Data-consistent neural networks for solving nonlinear inverse problems

DOI:
10.3934/ipi.2022037

Document Version
Accepted author manuscript

Manuscript submitted to AIMS Journal

doi:10.3934/xx.xxxxxxx

DATA-CONSISTENT NEURAL NETWORKS FOR SOLVING NONLINEAR INVERSE PROBLEMS

Yoeri Boink

Department of Applied Mathematics
University of Twente
Netherlands

Markus Haltmeier

Department of Mathematics
University of Innsbruck
Austria

Sean Holman

Department of Mathematics
University of Manchester
United Kingdom

Johannes Schwab*

Department of Mathematics
University of Innsbruck
Austria

(Communicated by the associate editor name)

Abstract. Data assisted reconstruction algorithms, incorporating trained neural networks, are a novel paradigm for solving inverse problems. One approach is to first apply a classical reconstruction method and then apply a neural network to improve its solution. Empirical evidence shows that plain two-step methods provide high-quality reconstructions, but they lack a convergence analysis as known for classical regularization methods. In this paper we formalize the use of such two-step approaches in the context of classical regularization theory. We propose data-consistent neural networks that can be combined with classical regularization methods. This yields a data-driven regularization method for which we provide a convergence analysis with respect to noise. Numerical simulations show that compared to standard two-step deep learning methods, our approach provides better stability with respect to out of distribution examples in the test set, while performing similarly on test data drawn from the distribution of the training set. Our method provides a stable solution approach to inverse problems that beneficially combines the known nonlinear forward model with available information on the desired solution manifold in training data.

2020 *Mathematics Subject Classification.* Primary: 65J20, 68T07, 65J22; Secondary: 45F05.

Key words and phrases. Deep learning, data-consistency, nonlinear inverse problems, neural networks, regularization, convergence rates.

*Corresponding author: Johannes Schwab.

1. **Introduction.** In recent years there has been massive interest in new machine learning based approaches to reconstruction in inverse problems. In this paper we consider a class of two-step methods for ill-posed nonlinear inverse problems which combine a known inversion technique with a second step given by a trained deep neural net. Existent works have looked at two-step networks without data-consistency. However our particular view is that the neural net should be designed so that individual reconstructions remain consistent with observed data. This means specifically that the neural net should preserve level sets of the forward operator, at least approximately if the data are noisy, and in this case we call the neural net *data-consistent*. Intuitively, data-consistent networks help us to pick a favourable reconstruction amongst the various possible reconstructions which are all consistent with the observed data based on a known forward model. The former is especially important if the inverse problem is highly underdetermined and the space of possible solutions is 'large'. Traditional regularization methods pursue a similar goal but we believe it is natural to let this choice be informed by a set of training data containing correct reconstructions. Because of this our approach, using data-consistent networks, may be considered a type of learned data-driven regularization for ill-posed problems.

This paper contains both theoretical and numerical results on data-consistent networks for ill-posed nonlinear inverse problems. On the theoretical side we prove convergence, under certain hypotheses, of reconstructions obtained by data-consistent networks to the correct solution as the noise level goes to zero. This proves that data-consistent networks can provide a data-driven regularization method for ill-posed problems in the standard technical sense. On the numerical side we focus on an ill-posed problem which is highly underdetermined, meaning that there are infinitely many solutions and the standard right inverse gives a reconstruction with severe artifacts. In this case classical post-processing methods could invent structure, that is not present in the data. We apply data-consistent networks to the problem of saturation as well as reconstruction from severely under-sampled and saturated Radon transform data. We compare the results of our data-consistent network to reconstructions using U-nets [55, 47] without data-consistency.

Background. We will now introduce the technical framework for our results. Let $(\mathbb{X}, \|\cdot\|)$ and $(\mathbb{Y}, \|\cdot\|)$ be Banach spaces[1] and let $\mathcal{F}\colon \mathbb{D} \subseteq \mathbb{X} \to \mathbb{Y}$ be a continuous, possibly nonlinear, mapping. We study the stable solution of the inverse problem

$$\text{Recover } x \in \mathbb{D} \text{ from } \mathcal{F}(x) = y\,, \tag{1}$$

where $y \in \mathcal{F}(\mathbb{D})$ are exact data. Furthermore, we are especially interested in the noisy data case where data $y^\delta \in \mathbb{Y}$ are given with $\|y - y^\delta\| \leq \delta$. An inversion method for exact data is a right inverse $\mathcal{G}_0\colon \mathcal{F}(\mathbb{D}) \to \mathbb{D}$ for $\mathcal{F}$,

$$\forall y \in \mathcal{F}(\mathbb{D})\colon \quad \mathcal{F}\mathcal{G}_0(y) = y\,. \tag{2}$$

The inversion method $\mathcal{G}_0$ therefore recovers elements in $\mathcal{G}_0\mathcal{F}(\mathbb{D}) = \mathrm{Fix}(\mathcal{G}_0\mathcal{F})$, the set of fixed points of $\mathcal{G}_0\mathcal{F}\colon \mathbb{D} \to \mathbb{D}$. We are mainly interested in the case where (1) is ill-posed, where no continuous right inverse $\mathcal{G}_0$ exists. If noisy data $y^\delta \in \mathbb{Y}$ are given with $\|\mathcal{F}(x) - y^\delta\| \leq \delta$ for $x \in \mathcal{G}_0\mathcal{F}(\mathbb{D})$, then $\mathcal{G}_0(y^\delta)$ is either not well defined or arbitrary far away from x. In this case one has to apply regularization methods to the data, which are stable approximations to $\mathcal{G}_0$.

[1] You can take $\mathbb{X}$ and $\mathbb{Y}$ as finite dimensional spaces $\mathbb{R}^n$ and $\mathbb{R}^q$ with the Euclidian norm if you are not familiar with infinite dimensional Banach spaces.

Well established regularization methods are quadratic Tikhonov regularization [32, 77, 76] and iterative regularization [78, 75]. Both methods are designed to approximate minimum norm solutions $\mathcal{G}_0(y) \in \arg\min\{\|x - x_0\| \mid \mathcal{F}(x) = y\}$ with fixed $x_0 \in \mathbb{X}$. However, for most applications minimum norm solutions are not the desired ones. One way to overcome this issue is by convex variational regularization [34], where one takes

$$\mathcal{G}_\alpha(y^\delta) \in \arg\min\left\{\frac{1}{2}\|\mathcal{F}(x) - y^\delta\|^2 + \alpha\mathcal{J}(x) \mid x \in \mathbb{D}\right\}, \tag{3}$$

which approximate $\mathcal{J}$-minimizing solutions $\mathcal{G}_0(y) \in \arg\min\{\mathcal{J}(x) \mid \mathcal{F}(x) = y\}$. Here the regularization functional $\mathcal{J}\colon \mathbb{X} \to [0, \infty]$ incorporates a-priori information and acts as criterion for selecting particular solutions. There are still several challenges related to variational regularization techniques. For example, computing $\mathcal{J}$-minimizing solutions requires time consuming iterative minimization schemes. Moreover, finding a regularization functional that well models solutions of interest is a difficult issue. Typical choices such as total variation or the ℓ^q-norm with respect to some frame enforce strong handcrafted prior assumptions that are often not met in practical applications.

Related work. Recently, several deep learning methods to solve inverse problems have been proposed [6]. Some approaches apply iterative methods, where operators are replaced by neural networks [8, 49, 17, 15, 9, 16], while others aim for a fully learned reconstruction scheme [13, 5]. Other approaches are inspired by solving the regularized linear or nonlinear inverse problem by means of truncated Neumann series with learned components [10, 12] or by minimizing a variational problem incorporating a learned regularization functional [21, 79, 87]. Another popular approach for imaging problems is to use a neural network as a second step after some initial reconstruction. Several such post processing methods have been considered in the literature [20, 47, 15, 72]. In these two-step approaches, the reconstruction network takes the form $\mathcal{R} = \Phi\mathcal{G}$ where $\mathcal{G}\colon \mathbb{Y} \to \mathbb{X}$ maps the data to the reconstruction space (reconstruction or backprojection layer) and $\Phi\colon \mathbb{X} \to \mathbb{X}$ is a neural network (NN) whose free parameters are adjusted to the training data. In particular, so called residual networks $\Phi = \mathrm{Id}_{\mathbb{X}} + \mathbf{U}$, where only the residual part $\mathbf{U}$ is trained [46, 47, 70, 45] show very accurate results for solving inverse problems.

In order to address the ill-posedness of linear inverse problems, regularizing networks of the form $\mathcal{R}_\alpha = \Phi_\alpha\mathcal{G}_\alpha$ were introduced in [82, 81]. Here $\mathcal{G}_\alpha\colon \mathbb{Y} \to \mathbb{X}$ defines any regularization and $\Phi_\alpha\colon \mathbb{X} \to \mathbb{X}$ are trained neural networks approximating data-consistent networks. In the linear case these networks are named nullspace networks because they only add parts in the kernel of $\mathcal{F}$ as proposed in [74, 81]. A somewhat similar approach was proposed by Bubba et al. [86], combining model-based sparse regularization and deep learning, so that only the part that is not in the information of the data is learned and the remaining part is handled by a controllable regularization method.

In this paper we extend the concept of nullspace networks and derive convergence and convergence rates for data-consistent network families $(\mathcal{R}_\alpha)_{\alpha>0}$ for nonlinear problems and provide some numerical examples. Our approach differs from learned iterations and unrolled schemes in that the application requires only one forward pass of a neural network and a single application of a reconstruction operator. Therefore the proposed approach in principle does not need any iterations with

multiple applications of the forward and adjoint operator and can allow fast application. However, it also addresses the problem of other such two-step procedures that they produce reconstructions that are not consistent with the data. Only by solving this problem is it possible to achieve convergence rates for these methods. Further the approach is very flexible since it is the combination of a neural network with a freely selectable classical reconstruction method.

Regularizing networks for nonlinear problems. Let $\mathcal{G}_0$ be any right inverse of $\mathcal{F}$ and $(\mathcal{G}_\alpha)_{\alpha>0}$ a regularization of $\mathcal{G}_0$, for example classical Tikhonov regularization. Further, let $(\Phi_\alpha)_{\alpha>0}$ be a family of Lipschitz continuous mappings $\Phi_\alpha \colon \mathbb{X} \to \mathbb{X}$ (networks). In this paper we show that under suitable assumptions the reconstruction networks

$$\mathcal{R}_\alpha \coloneqq \Phi_\alpha \mathcal{G}_\alpha \colon \mathbb{Y} \to \mathbb{X}, \tag{4}$$

define a convergent regularization method. Additionally, we derive rates at which the reconstruction error converges to zero as the noise level goes to zero.

A main condition for these results is that $\Phi_\alpha \mathcal{G}_\alpha \mathcal{F}$ converges pointwise to a network of the form $\Phi_0 \mathcal{G}_0 \mathcal{F}$ where $\Phi_0 \colon \mathbb{X} \to \mathbb{X}$ is data-consistent in the sense that $\mathcal{F}\Phi_0 z = \mathcal{F} z$ for $z \in \mathcal{G}_0\mathcal{F}(\mathbb{D})$. The latter property implies that Φ_0 preserves data-consistency of $\mathcal{G}_0$, meaning that if $\mathcal{G}_0(y)$ is a solution of (1), then $\Phi_0\mathcal{G}_0(y)$ is a solution of (1) too. Hence the goal of the learned mapping Φ_0 is to modify solutions of the inverse problem, while keeping data-consistency of the initial solution. We prove that the family of reconstruction networks $(\Phi_\alpha \mathcal{G}_\alpha)_{\alpha>0}$ is a convergent regularization method, and we provide convergence rates.

The benefits of $(\Phi_\alpha \mathcal{G}_\alpha)_{\alpha>0}$ over $(\mathcal{G}_\alpha)_{\alpha>0}$ are at least twofold. First, in the limit $\alpha \to 0$, the network $\Phi_0 \mathcal{G}_0 \mathcal{F}$ selects solutions in $\Phi_0\mathcal{G}_0\mathcal{F}(\mathbb{D})$ that can be trained to better reflect the desired image class than $\mathcal{G}_0\mathcal{F}(\mathbb{D})$. Second, for $\alpha > 0$, the networks Φ_α can be trained to partially undo the smoothing effect of $\mathcal{G}_\alpha\mathcal{F}$ and thereby allow for obtaining convergence rates for less regular elements than the original regularization method $(\mathcal{G}_\alpha)_{\alpha>0}$. The operator $\Phi_0\mathcal{G}_0$ can be seen as a right inverse that is learned from a suitable class of training data.

Outline. The paper is organized as follows. In section 2 we describe the regularization of nonlinear inverse problems and we define the proposed two-step data-driven regularization method. We introduce data-consistent networks, which allow the definition of *regularizing networks*, a regularization method which approximates a data-driven right inverse. We investigate under which assumptions these networks generate a convergent regularization method and we give examples of how such regularizing networks can be constructed. In section 3 we present a convergence analysis and derive convergence rates for the proposed method. Section 4 presents the mathematical description of the inverse problems considered in the numerical simulations, after which the implementation is explained in detail. Results for the simulations are shown in section 5. Additional simulation results can be found in the appendices. The paper concludes with a short summary of the established theory and the numerical simulations.

Glossary. To make the article easier to read, we provide a table that gives an overview of the nomenclature of terms and their meaning.

Furthermore it is our convention to write x as the variable in the domain of the forward mapping, i.e. $x \in \mathbb{D}$, and to write z as the variable after some right inverse has been applied, i.e. $z \in \mathcal{G}_0\mathcal{F}(\mathbb{D})$. For discretised functions and matrices, we will

use bold capital letters instead of the caligraphic letters. Neural networks are an exception to this; we denote both general functions between infinite-dimensional spaces (which we also call neural networks) and the corresponding parametrized functions on discrete spaces that are actually implemented on a computer with Φ and $\mathbf{U}$.

To increase readability, we write a composition of mappings, e.g. $\mathcal{F}$ and $\mathcal{G}_0$, as $\mathcal{G}_0\mathcal{F}(x)$ instead of $\mathcal{G}_0 \circ \mathcal{F}(x)$ or $\mathcal{G}_0(\mathcal{F}(x))$ throughout the paper.

Nomenclature	Description	Definition
$\mathbb{X}$	reconstruction space	Banach space with norm $\|\cdot\|$
$\mathbb{Y}$	data space	Banach space with norm $\|\cdot\|$
$\mathcal{F}$	forward operator	$\mathcal{F}\colon \mathbb{D} \subseteq \mathbb{X} \to \mathbb{Y}$
$\mathbb{D}$	domain of $\mathcal{F}$	$\mathbb{D} \subseteq \mathbb{X}$
δ	noise level	$\delta \geq 0$
$\mathcal{G}_0$	right inverse of $\mathcal{F}$	$\mathcal{G}_0\colon \mathcal{F}(\mathbb{D}) \to \mathbb{D}$
$\mathbb{X}_0$	reconstruction space with source condition	$\mathbb{X}_0 \subseteq \mathcal{G}_0\mathcal{F}(\mathbb{D})$
$\mathcal{J}$	regularization functional	$\mathcal{J}\colon \mathbb{X} \to [0,\infty]$
α	regularization parameter	$\alpha \in (0,\infty)$
α^*	parameter choice function	$\alpha^*\colon (0,\infty) \times \mathbb{Y} \to (0,\infty)$
Φ_0	data-consistent neural network	$\Phi_0\colon \mathbb{X} \to \mathbb{X}$ s.t. $\mathcal{F}\circ\Phi_0 = \mathcal{F}$
$(\mathcal{G}_\alpha)_{\alpha>0}$	regularization of $\mathcal{G}_0$	$\mathcal{G}_\alpha\colon \mathbb{Y} \to \mathbb{X}$
$(\Phi_\alpha)_{\alpha>0}$	family of Lipschitz mappings (networks)	$\Phi_\alpha\colon \mathbb{X} \to \mathbb{X}$
$(\mathcal{R}_\alpha)_{\alpha>0}$	regularizing networks	$\mathcal{R}_\alpha := \Phi_\alpha\mathcal{G}_\alpha$
$\mathcal{G}, \mathcal{G}^\delta, \mathcal{R}^\delta$	reconstruction algorithm	$\mathcal{G}, \mathcal{G}^\delta, \mathcal{R}^\delta\colon \mathbb{Y} \to \mathbb{X}$
$\mathbf{U}$	generic neural network	$\mathbf{U}\colon \mathbf{X} \to \mathbf{X}$
$\mathcal{P}$	projection operator	$\mathcal{P}\colon \mathbb{X} \to \mathbb{X}$
$\mathbf{P}$	discretized projection operator	$\mathbf{P}\colon \mathbf{X} \to \mathbf{X}$
$\mathbf{F}$	discretized saturation mapping	$\mathbf{F}\colon \mathbf{X} \to \mathbf{X}$
$\mathbf{F}_2$	discretized saturation in sinogram domain	$\mathbf{F}_2\colon \mathbf{Y} \to \mathbf{Y}$
$\mathbf{F}_1$	discretized Radon transform	$\mathbf{F}_1\colon \mathbf{X} \to \mathbf{Y}$
$\mathbf{F}_1^\dagger$	pseudo inverse of discr. Radon transform	$\mathbf{F}_1^\dagger\colon \mathbf{Y} \to \mathbf{X}$
$\mathbf{P}_C$	metric projection onto a convex set C	$\mathbf{P}_C\colon \mathbf{X} \to C$

2. Convergence of regularized data-consistent networks.

Throughout the rest of this paper, let $\mathcal{F}\colon \mathbb{D} \subseteq \mathbb{X} \to \mathbb{Y}$ be a continuous mapping between Banach spaces $\mathbb{X}$ and $\mathbb{Y}$. We study the stable solution of the inverse problem (1). In this section we introduce the regularizing networks and present the convergence analysis.

2.1. Regularization of inverse problems.

Let $\mathcal{G}_0\colon \mathcal{F}(\mathbb{D}) \to \mathbb{D}$ be a right inverse for $\mathcal{F}$. If (1) is ill-posed and y^δ are noisy data with $\|\mathcal{F}(x) - y^\delta\| \leq \delta$, then the reconstruction method $\mathcal{G}_0(y^\delta)$ is unstable, meaning arbitrarily far away from $\mathcal{G}_0\mathcal{F}(x)$ or even not defined. To obtain meaningful approximations of $\mathcal{G}_0\mathcal{F}(x)$, one has to apply regularization methods defined as follows.

Definition 2.1 (Regularization method)**.** Let $(\mathcal{G}_\alpha)_{\alpha>0}$ be a family of continuous mappings $\mathcal{G}_\alpha\colon \mathbb{Y} \to \mathbb{X}$. If for all $x \in \mathcal{G}_0\mathcal{F}(\mathbb{D})$ there exists a parameter choice function

$\alpha^*\colon (0,\infty)\times\mathbb{Y}\to(0,\infty)$ such that

$$0=\lim_{\delta\to 0}\sup\left\{\alpha^*(\delta,y^\delta)\mid y^\delta\in B_\delta(\mathcal{F}(x))\right\}$$
$$0=\lim_{\delta\to 0}\sup\left\{\|x-\mathcal{G}_{\alpha^*(\delta,y^\delta)}(y^\delta)\|\mid y^\delta\in B_\delta(\mathcal{F}(x))\right\},$$

where $B_\delta(\mathcal{F}(x))$ is the ball with radius δ around $\mathcal{F}(x)$. We call $((\mathcal{G}_\alpha)_{\alpha>0},\alpha^*)$ a regularization of $\mathcal{G}_0$. Moreover we call α^* an admissible parameter choice and $(\mathcal{G}_\alpha)_{\alpha>0}$ a regularization method for $\mathcal{G}_0$ at x.

Probably the best known regularization is quadratic Tikhonov regularization in Hilbert spaces [32]. Under the assumption that $\mathcal{F}$ is weakly sequentially closed, one shows that there exist solutions of (1) with minimal distance to a given point $x_0\in\mathbb{X}$ and that

$$\mathcal{T}_{\alpha,y^\delta}(x):=\frac{1}{2}\|\mathcal{F}(x)-y^\delta\|^2+\frac{\alpha}{2}\|x-x_0\|^2\quad\text{for } x\in\mathbb{D}\tag{5}$$

has at least one minimizer. We can define $\mathcal{G}_\alpha(y^\delta)$ as any minimizer of $\mathcal{T}_{\alpha,y^\delta}$. If the solution of (1) with minimal distance to x_0 is unique and denoted by $\mathcal{G}_0(y)$, then $((\mathcal{G}_\alpha)_{\alpha>0},\alpha^*)$ with a parameter choice satisfying $\delta^2/\alpha^*\to 0$ and $\alpha^*\to 0$ as $\delta\to 0$ is a regularization method for $\mathcal{G}_0$ [32, 77]. Research indicates that solutions with minimal distance to a fixed initial guess $x_0\in\mathbb{X}$ are too simple in many applications. The use of non-quadratic penalties has demonstrated to often give better results. Recently, deep learning methods have shown outstanding performance. Here solutions are defined by a neural network that maps the given data to a desired solution.

2.2. **Data-consistent networks.** The first ingredient for constructing regularizing two-step networks are data-consistent networks.

Definition 2.2 (Data-consistent network)**.** We call $\Phi_0\colon\mathbb{X}\to\mathbb{X}$ a data-consistent network if Φ_0 is Lipschitz continuous and $\forall z\in\mathcal{G}_0\mathcal{F}(\mathbb{D})\colon\mathcal{F}\Phi_0(z)=\mathcal{F}(z)$.

In data-consistent networks, if $z\in\mathcal{G}_0\mathcal{F}(\mathbb{D})$ is a solution of (1), then $\Phi_0(z)$ is solution of (1) too. In particular, $\Phi_0\mathcal{G}_0$ is a right inverse for $\mathcal{F}$ with solution set $\Phi_0\mathcal{G}_0\mathcal{F}(\mathbb{D})=\mathrm{Fix}(\Phi_0\mathcal{G}_0\mathcal{F})$. Data-consistent networks can be constructed by

$$\Phi_0(z)=\mathcal{P}_{z,0}\mathbf{U}(z),\tag{6}$$

where $\mathbf{U}\colon\mathbb{X}\to\mathbb{X}$ is a Lipschitz continuous trained neural network, and $\mathcal{P}_{z,0}\colon\mathbb{X}\to\mathbb{X}$ a Lipschitz continuous mapping with $\mathcal{P}_{z,0}(x)\in\mathcal{F}^{-1}(\{\mathcal{F}(z)\})=\{x\in\mathbb{D}\mid\mathcal{F}(x)=\mathcal{F}(z)\}$. The mapping $\mathcal{P}_{z,0}$ can be seen as a generalized projection on the preimage $\mathcal{F}^{-1}(\{\mathcal{F}(z)\})$. In the special case where $\mathcal{F}$ is a linear mapping, $\Phi_0(z)$ can be chosen as $\Phi_0(z)=z+\mathcal{P}_{\ker(\mathcal{F})}(\mathbf{U}(z)-z)$, where $\mathcal{P}_{\ker}$ is the projection on the kernel of $\mathcal{F}$ [81]. A visual comparison of data-consistent networks and standard post-processing networks is shown in figure 1.

The key ingredient of defining data-consistent networks is the computation of the generalized projection operator $\mathcal{P}_{z,0}$, which guarantees that the output of the network is projected onto a data-consistent solution. In practice for a given nonlinear operator $\mathcal{F}$ often there is no closed form expression of $\mathcal{P}_{z,0}$ and therefore it has to be approximated. For given data $y=\mathcal{F}(z)$ this can be done for example by using any iterative minimization algorithm initialized at $\mathbf{U}(z)$ (see figure 2) to minimize

$$\mathcal{P}_{z,0}\approx\arg\min_x\|\mathcal{F}(x)-\mathcal{F}(z)\|^2.\tag{7}$$

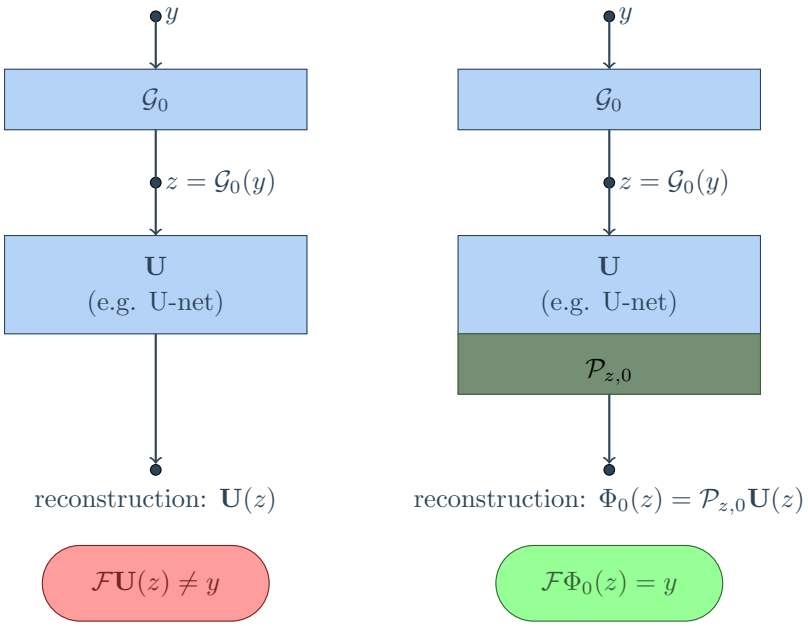


FIGURE 1. On the left, a standard post-processing network. The red box illustrates that the output in general does not reproduce the data under the forward operation $\mathcal{F}$. On the right a data-consistent network architecture. The green box illustrates that it does reproduce the data under the forward operation $\mathcal{F}$.

The minimization of this non-linear optimization can for example be computed by a Newton type algorithm with a fixed number of iterations. Of course this would not yield a completely data consistent solution, but nevertheless the data-discrepancy of the solution would be improved.

For ill-posed problems we cannot expect there to be a continuous inverse, and so we now introduce the concept of regularizing networks, which also account for the instability of the right inverse.

Definition 2.3 (Regularizing networks)**.** We call $(\mathcal{R}_\alpha \colon \mathbb{Y} \to \mathbb{X})_{\alpha>0}$ defined by $\mathcal{R}_\alpha := \Phi_\alpha \mathcal{G}_\alpha$ a family of regularizing networks if the following hold:
(R1) $(\mathcal{G}_\alpha \colon \mathbb{Y} \to \mathbb{X})_{\alpha>0}$ is a regularization of $\mathcal{G}_0$.
(R2) $(\Phi_\alpha \colon \mathbb{X} \to \mathbb{X})_{\alpha>0}$ are uniformly L-Lipschitz continuous mappings.
(R3) For some data-consistent network $\Phi_0 \colon \mathbb{X} \to \mathbb{X}$ we have

$$\forall x \in \mathbb{D} \colon \lim_{\alpha \to 0} \Phi_\alpha \mathcal{G}_\alpha \mathcal{F}(x) = \Phi_0 \mathcal{G}_0 \mathcal{F}(x)\,. \tag{8}$$

Regularizing networks are thus made up of a classical regularization method $\mathcal{G}_\alpha$ and a subsequently applied neural network Φ_α. The neural network can also be adapted to the noise level as the subscript α indicates. The network aims to improve the initial reconstruction in a way such that the difference between the input and the processed image in the data domain (after application of $\mathcal{F}$) is controlled. For high noise and larger α, more difference in data domain is tolerated, whereas for zero noise the processed reconstruction should produce exactly the same data. In

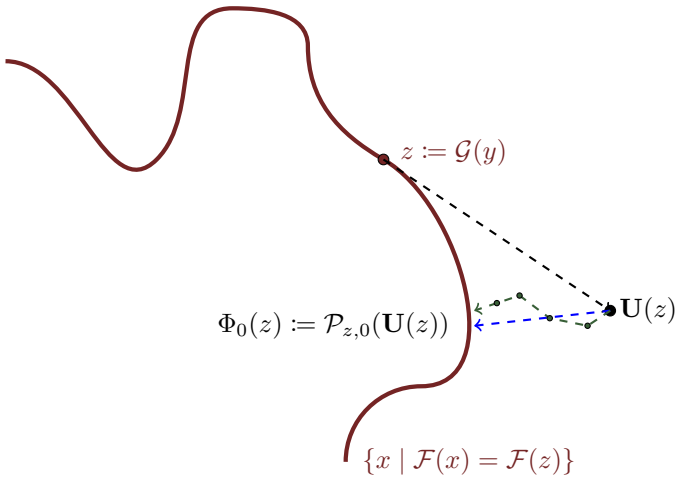


FIGURE 2. The figure illustrates the solution set for given data y and the output of the network $\mathbf{U}$ acting on the solution after applying a right inverse of $\mathcal{F}$. The generalized projection operator approximated by some iterative scheme or closed form operator is shown in green and blue respectively.

practice an important issue is to actually design networks that converge to a data-consistent limiting network as the noise level goes to zero. Next we give examples for a possible strategy to train such networks.

Example 2.4. Let $x_1, \dots, x_N \in \mathbb{X}$ be training signals, and $y_i = \mathcal{F}(x_i)$ and y_i^δ be the corresponding exact and noisy data, respectively. Further define the vectors of reconstructions $z := (z_1, \dots, z_N)$ and $z^\alpha := (z_1^\alpha, \dots, z_N^\alpha)$ where $z_i = \mathcal{G}_0(y_i)$ and $z_i^\alpha = \mathcal{G}_\alpha(y_i^\delta)$. The weights of the neural network are denoted by $\theta \in \Theta$. We write $(\Phi^\theta)_{\theta\in\Theta}$ for a neural network with given architecture, whose weights have not yet been fixed.

- One possible simple approach is to take the networks $\Phi_\alpha := \Phi_0$ for all $\alpha > 0$, where Φ_0 is the network obtained by minimizing the functional $\sum_i \|\Phi^\theta(z_i) - x_i\|^2 + r(\theta)$, where r is some regularizer for the weights that may be used to ensure a small Lipschitz constant. This approach is motivated by the bound for the Lipschitz constant $L(\Phi^\theta)$

$$L(\Phi^\theta) \le \prod_{i=1}^{l} \sqrt{\lambda_{\max}(\Theta_i^T \Theta_i)} \le \prod_{i=1}^{l} \|\Theta_i\|_F \le \left(\frac{1}{l} \sum_{i=1}^{l} \|\Theta_i\|_F \right)^l ,$$

  where Θ_i are the weight matrices of the ith layer, $\lambda_{\max}$ denotes the largest eigenvalue and $\|\cdot\|_F$ denotes the Frobenius norm. Other approaches to control the Lipschitz constant were proposed penalizing the gradient of the network [80, 79] for projecting the matrices back to the closest matrix with feasible Lipschitz constant after each gradient step [2]. Clearly, since Φ_0 is Lipschitz continuous and $(\mathcal{G}_\alpha)_{\alpha>0}$ is a regularization method, we have the desired limit in (2.3) for all $x \in \mathbb{D}$. If the data-consistency is incorporated in the network architecture, then condition 2.3 is satisfied.

- A more sophisticated approach is to choose the sequence of networks depending on the regularization parameter α. Here the networks Φ_α can be obtained by minimizing $\sum_i \|\Phi^\theta(z_i^\alpha) - x_i\|^2 + r(\theta)$. To enforce the data-consistency of the limiting network Φ_0 one could either choose the network architecture to be data-consistent, meaning $\mathcal{F}(\Phi^\theta(x)) = \mathcal{F}(x)$ for all $\theta \in \Theta$ and $x \in \mathbb{D}$. Alternatively, one may take the networks increasingly data-consistent, $\Phi_\alpha(z) = \mathcal{P}_{z,\alpha}\mathbf{U}_\alpha(z)$, where $\mathbf{U}_\alpha \colon \mathbb{X} \to \mathbb{X}$ are trained networks and $\mathcal{P}_{z,\alpha}$ is a Lipschitz continuous mapping with

$$\mathrm{Im}(\mathcal{P}_{z,\alpha}) \subseteq E_{\alpha,z} \coloneqq \{x \mid \|\mathcal{F}(x) - \mathcal{F}(z)\| \leq r(\alpha)\}$$

  and $\lim_{\alpha\to 0} r(\alpha) = 0$. Data-consistency is obtained in the limit. One example for $\mathcal{P}_{z,\alpha}$ is the metric projection on $E_{\alpha,z}$ which is a Lipschitz continuous mapping if $E_{\alpha,z}$ is convex. Note that there are no restrictions on the particular choice of the architecture of the networks $\mathbf{U}_\alpha$.
- Another network architecture guaranteeing data-consistency is given by

$$\Phi_\alpha(z) = \Phi^{\mathrm{dec}}\left(\mathbf{S} + \alpha\mathbf{T}\right)\Phi^{\mathrm{enc}}(z)\,. \tag{9}$$

  Here Φ^{enc} and Φ^{dec} denote an encoder and decoder network respectively, $\mathbf{S}$ denotes an α-independent network such that $\Phi^{\mathrm{dec}}\mathbf{S}\Phi^{\mathrm{enc}}$ is data-consistent and $\mathbf{T}$ denotes a network that is allowed to depend on α.

2.3. **Convergence analysis.** We have the following convergence result for regularizing networks according to Definition 2.3.

Theorem 2.5 (Regularizing networks)**.** *Any family of regularizing networks* $(\mathcal{R}_\alpha = \Phi_\alpha\mathcal{G}_\alpha)_{\alpha>0}$ *(see Definition 2.3) is a regularization for* $\Phi_0\mathcal{G}_0$ *in the sense of Definition 2.1.*

Proof. Let $x \in \Phi_0\mathcal{G}_0\mathcal{F}(\mathbb{D})$, $y^\delta \in \mathbb{Y}$ with $\|\mathcal{F}(x) - y^\delta\| \leq \delta$ and set $x_\alpha^\delta \coloneqq \Phi_\alpha\mathcal{G}_\alpha(y^\delta)$. Then

$$\begin{aligned}\|x - x_\alpha^\delta\| &= \|\Phi_0\mathcal{G}_0\mathcal{F}(x) - \Phi_\alpha\mathcal{G}_\alpha y^\delta\| \\ &\leq \|\Phi_0\mathcal{G}_0\mathcal{F}(x) - \Phi_\alpha\mathcal{G}_\alpha\mathcal{F}(x)\| + \|\Phi_\alpha\mathcal{G}_\alpha\mathcal{F}(x) - \Phi_\alpha\mathcal{G}_\alpha y^\delta\| \\ &\leq \|\Phi_\alpha\mathcal{G}_\alpha\mathcal{F}(x) - \Phi_0\mathcal{G}_0\mathcal{F}(x)\| + L\|\mathcal{G}_\alpha\mathcal{F}(x) - \mathcal{G}_\alpha y^\delta\|\,.\end{aligned}$$

Now if $\alpha^*(\delta, y^\delta)$ is an admissible parameter choice for $(\mathcal{G}_\alpha)_{\alpha>0}$, then

$$\begin{aligned}\sup\{\|x - \mathcal{R}_{\alpha^*(\delta,y^\delta)}(y^\delta)\| \mid y^\delta \in B_\delta(\mathcal{F}(x))\} &\leq \|\Phi_{\alpha^*(\delta,y^\delta)}\mathcal{G}_{\alpha^*(\delta,y^\delta)}\mathcal{F}(x) - \Phi_0\mathcal{G}_0\mathcal{F}(x)\| \\ &\quad + L\sup\{\|\mathcal{G}_{\alpha^*(\delta,y^\delta)}\mathcal{F}(x) - \mathcal{G}_{\alpha^*(\delta,y^\delta)}y^\delta\| \mid y^\delta \in B_\delta(\mathcal{F}(x))\}\,.\end{aligned} \tag{10}$$

According to 2.3 in Definition 2.3, the first term converges to zero. Because of

$$\|\mathcal{G}_{\alpha^*(\delta,y^\delta)}\mathcal{F}(x) - \mathcal{G}_{\alpha^*(\delta,y^\delta)}y^\delta\| \leq \|\mathcal{G}_{\alpha^*(\delta,y^\delta)}\mathcal{F}(x) - \mathcal{G}_0\mathcal{F}(x)\| + \|\mathcal{G}_0\mathcal{F}(x) - \mathcal{G}_{\alpha^*(\delta,y^\delta)}y^\delta\|$$

and the fact that $(\mathcal{G}_\alpha)_{\alpha>0}$ is a regularization method for $\mathcal{G}_0$ with parameter choice rule $\alpha = \alpha^*(\delta, y^\delta)$, the second term converges to zero as $\delta \to 0$. □

3. **Convergence rates.** Another important issue for regularization methods is the rate of approximation. This means specifically that there exists a decreasing function $f \colon (0,\infty) \to (0,\infty)$ such that $\lim_{\delta\to 0} f(\delta) = 0$ and $\|\mathcal{R}_{\alpha^*(\delta,y^\delta)}(y^\delta) - x\| \leq f(\delta)$ uniformly for all $y^\delta \in \mathbb{Y}$ with $\|\mathcal{F}(x) - y^\delta\| \leq \delta$.

3.1. **Reconstruction algorithms and convergence rates.** In the following we call $\mathcal{G}\colon \mathbb{Y} \to \mathbb{X}$ a reconstruction algorithm.

Definition 3.1 (Reconstruction error of an algorithm)**.** Let $\mathbb{X}_0 \subseteq \mathbb{X}$, $\delta > 0$ and $\mathcal{G}\colon \mathbb{Y} \to \mathbb{X}$ be a reconstruction algorithm. We call

$$\mathcal{E}(\mathcal{G}, \delta, \mathbb{X}_0) = \sup\{\|x - \mathcal{G}(y^\delta)\| \mid x \in \mathbb{X}_0 \wedge y^\delta \in \overline{B_\delta(\mathcal{F}(x))}\} \tag{11}$$

the reconstruction error of $\mathcal{G}$ over $\mathbb{X}_0$.

Definition 3.2 (Convergence rate of an algorithm)**.** Let $\mathbb{X}_0 \subseteq \mathbb{X}$, $r \in (0, 1]$ and for any $\delta > 0$, let $\mathcal{G}^\delta$ be a reconstruction algorithm. We say that $(\mathcal{G}^\delta)_{\delta>0}$ converges at rate δ^r over $\mathbb{X}_0$ if $\mathcal{E}(\mathcal{G}^\delta, \delta, \mathbb{X}_0) = \mathcal{O}(\delta^r)$ as $\delta \to 0$.

The concept of convergence rates in particular applies for reconstruction algorithms defined by regularization methods. In general, no convergence rate over $\mathcal{G}_0\mathcal{F}(\mathbb{D})$ is possible; they require restricting to proper subsets $\mathbb{X}_0 \subsetneq \mathcal{G}_0\mathcal{F}(\mathbb{D})$ [32, Proposition 3.11]. Source conditions define suitable sets $\mathbb{X}_0$ for classical Tikhonov regularization and related methods based on minimal norm solutions. We investigate the source conditions (transformed source sets) and convergence rates for regularizing networks where $(\mathcal{G}_\alpha)_{\alpha>0}$ is given by Tikhonov regularization in Example 3.5.

3.2. **Rates for the regularizing networks.** Our aim is to prove a convergence rate for $\mathcal{R}_{\alpha^*(\delta,y^\delta)}$ assuming a convergence rate for $\mathcal{G}_{\alpha^*(\delta,y^\delta)}$. Let $(\Phi_\alpha\mathcal{G}_\alpha)_{\alpha>0}$ be a regularizing network and α^* a parameter choice function. For any $\delta > 0$ we define the reconstruction algorithms $\mathcal{G}^\delta, \mathcal{R}^\delta\colon \mathbb{Y} \to \mathbb{X}$ and $\Phi^\delta\colon \mathbb{X} \to \mathbb{X}$ by

$$\begin{aligned}
\mathcal{G}^\delta(y) &\coloneqq \mathcal{G}_{\alpha^*(\delta,y)}(y),\\
\Phi^{\delta,y}(x) &\coloneqq \Phi_{\alpha^*(\delta,y)}(x),\\
\mathcal{R}^\delta(y) &\coloneqq \Phi_{\alpha^*(\delta,y)}\mathcal{G}_{\alpha^*(\delta,y)}(y),
\end{aligned}$$

for $x \in \mathbb{X}$ and $y \in \mathbb{Y}$. We will derive convergence rates for regularizing networks under the following assumptions.

Assumption 3.3 (Convergence rate conditions)**.** Let $\mathbb{X}_0 \subseteq \mathcal{G}_0\mathcal{F}(\mathbb{D})$ satisfy the following for some $r \in (0, 1]$

(N1) $\mathcal{E}(\mathcal{G}^\delta, \delta, \mathbb{X}_0) = \mathcal{O}(\delta^r)$ as $\delta \to 0$.

(N2) $\sup\{\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(z)\| \mid z \in \mathbb{X}_0 \wedge y^\delta \in \overline{B_\delta(\mathcal{F}(z))}\} = \mathcal{O}(\delta^r)$.

(N3) $\sup\{\|\Phi^{\delta,\mathcal{F}(z)}(z) - \Phi_0(z)\| \mid z \in \mathbb{X}_0\} = \mathcal{O}(\delta^r)$.

The first condition 3.3 means that $(\mathcal{G}_{\alpha^*(\delta,y^\delta)})_{\delta>0}$ converges at rate δ^r. Condition 3.3 is a stability estimate for $\mathcal{G}_{\alpha^*(\delta,y^\delta)}$. Condition 3.3 is uniform approximation assumption between $\Phi^{\delta,\mathcal{F}(\cdot)}$ and Φ_0 on $\mathbb{X}_0$.

Theorem 3.4 (Convergence rate for regularizing networks)**.** *Let $\mathbb{M}_0 = \Phi_0(\mathbb{X}_0)$. Under Assumption 3.3 we have $\mathcal{E}(\mathcal{R}^\delta, \delta, \mathbb{M}_0) = \mathcal{O}(\delta^r)$.*

Proof. Let $x \in \mathbb{M}_0$, $\|\mathcal{F}(x) - y^\delta\| \le \delta$, and $z \in \mathbb{X}_0$ s.t. $\Phi_0(z) = x$. Then

$$\begin{aligned}
\|\mathcal{R}^\delta(y^\delta) - x\| &\le \|\mathcal{R}^\delta(y^\delta) - \mathcal{R}^\delta\mathcal{F}(x)\| + \|\mathcal{R}^\delta\mathcal{F}(x) - x\|\\
&\le L\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(x)\| + \|\Phi_0\mathcal{G}^\delta\mathcal{F}(z) - \Phi_0(z)\|\\
&\qquad + \|\mathcal{R}^\delta\mathcal{F}(x) - \Phi_0\mathcal{G}^\delta\mathcal{F}(z)\|\\
&\le L\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(x)\| + L\|\mathcal{G}^\delta\mathcal{F}(z) - z\|
\end{aligned}$$

$$
\begin{aligned}
&\quad + \|\Phi^{\delta,\mathcal{F}(z)}\mathcal{G}^\delta\mathcal{F}(x) - \Phi_0\mathcal{G}^\delta\mathcal{F}(x)\| \\
&\le L\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(x)\| + L\|\mathcal{G}^\delta\mathcal{F}(z) - z\| \\
&\quad + \|\Phi^{\delta,\mathcal{F}(z)}\mathcal{G}^\delta\mathcal{F}(x) - \Phi^{\delta,\mathcal{F}(x)}\mathcal{G}_0\mathcal{F}(x)\| \\
&\quad + \|\Phi^{\delta,\mathcal{F}(z)}\mathcal{G}_0\mathcal{F}(x) - \Phi_0\mathcal{G}_0\mathcal{F}(x)\| \\
&\quad + \|\Phi_0\mathcal{G}_0\mathcal{F}(x) - \Phi_0\mathcal{G}^\delta\mathcal{F}(x)\| \\
&\le L\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(x)\| + L\|\mathcal{G}^\delta\mathcal{F}(z) - z\| \\
&\quad + \|\Phi^{\delta,\mathcal{F}(z)}\mathcal{G}_0\mathcal{F}(x) - \Phi_0\mathcal{G}_0\mathcal{F}(x)\| + 2L\|\mathcal{G}^\delta\mathcal{F}(x) - \mathcal{G}_0\mathcal{F}(x)\|.
\end{aligned}
$$

Each of the above terms are $\mathcal{O}(\delta^r)$: the first term due to the stability estimate 3.3, the second term due to 3.3, and the third term due to 3.3. For the fourth term we use that $\mathcal{G}_0\mathcal{F}(x) = \mathcal{G}_0\mathcal{F}(z) = \mathcal{G}_0\mathcal{F}\mathcal{G}_0\mathcal{F}(w) = \mathcal{G}_0\mathcal{F}(w) = z \in \mathbb{X}_0$ for some $w \in \mathbb{D}$. This implies that the fourth term is $\mathcal{O}(\delta^r)$, due to 3.3 again. □

In the following we give an explicit example of a classical regularization method combined with a sequence of regularizing networks, where Assumption 3.3 is satisfied and therefore Theorem 3.4 can be applied.

Example 3.5 (Regularizing networks combined with Tikhonov regularization)**.** Given a Gâteaux differentiable forward operator $\mathcal{F}$ we consider $(\mathcal{G}_\alpha)_{\alpha>0}$ defined by classical Tikhonov regularization (5), a data-consistent network $\Phi_0\colon \mathbb{X} \to \mathbb{X}$ and a sequence of regularizing networks $(\Phi_\alpha\colon \mathbb{X} \to \mathbb{X})_{\alpha>0}$ satisfying 3.3 for $r = 1/2$.

Corollary 1. *If we consider the set $\mathbb{M}_0 := \Phi_0(\mathbb{X}_0)$ where $\mathbb{X}_0$ is the source set of classical Tikhonov regularization and we assume that the networks $\Phi^{\delta,\mathcal{F}(.)}$ converge uniformly to Φ_0 on $\mathbb{X}_0$ at rate $\mathcal{O}(\delta^{1/2})$ as $\delta \to 0$, then $\mathcal{E}(\mathcal{R}^\delta, \delta, \mathbb{M}_0) = \mathcal{O}(\delta^{1/2})$*

Proof. The convergence rate condition 3.3 holds according to [32]. Since for $x = \Phi_0(z) \in \mathbb{M}_0$ we have $\mathcal{F}(x) = \mathcal{F}(z)$ (Definition 2.2) and because of the stability estimate for Tikhonov regularization [32] we have

$$
\begin{aligned}
&\sup\{\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(x)\| \mid x \in \mathbb{M}_0 \wedge y^\delta \in \overline{B_\delta(\mathcal{F}(x))}\} \\
&\qquad = \sup\{\|\mathcal{G}^\delta(y^\delta) - \mathcal{G}^\delta\mathcal{F}(z)\| \mid z \in \mathbb{X}_0 \wedge y^\delta \in \overline{B_\delta(\mathcal{F}(x))}\} = \mathcal{O}(\delta^{1/2}), \quad (12)
\end{aligned}
$$

which shows 3.3. Finally 3.3 holds by assumption and therefore the conditions of Theorem 3.4 are satisfied for $r = 1/2$. □

This shows one of the main benefits of the concept of regularizing networks, namely transforming the smoothness class $\mathbb{X}_0$ on which the basic regularization converges at a certain rate, to a different data dependent set $\Phi_0(\mathbb{X}_0)$ with possibly less regularity, while preserving the convergence rate.

4. Examples and numerical set-up.

In this section we provide a mathematical description of two general classes of nonlinear inverse problems (projection onto convex sets in section 4.1 and compositions of certain mappings in section 4.3) to which we can apply data-consistent networks, as well as specific examples of these problems that will be used for our simulation experiments. The specific examples consist of the recovery of saturated multivariate Gaussians and image reconstruction from saturated and sparsely sampled Radon-data. In each case, we define a nonlinear forward mapping $\mathcal{F}$ and derive a right inverse $\mathcal{G}_0$ and data-consistent network Φ_0. We also describe the network architecture for our neural networks in section 4.5.

Although the mathematical theory was worked out on infinite-dimensional spaces, the discontinuity of the right inverse and the non-uniqueness of the solution are serious problems even after discretization of the problem, which become noticeable through unstable numerical methods without the use of regularization methods [83].

4.1. **Projection onto a convex set.** For the first class of nonlinear inverse problems, we suppose the forward map $\mathcal{F} := \mathcal{P}_C \colon \mathbb{D} \to C$ is a metric projection on a closed convex set $C \subseteq \mathbb{D}$, i.e.

$$y = \mathcal{P}_C(x) := \operatorname*{arg\,min}_{\tilde{x} \in C} \{\|\tilde{x} - x\|\} \,. \tag{13}$$

The affine normal cone to C at x is defined as

$$\mathcal{N}_C(x) := \{\tilde{x} \in \mathbb{D} \mid \mathcal{P}_C(\tilde{x}) = x\} \,.$$

It is easily shown that any mapping that maps $x \in C$ to an element in the normal cone $\mathcal{N}_C(x)$ is a right inverse of $\mathcal{P}_C$. In particular, the projection $\mathcal{P}_{\mathcal{N}_C(x)} \colon C \to \mathbb{D}$,

$$\mathcal{P}_{\mathcal{N}_C(x)}(\hat{x}) := \operatorname*{arg\,min}_{\tilde{x} \in \mathcal{N}_C(x) \cap \mathbb{D}} \{\|\tilde{x} - \hat{x}\|\} \,, \tag{14}$$

defines a right inverse $\mathcal{G}_0$ of $\mathcal{P}_C$, since $\forall x \in C \colon \mathcal{P}_C \mathcal{P}_{\mathcal{N}_C(x)}(x) = x$. According to Definition 2.2, a data-consistent network Φ_0 satisfies that $\forall z \in \mathcal{G}_0 \mathcal{F}(\mathbb{D}) \colon \mathcal{F}\Phi_0(z) = \mathcal{F}(z)$. We define

$$\Phi_0(z) := \mathcal{P}_{\mathcal{N}_C(\mathcal{P}_C(z))}(\mathbf{U}(z)), \tag{15}$$

so this requirement is satisfied. Here $\mathbf{U} \colon \mathbb{X} \to \mathbb{X}$ is any Lipschitz continuous trained neural network (c.f. definition 2.2). See Figure 3 for a visual illustration.

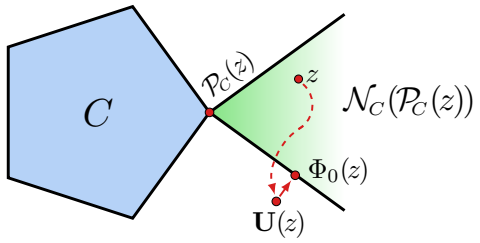


FIGURE 3. Two-dimensional visualization of the data-consistent network for the 'projection on convex set' problem explained in Section 4.1. The blue region indicates the convex set C, the green region (that extends infinitely to the right) indicates the affine normal cone to C at $\mathcal{P}_C(z)$. The output of the data-consistent network is indicated by $\Phi_0(z)$; it can be seen that it is obtained by taking the input z, applying a Lipschitz continuous neural network $\mathbf{U}(z)$ and projecting it to the normal cone $\mathcal{N}_C$ at $\mathcal{P}_C(z)$. This ensures that $\mathcal{P}_C(\Phi_0(z)) = \mathcal{P}_C(z)$, as required by the definition of a data-consistent network.

As an example of the 'projection on convex set' problem, we consider the inverse problem of recovering nonhomogeneously saturated signals in sections 4.2 and 4.4. In applications this can occur whenever the measurement device has a limited range, where the saturated signal can be either an image [3] or a any other measurement as for example in acoustics [4]. Saturation is also a problem in CT reconstruction,

where metal objects result in high-intensity artefacts in backprojection reconstructions [85] emerging from saturated signals in the measurement domain. Although our specific inverse problem does not treat the problem of metal artefact reduction explicitly, our numerical examples are inspired by real problems in medical imaging. In section 4.2 we consider the the saturation problem on images of multivariate Gaussians as a toy problem to investigate the advantages and disadvantages of the proposed data-consistent methods. Although we test our method here only on very simple images, the method is equally applicable to arbitrary images. In Section 4.3 we then study the saturation of Radon data combined with the inverse problem of undersampled computerized tomography.

4.2. **Spatially dependent saturation of multivariate Gaussians.** Next we mathematically describe the inverse problem of signal saturation which is an example of projection onto a convex set as described in section 4.1. Formally we define the domain $\mathbb{D} = \mathbb{X} := \ell^2(\Omega)$ and we define the saturation mapping as a projection on a convex set, as described in section 4.1. This means $\mathcal{F}(x) := \mathcal{P}_C(x)$, where

$$C := \left\{ x \in \ell^2(\Omega) \mid \forall r \in \Omega \colon x(r) \leq M(r) \right\}$$

where $M(r) \geq 0$ is the saturation value at location r. The operator $\mathcal{F}$ thus takes a signal x and cuts it off everywhere where it is greater than a given level M. Note that this level M does not have to be constant but can vary over the signal. The inverse problem now consists in recovering the original data from the corrupted signal. The corresponding right inverse for $y \in C$ is defined as $\mathcal{G}_0(y) = \mathcal{P}_{\mathcal{N}_C(y)}(y) = \mathrm{Id}(y)$. The projection defined by (13) is explicitly given by

$$\big[\mathcal{P}_C(x)\big](r) := \min\left\{x(r), M(r)\right\}.$$

Since $\mathcal{N}_C\mathcal{P}_C(z) = \{x \mid \mathcal{P}_C(x) = \mathcal{P}_C(z)\}$, (15) can be written pointwise as

$$\big[\Phi_0(z)\big](r) = \begin{cases} z(r) & \text{for } z(r) < M(r), \\ \max\left\{\big[\mathbf{U}(z)\big](r), M(r)\right\} & \text{for } z(r) \geq M(r). \end{cases} \tag{16}$$

We consider the square domain $\Omega := [-1, 1] \times [-1, 1]$. The spatially dependent saturation function is defined

$$M(r) := \begin{cases} 0.6 & \text{if } \|r\| \leq \frac{1}{2}, \\ 0 & \text{if } \|r\| > \frac{1}{2}. \end{cases} \tag{17}$$

Each image in the training or test set contains one centered multivariate Gaussian with diagonal covariance matrix, having standard deviations (σ_1, σ_2) independently randomly chosen in the interval $[0.24, 0.32]$. All images in the training and test set are scaled to obtain maximum values randomly chosen in the interval $[0.75, 1]$. Opposed to standard networks, one of the benefits of using a data-consistent network is that it is more robust to distribution shifts, when evaluating on test data. For this reason, a modified test set has been created, where the Gaussians have standard deviations in the interval $[0.12, 0.20]$ with maximum intensities in the interval $[0.6, 0.8]$. For the numerical implementation we consider the discretized domain $\bar{\Omega} := \mathbb{R}^{128\times128}$ as discretization of Ω.

The data-consistent network $\Phi_0(z)$, as described in (16), is compared with the neural network $\mathbf{U}(z)$ without data-consistency. We compare reconstruction quality for both the regular test set and modified test set. A description of the neural network architecture and training details are provided in section 4.5.

4.3. **Composition of mappings.** For the second general class of inverse problems, we consider forward mappings $\mathcal{F}\colon \mathbb{D} \to \mathbb{Y}$ that are themselves defined as a composition of two (possibly nonlinear) mappings:

$$\mathcal{F}(x) := \mathcal{F}_2(\mathcal{F}_1(x)), \qquad \text{where } \mathcal{F}_1\colon \mathbb{D} \to \mathbb{E} \text{ and } \mathcal{F}_2\colon \mathbb{E} \to \mathbb{Y},$$

where $\mathbb{E}$ is a Banach space.In section 4.4 we will consider such problems in which $\mathcal{F}_1$ is the Radon transform and $\mathcal{F}_2$ is a saturation operator as described in section 4.2. Furthermore we impose the restriction that the mapping $\mathcal{F}_2$ provides a data-consistent network that can be written as a projection

$$\Phi_0^{(2)}(z) := \mathcal{P}_{S\cap \mathrm{Im}(\mathcal{F}_1)}(\mathbf{U}_2(z)),$$

where $S \subseteq \mathbb{E}$. In particular, this is true for the mapping described in Section 4.1, when the projection on a normal cone also maps into the range of the operator $\mathcal{F}_1$. The projection onto the intersection in the data-consistent network can be implemented by an alternating projection algorithm, as explained in section 4.4. If we assume that $S \cap \mathrm{Im}(\mathcal{F}_1) \neq \emptyset$, then we define a data-consistent network Φ_0 for the full mapping as

$$\Phi_0(z) = \Phi_0^{(1)}\mathcal{G}_0^{(1)}\Phi_0^{(2)}\mathcal{G}_0^{(2)}\mathcal{F}_2\mathcal{F}_1(z) \tag{18}$$

where $\Phi_0^{(i)}$ and $\mathcal{G}_0^{(i)}$ are defined as the data-consistent network for $\mathcal{F}_i$ and the right inverse for $\mathcal{F}_i$ respectively. We check the data-consistent property (Definition 2.2) by

$$\begin{aligned}
\mathcal{F}\Phi_0(z) &= \mathcal{F}_2\mathcal{F}_1\Phi_0^{(1)}\mathcal{G}_0^{(1)}\Phi_0^{(2)}\mathcal{G}_0^{(2)}\mathcal{F}_2\mathcal{F}_1(z)\\
&= \mathcal{F}_2\mathcal{F}_1\mathcal{G}_0^{(1)}\Phi_0^{(2)}\mathcal{G}_0^{(2)}\mathcal{F}_2\mathcal{F}_1(z)\\
&= \mathcal{F}_2\Phi_0^{(2)}\mathcal{G}_0^{(2)}\mathcal{F}_2\mathcal{F}_1(z)\\
&= \mathcal{F}_2\mathcal{G}_0^{(2)}\mathcal{F}_2\mathcal{F}_1(z)\\
&= \mathcal{F}_2\mathcal{F}_1(z)\\
&= \mathcal{F}(z),
\end{aligned}$$

where we used in order: the data-consistent property of $\Phi_0^{(1)}$; the definition of a right-inverse $\mathcal{G}_0^{(1)}$ in combination with the projection on the range of $\mathcal{F}_1$; the data-consistent property of $\Phi_0^{(2)}$; and the definition of the right-inverse $\mathcal{G}_0^{(2)}$.

We note that this is not the only data-consistent network possible for such an inverse problem: one could also design a network that only makes use of either $\Phi_0^{(1)}$ or $\Phi_0^{(2)}$. However, (18) provides a network that is intuitively clear: an initial solution z is obtained by a classical regularization method, after which first a better 'guess' is made by applying a neural network on $\mathbb{E}$, followed by a neural network that makes a better guess on the reconstruction space $\mathbb{D}$, while keeping the solutions data-consistent throughout.

4.4. **Saturation of Radon transformed human chest images.** As an example of the 'composition of mapping' problem explained in section 4.3, we consider the inverse problem of reconstructing images of the human chest from saturated Radon measurements that are heavily undersampled in the angular variable. We consider the composition of two mappings as described in section 4.3, where $\mathcal{F}_1$ is a linear mapping that acts as the Radon transform and $\mathcal{F}_2$ is a nonlinear saturation mapping

as described in 4.2, that saturates the Radon signals at a constant value M. More precisely the Radon transform for an image x is defined by

$$\mathcal{F}_1(x)(\theta, s) = \int_{\mathbb{R}} x(s\theta + t\theta^{\perp})\,\mathrm{d}t,$$

for $(\theta, s) \in \mathbb{S}^1 \times \mathbb{R}$. Here θ in the unit sphere defines the projection direction and s the signed distance to the origin of the line over which x is integrated.

Next we define our discretized setup for the numerical studies. For this from now on, the associated discrete operators and also the corresponding domains and codomains will be denoted with bold capital letters. In our particular sparse sampling problem we assume we only have access to measurements at finitely many points in Ω, defined by $\Omega \coloneqq \{(\theta_k, s_l) \mid k = 1, \dots, n_a \text{ and } l = 1, \dots, n_s\}$. We choose a very small number of equidistantly sampled projection directions $n_a = 8$ and a typical number of lines $n_s = 288$ per direction, which is 3/2 times the number of pixels $n_x = 192$ per row and column of the discrete image. The domains and in the discretized setup are then given by the finite dimensional spaces $\mathbf{D} := \mathbf{X} = \mathbb{R}^{n_x \times n_x}$, and $\mathbf{Y} = \mathbb{R}^{n_a \times \frac{3}{2} n_x}$.

We now define all elements that are needed to obtain the data-consistent network (18). For the first mapping $\mathbf{F}_1$ we consider a discretization of the undersampled Radon transform, mapping an image to the projections along lines in a very low number of equidistantly distributed directions denoted n_a. Since the measurements contain only very few projection directions, this inverse problem is highly underdetermined and therefore has no unique solution. Using a basis ansatz the discrete angle Radon transform can be represented by a matrix $\mathbf{F}_1 \in \mathbb{R}^{n_a \cdot \frac{3}{2} n_x \times n_x^2}$, which is obtained as described in [11].As a right inverse we take the pseudo-inverse of $\mathbf{F}_1$, i.e. $\mathbf{G}_0^{(1)} := \mathbf{F}_1^{\dagger}$. Since the mapping is linear, the corresponding data-consistent network is a null-space network [81], i.e. $\mathbf{\Phi}_0^{(1)}(z) = z + \mathbf{P}_{\ker(\mathbf{F}_1)}\mathbf{U}(z)$, where $\mathbf{P}_{\ker(\mathbf{F}_1)} = \mathrm{Id} - \mathbf{F}_1^{\dagger}\mathbf{F}_1$. The second mapping $\mathbf{F}_2$ is a nonlinear saturation mapping $\mathbf{F}_2 = \mathbf{P}_C$, as described in 4.2, where the set Ω corresponds to the set of lines for which a value of the Radon transform is measured and $M(r) = 8$ for all $r \in \Omega$ is a constant clipping value. Its right inverse $\mathbf{G}_0^{(2)}$ and the data-consistent network $\mathbf{\Phi}_0^{(2)}$ are chosen as described in section 4.2, this time with constant saturation level $M = 8$. Finally, for this particular choice of $\mathbf{F}_2$, the projection on the intersection of convex sets reads $\mathbf{P}_{\mathcal{N}_C(\mathbf{P}_C(z)) \cap \mathrm{Im}(\mathbf{F}_1)}$, which can be achieved by the 'projection onto convex sets' (POCS) algorithm [1]: by alternatingly performing $\mathbf{P}_{\mathcal{N}_C(\mathbf{P}_C(z))}$ and $\mathbf{P}_{\mathrm{Im}(\mathbf{F}_1)} = \mathbf{F}_1\mathbf{F}_1^{\dagger}$, the resulting iteration converges linearly to a point on the intersection.

Training and test images were obtained from the LoDoPaB-CT dataset [14], which on its turn makes use of the LIDC/IDRI dataset [7]. In our work, we only make use of the high quality CT reconstructions in the LoDoPaB-CT dataset that we use as 'ground truth' for our setup. The images are scaled to 192×192 pixels, after which the mappings $\mathbf{F}_1$ and $\mathbf{F}_2$ are applied to obtain simulated sinograms. After that, pseudo-inverses $\mathbf{G}_0^{(2)}$ and $\mathbf{G}_0^{(1)}$ are applied to obtain the input for our data-consistent network. For this simulation experiment we have also created a modified test set to investigate how the trained networks generalize towards slightly modified data. For conciseness, the procedure to get from the regular test data to the modified test data is not explained in full detail. In short, the test set consists of images in the range of $\mathbf{F}_1^{\dagger}$ that produce sinograms that have a maximum below

or around the saturation level. This means that the saturation mapping $\mathbf{F}_2$ will not have a big effect on the unsaturated sinograms. Images in the modified test set look very similar to the ones in the regular test set, but they often show a small gradient at locations where the regular images show a piecewise constant structure. Some samples from the modified test set are shown in Figure 7 and appendix B.2.

The data-consistent network $\mathbf{\Phi}_0(z)$, as described in (18), is compared with two other networks: the first one applies a single neural network to the pseudo-inverse reconstruction; the second one first applies a neural network in the sinogram domain, then applies the pseudo-inverse of the Radon-transform, followed by a neural network in the image domain. Here the neural network in sinogram domain takes the Radon-measurements as an image, whose size depends on the measurement geometry. For completeness, we summarize the three networks below:

- One neural network: $\mathbf{N}_1(z) = \mathbf{U}_1(z)$.
- Two neural networks: $\mathbf{N}_2(z) = \mathbf{U}_1\mathbf{G}_0^{(1)}\mathbf{U}_2\mathbf{F}_1(z)$.
- Data-consistent network: $\mathbf{\Phi}_0(z) = \mathbf{\Phi}_0^{(1)}\mathbf{G}_0^{(1)}\mathbf{\Phi}_0^{(2)}\mathbf{G}_0^{(2)}\mathbf{F}_2\mathbf{F}_1(z)$.

We emphasize that the data-consistent networks, in terms of architecture, make use of the same neural networks $\mathbf{U}_1$ and $\mathbf{U}_2$ as the first two networks, i.e. $\mathbf{\Phi}_0^{(1)}(z) = z + \mathbf{P}_{\ker(\mathbf{F}_1)}\mathbf{U}_1(z)$ and $\mathbf{\Phi}_0^{(2)}$ makes use of $\mathbf{U}_2$ as defined in (16). A more detailed description of the neural network architecture and training details are provided in section 4.5.

Ideally, the data-consistent network is trained 'end-to-end', meaning that both $\mathbf{U}_1$ which is used in $\mathbf{\Phi}_0^{(1)}$, and $\mathbf{U}_2$ which is used in $\mathbf{\Phi}_0^{(2)}$, are trained at the same time. However, the application of the POCS algorithm is computationally intensive, since it requires iterative application of the mappings $\mathbf{F}_1$ and $\mathbf{F}_1^\dagger$. For this reason, we have chosen to first train $\mathbf{\Phi}_0^{(2)}$ to output the unsaturated sinogram, then perform the POCS algorithm and finally train $\mathbf{\Phi}_0^{(1)}$ to output the reconstructed image.

4.5. **Neural network architecture and training details.** In this work, the popular U-Net [55] is implemented as a neural network. By combining max-pooling and upsampling with residual connections, U-Net has shown to be effective in restoring or improving images that possess both small- and large-scale artefacts [47]. By using standard nonlinearities such as rectified linear units (ReLUs) and convolutions, the network is Lipschitz continuous, which is a requirement as described in Definition 2.2. The Lipschitz constant can be controlled by weight regularization, such as adding an L^2-loss on the weights in the loss function. We stress that instead of U-Net, any other Lipschitz continuous neural network can be chosen.

The U-Net was implemented as described in [47], although for each experiment some parameters were chosen slightly different to obtain optimal results. For all experiments, the network has a 'depth' of four, meaning four times max-pooling and upsampling. The U-Nets in the image domain perform the regular max-pooling and upsampling in two directions, while the U-Net in the sinogram domain performs these only in one direction, leaving the number of angles constant at 8. This is done because the neighboring angles in the sinogram show very little resemblance to each other and there are only 8. The 'width', or the amount of convolutions at every depth is chosen to be two. As in [47] the number of convolution channels doubles after each max-pooling; the number of channels at the start is stated in Table 1, since this was chosen differently for every simulation experiment. In all experiments a residual structure that is also apparent in [47] is used. The U-Net uses

	Exp. 1 ($\mathbf{U}/\mathbf{\Phi}_0$): image domain	Exp. 2 ($\mathbf{U}_1/\mathbf{\Phi}_0^{(1)}$): image domain	Exp. 2 ($\mathbf{U}_2/\mathbf{\Phi}_0^{(2)}$): sinogram domain
#training samples	1024	35584	35584
#validation samples	256	3522	3522
#test samples	1024	3553	3553
depth	4	4	4
width	2	2	2
#channels in top layer	8	16	16
convolution size	3×3	3×3	3×3
nonlinearity	ReLU	ReLU	ReLU
start learning rate	10^{-3}	10^{-3}	10^{-3}
final learning rate	10^{-4}	$2 \cdot 10^{-4}$	$2 \cdot 10^{-4}$
batch size	64	32	32
#epochs	1000	25	25

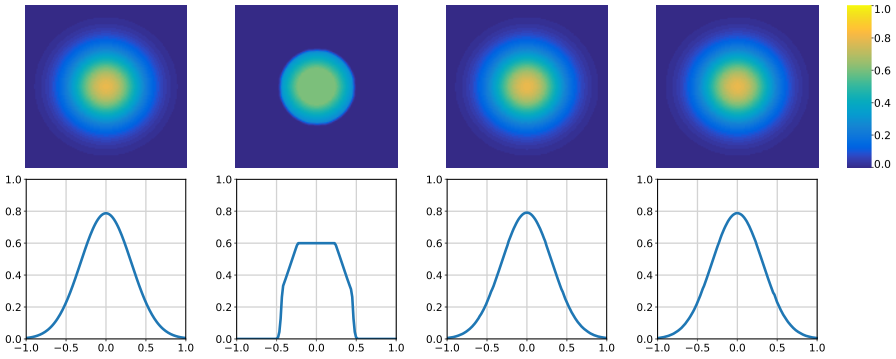
TABLE 1. U-Net parameter details for all simulation experiments.

3×3 convolutions with biases and applied a ReLU-activation after each convolution, except the last one. In all experiments, an L^2-loss function on the difference between output and ground truth is minimized, i.e. $\min_\theta \sum_i^N \|\mathbf{U}^\theta(z_i) - x_i\|_{\ell^2}^2$ for the U-Net $\mathbf{U}^\theta$, or $\min_\theta \sum_i^N \|\mathbf{\Phi}_0^\theta(z_i) - x_i\|_{\ell^2}^2$ for the data-consistent network $\mathbf{\Phi}_0^\theta$, both parametrized by the weights $\theta \in \Theta$. For optimization, the ADAM optimizer with exponentially decaying learning rate is chosen. The learning rates and batch sizes are stated in Table 1. Training was performed in TensorFlow, executed on a single GTX 1080 TI GPU.

5. **Numerical results.** In this section, the reconstruction quality of data-consistent networks is compared with the U-Nets that are not data-consistent. Since both, the proposed methods investigated in this work and deep learning post-processing with the U-net are two step approaches we decided to compare the proposed constrained and established unconstrained networks in the numerical studies. Besides visual comparison, the quality will be compared on the reconstructed images by means of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) but also in terms of data consistency, by measuring the discrepancy to the given data. For both experiments this will be done for the regular test set as well as the modified test set in order to investigate the capacity of the networks to deal with data outside of the training distribution.

5.1. **Spatially dependent saturation of multivariate Gaussians.** Multivariate Gaussians are saturated with a spatially dependent saturation function (17), as described in section 4.2. Results for one sample from the regular test set are shown in Figure 4. Here it can be seen that both U-Net and the data-consistent network provide a very accurate reconstruction. This is also reflected in the PSNR and SSIM values shown in Table 2. The pseudo-inverse reconstruction, which in

this case is just the measurement, is not a good one, since a lot of information is lost by applying the saturation mapping.



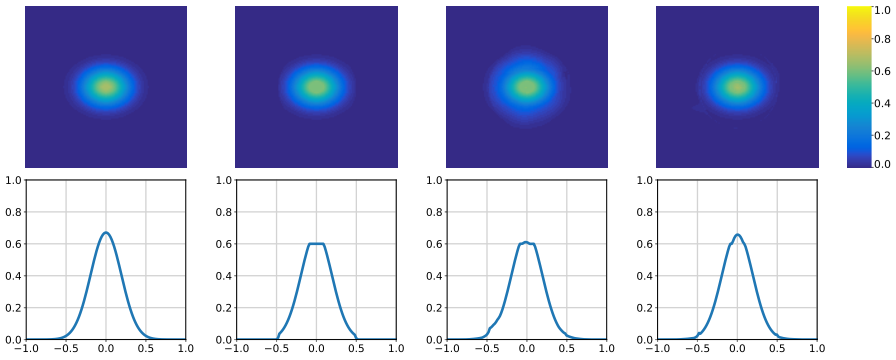
(A) Ground truth (B) Saturated (C) U-Net (D) Data-consistent

FIGURE 4. Reconstructions of a sample from the regular test set. In the bottom the horizontal central slice is shown. Both U-Net and data-consistent network provide an almost perfect reconstruction.

In Figure 5 the results for one sample are shown for the modified test set, which contains smaller Gaussians with a slightly lower intensity. Both U-Net and the data-consistent network are not perfectly able to fill in the missing information in the small Gaussians. This can be expected, since Gaussians of this size were not included in the training set. However, the data-consistent network does not deform the Gaussian at the location where it is not saturated, while U-Net does this slightly; for instance around -0.5 in the slice plot. This behaviour is also reflected in the PSNR and SSIM values in Table 2. Interestingly, the pseudo-inverse behaves very well if we just look at the values in the table, because the saturation mapping did not destroy a lot of the information in the Gaussian. Visual results of three more samples in the modified test set are shown in appendix A. It can be seen that U-Net tends to widen the Gaussians, since it was trained on Gaussians in the training set that were wider. Although the modified test set shows a very specific modification, it illustrates that a data-consistent network is beneficial over using an arbitrary neural network: by making use of the information that we have from the mapping $\mathbf{F}$, we obtain generalization capacity with respect to out of distribution examples.

	PSNR			SSIM		
	Pseudo-inverse	U-Net	Data-consistent	Pseudo-inverse	U-Net	Data-consistent
Regular set	24.2 ± 2.2	60.6 ± 2.1	66.7 ± 1.6	0.56 ± 0.08	1.00 ± 0.00	1.00 ± 0.00
Modified set	48.0 ± 7.8	36.9 ± 2.9	48.0 ± 4.4	0.99 ± 0.01	0.92 ± 0.03	0.97 ± 0.01

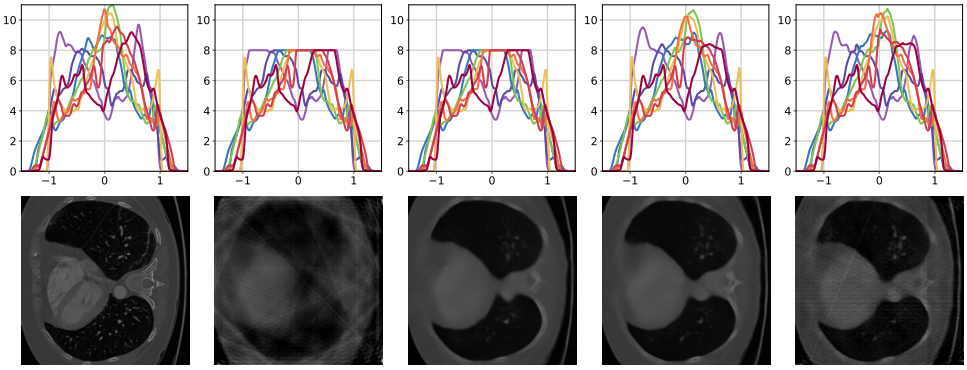
TABLE 2. Comparison of PSNR and SSIM for all reconstruction methods.



(A) Ground truth (B) Saturated (C) U-Net (D) Data-consistent

FIGURE 5. Reconstructions of a sample from the modified test set. In the bottom the horizontal central slice is shown. Data-consistency makes sure that intensity is only changed above the saturation level.

5.2. **Saturation of Radon transformed human chest images.** For one selected sample in the regular test set and one in the modified test set, all reconstructions are shown in Figures 6 and 7. These specific samples were selected because their PSNR values for the U-Nets and the data-consistent network show a similar relation to each other as the average PSNR values of the whole test set (c.f. Table 3).

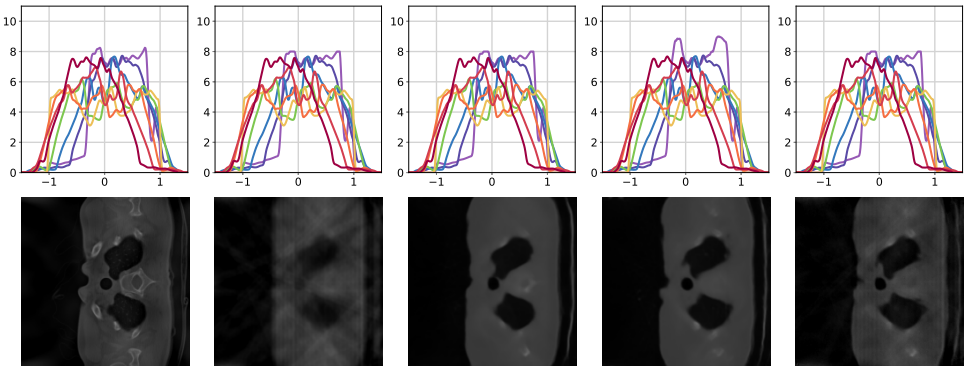


(A) Ground truth (B) Pseudo-inverse PSNR = 22.80 (C) One U-Net PSNR = 27.31 (D) Two U-Nets PSNR = 27.70 (E) Data-consistent PSNR = 27.01

FIGURE 6. Reconstructions of a typical sample from the regular test set. Top: reconstructed sinograms with all 8 angles in different colors. Bottom: reconstructed images (grayscale from 0 to 1).

In the top of Figure 6, the inputs of the right inverse $\mathbf{G}_0^{(1)}$ are shown. This corresponds to the saturated sinograms in case no or only one neural network is trained, and this corresponds to the output of the neural network in the sinogram

domain in case two neural networks are trained. The sinogram signals are plotted in a different color for each angle. The data-consistent network does not change the values of the sinogram that are below the saturation level ($M = 8$). Interestingly, the U-Net in the sinogram domain has learned not to do this as well to a large extent on the regular test set, only some values just below the saturation level are changed (for instance the purple line around -0.5), but values much lower are not changed at all. In the bottom of Figure 6, all reconstructions for this sample are shown. It can be seen that although PSNR values are similar, there is a clear visual difference: the data-consistent network shows more streak artefacts that are typical for discrete-angle Radon reconstructions, while the standard U-Nets provide over-smoothed reconstructions. In other words there is a trade off between preserving small structures and removing streak artifacts. Whereas the data-consistency prevents the neural network from smoothing out all image details, the unconstrained networks are superior in removing undersampling artifacts. However, since there are no image details that are visible in one of the reconstructions and not in the others, it can be concluded that all networks have a similar reconstruction quality with respect to the used measures in image space. Moreover, since only 8 angles were used in the Radon transform, only larger structures can be reconstructed. Some extreme samples, for which the PSNR value of the data-consistent solution is high, similar or low compared to the U-Net solutions, are shown in Appendix B.1. However we want to emphasize that the main intention of our method is improving image quality, but at the same time providing a solution that is compatible with the measured data. We have the opinion that the quality of the reconstruction should not only be measured visually but also in terms of how compatible it is to the given data. This is also reflected in Table 3, where the quality measures in the reconstruction space are comparable, whereas the data-fidelity of the data-consistent network is superior to the other methods on both test sets.



(A) Ground truth

(B) Pseudo-inverse PSNR = 30.02

(C) One U-Net PSNR = 28.29

(D) Two U-Nets PSNR = 29.08

(E) Data-consistent PSNR = 30.83

FIGURE 7. Reconstructions of a typical sample from the modified test set. Top: reconstructed sinograms with all 8 angles in different colors. Bottom: reconstructed images (grayscale from 0 to 1).

In the top of Figure 7, again all sinogram reconstructions are plotted, now for the modified test set. As the modified sinograms contain many values just below

or around the saturation level ($M = 8$), this is challenging for the regular U-Net. Indeed it can be seen that the data-consistent network does not change the saturated sinogram much, while the U-Net on the sinogram increases the purple line well beyond the saturation level. In the bottom of Figure 7, it can be seen that the ground truth possesses some 'smooth' regions, especially in the background on the left and right of the two dark inclusions. While both U-Nets create piecewise constant reconstructions that completely remove this gradient, the data-consistent network keeps these smooth regions: it generalizes better to test images that are not found in the training set by making use of the information in the operator. More extreme samples, where the PSNR value of the data-consistent solution is high, similar or low compared to the U-Net solutions, are shown in Appendix B.2. The same effect on smooth regions can be seen in these visualizations. In Table 3, it can be seen that for all networks the quality drops when the regular test set is replaced by the modified test set; however, this drop is only very small for the data-consistent network and is much bigger for the regular U-Nets. Note that the pseudo-inverse gives an increased PSNR for the modified test set, because the modified images were constructed to lie in the range of $\mathbf{F}$.

	PSNR			
	Pseudo-inverse	One U-Net	Two U-Nets	Data-consistent
Regular set	23.1 ± 2.3	30.5 ± 1.5	31.0 ± 1.5	30.1 ± 1.9
Modified set	29.1 ± 1.6	27.5 ± 1.7	28.3 ± 1.4	29.9 ± 1.2
	SSIM			
	Pseudo-inverse	One U-Net	Two U-Nets	Data-consistent
Regular set	0.50 ± 0.07	0.82 ± 0.04	0.83 ± 0.04	0.74 ± 0.07
Modified set	0.71 ± 0.07	0.74 ± 0.05	0.73 ± 0.05	0.75 ± 0.05
	Data-fidelity			
	Pseudo-inverse	One U-Net	Two U-Nets	Data-consistent
Regular set	6.1 ± 3.3	4.8 ± 1.5	3.9 ± 1.1	0.9 ± 0.4
Modified set	0.4 ± 0.2	11.9 ± 5.4	8.5 ± 2.8	0.6 ± 0.2

TABLE 3. Comparison of PSNR, SSIM and data-fidelity $\|\mathbf{F}(\tilde{x}) - \mathbf{F}(x)\|_{\ell^2}$ for all reconstruction methods, where $\tilde{x}$ is the reconstruction and x is the ground truth. Note that the data-fidelity of the pseudo-inverse in theory should be zero since it is a right inverse of the forward operator. The non-zero values result from numerical instabilities in the computation of this operator

Finally in Table 3 we check the data-fidelity of the solutions from all networks by computing $\|\mathbf{F}(\tilde{x}) - \mathbf{F}(x)\|_{\ell^2}$, where $\tilde{x}$ is the solution of the respective reconstruction method and x is the ground truth. Ideally, the data-fidelity should be zero for the pseudo-inverse and the data-consistent network. It can be seen that indeed

the data-fidelity is much lower for these solutions than for the U-Nets, although not completely zero. This is most likely due to numerical issues (especially for the pseudo-inverse of the regular set) and due to the fact that we saved the images in a 16-bit format between consecutive steps of the data-consistent network to not overload working memory.

6. **Conclusion.** In this paper we introduced data-consistent networks for nonlinear inverse problems. We presented a convergent regularization method by combining deep neural networks that converge to a data-consistent network with classical regularization methods. With the proposed data-driven regularization methods we are able to preserve convergence rates of classical methods over a transformed source set, which is adapted to some data set. This yields improved reconstructions for elements close to the training set, but at the same time data-consistent networks make use of the information from the forward mapping $\mathbf{F}$, which restricts the support of the output distribution to a physically plausible one and therefore increases generalization capacity with respect to examples not coming from the training distribution. This is particularly useful when the physics process is understood, but exact knowledge on real data is not available or when it is not possible to create a training set that reflects the true underlying data distribution. We showed that on a test set similar to the training set, our approach shows reconstruction results comparable to a classical post-processing network, whereas for instances not represented in the training set, the loss of performance is much less present. This demonstrates the robustness to distribution shifts of our approach.

REFERENCES

[1] Bauschke, H. & Borwein, J. On projection algorithms for solving convex feasibility problems. *SIAM Review*. **38**, 367-426 (1996)

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

[3] Honig, S. & Werman, M. Image declipping with deep networks. *2018 25Th IEEE International Conference On Image Processing (ICIP)*. pp. 3923-3927 (2018)

[4] Adler, A., Emiya, V., Jafari, M., Elad, M., Gribonval, R. & Plumbley, M. A constrained matching pursuit approach to audio declipping. *2011 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*. pp. 329-332 (2011)

[5] Boink, Y. & Brune, C. Learned SVD: solving inverse problems via hybrid autoencoding. *ArXiv:1912.10840*. (2019)

[6] Arridge, S., Maass, P., Öktem, O. & Schönlieb, C. Solving inverse problems using data-driven models. *Acta Numerica*. **28** pp. 1-174 (2019)

[7] Armato III, S., McLennan, G., Bidaut, L., McNitt-Gray, M., Meyer, C., Reeves, A., Zhao, B., Aberle, D., Henschke, C., Hoffman, E. & Others The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Medical Physics*. **38**, 915-931 (2011)

[8] Sun, J., Li, H., Xu, Z. & Others Deep ADMM-Net for compressive sensing MRI. *Advances In Neural Information Processing Systems*. pp. 10-18 (2016)

[9] Kofler, A., Haltmeier, M., Kolbitsch, C., Kachelrieß, M. & Dewey, M. A U-Nets cascade for sparse view computed tomography. *International Workshop On Machine Learning For Medical Image Reconstruction*. pp. 91-99 (2018)

[10] Gilton, D., Ongie, G. & Willett, R. Neumann networks for inverse problems in imaging. *ArXiv:1901.03707*. (2019)

[11] Lewitt, R. Multidimensional digital image representations using generalized Kaiser–Bessel window functions. *JOSA A*. **7**, 1834-1846 (1990)

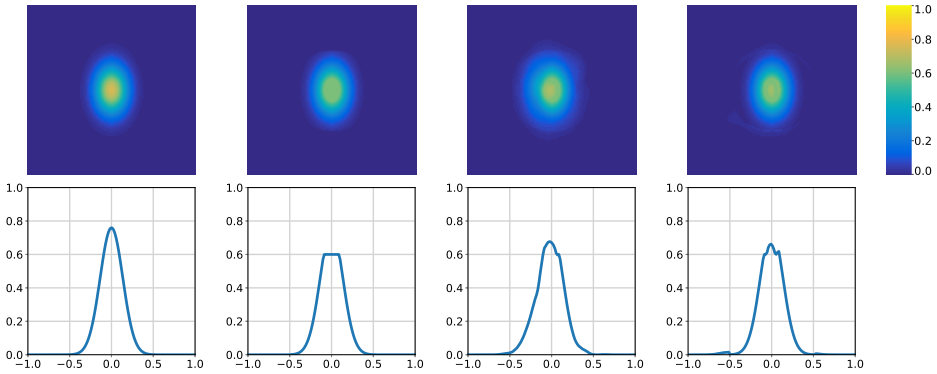
[12] Hoop, M., Lassas, M. & Wong, C. Deep learning architectures for nonlinear operator functions and nonlinear inverse problems. *ArXiv:1912.11090*. (2019)

[13] Zhu, B., Liu, J., Cauley, S., Rosen, B. & Rosen, M. Image reconstruction by domain-transform manifold learning. *Nature*. **555**, 487-492 (2018)
[14] Leuschner, J., Schmidt, M., Baguer, D. & Maaß, P. The LoDoPaB-CT Dataset: A Benchmark Dataset for Low-Dose CT Reconstruction Methods. *ArXiv:1910.01113*. (2019)
[15] Kobler, E., Klatzer, T., Hammernik, K. & Pock, T. Variational networks: connecting variational methods and deep learning. *German Conference On Pattern Recognition*. pp. 281-293 (2017)
[16] Boink, Y., Manohar, S. & Brune, C. A Partially Learned Algorithm for Joint Photoacoustic Reconstruction and Segmentation.. (IEEE,2019)
[17] Rick Chang, J., Li, C., Poczos, B., Vijaya, B. & Sankaranarayanan, A. One Network to Solve Them All–Solving Linear Inverse Problems Using Deep Projection Models. *Proceedings Of The IEEE International Conference On Computer Vision*. pp. 5888-5897 (2017)
[18] Benning, M. & Burger, M. Modern regularization methods for inverse problems. *Acta Numer.*. **27** pp. 1-111 (2018)
[19] Gupta, H., Jin, K., Nguyen, H., McCann, M. & Unser, M. CNN-Based Projected Gradient Descent for Consistent CT Image Reconstruction. *IEEE Trans. Med. Imag.*. **37**, 1440-1453 (2018)
[20] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proceedings Of The IEEE Conference On Computer Vision And Pattern Recognition*. pp. 770-778 (2016)
[21] Li, H., Schwab, J., Antholzer, S. & Haltmeier, M. NETT: Solving inverse problems with deep neural networks. *Inverse Problems, Online First*. **36**, 065005 (2020)
[22] Shalev-Shwartz, S. & Ben-David, S. Understanding machine learning: From theory to algorithms. (Cambridge university press,2014)
[23] Grasmair, M. Generalized Bregman distances and convergence rates for non-convex regularization methods. *Inverse Problems*. **26**, 115014, 16 (2010), http://dx.doi.org/10.1088/0266-5611/26/11/115014
[24] Hofmann, B., Kaltenbacher, B., Pöschl, C. & Scherzer, O. A convergence rates result for Tikhonov regularization in Banach spaces with non-smooth operators. *Inverse Problems*. **23**, 987-1010 (2007), http://dx.doi.org/10.1088/0266-5611/23/3/009
[25] Maas, A., Hannun, A. & Ng, A. Rectifier nonlinearities improve neural network acoustic models. *Proc. ICML*. **30** (2013)
[26] He, K., Zhang, X., Ren, S. & Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proc. ICCV*. pp. 1026-1034 (2015)
[27] Resmerita, E. On total convexity, Bregman projections and stability in Banach spaces. *J. Convex Anal.*. **11**, 1-16 (2004)
[28] LeCun, Y., Kavukcuoglu, K. & Farabet, C. Convolutional networks and applications in vision. *Circuits And Systems (ISCAS), Proceedings Of 2010 IEEE International Symposium On*. pp. 253-256 (2010)
[29] Butnariu, D. & Iusem, A. Totally convex functions for fixed points computation and infinite dimensional optimization. (Kluwer Academic Publishers, Dordrecht,2000), http://dx.doi.org/10.1007/978-94-011-4066-9
[30] Grasmair, M., Haltmeier, M. & Scherzer, O. Sparse regularization with lq penalty term. *Inverse Problems*. **24**, 055020, 13 (2008)
[31] Megginson, R. An introduction to Banach space theory. (Springer-Verlag, New York,1998), http://dx.doi.org/10.1007/978-1-4612-0603-3
[32] Engl, H., Hanke, M. & Neubauer, A. Regularization of inverse problems. (Kluwer Academic Publishers Group,1996)
[33] Natterer, F. & Wübbeling, F. Mathematical Methods in Image Reconstruction. (SIAM,2001)
[34] Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M. & Lenzen, F. Variational methods in imaging. (Springer,2009)
[35] King, J. & Chillingworth, D. Approximation of generalized inverses by iterated regularization. *Numer. Funct. Anal. Optim.*. **1**, 499-513 (1979)
[36] Lardy, L. A series representation for the generalized inverse of a closed linear operator. *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur. (8)*. **58**, 152-157 (1975)
[37] Burger, M. & Osher, S. Convergence rates of convex variational regularization. *Inverse Probl.*. **20**, 1411-1421 (2004)
[38] Finch, D., Haltmeier, M. & Rakesh Inversion of spherical means and the wave equation in even dimensions. *SIAM J. Appl. Math.*. **68**, 392-412 (2007)

[39] Grasmair, M., Haltmeier, M. & Scherzer, O. Necessary and sufficient conditions for linear convergence of ℓ^1 -regularization. *Comm. Pure Appl. Math.*. **64**, 161-182 (2011)
[40] Lorenz, D. Convergence rates and source conditions for Tikhonov regularization with sparsity constraints. *J. Inverse Ill-Posed Probl.*. **16**, 463-478 (2008)
[41] Haltmeier, M. Sampling conditions for the circular Radon transform. *IEEE Trans. Image Process.*. **25**, 2910-2919 (2016)
[42] Wang, S., Su, Z., Ying, L., Peng, X., Zhu, S., Liang, F., Feng, D. & Liang, D. Accelerating magnetic resonance imaging via deep learning. *IEEE 13th International Symposium On Biomedical Imaging (ISBI)*. pp. 514-517 (2016)
[43] Wang, G. A perspective on deep imaging. *IEEE Access*. **4** pp. 8914-8924 (2016)
[44] Würfl, T., Ghesu, F., Christlein, V. & Maier, A. Deep Learning Computed Tomography. *International Conference On Medical Image Computing And Computer-Assisted Intervention*. pp. 432-440 (2016)
[45] Antholzer, S., Haltmeier, M. & Schwab, J. Deep learning for photoacoustic tomography from sparse data. *Inverse Problems In Science And Engineering*. **27**, 987-1005 (2019)
[46] Hamilton, S. & Hauptmann, A. Deep D-bar: Real time Electrical Impedance Tomography Imaging with Deep Neural Networks. *IEEE Trans. Med. Imag.*. pp. 2367-2377 (2018)
[47] Jin, K., McCann, M., Froustey, E. & Unser, M. Deep convolutional neural network for inverse problems in imaging. *IEEE Trans. Image Process.*. **26**, 4509-4522 (2017)
[48] Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., Zhou, J. & Wang, G. Low-dose CT via convolutional neural network. *Biomed. Opt. Express*. **8**, 679-694 (2017)
[49] Adler, J. & Öktem, O. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Probl.*. **33**, 124007 (2017)
[50] Mallat, S. A wavelet tour of signal processing: The sparse way. (Elsevier/Academic Press, Amsterdam,2009)
[51] Candès, E. & Donoho, D. Recovering Edges In Ill-Posed Inverse Problems: Optimality Of Curvelet Frames. *Ann. Statist.*. **30**, 784-842 (2002)
[52] Daubechies, I. Ten Lectures on Wavelets. (SIAM,1992)
[53] Gribonval, R. & Schnass, K. Dictionary Identification — Sparse Matrix-Factorization via ℓ^1-Minimization. *IEEE Trans. Inf. Theory*. **56**, 3523-3539 (2010)
[54] Aharon, M., Elad, M. & Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Proc.*. **54**, 4311-4322 (2006)
[55] Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference On Medical Image Computing And Computer-Assisted Intervention*. pp. 234-241 (2015)
[56] Grasmair, M., Haltmeier, M. & Scherzer, O. The residual method for regularizing ill-posed problems. *Appl. Math. Comput.*. **218**, 2693-2710 (2011)
[57] Goodfellow, I., Bengio, Y. & Courville, A. Deep Learning. (MIT Press,2016), http://www.deeplearningbook.org
[58] Kuchment, P. & Kunyansky, L. Mathematics of thermoacoustic and photoacoustic tomography. *Eur. J. Appl. Math.*. **19** pp. 191-224 (2008)
[59] Wang, L. Multiscale photoacoustic microscopy and computed tomography. *Nature Phot.*. **3**, 503-509 (2009)
[60] Rosenthal, A., Ntziachristos, V. & Razansky, D. Acoustic inversion in optoacoustic tomography: A review. *Curr. Med. Imaging Rev.*. **9**, 318-336 (2013)
[61] Wang, K. & Anastasio, M. Photoacoustic and thermoacoustic tomography: image formation principles. *Handbook Of Mathematical Methods In Imaging*. pp. 781-815 (2011)
[62] Burgholzer, P., Bauer-Marschallinger, J., Grün, H., Haltmeier, M. & Paltauf, G. Temporal back-projection algorithms for photoacoustic tomography with integrating line detectors. *Inverse Probl.*. **23**, S65-S80 (2007)
[63] Paltauf, G., Nuster, R., Haltmeier, M. & Burgholzer, P. Photoacoustic tomography using a Mach-Zehnder interferometer as an acoustic line detector. *Appl. Opt.*. **46**, 3352-3358 (2007)
[64] Combettes, P. & Pesquet, J. Proximal splitting methods in signal processing. *Fixed-point Algorithms For Inverse Problems In Science And Engineering*. pp. 185-212 (2011)
[65] Ramlau, R. & Teschke, G. A Tikhonov-based projection iteration for nonlinear ill-posed problems with sparsity constraints. *Num. Math.*. **104**, 177-203 (2006)
[66] Daubechies, I., Defrise, M. & De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.*. **57**, 1413-1457 (2004)

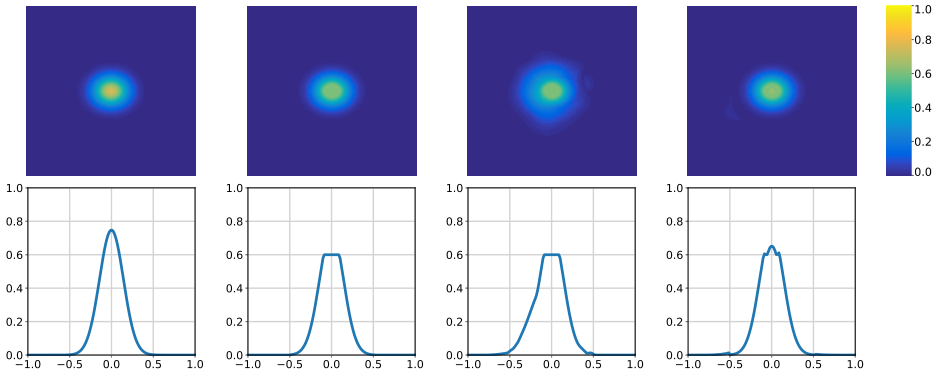
[67] Candes, E. & Donoho, D. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Comm. Pure Appl. Math.*. **57**, 219-266 (2004)
[68] Daubechies, I. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*. **41**, 909-996 (1988)
[69] Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., Zhou, J. & Wang, G. Low-dose CT via convolutional neural network. *Biomed. Opt. Express*. **8**, 679-694 (2017)
[70] Lee, D., Yoo, J. & Ye, J. Deep residual learning for compressed sensing MRI. *Biomedical Imaging (ISBI 2017), IEEE 14th International Symposium On*. pp. 15-18 (2017)
[71] Kang, E., Min, J. & Ye, J. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med. Phys.*. **44** (2017)
[72] Rivenson, Y., Göröcs, H., Zhang, Y., Wang, H. & Ozcan, A. Deep learning microscopy. *Optica*. **4**, 1437-1443 (2017)
[73] Schlemper, J., Caballero, J., Hajnal, J., Price, A. & Rueckert, D. A deep cascade of convolutional neural networks for dynamic MR image reconstruction. *IEEE Trans. Med. Imag.*. **37**, 491-503 (2018)
[74] Mardani, M., Gong, E., Cheng, J., Vasanawala, S., Zaharchuk, G., Alley, M., Thakur, N., Han, W., Pauly, J. & Others Deep generative adversarial networks for compressed sensing automates MRI. *ArXiv:1706.00051*. (2017)
[75] Kaltenbacher, B., Neubauer, A. & Scherzer, O. Iterative regularization methods for nonlinear ill-posed problems. (Walter de Gruyter GmbH & Co. KG, Berlin,2008)
[76] Tikhonov, A. & Arsenin, V. Solutions of Ill-Posed Problems. (John Wiley & Sons,1977)
[77] Morozov, V. Methods for Solving Incorrectly Posed Problems. (Springer Verlag,1984)
[78] Bakushinsky, A. & Kokurin, M. Iterative Methods for Approximate Solution of Inverse Problems. (Springer,2004)
[79] Lunz, S., Öktem, O. & Schönlieb, C. Adversarial regularizers in inverse problems. *Advances In Neural Information Processing Systems*. pp. 8507-8516 (2018)
[80] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. Improved training of wasserstein gans. *ArXiv Preprint ArXiv:1704.00028*. (2017)
[81] Schwab, J., Antholzer, S. & Haltmeier, M. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Problems*. **35**, 025008 (2019)
[82] Schwab, J., Antholzer, S. & Haltmeier, M. Big in Japan: Regularizing networks for solving inverse problems. *Journal Of Mathematical Imaging And Vision*. **62** pp. 445-455 (2020)
[83] Hansen, P. Discrete inverse problems: insight and algorithms. (SIAM,2010)
[84] Meyer, E., Raupach, R., Lell, M., Schmidt, B. & Kachelrieß, M. Normalized metal artifact reduction (NMAR) in computed tomography. *Medical Physics*. **37**, 5482-5493 (2010)
[85] Zhang, X., Wang, J. & Xing, L. Metal artifact reduction in x-ray computed tomography (CT) by constrained optimization. *Medical Physics*. **38**, 701-711 (2011)
[86] Bubba, T., Kutyniok, G., Lassas, M., März, M., Samek, W., Siltanen, S. & Srinivasan, V. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *Inverse Problems*. **35**, 064002 (2019)
[87] Mukherjee, S., Dittmer, S., Shumaylov, Z., Lunz, S., Öktem, O. & Schönlieb, C. Learned convex regularizers for inverse problems. *ArXiv Preprint ArXiv:2008.02839*. (2020)

Appendix A. **Additional results for saturated Gaussians.** In this appendix, three more samples from the modified test set of the saturated Gaussians are shown. It can be seen in Figures 8 and 9 that U-Net tends to slightly widen the Gaussian in some cases, since this was necessary for the wider Gaussians in the training set. Moreover it can be seen in Figures 9 and 10 that both U-Net and the data-consistent network sometimes fail to restore the top of the Gaussian adequately: the training on wider Gaussians is not directly generalised for smaller Gaussians.



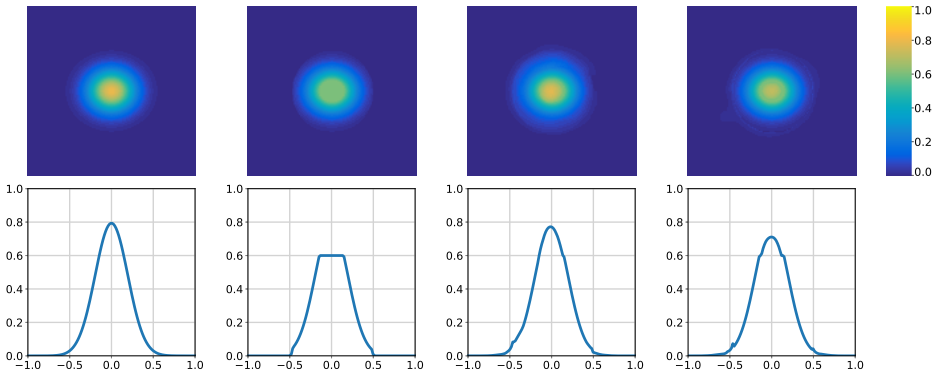
(A) Ground truth (B) Saturated (C) U-Net (D) Data-consistent

FIGURE 8



(A) Ground truth (B) Saturated (C) U-Net (D) Data-consistent

FIGURE 9

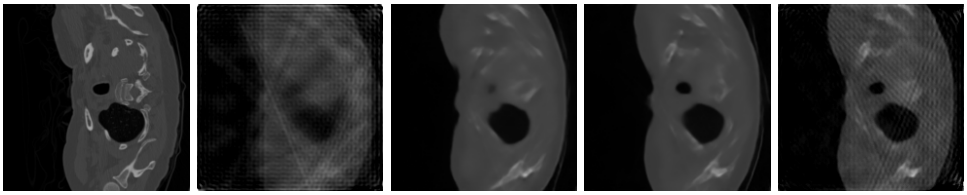


(A) Ground truth (B) Saturated (C) U-Net (D) Data-consistent

FIGURE 10

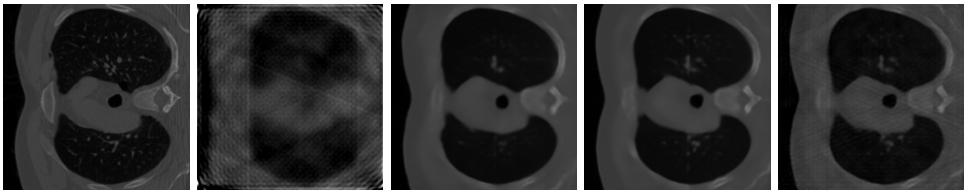
Appendix B. **Additional results for saturated Radon transform.** In this appendix, three additional samples from the regular and the modified test set of human chest images are shown. The samples have been selected based on their PSNR values: we show the samples in the test set for which the data-consistent network yields the highest relative PSNR value (Figures 11 and 14), a similar PSNR value (Figures 12 and 15), and the lowest relative PSNR value (Figures 13 and 16) when compared to the U-Nets.

B.1. **Regular test set.**



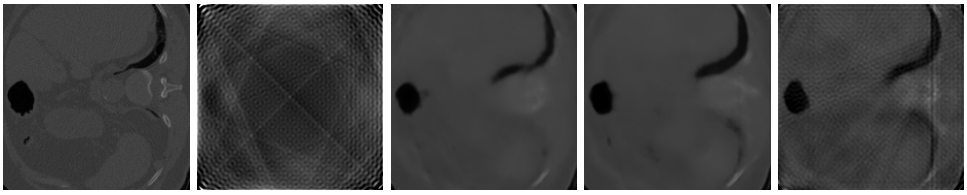
(A) Ground truth | (B) Pseudo-inverse PSNR = 29.01 | (C) One U-Net PSNR = 31.50 | (D) Two U-Nets PSNR = 32.29 | (E) Data-consistent PSNR = 34.10

FIGURE 11. Sample for which the data-consistent PSNR value is relatively high compared to the U-Net PSNR values (grayscale from 0 to 1).



(A) Ground truth | (B) Pseudo-inverse PSNR = 25.33 | (C) One U-Net PSNR = 30.09 | (D) Two U-Nets PSNR = 29.94 | (E) Data-consistent PSNR = 30.09

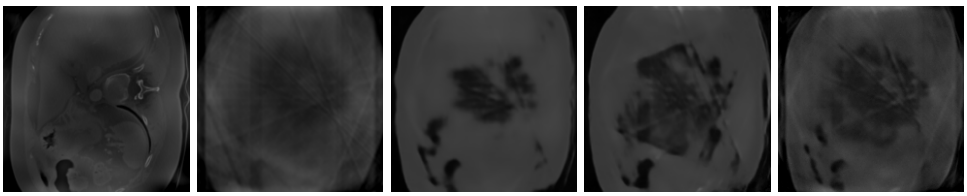
FIGURE 12. Sample for which the data-consistent PSNR value is approximately the same as the U-Net PSNR values (grayscale from 0 to 1).



(A) Ground truth | (B) Pseudo-inverse PSNR = 22.87 | (C) One U-Net PSNR = 29.78 | (D) Two U-Nets PSNR = 28.36 | (E) Data-consistent PSNR = 24.52

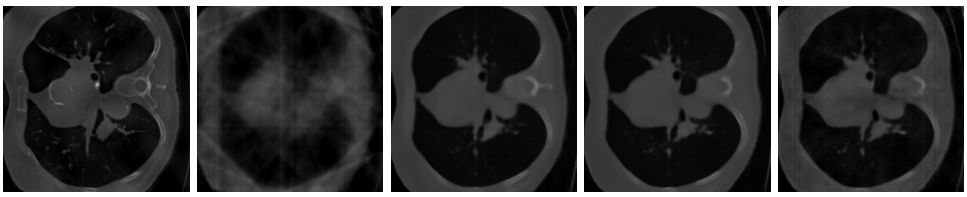
FIGURE 13. Sample for which the data-invariant PSNR value is relatively low compared to the U-Net PSNR values (grayscale from 0 to 1).

B.2. **Modified test set.** The ground truth images in the modified test set contain more regions which are non-constant, as opposed to the regular test set, which consists of piecewise constant images. The data-consistent network is better able to deal with these modifications in the images, as can be seen particularly well in Figure 14: The U-Nets create a piecewise constant dark structure in the middle, while the data-consistent network keeps it more smooth.



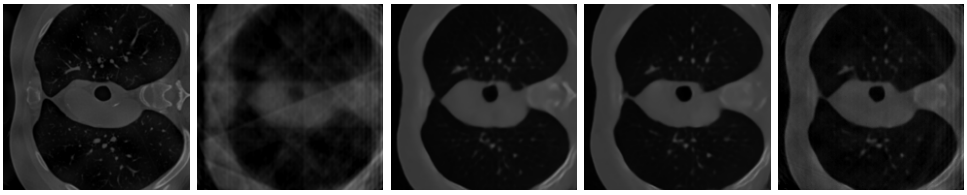
(A) Ground truth | (B) Pseudo-inverse PSNR = 34.76 | (C) One U-Net PSNR = 25.96 | (D) Two U-Nets PSNR = 27.61 | (E) Data-consistent PSNR = 33.61

FIGURE 14. Sample for which the data-invariant PSNR value is relatively high compared to the U-Net PSNR values (grayscale from 0 to 1).



(A) Ground truth | (B) Pseudo-inverse PSNR = 28.43 | (C) One U-Net PSNR = 31.50 | (D) Two U-Nets PSNR = 31.52 | (E) Data-consistent PSNR = 31.47

FIGURE 15. Sample for which the data-invariant PSNR value is approximately the same as the U-Net PSNR values (grayscale from 0 to 1).



(A) Ground truth | (B) Pseudo-inverse PSNR = 28.62 | (C) One U-Net PSNR = 31.05 | (D) Two U-Nets PSNR = 31.30 | (E) Data-consistent PSNR = 30.89

FIGURE 16. Sample for which the data-invariant PSNR value is relatively low compared to the U-Net PSNR values (grayscale from 0 to 1).

Received xxxx 20xx; revised xxxx 20xx; early access xxxx 20xx.

E-mail address: yoeriboink@gmail.com
E-mail address: markus.haltmeier@uibk.ac.at
E-mail address: sean.holman@manchester.ac.uk
E-mail address: schwab@mrc-lmb.cam.ac.uk