_\epsilon(\Theta)]$ is also small. Regarding the right hand side $\eta(A, \epsilon, X') - \eta$ of (14), it is noted that in this instance $\eta = \eta(\lambda)$ and $\eta(\lambda) \to \infty$ for $\lambda \to \infty$. Consequently, for sufficiently large values of $\lambda$, it is probable that ReNN does not fulfill the conditions of (14).

Moreover, minimizing $\Delta_\epsilon(\Theta)$ is crucial for enforcing the method’s stability, as proven by Theorem 4.1, where we have shown that in our hypothesis the stability constant $C^\epsilon_{\Psi^{\lambda,L}_{\Theta}} < 1$ for sufficiently small $\Delta_\epsilon(\Theta)$. Hence, effective training of ReNN should produce an accurate and stable reconstructor. The pseudocode to compute $\Psi^{\lambda,L}_{\Theta}$ is given in Algorithm 1.

**Algorithm 1** Regularized Neural Network (ReNN)

```plaintext
input a collection $\{x_i^{gt}\}_{i=1}^{N_0} \subseteq \mathcal{X}$ of data points, a noise level $\delta > 0$, $A \in \mathbb{R}^{m \times n}$ and a stable reconstructor $\Psi^{\lambda,L}$
for $i \leftarrow 1 : N_0$
do
Sample $e_i \sim \mathcal{N}(0, \sigma^2 I)$
Compute $y_i^\delta \leftarrow Ax_i^{gt} + e_i$
end for
Solve
$$\min_{\Psi_\Theta \in \mathcal{F}_{\Theta}} \sum_{i=1}^{N_0} ||\Psi_\Theta (y_i^\delta) - \Psi^{\lambda,L}_\Theta (y_i^\delta)||_2^2.$$ run a trained ReNN $\Psi^{\lambda,L}_{\Theta}$
```

4.4 Stabilization on NN and ReNN: St- approaches

In the remainder of this section, we discuss an application of the stabilizers, introduced in Section 3, to improve the stability of neural network-based reconstructors. We propose new reconstructors $\Psi \in \mathcal{S}^\Theta$, $\Psi = \gamma \circ \phi$ where $\gamma$ is a neural network based reconstructor. In particular, we consider $\phi$ as the Tikhonov $\epsilon$-stabilizer $\phi_k$ defined in Subsection 3.2 and obtained by $k$ iterations of the CGLS algorithm, with a suitable $k$. When $\gamma$ is chosen as NN, iNN, ReNN we obtain the $\epsilon$-stabilized reconstructors $\text{StNN}$, $\text{StiNN}$, and $\text{StReNN}$, respectively.

Note that, in this case, we can apply Theorem 3.2 with $\Psi_1 = \gamma \circ \phi_k$ and $\Psi_2 = \gamma$, and whenever we choose $\phi_k$ such that:

$$C^\epsilon_{\phi_k} \leq \frac{C^\epsilon_{\gamma}}{C^\epsilon_{\gamma}}, \quad (20)$$

the $\epsilon$-stabilized reconstructor $\Psi_1$ gets more stable than its unstabilized version $\Psi_2$. We remark that it is always possible to find a Tikhonov stabilizer $\phi_k$ fitting (20), by suitably tuning $\lambda$ and $k$. Clearly, this comes at the expense of accuracy as discussed in Proposition 3.3, but we will show that the accuracy does not suffer excessively, as evidenced by empirical results in Section 6.
5 Experimental setup

To assess the theoretical issues proposed, we conducted a series of experiments. It is important to highlight that all tests were carried out utilizing the same end-to-end U-net architecture. For details on the architecture and its training, you can refer to [37, 19]. In the following experiments, the stabilizer applied to all the considered reconstructors is obtained with $k = 3$ iterations of the CGLS algorithm on (8). The codes can be found in our GitHub repository at https://github.com/loibo/ToBeOrNotToBeStable.

As a test case, we consider image deblurring [30], a common inverse problem in imaging. In this case, $A$ is a block circulant matrix with circulant blocks obtained from a convolutional kernel with periodic boundary conditions [30]. In our experiments, we use the $11 \times 11$ Gaussian blur filter $K$:

$$K_{i,j} = e^{-\frac{i^2 + j^2}{\sigma^2}}, \quad i, j \in \{-5, \ldots, 5\}$$

with variance $\sigma^2 = 1.3$.

5.1 Dataset

Our results have been tested on the famous GoPro image dataset (https://seungjunnah.github.io/Datasets/goopro), introduced in [41], which is constituted by high-resolution RGB images. All the images have been cropped into patches of size $256 \times 256$ (without overlapping), converted into grayscale, normalized in $[0, 1]$, and labeled as $x^g_i$, $i = 1, \ldots, N_D$ with $N_D = 3614$. We generated the blurred and noisy data $y^\delta_i = Ax^g_i + e$, where $e \sim N(0, \delta^2 I)$. We need the following data sets to train the three considered neural networks-based reconstructors.

- For the NN training (see (17)) we consider the set $\mathbb{D} = \{(y^\delta_i, x^g_i)\}_{i=1}^{N_D}$ containing the couples of images constituted by the blurred noiseless datum (i.e. $\delta = 0$) and the exact $x^g_i$ target picture.

- For the iNN training (see (18)) we consider the set $\mathbb{D}_\delta = \{(y^\delta_i, x^g_i)\}_{i=1}^{N_D}$ containing the couples of images constituted by the blurred and noisy datum $y^\delta_i$ and the exact $x^g_i$ target picture.

- For the ReNN training (see (19)) we consider the set $\mathbb{D}^{\lambda,L}_\delta = \{(y^\delta_i, \Psi^{\lambda,L}(y^\delta_i))\}_{i=1}^{N_D}$ containing the couples of images constituted by the blurred and noisy datum $y^\delta_i$ and the target image computed by the Tikhonov reconstructor (using $L = I$ in (8)). In particular, we choose $\lambda$ heuristically and we computed $\Psi^{\lambda,L}(y^\delta_i)$ by means of the CGLS algorithm [28] to solve the normal equations of (8).

We finally split the $N_D$ data samples into train and test subsets, with $N_{\text{train}} = 2503$ and $N_{\text{test}} = 1111$.

5.2 Results evaluation

In order to estimate in our experiments the accuracy and the stability constants of a given reconstructor $\Psi$ we compute the empirical accuracy $\hat{\eta}^{-1}$ and the empirical stability constant $\hat{C}_\Psi$, over the test set $\mathcal{T}S$. They are respectively defined as:

$$\hat{\eta} = \sup_{x^g \in \mathcal{T}S} ||\Psi(Ax^g) - x^g||$$

and:

$$\hat{C}_\Psi = \sup_{x^g \in \mathcal{T}S} \frac{||\Psi(Ax^g + e) - x^g||}{||e||}$$

where $e \sim N(0, \delta^2 I)$ differs for each datum $x^g \in \mathcal{T}S$). Finally, we compute the empirical reconstruction error on the test set as:

$$\mathcal{E}(\Psi, \delta) = \sup_{x^g \in \mathcal{T}S} ||\Psi(y^\delta) - x^g||.$$
To be or not to be stable, that is the question: understanding neural networks for inverse problems

Figure 3: Results obtained by the NN and StNN reconstructors on a single test image $y^\delta$ with $\delta = 0$ (first row) and $\delta = 0.01$ (second row). The ground truth clean image is also reported for reference.

6 Numerical Results

In this section, we present the outcomes achieved in terms of empirical accuracy, stability, and reconstruction error for the solvers proposed in this study. The objective of this section is twofold: firstly, to validate the key theoretical findings established in the previous part of the paper, with a particular emphasis on the deep learning-based reconstructors introduced in Section 4; and secondly, to examine the impact of the stabilizer in scenarios where the noise levels exceed those the parameters were initially selected for.

6.1 Results with NN-based reconstructors

We begin by considering the NN and iNN approaches, and their stabilized counterparts, StNN and StiNN, assuming the availability of ground truth images $x^{gt}_i, i = 1, \ldots, N_D$.

The first experiment concerns NN and StNN. The first row of Figure 3 shows the reconstructions obtained with both the methods on one image $y_i$ from the test set (without noise added). To assess the stability of our frameworks concerning unseen noise on the data, we also tested the NN reconstructor on noisy images $y^\delta_i = y_i + e_i$ with $e_i \sim N(0, \delta^2 I)$ and $\delta = 0.01$. The second row of Figure 3 displays the reconstructions obtained on the same test image.
To be or not to be stable, that is the question: understanding neural networks for inverse problems

Table 2: Values of empirical accuracy and stability constant obtained for iNN and StiNN reconstructors, trained on noisy data with $\delta = 0.025$ and tested with different values of $\delta$.

| $\hat{\eta}$ | iNN  | StiNN |
|------------|------|-------|
| $\hat{\eta}^{-1}$ | 0.0707 | 0.0606 |
| $C^\epsilon_\Psi (\delta = 0.025)$ | 0.0899 | 0.0703 |
| $C^\epsilon_\Psi (\delta = 0.060)$ | 0.4309 | 0.2122 |
| $C^\epsilon_\Psi (\delta = 0.125)$ | 0.8385 | 0.6215 |

Table 1: Values of empirical accuracy and $\epsilon$-stability constant obtained by NN and StNN reconstructors, trained with $\delta = 0.0$.

|          | NN    | StNN  |
|----------|-------|-------|
| $\hat{\eta}$ | 0.1203 | 0.0616 |
| $C^\epsilon_\Psi (\delta = 0.01)$ | 36.7298 | 0.1579 |

Figure 4: Plots of the empirical error yielded by NN and StNN reconstructors for increasing values of $\delta$ in the test images.

From the images presented in Figure 3 and their SSIM values, it is observable that the NN reconstructor excels in restoring the blurred image, yet it demonstrates its unreliability as soon as even a minimal amount of noise is added to the data. In contrast, StNN emerges as an effective compromise between accuracy, as evidenced by the high-quality image in the first row with noise-free data, and stability, highlighted by the superior quality of the StNN image compared to the NN one in the second row under noisy conditions. The Table 1 reports the values of the empirical accuracy $\hat{\eta}^{-1}$ and empirical stability constant $C^\epsilon_\Psi$ for the considered methods on the whole test set. It confirms that there is a trade-off between accuracy and stability, as proved in Theorem 2.2, and that the stabilization strategy improves the value of $C^\epsilon_\Psi$ for NN.

To further investigate the different behavior of the two reconstructors for increasing values of $\delta$, in Figure 4 we plot the reconstruction error for $\delta \in [0, 0.03]$. The value of $\delta = 0$ used in the training is indicated with a star marker. We note that the StNN curve is characterized by a notably flat trajectory, in contrast to the NN curve which exhibits a rapid increase. This observation aligns with and reinforces the insights gathered from previous analyses.

In the second experimental setting, we considered the iNN reconstructor, trained by (18), with $\delta = 0.025$. Table 2 reports the empirical accuracy and stability computed for both iNN and its stabilized version, StiNN, when the methods are tested on noisy data with $\delta = 0.025, 0.060, 0.125$, respectively. The table shows that injecting noise in the observed data during training produces slightly less accurate but far more stable reconstructors (as visible by comparing the results with unseen noise in Table 2 to those in Table 1).

### 6.2 Results with ReNN-based reconstructors

In this subsection, we focus on the application of the proposed ReNN reconstructor and its stabilized variant StReNN on noisy data characterized by $\delta = 0.025$. It is important to recall that ReNN is trained following the methodology outlined in (19) and utilizes a dataset that does not include the exact $x^{gt}$ images.

The target images are the output of Tikhonov reconstructor $\Psi^{\lambda}L$ applied to the data $y^\delta_1, i = 1 \ldots N_D$. The Tikhonov regularization parameter $\lambda$ has been heuristically chosen as $\lambda = 0.31$ to obtain a small reconstruction error on the training set. The methods have been tested on noisy data with $\delta = 0.025, 0.060, 0.125$, respectively. The outcomes obtained in terms of accuracy and stability constants are reported in Table 3. In the final column of this table, we also include the metrics pertaining to the Tikhonov reconstructor. It is observed that the accuracy of the three methods is
 quite comparable. Notably, the stability of the regularized NN-based reconstructors surpasses that of the Tikhonov method. Furthermore, the application of stabilization to ReNN exhibits increasingly beneficial effects as the noise level in the data escalates, as evidenced in the table’s last row.

### 6.3 Comparison among reconstructors

In this final subsection we provide an overview of the results and compare the NN-based reconstructors with the ReNN-based ones. Figure 5 shows the output images of the reconstructors trained on noisy data $y^\delta_i$, $\delta = 0.025$ and tested on noisy data with $\delta = 0.060$. As previously observed, the stabilization technique is effective as demonstrated by the image quality and the SSIM value. Interestingly, comparing iNN and ReNN we observe that the ReNN output images inherit smoothness from the regularized images used as target in (19), and exhibits a higher SSIM. At last, ReNN also outperforms Tikhonov reconstructor in terms of SSIM.

In Figure 6a we plot the reconstruction error of the methods for increasing value of $\delta \in [0, 0.1]$. The value of $\delta = 0.025$ used in the training is indicated with a star marker. It is discernible that the blue iNN curve demonstrates a markedly steeper gradient, commencing from the minimal error value and escalating to the maximal. The red plot, representing StiNN, intersects the blue iNN curve at approximately $\delta = 0.055$, indicating a more stable behavior at higher noise levels. The remaining three curves, corresponding to the regularized approaches ReNN, StReNN, and Tikhonov, exhibit similar slopes and behaviors. They manifest elevated errors for smaller values of $\delta$, yet surpass the performance of iNN when $\delta > 0.07$, yielding results comparable to those of StiNN. Finally, Figure 6a presents the boxplots of the experimental accuracy achieved across $T = 20$ executions with varied random realizations. The limited variance in these plots indicates that the values of $\hat{\eta}^{-1}$ are remarkably consistent for each individual reconstructor, thereby affirming the robustness of our accuracy definition.

### 7 Conclusions

In this paper, we conducted a comprehensive theoretical analysis of a broad spectrum of reconstructors for addressing a discrete ill-posed inverse problem with noisy data. Our findings, particularly encapsulated in Theorem 2.2, establish...
that enhancing stability in these reconstructors invariably leads to a decrease in accuracy. Our focus was primarily on reconstructors that leverage neural networks.

In consideration of the trade-off theorem, our objective was to enhance the stability of reconstructors based on deep learning, while preserving their accuracy as much as possible. We based our analysis on the reconstructors represented by the popular end-to-end NN approach for image restoration and we also considered the extensively utilized noise injection stabilization technique, here referred to as iNN. As is commonly understood, these approaches are trained using datasets that include images with known ground truth.

We have proposed new deep learning-based approaches: (i) an additional reconstructor, ReNN, which is trained on noisy images and increases the stability of NN by inheriting regularization from a model-based scheme in its training; (ii) a stabilization technique which stabilizes the solving process by reducing the impact of the noise with few iterations of a model-based algorithm and it is applied to all the proposed reconstructors resulting in StNN, StiNN and StReNN.

We performed extensive numerical experiments on image deblurring and denoising, with results serving to substantiate the theoretical framework presented in our study. Firstly, we observe, from Table 1, Table 2 and Table 3, that the introduction of the proposed stabilizers reduces the stability constants of 99.6% in StNN, of about 50% in StiNN and StReNN, with a minimal accuracy loss of about 10−20% in StiNN and StReNN and of about 50% in StNN. Secondly, in cases where only noisy data are available and ground truth images are not accessible, the ReNN approach performs exceptionally well and represents a more stable alternative compared to the Tikhonov reconstructor, as demonstrated by Figure 3 and Figure 6 ReNN outperforms even NN when noise impacts the data.

We believe that this new approach for solving noisy linear inverse problems with stable deep learning-based tools is relevant in this field. It can be further theoretically extended to more general problems and formally applied in real imaging applications, as, for example, in [20].
To be or not to be stable, that is the question: understanding neural networks for inverse problems

**Funding**

This work has been partially supported by the GNCS - Gruppo Nazionale per il Calcolo Scientifico ["Apprendimento automatico e tecniche variationali per la tomografia” INdAM GNCS Project, grant code CUP,E55F55000270001] and by the U.S. National Science Foundation, grant codes DMS-2038118 and DMS-2208294.

**References**

[1] Ben Adcock and Nick Dexter. The gap between theory and practice in function approximation with deep neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):624–655, 2021.

[2] Jaweria Amjad, Jure Sokolić, and Miguel RD Rodrigues. On deep learning for inverse problems. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1895–1899. IEEE, 2018.

[3] Vegard Antun, Francesco Renna, Clarice Poon, Ben Adcock, and Anders C Hansen. On instabilities of deep learning in image reconstruction and the potential costs of ai. *Proceedings of the National Academy of Sciences*, 117(48):30088–30095, 2020.

[4] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[5] Richard Baraniuk, Mark A Davenport, Marco F Duarte, Chinmay Hegde, et al. An introduction to compressive sensing. *Connexions e-textbook*, pages 24–76, 2011.

[6] Johnathan M Bardsley, Sarah Knepper, and James Nagy. Structured linear algebra problems in adaptive optics imaging. *Advances in Computational Mathematics*, 35(2):103–117, 2011.

[7] Alexander Bastounis, Anders C Hansen, and Verner Vlačić. The mathematics of adversarial attacks in ai–why deep learning is unstable despite the existence of stable neural networks. *arXiv preprint arXiv:2109.06098*, 2021.

[8] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, 2021.

[9] Mario Bertero, Patrizia Boccacci, and Christine De Mol. *Introduction to inverse problems in imaging*. CRC press, 2021.

[10] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

[11] Alessandro Buccini and Lothar Reichel. An lp-lq minimization method with cross-validation for the restoration of impulse noise contaminated images. *Journal of Computational and Applied Mathematics*, 375:112824, 2020.

[12] Alessandro Buccini and Lothar Reichel. Generalized cross validation for lp-lq minimization. *Numerical Algorithms*, 88(4):1595–1616, 2021.

[13] Pasquale Cascarano, Elena Loli Piccolomini, Elena Morotti, and Andrea Sebastiani. Plug-and-play gradient-based denoisers applied to ct image enhancement. *Applied Mathematics and Computation*, 422:126967, 2022.

[14] Qing Chu, Stuart Jefferies, and James G Nagy. Iterative wavefront reconstruction for astronomical imaging. *SIAM Journal on Scientific Computing*, 35(5):S84–S103, 2013.

[15] Matthew J Colbrook, Vegard Antun, and Anders C Hansen. Can stable and accurate neural networks be computed?–on the barriers of deep learning and smale’s 18th problem. *arXiv preprint arXiv:2101.08286*, 2021.

[16] Mohammad Zalbargi Darestanti, Akshay S Chaudhari, and Reinhard Heckel. Measuring robustness in deep learning based compressive sensing. In *International Conference on Machine Learning*, pages 2433–2444. PMLR, 2021.

[17] Heinz W. Engl, Martin Hanke, and Andreas Neubauer. Regularization of inverse problems. 1996.

[18] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[19] Davide Evangelista, Elena Morotti, and Elena Loli Piccolomini. Rising: A new framework for model-based few-view CT image reconstruction with deep learning. *Computerized Medical Imaging and Graphics*, 103:102156, 2023.

[20] Davide Evangelista, Elena Morotti, Elena Loli Piccolomini, and James Nagy. Ambiguity in solving imaging inverse problems with deep-learning-based operators. *Journal of Imaging*, 9(7), 2023.
To be or not to be stable, that is the question: understanding neural networks for inverse problems

[21] Zalan Fabian, Reinhard Heckel, and Mahdi Soltanolkotabi. Data augmentation for deep learning based accelerated mri reconstruction with limited data. In International Conference on Machine Learning, pages 3057–3067. PMLR, 2021.

[22] Martin Genzel, Jan Macdonald, and Maximilian Marz. Solving inverse problems with deep neural networks-robustness included. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022.

[23] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572, 2014.

[24] Nina M Gotschling, Vegard Antun, Ben Adcock, and Anders C Hansen. The troublesome kernel: why deep learning for inverse problems is typically unstable. arXiv preprint arXiv:2001.01258, 2020.

[25] 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.

[26] Martin Hanke. Limitations of the l-curve method in ill-posed problems. BIT Numerical Mathematics, 36(2):287–301, 1996.

[27] Per Christian Hansen. Analysis of discrete ill-posed problems by means of the l-curve. SIAM review, 34(4):561–580, 1992.

[28] Per Christian Hansen. Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. SIAM, 1998.

[29] Per Christian Hansen. Discrete inverse problems: insight and algorithms. SIAM, 2010.

[30] Per Christian Hansen, James G Nagy, and Dianne P O’leary. Deblurring images: matrices, spectra, and filtering. SIAM, 2006.

[31] Yixing Huang, Alexander Preuhs, Günter Lauritsch, Michael Manhart, Xiaolin Huang, and Andreas Maier. Data consistent artifact reduction for limited angle tomography with deep learning prior. In International workshop on machine learning for medical image reconstruction, pages 101–112. Springer, 2019.

[32] Yixing Huang, Tobias Würfl, Katharina Breininger, Ling Liu, Günter Lauritsch, and Andreas Maier. Some investigations on robustness of deep learning in limited angle tomography. In Medical Image Computing and Computer Assisted Intervention--MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part I, pages 145–153. Springer, 2018.

[33] Chang Min Hyun, Seong Hyeon Baek, Mingyu Lee, Sung Min Lee, and Jin Keun Seo. Deep learning-based solvability of underdetermined inverse problems in medical imaging. Medical Image Analysis, 69:101967, 2021.

[34] Patricia M Johnson, Geunu Jeong, Kerstin Hammernik, Jo Schlemper, Chen Qin, Jinming Duan, Daniel Rueckert, Jingyu Ko, Nicola Pezzotti, Elwin De Weerdt, et al. Evaluation of the robustness of learned mr image reconstruction to systematic deviations between training and test data for the models from the fastmri challenge. In Machine Learning for Medical Image Reconstruction: 4th International Workshop, MLMIR 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings 4, pages 25–34. Springer, 2021.

[35] Chao Ma, Stephan Wojtowytsch, Lei Wu, et al. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don’t. arXiv preprint arXiv:2009.10713, 2020.

[36] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2574–2582, 2016.

[37] Elena Morotti, Davide Evangelista, and Elena Loli Piccolomini. A green prospective for learned post-processing in sparse-view tomographic reconstruction. Journal of Imaging, 7(8):139, 2021.

[38] Jan Nikolas Morshuis, Sergios Gatidis, Matthias Hein, and Christian F Baumgartner. Adversarial robustness of mr image reconstruction under realistic perturbations. In Machine Learning for Medical Image Reconstruction: 5th International Workshop, MLMIR 2022, Held in Conjunction with MICCAI 2022, Singapore, September 22, 2022, Proceedings, pages 24–33. Springer, 2022.

[39] Matthew J Muckley, Bruno Riemenschneider, Alireza Radmanesh, Sunwoo Kim, Geunu Jeong, Jingyu Ko, Yohan Jun, Hyungseob Shin, Dosik Hwang, Mahmoud Mostapha, et al. Results of the 2020 fastmri challenge for machine learning mr image reconstruction. IEEE transactions on medical imaging, 40(9):2306–2317, 2021.

[40] Jennifer L Mueller and Samuli Siltanen. Linear and nonlinear inverse problems with practical applications. SIAM, 2012.
[41] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In CVPR, 07 2017.

[42] Daniel Obmann, Linh Nguyen, Johannes Schwab, and Markus Haltmeier. Augmented nett regularization of inverse problems. *Journal of Physics Communications*, 5(10):105002, 2021.

[43] Arghya Pal and Yogesh Rathi. A review and experimental evaluation of deep learning methods for mri reconstruction. *The journal of machine learning for biomedical imaging*, 1, 2022.

[44] Allan Pinkus. Approximation theory of the mlp model in neural networks. *Acta numerica*, 8:143–195, 1999.

[45] Lothar Reichel and Giuseppe Rodriguez. Old and new parameter choice rules for discrete ill-posed problems. *Numerical Algorithms*, 63(1):65–87, 2013.

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

[47] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, and Frank Lenzen. Variational methods in imaging. 2009.

[48] Efrat Shimron, Jonathan I Tamir, Ke Wang, and Michael Lustig. Implicit data crimes: Machine learning bias arising from misuse of public data. *Proceedings of the National Academy of Sciences*, 119(13):e2117203119, 2022.

[49] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[50] Andrei Nikolaevich Tikhonov, A V Goncharsky, VV Stepanov, and Anatoly G Yagola. *Numerical methods for the solution of ill-posed problems*, volume 328. Springer Science & Business Media, 1995.

[51] Sean Twomey. On the numerical solution of fredholm integral equations of the first kind by the inversion of the linear system produced by quadrature. *Journal of the ACM (JACM)*, 10(1):97–101, 1963.

[52] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, 2003, volume 2, pages 1398–1402. Ieee, 2003.

[53] Weiwen Wu, Dianlin Hu, Wenxiang Cong, Hongming Shan, Shaoyu Wang, Chuang Niu, Pingkun Yan, Hengyong Yu, Varut Vardhanabhuti, and Ge Wang. Stabilizing deep tomographic reconstruction: Part a. hybrid framework and experimental results. *Patterns*, 3(5):100475, 2022.

[54] Weiwen Wu, Dianlin Hu, Wenxiang Cong, Hongming Shan, Shaoyu Wang, Chuang Niu, Pingkun Yan, Hengyong Yu, Varut Vardhanabhuti, and Ge Wang. Stabilizing deep tomographic reconstruction: Part b. convergence analysis and adversarial attacks. *Patterns*, 3(5):100475, 2022.

[55] Thomas Yu, Tom Hilbert, Gian Franco Piredda, Arun Joseph, Gabriele Bonanno, Salim Zenkhri, Patrick Omoumi, Meritxell Bach Cuadra, Erick Jorge Canales-Rodríguez, Tobias Kober, et al. Validation and generalizability of self-supervised image reconstruction methods for undersampled mri. *arXiv preprint arXiv:2201.12355*, 2022.

[56] Chi Zhang, Jinghan Jia, Burhaneddin Yaman, Steen Moeller, Sijia Liu, Mingyi Hong, and Mehmet Akçakaya. Instabilities in conventional multi-coil mri reconstruction with small adversarial perturbations. In *2021 55th Asilomar Conference on Signals, Systems, and Computers*, pages 895–899. IEEE, 2021.

[57] Zhengxia Zou, Tianyang Shi, Zhenwei Shi, and Jieping Ye. Adversarial training for solving inverse problems in image processing. *IEEE Transactions on Image Processing*, 30:2513–2525, 2021.