

ACCEPTED MANUSCRIPT • OPEN ACCESS

Invertible residual networks in the context of regularization theory for linear inverse problems

To cite this article before publication: Clemens Arndt *et al* 2023 *Inverse Problems* in press <https://doi.org/10.1088/1361-6420/ad0660>

Manuscript version: Accepted Manuscript

Accepted Manuscript is “the version of the article accepted for publication including all changes made as a result of the peer review process, and which may also include the addition to the article by IOP Publishing of a header, an article ID, a cover sheet and/or an ‘Accepted Manuscript’ watermark, but excluding any other editing, typesetting or other changes made by IOP Publishing and/or its licensors”

This Accepted Manuscript is © 2023 The Author(s). Published by IOP Publishing Ltd.



As the Version of Record of this article is going to be / has been published on a gold open access basis under a CC BY 4.0 licence, this Accepted Manuscript is available for reuse under a CC BY 4.0 licence immediately.

Everyone is permitted to use all or part of the original content in this article, provided that they adhere to all the terms of the licence <https://creativecommons.org/licenses/by/4.0>

Although reasonable endeavours have been taken to obtain all necessary permissions from third parties to include their copyrighted content within this article, their full citation and copyright line may not be present in this Accepted Manuscript version. Before using any content from this article, please refer to the Version of Record on IOPscience once published for full citation and copyright details, as permissions may be required. All third party content is fully copyright protected and is not published on a gold open access basis under a CC BY licence, unless that is specifically stated in the figure caption in the Version of Record.

View the [article online](#) for updates and enhancements.

Invertible residual networks in the context of regularization theory for linear inverse problems

Clemens Arndt* Alexander Denker* Sören Dittmer*[†] Nick Heilenkötter*
Meira Iske* Tobias Kluth* Peter Maass* Judith Nickel*

September 19, 2023

Abstract

Learned inverse problem solvers exhibit remarkable performance in applications like image reconstruction tasks. These data-driven reconstruction methods often follow a two-step procedure. First, one trains the often neural network-based reconstruction scheme via a dataset. Second, one applies the scheme to new measurements to obtain reconstructions. We follow these steps but parameterize the reconstruction scheme with invertible residual networks (iResNets). We demonstrate that the invertibility enables investigating the influence of the training and architecture choices on the resulting reconstruction scheme. For example, assuming local approximation properties of the network, we show that these schemes become convergent regularizations. In addition, the investigations reveal a formal link to the linear regularization theory of linear inverse problems and provide a nonlinear spectral regularization for particular architecture classes. On the numerical side, we investigate the local approximation property of selected trained architectures and present a series of experiments on the MNIST dataset that underpin and extend our theoretical findings.

1 Introduction

In inverse problems, one aims to recover underlying causes from measurements by reversing the forward measurement process. Naturally, they arise in several fields, e.g., medical imaging, non-destructive testing, and PDE-based models. One defines inverse problems via their ill-posed nature, i.e., reversing the measurement process is discontinuous, and obtaining reasonable reconstructions requires a stable reconstruction scheme. Usually, one defines the forward problem by possibly nonlinear forward operator $\mathcal{A} : X \rightarrow Y$ mapping between Banach or Hilbert spaces. Linear spectral regularizations for linear problems in Hilbert space settings were already well studied over two decades ago [17]. Also, more general nonlinear regularizations, e.g., sophisticated variational and iterative approaches, were the subject of extensive investigations for linear and nonlinear problems. We refer to the review [9] for a more general overview of this development. More recently, the linear spectral regularization approach was generalized to diagonal frames [15]. Frames provide larger flexibility by allowing advantageous representations of the data, e.g., via a set of images, instead of solely singular functions provided by the operator. Nonlinear spectral regularizations with nonlinear dependence on the measurement data were only considered in a few works, e.g., in the context of CG methods [17, Cor. 7.4].

During the last decade, these research directions have been accompanied by a highly dynamic development of learning-based methods for inverse problems (see [4] for an earlier review). Many promising methodological directions have arisen, such as end-to-end learned reconstructions [36, 20], learned postprocessing [21, 41], unrolled iteration schemes [18, 1, 19], plug-and-play priors [43, 39, 24], learned penalties in variational approaches [28, 31, 26, 2, 35], (deep) generative models [10, 13], regularization by architecture [42, 14, 3], and several more directions are developed. Besides the pure method development, an increasing number

*Center for Industrial Mathematics, University of Bremen, 28359 Bremen, Germany. Alphabetical author order. Emails: {carndt, adenker, heilenkoetter, iskem, pmaass, junickel}@uni-bremen.de, {sdittmer, tkluth}@math.uni-bremen.de.

[†]Cambridge Image Analysis Group, University of Cambridge, Cambridge CB3 0WA, UK

of works investigate theoretical justifications using the framework of regularization theory (see the recent survey [32] and the more detailed consideration in Section 1.1).

In the present work, we follow a general supervised learning approach for a reconstruction scheme based on the concept of invertible residual networks [8]. We use this architecture to provide a nonlinear regularization scheme for which we develop the general convergence theory by exploiting a local approximation property of the underlying network. Furthermore, instead of solving the inverse problem directly, we use a training strategy that aims to approximate the forward operator. We also introduce an architecture type acting on the singular function directions, and show that it provides a data-dependent nonlinear spectral regularization which is theoretically analyzed for specific shallow architectures and it is linked to the established linear spectral regularization theory. In addition, we underpin our theoretical findings with several numerical experiments.

The manuscript is structured as follows: Section 2 defines the problem setting and the theoretical framework. We then investigate general regularization properties in Section 3. In Section 4, we discuss specific architectures and the outcome of a training approach aiming for approximating the forward operator. We then relate it to the classical filter-based regularization theory. Following the theoretical investigation, Section 5 presents a series of numerical experiments. We conclude with a discussion and outlook in Section 6.

1.1 Related work

Empirically the success of learning-based methods has been established for several applications like various imaging modalities. Theoretically, there is still a gap, e.g., regarding the convergence properties of such reconstruction schemes. Recently an increasing number of works have examined this aspect. We recommend the excellent survey article [32], illustrating regularization properties of some learned methods [32, Fig.3].

One early example providing a convergent learned postprocessing approach is the deep null space network [41]. It utilizes established regularization methods and turns them into learned variants. While it replaces the concept of a minimum norm solution, it inherits convergence guarantees. [5] investigates the convergence of regularization methods trained on a finite set of training samples without access to the entire forward operator. Specifically, the authors consider the projections on the subspaces spanned by the dataset. Assumptions on the dataset ensure desired approximation properties for the convergence analysis of the regularization method. The authors then also consider the data-dependent projected problem in a variational Tikhonov-type framework and derive convergence of the regularization by assuming certain approximation properties of the projection being fulfilled for the desired solution.

Also, Tikhonov-type/variational methods and their classical regularization theory were used to obtain learned regularizations with convergence guarantees. In particular, learned convex regularizers [31, 33] and the network Tikhonov (NETT) approach [26, 35] investigated this.

Minimization of Tikhonov-type functionals is often performed by solving an equilibrium equation, e.g., obtained from first-order optimality conditions. This concept is generalized by the authors in the recent work [34] where a learned operator is introduced in the equilibrium equation replacing, for example, the component delivered by the regularizer. The authors formulate suitable assumptions on the learnable operator to guarantee convergence, and they also illustrate that residual networks fulfill the desired properties if the residual part in the network is contractive.

[16] recently suggested convergence results for plug-and-play reconstructions that are conceptually motivated by the Tikhonov theory. They guarantee convergence by assuming the required conditions on their parameterized family of denoising operators.

The deep image prior [42] proposed exploiting a network's training and architecture as a regularization to achieve better image representation. Based on this concept of unsupervised learning from one single measurement, the authors of [14] derived an analytic deep prior framework formulated as a bilevel optimization problem. This allowed them to verify regularization properties for particular cases by exploiting relations to the classical filter theory of linear regularizations. More recently, [3] investigated the equivalence between Ivanov regularization and analytic deep prior. They investigated the inclusion of early stopping and proved regularization properties [3].

An earlier work [12] proposed a data-driven approach to a learned linear spectral regularization; more recently, [7, 22] considered convergence aspects. In [22], the authors learn a scalar for each singular function direction in a filter-based reconstruction scheme, i.e., a linear regularization scheme. Due to the assumed

properties of noise and data distributions, their training outcome is equivalent to a standard Tikhonov regularization with a data- and noise-dependent linear operator included in the penalty term, which is diagonal with respect to the system of singular functions. The authors further verify convergence results concerning the data- and noise-dependent regularization parameters.

2 Problem setting - theoretical framework

We consider linear inverse problems based on the operator equation

$$\mathcal{A}x = y, \quad (2.1)$$

where $\mathcal{A}: X \rightarrow Y$ is a linear and bounded operator between Hilbert spaces X and Y . For simplification, $\|\mathcal{A}\| = 1$ is assumed, which can be easily obtained by a scaling of the operator. We aim to recover the unknown ground truth x^\dagger as well as possible by only having access to a noisy observation $y^\delta \in Y$ such that $\|y^\delta - \mathcal{A}x^\dagger\| \leq \delta$, where $\delta > 0$ is the noise level.

Instead of solving (2.1) directly, we define $A = \mathcal{A}^* \mathcal{A}$ and $z^\delta = \mathcal{A}^* y^\delta$ to get the operator equation

$$Ax = \mathcal{A}^* \mathcal{A}x = \mathcal{A}^* y = z \quad (2.2)$$

which only acts on X , i.e., we consider the normal equation with respect to \mathcal{A} . We propose a two-step data-based approach for solving (2.2), resp. (2.1), which we refer to as the *iResNet reconstruction approach*:

(I) Supervised training of an invertible neural network $\varphi_\theta: X \rightarrow X$ to approximate A .

(II) Using φ_θ^{-1} (or $\varphi_\theta^{-1} \circ \mathcal{A}^*$, respectively) to solve the inverse problem.

In general, the supervised training covers both cases, either training on noise-free data tuples $(x^{(i)}, Ax^{(i)})$ or on noisy tuples $(x^{(i)}, Ax^{(i)} + \eta^{(i)})$ with noise $\eta^{(i)}$ for a given dataset $\{x^{(i)}\}_{i \in \{1, \dots, N\}} \subset X$. To guarantee the invertibility, we use a residual network of the form

$$\varphi_\theta(x) = x - f_\theta(x) \quad (2.3)$$

and restrict f_θ to be Lipschitz continuous with $\text{Lip}(f_\theta) \leq L < 1$. We refer to this architecture, which is illustrated in Figure 1, as an invertible residual network (iResNet) [8]. Note that [8] considers concatenations of several of these invertible blocks while we focus on a single residual block.

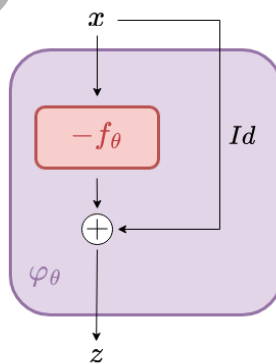


Figure 1: Illustration of the residual network architecture of φ_θ with residual function f_θ and skip connection.

The essential properties of the iResNets (like how it can be inverted) are summarized in the following lemma.

Lemma 2.1 (General properties of iResNets). *Let $\varphi_\theta, f_\theta: X \rightarrow X$, $\varphi_\theta(x) = x - f_\theta(x)$, where $\text{Lip}(f_\theta) \leq L < 1$. Then it holds:*

$$(i) \quad \text{Lip}(\varphi_\theta) \leq L + 1 \quad (2.4)$$

$$(ii) \quad \text{Lip}(\varphi_\theta^{-1}) \leq \frac{1}{1 - L} \quad (2.5)$$

(iii) For given $z \in X$, the unique $x = \varphi_\theta^{-1}(z)$ is given by $x = \lim_{k \rightarrow \infty} x^k$ where

$$x^{k+1} = f_\theta(x^k) + z \quad (2.6)$$

for arbitrary and given $x^0 \in X$.

Proof. The idea of the proof is taken from [8, Section 2]. The first assertion follows directly from

$$\text{Lip}(\varphi_\theta) \leq \text{Lip}(Id) + \text{Lip}(f_\theta). \quad (2.7)$$

Assertion (iii) follows from the fixed point theorem of Banach. Since f_θ is contractive, the iteration converges to a unique fixed point. This fixed point x fulfills

$$z = x + f_\theta(x) = \varphi(x). \quad (2.8)$$

Finally, assertion (ii) follows from

$$\begin{aligned} \|\varphi_\theta(x^1) - \varphi_\theta(x^2)\| &= \|x^1 - x^2 - (f_\theta(x^1) - f_\theta(x^2))\| \\ &\geq \|x^1 - x^2\| - \|f_\theta(x^1) - f_\theta(x^2)\| \geq (1 - L)\|x^1 - x^2\| \end{aligned} \quad (2.9)$$

by replacing $\varphi_\theta(x^i)$ with z^i and x^i with $\varphi_\theta^{-1}(z^i)$ for $i = 1, 2$. \square

Besides, we assume the existence of a singular value decomposition (SVD) $(u_j, v_j, \sigma_j)_{j \in \mathbb{N}}$ of the operator \mathcal{A} , i.e., $u_j \in Y$, $v_j \in X$, $\sigma_j > 0$ such that

$$\mathcal{A}x = \sum_{j=1}^{\infty} \sigma_j \langle x, v_j \rangle u_j \quad (2.10)$$

holds¹. All compact operators guarantee this. With respect to \mathcal{A} , the vectors v_j are eigenvectors, and σ_j^2 are the corresponding eigenvalues. The system $\{v_j\}_{j \in \mathbb{N}}$ builds an orthonormal basis of $\overline{\mathcal{R}(\mathcal{A})} = \overline{\mathcal{R}(\mathcal{A}^*)} = \mathcal{N}(\mathcal{A})^\perp = \mathcal{N}(\mathcal{A}^*)^\perp$. Due to the assumption $\|\mathcal{A}\| = 1$, it holds $\sigma_j \in (0, 1]$.

3 Regularization properties of the iResNet reconstruction

In this section, we discuss regularization properties in terms of well-definedness, stability, and convergence of the general reconstruction approach defined by the family of operators $T_L = \varphi_{\theta, L}^{-1} \circ \mathcal{A}^*$. Note that $L \in [0, 1)$ in the constraint of the Lipschitz constant of the residual function undertakes the role of the regularization parameter by $L \rightarrow 1$, as will be made apparent in this section.

3.1 General architecture and local approximation property

To highlight the essential dependence of the trained network on L , we write $\varphi_{\theta, L}$ and $f_{\theta, L}$ in the following. Please note that the set of possible network parameters and the underlying architecture might also depend on L . We thus consider a network parameter space $\Theta(L)$ depending on L .

Lemma 3.1 (Well-definedness and stability). *For $L \in [0, 1)$ and $\theta \in \Theta(L)$, let $f_{\theta, L} : X \rightarrow X$ be such that $\text{Lip}(f_{\theta, L}) \leq L$ and $\varphi_{\theta, L}(x) = x - f_{\theta, L}(x)$. Then, the reconstruction scheme $T_L = \varphi_{\theta, L}^{-1} \circ \mathcal{A}^* : Y \rightarrow X$ is well-defined and Lipschitz continuous.*

¹In case of $\dim(\mathcal{R}(\mathcal{A})) < \infty$, the number of singular values is finite.

Proof. Well-definedness follows immediately from Lemma 2.1 (iii) and Lipschitz continuity from (ii). \square

While well-definedness and continuity (stability) immediately follow from the construction of the reconstruction method T_L , the convergence property requires a specific approximation property, which we must guarantee during training. We do so by introducing an index function that generalizes the concept of convergence rates (see, e.g., [40]).

Definition 3.1. *An index function is a mapping $\psi: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$, which is continuous, strictly increasing and it holds $\psi(0) = 0$.*

Now, we can formulate a convergence result for the reconstruction $T_L(y^\delta)$ for $\delta \rightarrow 0$ and a suitable a priori parameter choice $L(\delta)$.

Theorem 3.1 (Convergence - local approximation property). *Let $x^\dagger \in X$ be a solution of the problem $\mathcal{A}x = y$ for $y \in R(\mathcal{A})$ and $y^\delta \in Y$ fulfill $\|y^\delta - y\| \leq \delta$. Furthermore, let the network parameters $\theta(L) \in \Theta(L)$ for $L \in [0, 1)$ be obtained in a way that the local approximation property*

$$\|\mathcal{A}^* \mathcal{A}x^\dagger - \varphi_{\theta(L), L}(x^\dagger)\| = \mathcal{O}((1-L)\psi(1-L)) \quad (\text{as } L \rightarrow 1) \quad (3.1)$$

holds for some index function ψ .

If $L: (0, \infty) \rightarrow [0, 1)$ fulfills

$$L(\delta) \rightarrow 1 \quad \wedge \quad \frac{\delta}{1-L(\delta)} \rightarrow 0 \quad \text{for } \delta \rightarrow 0, \quad (3.2)$$

then for $x_{L(\delta)}^\delta := T_{L(\delta)}(y^\delta) = (\varphi_{\theta(L(\delta)), L(\delta)}^{-1} \circ \mathcal{A}^*)(y^\delta)$ it holds

$$\|x_{L(\delta)}^\delta - x^\dagger\| \rightarrow 0 \quad \text{for } \delta \rightarrow 0. \quad (3.3)$$

Proof. For improved readability we write $\varphi_\delta := \varphi_{\theta(L(\delta)), L(\delta)}$ and $f_\delta := f_{\theta(L(\delta)), L(\delta)}$ in the remainder of the proof. Using Lemma 2.1, it holds

$$\begin{aligned} \|x_{L(\delta)}^\delta - x^\dagger\| &\leq \|\varphi_\delta^{-1}(\mathcal{A}^* y^\delta) - \varphi_\delta^{-1}(\mathcal{A}^* y)\| + \|\varphi_\delta^{-1}(\mathcal{A}^* y) - x^\dagger\| \\ &\leq \frac{\|\mathcal{A}^*\|}{1-L(\delta)} \|y^\delta - y\| + \|\varphi_\delta^{-1}(\mathcal{A}^* \mathcal{A}x^\dagger) - x^\dagger\| \\ &\leq \frac{\delta}{1-L(\delta)} + \|\varphi_\delta^{-1}(\mathcal{A}^* \mathcal{A}x^\dagger) - \varphi_\delta^{-1}(\varphi_\delta(x^\dagger))\| \\ &\leq \frac{\delta}{1-L(\delta)} + \frac{1}{1-L(\delta)} \|\mathcal{A}^* \mathcal{A}x^\dagger - \varphi_\delta(x^\dagger)\|. \end{aligned} \quad (3.4)$$

The assertion follows due to the assumptions (3.1) and (3.2) as $\lim_{L \rightarrow 1} \psi(1-L) = 0$. \square

Remark 3.1. *We can obtain a convergence rate of $\|x_{L(\delta)}^\delta - x^\dagger\| = \mathcal{O}(\delta^{\varepsilon/(1+\varepsilon)})$ for $\varepsilon > 0$, if (3.1) is fulfilled with the index function $\psi(\lambda) = \lambda^\varepsilon$, i.e.,*

$$\|\mathcal{A}^* \mathcal{A}x^\dagger - \varphi_{\theta(L), L}(x^\dagger)\| = \mathcal{O}((1-L)^{1+\varepsilon}) \quad (3.5)$$

and by choosing $1-L(\delta) \sim \delta^{1/(1+\varepsilon)}$.

Remark 3.2. *Note that the local approximation property (3.1) is a slightly stronger assumption, which is derived by exploiting the properties of the iResNet to remove the explicit dependency on the inverse within the constraint. A weaker but sufficient condition (which is implied by (3.1)) is*

$$\|\varphi_{\theta(L), L}^{-1}(\mathcal{A}^* \mathcal{A}x^\dagger) - x^\dagger\| \rightarrow 0 \quad \text{as } L \rightarrow 1. \quad (3.6)$$

The local approximation property (3.1) provides a relation between the approximation capabilities of the network architecture and the regularization properties of the iResNet reconstruction scheme, which also contains a certain kind of source condition. Note that the previous statement differs from common convergence results, which are stated not locally but for a broader range of solutions, i.e., for any $y \in \mathcal{R}(\mathcal{A})$. Here, the locality is strongly linked to the approximation capabilities of the underlying network, which can imply certain limitations. Thus, of particular interest for the convergence properties is the *approximation property set*

$$S = \{x \in X \mid \exists \text{ index function } \psi : x \text{ fulfills (3.1)}\}. \quad (3.7)$$

There are no specific assumptions on the design of the training of the network φ_θ made by (3.1). But since both architecture choice and training procedure are crucial for the resulting network parameters $\theta(L)$, they also have a strong impact on the local approximation property and the structure of S . This is discussed in more detail in the context of the specific shallow architectures in Section 4. In addition, we report numerical experiments regarding the local approximation property in Section 5.

A simple example illustrating the conditions under which a linear network satisfies the local approximation property is given in the following remark.

Remark 3.3 (Local approximation property for linear networks). *Assume that the network $\varphi_{\theta,L}$ is linear and the prediction error is bounded by $\|\varphi_{\theta(L(\delta)),L(\delta)}(x^{(i)}) - \mathcal{A}^*y^{\delta,(i)}\| \leq \zeta(\delta)$ with $\zeta(\delta) \in \mathbb{R}$ on a dataset $(x^{(i)}, y^{\delta,(i)})_{i=1,\dots,N}$ with $\|\mathcal{A}x^{(i)} - y^{\delta,(i)}\| \leq \delta$. Then, for x^\dagger given by $x^\dagger = \sum_{i=1}^N t_i x^{(i)}$ with $t_i \in \mathbb{R}$ we have*

$$\begin{aligned} \|\mathcal{A}^* \mathcal{A} x^\dagger - \varphi_{\theta(L(\delta)),L(\delta)}(x^\dagger)\| &= \left\| \mathcal{A}^* \mathcal{A} \sum_{i=1}^N t_i x^{(i)} - \sum_{i=1}^N t_i \varphi_{\theta(L(\delta)),L(\delta)}(x^{(i)}) \right\| \\ &\leq \sum_{i=1}^N |t_i| \|\mathcal{A}^* \mathcal{A} x^{(i)} - \varphi_{\theta(L(\delta)),L(\delta)}(x^{(i)})\| \\ &\leq \sum_{i=1}^N |t_i| \left(\|\mathcal{A}^* \mathcal{A} x^{(i)} - \mathcal{A}^* y^{\delta,(i)}\| + \|\mathcal{A}^* y^{\delta,(i)} - \varphi_{\theta(L(\delta)),L(\delta)}(x^{(i)})\| \right) \\ &\leq \sum_{i=1}^N |t_i| (\delta + \zeta(\delta)). \end{aligned} \quad (3.8)$$

As a result, Theorem 3.1 implies convergence of $x_{L(\delta)}^\delta$ to x^\dagger if

$$\frac{\delta + \zeta(\delta)}{1 - L(\delta)} \rightarrow 0 \quad \text{for } \delta \rightarrow 0 \quad (3.9)$$

and if $L(\delta)$ satisfies the conditions of (3.2). This example shows that in the case that the reconstruction error can be bounded by $\zeta(\delta)$ satisfying (3.9), the convergence directly translates to the linear span of the training data. Consequently, for test data similar to the training data, a simple criterion, that can be numerically validated to ensure convergence to the ground truth data, is obtained.

Similar to a source condition, (3.1) also has an influence on the type of solution x^\dagger one approximates in the limit $\delta \rightarrow 0$ if the operator \mathcal{A} has a non-trivial kernel. For example, if $\mathcal{R}(f_{\theta(L),L}) = \mathcal{N}(\mathcal{A})^\perp$, the local approximation property enforces that $P_{\mathcal{N}(\mathcal{A})} x^\dagger = 0$, i.e., together with the approximation of $\mathcal{A}^* \mathcal{A}$ one would consider the minimum norm solution. More generally, we can formulate the following property of S .

Lemma 3.2. *Let $x_1^\dagger, x_2^\dagger \in S$ both be solutions of $\mathcal{A}x = y$ for one $y \in \mathcal{R}(\mathcal{A})$. Then it holds $x_1^\dagger = x_2^\dagger$.*

Proof. For abbreviation, we write f_L and φ_L instead of $f_{\theta(L),L}$ and $\varphi_{\theta(L),L}$. Using $\mathcal{A}x_1^\dagger = \mathcal{A}x_2^\dagger$, it holds

$$\begin{aligned} &\|x_1^\dagger - x_2^\dagger\| \\ &\leq \|x_1^\dagger - \mathcal{A}^* \mathcal{A} x_1^\dagger - f_L(x_1^\dagger)\| + \|\mathcal{A}^* \mathcal{A} x_1^\dagger + f_L(x_1^\dagger) - \mathcal{A}^* \mathcal{A} x_2^\dagger - f_L(x_2^\dagger)\| + \|\mathcal{A}^* \mathcal{A} x_2^\dagger + f_L(x_2^\dagger) - x_2^\dagger\| \\ &= \|\varphi_L(x_1^\dagger) - \mathcal{A}^* \mathcal{A} x_1^\dagger\| + \|f_L(x_1^\dagger) - f_L(x_2^\dagger)\| + \|\mathcal{A}^* \mathcal{A} x_2^\dagger - \varphi_L(x_2^\dagger)\| \\ &\leq \|\mathcal{A}^* \mathcal{A} x_1^\dagger - \varphi_L(x_1^\dagger)\| + L \|x_1^\dagger - x_2^\dagger\| + \|\mathcal{A}^* \mathcal{A} x_2^\dagger - \varphi_L(x_2^\dagger)\|. \end{aligned} \quad (3.10)$$

Subtracting $L\|x_1^\dagger - x_2^\dagger\|$, it follows

$$(1 - L)\|x_1^\dagger - x_2^\dagger\| \leq \|\mathcal{A}^* \mathcal{A} x_1^\dagger - \varphi_L(x_1^\dagger)\| + \|\mathcal{A}^* \mathcal{A} x_2^\dagger - \varphi_L(x_2^\dagger)\|. \quad (3.11)$$

Since x_1^\dagger, x_2^\dagger both fulfill the local approximation property (3.1), there exists an index function ψ such that $(1 - L)\|x_1^\dagger - x_2^\dagger\| = \mathcal{O}((1 - L)\psi(1 - L))$ must hold. This implies $\|x_1^\dagger - x_2^\dagger\| = \mathcal{O}(\psi(1 - L))$, which is only possible for $x_1^\dagger = x_2^\dagger$. \square

3.2 Diagonal architecture

In order to investigate relations to established and well-studied spectral regularization methods, we continue with the introduction of a particular network design, which is exploited in the remainder of the manuscript. The idea of spectral regularization is to decompose the data using the SVD $(u_j, v_j, \sigma_j)_{j \in \mathbb{N}}$ of \mathcal{A} and apply filter functions to the singular values. Analogously, we consider a network architecture consisting of subnetworks $f_{\theta,j} : \mathbb{R} \rightarrow \mathbb{R}$, $j \in \mathbb{N}$, i.e.,

$$f_\theta(x) = \sum_{j \in \mathbb{N}} f_{\theta,j}(\langle x, v_j \rangle) v_j, \quad (3.12)$$

where all the components $\langle x, v_j \rangle$ of the data are processed separately. We refer to this architecture as *diagonal* architecture, which is also illustrated in Figure 2. The naming is motivated by the similarity of the structure of f_θ to a matrix multiplication with a diagonal matrix w.r.t the basis $\{v_j\}_{j \in \mathbb{N}}$. This architecture choice is, of course, less expressive than general architectures, but it has the useful benefit that this way, the iResNet can be analyzed like a filter-based regularization scheme. Besides, it enables us to simplify the infinite-dimensional inverse problem $\mathcal{A}x = y$ to 1D subproblems of the form $\sigma_j \langle x, v_j \rangle = \langle y, u_j \rangle$, which will be exploited in section 4.

In case of a diagonal architecture, the Lipschitz constraint $\text{Lip}(f_\theta) \leq L$ is fulfilled if $\text{Lip}(f_{\theta,j}) \leq L$ holds for all $j \in \mathbb{N}$, i.e., for any $x, y \in X$ it holds

$$\|f_\theta(x) - f_\theta(y)\|^2 = \sum_{j \in \mathbb{N}} |f_{\theta,j}(\langle x, v_j \rangle) - f_{\theta,j}(\langle y, v_j \rangle)|^2 \leq L^2 \sum_{j \in \mathbb{N}} |\langle x - y, v_j \rangle|^2 \leq L^2 \|x - y\|^2. \quad (3.13)$$

The local approximation property (3.1), in this case for $x^\dagger \in \mathcal{N}(\mathcal{A})^\perp$, implies

$$|f_{\theta(L),L,j}(x_j^\dagger) - (1 - \sigma_j^2)x_j^\dagger| = \mathcal{O}((1 - L)\psi(1 - L)) \quad (3.14)$$

for any $j \in \mathbb{N}$, where $x_j^\dagger = \langle x^\dagger, v_j \rangle$. Note that for infinite-dimensional operators, the constants in $\mathcal{O}((1 - L)\psi(1 - L))$ must be an ℓ^2 sequence with respect to j to obtain the implication in the opposite direction and thus equivalence.

The particular aim of using the diagonal architecture is now to find a filter function that defines a spectral regularization scheme that is equivalent to $T_L = \varphi_\theta^{-1} \circ \mathcal{A}^*$. Since filter-based regularization methods are usually linear and we want to also allow for nonlinear architectures, our filter function r_L must be data-dependent. Therefore, we define $r_L : \mathbb{R}_+ \times \mathbb{R} \rightarrow \mathbb{R}$ such that for $z \in \mathcal{R}(\mathcal{A}^*)$

$$\varphi_\theta^{-1}(z) = \sum_{j \in \mathbb{N}} r_L(\sigma_j^2, \langle z, v_j \rangle) \langle z, v_j \rangle v_j, \quad (3.15)$$

or for $y \in Y$

$$T_L(y) = \varphi_\theta^{-1}(\mathcal{A}^* y) = \sum_{j \in \mathbb{N}} r_L(\sigma_j^2, \sigma_j \langle y, u_j \rangle) \sigma_j \langle y, u_j \rangle v_j \quad (3.16)$$

holds. The first argument of r_L is the singular value (as usual), and the data-dependency comes via the second argument. This can be seen as a nonlinear extension to the established filter theory of linear regularization methods [38, 27], where r_L would depend on σ_j^2 only. The subnetworks thus play an important role in defining the filter functions via

$$(Id - f_{\theta,L,j})^{-1}(s) = r_L(\sigma_j^2, s)s. \quad (3.17)$$

For this to be well-defined, we need to assume that all singular values $\sigma_j > 0$ have a multiplicity of one. In case of $\sigma_j = \sigma_k$ for $j \neq k$ (multiplicity greater than one) one must ensure that $f_{\theta,L,j}$ and $f_{\theta,L,k}$ also coincide. In practice, this could be realized by sharing the weights of these networks.

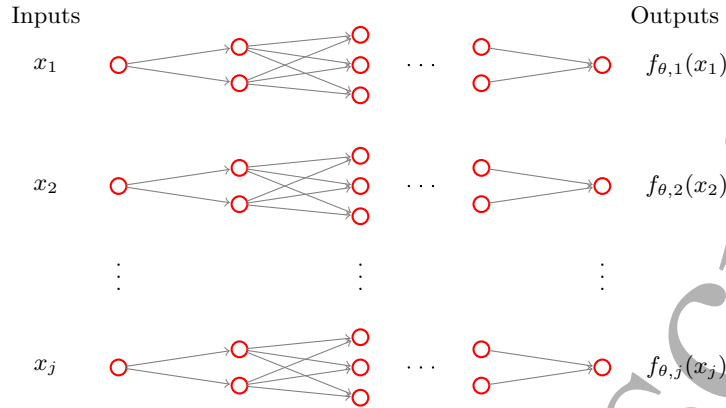


Figure 2: *Diagonal* structure of the residual function including subnetworks $f_{\theta,j} : \mathbb{R} \rightarrow \mathbb{R}$ acting on $x_j = \langle x, v_j \rangle$.

Remark 3.4. Some authors define filter functions in terms of the original problem with respect to the generalized inverse \mathcal{A}^\dagger such that

$$\varphi_\theta^{-1}(\mathcal{A}^*y) = \sum_{j \in \mathbb{N}} F_L(\sigma_j, \langle y, u_j \rangle) \frac{1}{\sigma_j} \langle y, u_j \rangle v_j, \quad \text{with } F_L(\sigma, s) = \sigma^2 r_L(\sigma^2, \sigma s). \quad (3.18)$$

Note that the choice of the representation, either in terms of r_L or in terms of F_L , depends on personal preference (compare, for example, [38] and [27]). In the following, we represent the filter function in terms of r_L .

One simple nonlinear dependence on s will be taken into account explicitly in the following, as it becomes relevant in later investigations. In analogy to the bias in neural network architectures, we consider a linear filter framework with bias, i.e.,

$$T_L(y) = \hat{b}_L + \sum_{j \in \mathbb{N}} \hat{r}_L(\sigma_j^2) \sigma_j \langle y, u_j \rangle v_j \quad (3.19)$$

where $\hat{b}_L \in X$ is a bias term and \hat{r}_L is a classical filter function. With additional simple assumptions on \hat{b}_L , this becomes a regularization method. For the sake of completeness, we provide the following result.

Lemma 3.3 (Filter-based spectral regularization with bias). *Let $\hat{r}_L : [0, \|\mathcal{A}\|^2] \rightarrow \mathbb{R}$ be a piecewise continuous function and let $b_L \in X$ for $L \in [0, 1)$. Furthermore, let*

$$(i) \lim_{L \rightarrow 1} \hat{r}_L(\sigma_j^2) = \frac{1}{\sigma_j^2} \text{ for any } \sigma_j, j \in \mathbb{N},$$

$$(ii) \exists 0 < C < \infty : \sigma_j^2 |\hat{r}_L(\sigma_j^2)| \leq C \text{ for any } \sigma_j, j \in \mathbb{N} \text{ and } L \in [0, 1),$$

$$(iii) \hat{b}_L \in X \text{ for } L \in [0, 1) \text{ and } \lim_{L \rightarrow 1} \|\hat{b}_L\| = 0$$

hold. Let $x^\dagger \in \mathcal{N}(\mathcal{A})^\perp$ be a solution of the problem $\mathcal{A}x = y$ for $y \in R(\mathcal{A})$, the operator $T_L : Y \rightarrow X$ be given by (3.19) and $y^\delta \in Y$ with $\|y - y^\delta\| \leq \delta$. In addition, let $L : (0, \infty) \rightarrow [0, 1)$ be such that

$$L(\delta) \rightarrow 1 \quad \wedge \quad \delta \sqrt{\sup \{ |\hat{r}_{L(\delta)}(\sigma_i)| \mid i \in \mathbb{N} \}} \rightarrow 0 \quad (3.20)$$

as $\delta \rightarrow 0$. Then, T_L is a regularization method and for $x_{L(\delta)}^\delta := T_{L(\delta)}(y^\delta)$ it holds $\lim_{\delta \rightarrow 0} \|x_{L(\delta)}^\delta - x^\dagger\| = 0$.

Proof. By defining $\tilde{x}_{L(\delta)}^\delta := T_{L(\delta)}(y^\delta) - \hat{b}_{L(\delta)}$ (removing the bias), we get an ordinary linear regularization method (see, e.g., [38, Corr. 3.3.4]). It holds

$$\|x_{L(\delta)}^\delta - x^\dagger\| \leq \|x_{L(\delta)}^\delta - \tilde{x}_{L(\delta)}^\delta\| + \|\tilde{x}_{L(\delta)}^\delta - x^\dagger\| = \|\hat{b}_{L(\delta)}\| + \|\tilde{x}_{L(\delta)}^\delta - x^\dagger\|. \quad (3.21)$$

Since $\|\hat{b}_{L(\delta)}\| \rightarrow 0$ by assumption and $\tilde{x}_{L(\delta)}^\delta \rightarrow x^\dagger$ by standard theory, T_L is a regularization method. \square

4 Approximation training of specialized architectures

Besides the architecture choice and the available data, the loss function for network training is an important ingredient. Having regularization properties in mind, a natural choice for training the iResNet is to approximate the forward operator $A: X \rightarrow X$, $A = \mathcal{A}^* \mathcal{A}$, on a given training dataset $\{x^{(i)}\}_{i \in \{1, \dots, N\}} \subset X$. This is also strongly motivated by the structure of the local approximation property (3.1) to obtain convergence guarantees. The training of $\varphi_\theta = Id - f_\theta$ then consists of solving

$$\min_{\theta \in \Theta} l(\varphi_\theta, A) = \min_{\theta \in \Theta} \frac{1}{N} \sum_i \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \quad \text{s.t. } \text{Lip}(f_\theta) \leq L \quad (4.1)$$

with $L < 1$ as a hyperparameter, which we refer to as *approximation training*. Here, we restrict ourselves to the noise-free case as the outcome of the approximation training is independent for $N \rightarrow \infty$ (assuming stochastic independence between data and noise).

In the following, we analyze to which extent φ_θ^{-1} acts as a regularized inverse of A for different simple diagonal iResNet architectures trained according to (4.1). We compute the particular network parameters θ and derive the corresponding data-dependent filter function $r_L: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ (see Section 3.2). Additionally, we check whether the local approximation property (3.1) is fulfilled. For this purpose, the assumptions on the architecture as well as on the dataset used for the training must be specified.

4.1 One-parameter-network – Tikhonov

We begin with a very simple linear network, which only has one single scalar learnable parameter. We show that this architecture is equivalent to Tikhonov regularization if trained appropriately.

Lemma 4.1. *Let $(v_j, \sigma_j^2)_j$ be the eigenvectors and eigenvalues of A and let $\varphi_\theta = Id - f_\theta$ be an iResNet which solves (4.1) with*

(i) $f_\theta = k(Id - A)$, where $\theta = k$ and $\Theta = \mathbb{R}$ (*architecture assumption*),

(ii) the training dataset $\{x^{(i)}\}_{i \in \{1, \dots, N\}}$ contains at least one $x^{(i)}$ s.t. $Ax^{(i)} \neq x^{(i)}$ (*dataset assumption*).

Then, the solution of (4.1) is $k = L$ and (3.15) holds with

$$r_L(\sigma^2, s) = \frac{1}{L} \cdot \frac{1}{\alpha + \sigma^2}, \quad \alpha = \frac{1 - L}{L}, \quad (4.2)$$

for any $s \in \mathbb{R}$.

The proof of the lemma can be found in Appendix A.1. The derived filter function corresponds to the Tikhonov method (with reference element)

$$\min_{x \in X} \|Ax - y^\delta\|^2 + \alpha \|x - \mathcal{A}^* y^\delta\|^2 \quad (4.3)$$

as one can see by considering the first-order optimality condition, which implies

$$x = \sum_j \frac{1 + \alpha}{\alpha + \sigma_j^2} \sigma_j \langle y^\delta, u_j \rangle v_j \quad (4.4)$$

and using the fact that $\frac{1}{L} = 1 + \alpha$. It is illustrated in Figure 3. We plot $\sigma^2 r_L(\sigma^2, s)$ since the multiplication of the filter function with σ^2 corresponds to the concatenation $\varphi_\theta^{-1} \circ A$ which emphasizes the regularizing effect.

Remark 4.1. *This particular choice of the residual function f_θ is one example where the local approximation property is not fulfilled for any $x^\dagger \in X$ except for eigenvectors corresponding to $\sigma_1^2 = 1$, i.e., $S = \text{span}\{v_1\}$. But the linear nature and the specific spectral representation of the residual network allow for the verification*

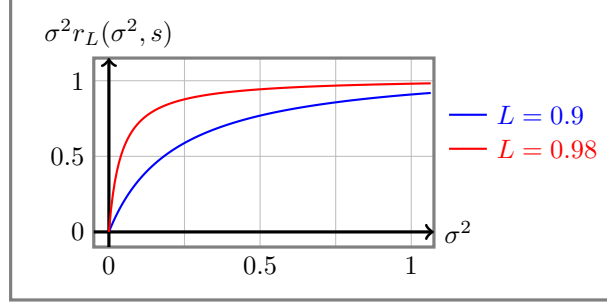


Figure 3: The plot depicts graphs of $\sigma^2 r_L(\sigma^2, s)$ as given in (4.2) to visualize the effect of the regularization, which is also known as standard Tikhonov regularization.

of the weaker constraint in (3.6) being highlighted in Remark 3.2 and being sufficient for convergence. For the residual function $f_{\theta(L),L} = L(Id - A)$ and $x^\dagger \in \mathcal{N}(A)^\perp$ we obtain

$$\begin{aligned} \|\varphi_{\theta(L),L}^{-1}(Ax^\dagger) - x^\dagger\|^2 &= \|((Id - L(Id - A))^{-1}A - Id)x^\dagger\|^2 \\ &= \sum_{j \in \mathbb{N}} \left(\underbrace{\frac{\sigma_j^2}{1 - L(1 - \sigma_j^2)}}_{=\sigma_j^2 r_L(\sigma_j^2)} - 1 \right)^2 |\langle x^\dagger, v_j \rangle|^2, \end{aligned} \quad (4.5)$$

which converges to zero for $L \rightarrow 1$.

Alternatively, convergence can be verified by standard arguments following, for example, the line of reasoning in the proof of [27, Thm. 3.3.3], where properties of the filter function $F_L(\sigma_j) = \sigma_j^2 r_L(\sigma_j^2)$ are exploited. Since F_L defines the filter function for Tikhonov regularization and is therefore known to fulfill the desired properties, we have convergence for arbitrary $x^\dagger \in \mathcal{N}(A)^\perp$.

4.2 Affine linear network – squared soft TSVD

We continue with a slightly more general affine linear architecture $f_\theta(x) = Wx + b$. It can be observed that the linear operator $W : X \rightarrow X$ only depends on A and L while the bias vector $b \in X$ is data dependent.

Lemma 4.2. Let $(v_j, \sigma_j^2)_j$ be the eigenvectors and eigenvalues of A and $\varphi_\theta = Id - f_\theta$ be an *iResNet* which solves (4.1) where

(i) $f_\theta(x) = Wx + b$, with $\theta = (W, b)$ and

$$\Theta = \left\{ (W, b) \in L(X) \times X \mid \exists (w_j)_{j \in \mathbb{N}}, (b_j)_{j \in \mathbb{N}} : W = \sum_{j \in \mathbb{N}} w_j \langle \cdot, v_j \rangle v_j, b = \sum_{j \in \mathbb{N}} b_j v_j \right\}, \quad (4.6)$$

(ii) the training dataset $\{x^{(i)}\}_{i \in \{1, \dots, N\}}$ and the mean values $\mu_j = \frac{1}{N} \sum_{i=1}^N \langle x^{(i)}, v_j \rangle$ fulfill

$$\forall j \in \mathbb{N} : \exists i \in \{1, \dots, N\} : \langle x^{(i)}, v_j \rangle \neq \mu_j. \quad (4.7)$$

Then, the solution of the training problem (4.1) is $w_j = \min\{1 - \sigma_j^2, L\}$, $b_j = \max\{0, 1 - L - \sigma_j^2\} \mu_j$ and $\varphi_\theta^{-1} \circ \mathcal{A}^*$ is equivalent to T_L in (3.19) with

$$\hat{r}_L(\sigma^2) = \frac{1}{\max\{\sigma^2, 1 - L\}}, \quad \hat{b}_L = \frac{1}{1 - L} \sum_{j \in \mathbb{N}} b_j v_j = \sum_{\sigma_j^2 < 1 - L} \frac{1 - L - \sigma_j^2}{1 - L} \mu_j v_j. \quad (4.8)$$

The proof of the lemma can be found in Appendix A.2. The filter function is illustrated in Figure 4.

Remark 4.2. Note that a similar filter function has been found in the context of an analytic deep prior approach for regularization [14]. The authors derived the filter function $F_L(\sigma) = \sigma^2 \hat{r}_L(\sigma^2)$, which is linear for small σ , and named it soft TSVD (soft truncated SVD). In contrast, the filter function in (4.8) is a polynomial of degree two for small σ , which we refer to as squared soft TSVD.

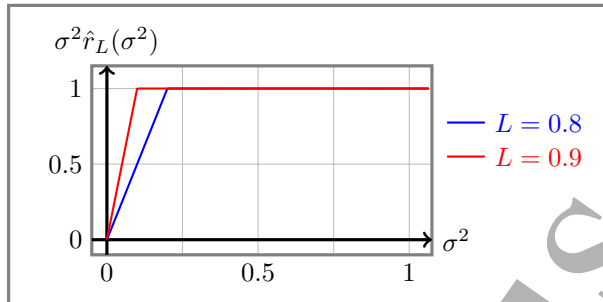


Figure 4: The plot depicts graphs of $\sigma^2 \hat{r}_L(\sigma^2)$ as given in (4.8) to visualize the effect of the regularization, which is similar to the known soft TSVD regularization.

Due to the affine linear nature of the found reconstruction scheme, regularization properties can immediately be derived by verifying certain properties of the filter function \hat{r}_L and the bias \hat{b}_L using Lemma 3.3. This is summarized in the following corollary.

Corollary 4.1. *The filter based method (3.19) with \hat{r}_L and \hat{b}_L given by (4.8) fulfills the properties (i)-(iii) in Lemma 3.3, i.e., in the setting of Lemma 3.3 it is a convergent regularization.*

The previous result provides a link to the classical theory for linear regularization operators. Besides the well-known classical theory, we further want to investigate the local approximation property (3.1) to verify whether the convergence property can also be obtained using Theorem 3.1. As a first step, we derive a technical source condition for x^\dagger , which implies the local approximation property.

Lemma 4.3. *Let the network φ_θ and the setting be that of Lemma 4.2. Assume $x^\dagger \in \mathcal{N}(A)^\perp$ and let ψ be an index function such that*

$$\exists \bar{\beta} \in (0, 1] : \forall \beta \in (0, \bar{\beta}) : \sum_{j: \sigma_j^2 \leq \beta} \langle x^\dagger, v_j \rangle^2 = \mathcal{O}(\psi(\beta)^2) \quad (4.9)$$

holds. Then there exists another index function $\tilde{\psi}$ such that the local approximation property (3.1) is fulfilled for x^\dagger .

Proof. With the setting of Lemma 4.2 and the trained parameters (W, b) of the network f_θ , it holds

$$\|\varphi_{\theta(L)}(x^\dagger) - Ax^\dagger\| = \|(Id - W)x^\dagger - b - Ax^\dagger\| \leq \|(Id - W - A)x^\dagger\| + \|b\|. \quad (4.10)$$

Now, we estimate the convergence order of both terms w.r.t $(1 - L) \rightarrow 0$.

For the first part, we exploit the simple computations

$$\begin{aligned}
\|(Id - W - A)x^\dagger\|^2 &= \left\| \sum_j (1 - w_j - \sigma_j^2) \langle x^\dagger, v_j \rangle v_j \right\|^2 \\
&= \sum_j (1 - \sigma_j^2 - \min\{1 - \sigma_j^2, L\})^2 \langle x^\dagger, v_j \rangle^2 \\
&= \sum_j \max\{1 - L - \sigma_j^2, 0\}^2 \langle x^\dagger, v_j \rangle^2 \\
&= \sum_{\sigma_j^2 \leq 1-L} (1 - L - \sigma_j^2)^2 \langle x^\dagger, v_j \rangle^2 \\
&\leq \sum_{\sigma_j^2 \leq 1-L} (1 - L)^2 \langle x^\dagger, v_j \rangle^2 \\
&= (1 - L)^2 \sum_{\sigma_j^2 \leq 1-L} \langle x^\dagger, v_j \rangle^2.
\end{aligned} \tag{4.11}$$

Using assumption (4.9), we obtain

$$(1 - L)^2 \sum_{\sigma_j^2 \leq 1-L} \langle x^\dagger, v_j \rangle^2 = \mathcal{O}\left((1 - L)^2 \psi(1 - L)^2\right). \tag{4.12}$$

Regarding the second part, it holds

$$\begin{aligned}
\|b\|^2 &= \sum_j b_j^2 = \sum_j \max\{0, 1 - L - \sigma_j^2\}^2 \mu_j^2 \\
&= \sum_{\sigma_j^2 \leq 1-L} (1 - L - \sigma_j^2)^2 \mu_j^2 \leq (1 - L)^2 \sum_{\sigma_j^2 \leq 1-L} \mu_j^2.
\end{aligned} \tag{4.13}$$

Since the values μ_j form by definition (see Lemma 4.2) an ℓ^2 sequence, there exists an index function $\tilde{\psi} \geq \psi$ such that

$$\sum_{\sigma_j^2 \leq 1-L} \mu_j^2 = \mathcal{O}\left(\tilde{\psi}(1 - L)^2\right). \tag{4.14}$$

Overall, we obtain

$$\|(Id - W - A)x^\dagger\| + \|b\| = \mathcal{O}\left((1 - L)^2 \tilde{\psi}(1 - L)^2\right), \tag{4.15}$$

thus, the local approximation property (3.1) is fulfilled. \square

While (4.9) is quite a technical source condition, standard source conditions of the form

$$\exists w \in X, \mu > 0: \quad x^\dagger = A^\mu w \tag{4.16}$$

imply it. In this case, the index function ψ is of the form $\psi(\beta) = \beta^\mu$. The proof of the following corollary reveals the exact relation between standard source conditions and (4.9).

Corollary 4.2. *We assume that the setting of Lemma 4.2 holds. Let \mathcal{A} be compact. Then, for any $x^\dagger \in \mathcal{N}(\mathcal{A})^\perp$ the local approximation property (3.1) is fulfilled (i.e., $S = \mathcal{N}(\mathcal{A})^\perp$).*

Proof. Let $x^\dagger \in \mathcal{N}(\mathcal{A})^\perp$ be arbitrary. By construction, $A = \mathcal{A}^* \mathcal{A}$ is self-adjoint and non-negative. As A is also compact, we can deduce from [29, Thm. 1]: For any $\varepsilon > 0$ there exists an index function $\psi : [0, 1] \rightarrow [0, \infty)$ such that $x^\dagger \in \{x \in X \mid x = \psi(A)w, \|w\| \leq (1 + \varepsilon)\|x^\dagger\|\}$.

This implies that there exists a $w \in X$ such that $x^\dagger = \psi(A)w$. We thus obtain for $\beta > 0$

$$\begin{aligned} \sum_{\sigma_j^2 \leq \beta} \langle x^\dagger, v_j \rangle^2 &= \sum_{\sigma_j^2 \leq \beta} \langle \psi(A)w, v_j \rangle^2 = \sum_{\sigma_j^2 \leq \beta} \psi(\sigma_j^2)^2 \langle w, v_j \rangle^2 \\ &\leq \sum_{\sigma_j^2 \leq \beta} \psi(\beta)^2 \langle w, v_j \rangle^2 \leq \|w\|^2 \psi(\beta)^2 \leq (1 + \varepsilon)^2 \|x^\dagger\|^2 \psi(\beta)^2. \end{aligned} \quad (4.17)$$

By Lemma 4.3, we get the desired local approximation property (3.1). \square

The verification of regularization properties by Theorem 3.1 in terms of the local approximation property is thus in line with the existing theory exploited in Lemma 3.3, and it further illustrates the character of the local approximation property combining the approximation capabilities of the residual network and a source condition on the desired solution.

4.3 Linear network with ReLU activation

In the following, we include selected nonlinearities as activation functions in a shallow network, allowing for an analytical investigation of the resulting reconstruction scheme. Here, we start with a ReLU activation, which requires a certain assumption on the training dataset depending on the chosen nonlinearity. For simplicity, we do not include a bias in the architecture, in contrast to the architecture from the last section.

Lemma 4.4. *Let $(v_j, \sigma_j^2)_j$ be the eigenvectors and eigenvalues of A and $\varphi_\theta = Id - f_\theta$ be an iResNet which solves (4.1) with*

(i) $f_\theta(x) = \phi(Wx)$, where $\theta = W$ and $\Theta = \{W \in L(X) \mid \exists (w_j)_{j \in \mathbb{N}} : W = \sum_{j \in \mathbb{N}} w_j \langle \cdot, v_j \rangle v_j\}$ and ϕ being the ReLU function w.r.t. the eigenvectors, i.e., $\phi(x) = \sum_{j \in \mathbb{N}} \max(0, \langle v_j, x \rangle) v_j$,

(ii) for every eigenvector v_j of A , the training dataset $\{x^{(i)}\}_{i \in \{1, \dots, N\}}$ contains at least one $x^{(i)}$ s.t. $\langle x^{(i)}, v_j \rangle > 0$.

Then, the solution of (4.1) is $W = \sum_{j \in \mathbb{N}} w_j \langle \cdot, v_j \rangle v_j$, $w_j = \min\{1 - \sigma_j^2, L\}$ and (3.15) holds with

$$r_L(\sigma^2, s) = \begin{cases} \frac{1}{\max\{\sigma^2, 1-L\}} & \text{if } s \geq 0, \\ 1 & \text{if } s < 0. \end{cases} \quad (4.18)$$

The proof of the lemma can be found in Appendix A.3. The obtained filter function is now characterized by a varying behavior depending on the actual measurement y . This is expressed via the variable s which represents the coefficients $\langle z, v_i \rangle = \sigma_i \langle y, u_i \rangle$ (see (3.15), (3.16)). Whenever the coefficient is positive, the reconstruction scheme behaves like the squared soft TSVD without bias discussed in Section 4.2, i.e., they share the same filter function for those $x^\dagger \in \mathcal{N}(A)$. In all other cases, the reconstruction method does not change the coefficients of the data, i.e., it behaves like the identity. Due to the relationship to the squared soft TSVD, we can immediately specify those x^\dagger fulfilling the local approximation property, i.e., for the ReLU network, we have

$$S = \{x \in \mathcal{N}(A)^\perp \mid \forall j \in \mathbb{N} : \langle v_j, x \rangle \geq 0\}. \quad (4.19)$$

Thus, the nonlinearity in the network architecture introduces restricted approximation capabilities as well as convergence guarantees on a limited subset of X only.

4.4 Linear network with soft thresholding activation

At last, we want to analyze a network with soft thresholding activation, e.g., known from LISTA [18]. This function is known to promote sparsity since it shrinks all coefficients and sets those under a certain threshold to zero. That is why only training data with sufficiently large coefficients matters for the result of the training, and the condition $|\langle x, v_j \rangle| > \frac{\alpha_j}{L}$ is crucial in the following lemma.

Lemma 4.5. *Let $(v_j, \sigma_j^2)_j$ be the eigenvectors and eigenvalues of A , $\alpha = (\alpha_j)_j$, $\alpha_j \geq 0$ and $\varphi_\theta = Id - f_\theta$ be an iResNet which solves (4.1) with*

(i) $f_\theta(x) = \phi_\alpha(Wx)$, where $\theta = W$, $\Theta = \{W \in L(X) \mid \exists (w_j)_{j \in \mathbb{N}} : W = \sum_{j \in \mathbb{N}} w_j \langle \cdot, v_j \rangle v_j\}$ and ϕ_α is the soft thresholding function w.r.t. the eigenvectors, i.e., $\phi_\alpha(x) = \sum_{j \in \mathbb{N}} \text{sign}(\langle x, v_j \rangle) \max(0, |\langle x, v_j \rangle| - \alpha_j) v_j$,

(ii) the training data set is $\{x^{(i)}\}_{i \in \{1, \dots, N\}}$, where for any $j \in \mathbb{N} : \exists i : |\langle x^{(i)}, v_j \rangle| > \frac{\alpha_j}{L}$.

Then, a solution of (4.1) is $W = \sum_{j \in \mathbb{N}} w_j \langle \cdot, v_j \rangle v_j$, $w_j = \min\{\frac{\alpha_j}{p_{L,j}} + 1 - \sigma_j^2, L\}$, $p_{L,j} = \frac{\sum_{i \in I_j(L)} |\langle x^{(i)}, v_j \rangle|^2}{\sum_{i \in I_j(L)} |\langle x^{(i)}, v_j \rangle|}$, $I_j(L) = \{i \in \{1, \dots, N\} \mid |\langle x^{(i)}, v_j \rangle| > \frac{\alpha_j}{L}\}$ and (3.15) holds with

$$r_L(\sigma_j^2, s) = \begin{cases} \frac{1}{\max\{\sigma_j^2 - \frac{\alpha_j}{p_{L,j}}, 1-L\}} \frac{|s| - \alpha_j}{|s|} & \text{if } |s| > \frac{\alpha_j}{w_j}, \\ 1 & \text{if } |s| \leq \frac{\alpha_j}{w_j}. \end{cases} \quad (4.20)$$

For singular values $\sigma_j^2 = 1$, w_j is not uniquely determined.

The proof of the lemma can be found in Appendix A.4. It follows the same line of reasoning as in the previous sections but is more technical due to the effects of the nonlinearity.

So far, the filter function r_L in Lemma 4.5 is only defined on the discrete values σ_j^2 (and not continuous for $\sigma^2 \in [0, 1]$) since it depends on the coefficients $p_{L,j}$, α_j and w_j . However, if we assume continuous extensions $p_L(\sigma^2)$ with $p_L(\sigma_j^2) = p_{L,j}$, $w_L(\sigma^2)$ with $w_L(\sigma_j^2) = w_j$, and $\alpha(\sigma^2)$ with $\alpha(\sigma_j^2) = \alpha_j$, the function $r_L = r_L(\sigma^2, s)$ also becomes continuous. The continuity at the point $|s| = \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}$ is assured by

$$\frac{1}{\max\{\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}, 1-L\}} \frac{\frac{\alpha(\sigma^2)}{w_L(\sigma^2)} - \alpha(\sigma^2)}{\frac{\alpha(\sigma^2)}{w_L(\sigma^2)}} = \frac{1 - w_L(\sigma^2)}{\max\{\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}, 1-L\}} = 1 = r_L(\sigma^2, s). \quad (4.21)$$

To be able to interpret the filter function, suitable values for s representing the coefficients $\langle z, v_i \rangle = \sigma_i \langle y, u_i \rangle$ (see (3.15), (3.16)) need to be considered. One reasonable option is to choose s according to the data on which φ_θ has been trained. We thus consider an eigenvector v_j scaled with the coefficient $p_{L,j}$. Since φ_θ minimizes (4.1), we expect $\varphi_\theta(p_{L,j} v_j) \approx p_{L,j} A v_j = p_{L,j} \sigma_j^2 v_j$ and $\varphi_\theta^{-1}(p_{L,j} \sigma_j^2 v_j) \approx p_{L,j} v_j$, respectively. Accordingly, we choose $|s|$ proportional to $p_L(\sigma^2) \sigma^2$, i.e., $|s| = \gamma p_L(\sigma^2) \sigma^2$ for different values $\gamma > 0$. Hence, the case $\gamma = 1$ corresponds to a test vector z with coefficients $|\langle z, v_i \rangle| = |s| = p_L(\sigma_j^2) \sigma_j^2$, which perfectly fits to the training data. Analogously, the cases $\gamma < 1$ (and $\gamma > 1$, respectively) correspond to test data whose coefficients are smaller (or bigger, respectively) than the average coefficients of the training data.

For $\gamma = 1$, the filter function r_L can be written in a form, which allows for an easier interpretation. It holds

$$r_L(\sigma^2, \pm p_L(\sigma^2) \sigma^2) = \begin{cases} 1 & \text{if } \sigma^2 \leq \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)}, \\ \frac{1}{1-L} \left(1 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2) \sigma^2}\right) & \text{if } \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)} < \sigma^2 \leq \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L, \\ \frac{1}{\sigma^2} & \text{if } \sigma^2 > \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L. \end{cases} \quad (4.22)$$

The derivation can be found in Appendix A.5. Note that the filter function depends especially on the quotient of $\alpha(\sigma^2)$ (soft thresholding parameter) and $p_L(\sigma^2)$ (property of training data). To visualize the influence, we depicted the graph in Figure 5 for two different (constant) values of $\frac{\alpha(\sigma^2)}{p_L(\sigma^2)}$. As can be seen, the graph of $\sigma^2 \mapsto \sigma^2 r(\sigma^2, p_L(\sigma^2) \sigma^2)$ has two kinks. The first one depends mainly on the choice of $\frac{\alpha(\sigma^2)}{p_L(\sigma^2)}$, the second one mainly on the choice of L .

For $\gamma \neq 1$, the filter functions cannot be written in a similarly compact and easy form as in (4.22). Instead, we illustrate them in Figure 6. In contrast to the case of $\gamma = 1$, the graph of $\sigma^2 \mapsto \sigma^2 r(\sigma^2, p_L(\sigma^2) \sigma^2)$ is not equal to one for the larger values of σ^2 .

Finally, we want to analyze to which extent φ_θ fulfills the local approximation property (3.1). The illustration of the filter function in Figure 5 implies that convergence can only be possible if both kinks tend to zero. So, we need $L \rightarrow 1$ and $\alpha \rightarrow 0$. Thus, the structure of the nonlinearity (soft thresholding) has a severe influence on the regularization properties of the reconstruction scheme. But in contrast to the ReLU architecture, the soft thresholding operator provides the opportunity to control its similarity to a linear operator, which can be controlled via the coefficients α . In the following lemma, we discuss in more

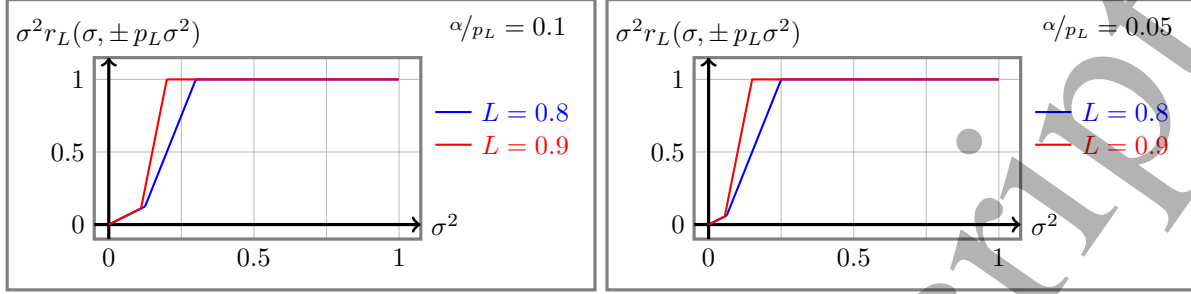


Figure 5: The plots depict graphs of $\sigma^2 r_L(\sigma, \pm p_L \sigma^2)$ as given in (4.20) and (4.22). The choice $s = p_L(\sigma^2)\sigma^2$ corresponds to the case that the test data is similar to the average training data. In this case, r_L shows a quite similar behavior as the squared soft TSVD function.

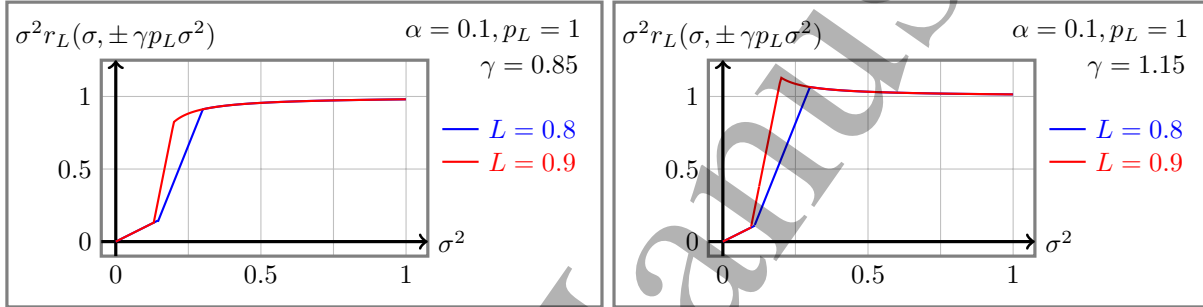


Figure 6: The plots depict graphs of $\sigma^2 r_L(\sigma, \pm \gamma p_L(\sigma^2)\sigma^2)$ as given in (4.20). The case $\gamma \neq 1$ corresponds to test data, which differs from the average training data. Especially for larger singular values σ^2 (to the right of the two kinks), the filter function shows a suboptimal behavior compared to the squared soft TSVD filter function.

detail how α can be chosen depending on the regularization parameter L to obtain the desired regularization properties.

Lemma 4.6. *Let all assumptions of Lemma 4.5 hold. Further, assume that \mathcal{A} is compact and let $p^\dagger = \sum_{j \in \mathbb{N}} p_j^\dagger v_j$ be given by $p_j^\dagger = \frac{\sum_{i=1}^N |\langle x^{(i)}, v_j \rangle|^2}{\sum_{i=1}^N |\langle x^{(i)}, v_j \rangle|}$. In addition, consider $x^\dagger \in X$ as well as strictly monotonic and continuous architecture parameter choices $\alpha_j, \beta : (0, 1) \rightarrow [0, \infty)$ with*

$$\alpha_j(L) \leq p_j^\dagger \beta(L), \text{ with } \beta(L) = \mathcal{O}((1-L)\psi(1-L)) \quad (4.23)$$

for an index function $\psi : [0, 1] \rightarrow [0, \infty)$.

Then, the local approximation property (3.1) holds for any $x^\dagger \in \mathcal{N}(A)^\perp$.

Proof. We first verify that $p_L p^\dagger \in X$ ($p_L = \sum_{j \in \mathbb{N}} p_{L,j} v_j$). For this, we exploit for each $j \in \mathbb{N}$ that

$$\frac{|\langle x^{(i)}, v_j \rangle|}{\sum_{i=1}^N |\langle x^{(i)}, v_j \rangle|} \leq 1 \quad (4.24)$$

for any i . We thus obtain via Young's inequality

$$\|p^\dagger\|^2 = \sum_{j \in \mathbb{N}} (p_j^\dagger)^2 \leq \sum_{j \in \mathbb{N}} \left(\sum_{i=1}^N |\langle x^{(i)}, v_j \rangle| \right)^2 \leq 2^{N-1} \sum_{i=1}^N \|x^{(i)}\|^2 < \infty. \quad (4.25)$$

Analogously for p_L . Due to the finite nature of the data set and the properties of α we immediately have $p_L \rightarrow p^\dagger$ for $L \rightarrow 1$ and $p_L = p^\dagger$ for L being sufficiently close to 1.

We now continue with the approximation property, making use of the notation $x_j := \langle x^\dagger, p_j \rangle$ and a sufficiently large $L < 1$ where $w_j = \min\{\frac{\alpha_j(L)}{p_j^\dagger} + 1 - \sigma_j^2, L\} \geq 0$:

$$\begin{aligned}
\|\varphi_{\theta(L)}(x^\dagger) - Ax^\dagger\|^2 &= \sum_{j \in \mathbb{N}} \left(\max(0, w_j |x_j| - \alpha_j(L)) - (1 - \sigma_j^2) |x_j| \right)^2 \\
&\leq 2 \sum_{j \in \mathbb{N}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2 + (w_j |x_j| - \max(0, w_j |x_j| - \alpha_j(L)))^2 \\
&= 2 \sum_{j \in \mathbb{N}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2 + (\max(0, w_j |x_j|) - \max(0, w_j |x_j| - \alpha_j(L)))^2 \\
&\leq 2 \sum_{j \in \mathbb{N}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2 + \alpha_j(L)^2 \\
&\leq 2 \sum_{j \in \mathbb{N}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2 + 2\beta(L)^2 \|p^\dagger\|^2
\end{aligned} \tag{4.26}$$

due to Young's inequality, the Lipschitz continuity of the ReLU function, and the assumption on α_j . We further obtain

$$\begin{aligned}
\sum_{j \in \mathbb{N}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2 &= \underbrace{\sum_{j: \sigma_j^2 \leq 1-L + \frac{\alpha_j(L)}{p_j^\dagger}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2}_{=: (I)} \\
&+ \underbrace{\sum_{j: \sigma_j^2 > 1-L + \frac{\alpha_j(L)}{p_j^\dagger}} (1 - \sigma_j^2 - w_j)^2 |x_j|^2}_{=: (II)}.
\end{aligned} \tag{4.27}$$

We thus need to derive estimates for (I) and (II).

$$\begin{aligned}
(I) &= \sum_{j: \sigma_j^2 \leq 1-L + \frac{\alpha_j(L)}{p_j^\dagger}} (1 - \sigma_j^2 - L)^2 |x_j|^2 \\
&\leq \sum_{j: \sigma_j^2 \leq 1-L + \beta(L)} (1 - \sigma_j^2 - L)^2 |x_j|^2 \\
&\leq \sum_{j: \sigma_j^2 \leq 1-L} (1 - \sigma_j^2 - L)^2 |x_j|^2 + \sum_{j: \sigma_j^2 > 1-L \wedge \sigma_j^2 \leq 1-L + \beta(L)} (1 - \sigma_j^2 - L)^2 |x_j|^2 \\
&\leq \sum_{j: \sigma_j^2 \leq 1-L} (1 - L)^2 |x_j|^2 + \sum_{j: \sigma_j^2 > 1-L \wedge \sigma_j^2 \leq 1-L + \beta(L)} \beta(L)^2 |x_j|^2 \\
&\leq (1 - L)^2 \sum_{j: \sigma_j^2 \leq 1-L} |x_j|^2 + \beta(L)^2 \|x^\dagger\|^2,
\end{aligned} \tag{4.28}$$

where we again exploit Young's inequality and the properties of a_j . For (II) we immediately obtain

$$(II) = \sum_{j: \sigma_j^2 > 1-L + \frac{\alpha_j(L)}{p_j^\dagger}} \left(\frac{\alpha_j(L)}{p_j^\dagger} \right)^2 |x_j|^2 \leq \beta(L)^2 \|x^\dagger\|^2. \tag{4.29}$$

Due to the properties of β , (II) already has the desired characteristics for $L \rightarrow 1$. In contrast, (I) requires some further calculations. Following the same line of reasoning as in the proof of Corollary 4.2 we can find

an index function ψ' for any $\varepsilon > 0$ such that

$$\begin{aligned} \sum_{j: \sigma_j^2 \leq 1-L} \langle x^\dagger, v_j \rangle^2 &\leq (1 + \varepsilon)^2 \|x^\dagger\|^2 \psi'(1-L)^2 \\ &= \mathcal{O}(\psi'(1-L)^2). \end{aligned} \quad (4.30)$$

Combining this with (4.26), (4.28) and (4.29) we obtain the desired result. \square

Remark 4.3. *Analogous to the line of reasoning in the proof of Lemma 4.3, we split the series into two sums, (I) and (II). (I) takes care of small singular values and needs to be related to a property of the element x^\dagger in terms of a certain kind of source condition. The second sum (II) is somehow related to the approximation properties of the underlying architecture. In the proof of Lemma 4.3 it is immediately zero for any x^\dagger which is due to the linear network assumption therein. In contrast, in the previous proof we had to carefully control the behavior of this term since it is strongly influenced by the structure of the nonlinearity.*

In general, the previous lemma serves as an example of the interplay between the approximation capabilities of the network and the resulting regularization properties.

5 Numerical experiments

In this section, we present numerical results in order to compare our theoretical findings from Section 3 and Section 4 to its corresponding numerical implementations and extend these by experiments in a more general setting by learning from noisy measurements. To this end, we use a Radon transform with 30 angles and 41 detector pixels as our forward operator. We discretize the Radon transform for 28×28 px images. This results in a linear forward operator $\mathcal{A} : \mathbb{R}^{28 \times 28} \rightarrow \mathbb{R}^{30 \times 41}$. For the training of the different iResNet architectures, we use the well-known MNIST dataset of handwritten digits [25]. This dataset is split into 60 000 images for training and 10 000 images for testing. For our experiments, we treat the 28×28 images as column vectors of length 784 and use fully connected layers in all network architectures. We optimize the networks using Adam [23] and a learning rate scheduling scheme.

There are multiple ways to satisfy the Lipschitz constraint during training. Bungert et al. [11] use an additional penalty term in the loss function. However, this method does not strictly enforce the constraint. Behrmann et al. [8] use contractive nonlinearities and only constrain the Lipschitz constant of each individual linear mapping. They compute the Lipschitz constant using a power iteration and normalize the weights after each training step. We observe an improvement of convergence when directly parameterizing the weights to fulfill a specific Lipschitz constant by extending the approach in [30].

In Section 5.1, we show experiments on the local approximation property in Theorem 3.1. To this end, we use the diagonal architecture proposed in Section 3.2. The resulting data-dependent filter functions are visualized in Section 5.2, and the convergence property is considered in Section 5.3.

We implement each subnetwork $f_{\theta,j}$, $j = 1, \dots, n$, (see (3.12)) as a small fully-connected neural network with independent weights for each j , where n denotes the total number of singular values in the discrete setting. Each subnetwork consists of three layers, where the first two layers $f_{\theta,j}^k$, $k = 1, 2$ each contain 35 hidden neurons, and its final layer $f_{\theta,j}^3$ contains 1 neuron. Every layer is equipped with a linear matrix multiplication and corresponding additive bias and a ReLU activation function ($k = 1, 2$). Accordingly, each subnetwork has 1366 trainable parameters. An illustration of the architecture is provided in Figure 7. Altogether, the parameters of the subnetworks determine the parameters $\theta \in \Theta(L)$ of $\varphi_\theta = \text{Id} - f_\theta$, where $\Theta(L)$ includes the network parameters as well as the Lipschitz constraint being realized by constraints on the network parameter. Here, we enforce the Lipschitz constants $\text{Lip}(f_{\theta,j}) \leq L$, $j = 1, \dots, n$, by constraining $\text{Lip}(f_{\theta,j}^k) \leq 1$ for $k = 1, 2$ and $\text{Lip}(f_{\theta,j}^3) \leq L$. In the following, we thus write $\theta(L)$ in order to give emphasis to the regularization parameter. Our training objective is the minimization of the *approximation loss* (4.1), i.e., minimize the loss

$$l(\varphi_\theta, A) = \frac{1}{N} \sum_i \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \quad (5.1)$$

subject to $\text{Lip}(f_\theta) \leq L$.

The source code corresponding to the experiments in this section is available at <https://gitlab.informatik.uni-bremen.de/inn4ip/iresnet-regularization>.

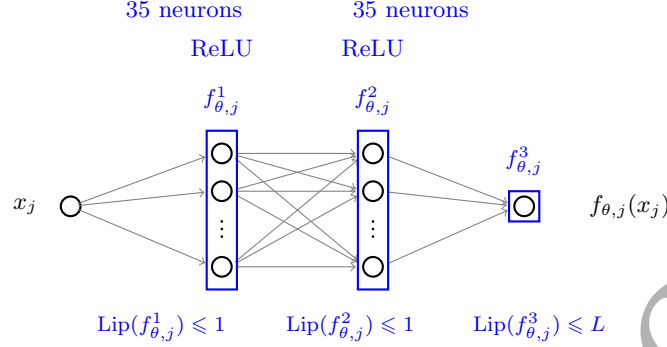


Figure 7: Illustration of the subnetworks $f_{\theta,j} : \mathbb{R} \rightarrow \mathbb{R}$, $j = 1, \dots, n$. Each having 1366 trainable parameters.

5.1 Local approximation property

According to Theorem 3.1, we expect $\varphi_{\theta(L)}^{-1}$ to act as a regularizer as soon as the local approximation property (3.1) is fulfilled. Note that the architecture does not change for varying L in our experiments such that we write $\varphi_{\theta(L)}$ instead of $\varphi_{\theta(L),L}$. To be able to observe this behavior of our trained network, the local approximation property needs to be verified with respect to the trained network parameters. Therefore, we evaluate the approximation error with respect to fixed data samples

$$\mathcal{E}_{x^{(i)}}(\varphi_{\theta(L)}, A) = \|\varphi_{\theta(L)}(x^{(i)}) - Ax^{(i)}\| \quad (5.2)$$

as well as the mean approximation error over the test data set

$$\mathcal{E}_{\text{mean}}(\varphi_{\theta(L)}, A) = \frac{1}{N} \sum_i \|\varphi_{\theta(L)}(x^{(i)}) - Ax^{(i)}\|. \quad (5.3)$$

Figure 8 indicates superlinear convergence of $\mathcal{E}_{\text{mean}}$ for $L \rightarrow 1$, i.e., the existence of an index function ψ such that the local approximation property (3.1) is satisfied within our test data set on average. This underpins the capability of the chosen network architecture and training to locally approximate the operator A in terms of (3.1). Furthermore, the evaluation of $\mathcal{E}_{x^{(1)}}$ shows that the chosen sample $x^{(1)}$ fulfills the superlinear convergence and behaves very similarly to the mean error over the test data set. However, from the selection of data sample $x^{(2)}$ and corresponding error $\mathcal{E}_{x^{(2)}}$, we notice that the local approximation property does not hold for all samples of our test data set. Figure 8 additionally shows that some coefficients $x_j^{(2)}$ of $x^{(2)}$, corresponding to large singular values, severely deviate from the corresponding mean values with respect to the data distribution. This effect is reduced for $x^{(1)}$. Therefore, the slow, respectively non-, convergence of $\mathcal{E}_{x^{(2)}}$ could possibly be explained by the fact that structures outside of the core of the distribution of chosen singular values have not been learned properly during network training.

5.2 Data-dependent filter functions for diagonal architecture

For the experiments in this and the subsequent section, we train networks $\varphi_{\theta(L)}$ with Lipschitz bounds $L \in \{L_m = 1 - 1/3^m \mid m = 1, \dots, 5\}$. We also include additive Gaussian noise in the network training via minimizing the approximation loss (4.1). More precisely, for each training sample $x^{(i)}$ we generate $Ax^{(i)} + \eta^{(i)}$ with $\eta^{(i)} \sim \mathcal{N}(0, \delta_\ell \text{Id})$ and relative noise level $\delta_\ell = \hat{\delta}_\ell \cdot \text{std}_{\text{MNIST}}$, where $\text{std}_{\text{MNIST}}$ denotes the averaged standard deviation of the coefficients $\langle x^{(i)}, v_j \rangle$ of the MNIST data set (standard deviation with respect to i , mean with respect to j) and

$$\hat{\delta}_\ell = \begin{cases} \left(\frac{1}{3}\right)^{7-\ell} & \text{for } 0 < \ell < 7 \\ 0 & \text{for } \ell = 0. \end{cases} \quad (5.4)$$

The loss function l then amounts to

$$\min_{\theta \in \Theta(L)} l(\varphi_\theta, A) = \min_{\theta \in \Theta(L)} \frac{1}{N} \sum_{i=1}^N \|\varphi_\theta(x^{(i)}) - Ax^{(i)} - \eta^{(i)}\|^2. \quad (5.5)$$

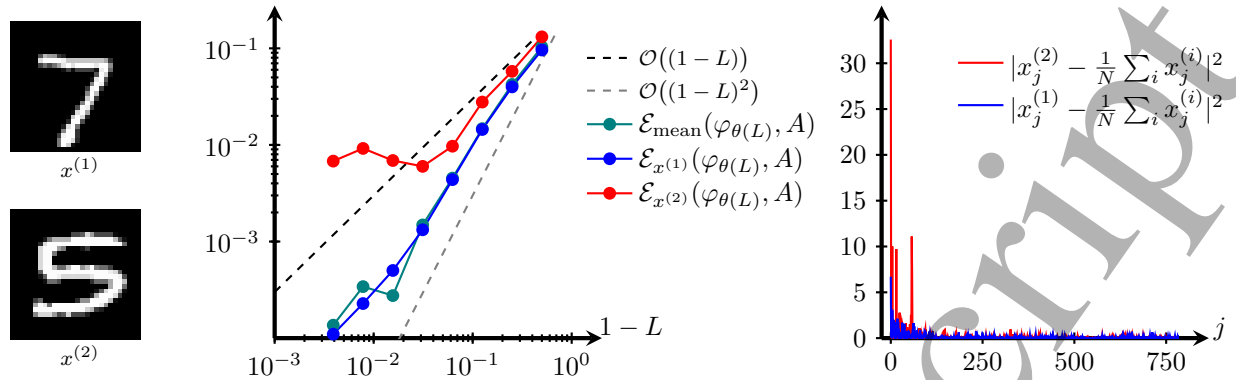


Figure 8: Test samples $x^{(1)}$ and $x^{(2)}$ (left). Evaluations of $\mathcal{E}_{\text{mean}}(\varphi_{\theta(L_m)}, A)$, $\mathcal{E}_{x^{(1)}}(\varphi_{\theta(L_m)}, A)$ and $\mathcal{E}_{x^{(2)}}(\varphi_{\theta(L_m)}, A)$ for $L_m = 1 - 1/2^m$ with $m = 1, \dots, 8$ on the MNIST test data set (middle). Squared absolute differences of the coefficients $x_j^{(k)} = \langle x^{(k)}, v_j \rangle$ for $k = 1, 2$ to the mean value of corresponding coefficients in the test data set (right).

Note that this includes (4.1) in the noise-free case. Trained networks on noisy samples with noise level δ with a particular Lipschitz bound L are denoted by $\varphi_{\theta(L, \delta)}$. The noise-free training outcome is denoted by $\varphi_{\theta(L)}$.

Utilizing identical network architectures as in the previous subsection, we evaluate the learned filter functions of chosen networks. Analogously to (3.19) the reconstruction can be written as

$$T_L(y) = \varphi_{\theta}^{-1}(\mathcal{A}^* y) = \hat{b}_L + \sum_{j \in \mathbb{N}} \hat{r}_L(\sigma_j^2, \sigma_j \langle y, u_j \rangle) \sigma_j \langle y, u_j \rangle v_j \quad (5.6)$$

for $\hat{b}_L \in X = \mathbb{R}^n$. As stated in Section 3.2, the learned filter function of a trained network with diagonal architecture follows immediately from its subnetworks. Due to the additional bias, we make a small adaption to (3.17), which gives

$$(\text{Id} - f_{\theta, j})^{-1}(s) - \hat{b}_{L, j} = r_L(\sigma_j^2, s) s \quad \text{for } s \in \mathbb{R}, \quad (5.7)$$

where each entry $\hat{b}_{L, j} = \langle b_L, v_j \rangle$ corresponds to the axis intercept of a subnetwork $\varphi_{\theta, j}$, $j = 1, \dots, n$, i.e., $\hat{b}_{L, j} = (\text{Id} - f_{\theta, j})^{-1}(0)$. Since $f_{\theta, j}$ corresponds to σ_j or σ_j^2 , respectively, we also write $\hat{b}_{L, j} = \hat{b}_L(\sigma_j^2)$.

Adapting to the mean values with respect to the distribution of each coefficient, we compute

$$\mu_j := \frac{1}{N} \sum_{i=1}^N \langle x^{(i)}, v_j \rangle \quad \text{for } j = 1, \dots, n \quad (5.8)$$

and evaluate the filter function with respect to its second variable at the continuous extension $\sigma^2 \mu(\sigma^2)$ with $\mu(\sigma_j^2) = \mu_j$, $j = 1, \dots, n$.

From Figure 9 we notice that the regularization of smaller singular values increases with decreasing Lipschitz bound L for the mean MNIST data sample. This is in line with the regularization theory, as L serves as our regularization parameter. Thus, the filter plots in Figure 9 visualize how L acts as a regularization parameter. The trained diagonal networks $\varphi_{\theta(L_m)}$ for L_2 and L_5 show a slightly different behavior at $\sigma_j^2 \approx 0.27$. The distributions of the coefficients $\langle x, v_j \rangle_{j=1, \dots, n}$ are treated independently and in this particular singular value we observed a wider distribution of coefficients $\langle x, v_j \rangle$ in the dataset. The inherent structure within the dataset might be one possible explanation for this outlier. In general, when neglecting the previously mentioned outlier, for the mean MNIST sample one can observe a similarity to the squared soft TSVD filter function (see Section 4.2) to some extent. In addition, the observed decay of $\|\hat{b}_L\|$ with increased L is also in line with the necessary condition for a classical filter function with bias to become a convergent regularization (see Lemma 3.3). The observed increase for the largest L is most likely caused by a numerical instability when evaluating $(\text{Id} - f_{\theta, j})^{-1}(0)$ via the fixed point iteration performed for 30 iterations.

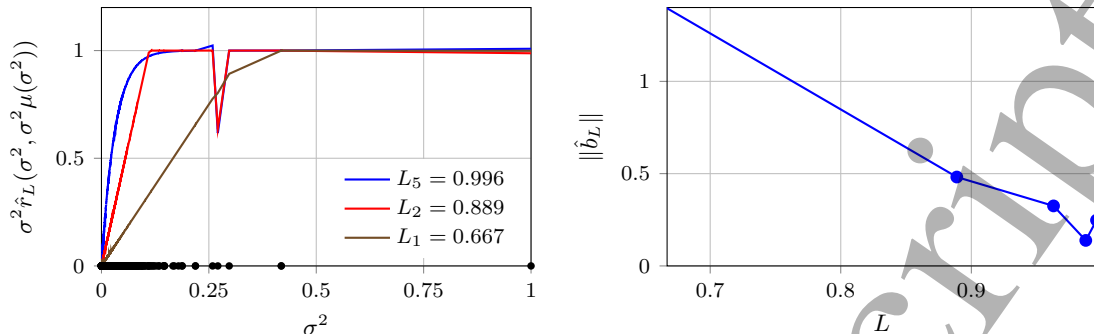


Figure 9: Learned filter functions $r_L(\sigma^2, \sigma^2 \mu(\sigma^2))$ of the iResNet with diagonal architecture and zero noise level δ_0 for different Lipschitz constraints L (left) where the eigenvalues σ_j^2 are highlighted as black bullets on the x-axis. $\|\hat{b}_L\|$ corresponding to the axis intersection $\hat{b}_L = \sum_j \hat{b}_{L,j} v_j$, evaluated for $L_m, m = 1, \dots, 5$ (right).

Figure 10 includes reconstructions of a fixed sample from the test data set. This example illustrates the effect of a regularization induced by the Lipschitz bound even if the training takes noise into account. It becomes stronger for small values of L , which coincides with our previous observations. Reconstructions resulting from increased noise levels require stronger regularizations in order to improve reconstruction quality. Therefore, the best reconstructions in case of $\hat{\delta}_0$ and $\hat{\delta}_1$ result from $\varphi_{\theta(L_3, \delta_0)}^{-1}$ and $\varphi_{\theta(L_3, \delta_1)}^{-1}$. In comparison, $\varphi_{\theta(L_2, \delta_4)}^{-1}$ and $\varphi_{\theta(L_2, \delta_3)}^{-1}$ provide improved reconstructions for the cases $\hat{\delta}_4$ and $\hat{\delta}_3$. Moreover, at L_1 we notice similar blurred structures in the background of the reconstructed digit for all noise levels. One might argue that its structure compares to a handwritten digit itself, making the learned background pattern being encoded in the bias \hat{b}_L suitable to the data set. These additional observations from Figure 10 indicate a dependency of the regularization on learned data structures. The corresponding filter functions being illustrated in the right column of Figure 10 show a similar behavior for all training noise levels which underpins that the outcome of the approximation training is independent of the noise for a sufficiently large number of training samples. The outlier in the filter function for the mean sample $(\mu_j)_j$ can also be observed in Figure 10. In addition, this test sample has a slightly different behavior with respect to the second singular value. Note that the seemingly linear behavior for $\sigma_j^2 > 0.4$ is only due to the fact that this is the linear continuous extension between the first and second singular value. In summary, the resulting filter behaves similarly to the squared soft TSVD filter independent of the training noise and with exceptions at two singular values.

5.3 Convergence property

After investigating approximation capabilities of trained networks with respect to the operator A in Section 5.1 in the noise-free case and extending the training to the noisy case in Section 5.2, we continue verifying the convergence property with respect to different noise levels. We analyze the convergence property by computing the averaged reconstruction error

$$\text{MSE}_{\text{reco}}^{\delta_\ell}(\varphi_{\theta(L, \delta)}, A) = \frac{1}{N} \sum_{i=1}^N \|x^{(i)} - \varphi_{\theta(L, \delta)}^{-1}(Ax^{(i)} + \eta^{(i)})\|^2, \quad (5.9)$$

including the noise levels δ_ℓ and noise samples $\eta^{(i)} \sim \mathcal{N}(0, \delta_\ell Id)$ in the reconstruction process where the network has been trained on noise level δ . We thus can evaluate the noise-free training case, which solely aims to impart data dependency, on reconstructions from noisy data, and the noisy training case where training and test noise share the same level. Reconstructions, i.e., the inverse of the iResNet, are obtained by using 30 fixed-point iterations.

Figure 11 shows that $\varphi_{\theta(L, \delta_0)}^{-1}$ as well as $\varphi_{\theta(L, \delta_\ell)}^{-1}$ provides more accurate reconstructions with respect to (5.9) at large L and low noise levels, whereas this behavior is reversed for decreasing Lipschitz bound and increasing noise levels. This is consistent with the regularization theory and the visualized reconstructions in Figure 10, as high noise levels require strong regularizations and vice versa. The behavior of

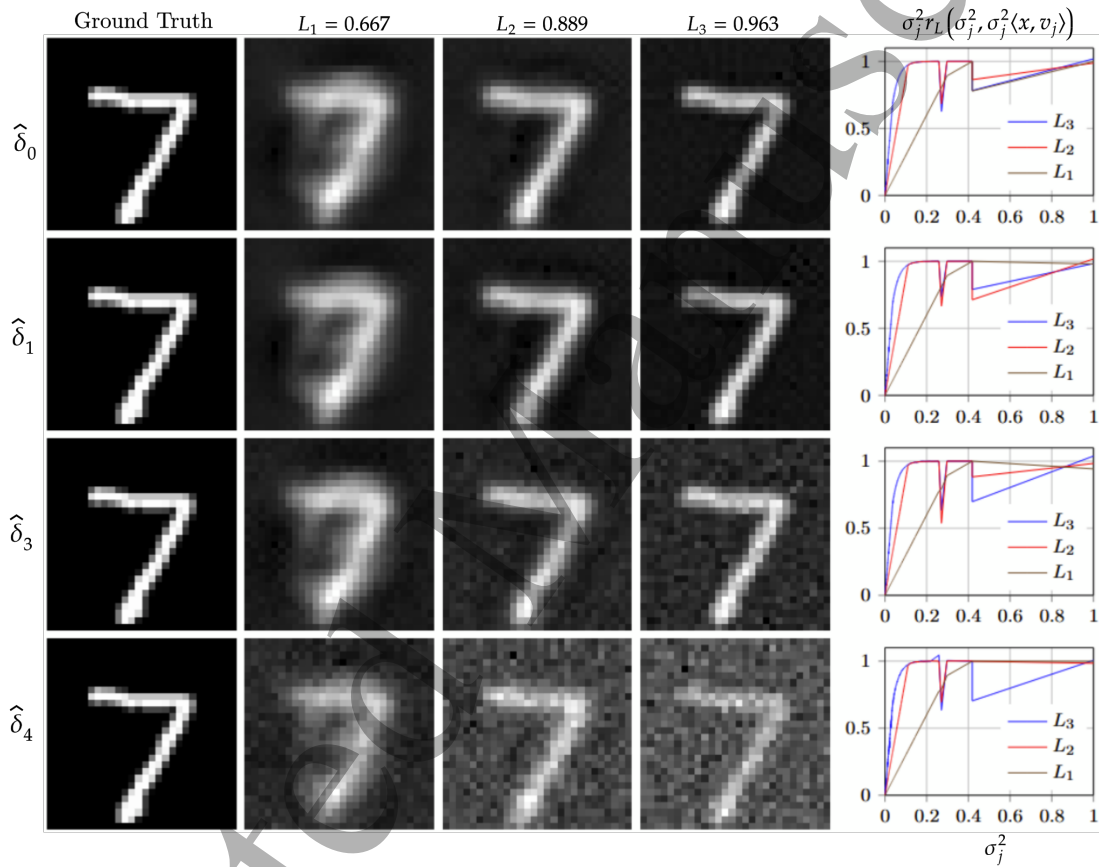


Figure 10: Reconstructions of an MNIST sample \tilde{x} from the test data set by computing $\varphi_{\theta(L_m, \delta_\ell)}^{-1}(A\tilde{x} + \tilde{\eta})$ with $\tilde{\eta} \sim \mathcal{N}(0, \delta_\ell Id)$ for Lipschitz bounds L_m (columns) and noise levels $\delta_\ell = \hat{\delta}_\ell \cdot \text{std}_{\text{MNIST}}$ with $\ell = 0, 1, 3, 4$ (rows). The last column depicts the filter functions at $L_1, L_2,$ and L_3 for each noise level with respect to the sample \tilde{x} .

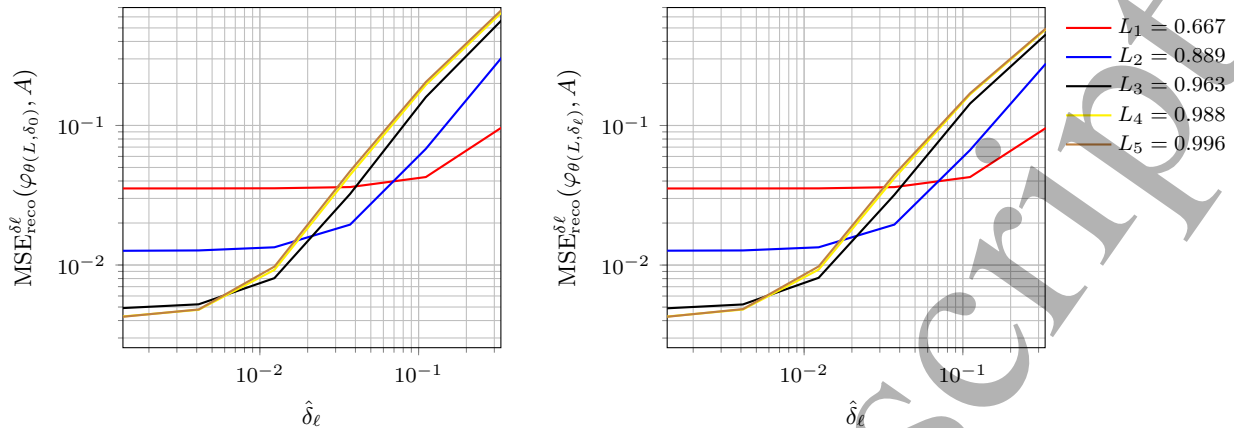


Figure 11: Reconstruction errors $\text{MSE}_{\text{reco}}^{\delta_\ell}(\varphi_{\theta(L, \delta_0)}, A)$ with training on noise-free samples and reconstruction from noisy samples (*left*) and $\text{MSE}_{\text{reco}}^{\delta_\ell}(\varphi_{\theta(L, \delta_\ell)}, A)$ with training and reconstruction on the same noise level δ_ℓ (*right*) for $\varphi_{\theta(L_m, \delta_\ell)}$ over different noise levels $\delta_\ell = \hat{\delta}_\ell \cdot \text{std}_{\text{MNIST}}$, $\ell = 0, \dots, 6$, and Lipschitz bounds L_m , $m = 1, \dots, 5$.

$\text{MSE}_{\text{reco}}^{\delta_\ell}(\varphi_{\theta(L, \delta_\ell)}, A)$ for small L and small noise levels δ_ℓ is rather intuitive, since its approximation capability is limited as a consequence of strong regularization. In addition, from Figure 11 one can also extract a suitable candidate for the parameter choice $L(\delta)$ to obtain convergence. The similarity in the behavior of $\varphi_{\theta(L, \delta_0)}^{-1}$ and $\varphi_{\theta(L, \delta_\ell)}^{-1}$ underpins that the outcome of the approximation training is independent of noise if data and noise are independent.

6 Discussion and Outlook

In the present work, we developed and investigated the regularization theory for the proposed iResNet reconstruction approach providing a learned method from data samples. The network’s local approximation property is fundamental to delivering a convergent regularization scheme. It comprises approximation properties of the architecture and training, a definition of solution type, and a source condition. The approximation loss used for training is motivated by this property. In the most general version, the framework can be understood as a fully-learned end-to-end reconstruction scheme with minor limitations as it relies on the concatenation with \mathcal{A}^* , i.e., some a priori knowledge on the forward operator is at least included in a hand-crafted way in the reconstruction scheme. Introducing a diagonal architecture type relying on the SVD of the forward operator \mathcal{A} allowed for an analytical investigation of the resulting learned nonlinear spectral reconstruction method, which becomes a convergent regularization when fulfilling the local approximation property. The analysis of trained shallow architectures revealed the link between the classical linear filter-based regularization theory and the concept of the local approximation property, and it illustrated the interplay between the approximation capability of the nonlinear network and the source condition. In addition, we validated and extended the theoretical findings by a series of numerical experiments on the MNIST data set and provided further insights into the learned nonlinear spectral regularization, such as similarity to the analytically determined linear regularization (squared soft TSVD) and data-dependency of the learned regularization.

Our iResNet method using the diagonal architecture can be seen as a generalization of the learned linear spectral regularization considered in [22]. Using a different loss, which measures the reconstruction error, the authors of [22] obtain learned filter functions corresponding to Tikhonov regularization with data- and noise-dependent regularization parameters for each singular vector. An extension of the iResNet approach to this kind of training loss is thus desirable for future comparison.

The present work also serves as a starting point for various future research directions:

As we obtain the inverse of the residual network via a fixed point iteration, findings in the context of the iResNet reconstruction may be related to learned optimization methods [6], maximally monotone

operators [37], or potentially to plug-and-play methods [16]. Investigations at the interface to these methods are far beyond the present work and will remain future work.

The concept of a local criterion to guarantee convergence of the iResNet reconstruction method provides further opportunities to investigate the regularization properties of learned reconstruction schemes. In a different context, the authors of [5] also consider a local property. They consider a learned projection onto subspaces concatenated with the forward operator, i.e., they consider a projected linear problem and combine it with Tikhonov regularization. There, the authors guarantee convergence of the regularization method by the Tikhonov theory, including a different but also localized assumption. The findings in the present work and the literature thus reveal the necessity to explicitly take the approximation capabilities of the learned schemes and the training procedure into account. Immediate future research includes a comprehensive numerical study on how architectural choices such as particular activation functions, e.g., being strictly monotone but nonlinear, influence the desired regularization properties and reconstruction quality. Further potential future investigations in this context are twofold. On the one hand, one may exploit universal approximation properties of the underlying architectures to guarantee convergence for a sufficiently large subset of X , e.g., by verifying the local approximation property. On the other hand, one may adapt the definition of regularization to the approximate and data-based nature of the learning-based setting.

Besides the more general open research questions at the interface to other methods and data-based concepts, the iResNet reconstruction approach itself provides various potential future works. The observed similarity of the learned nonlinear filter functions in the numerical experiments to the analytical solution of the affine network with bias (Section 4.2) immediately raises the question: How much does the data's structure influence the resulting reconstruction scheme? Phrased differently, one needs to test the limits of the proposed approximation training. Here, it would be desirable to change the perspective to a Bayesian one considering data and noise distributions and further investigate the training outcome, including additional variations of the training loss and its influence on the local approximation property. Natural extension and generalizations, such as including learned solution types, e.g., by combining our approach with null space networks [41] or relaxing the diagonal assumption of the architecture, remain future work.

In summary, the present first work on learned iResNet regularization schemes builds the basis for various future works of theoretical and numerical nature.

Acknowledgments

A. Denker, N. Heilenkötter, and M. Iske acknowledge the support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project number 281474342/GRK2224/2. T. Kluth acknowledges support from the DELETO project funded by the Federal Ministry of Education and Research (BMBF, project number 05M20LBB).

A Proofs of Section 4

A.1 Proof of Lemma 4.1

Equivalent filter function to the one-parameter-network.

Proof. At first, we observe that the Lipschitz constraint in (4.1) is fulfilled for $|k| \leq L$. Thus, (4.1) is equivalent to

$$\begin{aligned}
 & \min_{|k| \leq L} \sum_i \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \\
 \Leftrightarrow & \min_{|k| \leq L} \sum_i \|x^{(i)} - kx^{(i)} + kAx^{(i)} - Ax^{(i)}\|^2 \\
 \Leftrightarrow & \min_{|k| \leq L} \sum_i (1-k)^2 \|x^{(i)} - Ax^{(i)}\|^2.
 \end{aligned} \tag{A.1}$$

Since there is one $x^{(i)}$ s.t. $Ax_i \neq x^{(i)}$ the solution is obviously $k = L$ and we have $\varphi_\theta(x) = x - L(x - Ax)$.

Now we use the SVD of \mathcal{A} to solve $\varphi_\theta(\bar{x}) = z$ for $z \in \mathcal{R}(A)$ component wise. For all $j \in \mathbb{N}$ it holds

$$\begin{aligned} & \langle \bar{x} - L(\bar{x} - A\bar{x}), v_j \rangle = \langle z, v_j \rangle \\ \Leftrightarrow & (1 - L)\langle \bar{x}, v_j \rangle + L\langle \bar{x}, Av_j \rangle = \langle z, v_j \rangle \\ \Leftrightarrow & (1 - L + L\sigma_j^2)\langle \bar{x}, v_j \rangle = \langle z, v_j \rangle \\ \Leftrightarrow & \langle \bar{x}, v_j \rangle = \frac{1}{1 - L + L\sigma_j^2} \langle z, v_j \rangle \end{aligned} \quad (\text{A.2})$$

Thus, the filter function

$$r_L(\sigma^2, s) = \frac{1}{1 - L + L\sigma^2} = \frac{1}{L} \cdot \frac{1}{\alpha + \sigma^2}, \quad \alpha = \frac{1 - L}{L} \quad (\text{A.3})$$

fulfills (3.15). \square

A.2 Proof of Lemma 4.2

Equivalent filter function to the linear network.

Proof. At first, we observe that the Lipschitz constraint in (4.1) is fulfilled if the eigenvalues $(w_j)_j$ of W are restricted by $|w_j| \leq L$. Defining $\Theta_L = \{(W, b) \in \Theta \mid |w_j| \leq L\}$, (4.1) is equivalent to

$$\begin{aligned} & \min_{\theta \in \Theta_L} \sum_{i=1}^N \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \\ \Leftrightarrow & \min_{(W, b) \in \Theta_L} \sum_{i=1}^N \|x^{(i)} - Wx_i - b - Ax^{(i)}\|^2 \\ \Leftrightarrow & \min_{(W, b) \in \Theta_L} \sum_{i=1}^N \|P_{\mathcal{N}(A)}(x^{(i)})\|^2 + \|P_{\mathcal{N}(A)^\perp}(x^{(i)}) - Wx_i - b - Ax^{(i)}\|^2. \end{aligned} \quad (\text{A.4})$$

The part with $P_{\mathcal{N}(A)}(x^{(i)})$ is independent of W and b and therefore not relevant. Furthermore, it holds

$$\begin{aligned} P_{\mathcal{N}(A)^\perp}(x^{(i)}) - Wx_i - b - Ax^{(i)} &= \sum_{j \in \mathbb{N}} \langle x, v_j \rangle v_j - \sum_{j \in \mathbb{N}} (w_j \langle x, v_j \rangle + b_j) v_j - \sum_{j \in \mathbb{N}} \sigma_j^2 \langle x, v_j \rangle v_j \\ &= \sum_{j \in \mathbb{N}} \left((1 - w_j - \sigma_j^2) \langle x, v_j \rangle - b_j \right) v_j. \end{aligned} \quad (\text{A.5})$$

Thus, the minimizing problem can be written as

$$\min_{|w_j| \leq L, (b_j) \in \ell^2} \sum_{i=1}^N \sum_{j \in \mathbb{N}} \left((1 - w_j - \sigma_j^2) \langle x, v_j \rangle - b_j \right)^2. \quad (\text{A.6})$$

We can solve this problem for each j separately. For determining b_j we set the derivative to zero and get

$$\begin{aligned} 0 &\stackrel{!}{=} -2 \sum_{i=1}^N \left((1 - w_j - \sigma_j^2) \langle x^{(i)}, v_j \rangle - b_j \right) \\ \Leftrightarrow & b_j \stackrel{!}{=} \frac{1}{N} (1 - w_j - \sigma_j^2) \sum_{i=1}^N \langle x^{(i)}, v_j \rangle = (1 - w_j - \sigma_j^2) \mu_j. \end{aligned} \quad (\text{A.7})$$

Since for every i , the sequence $(\langle x^{(i)}, v_j \rangle)_{j \in \mathbb{N}}$ is in ℓ^2 , $(b_j) \in \ell^2$ is also fulfilled. It remains to solve

$$\min_{|w_j| \leq L} \sum_{i=1}^N \left((1 - w_j - \sigma_j^2) (\langle x, v_j \rangle - \mu_j) \right)^2. \quad (\text{A.8})$$

By assumption, for every v_j there is at least one $x^{(i)}$ s.t. $\langle x^{(i)}, v_j \rangle \neq \mu_j$, thus, the problem can be simplified to

$$\min_{|w_j| \leq L} (1 - w_j - \sigma_j^2)^2 \quad \forall j \in \mathbb{N}. \quad (\text{A.9})$$

The obvious solution is $w_j = \min\{1 - \sigma_j^2, L\}$, since $1 - \sigma_j^2$ is by assumption always positive.

Now we use the SVD of \mathcal{A} to solve $\varphi_\theta(\bar{x}) = z$ for some $z \in \mathcal{R}(A)$ component-wise. For all $j \in \mathbb{N}$ it holds

$$\begin{aligned} & \langle \bar{x}, v_j \rangle - w_j \langle \bar{x}, v_j \rangle - b_j = \langle z, v_j \rangle \\ \Leftrightarrow & (1 - w_j) \langle \bar{x}, v_j \rangle = \langle z, v_j \rangle + b_j \\ \Leftrightarrow & \langle \bar{x}, v_j \rangle = \frac{1}{1 - w_j} \langle z, v_j \rangle + \frac{b_j}{1 - w_j} \\ \Leftrightarrow & \langle \bar{x}, v_j \rangle = \frac{1}{\max\{\sigma_j^2, 1 - L\}} \langle z, v_j \rangle + \frac{\max\{0, 1 - L - \sigma_j^2\}}{\max\{\sigma_j^2, 1 - L\}} \mu_j. \end{aligned} \quad (\text{A.10})$$

Thus, using the filter function and bias

$$\hat{r}_L(\sigma^2) = \frac{1}{\max\{\sigma^2, 1 - L\}}, \quad \hat{b}_L = \sum_{\sigma_j^2 < 1 - L} \frac{1 - L - \sigma_j^2}{1 - L} \mu_j v_j, \quad (\text{A.11})$$

the reconstruction schemes $\varphi_\theta^{-1} \circ \mathcal{A}^*$ and T_L from (3.19) are equivalent. \square

A.3 Proof of Lemma 4.4

Equivalent filter function to the ReLU-network.

Proof. At first, we observe that the Lipschitz constraint in (4.1) is fulfilled if the eigenvalues $(w_j)_j$ of W are restricted by $|w_j| \leq L$. Thus, (4.1) is equivalent to

$$\begin{aligned} & \min_{(w_j), |w_j| \leq L} \sum_i \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_i \|x^{(i)} - \phi(Wx_i) - Ax^{(i)}\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_i \left\| P_{\mathcal{N}(A)}(x^{(i)}) + \sum_{j \in \mathbb{N}} \langle x^{(i)}, v_j \rangle v_j - \sum_{j \in \mathbb{N}} \max\{0, w_j \langle x^{(i)}, v_j \rangle\} v_j - \sum_{j \in \mathbb{N}} \sigma_j^2 \langle x^{(i)}, v_j \rangle v_j \right\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_i \left\| P_{\mathcal{N}(A)}(x^{(i)}) \right\|^2 + \left\| \sum_{j \in \mathbb{N}} \left((1 - \sigma_j^2) \langle x^{(i)}, v_j \rangle - \max\{0, w_j \langle x^{(i)}, v_j \rangle\} \right) v_j \right\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_i \sum_{j \in \mathbb{N}} \left((1 - \sigma_j^2) \langle x^{(i)}, v_j \rangle - \max\{0, w_j \langle x^{(i)}, v_j \rangle\} \right)^2. \end{aligned} \quad (\text{A.12})$$

At first we focus on the components $\langle x^{(i)}, v_j \rangle$ which are negative or zero. They do not influence the solution of the problem since

$$\min_{|w_j| \leq L} \left(-(1 - \sigma_j^2) - \max\{0, -w_j\} \right)^2 \quad \forall j \in \mathbb{N} \quad (\text{A.13})$$

is solved by any $w_j \in [0, L]$, as $1 - \sigma_j^2$ is by assumption always positive. Thus, it suffices to consider the cases $\langle x^{(i)}, v_j \rangle > 0$, which lead to

$$\min_{|w_j| \leq L} (1 - \sigma_j^2 - \max\{0, w_j\})^2 \quad \forall j \in \mathbb{N}. \quad (\text{A.14})$$

The obvious solution is $w_j = \min\{1 - \sigma_j^2, L\}$.

Now we use the SVD of \mathcal{A} to solve $\varphi_\theta(\bar{x}) = z$ for $z \in \mathcal{R}(A)$ or

$$\bar{x} - \sum_{j \in \mathbb{N}} \max\{0, w_j \langle \bar{x}, v_j \rangle\} v_j = \sum_{j \in \mathbb{N}} \langle z, v_j \rangle v_j, \quad (\text{A.15})$$

respectively, component-wisely. For all $j \in \mathbb{N}$ it holds

$$\begin{aligned} & \langle \bar{x}, v_j \rangle - \max\{0, w_j \langle \bar{x}, v_j \rangle\} = \langle z, v_j \rangle \\ \Leftrightarrow & \min\{\langle \bar{x}, v_j \rangle, (1 - w_j) \langle \bar{x}, v_j \rangle\} = \langle z, v_j \rangle. \end{aligned} \quad (\text{A.16})$$

Since $1 - w_j > 0$, the signs of $\langle \bar{x}, v_j \rangle$ and $\langle z, v_j \rangle$ must coincide. In case of positive signs, the minimum on the left hand side takes the value $(1 - w_j) \langle \bar{x}, v_j \rangle$, in case of negative signs, it is $\langle \bar{x}, v_j \rangle$. Thus, we get

$$\langle \bar{x}, v_j \rangle = \begin{cases} \frac{1}{1 - w_j} \langle z, v_j \rangle & \text{if } \langle z, v_j \rangle \geq 0, \\ \langle z, v_j \rangle & \text{if } \langle z, v_j \rangle < 0. \end{cases} \quad (\text{A.17})$$

Using $1 - w_j = \max\{\sigma_j^2, 1 - L\}$, it can be seen that

$$r_L(\sigma^2, s) = \begin{cases} \frac{1}{\max\{\sigma^2, 1 - L\}} & \text{if } s \geq 0, \\ 1 & \text{if } s < 0 \end{cases} \quad (\text{A.18})$$

fulfills (3.15). \square

A.4 Proof of Lemma 4.5

Equivalent filter function to the soft-thresholding-network.

Proof. At first, we observe that the Lipschitz constraint in (4.1) is fulfilled if the eigenvalues $(w_j)_j$ of W are restricted by $|w_j| \leq L$. Besides, $f_\theta(x)$ can be written as

$$\begin{aligned} \phi_\alpha(Wx) &= \sum_{j \in \mathbb{N}} \text{sign}(\langle Wx, v_j \rangle) \max(0, |\langle Wx, v_j \rangle| - \alpha_j) v_j \\ &= \sum_{j \in \mathbb{N}} \text{sign}(w_j \langle x, v_j \rangle) \max(0, |w_j \langle x, v_j \rangle| - \alpha_j) v_j. \end{aligned} \quad (\text{A.19})$$

For any $x^{(i)}$ from the training data set, it follows

$$\begin{aligned} & x^{(i)} - \phi_\alpha(Wx^{(i)}) - Ax^{(i)} \\ &= P_{\mathcal{N}(A)}(x^{(i)}) + \sum_{j \in \mathbb{N}} \langle x^{(i)}, v_j \rangle v_j - \sum_{j \in \mathbb{N}} \text{sign}(w_j \langle x^{(i)}, v_j \rangle) \max(0, |w_j \langle x^{(i)}, v_j \rangle| - \alpha_j) v_j - \sum_{j \in \mathbb{N}} \sigma_j^2 \langle x^{(i)}, v_j \rangle v_j \\ &= P_{\mathcal{N}(A)}(x^{(i)}) + \sum_{j \in \mathbb{N}} \left[x_j^{(i)} - \text{sign}(w_j x_j^{(i)}) \max(0, |w_j x_j^{(i)}| - \alpha_j) - \sigma_j^2 x_j^{(i)} \right] v_j \end{aligned} \quad (\text{A.20})$$

where $x_j^{(i)} := \langle x^{(i)}, v_j \rangle$. Therefore, (4.1) is equivalent to

$$\begin{aligned} & \min_{(w_j), |w_j| \leq L} \sum_i \|\varphi_\theta(x^{(i)}) - Ax^{(i)}\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_i \left\| x^{(i)} - \phi_\alpha(Wx^{(i)}) - Ax^{(i)} \right\|^2 \\ \Leftrightarrow & \min_{(w_j), |w_j| \leq L} \sum_j \sum_i \left(x_j^{(i)} - \text{sign}(w_j x_j^{(i)}) \max(0, |w_j x_j^{(i)}| - \alpha_j) - \sigma_j^2 x_j^{(i)} \right)^2. \end{aligned} \quad (\text{A.21})$$

For each j , we have to find the solution of

$$\begin{aligned}
& \min_{|w_j| \leq L} \sum_i \left(x_j^{(i)} - \text{sign}(w_j x_j^{(i)}) \max(0, |w_j x_j^{(i)}| - \alpha_j) - \sigma_j^2 x_j^{(i)} \right)^2 \\
& \Leftrightarrow \min_{|w_j| \leq L} \sum_i \left((1 - \sigma_j^2) \text{sign}(x_j^{(i)}) |x_j^{(i)}| - \text{sign}(w_j) \text{sign}(x_j^{(i)}) \max(0, |w_j x_j^{(i)}| - \alpha_j) \right)^2 \\
& \Leftrightarrow \min_{|w_j| \leq L} \sum_i \left((1 - \sigma_j^2) |x_j^{(i)}| - \text{sign}(w_j) \max(0, |w_j x_j^{(i)}| - \alpha_j) \right)^2. \tag{A.22}
\end{aligned}$$

Note that for $\sigma_j^2 = 1$, any $w_j \in \left[-\frac{\alpha_j}{\max^{(i)} |x_j^{(i)}|}, \frac{\alpha_j}{\max_i |x_j^{(i)}|} \right]$ solves the problem and hence the solution is not unique. In all other cases, we have $1 - \sigma_j^2 > 0$, thus, only a positive sign of w_j makes sense and we restrict ourselves to this case. Due to $w_j \geq 0$, we have

$$\left((1 - \sigma_j^2) |x_j^{(i)}| - \max(0, |w_j x_j^{(i)}| - \alpha_j) \right)^2 = \begin{cases} \left((1 - \sigma_j^2) |x_j^{(i)}| \right)^2 & \text{if } |w_j x_j^{(i)}| \leq \alpha_j, \\ \left((1 - \sigma_j^2 - w_j) |x_j^{(i)}| + \alpha_j \right)^2 & \text{if } |w_j x_j^{(i)}| > \alpha_j. \end{cases} \tag{A.23}$$

As the first case is an upper bound of the second case, it is highly desirable to choose w_j large enough such that all training samples are contained in the second case. But we also need to take into account the upper bound $w_j \leq L$. Due to the data set assumption (ii), the minimization problem (A.22) becomes

$$\min_{w_j \in [0, L]} \sum_{i \in I_j(L)} \left((1 - \sigma_j^2 - w_j) |x_j^{(i)}| + \alpha_j \right)^2. \tag{A.24}$$

The minimizer is

$$w_j = \begin{cases} \frac{\alpha_j}{p_j} + 1 - \sigma_j^2 & \text{if } \frac{\alpha_j}{p_j} + 1 - \sigma_j^2 \leq L, \\ L & \text{else,} \end{cases} \tag{A.25}$$

where $p_{L,j} = p_j = \frac{\sum_{i \in I_j(L)} |x_j^{(i)}|^2}{\sum_{i \in I_j(L)} |x_j^{(i)}|}$. Equivalently, we can write $w_j = \min\{\frac{\alpha_j}{p_j} + 1 - \sigma_j^2, L\}$.

Now we use the SVD of \mathcal{A} to solve $\varphi_\theta(\bar{x}) = z$ for $z \in \mathcal{R}(A)$ or

$$\bar{x} - \sum_{j \in \mathbb{N}} \text{sign}(w_j \langle \bar{x}, v_j \rangle) \max(0, |w_j \langle \bar{x}, v_j \rangle| - \alpha_j) v_j = \sum_{j \in \mathbb{N}} \langle z, v_j \rangle v_j, \tag{A.26}$$

respectively, componentwise. For all $j \in \mathbb{N}$ it holds

$$\begin{aligned}
& \langle \bar{x}, v_j \rangle - \text{sign}(w_j \langle \bar{x}, v_j \rangle) \max(0, |w_j \langle \bar{x}, v_j \rangle| - \alpha_j) = \langle z, v_j \rangle \\
& \Leftrightarrow \text{sign}(\langle \bar{x}, v_j \rangle) (|\langle \bar{x}, v_j \rangle| - \max(0, |w_j \langle \bar{x}, v_j \rangle| - \alpha_j)) = \langle z, v_j \rangle. \tag{A.27}
\end{aligned}$$

Thus, the sign of $\langle \bar{x}, v_j \rangle$ is the same as the one of $\langle z, v_j \rangle$. Assuming $w_j |\langle \bar{x}, v_j \rangle| \leq \alpha_j$, we obtain

$$|\langle \bar{x}, v_j \rangle| = |\langle z, v_j \rangle| \quad \text{if } |\langle z, v_j \rangle| \leq \frac{\alpha_j}{w_j} \tag{A.28}$$

and assuming $w_j |\langle \bar{x}, v_j \rangle| > \alpha_j$, we obtain

$$|\langle \bar{x}, v_j \rangle| = \frac{|\langle z, v_j \rangle| - \alpha_j}{1 - w_j} \quad \text{if } \frac{|\langle z, v_j \rangle|}{1 - w_j} - \frac{\alpha_j}{1 - w_j} > \frac{\alpha_j}{w_j}. \tag{A.29}$$

In total, this leads to

$$\langle \bar{x}, v_j \rangle = \begin{cases} \langle z, v_j \rangle & \text{if } |\langle z, v_j \rangle| \leq \frac{\alpha_j}{w_j}, \\ \text{sign}(\langle z, v_j \rangle) \frac{|\langle z, v_j \rangle| - \alpha_j}{1 - w_j} & \text{if } |\langle z, v_j \rangle| > \frac{\alpha_j}{w_j}. \end{cases} \tag{A.30}$$

Using $1 - w_j = \max\{\sigma_j^2 - \frac{\alpha_j}{p_j}, 1 - L\}$, we see that the filter function

$$r_L(\sigma_j^2, s) = \begin{cases} 1 & \text{if } |s| \leq \frac{\alpha_j}{w_j}, \\ \frac{1}{\max\{\sigma_j^2 - \frac{\alpha_j}{p_j}, 1 - L\}} \frac{|s| - \alpha_j}{|s|} & \text{if } |s| > \frac{\alpha_j}{w_j} \end{cases} \quad (\text{A.31})$$

fulfills (3.15). F_L is then derived by exploiting (3.18). \square

A.5 Computation of filter function (4.22)

Assuming continuous extensions $p_L(\sigma^2)$ s.t. $p_L(\sigma_j^2) = p_{L,j}$, $w_L(\sigma^2)$ s.t. $w_L(\sigma_j^2) = w_j$, and $\alpha(\sigma^2)$ s.t. $\alpha(\sigma_j^2) = \alpha_j$, we can write the filter function from Lemma 4.5 as

$$r_L(\sigma^2, s) = \begin{cases} \frac{1}{\max\{\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}, 1 - L\}} \frac{|s| - \alpha(\sigma^2)}{|s|} & \text{if } |s| > \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}, \\ 1 & \text{if } |s| \leq \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}. \end{cases} \quad (\text{A.32})$$

Besides, it holds $w_L(\sigma^2) = \min\left\{\frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - \sigma^2, L\right\} \leq L$, also by Lemma 4.5. Now, we choose $|s| = p_L(\sigma^2)\sigma^2$ and distinguish between three different cases (small, medium and large values of σ^2).

At first, we consider the case $\sigma^2 \leq \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)}$. Then, it holds

$$|s| = p_L(\sigma^2)\sigma^2 \leq \frac{\alpha(\sigma^2)}{L} \leq \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}. \quad (\text{A.33})$$

Thus, it follows

$$r_L(\sigma^2, \pm p_L(\sigma^2)\sigma^2) = 1. \quad (\text{A.34})$$

Secondly, we consider the case $\frac{\alpha(\sigma^2)}{L p_L(\sigma^2)} < \sigma^2 \leq \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L$. We shortly make sure, that the lower bound is in fact smaller than the upper one. The assumptions of Lemma 4.5 imply that $p_L(\sigma^2) > \frac{\alpha(\sigma^2)}{L}$ holds. From this, we can follow

$$\begin{aligned} & p_L(\sigma^2)(1 - L) > \frac{\alpha(\sigma^2)}{L} - \frac{L\alpha(\sigma^2)}{L} \\ \Rightarrow & \alpha(\sigma^2) + p_L(\sigma^2)(1 - L) > \frac{\alpha(\sigma^2)}{L} \\ \Rightarrow & \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L > \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)}. \end{aligned} \quad (\text{A.35})$$

Besides, note that $\sigma^2 \leq \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L$ implies $w_L(\sigma^2) = L$. Thus, it holds

$$|s| = p_L(\sigma^2)\sigma^2 > \frac{\alpha(\sigma^2)}{L} = \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}. \quad (\text{A.36})$$

Accordingly, we obtain

$$\begin{aligned} r_L(\sigma^2, \pm p_L(\sigma^2)\sigma^2) &= \frac{1}{\max\{\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}, 1 - L\}} \frac{p_L(\sigma^2)\sigma^2 - \alpha(\sigma^2)}{p_L(\sigma^2)\sigma^2} \\ &= \frac{1}{1 - L} \frac{p_L(\sigma^2)\sigma^2 - \alpha(\sigma^2)}{p_L(\sigma^2)\sigma^2} = \frac{1}{1 - L} \left(1 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)\sigma^2}\right). \end{aligned} \quad (\text{A.37})$$

At last, we consider the case $\sigma^2 > \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L$. Note that this implies $w_L(\sigma^2) = \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - \sigma^2$. Besides it holds $\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} > 0$, which implies

$$\begin{aligned} & \left(\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} \right) (1 - \sigma^2) > 0 \\ \Rightarrow & \sigma^2 \left(\frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - \sigma^2 \right) > \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} \\ \Rightarrow & |s| = p_L(\sigma^2) \sigma^2 > \frac{\alpha(\sigma^2)}{\frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - \sigma^2} = \frac{\alpha(\sigma^2)}{w_L(\sigma^2)}. \end{aligned} \quad (\text{A.38})$$

Accordingly, we obtain

$$\begin{aligned} r_L \left(\sigma^2, \pm p_L(\sigma^2) \sigma^2 \right) &= \frac{1}{\max\left\{ \sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}, 1 - L \right\}} \frac{p_L(\sigma^2) \sigma^2 - \alpha(\sigma^2)}{p_L(\sigma^2) \sigma^2} \\ &= \frac{1}{\sigma^2 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2)}} \frac{p_L(\sigma^2) \sigma^2 - \alpha(\sigma^2)}{p_L(\sigma^2) \sigma^2} \\ &= \frac{p_L(\sigma^2) \sigma^2 - \alpha(\sigma^2)}{(p_L(\sigma^2) \sigma^2 - \alpha(\sigma^2)) \sigma^2} = \frac{1}{\sigma^2}. \end{aligned} \quad (\text{A.39})$$

Thus, in total, it holds

$$r_L \left(\sigma^2, \pm p_L(\sigma^2) \sigma^2 \right) = \begin{cases} 1 & \text{if } \sigma^2 \leq \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)}, \\ \frac{1}{1-L} \left(1 - \frac{\alpha(\sigma^2)}{p_L(\sigma^2) \sigma^2} \right) & \text{if } \frac{\alpha(\sigma^2)}{L p_L(\sigma^2)} < \sigma^2 \leq \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L, \\ \frac{1}{\sigma^2} & \text{if } \sigma^2 > \frac{\alpha(\sigma^2)}{p_L(\sigma^2)} + 1 - L. \end{cases} \quad (\text{A.40})$$

References

- [1] J. Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.
- [2] G. S. Alberti, E. De Vito, M. Lassas, L. Ratti, and M. Santacesaria. Learning the optimal tikhonov regularizer for inverse problems. *Advances in Neural Information Processing Systems*, 34:25205–25216, 2021.
- [3] C. Arndt. Regularization theory of the analytic deep prior approach. *Inverse Problems*, 38(11):115005, 2022.
- [4] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.
- [5] A. Aspri, Y. Korolev, and O. Scherzer. Data driven regularization by projection. *Inverse Problems*, 36(12):125009, 2020.
- [6] S. Banert, A. Ringh, J. Adler, J. Karlsson, and O. Oktem. Data-driven nonsmooth optimization. *SIAM Journal on Optimization*, 30(1):102–131, 2020.
- [7] H. Bauermeister, M. Burger, and M. Moeller. Learning spectral regularizations for linear inverse problems. In *NeurIPS 2020 Workshop on Deep Learning and Inverse Problems*.
- [8] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pages 573–582. PMLR, 2019.

- [9] M. Benning and M. Burger. Modern regularization methods for inverse problems. *Acta Numerica*, 27:1–111, 2018.
- [10] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. Compressed sensing using generative models. In *International Conference on Machine Learning*, pages 537–546. PMLR, 2017.
- [11] L. Bungert, R. Raab, T. Roith, L. Schwinn, and D. Tenbrinck. Clip: Cheap lipschitz training of neural networks. In *Scale Space and Variational Methods in Computer Vision: 8th International Conference, SSVM 2021, Virtual Event, May 16–20, 2021, Proceedings*, pages 307–319. Springer, 2021.
- [12] J. Chung, M. Chung, and D. P. O’Leary. Designing optimal spectral filters for inverse problems. *SIAM Journal on Scientific Computing*, 33(6):3132–3152, 2011.
- [13] A. Denker, M. Schmidt, J. Leuschner, and P. Maass. Conditional invertible neural networks for medical imaging. *Journal of Imaging*, 7(11):243, 2021.
- [14] S. Dittmer, T. Kluth, P. Maass, and D. Otero Bague. Regularization by Architecture: A Deep Prior Approach for Inverse Problems. *Journal of Mathematical Imaging and Vision*, 62:456–470, 2020.
- [15] A. Ebner, J. Frikel, D. Lorenz, J. Schwab, and M. Haltmeier. Regularization of inverse problems by filtered diagonal frame decomposition. *Applied and Computational Harmonic Analysis*, 62:66–83, 2023.
- [16] A. Ebner and M. Haltmeier. Plug-and-play image reconstruction is a convergent regularization method. *arXiv preprint arXiv:2212.06881*, 2022.
- [17] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- [18] K. Gregor and Y. LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML’10, pages 399–406, Madison, WI, USA, 2010. Omnipress.
- [19] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge. Model-based learning for accelerated, limited-view 3-d photoacoustic tomography. *IEEE transactions on medical imaging*, 37(6):1382–1393, 2018.
- [20] J. He, Y. Wang, and J. Ma. Radon inversion via deep learning. *IEEE transactions on medical imaging*, 39(6):2076–2087, 2020.
- [21] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.
- [22] S. Kabri, A. Auras, D. Riccio, H. Bauermeister, M. Benning, M. Moeller, and M. Burger. Convergent data-driven regularizations for ct reconstruction. *arXiv preprint arXiv:2212.07786*, 2022.
- [23] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
- [24] R. Laumont, V. D. Bortoli, A. Almansa, J. Delon, A. Durmus, and M. Pereyra. Bayesian imaging using plug & play priors: when langevin meets tweedie. *SIAM Journal on Imaging Sciences*, 15(2):701–737, 2022.
- [25] Y. LeCun. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- [26] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. Nett: Solving inverse problems with deep neural networks. *Inverse Problems*, 36(6):065005, 2020.
- [27] A. K. Louis. *Inverse und schlecht gestellte Probleme*. Teubner, 1989.

- [28] S. Lunz, O. Öktem, and C.-B. Schönlieb. Adversarial regularizers in inverse problems. *Advances in neural information processing systems*, 31, 2018.
- [29] P. Mathé and B. Hofmann. How general are general source conditions? *Inverse Problems*, 24(1):015009, 2008.
- [30] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.
- [31] S. Mukherjee, S. Dittmer, Z. Shumaylov, S. Lunz, O. Öktem, and C.-B. Schönlieb. Learned convex regularizers for inverse problems. *arXiv preprint arXiv:2008.02839*, 2020.
- [32] S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra, and C.-B. Schönlieb. Learned reconstruction methods with convergence guarantees: A survey of concepts and applications. *IEEE Signal Processing Magazine*, 40(1):164–182, 2023.
- [33] S. Mukherjee, C.-B. Schönlieb, and M. Burger. Learning convex regularizers satisfying the variational source condition for inverse problems. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021.
- [34] D. Obmann and M. Haltmeier. Convergence analysis of equilibrium methods for inverse problems. *arXiv preprint arXiv:2306.01421*, 2023.
- [35] D. Obmann, L. Nguyen, J. Schwab, and M. Haltmeier. Augmented nett regularization of inverse problems. *Journal of Physics Communications*, 5(10):105002, 2021.
- [36] C. Oh, D. Kim, J.-Y. Chung, Y. Han, and H. Park. Eter-net: End to end mr image reconstruction using recurrent neural network. In *Machine Learning for Medical Image Reconstruction: First International Workshop, MLMIR 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 1*, pages 12–20. Springer, 2018.
- [37] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux. Learning maximally monotone operators for image recovery. *SIAM Journal on Imaging Sciences*, 14(3):1206–1237, 2021.
- [38] A. Rieder. *Keine Probleme mit inversen Problemen: eine Einführung in ihre stabile Lösung*. Springer-Verlag, 2013.
- [39] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin. Plug-and-play methods provably converge with properly trained denoisers. In *International Conference on Machine Learning*, pages 5546–5557. PMLR, 2019.
- [40] T. Schuster, B. Kaltenbacher, B. Hofmann, and K. S. Kazimierski. *Regularization Methods in Banach Spaces*. De Gruyter, Berlin, Boston, 2012.
- [41] J. Schwab, S. Antholzer, and M. Haltmeier. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Problems*, 35(2):025008, 2019.
- [42] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. *International Journal of Computer Vision*, 128:1867–1888, 2020.
- [43] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948. IEEE, 2013.