Deep Decomposition Learning for Inverse Imaging Problems

Dongdong Chen and Mike E. Davies

School of Engineering, University of Edinburgh, UK
{d.chen, mike.davies}@ed.ac.uk

Abstract. Deep learning is emerging as a new paradigm for solving inverse imaging problems. However, the deep learning methods often lack the assurance of traditional physics-based methods due to the lack of physical information considerations in neural network training and deploying. The appropriate supervision and explicit calibration by the information of the physic model can enhance the neural network learning and its practical performance. In this paper, inspired by the geometry that data can be decomposed by two components from the null-space of the forward operator and the range space of its pseudo-inverse, we train neural networks to learn the two components and therefore learn the decomposition, *i.e.* we explicitly reformulate the neural network layers as learning range-nullspace decomposition functions with reference to the layer inputs, instead of learning unreferenced functions. We empirically show that the proposed framework demonstrates superior performance over recent deep residual learning, unrolled learning and nullspace learning on tasks including compressive sensing medical imaging and natural image super-resolution. Our code is available at `https://github.com/edongdongchen/DDN`.

Keywords: decomposition learning, physics, inverse problems

1 Introduction

We consider a linear inverse problem of the form :

$$\mathbf{y}_\epsilon = \mathbf{H}\mathbf{x} + \epsilon, \tag{1}$$

where the goal is to recover the unknown signal $\mathbf{x} \in \mathbb{R}^D$ from the noisy measurement $\mathbf{y}_\epsilon \in \mathbb{R}^d$ with typical dimension $D \gg d$, and $\mathbf{H} : \mathbb{R}^D \to \mathbb{R}^d$ is the forward operator which models the response of the acquisition device or reconstruction system, while $\epsilon \in \mathbb{R}^d$ represents the measurement noise intrinsic to the acquisition process.

Inverse problems have wide applications in computer vision, medical imaging, optics, radar, and many other fields. The forward operator $\mathbf{H}$ in (1) could represent various inverse problems, from *e.g.* an identity operator for image denoising, to convolution operators for image deblurring, random sensing matrices for compressive sensing(CS), filtered subsampling operators for super-resolution

(SR), (undersampled) Fourier transform for magnetic resonance imaging (MRI) and the (subsampled) Radon transform in computed tomography (CT). The inverse problems in (1) are often noisy and ill-posed since the operator $\mathbf{H}$ has a non-trivial null space. Such under-determined systems are extremely difficult to solve and the solutions are very sensitive to the input data. The classical approach for solving them have traditionally been model-based [4,9], which typically aim to regularize the solutions by constraining them to be consistent with prior knowledge about the signal and usually can only be solved iteratively.

More recently, due to the powerful representation learning and transformation ability, deep neural networks (DNN) have emerged as a new paradigm for inverse problems. The community has already taken significant steps in this direction, with deep neural networks being successfully applied to a wide variety of inverse problems [25]. For example, [5,48] use a fully connected feedfoward neural network for image denoisng and inpainting. [12,22] learn end-to-end mappings between $\mathbf{y}_\epsilon$ and $\mathbf{x}$ with vanilla convolutional neural networks (CNN). [51,24] further use CNNs with residual blocks [17] and skip connections to improve the neural network performance. [28,36,20] learn downsampling and upsampling feature maps with encoder-decoder CNNs. [8,50] use autoencoders for learning new representations for $\mathbf{x}$ and $\mathbf{y}_\epsilon$ to solve the inverse problems. [47] use CNN as a prior and train with early stopping criteria to recover a single image from its observation. [33] use CNN to perform denoising-based approximate message passing. [46,31] unfold the model-based optimizations with DNN. [34,3] use generative models for natural images to recover images from Gaussian measurements.

However, the DNN itself in the above deep learning-based approaches often lack the guarantees of traditional physics-based methods as they are purely data-driven and learning-based. In addition, the designing of DNNs is usually complicated and has poor intuitive interpretation when they are decoupled from the inverse problem of interest. Furthermore, it is a commonly held belief in the inverse problems community that using the physics is preferable to relying solely on data [41,30]. This raises a number of questions: is a purely data-driven neural network the best way to solve an inverse problem? Does physical information facilitate neural networks to find a better inverse problem solution? How should one best make use of the prior physical (acquisition) information? All of these above questions inspire us to think about whether the introduction of physical information in neural networks would be beneficial to the training of deep learning methods to better solve inverse problems.

We present *Deep Decomposition learning Network* (DDN) as a way of using physics to reconstruct image from its measurement $\mathbf{y}_\epsilon$ ($\epsilon \neq 0$). Relying on the range-nullspace decomposition of data and the recent null-space learning [30,41], also known as affine projected network in [44], we propose to use two sets of neural network layers to separately capture the residuals lying on the range of $\mathbf{H}^\dagger$ and the nullspace of $\mathbf{H}$. By incorporating the two learned residuals with the pseudo-inverse input, the proposed framework DDN is able to recover a plausible image and preserve data consistency with respect to the measurements $\mathbf{y}_\epsilon$.

Decomposition learning can be used with various inverse imaging problems. In this paper, we follow a compressive sensing magnetic resonance fingerprinting (CS-MRF) reconstruction task [10,27]: reconstructing brain MR images sequence from highly undersampled and noisy Fourier space data. We show that the DDN outperforms recent related deep learning methods including deep residual learning [20] which is physics-free, deep nullspace learning [44,30,41] which neglected the noise component, and deep unrolled learning [15,16,38] which retained the undesirable iterative nature of model-based systems. We also performed an ablation study on the natural image super-resolution (SR) task, and the results show that simultaneous learning of two residuals is outperforming the way that does not use physics or only learn one of the residuals. Finally, all the experiments demonstrate that decomposition learning can facilitate the generalization of the deep neural network for inverse problems.

2 Background

2.1 Deep learning for the inverse problem

Depending on whether the physical acquisition information with respect to $\mathbf{H}$ is used during DNN training and testing, we divide the deep learning approaches into two categories: *Physics-free* and *Physics-engaged.*

Physics-free. The DNN aims to learn a direct mapping from $\mathbf{y}_\epsilon$ (or its projection, e.g. $\mathbf{H}^\dagger \mathbf{y}_\epsilon$) to $\mathbf{x}$ without exploiting the knowledge of $\mathbf{H}$ at any point in the training or testing process (with the exception of the input). The general principle is that, given enough training data, we should be able to design a proper DNN to learn everything we need to know about $\mathbf{H}$ to successfully estimate $\mathbf{x}$ directly. The success of this approach is likely to depend on the complexity of the forward operator $\mathbf{H}$. However, it has been observed to work well for numerous computer vision tasks, such as denoising and inpainting [48], superresoluion [12] and deblurring [49,23]. The DNN can be trained by a sole least squares loss [48,28,20] or a combination of least squares loss and auxiliary losses such as adversarial loss [24,23,37,43]. In general, this approach requires large quantities of training data because it is required to not only learn the geometry of the image space containing $\mathbf{x}$, but also aspects of $\mathbf{H}$. Hence, an interesting question is how hard (relatively) are each of these components to learn and how important is it to incorporate $\mathbf{H}$ into the learning process. When the forward problem is too complex such that it can not be incorporated into the DNN model it will always be necessary to go Physics-free. Finally, since direct estimation using a DNN for solving inverse problems is essentially a form of regression, there is a potential generalization issue with such physic-free DNN approaches.

Physics-engaged. The most widely used strategy considering physics of $\mathbf{H}$ in deep learning approaches is through a model-based approach, in which one or more pretrained DNNs are used within a more traditional iterative physics-engaged model-based framework such as [4,9]. As mentioned before, the inverse problem (1) typically boils down to solving an optimisation problem broadly of

the following form:

$$\arg\min_{\mathbf{x}} f(\mathbf{x}) + \lambda\phi(\mathbf{x}), \tag{2}$$

where the first term $f(\mathbf{x})$ aims to enforce data fidelity, *e.g.* f could be the MSE between $\mathbf{Hx}$ and $\mathbf{y}_\epsilon$, while the regularizer ϕ allows us to insert knowledge onto the solution $\mathbf{x}$, and $\lambda \in \mathbb{R}^+$ controls the strength of the regularization. Typically there is no closed-form solution to (2) and it usually needs to be solved iteratively. This has led to the following proposed uses for pretrained DNNs: (i) use DNN to replace the proximal operator associated with $\phi(\mathbf{x})$ in a proximal gradient algorithm [15,16,38], (ii) use DNN to replace the gradient $\nabla\phi$ in an unrolled optimization method [7,46,31], (iii) directly replace the regularizer ϕ with DNN [26,39], (iv) use DNN as a generative model to generate $\mathbf{x}$ from a latent code that needs to be estimated [3,42]. These iterative methods are Physics-engaged, as they actually use the regularizer along with the forward model and observation by minimizing the disparity between the oracle and its reconstruction.

As an exception to the above physics-engaged deep learning approaches, there have been some recent studies aimed at explicitly using $\mathbf{H}$-related information during the DNN training process in an end-to-end manner. For example, [41,30] explicitly learn the nullspace component of $\mathbf{x}$ with respect to $\mathbf{H}$. However, this separate nullspace learning does not deal with the presence of noise in the input nor with situations where no nullspace exists. Another interesting direction presented in [11] considers a Neumann series expansion of linear operators to approximate the inverse mapping of (2). However, this requires the network to precondition and store the results of each iteration in order to accumulate the approximate Neumann series to solve the inverse problem.

In this paper, inspired by the nullspace method of [30,41,44], we explore the possibility of a more flexible end-to-end neural network structure that is capable of exploiting both the range and null space structures of the inverse problem. Before discussing the proposed method, let us briefly recall the Range-Nullspace decomposition of data.

2.2   Range-Nullspace ($\mathcal{R}$-$\mathcal{N}$) Decomposition

Given a linear forward operator $\mathbf{H} \in \mathbb{R}^{d \times D}$ and its right pseudo inverse $\mathbf{H}^\dagger \in \mathbb{R}^{D \times d}$, which satisfies $\mathbf{HH}^\dagger = \mathbf{I}_d$, it holds that $\mathbb{R}^D = \mathcal{R}(\mathbf{H}^\dagger) \oplus \mathcal{N}(\mathbf{H})$, which implies that for any sample $\forall \mathbf{x} \in \mathbb{R}^D$ there exists two unique elements $\mathbf{x}^+ \in \mathcal{R}(\mathbf{H}^\dagger)$ and $\mathbf{x}^\perp \in \mathcal{N}(\mathbf{H})$ such that $\mathbf{x} = \mathbf{x}^+ + \mathbf{x}^\perp$. Therefore we define the following range-nullspace ($\mathcal{R}$-$\mathcal{N}$) decomposition,

Definition 1. $\mathcal{R}$-$\mathcal{N}$ *Decomposition:* Let $\mathcal{P}_r \triangleq \boldsymbol{H}^\dagger \boldsymbol{H}$ *be the operator that projects the sample* $\boldsymbol{x}$ *from sample domain to the range of* $\boldsymbol{H}^\dagger$, *and denote by* $\mathcal{P}_n \triangleq (\boldsymbol{I}_D - \boldsymbol{H}^\dagger \boldsymbol{H})$ *the operator that projects* $\boldsymbol{x}$ *to the null space of* $\boldsymbol{H}$. *Then* $\forall \boldsymbol{x} \in \mathbb{R}^D$, *there exists the unique decomposition:*

$$\boldsymbol{x} = \mathcal{P}_r(\boldsymbol{x}) + \mathcal{P}_n(\boldsymbol{x}), \tag{3}$$

where we will call $\mathcal{P}_r(\boldsymbol{x})$ *and* $\mathcal{P}_n(\boldsymbol{x})$ *the* r-component *and* n-component *of* $\boldsymbol{x}$, *respectively.*

Remark 1. In this paper we will only focus on the above decomposition. However, we comment that in principle the pseudo-inverse, $\mathbf{H}^\dagger$, could be replaced by any general right inverse of $\mathbf{H}$ in the above decomposition which might provide added flexibility in certain inverse problems.

Thus, the task of solving an inverse problem is to find these two components $\mathcal{P}_r(\mathbf{x})$ and $\mathcal{P}_n(\mathbf{x})$ based on the observed data, $\mathbf{y}_\epsilon$. The simple linear estimator to solve this problem is to use the approximation:

$$\mathbf{x}^* = \mathbf{H}^\dagger \mathbf{y}_\epsilon. \tag{4}$$

This estimator enjoys global and exact data-consistency, *i.e.* $\mathbf{H}\mathbf{x}^* \equiv \mathbf{y}_\epsilon$, which is an important consideration when solving inverse problems [30]. However, comparing (4) with (3) we can see that this is achieved by simply setting the nullspace component to zero: $\mathcal{P}_n(\mathbf{H}^\dagger \mathbf{y}_\epsilon) = 0$. In general this provides a poor solution for ill-posed problems. Thus it is necessary to further estimate the missing component $\mathcal{P}_n(\mathbf{x})$. Such an estimator is necessarily nonlinear.

Nullspace learning. Recently, [30,41,44] explored the use of a neural network $\mathcal{G}$ to feed a refined backprojection $\mathcal{G}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)$ to the null-space projection operator $\mathcal{P}_n$, then the reconstruction in (4) is reformulated as

$$\mathbf{x}^* = \mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_n(\mathcal{G}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)), \tag{5}$$

where the network $\mathcal{G}$ is suggested to be trained by minimizing the error between $\mathbf{x}$ and $\mathbf{x}^*$. Note the solution (5) enjoys global data consistency, *i.e.* $\mathbf{H}\mathbf{x}^* \equiv \mathbf{y}_\epsilon$. However, the solution (5) unfortunately, only works for the noise-free situation, and does not allow any denoising in the range $\mathcal{R}(\mathbf{H}^\dagger)$. Indeed, (5) can only denoise in the nullspace and the denoising ability is therefore worst-case bounded by $\|\epsilon\|/\|\mathbf{H}\|$ since $\|\mathbf{H}(\mathbf{x} - \mathbf{x}^*)\| = \|\epsilon\|$. The noise may further limit the ability to predict the null space component from the noisy measurements. Although it is reminiscent of decoupling the neural network denoiser from the inverse problem, it does not benefit from this since the training needs to be tailored to the task [38], which will be confirmed in our experiments.

3 Deep Decomposition Learning

Inspired by nullspace learning [44,41,30] we aim to remove the range space denoising deficiency while still exploiting the nullspace property. Let us consider the case $\epsilon \neq 0$ in (1) which is more practical. By the $\mathcal{R}$-$\mathcal{N}$ decomposition (3), it holds that $\mathbf{x}$ can be exactly recovered by,

$$\mathbf{x} = \mathbf{H}^\dagger \mathbf{y}_\epsilon - \mathbf{H}^\dagger \epsilon + \mathcal{P}_n(\mathbf{x}). \tag{6}$$

However, as mentioned before, in the scenario of the inverse problem, both the oracle image $\mathbf{x}$ and the noise term ϵ in (6) are still unknown and need to be recovered.

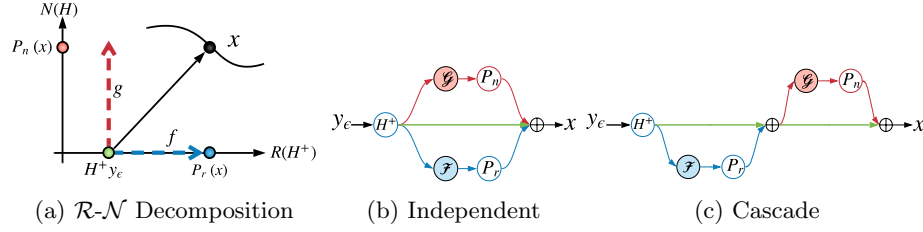(a) $\mathcal{R}$-$\mathcal{N}$ Decomposition   (b) Independent   (c) Cascade

Fig. 1: Decomposition learning (a) and its two corresponding network architectures in (b) and (c). Decomposition learning trains neural network to recovery the range component $\mathcal{P}_r(\mathbf{x}) = \mathbf{H}^\dagger \mathbf{H} \mathbf{x}$ and nullspace component $\mathcal{P}_n(\mathbf{x}) = \mathbf{x} - \mathcal{P}_r(\mathbf{x})$ of oracle image $\mathbf{x}$ from its coarse reconstruction $\mathbf{H}^\dagger \mathbf{y}_\epsilon$. The residuals between $\mathbf{H}^\dagger \mathbf{y}_\epsilon$ and $\mathcal{P}_r(\mathbf{x})$ and $\mathcal{P}_n(\mathbf{x})$ are recovered by $f = \mathcal{P}_r(\mathcal{F})$ and $g = \mathcal{P}_n(\mathcal{G})$ respectively, where $\mathcal{F}$ and $\mathcal{G}$ are two sets of neural layers need to train. Different from direct learn the residual between $\mathbf{x}$ and $\mathbf{H}^\dagger \mathbf{y}_\epsilon$ without using physics, $\mathbf{x}$ is instead reconstructed by $\mathbf{H}^\dagger \mathbf{y}_\epsilon + f + g$ in a decomposition learning mechanism. This method enables physics to engage in neural network training and testing and generate more physics plausible reconstruction.

We address this problem by using two neural networks and introducing the decomposition learning framework (illustrated in Fig.1a). Instead of hoping a single neural network will directly fit a desired underlying mapping between $\mathbf{H}^\dagger \mathbf{y}_\epsilon$ and $\mathbf{x} = \mathcal{P}_r(\mathbf{x}) + \mathcal{P}_n(\mathbf{x})$, we explicitly let two networks, denoted by $\mathcal{F}$ and $\mathcal{G}$ fit the two mappings from $\mathbf{H}^\dagger \mathbf{y}_\epsilon$ to the residual r-component $\mathcal{P}_r(\mathbf{x}) - \mathbf{H}^\dagger \mathbf{y}_\epsilon$ and the n-component $\mathcal{P}_n(\mathbf{x})$, respectively. In particular, the output of $\mathcal{F}$ should be bounded by the magnitude of noise ϵ, while $\mathcal{G}$ should be a smooth, *i.e.* a Lipschitz continuous neural network, since $\mathcal{G}$ is essentially a nullspace network, which does not need to be strongly bounded but should be regularized in order to get reasonable generalization. Therefore, the oracle $\mathbf{x}$ is decomposed as the sum of a linear component $\mathbf{H}^\dagger \mathbf{y}_\epsilon$ (the input), a bounded residual component $\mathcal{P}_r \circ \mathcal{F} \in \mathcal{R}(\mathbf{H}^\dagger)$ and a smooth n-component $\mathcal{P}_n \circ \mathcal{G} \in \mathcal{N}(\mathbf{H})$.

We consider two versions of DDN estimators. First, we define an independent connection architecture (Fig.1b) estimator $\mathcal{A}_i$ using the $\mathcal{R}$-$\mathcal{N}$ decomposition,

$$\mathcal{A}_i(\mathbf{y}_\epsilon) \triangleq \mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_r(\mathcal{F}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)) + \mathcal{P}_n(\mathcal{G}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)). \tag{7}$$

where there are no interactions between $\mathcal{F}$ and $\mathcal{G}$.

An alternative (but essentially equivalent) mapping $\mathcal{A}_c$ from $\mathbf{y}_\epsilon$ to $\mathbf{x}$ which is related to (7) uses a cascade of networks (Fig.1c), *i.e.* first denoising with $\mathcal{F}$, then feeding the denoised $\mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_r(\mathcal{F}(\mathbf{H}^\dagger \mathbf{y}_\epsilon))$ into $\mathcal{G}$ such that:

$$\mathcal{A}_c(\mathbf{y}_\epsilon) \triangleq \mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_r(\mathcal{F}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)) + \mathcal{P}_n(\mathcal{G}(\mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_r(\mathcal{F}(\mathbf{H}^\dagger \mathbf{y}_\epsilon)))). \tag{8}$$

Intuitively (8) is preferable since it is no more complex than the independent network topology and provides the nullspace network $\mathcal{G}$ with a range denoised input, thereby reducing the learning burden needed to be done by $\mathcal{G}$. Our experiments also verified the cascade connections typically perform better than the

independent one. Note that the independent model is by its nature a shallower but wider network than the cascade model which by construction is deeper, but given the decomposition both networks have essentially the same complexity.

The $\mathcal{R}$-$\mathcal{N}$ decomposition learning gives the outputs of the two networks $\mathcal{F}$ and $\mathcal{G}$ clear interpretability: the output from $\mathcal{F}$ estimates the noise component, while the output from $\mathcal{G}$ represents the component that is inferred from the implicit image model as it was not measured at all by $\mathbf{y}_\epsilon$. Second, the DDN estimator $\mathcal{A}(\mathbf{y}_\epsilon)$ defined in (7) and (8) offers the ability to denoise both the r-component $\mathcal{P}_r(\mathbf{x})$ and the n-component $\mathcal{P}_n(\mathbf{x})$, and if $\|\mathbf{H}\mathcal{F}\| \leq \|\epsilon\|$, the solution enjoys a relaxed notion of data-consistency (in the spirit of the discrepancy principle popular in inverse problems), which convergences to exact data consistency when $\|\mathbf{H}\mathcal{F} - \epsilon\| \to 0$, that makes the deep learning solutions more physically plausible. Finally, in the cascade architecture (Fig. 1c) we calculate $\mathcal{P}_r(\mathcal{F})$ first. If we had cascaded $\mathcal{P}_n(\mathcal{G}(\mathbf{H}^\dagger\mathbf{y}_\epsilon)) + \mathbf{H}^\dagger\mathbf{y}_\epsilon$ first we would have lost the $\mathcal{R}$-$\mathcal{N}$ decomposition as the input to $\mathcal{F}$ would live in the full space. The input to $\mathcal{G}$ would also have range noise, as in the independent architecture, thereby increasing the training burden on $\mathcal{G}$ network.

3.1   Training strategy

Let $\mathbf{X} = \{(\mathbf{y}_\epsilon^{(i)}, \mathbf{x}^{(i)})\}_{i=1}^N$ denote a training set of N samples, where $\mathbf{x}^{(i)}$ and $\mathbf{y}_\epsilon^{(i)}$ are the clean oracle signal and its noisy measurement given by (1). Denote by $\ell(x, y)$ the loss function, which measures the discrepancy between x and y. In this paper we used ℓ_2 to compute ℓ. Given an estimator $\mathcal{A}$, we jointly training $\mathcal{F}$ and $\mathcal{G}$ by solving a single optimization program,

$$\min_{\mathcal{F},\mathcal{G}} \ell_{\mathrm{emp}}(\mathcal{A}) + \lambda_1\phi_1(\mathcal{F}) + \lambda_2\phi_2(\mathcal{G}), \tag{9}$$

where the first term $\ell_{\mathrm{emp}}(\mathcal{A}) \triangleq \frac{1}{N}\sum_{\mathbf{y}_\epsilon^{(i)},\mathbf{x}^{(i)}\in\mathbf{X}} \ell(\mathcal{A}(\mathbf{y}_\epsilon^{(i)}), \mathbf{x}^{(i)})$ is $\mathcal{A}$'s empirical loss associated with the training set $\mathbf{X}$ and serves the data-fidelity, and $\mathcal{A}$ take the form either in (7) or in (8). In this paper, we set $\phi_1(\mathcal{F}) = \sum_{i=1}^N \ell(\mathbf{H}\mathcal{F}(\mathbf{H}^\dagger\mathbf{y}_\epsilon^{(i)}), \epsilon^{(i)})$ in order to encourage the data discrepancy term to be small. We then set ϕ_2 as the weight decay term to control the Lipschitz of the network [41] and to encourage good generalization [21]. It worth noting we can flexibly define different ϕ_1 and ϕ_2 for $\mathcal{F}$ and $\mathcal{G}$ to impose the desired boundedness and smoothness conditions, such that to tune the networks to their specific tasks.

Due to the decomposition $\mathbf{x} = \mathcal{P}_r(\mathbf{x}) + \mathcal{P}_n(\mathbf{x})$ in the independent architecture there are no interactions between the respective targets of $\mathcal{F}$ and $\mathcal{G}$, therefore one can decouple (9) into two independent sub-optimizations to train $\mathcal{F}$ and $\mathcal{G}$ separately, *i.e.* $\min_{\mathcal{F}}(\mathcal{P}_r(\mathcal{F}), \mathcal{P}_r(\mathbf{X})) + \lambda_1\phi_1$ and $\min_{\mathcal{G}}(\mathcal{P}_n(\mathcal{G}), \mathcal{P}_n(\mathbf{X})) + \lambda_2\phi_2$. While joint training and decoupled training are theoretically equivalent, in practice, the decoupled training enjoys more intuitive interpretability, and it is easier to control $\mathcal{F}$ and $\mathcal{G}$ each to achieve better convergence results. However, the joint training is slightly more efficient than the decoupled because the networks can be trained simultaneously. Accordingly we use the cascade architecture as our DDN in our experiments and jointly train it using (9).

3.2   The relationship to other work

In the noise-free case ($\epsilon = 0$), the decomposition learning (6) reduces naturally to vanilla nullspace learning (5). Thus a DDN can be regarded as a generalized nullspace network. While in the noisy case, one might be tempted to consider adopting a separate generic denoiser to preprocess the measurements $\mathbf{y}_\epsilon$. However, such denoisers are typically built in a manner decoupled from the inverse problem of interest, therefore to train a nullspace network with such denoised measurements could amplify the reconstruction error, causing more inconsistent results [38]. In contrast, the denoising process in the DDN is not decoupled from the inverse problem but integrated into a unified learning model. The experiments show this not only improves the quality of the results but in addition, helps the model's generalization. Note DDN is different from the recent heuristic decomposition network [35] which broadly separates the image into structure part and detail part. In contrast, we identify the natural $\mathcal{R}$-$\mathcal{N}$ decomposition induced by $\mathbf{H}$ that also allows us to select appropriate loss functions for the different components.

Our decomposition learning can also be regarded as a special gated neural network. To be specific, if we rewrite (7) as

$$\mathcal{A} = \mathbf{T}\mathcal{F}(\mathbf{z}) + (\mathbf{I} - \mathbf{T})\mathcal{G}(\mathbf{z}) + \mathbf{z}, \tag{10}$$

where $\mathbf{z} = \mathbf{H}^\dagger \mathbf{y}_\epsilon$, $\mathbf{T} = \mathbf{H}^\dagger \mathbf{H}$, it can be seen the output of $\mathcal{F}$ and $\mathcal{G}$ are gated in terms of $\mathbf{T}$ and $\mathbf{I} - \mathbf{T}$. The importance of the two components is determined by the physics in terms of $\mathbf{H}^\dagger \mathbf{H}$. This is different from previous gated networks such as [18,45] in which the model is gated by some bounded numerical function such as a sigmoid or hardlim which are not typically related to the physics of the forward model or its inverse. Our method can also be regarded as a generalized residual learning [17], *i.e.* we decompose the residual $\mathcal{A} - \mathbf{z}$ into two components in $\mathcal{R}(\mathbf{H}^\dagger)$ and $\mathcal{N}(\mathbf{H})$ with more explicit interpretability. In particular, in the absence of the nullspace $\mathcal{N}(\mathbf{H})$ or in the case $\mathbf{H} = \mathbf{I}$ such that $\mathcal{P}_r = \mathbf{I}$ and $\mathcal{P}_n = 0$, *i.e.* there is no nullspace learning, $\mathcal{G}$ will be irrelevant and only $\mathcal{F}$ will be learnable, and the decomposition learning will be reduced to a non-gated neural network and equivalent to the standard residual learning.

Remark 2. A key aspect of the DDN framework is access to the projection operators $\mathcal{P}_n$ and $\mathcal{P}_r$. The complexity will depend on how easy it is to approximate $\mathbf{H}^\dagger$, or another left inverse. While many computer vision problems admit an easily computable inverse, such as: deblurring, inpainting and the ones considered in this paper - compressed sensing and super-resolution, for more general inverse problems this approximation is more challenging and may even be unstable. However, it is possible to approximate $\mathbf{z} = \mathbf{H}^\dagger \mathbf{y}$ by a regularized linear operator, e.g. $\arg\min_\mathbf{z} \|\mathbf{y} - \mathbf{H}\mathbf{z}\|^2 + \lambda\|\mathbf{z}\|^2$. Here we can leverage the wealth of literature in inverse problems dedicated to efficiently computing this, e.g. using preconditioned conjugate gradient solvers, etc. *Unrolled* physics-engaged deep learning solutions [15,16,38] also solve a similar optimization problem but with an integrated complex DNN regularizer, and therefore resort to slower proximal

Table 1: Four DNN solvers $\mathcal{A}$ that use different mechanisms for using physics to solve inverse problems. Note the residual learning is physic-free and $\text{Prox}_{\mathbf{H}}$ is the proximal operator in proximal gradient descent optimization.

Mechanisim	Residual learning	Unrolled learning	Nullspace learning	Decomposition learning
Formulation	$\mathcal{A} = \mathbf{I} + \mathcal{G} \circ \mathcal{F}$	$\text{Prox}_{\mathbf{H}} \leftarrow \mathcal{G} \circ \mathcal{F}$	$\mathcal{A} = \mathbf{I} + \mathcal{P}_n(\mathcal{G} \circ \mathcal{F})$	$\mathcal{A} = \mathbf{I} + \mathcal{P}_r(\mathcal{F}) + \mathcal{P}_n(\mathcal{G})$

first order methods. In contrast, the DDN allows us to separate the DNN mappings from this optimization. For such challenging problems we doubt whether a similar complexity physics-free solution would be competitive.

4 Experiments

We designed our experiments to address the following questions:

- *Does the proposed deep decomposition learning help the neural network produce superior results on different inverse problem tasks? Does the DDN enjoy better generalization?*
- *Which training strategy, e.g. jointly or decoupled, is best for training the DDN? Which connection type,* e.g. *the independent or cascade one, is better?*
- *How important is each component $\mathcal{P}_r(\mathcal{F})$ and $\mathcal{P}_n(\mathcal{G})$ to DDN?*

To do that, we validate the effectiveness of our proposed decomposition learning on the inverse problems of compressive sensing MR fingerprinting (CS-MRF) [27,10] reconstruction and natural images super-resolution. While our focus is mostly on CS-MRF, the SR experiments are included to shed some light on the ablation analysis.

4.1 Implementation

Our goal here is not to explore the potential designs of the neural networks, $\mathcal{F}$ and $\mathcal{G}$, but the usage of physics in the neural network. Therefore, we use a simple four layers CNN as $\mathcal{F}$ and directly apply UNet [40], which is commonly used in the field of image reconstruction [20,16,47], as $\mathcal{G}$. In particular, all the layers in $\mathcal{F}$ are with 3×3 kernels where the first three layers undergo ReLU activation and the 2nd and 3rd layers were followed by batch normalization and ReLU activation, and a fixed number of c, 64, 64, c feature maps for each layer where c is the number of input channels. We compare the proposed *decomposition learning* (DDN) with *residual learning* (ResUnet, [20]), *unrolled learning* (NPGD, [31]) and *nullspace learning* (NSN, [32,41]). The above different mechanisms are summarized in Table 1. We build NPGD with a fixed 3 recurrent steps [31]. It is important to note that we keep the base neural network topology and the number of parameters the same for both our DDN and the competitors.

We quantify the image reconstruction quality using Normalized MSE (NMSE), Peak Signal to Noise Ratio (PSNR), *i.e.* $\text{PSNR} = 10 \log_{10}(255/\text{MSE})^2$, and generalization error (GE)-the difference between the expected loss and the training

loss, is evaluated by $\text{GE}(\mathcal{A}) = |\ell(\mathcal{A}, \mathcal{X}_{\text{test}}) - \ell(\mathcal{A}, \mathcal{X}_{\text{train}})|$ where $\ell(\mathcal{A}, \mathcal{X})$ denotes the loss evaluated over the data $\mathcal{X}$. We used ADAM to optimize the DDN and tuned the λ_1 and λ_2 for specific inverse problems. All networks used $\mathbf{H}^{\dagger}\mathbf{y}_{\epsilon}$ as the input, were implemented in PyTorch, and trained on NVIDIA 2080Ti GPUs.

4.2   CS-MRF reconstruction

Magnetic Resonance Fingerprinting (MRF) [27] recently emerged to accelerate the acquisition of tissues' quantitative MR characteristics include T_1, T_2, off-resonance frequency, etc. The compressive sensing MRF (CS-MRF) [10] is typically involves two goals, reconstructing a reasonable MR fingerprint (multi-coil sensitivities, *i.e.* MR images sequence) $\mathbf{x}$ from its undersampled k-space measurement $\mathbf{y}_{\epsilon}$, and at the same time using pattern recognition techniques such as dictionary matching [27] or neural networks [13,6,14], to query the corresponding quantitative map $\mathbf{m}$, such that achieve quantification of MRI. Accordingly, the data acquisition of CS-MRF can be boiled down to $\mathbf{y}_{\epsilon} = \mathbf{H}(\mathbf{x}) + \epsilon$ where the forward operator $\mathbf{H}$ consists a Fourier transformation $\mathcal{F}$ followed by a per-frame and time-varying subsampling operator $\mathcal{S}$, *i.e.* $\mathbf{H} = \mathcal{S} \circ \mathcal{F}$. For the simplicity, we use the Cartesian sampling as example and now $\mathbf{H}^{\dagger} = \mathcal{F}^{-1} \circ \mathcal{S}^{\top}$.

Note that $\mathbf{x}$ can be generated by the Bloch equation using its tissue property/parameters map $\mathbf{m}$, thus predicting $\mathbf{m}$ from $\mathbf{x}$ is another inverse problem and is nonlinear. Recent studies show that this nonlinear inverse problem $\mathbf{x} \rightarrow \mathbf{m}$ can be effectively solved by neural networks [13,6,14]. Here we mainly focus on the inverse problem $\mathbf{y}_{\epsilon} \rightarrow \mathbf{x}$, *i.e.* recover $\mathbf{x}$ from its noisy measurement $\mathbf{y}_{\epsilon}$. We then use the UNet-based network [13] that pre-trained on the clean oracle MRF images to predict the tissue property map $\mathbf{m}$ for a given MRF image $\mathbf{x}$.

Dataset. We use the simulated human brain MRF dataset in [6]. The ground truth parametric tissue property maps are collected from 8 volunteers (2k slices) using MAGIC [29] quantitative MRI protocol with Cartesian sampling. These parametric maps are then used to simulate MRF acquisition using the Fast imaging with Steady State Precession (FISP) [19] protocol and Cartesian sampling. In particular, we set the dimension reduced 10 channel MRF image $\mathbf{x}$ with scale $D = 200 \times 200$, the compression ratio is $d/D = 1/40$, and the noise level $\sigma_{\epsilon} = 0.01$. In order to evaluate the generalization ability of different methods under the extreme case that *scarce* labelled anatomical quantitative MRI datasets that are available for training, small numbers of samples from the first 7 patients are randomly picked for training, *e.g.* $N = 100, 50, 10$ in our experiments and 20 slices from the last volunteer are picked for the test.

Comparison results Table 2 reports the inferring time (IT), NMSE and GE for the reconstruction of MRF images and the PSNRs for the tissue maps prediction results. Due to space limitation, we only list the PSNR results of T_1 here. Fig.2a and Fig.2b show the reconstruction of a test MRF image $\mathbf{x}$'s first channel and $\mathbf{x}$'s T_1 map prediction, respectively. In summary, we make three key findings.

Table 2: Comparison results on CS-MRF reconstruction task.

	IT (sec)	X (NMSE$\times10^{-2}$/GE)			T_1 (PSNR dB)		
		$N=100$	$N=50$	$N=10$	$N=100$	$N=50$	$N=10$
ResUnet[20]	0.005	0.62/0.027	0.743/0.069	1.119/0.164	17.10 ± 2.64	17.80 ± 1.93	15.18 ± 1.49
NPGD[31]	0.15	0.58/0.024	0.980/0.118	1.689/0.255	18.05 ± 2.02	15.49 ± 1.47	13.67 ± 1.31
NSN[41]	0.04	0.57/0.031	0.740/0.068	1.119/0.159	18.49 ± 2.39	18.03 ± 1.65	15.25 ± 1.53
DDN	0.06	0.51/0.019	0.711/0.062	1.056/0.139	18.92 ± 2.04	18.06 ± 1.54	15.90 ± 1.49

First, DDN obtained the best reconstruction results and the best generalization ability in all cases (Table 2). When training with fewer training samples *e.g.* 50 and 10, NSN and NPGD perform better than NPGD. This demonstrates that a neural network that improperly-trained using a few samples is not suitable as a proximal operator. ResUnet is more stable than NPGD because the former does not need to be used in another optimization algorithm which is irrelevant to the neural network training.

Second, let τ denotes the number of times a particular model accesses the physical model (*e.g.* $\mathbf{H}$, $\mathbf{H}^\dagger$). Clearly, $\tau_{\text{ResUnet}} = 0$, $\tau_{\text{NSN}} = 2$, $\tau_{\text{DDN}} = 2\times\tau_{\text{NSN}} = 4$ and $\tau_{\text{NPGD}} = 2\times3 = 6$ since NPGD needs to access both $\mathbf{H}$ and $\mathbf{H}^\top$ in a proximal gradient descent iteration. As can be seen, the inferring time of ResUnet is very short due to its essence of the pure physics-free model. Although NPGD requires longer inferring time (2.5 times DDN), it only performs well when training using 100 samples, *i.e.* fewer samples are not sufficient to train a good neural proximal operator [38,31]. This also shows the usage of physics $\mathcal{P}_r$ and $\mathcal{P}_n$ in the proposed decomposition learning is reasonable since it obtained best results within an acceptable inferring time.

Finally, when the ResUnet, NPGD and NSN are trained with 50 or 10 samples, there exist lots of artifacts in their reconstructions (see Fig.2a and Fig.2b). In contrast, DDN can stably output reasonable MRF image reconstructions and such that lead to a physics plausible prediction of T_1 map. This shows the use of decomposition learning can help improve the generalization ability of the neural network and alleviate the dependence of deep learning performance on large-volume training samples.

4.3   Ablation study

We are interested in studying (1) the impact of different training strategies (jointly/decoupled) and different connection types (independent/cascade) to DDN and (2) the importance of component $\mathcal{P}_r$ and $\mathcal{P}_n$ to DDN. To do that, we consider the Super-resolution (SR) task and train the models on the BSDS300 [1] and test on the benchmark dataset Set5[2]. We super-resolve low-resolution (LR) images by a scale factor of 2. The LR images are formed by downsampling with bi-cubic anti-aliasing the HR images [12], and then corrupting them with Gaussian noise with a standard deviation $\sigma_\epsilon = 0.1$ (10% noise). The presence of anti-aliasing in
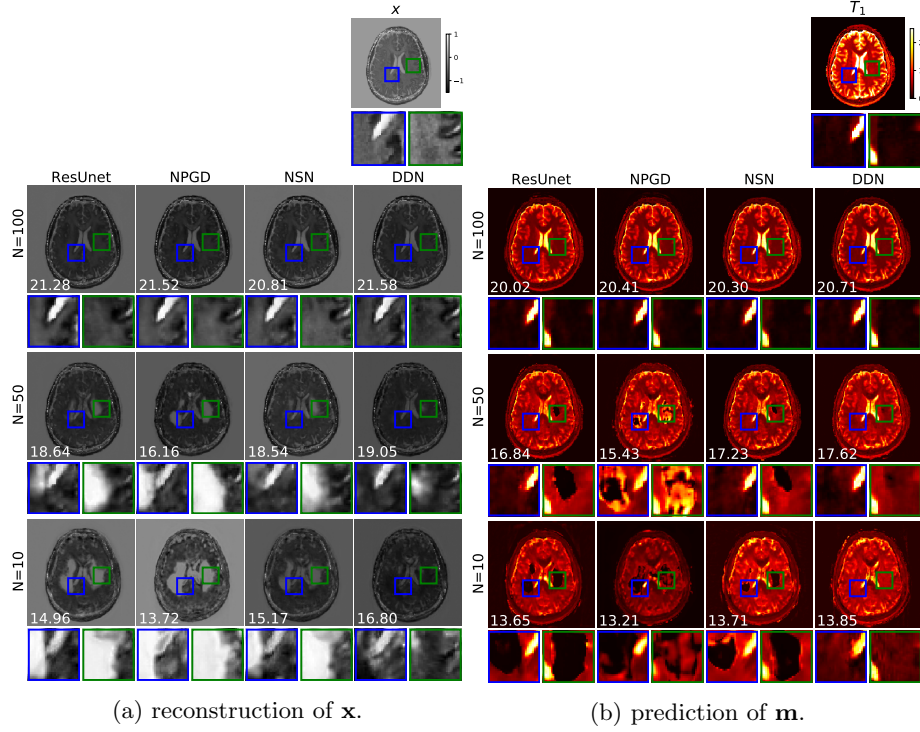
(a) reconstruction of **x**.   (b) prediction of **m**.

Fig. 2: (a) Reconstruction of MRF image **x** from its noisy measurement $\mathbf{y}_\epsilon$ ($\sigma_\epsilon = 0.01$) and (b) its corresponding T_1 map prediction. From top to bottom are the groundtruth x and m and their corresponding reconstructions by ResUnet [20], NPGD[31], NSN[41] and DDN (proposed) trained with 100, 50 and 10 sample, respectively.

the downsampling means that the bi-cubic upsampling operator is a reasonable approximation to $\mathbf{H}^\dagger$, and this is what we use in our DDN. Training is performed using 40×40 RGB image patches (LR). The quantitative results presented are then evaluated on the luminance (Y) channel. We *emphasize* that our goal here is not to achieve state-of-the-art performance on the SR task, but a simple scenario to study the behaviour of ablation analysis for decomposition learning.

Training strategy study We trained DDN using different training strategies. The test results are reported in the Table 3 which demonstrates the cascade architecture always performs slightly better than the independent one in term of both PSNR and GE. Note that for the independent model the joint and decoupled training are exactly equivalent. We also find that $\mathcal{F}$ and $\mathcal{G}$ benefit from more iterations in the decoupled training, especially for very noisy cases. In contrast, joint training was observed to be more efficient. In all experiments, we therefore use the cascade architecture with joint training for the DDNs.

Table 3: Comparison results of ablation study on image SR task ($\times 2$, $\sigma_\epsilon = 0.1$).

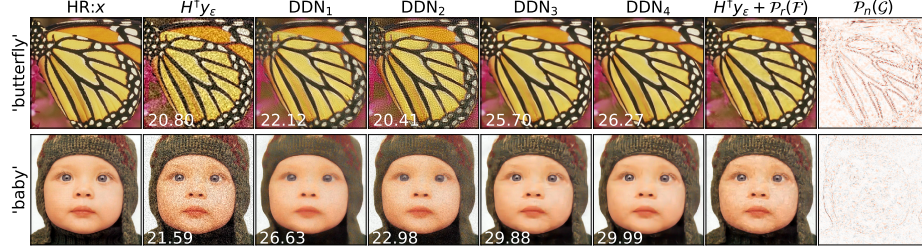| Metric	Training strategy study			Ablation study			
	Independent Joint/Decoupled	Cascade Joint	Decoupled	DDN_1	DDN_2	DDN_3	DDN_4
PSNR (dB)	26.15	26.30	26.38	22.31	20.95	25.82	26.30
GE (10^{-3})	1.24	1.26	1.24	1.75	1.41	1.83	1.26

Fig. 3: Visualization of DDN with different specs on the SR task ($\times 2$, $\sigma_\epsilon = 0.1$). From the left to right are the oracle HR image $\mathbf{x}$, reconstructions by $\mathbf{H}^\dagger \mathbf{y}_\epsilon$, $\text{DDN}_1(\boxtimes\mathcal{P}_r, \boxtimes\mathcal{P}_n)$, $\text{DDN}_2(\boxtimes\mathcal{P}_r, \boxdot\mathcal{P}_n)$, $\text{DDN}_3(\boxdot\mathcal{P}_r, \boxtimes\mathcal{P}_n)$, $\text{DDN}_4(\boxdot\mathcal{P}_r, \boxdot\mathcal{P}_n)$, and the r-component $\mathbf{H}^\dagger \mathbf{y}_\epsilon + \mathcal{P}_r(\mathcal{F})$ and n-component $\mathcal{P}_n(\mathcal{G})$ recovered by DDN_4.

Importance of $\mathcal{P}_r$ and $\mathcal{P}_n$ We trained four DDN variants by different usages of $\mathcal{P}_r$ and $\mathcal{P}_n$: DDN_1 - deactivate both $\mathcal{P}_r$ and $\mathcal{P}_n$, DDN_2 - only use $\mathcal{P}_n$, DDN_3 - only use $\mathcal{P}_r$ and DDN_4 - use both $\mathcal{P}_r$ and $\mathcal{P}_n$. The statistics PSNRs and GEs of the reconstruction are listed in Table 3. Two reconstruction examples and the learnt r-component and n-component by DDN_4 are shown in Fig.3. From the results, we have the following conclusions.

First, it can be seen that the DDN_4 achieves the best results which used both projectors $\mathcal{P}_r$ and $\mathcal{P}_n$. DDN_2 is essentially a nullspace network and performed poorly in the presence of noise, as in this scenario the denoising task plays a significant role in the inverse problem. DDN_1 is equivalent to residual learning and is purely learning-based and physics-free, it consistently provides better performance than DDN_2, but it has the higher GEs. This suggests that introducing physics into deep learning models can facilitate the neural network to enjoy better generalization for solving inverse problems.

Second, DDN_3 performs better than DDN_1 and DDN_2 but worse than DDN_4. It is because DDN_3 considers denoising by recovering the r-component of data but fails to learn the n-component. This demonstrates integrating $\mathcal{P}_r$ and $\mathcal{P}_n$ into the proposed decomposition learning allows the DDN to remove structural noise and to simultaneously accurately approximate the r-component $\mathcal{P}_r(\mathbf{x})$ (see the penultimate column in Fig.3) and predict the n-component $\mathcal{P}_n(\mathbf{x})$ (see the last column in Fig.3) from its noisy measurement. From the above discussions, we conclude that decomposition learning is well-principled, structurally simple, and highly interpretable.

Finally, we emphasize that given a linear forward operator $\mathbf{H}$, the decomposition learning naturally exists and is easy to define. One can plug the decomposition learning, into other existing specialized deep learning solvers for different inverse problems, with which we believe one could increase the performance limit of the deep learning solvers.

5   Conclusion

In this paper, we have proposed a deep decomposition learning framework for building an end-to-end and physics-engaged neural network solution for inverse problems. We have explicitly reformulated the neural network layers to learn range-nullspace decomposition functions with reference to the layer inputs, instead of learning unreferenced functions. We have shown that the decomposition networks not only produce superior results, but also enjoy good interpretability and generalization. We have demonstrated the advantages of decomposition learning on CS-MRF and image super-resolution examples. In future work, we will explore adapting the proposed deep decomposition learning to more challenging inverse problems such as tomographic imaging.

Acknowledgements

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 126–135 (2017)
2. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding (2012)
3. Bora, A., Jalal, A., Price, E., Dimakis, A.G.: Compressed sensing using generative models. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 537–546. JMLR. org (2017)
4. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning **3**(1), 1–122 (2011)
5. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2392–2399. IEEE (2012)
6. Chen, D., Golbabaee, M., Gómez, P.A., Menzel, M.I., Davies, M.E.: Deep fully convolutional network for mr fingerprinting. In: International Conference on Medical Imaging with Deep Learning (2019)
7. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1256–1272 (2016)
8. Cui, Z., Chang, H., Shan, S., Zhong, B., Chen, X.: Deep network cascade for image super-resolution. In: European Conference on Computer Vision. pp. 49–64. Springer (2014)
9. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences **57**(11), 1413–1457 (2004)
10. Davies, M., Puy, G., Vandergheynst, P., Wiaux, Y.: A compressed sensing framework for magnetic resonance fingerprinting. SIAM Journal on Imaging Sciences **7**(4), 2623–2656 (2014)
11. Davis, G., Greg, O., Rebecca, W.: Neumann networks for linear inverse problems in imaging. IEEE Transactions on Computational Imaging (2019)
12. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
13. Fang, Z., Chen, Y., Liu, M., Xiang, L., Zhang, Q., Wang, Q., Lin, W., Shen, D.: Deep learning for fast and spatially-constrained tissue quantification from highly-accelerated data in magnetic resonance fingerprinting. IEEE transactions on medical imaging (2019)
14. Golbabaee, M., Chen, D., Gómez, P.A., Menzel, M.I., Davies, M.E.: Geometry of deep learning for magnetic resonance fingerprinting. In: the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (2019)
15. Gu, S., Timofte, R., Van Gool, L.: Integrating local and non-local denoiser priors for image restoration. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 2923–2928. IEEE (2018)
16. Gupta, H., Jin, K.H., Nguyen, H.Q., McCann, M.T., Unser, M.: Cnn-based projected gradient descent for consistent ct image reconstruction. IEEE transactions on medical imaging **37**(6), 1440–1453 (2018)

17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
18. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
19. Jiang, Y., Ma, D., Seiberlich, N., Gulani, V., Griswold, M.A.: Mr fingerprinting using fast imaging with steady state precession (fisp) with spiral readout. Magnetic resonance in medicine **74**(6), 1621–1631 (2015)
20. Jin, K.H., McCann, M.T., Froustey, E., Unser, M.: Deep convolutional neural network for inverse problems in imaging. IEEE Transactions on Image Processing **26**(9), 4509–4522 (2017)
21. Kawaguchi, K., Kaelbling, L.P., Bengio, Y.: Generalization in deep learning. arXiv preprint arXiv:1710.05468 (2017)
22. Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., Ashok, A.: Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 449–458 (2016)
23. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8183–8192 (2018)
24. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4681–4690 (2017)
25. Lucas, A., Iliadis, M., Molina, R., Katsaggelos, A.K.: Using deep neural networks for inverse problems in imaging: beyond analytical methods. IEEE Signal Processing Magazine **35**(1), 20–36 (2018)
26. Lunz, S., Öktem, O., Schönlieb, C.B.: Adversarial regularizers in inverse problems. In: Advances in Neural Information Processing Systems. pp. 8507–8516 (2018)
27. Ma, D., Gulani, V., Seiberlich, N., Liu, K., Sunshine, J.L., Duerk, J.L., Griswold, M.A.: Magnetic resonance fingerprinting. Nature **495**(7440), 187–192 (2013)
28. Mao, X., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Advances in neural information processing systems. pp. 2802–2810 (2016)
29. Marcel, W., AB, S.: New technology allows multiple image contrasts in a single scan. GESIGNAPULSE.COM/MR **SPRING**, 6–10 (2015)
30. Mardani, M., Gong, E., Cheng, J.Y., Vasanawala, S., Zaharchuk, G., Xing, L., Pauly, J.M.: Deep generative adversarial networks for compressed sensing automates mri. IEEE Transactions on Medical Imaging **38**(1), 167–179 (2019)
31. Mardani, M., Sun, Q., Donoho, D., Papyan, V., Monajemi, H., Vasanawala, S., Pauly, J.: Neural proximal gradient descent for compressive imaging. In: Advances in Neural Information Processing Systems. pp. 9573–9583 (2018)
32. Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1781–1790 (2017)
33. Metzler, C., Mousavi, A., Baraniuk, R.: Learned d-amp: Principled neural network based compressive image recovery. In: Advances in Neural Information Processing Systems. pp. 1772–1783 (2017)

34. Mousavi, A., Patel, A.B., Baraniuk, R.G.: A deep learning approach to structured signal recovery. In: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton). pp. 1336–1343. IEEE (2015)
35. Pan, J., Liu, S., Sun, D., Zhang, J., Liu, Y., Ren, J., Li, Z., Tang, J., Lu, H., Tai, Y.W., et al.: Learning dual convolutional neural networks for low-level vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3070–3079 (2018)
36. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
37. Quan, T.M., Nguyen-Duc, T., Jeong, W.K.: Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss. IEEE transactions on medical imaging **37**(6), 1488–1497 (2018)
38. Rick Chang, J., Li, C.L., Poczos, B., Vijaya Kumar, B., Sankaranarayanan, A.C.: One network to solve them all–solving linear inverse problems using deep projection models. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5888–5897 (2017)
39. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: Regularization by denoising (red). SIAM Journal on Imaging Sciences **10**(4), 1804–1844 (2017)
40. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
41. Schwab, J., Antholzer, S., Haltmeier, M.: Deep null space learning for inverse problems: Convergence analysis and rates. Inverse Problems (2019)
42. Shah, V., Hegde, C.: Solving linear inverse problems using gan priors: An algorithm with provable guarantees. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4609–4613. IEEE (2018)
43. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4570–4580 (2019)
44. Sønderby, C.K., Caballero, J., Theis, L., Shi, W., Huszár, F.: Amortised map inference for image super-resolution. arXiv preprint arXiv:1610.04490 (2016)
45. Srivastava, R.K., Greff, K., Schmidhuber, J.: Training very deep networks. In: Advances in neural information processing systems. pp. 2377–2385 (2015)
46. Sun, J., Li, H., Xu, Z., et al.: Deep admm-net for compressive sensing mri. In: Advances in neural information processing systems. pp. 10–18 (2016)
47. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9446–9454 (2018)
48. Xie, J., Xu, L., Chen, E.: Image denoising and inpainting with deep neural networks. In: Advances in neural information processing systems. pp. 341–349 (2012)
49. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Advances in neural information processing systems. pp. 1790–1798 (2014)
50. Zeng, K., Yu, J., Wang, R., Li, C., Tao, D.: Coupled deep autoencoder for single image super-resolution. IEEE transactions on cybernetics **47**(1), 27–37 (2015)
51. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)