Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling

Xu, Xiaojian
Liu, Jiaming
Sun, Yu
Wohlberg, Brendt Egon
Kamilov, Ulugbek

Boosting the Performance of Plug-and-Play Priors via Denoiser Scaling

Xiaojian Xu*, Jiaming Liu†, Yu Sun*,
Brendt Wohlberg‡, and Ulugbek S. Kamilov*,†

February 27, 2020

Abstract

Plug-and-play priors (PnP) is an image reconstruction framework that uses an image denoiser as an imaging prior. Unlike traditional regularized inversion, PnP does not require the prior to be expressible in the form of a regularization function. This flexibility enables PnP algorithms to exploit the most effective image denoisers, leading to their state-of-the-art performance in various imaging tasks. In this paper, we propose a new *denoiser scaling* technique to explicitly control the amount of PnP regularization. Traditionally, the performance of PnP algorithms is controlled via intrinsic parameters of the denoiser related to the noise variance. However, many powerful denoisers, such as the ones based on convolutional neural networks (CNNs), do not have tunable parameters that would allow controlling their influence within PnP. To address this issue, we introduce a scaling parameter that adjusts the magnitude of the denoiser input and output. We theoretical justify the denoiser scaling from the perspectives of proximal optimization, statistical estimation, and consensus equilibrium. Finally, we provide numerical experiments demonstrating the ability of denoiser scaling to systematically improve the performance of PnP for denoising CNN priors that do not have explicitly tunable parameters.

1 Introduction

Image formation is naturally posed as an inverse problem, which is often ill-posed. Regularized inversion is a widely adopted framework for dealing with such ill-posed inverse problems by taking advantage of prior information regarding the unknown image. Examples of popular imaging priors include nonnegativity, transform-domain sparsity, and self-similarity [1–6]. Since imaging priors are often nondifferentiable, proximal algorithms [7] – such as variants of iterative shrinkage/thresholding algorithm (ISTA) [8–11] and alternating direction method of multipliers (ADMM) [12–15] – are extensively used in image reconstruction. These algorithms avoid differentiating the regularizer by using the *proximal operator* [7], a mathematical concept that corresponds to regularized image denoising.

Plug-and-play priors (PnP) [16] is a framework for regularized inversion that does not require the image prior to be expressible in the form of a regularization function. Recent results have shown that by using advanced image denoisers in iterative image reconstruction, PnP algorithms achieve state-of-the-art performance in many imaging problems [17–23]. PnP algorithms have also been successfully combined with powerful denoising *convolutional neural nets (CNNs)* for exploiting learned imaging priors while enforcing fidelity to the measured data [24–28]. (A similar strategy for using general image denoisers has also been adopted in related frameworks known as *approximate message passing (AMP)* [29–32] and *regularization by denoising (RED)* [33–38].)

---

*Department of Computer Science & Engineering, Washington University in St. Louis, St. Louis, MO 63130.
†Department of Electrical & Systems Engineering, Washington University in St. Louis, St. Louis, MO 63130.
‡Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545 USA.

Algorithm 1 PnP-ADMM	**Algorithm 2** PnP-ISTA/PnP-FISTA
1: **input:** $\boldsymbol{x}^0$, $\boldsymbol{s}^0 = \boldsymbol{0}$, $\gamma > 0$, and $\mu > 0$	1: **input:** $\boldsymbol{x}^0 = \boldsymbol{s}^0$, $\gamma > 0$, $\mu > 0$, and $\{q_k\}_{k\in\mathbb{N}}$
2: **for** $k = 1, 2, \ldots$ **do**	2: **for** $k = 1, 2, \ldots$ **do**
3: $\boldsymbol{z}^k = \mathsf{prox}_{\gamma g}(\boldsymbol{x}^{k-1} + \boldsymbol{s}^{k-1})$	3: $\boldsymbol{z}^k = \boldsymbol{s}^{k-1} - \gamma \nabla g(\boldsymbol{s}^{k-1})$
4: $\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k - \boldsymbol{s}^{k-1})$	4: $\boldsymbol{x}^k = \mathsf{D}_\sigma(\boldsymbol{z}^k)$
5: $\boldsymbol{s}^k = \boldsymbol{s}^{k-1} + (\boldsymbol{x}^k - \boldsymbol{z}^k)$	5: $\boldsymbol{s}^k = \boldsymbol{x}^k + ((q_{k-1} - 1)/q_k)(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$

The price of the convenience of using a general denoiser is that PnP algorithms lose interpretability as optimization methods for minimizing explicit convex objective functions. This complicates their theoretical analysis, a problem that was extensively addressed in several recent publications [17,18,22,24,27]. A distinct but related problem is the lack of a *regularization parameter* for adjusting the relative strength between the prior and the data fidelity. Unlike traditional regularized inversion, PnP does not have an *explicit* relationship between the regularization parameter and the intrinsic parameters of the denoiser. Instead, current PnP algorithms rely on indirect proxies for adjusting the relative strength of the prior, such as the noise-variance parameter (available for some denoisers) or the parameters of the iterative algorithms. However, many denoisers, such as those based on CNNs, do no have tunable parameters controlling their influence within PnP. This leads to their suboptimal performance, and requires training several CNN instances at multiple noise levels.

To address this issue, we introduce a tunable regularization parameter for PnP, which is independent from the intrinsic parameters of the denoiser or iterative algorithms. More specifically, the key contributions of this paper are as follows:

- We introduce a new *denoiser scaling* technique that simply scales the denoiser input by a positive constant and its output by the inverse of the same constant. The technique is broadly applicable to all PnP algorithms, and provides a mechanism to adjust the denoiser strength in a way that is independent of traditional approaches.

- We present a detailed theoretical justification of denoiser scaling for several classes of denoisers. We show that, unlike the intrinsic parameters of the denoiser, the new scaling parameter can be explicitly related to the trade-off between the data-fidelity and the prior.

- We extensively validate denoiser scaling by showing its potential to address the suboptimal performance of denoising CNNs within PnP algorithms. Our results show that denoiser scaling is a simple yet effective approach for boosting the performance of CNN priors within PnP.

2   Background

Consider the recovery of an unknown image $\boldsymbol{x} \in \mathbb{R}^n$ from noisy measurements $\boldsymbol{y} \in \mathbb{R}^m$. When this inverse problem is ill-posed, it is essential to include prior information on the unknown image to regularize the solution. A common approach is to formulate the problem as regularized inversion, expressed as an optimization problem of the form

$$\widehat{\boldsymbol{x}} = \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min}\, f(\boldsymbol{x}) \quad \text{with} \quad f(\boldsymbol{x}) = g(\boldsymbol{x}) + \lambda h(\boldsymbol{x}), \tag{1}$$

where g is the data-fidelity term, h is the regularizer, and $\lambda > 0$ is a regularization parameter that adjusts their relative strengths. For example, by setting

$$g(\boldsymbol{x}) = -\log(p_{\boldsymbol{y}|\boldsymbol{x}}(\boldsymbol{y}|\boldsymbol{x})) \quad \text{and} \quad h(\boldsymbol{x}) = -(1/\lambda)\log(p_{\boldsymbol{x}}(\boldsymbol{x})),$$

where $p_{\boldsymbol{y}|\boldsymbol{x}}$ denotes the likelihood function characterizing the imaging system and $p_{\boldsymbol{x}}$ denotes a probability distribution over $\boldsymbol{x}$, one obtains the classical *maximum-a-posteriori probability (MAP)* estimator. The regularized inversion framework (1) can accommodate a variety of data-fidelity and regularization terms. For

example, the combination of a linear measurement model under an additive white Gaussian noise (AWGN) and isotropic *total variation (TV)* regularization [1,11] is obtained by setting

$$g(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x}\|_2^2 \quad \text{and} \quad h(\boldsymbol{x}) = \sum_{i=1}^{n} \|[\boldsymbol{D}\boldsymbol{x}]_i\|_2,$$

where $\boldsymbol{D}$ is the discrete image gradient. Isotropic TV corresponds to the sparsity-promoting ℓ_1-norm prior on the magnitude of the image gradient.

A large number of regularizers used in the context of imaging inverse problems, including ℓ_1-norm and TV, are nonsmooth. Proximal algorithms [7] enable efficient minimization of nonsmooth functions, without differentiating them, by using the *proximal operator*, defined as

$$\mathsf{prox}_{\tau h}(\boldsymbol{z}) \;:=\; \underset{\boldsymbol{x}\in\mathbb{R}^n}{\arg\min} \left\{ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2 + \tau h(\boldsymbol{x}) \right\}, \tag{2}$$

where $\tau > 0$ is a scaling parameter that controls the influence of h. Note that the proximal operator can be interpreted as a MAP image denoiser for AWGN with variance of τ.

The observation that the proximal operator is an image denoiser for AWGN prompted the development of PnP [16], where the operator $\mathsf{prox}_{\tau h}(\cdot)$, within a proximal algorithm, is replaced with a more general image denoiser $\mathsf{D}(\cdot)$, such as BM3D [39] or DnCNN [40]. The PnP-ADMM algorithm [16] is summarized in Algorithm 1, where in analogy to $\mathsf{prox}_{\tau h}(\cdot)$ we also introduce the parameter $\sigma > 0$ for the denoiser $\mathsf{D}_\sigma(\cdot)$. An alternative algorithm, PnP-ISTA [20], is summarized Algorithm 2. When the values for $\{q_k\}$ in this algorithm are set such that $q_k = 1$ for all $k \geq 1$, the algorithm corresponds to the traditional form of ISTA [11]. Alternatively, when the values for $\{q_k\}$ are adapted as

$$q_k = \frac{1}{2}\left(1 + \sqrt{1 + 4q_{k-1}^2}\right), \tag{3}$$

the algorithm corresponds to the accelerated variant of ISTA, known as Fast ISTA (FISTA) [41]. In this paper, we will use the sequence $\{q_k\}$ as a mechanism for switching between the methods. As extensively discussed in [24], both PnP-ADMM and PnP-ISTA have the same set of fixed points, but the algorithms offer complimentary strategies for handling the data-fidelity term, and have different convergence behaviour. In particular, PnP-ADMM is known to be fast for forward operators that can be inverted efficiently [42–44], while PnP-ISTA is well suited to nonlinear forward models where $\mathsf{prox}_{\gamma g}(\cdot)$ is computationally expensive to evaluate [20,45].

In traditional proximal optimization, the scaling parameter τ of the proximal operator (2) is directly related to the regularization parameter λ. For example, by setting $\tau = \gamma\lambda$ within traditional ADMM or FISTA, one minimizes the objective function in (1). However, this explicit relationship between the scaling parameter and the regularization parameter is lost in the context of more general denoisers. Since some popular image denoisers, such as BM3D, accept a parameter corresponding to the noise variance, current PnP algorithms generally treat it as a proxy for the regularization parameter. For example, if σ in the notation for $\mathsf{D}_\sigma(\cdot)$ in Algorithm 1 and Algorithm 2 denotes the standard deviation parameter accepted by the denoiser, the common strategy is to set it as $\sigma = \sqrt{\gamma\lambda}$ [17]. However, this strategy does not work with all denoisers, since some do not have a dedicated parameter for noise variance. In particular, many denoising CNNs do not have a parameter for the noise standard deviation, which is often addressed by training multiple neural nets at different noise levels and using σ to select the most suitable one for a given problem. The denoiser scaling technique introduced in the next section enables the control of the regularization strength for denoisers that have no intrinsic parameters analogous to σ.

3 Proposed Method

We introduce denoiser scaling for explicitly controlling the regularization strength in PnP. Remarkably, the technique can be theoretically justified from multiple perspectives, including from that of proximal

optimization, statistical estimation, and consensus equilibrium [22]. Our experimental results in Section 4 corroborate the ability of denoiser scaling to control the relative influence of the denoiser.

3.1 Denoiser scaling

Consider an image denoiser $\mathsf{D} : \mathbb{R}^n \to \mathbb{R}^n$, where we omit the parameter σ from the notation as it is not available for all denoisers. We define the scaled denoiser as

$$\mathsf{D}_\mu(\boldsymbol{z}) \coloneqq (1/\mu)\mathsf{D}(\mu\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n, \tag{4}$$

where we will refer to the parameter $\mu > 0$ as the *denoiser scaling parameter*. In the rest of this paper, we assume that the denoisers in line 4 of Algorithm 1 and Algorithm 2 correspond to the scaled denoiser $\mathsf{D}_\mu(\cdot)$. Note that the scaling in (4) is *complimentary* to any intrinsic parameter of $\mathsf{D}(\cdot)$. For example, if the underlying denoiser $\mathsf{D}(\cdot)$ additionally accepts σ as a parameter, $\mathsf{D}_\mu(\cdot)$ will also accept the same parameter. However, as discussed below, the parameter μ will enable control of the strength of regularization when σ is not available.

3.2 Proximal operator denoisers

We first consider the case in which $\mathsf{D}(\cdot)$ is an *implicit* proximal operator of some *unknown* h, which is a common interpretation for PnP algorithms [18]. For convenience, we assume h to be closed, convex, and proper [7]; however, this assumption can be dropped as long as $\mathsf{prox}_h(\cdot)$ is well defined for the given h. We state the following result for the scaled denoisers.

Proposition 1. Suppose $\mathsf{D}(\boldsymbol{z}) = \mathsf{prox}_h(\boldsymbol{z})$, where h is a closed, convex, and proper function. Then, we have

$$\mathsf{D}_\mu(\boldsymbol{z}) = \mathsf{prox}_{\mu^{-2}h(\mu\cdot)}(\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{5}$$

Proof. See Appendix A. $\qquad\square$

Proposition 1 indicates that by scaling an implicit proximal operator, one directly adjusts the strength of regularization via the scaling of the regularizer by $1/\mu^2$ and of the input to h by μ. While the relationship between μ and the regularization parameter in front of h is not linear, μ still provides an explicit mechanism to tune the denoiser. If the denoiser corresponds to a 1-*homogeneous* regularizer h, we have $h(\mu\cdot) = \mu \cdot h(\cdot)$, which directly implies

$$\mathsf{D}_\mu(\boldsymbol{z}) = \mathsf{prox}_{\mu^{-1}h}(\boldsymbol{z}), \quad \boldsymbol{z} \in \mathbb{R}^n. \tag{6}$$

This means that the denoiser scaling becomes equivalent to tuning the traditional regularization parameter in regularized inversion. Since any norm and semi-norm is 1-homogeneous, the strength of many implicit and explicit regularizers, such as the ℓ_1-norm or TV penalty, can be directly adjusted through denoiser scaling. This equivalence is confirmed numerically for the TV denoiser in Section 4.

3.3 Mean squared error optimal denoisers

We now consider the case of a denoiser that performs the minimum mean-squared error (MMSE) estimation of a vector from its AWGN corrupted version [46–48]. MMSE denoisers are optimal with respect to the ubiquitous image-quality metrics, such as signal-to-noise ratio (SNR). Additionally, many popular denoisers (such as BM3D and certain denoising CNNs) are often interpreted as *empirical* MMSE denoisers. We state the following result for the scaled MMSE denoisers.

Proposition 2. Suppose $\mathsf{D}(\cdot)$ computes the MMSE solution of the following denoising problem

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{n} \quad \text{with} \quad \boldsymbol{x} \sim p_{\boldsymbol{x}} \quad \text{and} \quad \boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),$$

where $\boldsymbol{I}$ is an identity matrix. Then, the denoiser (4) computes the MMSE solution of

$$\boldsymbol{z} = \boldsymbol{u} + \boldsymbol{e} \quad \text{with} \quad \boldsymbol{u} \sim p_{\boldsymbol{x}}(\mu\cdot) \quad \text{and} \quad \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \mu^{-2}\boldsymbol{I}).$$

Figure 1: Test images used for the quantitative performance evaluation. From left to right: *Cameraman, House, Pepper, Starfish, Butterfly, Plane, Parrot.*

Table 1: SNR performances of several image denoisers at different noise levels.

Input SNR	TV		BM3D			DnCNN*		
	Scaled	Optimized	Unscaled	Scaled	Optimized	Unscaled	Scaled	Optimized
15 dB	22.77	22.77	16.58	24.55	24.51	16.49	24.30	24.21
20 dB	25.80	25.80	25.58	27.34	27.34	24.83	27.42	27.23
25 dB	29.09	29.09	29.61	30.43	30.43	29.83	30.51	30.35
30 dB	32.66	32.66	33.49	33.63	33.73	33.54	33.82	33.72

Proof. See Appendix B. □

Similarly to Proposition 1, Proposition 2 indicates that the scaling of the MMSE denoiser enables the direct control of the strength of regularization via the scaling of the noise variance by $1/\mu^2$ and of the input to the prior $p_{\boldsymbol{x}}$ by μ. If $p_{\boldsymbol{x}}$ assigns equal probabilities to all images that have undergone rescaling, we have $p_{\boldsymbol{x}}(\mu \cdot) = p_{\boldsymbol{x}}(\cdot)$, which implies that the scaled denoiser directly adjusts the variance of AWGN in the MMSE estimation.

3.4 Consensus equilibrium interpretation

Consensus equilibrium (CE) [22] is a recent framework for interpreting the solution of regularized inversion methods in terms of a set of balancing equations for the forward and prior models, without an explicit cost function The solutions obtained by both PnP-ADMM and PnP-ISTA can be expressed in terms of the same set of CE equations

$$\boldsymbol{x} = \mathsf{F}(\boldsymbol{x} + \boldsymbol{s}) \tag{7a}$$
$$\boldsymbol{x} = \mathsf{D}(\boldsymbol{x} - \boldsymbol{s}), \tag{7b}$$

where $\mathsf{F}(\cdot) \coloneqq \mathsf{prox}_{\gamma g}(\cdot)$ and $\gamma > 0$ is an algorithm tuning parameter. We use the CE framework to state the following result for the scaled denoisers.

Proposition 3. Let g be a smooth, convex function and $\mathsf{D}(\cdot)$ be a continuous denoiser. The fixed point $(\boldsymbol{x}, \boldsymbol{s})$ of PnP-ADMM and PnP-ISTA for the scaled denoiser satisfies

$$\begin{cases} \mu\boldsymbol{x} = \mathsf{prox}_{(\gamma\mu^2)g(\cdot/\mu)}(\mu\boldsymbol{x} + \boldsymbol{s}) \\ \mu\boldsymbol{x} = \mathsf{D}(\mu\boldsymbol{x} - \boldsymbol{s}). \end{cases}$$

Proof. See Appendix C. □

This result establishes a direct relationship between the scaling parameter $\mu > 0$ and the rescaling of g with respect to the denoiser. Note that while Proposition 1 and 2 discuss the impact of denoisier scaling on the implicit prior, Proposition 3 highlights its impact on the relative influence between the denoiser and the data-fidelity term via the weighting in front of g. Since our only assumption is the continuity of the denoiser, Proposition 3 also relaxes the assumptions on the denoiser. While the relationship between μ and the set of equilibrium points is nontrivial, the proposition implies that one can still adjust the amount of regularization by tuning μ.
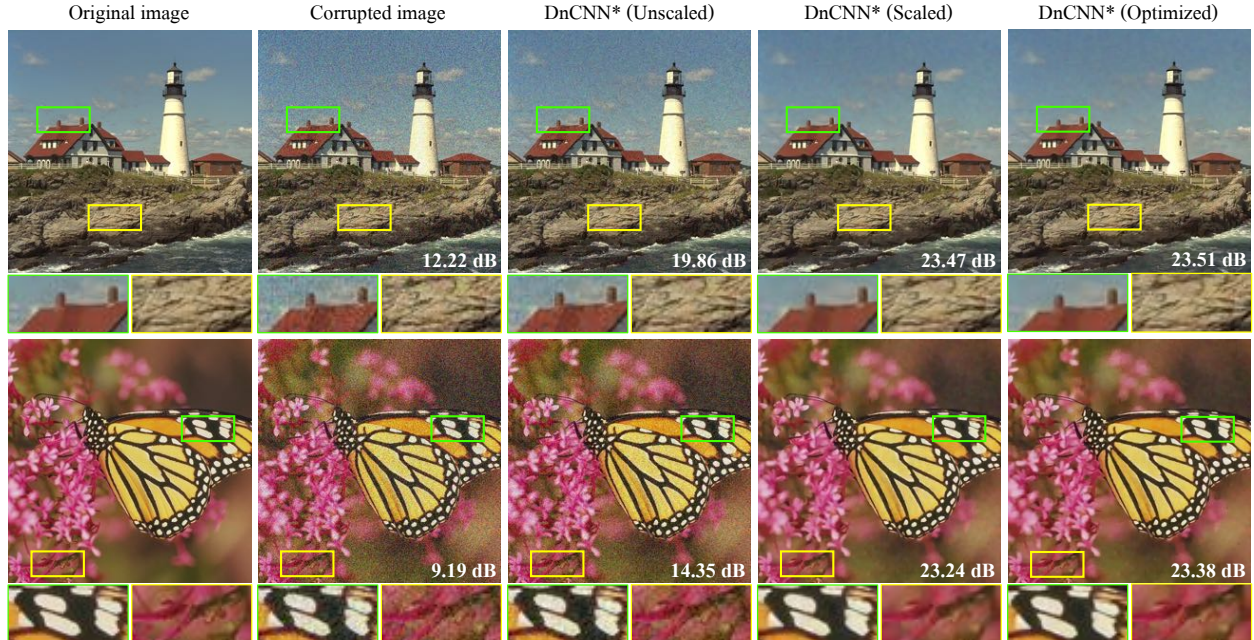
Figure 2: Illustration of denoiser scaling on two color images: *Lighthouse* (top) and *Monarch* (bottom). The noise levels in the top and bottom rows are $\sigma = 30$ and $\sigma = 40$, respectively. DnCNN* (Optimized) corresponds to the CNN denoiser trained using the correct noise levels. On the other hand, DnCNN* (Unscaled) and DnCNN* (Scaled) use the same CNN trained at a mismatched noise level of $\sigma = 20$. By adjusting μ, DnCNN* trained at a suboptimal σ can be made to match the performance of DnCNN* trained using the correct noise level.

4 Numerical Validation

In this section, we demonstrate the ability of denoiser scaling to boost the performance of PnP. Our experiments consider imaging inverse problems of the form $\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{e}$, where $\boldsymbol{y} \in \mathbb{R}^m$ denotes the measurements, $\boldsymbol{e} \in \mathbb{R}^m$ is a vector of AWGN with zero mean and standard deviation σ, and $\boldsymbol{H} \in \mathbb{R}^{m \times n}$ denotes the forward operator. We focus on three inverse problems: image denoising, subsampled Fourier inversion, which commonly used in magnetic resonance imaging (MRI), and single image super-resolution (SR), where the forward models correspond to the identity matrix, radially subsampled two-dimensional Fourier transform, and blurring-downsampling operator, respectively. For each simulation, the measurements are corrupted with AWGN quantified through the input signal-to-noise ratio (SNR). We will consider three denoisers for PnP: TV, BM3D, and our own residual DnCNN*. DnCNN* is a simplified variant of the standard DnCNN [40], where our simplifications correspond to the removal of batch-normalization layers and reduction in the total number of layers (see Appendix D for details). This simplification reduces the computational cost of applying the denoiser across multiple PnP iterations. We use σ to parameterize BM3D and DnCNN*. For BM3D σ represents the parameter of the denoiser represensing the standard deviation of noise and for DnCNN* it represents the standard deviation of the noise used for training the CNN. We follow [49] and use 400 images of size 180×180 to train three DnCNN* instances, on natural grayscale images, for the removal of AWGN at three noise levels, $\sigma \in \{1, 5, 10\}$. For some of the experiments, we also use a constrained variant of BM3D, where σ is restricted to the same set. For the medical knee images, we follow the work in [38] to train our 7-layer DnCNN* on NYU fastMRI dataset [50] for $\sigma \in \{1, 5, 10\}$. For color images, we use DnCNN* trained on 4744 images from the Waterloo Exploration Database [51] and use the CBM3D denoiser [52]. For all experiments in SR and Fourier, we use PnP-FISTA as the reconstruction algorithm. All the quantitative
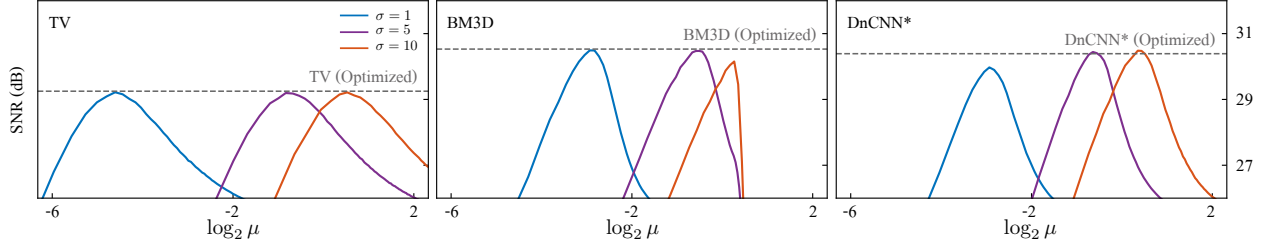
Figure 3: The influence of the scaling parameter μ on the denoising performance for *Pepper* for AWGN with input SNR of 25 dB ($\sigma = 7.23$). We show the SNR evolution against μ for the variants of TV, BM3D, and DnCNN* designed for the mismatched noise levels. The horizontal line shows the performance of the corresponding denoiser optimized for input SNR of 25 dB. Note how by adjusting μ, one can achieve nearly optimal performance for all three denoisers.

Table 2: Average SNRs obtained for different inverse problems and image denoisers.

Forward model	Input SNR	TV		BM3D			DnCNN*	
		Scaled	Optim	Unscaled	Scaled	Optim	Unscaled	Scaled
Fourier	30 dB	27.15	27.15	27.29	28.49	28.51	27.92	28.44
	40 dB	27.79	27.79	28.97	29.17	29.21	29.72	29.89
Super-resolution	30 dB	19.86	19.87	14.32	20.65	20.64	13.59	20.05
	40 dB	22.42	22.41	22.64	22.78	22.71	23.03	23.07

results in the tables are averaged over seven test images shown in Figure 1 with hyperparameters optimized individually for each image for the best SNR performance using grid search. All visual results are shown with SNR value displayed directly on the images. None of the test images were used in training.

Table 1 shows the ability of denoiser scaling to adjust the denoising strength for three different image denoisers at four input SNR levels: 15 dB, 20 dB, 25 dB and 30 dB. In the table, TV (Optimized) is obtained by tuning the regularization parameter λ for each test image. For TV (Scaled), we fix $\lambda = 1$ and tune the scaling parameter μ for the best result. The performances of BM3D (Unscaled and Scaled) and DnCNN* (Unscaled and Scaled) correspond to the best instance selected from the limited set of $\sigma \in \{1, 5, 10\}$. For reference, we also show BM3D (Optimized), which uses fully optimized σ, and DnCNN (Optimized), which is trained on noisy images with the true input SNR. Table 1 highlights the equivalence between TV with an optimized λ and TV with $\lambda = 1$, but optimized μ, which validates eq. (6). Table 1 also shows that denoiser scaling significantly improves the performance of sub-optimally tuned BM3D and DnCNN* to achieve the performance of the corresponding denoiser with optimized σ.

Figure 2 visually illustrates the performance of denoiser scaling on the problem of color image denoising for AWGN of $\sigma = 30$ (top) and $\sigma = 40$ (bottom). In the figure, DnCNN* (Optimized) denotes to the denoiser trained using the dataset with the correct noise level. On the other hand, DnCNN* (Unscaled) and DnCNN* (Scaled) correspond to the same CNN instance, trained for noise level $\sigma = 20$. DnCNN* (Unscaled) uses $\mu = 1$, while DnCNN* (Scaled) optimizes μ for the best SNR performance. By simply adjusting μ, the suboptimaly trained DnCNN* achieves the performance of DnCNN* trained using the correct noise level.

Figure 3 considers the problem of denoising an image with the input SNR of 25 dB, which corresponds to the noise level $\sigma = 7.23$. The figure shows the influence of the parameter μ for improving the performance of denoisers at mismatched values of $\sigma = 1$, 5, and 10. The SNR value after denoising is plotted against the logarithm of μ. For TV, we fixed the regularization parameter λ to its optimal value for the noise levels $\sigma \in \{1, 5, 10\}$ and then adjusted the parameter μ. For each plot, we also provide the performance of the denoisers with optimized σ. The comparison between unscaled denoisers at $\mu = 1$ and their scaled
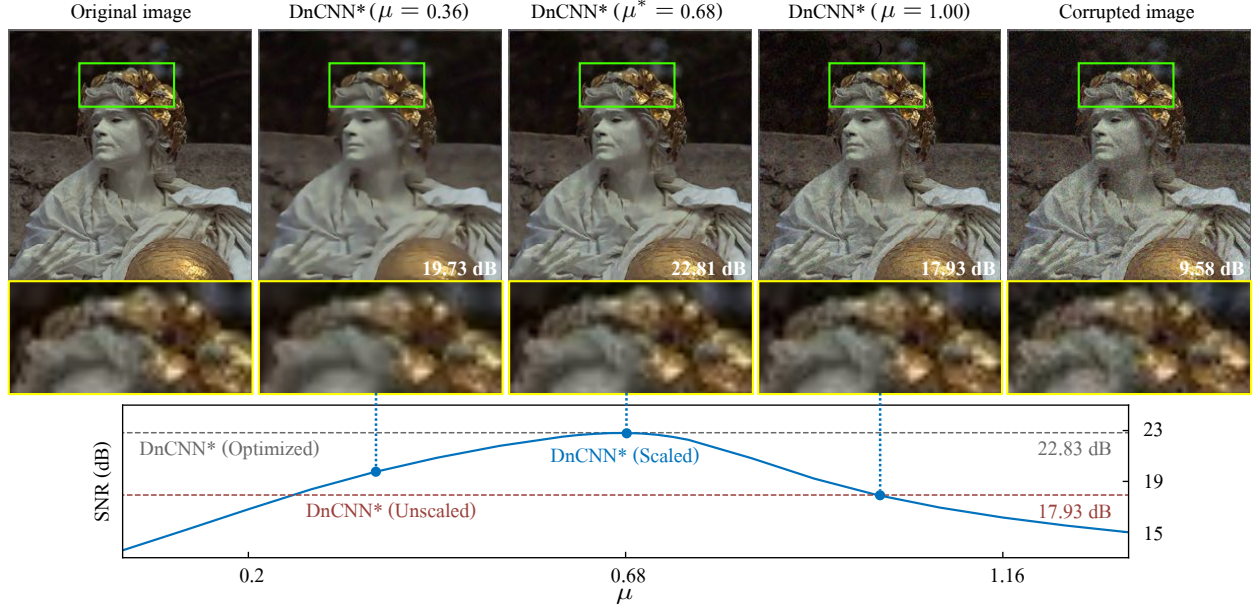
Figure 4: The influence of the denoiser scaling parameter μ on the denoising performance of DnCNN* on the color image *Statue*. The noise in the image corresponds to $\sigma = 30$, while DnCNN* was trained for the removal of noise corresponding to $\sigma = 20$. The top row images illustrate the visual performance at μ values of 0.36, 0.68, and 1.00. The bottom plot shows the SNR evolution against the parameter μ for a wider range of values. The scaled DnCNN* achieves its best performance at $\mu^* = 0.68$. Note how unscaled DnCNN* ($\mu = 1.00$) leads to an insufficient amount of regularization, while a smaller scaling parameter $\mu = 0.36$ leads to oversmoothing. This figure highlights the ability of μ to control the strength of regularization with a CNN denoiser.

counterparts demonstrate the potential of denoiser scaling to influence the final performance, validating the theoretical conclusion that the denoiser scaling directly controls the strength of regularization. This is particularly appealing for DnCNN*, which does not have a tunable parameter σ.

Figure 4 visually and quantitatively illustrates the influence of the parameter μ for color image denoising. The DnCNN* trained on noise level $\sigma = 20$ is applied for denoising an image with a noise level $\sigma = 30$. The performance of scaled DnCNN achieves its peak SNR at $\mu^* = 0.68$ and leads to either under- or over-regularization at either side of this value.

Table 2 shows results of image reconstruction for both Fourier and SR under two input noise levels 30 dB and 40 dB. Here, we fix $\sigma = 1$ for both BM3D and DnCNN*. We aslo show BM3D (Optimized), which uses fully optimized σ. For Fourier, the measurement ratio is set to be approximately $m/n = 1/3$. For SR, the low resolution (LR) image is simulated by convolving the high resolution image (HR) with a motion-blur kernel of size 19×19 from [53], followed by down-sampling with scale factor 2. Note that TV (Scaled) has a fixed $\lambda = 1$ and a scaling parameter μ optimized for each test image. Some visual results are shown in Figures 5, 6, and 7. These results clearly highlight the potential of denoiser scaling to significantly boost the performance of PnP. The results also highlight that the amount of improvement depends on the inverse problem in question (compare, for example, the results for SR under 30 dB and 40 dB of noise). This is expected since a specific regularizer might be already well suited for a given forward model (even without any denoiser scaling).
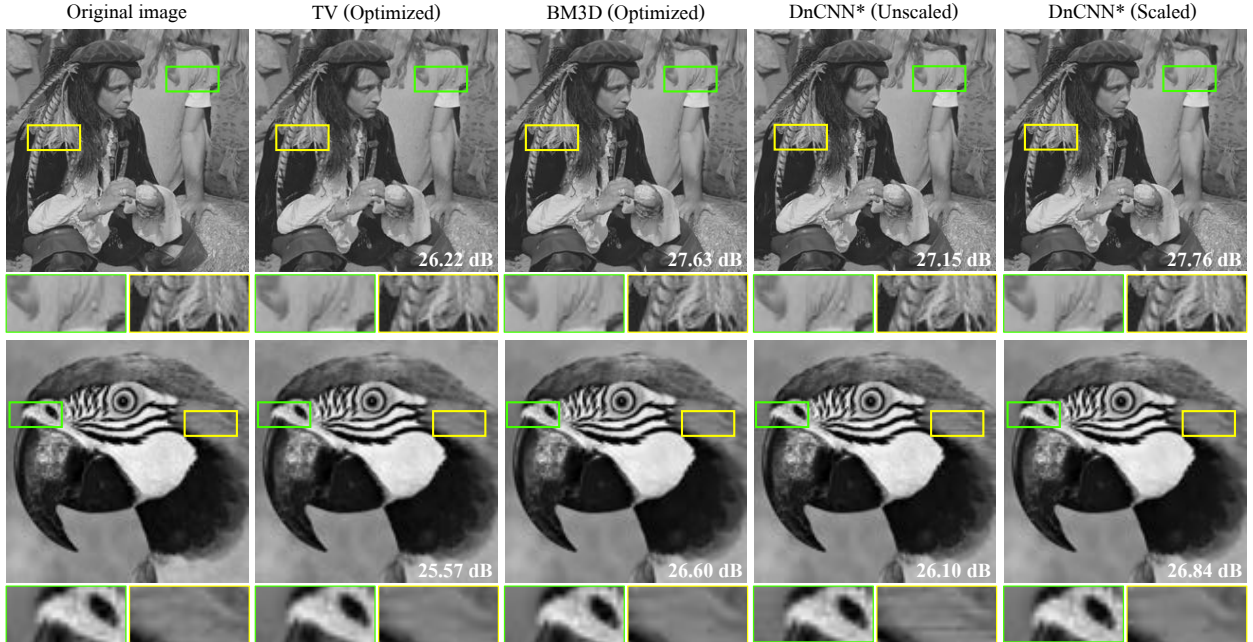
8

Figure 5: Visual illustration of denoiser scaling on the subsampled Fourier operator and two natural images: *Man* (top) and *Parrot* (bottom). The sampling rate is $m/n = 1/3$ and input SNR is 30 dB. DnCNN* is selected from $\sigma \in \{1, 5, 10\}$ that produces the best SNR performance. DnCNN* (Scaled) relies on the same CNN selected by DnCNN* (Unscaled). This result illustrates the potential of DnCNN* (Scaled) to boost the performance of DnCNN* (Unscaled) by reducing the artifacts.

5   Conclusion

We have presented a simple, but effective, technique for improving the performance of PnP algorithms. The approach is justified theoretically by connecting the denoiser scaling with the strength of the effective regularization introduced by PnP. The proposed technique is shown to be particularly valuable when PnP is used with CNN denoisers that have no explicit tunable parameters. Our experimental results show the potential of denoiser scaling to significantly improve the performance of PnP algorithms across several inverse problems. While we have focused on PnP, the denoiser scaling approach can be applied more broadly to improve the performance of related methods, such as AMP and RED.

A   Proof of Proposition 1

The proof below is a direct consequence of definition (2). It is similar to the derivation of other properties of the proximal operator that have been extensively described in the literature (see for example [54, Ch. 6]).
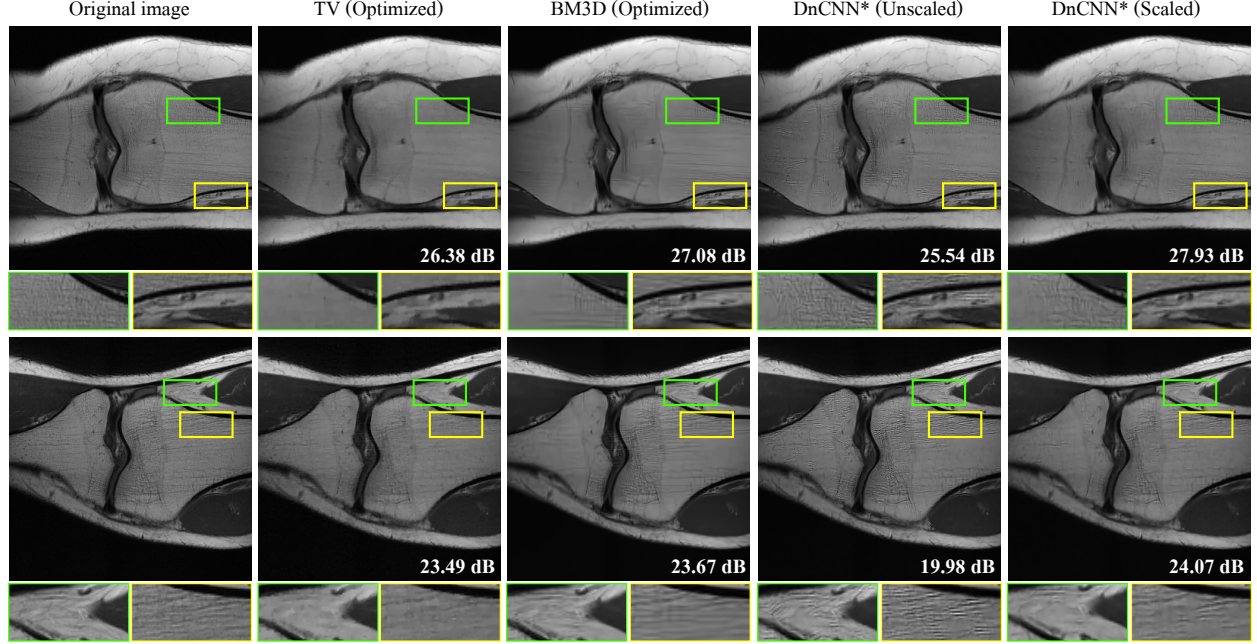
Figure 6: Visual illustration of denoiser scaling on the subsampled Fourier operator and two medical images: *Knee 1* (top) and *Knee 2* (bottom) from the fastMRI dataset. The sampling ration is $m/n = 1/3$ and input SNR is 30 dB. DnCNN* is selected from $\sigma \in \{1, 5, 10\}$ that produces the best SNR performance. DnCNN* (Scaled) relies on the same CNN selected by DnCNN* (Unscaled). Note how DnCNN* (Scaled) improves the visual quality of results compared to DnCNN* (Unscaled).

For any $\boldsymbol{z} \in \mathbb{R}^n$, we have that

$$
\begin{aligned}
\mathsf{D}_\mu(\boldsymbol{z}) &= (1/\mu) \cdot \mathsf{prox}_h(\mu \cdot \boldsymbol{z}) \\
&= (1/\mu) \cdot \arg \min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|\boldsymbol{x} - \mu \boldsymbol{z}\|_2^2 + h(\boldsymbol{x}) \right\} \\
&= (1/\mu) \cdot \arg \min_{\boldsymbol{x}} \left\{ \frac{1}{2} \|(\boldsymbol{x}/\mu) - \boldsymbol{z}\|_2^2 + (1/\mu^2) h(\boldsymbol{x}) \right\} \\
&= (1/\mu) \cdot \mu \cdot \arg \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \|\boldsymbol{u} - \boldsymbol{z}\|_2^2 + (1/\mu^2) h(\mu \boldsymbol{u}) \right\} \\
&= \mathsf{prox}_{(1/\mu^2)h(\mu \cdot)}(\boldsymbol{z}),
\end{aligned}
$$

where in the second and the last lines we used the definition of the proximal operator, and in the fourth line we performed the variable change $\boldsymbol{u} = \boldsymbol{x}/\mu$.
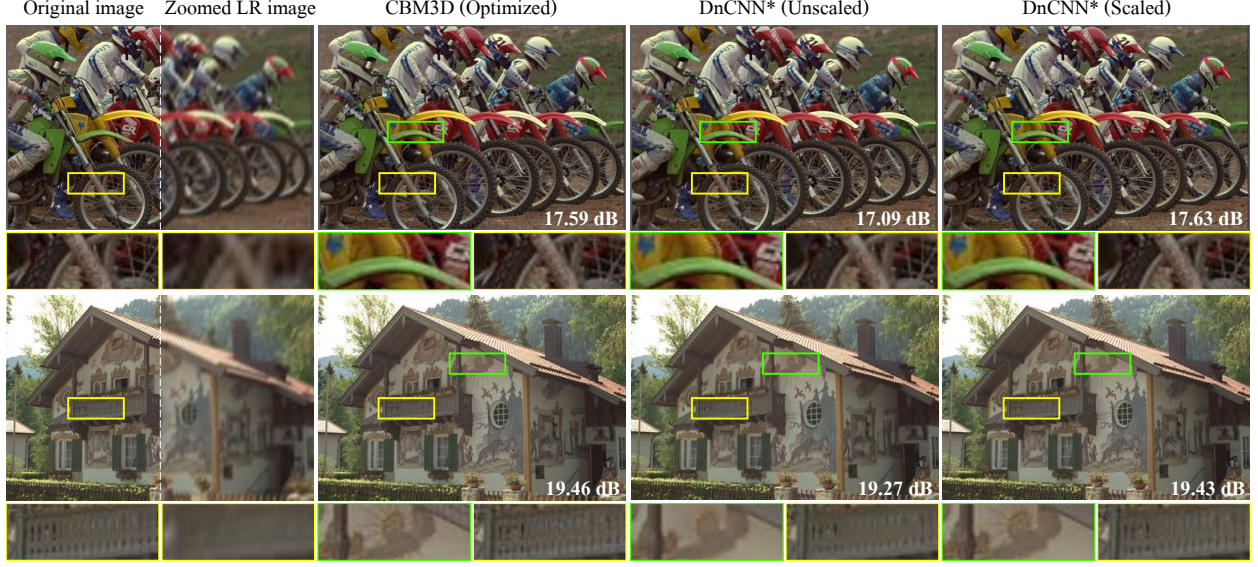
Figure 7: Results of SR simulation on color image *Bikes* (top) and *Paintedhouse* (bottom). The input SNR is 40 dB. DnCNN* is selected from $\sigma \in \{1, 5, 10\}$ that produces the best SNR performance. DnCNN* (Scaled) relies on the same CNN selected by DnCNN* (Unscaled). One can see while DnCNN* blur out the details in some regions, DnCNN* (Scaled) can generate images with more details and sharper edges.

B   Proof of Proposition 2

This result is a direct consequence of the definition of the MMSE denoiser. For any $\boldsymbol{z} \in \mathbb{R}^n$, we have that

$$
\begin{aligned}
\mathsf{D}_\mu(\boldsymbol{z}) &= (1/\mu) \cdot \mathsf{D}(\mu \cdot \boldsymbol{z}) \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} \boldsymbol{x}\phi_1(\boldsymbol{x} - \mu\boldsymbol{z})p_{\boldsymbol{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int_{\mathbb{R}^n} \phi_1(\boldsymbol{x} - \mu\boldsymbol{z})p_{\boldsymbol{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}} \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} \boldsymbol{x}\phi_{(1/\mu^2)}(\boldsymbol{x}/\mu - \boldsymbol{z})p_{\boldsymbol{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int_{\mathbb{R}^n} \phi_{(1/\mu^2)}(\boldsymbol{x}/\mu - \boldsymbol{z})p_{\boldsymbol{x}}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}} \\
&= (1/\mu) \cdot \frac{\int_{\mathbb{R}^n} (\mu\boldsymbol{u})\phi_{(1/\mu^2)}(\boldsymbol{u} - \boldsymbol{z})p_{\boldsymbol{x}}(\mu\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}}{\int_{\mathbb{R}^n} \phi_{(1/\mu^2)}(\boldsymbol{u} - \boldsymbol{z})p_{\boldsymbol{x}}(\mu\boldsymbol{u})\,\mathrm{d}\boldsymbol{u}},
\end{aligned}
$$

where we defined the probability density function of AWGN of variance $\nu > 0$ as

$$
\phi_\nu(\boldsymbol{x}) := \frac{1}{\sqrt{2\pi\nu}}\mathrm{e}^{-\frac{\|\boldsymbol{x}\|^2}{2\nu}}.
$$

The final line corresponds to the MMSE estimate of a random variable $\boldsymbol{u} \sim p_{\boldsymbol{x}}(\mu\cdot)$ from AWGN of variance $1/\mu^2$.
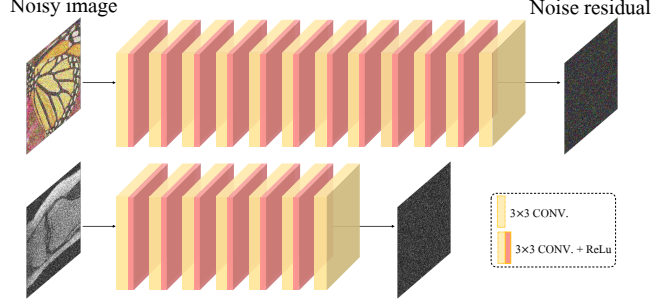
Figure 8: The architecture of two variants of DnCNN* we use in our simulations. DnCNN* (top) is applied on natural images, and DnCNN* (bottom) is applied on the medical knee images. Both neural nets are trained to predict the AWGN from the input. The final desired denoiser D is obtained by simply subtracting the predicted noise from the input $\mathsf{D}(\boldsymbol{x}) = \boldsymbol{x} - \mathrm{DnCNN}^*(\boldsymbol{x})$.

C   Proof of Proposition 3

It has already been shown that for any continuous denoiser both PnP-ADMM and PnP-ISTA have the same set of fixed points [24]. Consider any fixed point $\boldsymbol{x}$ of PnP-ISTA

$$
\begin{aligned}
& \boldsymbol{x} = \mathsf{D}_\mu(\boldsymbol{x} - \gamma \nabla g(\boldsymbol{x})) \\
\Leftrightarrow \quad & \mu \boldsymbol{x} = \mathsf{D}(\mu \boldsymbol{x} - \gamma \mu \nabla g(\boldsymbol{x})) \\
\Leftrightarrow \quad & \begin{cases} \mu \boldsymbol{x} = \mathsf{D}(\mu \boldsymbol{x} - \boldsymbol{z}) \\ \boldsymbol{z} = \gamma \mu \nabla g(\boldsymbol{x}) \end{cases} \\
\Leftrightarrow \quad & \begin{cases} \mu \boldsymbol{x} = \mathsf{D}(\mu \boldsymbol{x} - \boldsymbol{z}) \\ \mu \boldsymbol{x} = \mathsf{prox}_{(\gamma \mu^2)g(\cdot/\mu)}(\mu \boldsymbol{x} + \boldsymbol{z}), \end{cases}
\end{aligned}
$$

which directly leads to the result. To see the last equivalence, assume that g is a smooth and convex function and note that

$$
\begin{aligned}
\mu \boldsymbol{x} &= \mathsf{prox}_{(\mu^2 \gamma)g(\cdot/\mu)}(\mu \boldsymbol{x} + \boldsymbol{z}) \\
&= \arg \min_{\boldsymbol{u}} \left\{ \frac{1}{2} \|\boldsymbol{u} - (\mu \boldsymbol{x} + \boldsymbol{z})\|_2^2 + (\gamma \mu^2)g(\boldsymbol{u}/\mu) \right\} \\
\Leftrightarrow \quad & \mu \boldsymbol{x} - (\mu \boldsymbol{x} + \boldsymbol{z}) + (\gamma \mu^2) \cdot (1/\mu) \cdot \nabla g(\mu \boldsymbol{x}/\mu) = \boldsymbol{0} \\
\Leftrightarrow \quad & \boldsymbol{z} = \gamma \mu \nabla g(\boldsymbol{x}),
\end{aligned}
$$

where we used the optimality conditions.

D   Architecture of the DnCNN* denoiser

Two variants of the residual DnCNN*, shown in Figure 8, are used in our simulations. The DnCNN* of 12 convolutional layers is used for natural images. The DnCNN* of 7 layers from [38] is used for the knee images from the NYU fastMRI dataset [50]. The latter has a bounded Lipschitz constant $L = 2$, providing a necessary but not sufficient condition for D to be a nonexpansive denoiser. As discussed in [38], the Lipschitz constant is controlled via spectral-normalization [55]. In both neural nets, the first layer convolves the $H \times W \times C$ input to $H \times W \times 64$ features maps by using 64 filters of size 3×3. Here W and H represents the height and width of input image, and C represents the number of image channels. For example, $C = 1$ for gray image and $C = 3$ for color image. The last layer is a single convolutional layer, generating the

final output image with C channels by convolving the feature maps with a $3 \times 3 \times 64$ filter. The layers in-between are convolutional layers, each having 64 filters of size $3 \times 3 \times 64$. All layers except for the last one is followed by a rectified linear units (ReLU) layer. Every convolution is performed with a stride $= 1$, so that the intermediate feature maps share the same spatial size of the input image.

References

[1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.

[2] M. A. T. Figueiredo and R. D. Nowak, "Wavelet-based image estimation: An empirical bayes approach using Jeffreys' noninformative prior," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1322–1331, Sep. 2001.

[3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[4] A. Danielyan, V. Katkovnik, and K. Egiazarian, "BM3D frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, April 2012.

[5] Z. Yang and M. Jacob, "Nonlocal regularization of inverse problems: A unified variational framework," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3192–3203, Aug. 2013.

[6] U. S. Kamilov, "A parallel proximal algorithm for anisotropic total variation minimization," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 539–548, Feb. 2017.

[7] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, 2014.

[8] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.

[9] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[10] J. Bect, L. Blanc-Feraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. ECCV*, Springer, Ed., vol. 3024, New York, 2004, pp. 1–13.

[11] A. Beck and M. Teboulle, "Fast gradient-based algorithm for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[12] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, pp. 293–318, 1992.

[13] M. V. Afonso, J. M.Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[14] M. K. Ng, P. Weiss, and X. Yuan, "Solving constrained total-variation image restoration and reconstruction problems via alternating direction methods," *SIAM J. Sci. Comput.*, vol. 32, no. 5, pp. 2710–2736, Aug. 2010.

[15] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[16] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Process. and Inf. Process. (GlobalSIP)*, Austin, TX, USA, December 3-5, 2013, pp. 945–948.

[17] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comp. Imag.*, vol. 3, no. 1, pp. 84–98, Mar. 2017.

[18] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comp. Imag.*, vol. 2, no. 4, pp. 408–423, Dec. 2016.

[19] S. Ono, "Primal-dual plug-and-play image restoration," *IEEE Signal Process. Lett.*, vol. 24, no. 8, pp. 1108–1112, 2017.

[20] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, Dec. 2017.

[21] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep CNN denoiser prior for image restoration," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[22] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization free reconstruction using consensus equilibrium," *SIAM J. Imaging Sci.*, vol. 11, no. 3, pp. 2001–2020, 2018.

[23] A. M. Teodoro, J. M. Bioucas-Dias, and M. Figueiredo, "A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 451–463, Jan. 2019.

[24] Y. Sun, B. Wohlberg, and U. S. Kamilov, "An online plug-and-play algorithm for regularized image reconstruction," *IEEE Trans. Comput. Imaging*, vol. 5, no. 3, pp. 395–408, Sep. 2019.

[25] T. Tirer and R. Giryes, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1220–1234, 2019.

[26] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. IEEE Int. Conf. Comp. Vis. (ICCV)*, Venice, Italy, October 22-29, 2017, pp. 1799–1808.

[27] E. K. Ryu, J. Liu, S. Wnag, X. Chen, Z. Wang, and W. Yin, "Plug-and-play methods provably converge with properly trained denoisers," in *Proc. 36th Int. Conf. Machine Learning (ICML)*, Long Beach, CA, USA, Jun. 2019.

[28] G. Song, Y. Sun, J. Liu, Z. Wang, and U. S. Kamilov, "A new recurrent plug-and-play prior based on the multiple self-similarity network," *IEEE Signal Process. Lett.*, 2020, arXiv:1907.11793 [eess.IV].

[29] J. Tan, Y. Ma, and D. Baron, "Compressive imaging via approximate message passing with image denoising," *IEEE Trans. Signal Process.*, vol. 63, no. 8, pp. 2085–2092, Apr. 2015.

[30] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[31] C. A. Metzler, A. Maleki, and R. Baraniuk, "BM3D-PRGAMP: Compressive phase retrieval based on BM3D denoising," in *Proc. IEEE Int. Conf. Image Proc.*, Phoenix, AZ, USA, September 25-28, 2016, pp. 2504–2508.

[32] A. Fletcher, S. Rangan, S. Sarkar, and P. Schniter, "Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis," in *Proc. Advances in Neural Information Processing Systems 32*, Montréal, Canada, 2018.

[33] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imaging Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[34] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," in *Proc. Advances in Neural Information Processing Systems 31*, Long Beach, CA, USA, Dec. 2017.

[35] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, "prDeep: Robust phase retrieval with a flexible deep network," in *Proc. 35th Int. Conf. Machine Learning (ICML)*, Stockholm, Sweden, Jun. 2018.

[36] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE Trans. Comput. Imag.*, vol. 5, no. 1, pp. 52–67, Mar. 2019.

[37] G. Mataev, P. Milanfar, and M. Elad, "DeepRED: Deep image prior powered by RED," in *Proc. IEEE Int. Conf. Comp. Vis. Workshops (ICCVW)*, Oct. 2019.

[38] Y. Sun, J. Liu, and U. S. Kamilov, "Block coordinate regularization by denoising," in *Proc. Advances in Neural Information Processing Systems 33*, Vancouver, BC, Canada, Dec. 2019.

[39] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 16, pp. 2080–2095, Aug. 2007.

[40] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[41] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[42] A. Matakos, S. Ramani, and J. Fessler, "Accelerated edge-preserving image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 2019–2029, May 2013.

[43] M. Almeida and M. Figueiredo, "Deconvolving images with unknown boundaries using the alternating direction method of multipliers," *IEEE Trans. Ima*, vol. 22, no. 8, pp. 3074–3086, Aug. 2013.

[44] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 301–315, Jan. 2016.

[45] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, "Optical tomographic image reconstruction based on beam propagation and sparse regularization," *IEEE Trans. Comp. Imag.*, vol. 2, no. 1, pp. 59–70, Mar. 2016.

[46] U. S. Kamilov, P. Pad, A. Amini, and M. Unser, "MMSE estimation of sparse Lévy processes," *IEEE Trans. Signal Process.*, vol. 61, no. 1, pp. 137–147, Jan. 2013.

[47] A. Kazerouni, U. S. Kamilov, E. Bostan, and M. Unser, "Bayesian denoising: From MAP to MMSE using consistent cycle spinning," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 249–252, Mar. 2013.

[48] R. Gribonval and P. Machart, "Reconciling "priors" & "priors" without prejudice?" in *Proc. Advances in Neural Information Processing Systems 26*, Lake Tahoe, NV, USA, Dec. 2013, pp. 2193–2201.

[49] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[50] Zbontar *et al.*, "fastMRI: An open dataset and benchmarks for accelerated MRI," 2018, arXiv:1811.08839. [Online]. Available: http://arxiv.org/abs/1811.08839

[51] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1004–1016, 2016.

[52] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. Image Proc. (ICIP 2017)*, San Antonio, TX, USA, 2007.

[53] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov, "Image restoration using total variation regularized deep image prior," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Process.*, Brighton, UK, May 2019, pp. 7715–7719.

[54] A. Beck, *First-Order Methods in Optimization.* SIAM, 2017.

[55] H. Sedghi, V. Gupta, and P. M. Long, "The singular values of convolutional layers," in *International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA, May 2019.