



**HAL**  
open science

# Deep learning for a space-variant deconvolution in galaxy surveys

Florent Sureau, Alexis Lechat, Jean-Luc Starck

► **To cite this version:**

Florent Sureau, Alexis Lechat, Jean-Luc Starck. Deep learning for a space-variant deconvolution in galaxy surveys. *Astronomy and Astrophysics - A&A*, 2020, 641, pp.A67. 10.1051/0004-6361/201937039 . cea-02936950

**HAL Id: cea-02936950**

**<https://cea.hal.science/cea-02936950>**

Submitted on 11 Sep 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Deep learning for a space-variant deconvolution in galaxy surveys<sup>★</sup>

F. Sureau<sup>1</sup>, A. Lechat<sup>1,2</sup>, and J.-L. Starck<sup>1</sup>

<sup>1</sup> Laboratoire AIM, CEA, CNRS, Université Paris-Saclay, Université Paris Diderot, Sorbonne Paris Cité, 91191 Gif-sur-Yvette, France  
e-mail: florent.sureau@cea.fr

<sup>2</sup> ONERA – The French Aerospace Lab, 6 chemin de la Vauve aux Granges, BP 80100, 91123 PALAISEAU cedex, France

Received 1 November 2019 / Accepted 21 June 2020

## ABSTRACT

The deconvolution of large survey images with millions of galaxies requires developing a new generation of methods that can take a space-variant point spread function into account. These methods have also to be accurate and fast. We investigate how deep learning might be used to perform this task. We employed a U-net deep neural network architecture to learn parameters that were adapted for galaxy image processing in a supervised setting and studied two deconvolution strategies. The first approach is a post-processing of a mere Tikhonov deconvolution with closed-form solution, and the second approach is an iterative deconvolution framework based on the alternating direction method of multipliers (ADMM). Our numerical results based on GREAT3 simulations with realistic galaxy images and point spread functions show that our two approaches outperform standard techniques that are based on convex optimization, whether assessed in galaxy image reconstruction or shape recovery. The approach based on a Tikhonov deconvolution leads to the most accurate results, except for ellipticity errors at high signal-to-noise ratio. The ADMM approach performs slightly better in this case. Considering that the Tikhonov approach is also more computation-time efficient in processing a large number of galaxies, we recommend this approach in this scenario.

**Key words.** methods: statistical – methods: data analysis – methods: numerical

## 1. Introduction

The deconvolution of large galaxy survey images requires that the spatial variation of the point spread function (PSF) across the field of view is taken into account. The PSF field is usually estimated beforehand through parametric models and simulations as in [Krist et al. \(2011\)](#) or is directly estimated from the (noisy) observations of stars in the field of view ([Bertin 2011](#); [Kuijken et al. 2015](#); [Zuntz et al. 2018](#); [Mboula et al. 2016](#); [Schmitz et al. 2020](#)). Even when the PSF is known perfectly, this ill-posed deconvolution problem is challenging, in particular because of the size of the image that is to be processed. [Starck et al. \(2000\)](#) proposed an object-oriented deconvolution that consists of first detecting galaxies and then deconvolving each object independently, taking the PSF at the position of the center of the galaxy into account (but not the variation in the PSF field at the galaxy scale). Following this idea, [Farrens et al. \(2017\)](#) introduced a space-variant deconvolution approach for galaxy images that is based on two regularization strategies: using either a sparse prior in a transformed domain ([Starck et al. 2015a](#)), or trying to learn without supervision a low-dimensional subspace for galaxy representation using a low-rank prior on the recovered galaxy images. When a sufficient number of galaxies were processed jointly (more than 1000), the authors found that the low-rank approach provided significantly lower ellipticity errors than sparsity. This illustrates the importance of learning adequate representations for galaxies. To proceed in learn-

ing, supervised deep-learning techniques that use databases of galaxy images might be employed to learn complex mappings that might regularize our deconvolution problem. Deep convolutional architectures have also proved to be computationally efficient in processing large numbers of images when the model has been learned. They are therefore promising in the context of modern galaxy surveys.

In recent years, deep-learning approaches have been proposed in a large number of inverse problems with high empirical success. This might be explained by the expressivity of the deep architectures as shown in the theoretical works for simple architecture in [Eldan & Shamir \(2015\)](#), [Safran & Shamir \(2017\)](#), and [Petersen & Voigtlaender \(2018\)](#). New architectures or new optimization strategies have been proposed as well to increase the learning performance (e.g., [Kingma & Ba 2014](#); [Ioffe & Szegedy 2015](#); [He et al. 2016](#); [Szegedy et al. 2016](#)). Their success also depends on the huge datasets that were collected in the different applications to train the networks, and on the increased computing power available to process them.

We investigate here two different strategies to interface deep-learning techniques with space-variant deconvolution approaches inspired by convex optimization. In Sect. 2 we review deconvolution techniques based on convex optimization and deep-learning schemes. The space-variant deconvolution is presented in Sect. 3, where the two proposed methods are described. The first method uses a deep neural network (DNN) to post-process a Tikhonov deconvolution, and the second includes a DNN that is trained to denoise in an iterative algorithm derived from convex optimization. The neural network architecture proposed for deconvolution is also presented in Sect. 3. The experiment settings are described in Sect. 4 and the results are presented in Sect. 5. We conclude in Sect. 6.

<sup>★</sup> In the spirit of reproducible research, the codes will be made freely available on the CosmoStat website ([www.cosmostat.org](http://www.cosmostat.org)). The testing datasets will also be provided to repeat the experiments performed in this paper.

## 2. Image deconvolution in the deep-learning era

### 2.1. Deconvolution before deep learning

The standard deconvolution problem consists of solving the linear inverse problem  $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}$ , where  $\mathbf{Y}$  is the observed noisy data,  $\mathbf{X}$  is the unknown solution,  $\mathbf{H}$  is the matrix related to the PSF, and  $\mathbf{N}$  is the noise. Images  $\mathbf{Y}$ ,  $\mathbf{X}$ , and  $\mathbf{N}$  are represented by a column vector of  $n_p$  pixels arranged in lexicographic order, with  $n_p$  being the total number of pixels, and  $\mathbf{H}$  is a  $n_p \times n_p$  matrix. Modern deconvolution techniques typically solve this ill-posed inverse problem (i.e., without a unique and stable solution) by modeling the forward problem based on physics and adding the regularization penalty term  $\mathcal{R}(\mathbf{X})$ , which can be interpreted as enforcing some constraints on the solution. This leads to the minimization

$$\arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \mathcal{R}(\mathbf{X}), \quad (1)$$

where  $\|\cdot\|_F$  is the Frobenius norm. The simplest (and oldest) corresponding regularization is the Tikhonov regularization (Tikhonov et al. 1977; Hunt 1972; Twomey 1963), where  $\mathcal{R}(\mathbf{X})$  is a quadratic term,  $\mathcal{R}(\mathbf{X}) = \frac{\lambda}{2} \|\mathbf{L}\mathbf{X}\|_F^2$ . The closed-form solution of this inverse problem is given by

$$\tilde{\mathbf{X}} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{L}^T \mathbf{L})^{-1} \mathbf{H}^T \mathbf{Y}, \quad (2)$$

which involves the Tikhonov linear filter  $(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{L}^T \mathbf{L})^{-1} \mathbf{H}^T$ . The simplest version is when  $\mathbf{L} = \mathbf{Id}$ , which penalizes solutions with high energy. When the PSF is space invariant, the matrix  $\mathbf{H}$  is block circulant, and the inverse problem can then be written as a simple convolution product. It is also easy to see that the Wiener deconvolution corresponds to a specific case of the Tikhonov filter; see Bertero & Boccacci (1998) for more details.

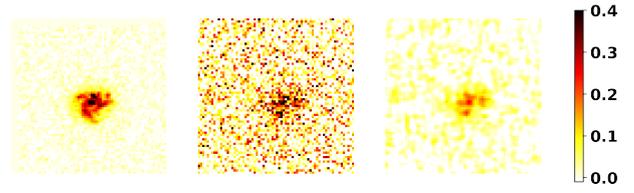
This rather crude deconvolution is illustrated in Fig. 1 in a scenario with a low signal-to-noise ratio (S/N). It shows the oversmoothing of the galaxy image, loss of energy in the recovered galaxy, and colored noise that is due to the inverse filter.

Most advanced methods are nonlinear and generally involve iterative algorithms. The literature on image processing in advanced regularization techniques applied to deconvolution is copious: adding some prior information on  $\mathbf{X}$  in a Bayesian paradigm (Bioucas-Dias 2006; Krishnan et al. 2009; Orioux et al. 2010) or assuming  $\mathbf{X}$  to belong to some classes of images that are to be recovered (e.g., using total variation regularization (Oliveira et al. 2009; Cai et al. 2010), sparsity in fixed representations (Starck et al. 2003; Pesquet et al. 2009; Pustelnik et al. 2016) or learned through dictionary learning Mairal et al. 2008; Lou et al. 2011; Jia & Evans 2011), by constraining the solution to belong to some convex subsets (such as ensuring that the final galaxy image is positive).

For instance, a very efficient approach for galaxy image deconvolution is based on sparse recovery, which consists of minimizing

$$\arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_2^2 + \lambda \|\Phi^T \mathbf{X}\|_1, \quad (3)$$

where  $\Phi$  is a matrix related to a fixed transform (Fourier, wavelet, curvelets, etc.) or that can be learned from the data or a training dataset (Starck et al. 2015b). The  $\ell_1$  norm in the regularization term is known to reinforce the sparsity of the solution; see Starck et al. (2015b) for a review of sparsity. Sparsity



**Fig. 1.** Deconvolution with Tikhonov regularization. *From left to right*, we show the galaxy image from the HST that we used for the simulation, the observed galaxy at  $S/N$  20 (see below for our definition of  $S/N$ ), and the deconvolved image computed from Eq. (2).

was found to be extremely efficient for different inverse problems in astrophysics, such as an estimation of the cosmic microwave background (CMB; Bobin et al. 2014), estimation of compact sources in CMB missions (Sureau et al. 2014), recovery of a weak-lensing map (Lanusse et al. 2016), or reconstruction of a radio-interferometry image (Garsden et al. 2015). We compare our deconvolution techniques with this sparse-deconvolution approach here.

Iterative convex optimization techniques have been devised to solve Eq. (3) (see, e.g., Beck & Teboulle 2009; Zibulevsky & Elad 2010; Combettes et al. 2011; Chambolle & Pock 2011; Afonso et al. 2011; Condat 2013; Combettes & Vu 2014), with well-studied convergence properties, but with a high computing cost when adaptive representation is used for galaxies. This problem opens the way to a new generation of methods.

### 2.2. Toward deep learning

Recently, deep-learning techniques have been proposed to solve inverse problems using the collected dataset and/or the advances in simulations, and have been applied for the deconvolution of galaxy images. These approaches have proved to be able to learn complex mappings in the supervised setting, and to be computationally efficient when the model has been learned. Without being exhaustive, we review here some recent work on deconvolution using DNNs. We identified three different strategies for using DNN in a deconvolution problem.

The inverse convolution filter can be directly approximated using convolutional neural networks (Xu et al. 2014; Schuler et al. 2016). In our application with space-variant deconvolution and known kernels, a complicated blind deconvolution like this is clearly not necessary and would require a large amount of data to try learning information that is already provided by the physics included in the forward model. In the early years of using sparsity for deconvolution, a two-step approach was proposed that consisted of first applying a simple linear deconvolution such as the pseudo-inverse or the Tikhonov filter, which allows noise to enter the solution, and in the second step, applying a sparse denoising (see the wavelet-vaguelette decomposition; Donoho 1995; Kalifa et al. 2003, more general regularization; Guerrero-Colon & Portilla 2006, or the forward method; Neelamani et al. 2004). Similarly, the second step has been replaced by denoising or removing artifacts using a multilayer perceptron (Schuler et al. 2013), or more recently, using U-nets (Jin et al. 2017). CNNs are well adapted to this task because the form of a CNN mimics unrolled iterative approaches when the forward model is a convolution. In another application, convolutional networks such as deep convolutional framelets have also been applied to remove artifacts from reconstructed CT images (Ye et al. 2018a). One advantage of this decoupling approach is the ability to quickly process a large amount of

data when the network has been learned if the chosen deconvolution has a closed-form expression. The third strategy uses iterative approaches that are often derived from convex optimization coupled with deep-learning networks. Several schemes have been devised to solve generic inverse problems. The first option, called unrolling or unfolding (see [Monga et al. \(2019\)](#) for a detailed review), is to mimic a few iterations of an iterative algorithm with DNNs so as to capture in the learning phase the effect of (1) the prior ([Mardani et al. 2017](#)), (2) the hyperparameters ([Mardani et al. 2017](#); [Adler & Öktem 2017, 2018](#); [Bertocchi et al. 2018](#)), (3) the updating step of a gradient descent ([Adler & Öktem 2017](#)), or (4) the whole update process ([Gregor & LeCun 2010](#); [Adler & Öktem 2018](#); [Mardani et al. 2018](#)). These approaches allow a fast approximation of iterative algorithms ([Gregor & LeCun 2010](#)), a better hyperparameter selection ([Bertocchi et al. 2018](#)), and/or provide new algorithms in a supervised way ([Adler & Öktem 2017](#); [Mardani et al. 2018](#); [Adler & Öktem 2018](#)) that are better adapted to processing a specific dataset. This approach has noticeably been used recently for blind deconvolution ([Li et al. 2019](#)). Finally, an alternative is using iterative proximal algorithms from convex optimization (e.g., in the framework of the alternating direction method of multiplier plug&play (ADMM PnP; [Venkatakrisnan et al. 2013](#); [Sreehari et al. 2016](#); [Chan et al. 2017](#)), or regularization by denoising; [Romano et al. 2017](#); [Reehorst & Schniter 2018](#)), where the proximity operator related to the prior is replaced by a DNN ([Meinhardt et al. 2017](#); [Bigdeli et al. 2017](#); [Gupta et al. 2018](#)) or by a series of DNN that are trained in different denoising settings, as in [Zhang et al. \(2017\)](#).

The last two strategies are therefore more adapted to the problem we target here, and we investigate below how they can be applied and how they perform compared to modern methods in the space-variant deconvolution of galaxy images.

### 2.3. Deep deconvolution and sparsity

It is interesting to note that connections exist between sparse recovery methodology and DNN. The first features learned in convolutive deep neural networks typically correspond to edges at particular orientation and location in the images ([LeCun et al. 2015](#)), which is also what the wavelet transforms extract at different scales. Similar observations were noted for features learned with a CNN in the context of cosmological parameter estimations from weak-lensing convergence maps ([Ribli et al. 2019](#)). Understanding mathematically how the architecture of such networks captures progressively powerful invariants can also be approached through wavelets and their use in the wavelet-scattering transform ([Mallat 2016](#)). [Meinhardt et al. \(2017\)](#) has shown that using a denoising neural network instead of a proximal operator (e.g., soft-thresholding in wavelet space in sparse recovery) during the minimization iterations improves the deconvolution performance. They also claimed that the noise level used to train the neural network behaves like the regularization parameter in sparse recovery. The convergence of the algorithm is no longer guaranteed, but the authors observed experimentally that their algorithms stabilized, and they expressed their fixed points. Finally, the two parts of U-nets are very similar to synthesis and analysis concepts in sparse representations. This has motivated the use of wavelets to implement average pooling and unpooling in the expanding path in U-nets ([Ye et al. 2018a](#); [Han & Ye 2018](#)). Some other connection can be made with the soft-autoencoder in [Fan et al. \(2018\)](#), which introduces a pair of ReLU units that emulate soft-thresholding. This accen-

tuates the comparison with cascade-wavelet-shrinkage systems. We thus observe exchanges between the two fields, in particular for U-net architectures, with significant differences, however, such as the construction of a very rich dictionary in U-nets that is possible through the use of a large training dataset, and nonlinearities at every layer that are essential to capture invariants in the learning phase.

### 3. Image deconvolution with space-variant PSF

In the case of a space-variant deconvolution problem, we can write the same deconvolution equation as before,  $\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{N}$ , but  $\mathbf{H}$  is no longer a block circulant matrix, and manipulating such a huge matrix is not possible in practice. As in [Farrens et al. \(2017\)](#), we instead consider an object-oriented deconvolution by first detecting  $n_g$  galaxies with  $n_p$  pixels each and then independently deconvolving each object using the PSF at the position of the center of the galaxy. We use the following definitions: the observations of  $n_g$  galaxies with  $n_p$  pixels are collected in  $\mathbf{Y} \in \mathbb{R}^{n_p \times n_g}$  (as before, each galaxy is represented by a column vector arranged in lexicographic order), the galaxy images that are to be recovered are similarly collected,  $\mathbf{X} \in \mathbb{R}^{n_p \times n_g} = [\mathbf{x}_i]_{i=1\dots n_g}$ , and the convolution operator with the different kernels is noted  $\mathcal{H}$ . This corresponds to applying in parallel a convolution matrix  $\mathbf{H}_i$  to a galaxy  $\mathbf{x}_i$  ( $\mathbf{H}_i$  typically is a block-circulant matrix after zero-padding that we perform on the images [Andrews & Hunt 1977](#)). Noise is noted  $\mathbf{N} \in \mathbb{R}^{n_p \times n_g}$  as before and is assumed to be additive white Gaussian noise. With these definitions, we now have

$$\mathbf{Y} = \mathcal{H}(\mathbf{X}) + \mathbf{N}, \quad (4)$$

or more precisely,

$$\{\mathbf{y}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{n}_i\}_{i=1\dots n_g}, \quad (5)$$

for block-circulant  $\{\mathbf{H}_i\}_{i=1\dots n_g}$ , which illustrates that we consider multiple local space-invariant convolutions in our model (ignoring the very small variations in PSF at the scale of the galaxy, as done in practice [Kuijken et al. 2015](#); [Mandelbaum et al. 2015](#); [Zuntz et al. 2018](#)). The deconvolution problem of finding  $\mathbf{X}$  knowing  $\mathbf{Y}$  and  $\mathcal{H}$  is therefore considered as a series of independent ill-posed inverse problems. To avoid multiple solutions (due to a nontrivial null space of  $\{\mathbf{H}_i\}_{i=1\dots n_g}$ ) or an unstable solution (bad conditioning of these matrices), we need to regularize the problem as in standard deconvolution approaches that were developed for space-invariant convolutions. This amounts to solving the following inverse problem:

$$\arg \min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_F^2 + \mathcal{R}(\mathbf{X}), \quad (6)$$

and in general, we choose separable regularizers so that we can handle the different deconvolution problems in parallel:

$$\left\{ \arg \min_{\mathbf{x}_i} \frac{1}{2} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{x}_i\|_2^2 + \mathcal{R}(\mathbf{x}_i) \right\}_{i=1\dots n_g}. \quad (7)$$

[Farrens et al. \(2017\)](#) proposed two methods to perform this deconvolution using either a sparse or a low-rank prior. In the former, each galaxy is assumed to be sparse in the wavelet domain. This leads to the minimization

$$\operatorname{argmin}_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_2^2 + \|\mathbf{W}^{(k)} \odot \Phi(\mathbf{X})\|_1 \quad \text{s.t.} \quad \mathbf{X} \geq 0, \quad (8)$$

where  $\mathbf{W}^{(k)}$  is a weighting matrix. Each galaxy is then deconvolved independently of the others. As there are many similarities between galaxy images, in the latter method [Farrens et al. \(2017\)](#) proposed a joint restoration process, where the matrix  $\mathbf{X}$  has a low rank. This is enforced by adding a nuclear norm penalization instead of the sparse regularization, as follows:

$$\operatorname{argmin}_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_2^2 + \lambda \|\mathbf{X}\|_* \quad \text{s.t.} \quad \mathbf{X} \geq 0, \quad (9)$$

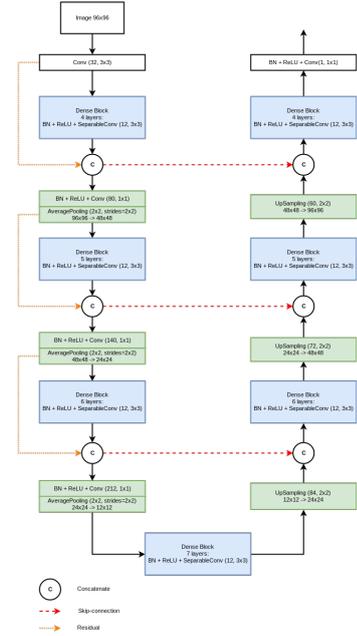
where  $\|\mathbf{X}\|_* = \sum_k \sigma_k(\mathbf{X})$ , and  $\sigma_k(\mathbf{X})$  denotes the  $k$ th largest singular value of  $\mathbf{X}$ .

It was shown that the second approach outperforms sparsity techniques as soon as the number of galaxies in the field is larger than 1000 ([Farrens et al. 2017](#)).

### 3.1. Neural network architectures

The DNN allows us to extend the previous low-rank minimization by using existing databases and learning more features from the data in a supervised way, compared to what we could do with the simple singular value decomposition used for nuclear norm penalization. The choice of network architecture is crucial for the performance. We identified three different features that are important for our application: (1) the forward model and the task imply that the network should be translation equivariant, (2) the model should include some multiscale processing based on the fact that we should be able to capture distant correlations, and (3) the model should minimize the number of trainable parameters for a given performance, so as to be efficient (lower GPU memory consumption), which is also important to facilitate the learning. These objectives are hopefully not contradictory: the first consideration leads to the use of convolutional layers, while the second implies a structure such as the U-net ([Ronneberger et al. 2015](#)), which has previously been used to solve inverse problems ([Jin et al. 2017](#)), or it leads to the deep convolutional framelets ([Ye et al. 2018a](#)). Because these architectures allow rapidly increasing the receptive field in the layers of the network, however, they can compete with their smaller number of parameters against CNNs with a larger number of layers and therefore more parameters.

We therefore selected a global U-net structure as in [Jin et al. \(2017\)](#), but included several modifications. We first replaced 2D convolutions by 2D separable convolutions ([Chollet 2016, 2017](#)). The separable convolutions allow us to limit the number of parameters in the model by assuming that spatial correlations and correlations across feature maps can be captured independently. This has previously led to configurations that outperform architectures with non-separable convolution with a larger number of parameters ([Chollet 2016, 2017](#)). We also changed the convolutional layers at each “scale” by using dense blocks ([Huang et al. 2017](#)). Dense blocks also allow us to reduce the number of parameters by propagating all prior feature maps to the input of the current layer through concatenation. This was claimed to enable feature reuse, preservation of information, and to limit vanishing gradients in the learning. We also changed the pooling step because we observed that maximum-pooling leads to oversegmentation of our final estimates. This is alleviated by the use of average-pooling. Finally, we removed the skip connection between the input and output layers introduced by [Jin et al. \(2017\)](#), which proved to be detrimental to the performance of the network, especially at low S/N. The dense blocks may also have preserved the flow of relevant information better and limited the interest in using residual learning.



**Fig. 2.** DNN model used in this work. The global architecture is a U-net, which we slightly modified for performance and to limit the number of model parameters.

The two first modification significantly limit the number of parameters per scale of the U-net and potentially allow for more scales to be used for a given budget of the number of trainable parameters. We call our network XDense U-net and display it in Fig. 2. We describe below how a network like this can be used in two different ways in order to perform the space-variant deconvolution.

### 3.2. Tikhonet: Tikhonov deconvolution post-processed by a DNN

The Tikhonov solution for the space-variant PSF deconvolution is

$$\operatorname{argmin}_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \mathcal{H}(\mathbf{X})\|_F^2 + \|\mathcal{L}(\mathbf{X})\|_F^2, \quad (10)$$

where  $\mathcal{L}$  is built similarly as  $\mathcal{H}$ . The closed-form solution of this linear inverse problem is given by

$$\left\{ \tilde{\mathbf{x}}_i = \left( \mathbf{H}_i^T \mathbf{H}_i + \lambda_i \mathbf{L}_i^T \mathbf{L}_i \right)^{-1} \mathbf{H}_i^T \mathbf{y}_i \right\}_{i=1 \dots n_g}, \quad (11)$$

which for each galaxy involves a different Tikhonov filter  $\left( \mathbf{H}_i^T \mathbf{H}_i + \lambda_i \mathbf{L}_i^T \mathbf{L}_i \right)^{-1} \mathbf{H}_i^T$ . We chose  $\mathbf{L}_i = \mathbf{Id}$ , and the regularization parameter  $\lambda_i$  was different for each galaxy, depending on its S/N (see Sect. 3.4 for more details). The final estimate is then only

$$\{\hat{\mathbf{x}}_i = \mathcal{N}_\theta(\tilde{\mathbf{x}}_i)\}_{i=1 \dots n_g}, \quad (12)$$

where the neural network predictions based on its parameters  $\theta$  for some inputs  $\mathbf{y}$  are written  $\mathcal{N}_\theta(\mathbf{y})$ .

The success of the first approach therefore lies in the supervised learning of the mapping between the Tikhonov deconvolution of Eq. (11) and the targeted galaxy. We call this two-step approach Tikhonet, and the rather simple training process is described in Algorithm 1.

---

**Algorithm 1** DNN training in the Tikhonet approach
 

---

```

1: Initialization: Prepare noise-free training set, choose noise
   parameters (S/N range) and validation set. Choose architecture
   for network  $\mathcal{N}$ , learning parameters (optimizer and its
   parameters, batch size  $B$ , and number of batches  $n_{\text{batch}}$ , number
   of epochs  $n_{\text{epoch}}$ ), and cost function to minimize (here
   mean-square error).
2: for  $n = 1$  to  $n_{\text{epoch}}$  do {loop over epochs}
3:   for  $b = 1$  to  $n_{\text{batch}}$  do {loop over batches}
4:     for  $i = 1$  to  $B$  do {loop over galaxies in batch}
5:       Add random noise to obtain a realization in the S/N
       range chosen
6:       Compute the Tikhonov solution  $\tilde{\mathbf{x}}_i$  using Eq. (2)
7:     end for
8:     Predict  $\hat{\mathbf{x}}_i$  (Eq. (12)) and update network parameters  $\theta$ 
       according to the cost function.
9:   end for
10: end for
11: return  $\mathcal{N}_\theta$ 
    
```

---

### 3.3. ADMMnet: DNNs as constraint in ADMM plug-and-play

The second approach we investigated is using the ADMM PnP framework with a DNN. The ADMM is an augmented Lagrangian technique developed to solve convex problems under linear equality constraints (see, e.g., [Boyd et al. 2010](#)). It operates by decomposing the minimization problem into subproblems solved sequentially. One iteration consists of first solving a minimization problem that typically involves the data fidelity term, then solving a second minimization problem that involves the regularization term, and an update of the dual variable completes the iteration.

It has previously been noted that the first two substeps can be interpreted as an inversion step followed by a denoising step ([Venkatakrishnan et al. 2013](#); [Sreehari et al. 2016](#); [Chan et al. 2017](#)). The steps are coupled by the augmented Lagrangian term and the dual variable. These works suggested using this ADMM structure with nonlinear denoisers in the second step in an approach called ADMM PnP. This has been proposed more recently to be implemented through DNNs ([Meinhardt et al. 2017](#)).

In the following, we adopt this iterative approach based on the ADMM PnP because first, it separates the inversion step and the use of the DNN, which offers flexibility by adding convex constraints to the cost function that can be handled with convex optimization. Second, it lessens the cost of learning by learning essentially a denoiser or a projector. This means fewer networks and fewer parameters to learn jointly compared to unfolding approaches, where each iteration corresponds to a different network. Finally, by iterating between the steps, the output of the network is propagated to the forward model to be compared with the observations. Large discrepancies are avoided in this way, which is different from the Tikhonet approach, where the output of the network is not used in a likelihood.

The training of the network  $\mathcal{N}_\theta$  in this case is similar to Algorithm 1, except that the noise-free training set is composed of noise-free target images instead of noise-free convolved images, and the noise that is added has a constant standard deviation. The algorithm for deconvolving a galaxy is presented in Algorithm 2 and is derived from [Chan et al. \(2017\)](#). The application of the network is illustrated in red. We call this approach ADMMnet. The first step consists of solving the following regularized deconvolution problem at iteration  $k$

---

**Algorithm 2** Proposed ADMM deep plug&play for deconvolution of a galaxy image
 

---

```

1: Initialize: set  $\rho_0, \rho_{\text{max}}, \eta \in [0, 1), \gamma > 1, \Delta_0 = 0, \mathbf{X}^{(0)} = \mathbf{0}, \mathbf{Z}^{(0)} = \mathbf{0}, \boldsymbol{\mu}^{(0)} = \mathbf{0}, \epsilon$ 
2: for  $k = 0$  to  $N_{it}$  do {main loop}
3:   Deconvolution subproblem:  $\mathbf{X}^{(k+1)} =$ 
     FISTA( $\mathbf{Y}, \mathbf{X}^{(k)}, \mathbf{Z}^{(k)}, \boldsymbol{\mu}^{(k)}, \rho_k$ )
4:   Denoising subproblem:  $\mathbf{Z}^{(k+1)} = \mathcal{N}_\theta(\mathbf{X}^{(k+1)} + \boldsymbol{\mu}^{(k)})$ 
5:   Lagrange multiplier update:  $\boldsymbol{\mu}^{(k+1)} = \boldsymbol{\mu}^{(k)} + (\mathbf{X}^{(k+1)} - \mathbf{Z}^{(k+1)})$ 
6:    $\Delta_{k+1} = \frac{1}{\sqrt{\eta}} (\|\mathbf{X}^{(k+1)} - \mathbf{X}^{(k)}\|_2 + \|\mathbf{Z}^{(k+1)} - \mathbf{Z}^{(k)}\|_2 + \|\boldsymbol{\mu}^{(k+1)} - \boldsymbol{\mu}^{(k)}\|_2)$ 
7:   if  $\Delta_{k+1} \geq \eta \Delta_k$  and  $\rho_{k+1} \leq \rho_{\text{max}}$  then
8:      $\rho_{k+1} = \gamma \rho_k$ 
9:   else
10:     $\rho_{k+1} = \rho_k$ 
11:   end if
12:   if  $\|\mathbf{Z}^{(k+1)} - \mathbf{X}^{(k+1)}\|_2 < \epsilon$  then
13:     stop
14:   end if
15: end for
16: return  $\{\mathbf{X}^{(k+1)}\}$ 
    
```

---

using the accelerated iterative convex algorithm FISTA ([Beck & Teboulle 2009](#)):

$$\left\{ \arg \min_{\mathbf{x}_i} \frac{1}{2\sigma^2} \|\mathbf{y}_i - \mathbf{H}_i \mathbf{x}_i\|_2^2 + \iota_C(\mathbf{x}_i) + \frac{\rho}{2} \|\mathbf{x}_i - \mathbf{z}_i^{(k)} + \boldsymbol{\mu}^{(k)}\|_2^2 \right\}_{i=1 \dots n_g}, \quad (13)$$

where  $\iota_C$  is the characteristic function of the non-negative orthant, to enforce the non-negativity of the solution. The DNN used in the second step is employed as an analogy with a denoiser (or as a projector), as presented above. The last step controls the augmented Lagrangian parameter and ensures that this parameter is increased when the change in the estimates is insufficient. This continuation scheme is also important, as noted in [Chan et al. \(2017\)](#), because a progressive increase of the augmented Lagrangian parameter ensures stabilization of the algorithm. The convergence of this scheme is not guaranteed, and in contrast to the convex case, the augmented Lagrangian parameter  $\rho$  is expected to affect the solution. Finally, because the target galaxy is obtained after reconvolution with a target PSF to avoid aliasing (see Sect. 4), we also reconvolved the ADMMnet solution with this target PSF to obtain our final estimate.

### 3.4. Implementation and choice of parameters for the network architecture

We describe here our practical choices for the implementation of the algorithms. For the Tikhonet, the hyperparameter  $\lambda_i$  that controls the balance in between the data fidelity term and the quadratic regularization in Eq. (11) needs to be set for each galaxy. This can be done either manually by selecting a value as a function of an estimate of the S/N, or using automated procedures such as a generalized cross-validation (GCV; [Golub et al. 1979](#)), the L-curve method ([Hansen & O’Leary 1993](#)), the Morozov discrepancy principle ([Engl et al. 1996](#)), various Stein unbiased risk estimate (SURE) minimizations ([Eldar 2009](#); [Pesquet et al. 2009](#); [Deledalle et al. 2014](#)), or a hierarchical Bayesian framework ([Orioux et al. 2010](#); [Pereyra et al. 2015](#)). We compared these approaches and report the results obtained by the

SURE prediction risk minimization, which led to the best results with the GCV approach.

For the ADMM, the parameters  $\rho_0$ ,  $\rho_{\max}$ ,  $\eta$ ,  $\epsilon$ , and  $\gamma$  were selected manually as a balance between quickly stabilizing the algorithm (in particular, high  $\rho$ ) and favoring the minimization of the data fidelity term in the first steps (low  $\rho$ ). We specifically investigated the choice of  $\rho_0$ , which illustrates the effect of the continuation scheme on the solution.

The DNNs were coded in Keras<sup>1</sup> with Tensorflow<sup>2</sup> as backend. For the proposed XDense U-net, four scales were selected with an increasing number of layers for each scale (to capture distant correlations). Each separable convolution was composed of  $3 \times 3$  spatial filters, and a growth factor of 12 was selected for the dense blocks. The total number of trainable parameters was 184301. We also implemented a classical U-net to test the efficiency of the proposed XDense U-net architecture. For this U-net, we chose three scales with two layers per scale and 20 feature maps per layer in the first scale, which gave us 206381 trainable parameters (12% more than the XDense U-net implementation). In both networks we used batch normalization and rectified linear units for the activation. We also tested weighted sigmoid activations for the approach we propose here (or swish in Elfwing et al. 2018), which seems to improve the results slightly, but at the cost of increasing the computational burden. We therefore did not use them in the following results.

In the training phase, we used 20 epochs, a batch size of 32, and selected the Adam optimizer (we kept the default parameters) to minimize the mean-square error (MSE) cost function. After each epoch, we only saved the network parameters when they improved the MSE in the validation set.

## 4. Experiments

In this section we describe how we generated the simulations we used for learning networks and testing our deconvolution schemes. We also report the criteria that we used to compare the different deconvolution techniques.

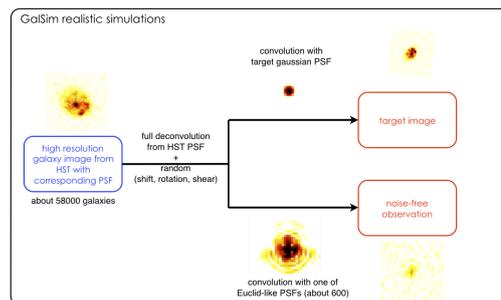
### 4.1. Dataset generation

We used GalSim<sup>3</sup> (Rowe et al. 2015) to generate realistic images of galaxies to train our networks and test our deconvolution approaches. We essentially followed the approach used in GREAT3 (Mandelbaum et al. 2014) to generate the realistic space branch from high-resolution HST images, but chose the PSFs in a set of 600 Euclid-like PSFs (the same as in Farrens et al. 2017). The process is illustrated in Fig. 3.

An HST galaxy was randomly selected from the set of about 58 000 galaxies that were used in the GREAT3 challenge, deconvolved with its PSF. We then applied random shift (taken from a uniform distribution in  $[-1, 1]$  pixel), rotation, and shear. The same cut in S/N was performed as in GREAT3 (Mandelbaum et al. 2014) to obtain a realistic set of galaxies that would be observed in an S/N range  $[20, 100]$  when the noise level is the same as in GREAT3. We here used the same definition of the S/N as in this challenge:

$$S/N(\mathbf{X}_i) = \frac{\|\mathbf{X}_i\|_2}{\sigma}, \quad (14)$$

where  $\sigma$  is the standard deviation of the noise. This S/N corresponds to an optimistic S/N for a detection when the galaxy



**Fig. 3.** Setup for a GalSim simulated realistic galaxy. In the upper branch we obtain the targeted galaxy image. In the lower branch, we simulate the corresponding Euclid-like observed galaxy. A log-scale was adopted for the PSFs to illustrate their complicated structure.

profile  $\mathbf{X}$ ; is known. In other (experimental) definitions, the minimum S/N is indeed closer to 10, similarly to what is usually considered in weak-lensing studies (Mandelbaum et al. 2014).

To obtain the target image in a  $96 \times 96$  grid with a pixel size  $0.05''$ , when the cut in S/N was passed, we first convolved the HST deconvolved galaxy image with a Gaussian PSF with a full width at half maximum  $FWHM = 0.07''$  to ensure that no aliasing occurred after the subsampling. To simulate the observed galaxy without additional noise, we convolved the HST deconvolved image with a PSF that we randomly selected among about 600 Euclid-like PSFs (the same set as was used in Farrens et al. 2017). The same galaxy rotated by  $90^\circ$  was also simulated as in GREAT3.

Because we used real HST galaxies as input, noise from HST images propagated to our target and observed images and was colored by the deconvolution and reconvolution process. We did not decide to denoise the original galaxy images to avoid losing substructures in the target images (and making them less realistic), and as this noise level is lower than the noise added in our simulations, we expect it to change our results only marginally and the ranking of methods not at all.

This process was repeated so that we finally had about 210 000 simulated observed galaxies and their corresponding targets. For the learning, 190 000 galaxies were employed, and for the validation set, 10 000 galaxies. The additional 10 000 galaxies were used to test our approaches.

In the learning phase, additive white Gaussian noise was added to the galaxy batches, and the standard deviation was chosen so that we obtained a galaxy in a prescribed S/N range. For the Tikhonet, we randomly chose an S/N in the range  $[20, 100]$  for each galaxy in the batch, which corresponds to selecting galaxies from the detection limit to galaxies with observable substructures, as illustrated in Fig. 4. For the ADMMnet, we learned a denoising network for a constant noise standard deviation of  $\sigma = 0.04$  (same level as in GREAT3).

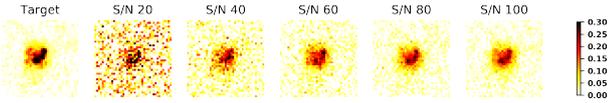
We then tested the relative performance of the different approaches in a test set for fixed values of  $S/N \in \{20, 40, 60, 80, 100\}$  to better characterize (and distinguish) them, and for a fixed standard deviation of  $\sigma = 0.04$ , which corresponds to what was simulated in GREAT3 for the real galaxy space branch to obtain results for a representative observed galaxy set. The corresponding S/N distribution in the last scenario is represented in Fig. 5. All the techniques were compared in exactly the same test sets.

When we tested at different S/N in the ADMMnet approach, we adjusted the noise level in the galaxy images to the noise level in the learning phase. We therefore rescaled the galaxy images to

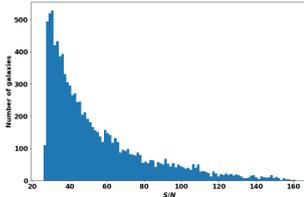
<sup>1</sup> <https://keras.io>

<sup>2</sup> <https://www.tensorflow.org/>

<sup>3</sup> <https://github.com/GalSim-developers/GalSim>



**Fig. 4.** Range of the S/N we used for to train and test the simulations. *From left to right:* targeted galaxy image, and the observed convolved images at increasing S/N. In our definition, S/N 20 is barely at the galaxy detection limit, while galaxy substructures can be visualized at S/N 100.



**Fig. 5.** Distribution of the S/N of simulated galaxies for constant noise simulations ( $\sigma = 0.04$ ). The distribution peak is at about S/N 30, and the mean at S/N 54.

reach this targeted noise level, based on the noise level estimation in the images. We used a robust standard procedure based on computing the median absolute deviation in the wavelet domain for this (using orthogonal daubechies wavelets with three vanishing moments).

#### 4.2. Quality criteria

The performance of the deconvolution schemes was measured according to two criteria that are related to pixel error and shape measurement errors. For the pixel error we selected a robust estimator,

$$P_{\text{err}}(\widehat{\mathbf{X}}) = \text{MED} \left( \frac{\|\widehat{\mathbf{x}}_i - \mathbf{x}_i^{(t)}\|_2^2}{\|\mathbf{x}_i^{(t)}\|_2^2} \right)_{i=1 \dots n_g}, \quad (15)$$

where  $\mathbf{x}_i^{(t)}$  is the targeted value, and MED is the median over the relative MSE computed for each galaxy  $\mathbf{x}_i$  in the test set in a central window of  $41 \times 41$  pixels common to all approaches.

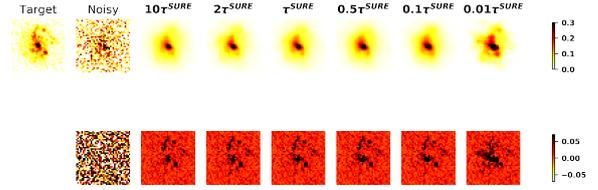
For the shape measurement errors, we computed the ellipticity using a Kaiser-Squires-Broadhurst approach implemented in shapelens<sup>4</sup> (Kaiser et al. 1995; Viola et al. 2011), which additionally computes an adapted circular weight function from the data.

We first applied this method to the targets and also took the target isotropic Gaussian PSF into account to obtain reference complex ellipticities  $\epsilon_i$  and windows. We then computed the complex ellipticity  $\widehat{\epsilon}_i$  of the deconvolved galaxies using the same circular weight functions as their target counterpart. Finally, we computed

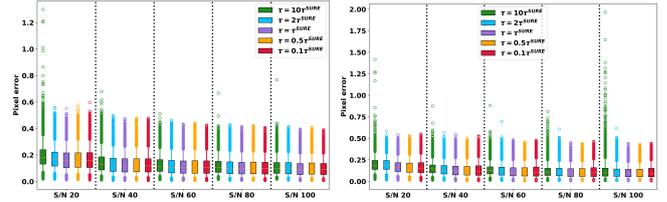
$$\epsilon_{\text{err}}(\widehat{\mathbf{X}}) = \text{MED} \left( \|\widehat{\epsilon}_i - \epsilon_i\|_2 \right)_{i=1 \dots n_g} \quad (16)$$

to obtain a robust estimate of the ellipticity error in the windows set up by the target images, again in a central window of  $41 \times 41$  pixels common to all approaches. We also report the distribution of pixel and ellipticity errors prior to applying the median when finer assessments need to be made.

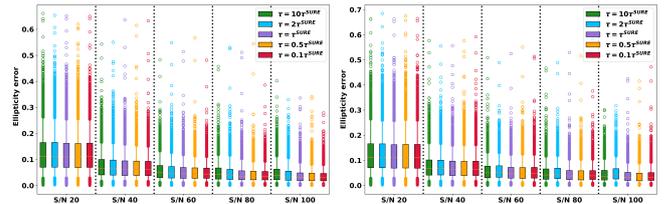
<sup>4</sup> <https://github.com/pmelchior/shapelens>



**Fig. 6.** Visual effect of the hyperparameter choice for the Tikhonet approach at S/N 20. *Top:* target and observations, followed by estimates with different multiplicative factors for the hyperparameter based on SURE. *Bottom:* residuals associated with the top row.



**Fig. 7.** Effect of the hyperparameter multiplicative factor value for the Tikhonet approach using the proposed XDense U-net architecture (*left*) and classical U-net architecture (*right*) in terms of pixel errors. The box indicates quartiles, and the vertical bars encompass 90% of the data. Outliers are displayed with circles.



**Fig. 8.** Effect of the hyperparameter multiplicative factor value for the Tikhonet approach using the proposed XDense U-net architecture (*left*) and classical U-net architecture (*right*) in terms of ellipticity errors. The box indicates quartiles, and the vertical bars encompass 90% of the data. Outliers are displayed with circles.

## 5. Results

### 5.1. Setting the Tikhonet architecture and hyperparameters

For the Tikhonet, the key parameters to set are the hyperparameters  $\lambda_i$  in Eq. (11). In Fig. 6 these hyperparameters are set to the parameters that minimize the SURE multiplied by factors, which range from 10 to 0.01 for the proposed X-Dense architecture at S/N 20 (similar visual results are obtained for the classical U-net). It appears that for the lowest factor, which corresponds to the smallest regularization of deconvolution (i.e., more noise added to the deconvolved image), the Tikhonet is unable to perform as well as for intermediate values, in particular for exactly the SURE minimizer.

This is confirmed in Fig. 7, which reports the pixel errors for the two proposed X-Dense and classical architecture. Figure 8 shows the ellipticity errors.

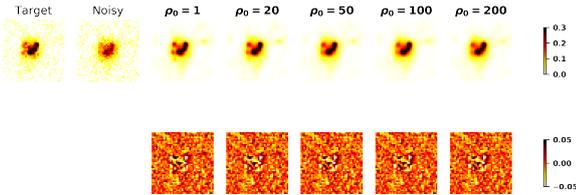
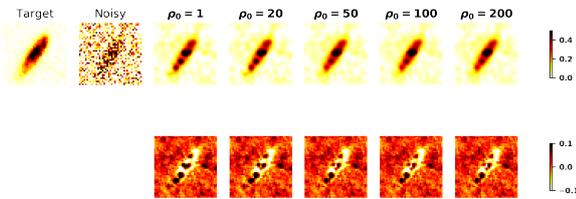
For both architectures, the best results in terms of pixel or ellipticity errors are consistently obtained across all S/N that we tested for values of the multiplicative factor between 0.1 and 1. Higher multiplicative factors also lead to larger extreme errors, in particular, at low S/N. In the following, we therefore set this parameter to the SURE minimizer.

Concerning the choice of architecture, Fig. 7 illustrates that the XDense U-net provides less extreme outliers in pixel errors

**Table 1.** Comparison of U-net architectures for the SURE selected hyperparameter.

	$S/N$ 20	$S/N$ 40	$S/N$ 60	$S/N$ 80	$S/N$ 100
Median pixel error	<b>0.157</b> (0.163)	<b>0.117</b> (0.121)	<b>0.105</b> (0.106)	<b>0.097</b> (0.097)	<b>0.090</b> (0.093)
Median ellipticity error	<b>0.109</b> (0.110)	<b>0.063</b> (0.064)	<b>0.045</b> (0.046)	<b>0.035</b> (0.038)	<b>0.030</b> (0.033)

**Notes.** The first number is obtained with the XDense U-net architecture, and the second in parentheses with the classical U-net architecture.

**Fig. 9.** Visual effect of initializing  $\rho$  for the ADMMnet for  $S/N$  100. *Top:* target and observations, followed by estimates with different augmented Lagrangian parameters  $\rho_0$ . *Bottom:* residuals associated with the top row.**Fig. 10.** Visual effect of initializing  $\rho$  for the ADMMnet for  $S/N$  20. *Top:* target and observations, followed by estimates with different augmented Lagrangian parameters  $\rho_0$ . *Bottom:* residuals associated with the top row.

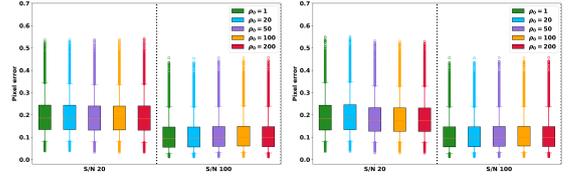
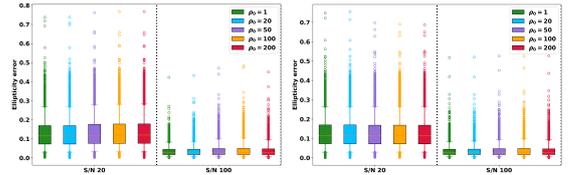
for a multiplicative factor of 10 regardless of  $S/N$ , which is far from providing the best results, however. When we study the median error values in Table 1 for the SURE minimizers more closely, slightly better results are consistently obtained for the proposed XDense U-net architecture. In this experiment, the XDense obtains 4% (3%) fewer pixel errors at  $S/N$  20 ( $S/N$  100), and the most significant difference is an about 8% improvement in ellipticity measurement at  $S/N$  100.

## 5.2. Setting the ADMMnet architecture and hyperparameters

For the ADMMnet, we manually set the hyperparameters  $\rho_{\max} = 200$ ,  $\epsilon = 0.01$  to lead to an ultimate stabilization of the algorithm,  $\eta = 0.5$  and  $\gamma = 1.4$ , to explore intermediate  $\rho$  values, and we investigated the choice of parameter  $\rho_0$  to illustrate the effect of the continuation scheme on the solution. This is illustrated in Fig. 9 at high  $S/N$  and Fig. 10 at low  $S/N$  for the proposed XDense U-net architecture. When  $\rho_0$  is small, higher frequencies are recovered in the solution, as illustrated in galaxy substructures in Fig. 9, but this could lead to artifacts at low  $S/N$ , as illustrated in Fig. 10.

Quantitative results for the two architectures are presented in Fig. 11 for pixel errors and in Fig. 12 for ellipticity errors. The error distribution is very stable with respect to the hyperparameter  $\rho_0$  value, and it is similar for both architectures.

The pixel error at high  $S/N$  and low  $S/N$  for both architectures shows that the lowest pixel errors in terms of the median are obtained at low  $S/N$  for larger  $\rho_0$ , while at high  $S/N$ ,  $\rho_0 = 1$

**Fig. 11.** Effect of the hyperparameter  $\rho_0$  value for the ADMMnet in terms of pixel errors for the proposed XDense U-net (*left*) and classical U-net (*right*). The box indicates quartiles, and the vertical bars encompass 90% of the data. Outliers are displayed with circles.**Fig. 12.** Effect of the hyperparameter choice  $\rho_0$  for the ADMMnet in terms of ellipticity error for the proposed XDense U-net (*left*) and classical U-net (*right*). The box indicates quartiles, and the vertical bars encompass 90% of the data. Outliers are displayed with circles.**Table 2.** Comparison of U-net architectures for median errors.

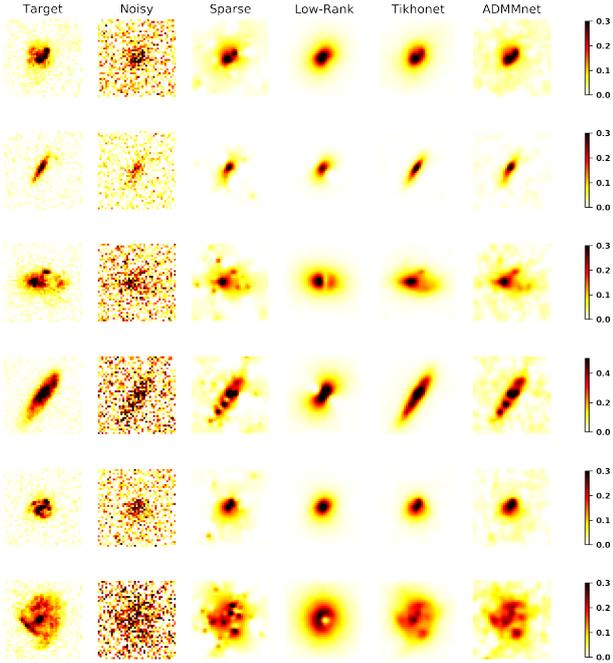
	$S/N$ 20				
	$\rho_0 = 1$	$\rho_0 = 20$	$\rho_0 = 50$	$\rho_0 = 100$	$\rho_0 = 200$
Median pixel error	0.186 ( <b>0.184</b> )	<b>0.185</b> (0.186)	0.182 ( <b>0.176</b> )	0.183 ( <b>0.175</b> )	0.182 ( <b>0.175</b> )
Median ellipticity error	<b>0.114</b> (0.116)	<b>0.114</b> (0.116)	0.118 ( <b>0.115</b> )	0.119 ( <b>0.115</b> )	0.119 ( <b>0.115</b> )
	$S/N$ 100				
	$\rho_0 = 1$	$\rho_0 = 20$	$\rho_0 = 50$	$\rho_0 = 100$	$\rho_0 = 200$
Median pixel error	<b>0.095</b> (0.096)	<b>0.096</b> (0.097)	<b>0.098</b> (0.099)	<b>0.099</b> (0.099)	<b>0.097</b> (0.098)
Median ellipticity error	<b>0.028</b> (0.028)	0.028 ( <b>0.028</b> )	0.029 ( <b>0.029</b> )	0.029 ( <b>0.029</b> )	0.029 ( <b>0.028</b> )

**Notes.** The first number is obtained with the proposed XDense U-net architecture, and the second in parentheses with the classical U-net architecture.

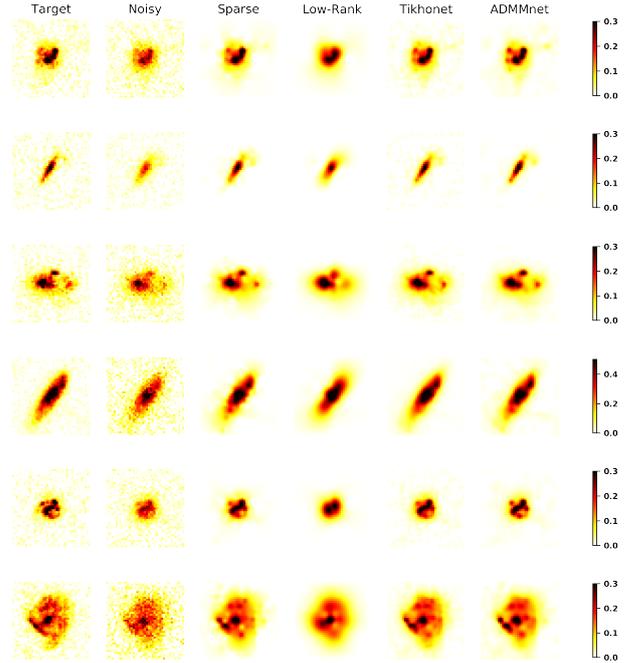
is the best. In terms of ellipticity errors,  $\rho_0 = 1$  allows us to consistently obtain the best results at low and high  $S/N$ .

To better compare the differences between the two architectures, the median errors are reported in Table 2. At low  $S/N$ , the classical U-net performance varies more than that of the XDense, and at  $S/N$  20, the best results are obtained for the classical U-net approach (4% improvement over X-Dense). At high  $S/N$ , however, the best results are consistently obtained across  $\rho_0$  values with the proposed XDense U-net (but the improvement is only 1% over the U-net for the best  $\rho_0 = 1$ ). Finally, the best results for the ellipticity median errors are obtained for the lowest value  $\rho_0 = 1$  for both architectures, and the proposed XDense U-net performs slightly better than the classical U-net (about 1% better at low and high  $S/N$ ).

Overall, this illustrates that the continuation scheme has a weak effect, specifically, on the ellipticity errors, and that the best results are obtained for different  $\rho_0$  and network architectures when the pixel or ellipticity errors are considered, depending on the  $S/N$ . The classical U-net allows smaller pixel errors than the proposed XDense at low  $S/N$ , but also leads to slightly higher pixel errors at higher  $S/N$  and ellipticity errors at both low and high  $S/N$ . In practice, we keep in the following  $\rho_0 = 1$  for the proposed XDense U-net approach for further comparison with other deconvolution approaches because the pixel error varies slowly with this architecture as a function of  $\rho_0$ .



**Fig. 13.** Deconvolved images with the various approaches for  $S/N$  20. Each row corresponds to a different processed galaxy. *From left to right:* image to recover, observation with a noise realization, sparse and low-rank approaches, and the Tikhonet and ADMMnet results.



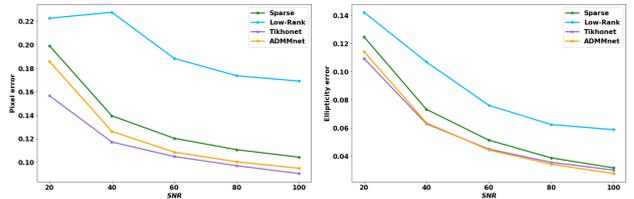
**Fig. 14.** Deconvolved images with the various approaches for  $S/N$  100. Each row corresponds to a different processed galaxy. *From left to right:* image to recover, observation with a noise realization, sparse and low-rank approaches, and the Tikhonet and ADMMnet results.

### 5.3. DNN versus sparsity and low-rank approaches

We compared our two deep-learning schemes with the XDense U-net architecture and the hyperparameters set as described in the previous sections with the sparse and low-rank approaches of Farrens et al. (2017), which are implemented in `sf_deconvolve`<sup>5</sup>. For the two methods, we used all parameters selected by default, reconvolved the recovered galaxy images with the target PSF, and selected the central  $41 \times 41$  pixels of the observed galaxies to be processed in particular to speed up the computation of the singular value decomposition used in the low-rank constraint (and therefore of the whole algorithm), as in Farrens et al. (2017). All comparisons were made in this central region of the galaxy images.

We now report the results for a variety of galaxies that we recovered at different  $S/N$  for the sparse and low-rank deconvolution approaches and for the Tikhonet and ADMMnet. We first display several results at low  $S/N$  in Fig. 13 to illustrate the robustness of the various deconvolution approaches. Important artifacts appear in the sparse approach, which illustrates the difficulty of recovering the galaxy images in this high-noise scenario: the retained noise in the deconvolved images leads to these point-like artifacts.

For the low-rank approach, low frequencies seems to be partially well recovered, but artifacts appear for elongated galaxies in the direction of the minor axis. Finally, both Tikhonet and ADMMnet seem to recover the low-frequency information better, but the galaxy substructures are essentially lost. The ADMMnet in this situation appears to recover sharper images, but they have more propagated noise or artifacts than the Tikhonet approach, with similar features as for the sparse approach, but with fewer point-like artifacts.



**Fig. 15.** Deconvolution quality criteria for the different deconvolution schemes. *Left:* median pixel error. *Right:* median ellipticity error.

We also display these galaxies at a much higher  $S/N$  in Fig. 14 to assess the ability of the various deconvolution schemes to recover galaxy substructures in a low-noise scenario.

The low-rank approach displays fewer artefacts than at low  $S/N$ , but still does not seem to be able to adequately represent elongated galaxies or directional substructures in the galaxy. The reason probably is that the low-rank approach does not adequately cope with translations, which leads to overly smooth solutions. In contrast, Tikhonet, ADMMnet, and sparse recovery lead to a recovery of galaxy substructures. Overall, the two proposed deconvolution approaches using DNNs lead to the best visual results regardless of  $S/N$ .

The quantitative deconvolution criteria are presented in Fig. 15. For the median pixel error, this figure illustrates that Tikhonet and ADMMnet both perform better than the sparse and low-rank approaches in recovering the galaxy intensity values, regardless of the  $S/N$ . In these noise settings, the low-rank approach performed consistently poorer than sparsity. For the pixel errors, the median errors obtained with the sparse approach are 27% (5%) larger at  $S/N$  20 ( $S/N$  100) than those of the Tikhonet results. The Tikhonet approach also appears to perform slightly better than the ADMMnet with this criterion.

The best results for the shape measurement errors are obtained with the Tikhonet approach at low  $S/N$  (up to  $S/N$  40),

<sup>5</sup> [https://github.com/sfarrens/sf\\_deconvolve](https://github.com/sfarrens/sf_deconvolve)

**Table 3.** Criteria for constant noise simulations ( $\sigma = 0.04$ ).

	Sparse	Low-rank	Tikhonet	ADMMnet
Median pixel error	0.130	0.169	<b>0.112</b>	0.119
Median ellipticity error	0.061	0.072	<b>0.053</b>	0.054

**Notes.** The best results are indicated in bold.

**Table 4.** Computing time for the various approaches (in hours).

Method	Learning	Processing 10 000 galaxies
Sparse	/	24.7
Low-rank	5.2	
Tikhonet	21.5	0.1
ADMMnet	16.2	20.3

and then the ADMMnet outperforms the other approaches at higher S/N. In terms of ellipticity errors, the median errors of the sparse approach are 14% (5%) larger at  $S/N$  20 ( $S/N$  100) than for the Tikhonet results. Finally, the low-rank approach performs poorest regardless of the S/N. To summarize, these results clearly favor the choice of the DNN approach because it consistently produces lower errors regardless of S/N.

This is confirmed by a realistic distribution of galaxy S/N, as shown in Table 3. The proposed deep-learning approaches perform similarly in median pixel and ellipticity errors, and they outperform the sparse and low-rank approaches: the median pixel error is reduced by almost 14% (9%) for the Tikhonet (ADMMnet) approach compared to sparse recovery, and the ellipticity errors are reduced by about 13% for both approaches. Higher differences are observed for the low-rank approach.

#### 5.4. Computing time

Finally, we also report in Table 4 the time required to learn the networks and process the set of 10 000 galaxies on the same GPU/CPU. This is a crucial aspect when a large number of galaxies is to be processed, such as in modern surveys. Of the DNNs, learning the parameters of the denoising network for the ADMMnet is faster than those of the post-processing network in the Tikhonet approach because the latter requires each batch to be deconvolved. However, when the network parameters have been learned, the Tikhonet approach based on a closed-form deconvolution is the fastest to process a large number of galaxies (about 0.05 s per galaxy). On the other hand, learning and restoring 10 000 galaxies is quite fast for the low-rank approach, while iterative algorithms such as ADMMnet or the primal-dual algorithm for sparse recovery are similar in terms of computing time (about 7 to 10 s per galaxy). All these computing times might be reduced, however, when the restoration of different galaxy images is performed in parallel. This is not implemented so far.

## 6. Conclusions

We have proposed two new space-variant deconvolution strategies for galaxy images based on deep neural networks while keeping all knowledge of the PSF in the forward model: the Tikhonet approach is a post-processing approach of a simple Tikhonov deconvolution with a DNN, and the ADMMnet approach is based on regularization by a DNN denoiser inside an iterative ADMM PnP algorithm for deconvolution.

We proposed to use a DNN architecture for the galaxy processing that is based on the U-net, which is particularly adapted to deconvolution problems, with small modifications implemented (dense blocks of separable convolutions, and no skip connection) to decrease the number of parameters that are to be learned compared to a classical U-net implementation. We finally evaluated these approaches compared to the deconvolution techniques in Farrens et al. (2017) in simulations of realistic galaxy images derived from HST observations, with realistically sampled sparse-variant PSFs and noise, processed with the GalSim simulation code. We investigated in particular how to set the hyperparameters in the two approaches: the Tikhonov hyperparameter for the Tikhonet approach, and the continuation parameters for the ADMMnet approach. We compared our proposed XDense U-net architecture with a classical U-net implementation. Our main findings are listed below.

In the Tikhonet and ADMMnet approaches, the hyperparameters affect the performance, but the results are quite stable in a range of values for these hyperparameters. In particular for the Tikhonet approach, the SURE minimizer is within this range. For the ADMMnet, more hyperparameters need to be set, and the initialization of the augmented Lagrangian parameter affects the performance: small parameters lead to higher frequencies in the images, and larger parameters lead the recovery of overly smooth galaxies.

Compared to the classical implementation, the XDense U-net leads to consistently improved criteria for the Tikhonet approach. The situation is more balanced for the ADMMnet, where smaller pixel errors can be achieved at low S/N with the classical architecture (with high hyperparameter values), but the XDense U-net provides the best results for pixel errors at high S/N and ellipticity errors at high and low S/N (with low hyperparameter values). The ranking of the methods is unchanged, regardless of which architecture with their best hyperparameter value is selected.

The two methods visually outperform the sparse recovery and low-rank techniques, which display artifacts at the low S/N we probed (also for high S/N in the low-rank approach). This is also confirmed in all S/N ranges and for a realistic distribution of S/N. In the latter, an improvement of about 14% is achieved in terms of the median pixel error and an improvement of about 13% for the median shape measurement errors for the Tikhonet approach compared to sparse recovery.

In the DNN approaches, the Tikhonet approach outperforms the ADMMnet approach in terms of median pixel errors regardless of the S/N, and in terms of median ellipticity errors for low S/N ( $S/N < 40$ ). At higher S/N, the ADMMnet approach leads to slightly smaller ellipticity errors. The Tikhonet approach is fastest when the network parameters have been learned. About 0.05 s are required to process a galaxy. This is to be compared with sparse and ADMMnet iterative deconvolution approaches, which take about 7–10 s per galaxy.

Note that ADMMnet approach can further be improved, as additional constraints could be added easily to the framework (while the success of the Tikhonet approach also lies in the ability to compute a closed-form solution for the deconvolution step). Overall, these results illustrate that the Tikhonet is the best approach in this scenario to fastly process a large number of galaxies with high accuracy.

*Acknowledgements.* The authors thank the GalSim developers/GREAT3 collaboration for publicly providing simulation codes and galaxy databases, and the developers of sf\_deconvolve and shapelens as well for their code publicly available.

## References

- Adler, J., & Öktem, O. 2017, *Inverse Prob.*, **33**, 124007
- Adler, J., & Öktem, O. 2018, *IEEE Trans. Med. Imaging*, **37**, 1322
- Afonso, M., Bioucas-Dias, J., & Figueiredo, M. 2011, *IEEE Trans. Image Process.*, **20**, 681
- Andrews, H. C., & Hunt, B. R. 1977, *Digital Image Restoration* (Englewood Cliffs, NJ: Prentice-Hall)
- Beck, A., & Teboulle, M. 2009, *SIAM J. Imaging Sci.*, **2**, 183
- Bertero, M., & Boccacci, P. 1998, *Introduction to Inverse Problems in Imaging* (Institute of Physics)
- Bertin, E. 2011, in *Astronomical Data Analysis Software and Systems XX*, eds. I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, *ASP Conf. Ser.*, **442**, 435
- Bertocchi, C., Chouzenoux, E., Corbineau, M. C., Pesquet, J. C., & Prato, M. 2018, ArXiv e-prints [arXiv:1812.04276]
- Bigdeli, S. A., Jin, M., Favaro, P., & Zwicker, M. 2017, *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17* (USA: Curran Associates Inc.), 763
- Bioucas-Dias, J. 2006, *IEEE Trans. Image Process.*, **15**, 937
- Bobin, J., Sureau, F., Starck, J.-L., Rassat, A., & Paykari, P. 2014, *A&A*, **563**, A105
- Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. 2010, *Mach. Learn.*, **3**, 1
- Cai, J., Osher, S., & Shen, Z. 2010, *Multiscale Model. Simul.*, **8**, 337
- Chambolle, A., & Pock, T. 2011, *J. Math. Imaging Vision*, **40**, 120
- Chan, S. H., Wang, X., & Elgendy, O. 2017, *IEEE Trans. Comput. Imaging*, **3**, 84
- Chollet, F. 2016, 2017, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1800
- Combettes, P. L., & Pesquet, J. C. 2011, in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, eds. H. Bauschke, R. Burachik, & P. Combettes (Springer), 185
- Combettes, P. L., & Vu, B. C. 2014, *Optimization*, **63**, 1289
- Condat, L. 2013, *J. Optim. Theory App.*, **158**, 460
- Deledalle, C., Vaiter, S., Fadili, J., & Peyré, G. 2014, *SIAM J. Imaging Sci.*, **7**, 2448
- Donoho, D. L. 1995, *Appl. Comput. Harmonic Anal.*, **2**, 101
- Eldan, R., & Shamir, O. 2015, *Conference on Learning Theory*
- Eldar, Y. 2009, *IEEE Trans. Signal Proc.*, **57**, 471
- Elfving, S., Uchibe, E., & Doya, K. 2018, *Neural Networks*, **107**, 3, Special issue on deep reinforcement learning
- Engl, H. W., Hanke, M., & Neubauer, A. 1996, *Regularization of Inverse Problems* (Dordrecht: Kluwer)
- Fan, F., Li, M., Teng, Y., & Wang, G. 2018, ArXiv e-prints [arXiv:1812.11675]
- Farrens, S., Starck, J.-L., & Mboula, F. 2017, *A&A*, **601**
- Garsden, H., Girard, J. N., Starck, J. L., et al. 2015, *A&A*, **575**, A90
- Golub, G. H., Heath, M., & Wahba, G. 1979, *Technometrics*, **21**, 215
- Gregor, K., & LeCun, Y. 2010, *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10* (USA: Omnipress), 399
- Guerrero-Colon, J. A., & Portilla, J. 2006, in *2006 International Conference on Image Processing*, 625
- Gupta, H., Jin, K. H., Nguyen, H. Q., McCann, M. T., & Unser, M. 2018, *IEEE Trans. Med. Imaging*, **37**, 1440
- Han, Y., & Ye, J. C. 2018, *IEEE Trans. Med. Imaging*, **37**, 1418
- Hansen, P. C., & O'Leary, D. 1993, *SIAM J. Sci. Comput.*, **14**, 1487
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770
- Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. 2017, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261
- Hunt, B. R. 1972, *IEEE Transactions on Automatic and Control*, **AC-17**, 703
- Ioffe, S., & Szegedy, C. 2015, *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15 (JMLR.org)*, **37**, 448
- Jia, C., & Evans, B. L. 2011, in *2011 18th IEEE International Conference on Image Processing*, 681
- Jin, K. H., McCann, M. T., Froustey, E., & Unser, M. 2017, *IEEE Trans. Image Process.*, **26**, 4509
- Kaiser, N., Squires, G., & Broadhurst, T. 1995, *ApJ*, **449**, 460
- Kalifa, J., Mallat, S., & Rouge, B. 2003, *IEEE Trans. Image Process.*, **12**, 446
- Kingma, D., & Ba, J. 2014, *International Conference on Learning Representations*
- Krishnan, D., & Fergus, R. 2009, in *Advances in Neural Information Processing Systems*, eds. Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, & A. Culotta (Curran Associates, Inc.), **22**, 1033
- Krist, J. E., Hook, R. N., & Stoehr, F. 2011, in *Optical Modeling and Performance Predictions V*, Proc. SPIE, 8127, 81270J
- Kuijken, K., Heymans, C., Hildebrandt, H., et al. 2015, *MNRAS*, **454**, 3500
- Lanusse, F., Starck, J.-L., Leonard, A., & Pires, S. 2016, *A&A*, **591**, A2
- LeCun, Y., Bengio, Y., & Hinton, G. 2015, *Nature*, **521**, 436
- Li, Y., Tofighi, M., Monga, V., & Eldar, Y. C. 2019, *ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7675
- Lou, Y., Bertozzi, A. L., & Soatto, S. 2011, *J. Math. Imaging Vision*, **39**, 1
- Mairal, J., Sapiro, G., & Elad, M. 2008, *Multiscale Model. Simul.*, **7**, 214
- Mallat, S. 2016, *Philos. Trans. R. Soc. A: Math. Phys. Eng. Sci.*, **374**, 20150203
- Mandelbaum, R., Rowe, B., Bosch, J., et al. 2014, *ApJS*, **212**, 5
- Mandelbaum, R., Rowe, B., Armstrong, R., et al. 2015, *MNRAS*, **450**, 2963
- Mardani, M., Gong, E., Cheng, J. Y., Pauly, J. M., & Xing, L. 2017, *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, CAMSAP 2017, Curaçao, The Netherlands, December 10–13, 2017*, 1
- Mardani, M., Sun, Q., Vasawanala, S., et al. 2018, *Proceedings of the 32Nd International Conference on Neural Information Processing Systems, NIPS'18* (USA: Curran Associates Inc.), 9596
- Mboula, F., Starck, J.-L., Okumura, K., Amiaux, J., & Hudelot, P. 2016, *Inverse Prob.*, **32**
- Meinhardt, T., Moller, M., Hazirbas, C., & Cremers, D. 2017, *Proceedings of the IEEE International Conference on Computer Vision*, 1781
- Monga, V., Li, Y., & Eldar, Y. C. 2019, *Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing*
- Neelamani, R., Choi, Hyeokho, & Baraniuk, R. 2004, *IEEE Trans. Signal Process.*, **52**, 418
- Oliveira, J. P., Bioucas-Dias, J. M., & Figueiredo, M. A. 2009, *Signal Process.*, **89**, 1683
- Orieux, F., Giovannelli, J., & Rodet, T. 2010, in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, 1350
- Pereyra, M., Bioucas-Dias, J. M., & Figueiredo, M. A. T. 2015, in *2015 23rd European Signal Processing Conference (EUSIPCO)*, 230
- Pesquet, J., Benazza-Benyahia, A., & Chau, C. 2009, *IEEE Trans. Signal Process.*, **57**, 4616
- Petersen, P., & Voigtlaender, F. 2018, *Neural Networks*, **108**, 296
- Pustelnik, N., Benazza-Benhayia, A., Zheng, Y., & Pesquet, J.-C. 2016, *Wiley Encyclopedia of Electrical and Electronics Engineering*
- Reehorst, E. T., & Schniter, P. 2018, *IEEE Trans. Comput. Imaging*, **1**
- Ribli, D., Pataki, B. Á., & Csabai, I. 2019, *Nat. Astron.*, **3**, 93
- Romano, Y., Elad, M., & Milanfar, P. 2017, *SIAM J. Imaging Sci.*, **10**, 1804
- Ronneberger, O., Fischer, P., & Brox, T. 2015, in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, eds. N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Cham: Springer International Publishing), 234
- Rowe, B., Jarvis, M., Mandelbaum, R., et al. 2015, *Astron. Comput.*, **10**, 121
- Safran, I., & Shamir, O. 2017, in *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17 (JMLR.org)*, **70**, 2979
- Schmitz, M. A., Starck, J. L., Ngole Mboula, F., et al. 2020, *A&A*, **636**, A78
- Schuler, C. J., Burger, H. C., Harmeling, S., & Schölkopf, B. 2013, *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 1067
- Schuler, C. J., Hirsch, M., Harmeling, S., & Schölkopf, B. 2016, *IEEE Trans. Pattern Anal. Mach. Intell.*, **38**, 1439
- Sreehari, S., Venkatakrishnan, S., Wohlberg, B., et al. 2016, *IEEE Trans. Comput. Imaging*, **2**, 408
- Starck, J.-L., Bijaoui, A., Valtchanov, I., & Murtagh, F. 2000, *A&AS*, **147**, 139
- Starck, J.-L., Nguyen, M., & Murtagh, F. 2003, *Signal Process.*, **83**, 2279
- Starck, J.-L., Murtagh, F., & Bertero, M. 2015a, *Starlet Transform in Astronomical Data Processing* (New York, NY: Springer), 2053
- Starck, J.-L., Murtagh, F., & Fadili, J. 2015b, *Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis* (Cambridge University Press)
- Sureau, F. C., Starck, J.-L., Bobin, J., Paykari, P., & Rassat, A. 2014, *A&A*, **566**, A100
- Szegedy, C., Ioffe, S., & Vanhoucke, V. 2016, *AAAI Conference on Artificial Intelligence*
- Tikhonov, A. N., & Arsenin, V. Y. 1977, *Solutions of Ill-posed problems*, ed. W. H. Winston
- Twomey, S. 1963, *J. ACM*, **10**, 97
- Venkatakrishnan, S. V., Bouman, C. A., & Wohlberg, B. 2013, *2013 IEEE Global Conference on Signal and Information Processing*, 945
- Viola, M., Melchior, P., & Bartelmann, M. 2011, *MNRAS*, **410**, 2156
- Xu, L., Ren, J., Liu, C., & Jia, J. 2014, *Adv. Neural Inf. Process. Syst.*, **2**, 1790
- Ye, J. C., Han, Y., & Cha, E. 2018a, *SIAM J. Imaging Sci.*, **11**, 991
- Zhang, K., Zuo, W., Gu, S., & Zhang, L. 2017, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2808
- Zibulevsky, M., & Elad, M. 2010, *IEEE Signal Process. Mag.*, **27**, 76
- Zuntz, J., Sheldon, E., Samuroff, S., et al. 2018, *MNRAS*, **481**, 1149