
It Has Potential: Gradient-Driven Denoisers for Convergent Solutions to Inverse Problems

Regev Cohen

Verily Research, Israel
regevcohen@google.com

Yochai Blau

Google Research, Israel
yochaib@google.com

Daniel Freedman

Verily Research, Israel
danielfreedman@google.com

Ehud Rivlin

Verily Research, Israel
ehud@google.com

Abstract

In recent years there has been increasing interest in leveraging denoisers for solving general inverse problems. Two leading frameworks are regularization-by-denoising (RED) and plug-and-play priors (PnP) which incorporate explicit likelihood functions with priors induced by denoising algorithms. RED and PnP have shown state-of-the-art performance in diverse imaging tasks when powerful denoisers are used, such as convolutional neural networks (CNNs). However, the study of their convergence remains an active line of research. Recent works derive the convergence of RED and PnP methods by treating CNN denoisers as approximations for maximum a posteriori (MAP) or minimum mean square error (MMSE) estimators. Yet, state-of-the-art denoisers cannot be interpreted as either MAP or MMSE estimators, since they typically do not exhibit symmetric Jacobians. Furthermore, obtaining stable inverse algorithms often requires controlling the Lipschitz constant of CNN denoisers during training. Precisely enforcing this constraint is impractical, hence, convergence cannot be completely guaranteed. In this work, we introduce image denoisers derived as the gradients of smooth scalar-valued deep neural networks, acting as potentials. This ensures two things: (1) the proposed denoisers display symmetric Jacobians, allowing for MAP and MMSE estimators interpretation; (2) the denoisers may be integrated into RED and PnP schemes with backtracking step size, removing the need for enforcing their Lipschitz constant. To show the latter, we develop a simple inversion method that utilizes the proposed denoisers. We theoretically establish its convergence to stationary points of an underlying objective function consisting of the learned potentials. We numerically validate our method through various imaging experiments, showing improved results compared to standard RED and PnP methods, and with additional provable stability.

1 Introduction

General Recovery Inverse problems aim to recover an unknown signal from its observed corrupted or compressed version. This fundamental task appears in a variety of fields such as machine learning, signal processing, computer vision and medical imaging. When a model for the degradation process is available, one may derive a data fidelity term which relates the measurements to the unknown signal through a forward operator. However, this term alone typically leads to ill-posed optimization problems with unstable solutions, requiring the incorporation of additional knowledge of the desired

signal into the inversion process. This has given rise to diverse regularization methods which have been investigated extensively over the last few decades.

RED and PnP A very basic inverse problem is denoising, in which a clean signal is to be recovered from its measurements contaminated with additive noise. During the last period of time there has been an overwhelming advancement in the ability to perform denoising, creating techniques that approach optimal performance [11, 29, 30]. This striking progress in denoising has been the motivation for the plug-and-play priors (PnP) [48, 10, 41] methodology, which introduced the paradigm of using denoisers in order to solve general inverse problems. PnP incorporates implicit regularization into the inverse solutions by replacing proximal operators in common optimization algorithms with powerful denoisers. This simple modification has led to remarkable performance in diverse inverse tasks [41, 50, 38, 22, 56, 3, 51], as well as to the development of an arsenal of PnP-based techniques which includes proximal-type methods but also goes beyond them [33, 7, 18, 15, 5, 55, 14, 17]. However, the lack of a clearly formulated regularization term corresponding to the denoiser poses a serious challenge in ensuring the stability of PnP-based methods, and typically requires careful parameter-tuning [49].

To fill this void, the regularization by denoising (RED) [39] framework presents an explicit Laplacian-based regularization that relies on a denoiser. Under certain conditions on the denoiser [39, 37], the RED functional is convex and its gradient is given by the denoising residual, allowing for easy incorporation in standard optimization procedures. RED has demonstrated impressive recovery performance while providing global convergence guarantees. However, various common denoisers do not satisfy the conditions required by RED [37]. In such cases, there exists no objective function which justifies the RED method [37] and the corresponding convergence analysis does not hold. Thus RED, as well as PnP, shows state-of-the-art empirical results but without provable stability. A comprehensive overview of PnP and RED is given in [3].

Related Work The study of the stability of both RED and PnP methods has become an active line of research in recent years. The convergence of two PnP techniques was proven in [10] and [47] under the assumption of bounded denoisers. When minimum mean squared error (MMSE) denoisers are used, the PnP framework was shown to converge vis-a-vis an explicit objective function [53]. Similarly, [7, 18] proved the convergence of approximate message passing algorithms with MMSE denoisers for random forward operators. Multiple studies [41, 43, 44, 9, 45] derived the convergence of PnP by relying on nonexpansive denoisers. The latter has been the core assumption of [8], which explained the PnP framework, including its stability, through the notion of consensus equilibrium. A similar condition was used in [37], which adopted the score-matching framework to provide a new interpretation of the RED algorithms and their convergence. [42] presented a block-coordinate RED method and derived its fixed-point convergence for block-nonexpansive denoisers. The work of [36] introduced a convergent PnP-type method using a maximally monotone denoiser, created by training a nonexpansive neural network in a supervised fashion. Alternatively, [40] provided convergence guarantees for certain PnP variants that use trained denoisers whose residuals are nonexpansive. This type of denoiser was used in [27] to develop Bayesian inference with PnP priors, including the extension of the PnP framework to Monte Carlo sampling. The authors of [24, 23] employ denoisers within Langevin dynamics to sample from the posterior distribution of any linear inverse problem. A different strategy was studied in [35], which proposed a special inner product derived from the given linear denoiser, and thus proved the convergence of PnP based on the theory of averaged nonexpansive operators. Recent work [12] introduced RED via fixed-projection framework (RED-PRO), which utilizes the fixed-points of the denoisers as regularization. RED-PRO unifies RED and PnP and derives their convergence for general demicontractive denoisers, including nonexpansive ones.

Recent studies, mentioned above, are motivated by or rely on the assumption that the denoiser is an MMSE or maximum a posteriori (MAP) estimator. However, such estimators are in the form of gradients, hence, they must exhibit symmetric Jacobians [20, 19, 21] amongst other properties. Furthermore, Jacobian symmetry offers additional advantages as it often leads to stable algorithms and allows one to better understand the behavior of such non-linear estimators [34]. Unfortunately, many state-of-the-art denoisers do not fulfill this condition [37]. An additional drawback of current RED and PnP methods is that the Lipschitz constant of the denoiser often has to be controlled to create a nonexpansive operator, which in turn ensures convergence. This is typically done by learning a deep network and adding a regularization term to the training loss function, whose purpose is to bound the Lipschitz constant of the resultant denoiser. However, exactly enforcing this property of the denoiser is impractical and it is only approximated in practice.

Contributions In this work, we develop a provably convergent technique which exploits denoisers for solving general inverse problems. (1) We introduce image denoisers that are constructed as the gradients of smooth potential functions, represented by deep neural networks and trained in a supervised fashion. (2) Equipped with these learned potential networks, we formulate a general inverse framework where we utilize our denoising potentials as an explicit regularization. (3) We use the existence of an underlying objective function to incorporate the proposed denoisers into standard optimization schemes with backtracking step size. Specifically, we present a simple gradient descent algorithm where the step size is tuned automatically to provably guarantee convergence to stationary points of the objective function. (4) We numerically validate our method through imaging experiments, showing improved results compared to RED and PnP, with additional provable stability.

2 Background

Inverse Problems: Formulation We consider the recovery of an unknown signal $\mathbf{x}^* \in \mathbb{R}^{d_x}$ from its corrupted observation $\mathbf{y} \in \mathbb{R}^{d_y}$. The inversion task is traditionally formulated as an optimization problem of the form

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{d_x}} f(\mathbf{x}; \mathbf{y}) + \lambda g(\mathbf{x}), \quad (1)$$

where $f(\cdot; \mathbf{y})$ is a fidelity term related to the observations \mathbf{y} , $g(\cdot)$ is a regularization functional that incorporates prior information or known statistics of the true signal \mathbf{x}^* to avoid ill-posed problems, and $\lambda > 0$ balances between the terms. A common model for the measurements is $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$ where \mathbf{A} is a linear degradation (forward) operator and \mathbf{w} is an i.i.d. white Gaussian noise, leading to the following log-likelihood function

$$f(\mathbf{x}; \mathbf{y}) = \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2. \quad (2)$$

The above model applies to a wide variety of inverse problems such as compressed sensing, super resolution, deblurring, inpainting and much more.

In the following, we consider a general functional $f(\cdot)$ where we suppress its dependency on the observations \mathbf{y} for brevity. Moreover, both $g(\cdot)$ and $f(\cdot)$ are typically assumed to be convex where the former is generally non-differentiable, while the latter is a smooth function. By the definition of smooth functions, there exists a Lipschitz constant $L_f > 0$ such that the gradient ∇f of $f(\cdot)$ satisfies

$$\|\nabla f(\mathbf{x}_1) - \nabla f(\mathbf{x}_2)\| \leq L_f \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d_x}. \quad (3)$$

Proximal Methods The minimization problem (1) has been investigated extensively over the decades, leading to a series of diverse approaches. Many common optimization techniques for solving (1) rely on the proximal operator of $g(\cdot)$, defined as

$$\mathcal{P}_{\lambda g}(\mathbf{x}) \triangleq \arg \min_{\mathbf{z} \in \mathbb{R}^{d_x}} \frac{1}{2} \|\mathbf{x} - \mathbf{z}\|^2 + \lambda g(\mathbf{z}). \quad (4)$$

The proximal operator, which is a MAP estimator, provides an efficient way to overcome the lack of differentiability of $g(\cdot)$, especially when it admits a closed-form expression, for example in the case of the ℓ_1 -norm. A simple representative of this family of algorithms is the proximal gradient descent (PGD) method which updates the solution iteratively as follows

$$\mathbf{x}_{k+1} = \mathcal{P}_{\lambda g}(\mathbf{x}_k - \mu_k \nabla f(\mathbf{x}_k)) \quad (5)$$

where $\mu_k > 0$ is the step size. PGD has shown great performance in various applications along with strong convergence guarantees.

The PnP Framework Notice that the proximal operator can be seen, by definition, as the solution to a denoising problem. This observation serves as the basis of the PnP framework which extends standard proximal methods by replacing the proximal operator with powerful denoisers. Following this rationale, (5) is modified to create the PnP-PGD method that performs the following update rule

$$\mathbf{x}_{k+1} = \mathcal{D}(\mathbf{x}_k - \mu_k \nabla f(\mathbf{x}_k)), \quad (\text{PnP-PGD})$$

where $\mathcal{D} : \mathbb{R}^{d_x} \rightarrow \mathbb{R}^{d_x}$ is an off-the-shelf denoiser.

The RED Framework An alternative strategy for leveraging advanced denoisers is the RED framework which utilizes the denoiser to construct an explicit regularization $g(\mathbf{x}) = \frac{1}{2} \langle \mathbf{x}, \mathbf{x} - \mathcal{D}(\mathbf{x}) \rangle$. Under certain conditions on the denoiser [39, 37], $g(\cdot)$ is convex and its gradient is readily given by the denoising residual $\nabla g(\mathbf{x}) = \mathbf{x} - \mathcal{D}(\mathbf{x})$. This allows for the solution of (1) by standard algorithms, e.g., steepest (or gradient) descent

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{\mu_k}{1 + \lambda} \left(\nabla f(\mathbf{x}_k) + \lambda (\mathbf{x}_k - \mathcal{D}(\mathbf{x}_k)) \right). \quad (\text{RED-SD})$$

Surprisingly, RED-based methods have been found to be effective also when the required conditions on the denoiser are not satisfied and the expression for the regularization term is invalid. In such cases, RED uses the denoiser as an implicit regularization, similar to PnP.

Challenges in RED and PnP Both RED and PnP provide extremely efficient means for exploiting powerful denoisers to solve inverse problems. Yet their analysis poses two closely related challenges:

1. **Asymmetric Denoisers:** Many studies on RED and PnP [53, 27, 43, 44, 9, 45, 52, 42] motivate or base their derivations on the assumption that the denoiser is an approximation of either MAP or MMSE estimator. It has been proven that when the Jacobian of such estimators is well-defined it must be symmetric [20, 19, 21]. However, RED and PnP typically reach state-of-the-art results by relying on denoisers which do not exhibit this property, as in the case of the popular DnCNN technique; as a result, they lose their interpretation as optimization solvers. In such cases, while there are alternative optimization-free explanations for RED and PnP [8, 37], it is challenging to properly analyze their behavior and characterize the nature of their solutions.

2. **Convergence Issues:** Continuing the above line of reasoning, often there exists no regularization that explains the denoiser, namely no functional $g(\cdot)$ for which $\nabla g(\mathbf{x}) = \mathcal{D}(\mathbf{x})$ or $\nabla g(\mathbf{x}) = \mathbf{x} - \mathcal{D}(\mathbf{x})$ [37]. This lack of a potential function creates difficulties in studying the theoretical convergence of the RED and PnP algorithms. However, a common sufficient condition to guarantee convergence is for the denoiser $\mathcal{D}(\cdot)$ to be (averaged) nonexpansive [13], implying that

$$\|\mathcal{D}(\mathbf{x}_1) - \mathcal{D}(\mathbf{x}_2)\| \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|, \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^{d_x}, \quad (6)$$

for some Lipschitz constant $0 < L \leq 1$. However, most predefined denoisers do not satisfy this assumption. An alternative approach [40, 36, 46] is to learn a nonexpansive denoiser by adding a regularization term to the training loss which constrains its Lipschitz constant. Unfortunately, this remains an open challenge since in general there are no practical means to absolutely enforce the resultant denoiser to be nonexpansive.

Our work addresses both of these challenges. We first introduce a new class of denoisers which stem from an explicit potential functional and thus demonstrate symmetric Jacobian by their construction. We then derive an inverse technique that utilizes our denoisers as well as their potentials to ensure convergence. Three important remarks are to be made with regard to work presented below. First, we emphasize that the proposed denoisers are not intended to compete with state-of-the-art denoisers, but rather are developed to exhibit favorable properties that benefit other inversion tasks. Second, our proposed recovery method is not derived to compete with end-to-end learned solutions to specific inverse problems [31, 2, 25, 26]. Rather, our learned denoisers, along with our optimization framework, can be used for *all* inverse problems (of the form in Eq. (1)) without additional training. Lastly, while we present a specific inversion method, the potential-driven denoisers can benefit many RED and PnP algorithms, hence, our work complements these frameworks.

3 Recovery by Potential-Driven Denoisers

In this section, we present our denoisers driven by learned potential functions. Designed as gradient operators, the denoisers display symmetric Jacobians, allowing for a MAP or an MMSE estimator interpretation. More importantly, we use the denoising potentials as an explicit regularization for general inverse problems, leading to provably convergent optimization schemes which utilize our proposed denoisers. Unlike previous works, we do not require bounding or approximation of the Lipschitz constants of the denoisers.

3.1 Potential-Driven Denoisers

MAP and MMSE Denoisers as Gradients Two fundamental forms of denoisers are MAP and MMSE estimators. The theoretical properties of MAP and MMSE denoisers are well-understood and

they serve as the basis of many works which study the convergence of RED and PnP methods. The following lemmas reveals that MAP and MMSE denoisers share a common structure:

Lemma 1 (Corollary 1 [21]). *Assume the denoiser is a proximal operator $\mathcal{D} \triangleq \mathcal{P}_\phi$ of some scalar-valued function ϕ . Then, there exists a convex scalar-valued function $\psi(\cdot)$ such that $\mathcal{D} = \nabla\psi$.*

Lemma 2 (Tweedie’s Formula [16]). *Assume the denoiser is an MMSE estimator \mathcal{D} of images contaminated with white Gaussian noise of variance σ^2 . Then, $\mathcal{D} \triangleq I + \sigma^2 \nabla \log p(\cdot)$ where I is the identity operator and $p(\cdot)$ denotes a smooth probability density of images.*

The above results imply that for any MAP or MMSE denoiser D there exists a scalar-valued function ρ such that the denoiser, or equivalently its residual, can be expressed as $D = \nabla\rho$. This observation sets the main foundation of our gradient-driven denoisers, described next.

Construction of Gradient-Based Denoisers Now, we consider the task of training a deep network to perform denoising. While we focus on image denoising, the following derivations can be generalized to any signal of interest. Motivated by the above observation, we design a denoising deep network which is the gradient of scalar-valued function, acting as a potential function. Three possible approaches to construct gradient-driven denoisers are as follows:

1. Directly design a deep network which is the gradient of a certain potential function. This approach avoids building explicitly the potential network. In the Supplemental we describe certain conditions under which a given network is a gradient.
2. Compose a potential function using an arbitrary deep network N_θ whose input is an image. Simple examples of this approach are $\langle \mathbf{x}, N_\theta(\mathbf{x}) \rangle$, $\|N_\theta(\mathbf{x})\|^2$, $\|\mathbf{x} - N_\theta(\mathbf{x})\|^2$, etc. Note that in general the network N_θ does not equal to the gradient of the composite potential.
3. Model the potential function as scalar-valued deep network and use its gradient as the denoiser.

Here we focus on the last approach since it is the most general. We construct a potential function $g_\theta(\cdot) \in \mathbb{R}$ defined as a deep neural network with parameters θ . As illustrated in Fig. 1, the potential network follows a typical structure of feed-forward network with skip-connections except from two main modifications:

- The output of the network is a single scalar.
- The activations are continuously differentiable and smooth functions.

Mathematically, the potential network consists of $M > 0$ layers described recursively as follows

$$\mathbf{z}_{m+1} = s_m(\mathcal{W}_m \mathbf{z}_m + \mathcal{V}_m \mathbf{x} + b_m), \quad m = 0, \dots, M - 1. \quad (7)$$

Here, $\mathbf{x} \in \mathbb{R}^{d_x}$ is the input noisy image, $\mathbf{z}_m \in \mathbb{R}^{d_m}$ are the hidden variables where $\mathbf{z}_0 \equiv 0$ and $g_\theta(\mathbf{x}) = \mathbf{z}_M$ is the output of the network. The parameters of the network are $\theta \triangleq \{b_m, \mathcal{W}_m, \mathcal{V}_m, \}_{m=0}^{M-1}$ where b_m is a bias term and $\mathcal{W}_m, \mathcal{V}_m$ are linear operators with $\mathcal{W}_0 \equiv 0$. The activation functions are denoted by $s_m : \mathbb{R} \rightarrow \mathbb{R}$ and applied element-wise.

To allow the network to accept inputs of variable sizes, we choose the linear mappings to be standard convolutions. We ensure the output of the network is a scalar by setting the number of output channels of the last convolution to be one, and using global average pooling as the final activation function. Another important design choice is that all activations are continuously differentiable and smooth functions. This creates a potential network that is smooth as the composition of smooth functions, which is crucial for guaranteeing convergence of our inversion method, presented in Section 3.2.

Given the potential function $g_\theta(\cdot)$, we define our denoiser $\mathcal{D}_\theta(\cdot)$ as

$$\mathcal{D}_\theta(\mathbf{x}) \triangleq \mathbf{x} - \nabla g_\theta(\mathbf{x}). \quad (8)$$

Alternatively, we can write or $\mathcal{R}_\theta = \nabla g_\theta$ where \mathcal{R}_θ is the denoising residual. This construction ensures that the resultant denoiser, termed gradient-driven denoising CNN (GraDnCNN), is indeed a gradient of an explicit function, and thus it possesses a symmetric Jacobian. This form of gradient-driven denoisers represents more faithfully MAP and MMSE estimators, hopefully leading to better

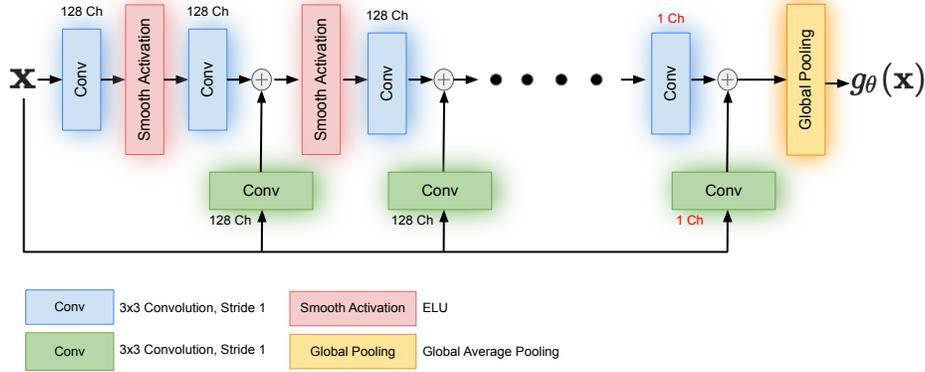


Figure 1: Network Architecture.

approximations. Finally, we perform supervised residual learning and train the denoiser (or equivalently the potential) by minimizing a chosen loss function $\mathcal{L}(\cdot, \cdot)$ with respect to the parameters

$$\theta^* = \arg \min_{\theta} \sum_n \mathcal{L}(\mathbf{x}_n - \mathbf{x}_n^*, \mathcal{R}_{\theta}(\mathbf{x}_n)), \quad (9)$$

where $\{\mathbf{x}_n^*, \mathbf{x}_n\}$ are pairs of clean and noisy images respectively.

Special Potentials: Convex Functions and Difference of Convex Functions An additional interesting path we follow is setting all $\{\mathcal{W}_m\}$ (blue blocks in Fig. 1) to be non-negative and all activation functions $\{s_m\}$ (red blocks in Fig. 1) to be convex and non-decreasing mappings. This creates a potential function $g_{\theta}(\cdot)$ which is an input convex neural network (ICNN) [4], leading to a denoiser called denoising ICNN (DnICNN). As we explain later in Section 3.2, the use of convex potentials leads to convex inverse problems, thus it facilitates the optimization process and allows to derive recovery methods with global convergence guarantees.

The down side of convex potentials is that the non-negative weights $\{\mathcal{W}_m\}$ may limit the expressivity of the network. Hence, we propose an alternative where the potential network is defined as the difference of two ICNNs

$$g_{\theta}(\mathbf{x}) \triangleq g_{\theta_1}(\mathbf{x}) - g_{\theta_2}(\mathbf{x}). \quad (10)$$

The resultant potential enjoys favorable properties as the difference of two convex functions [28] and it can approximate non-convex functions (unlike the convex potentials). Thus, we present a third denoiser, referred to as denoising by difference of ICNNs (DnDICNN), which acts as a compromise between GraDnCNN and DnICNN. In the following section, we study the properties of these three denoisers in the context of our recovery scheme, both in terms of stability and recovery performance.

3.2 Convergent Inversion Method

Inversion by Descent Given a trained gradient-driven denoiser, we wish to exploit the denoising power within general recovery processes. The existence of an underlying learned potential $g_{\theta}(\cdot)$ allows us to formulate the recovery as the following problem:

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{d_x}} f(\mathbf{x}) + \lambda g_{\theta}(\mathbf{x}). \quad (11)$$

Here we assume $f(\cdot)$ is a continuously differentiable smooth function, typically the log-likelihood of the observations given by (2). The potential network acts as a regularization term or a smooth image prior which creates a well-posed problem. Furthermore, by our construction the potential $g_{\theta}(\cdot)$ is smooth, thus, the complete objective, denoted by $F(\cdot)$, is smooth with some unknown Lipschitz constant $L > 0$. This facilitates the optimization process considerably and we can solve (11) via a simple gradient descent algorithm

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \nabla F(\mathbf{x}_k), \quad (12)$$

where we start with some initial solution \mathbf{x}_0 . Notice the above iteration can be written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \left(\nabla f(\mathbf{x}_k) + \lambda (\mathbf{x}_k - \mathcal{D}_{\theta}(\mathbf{x}_k)) \right) \quad (13)$$

where $\mathcal{D}_\theta(\cdot)$ is our denoiser originating from $g_\theta(\cdot)$.

Backtracking Step Size Note that the step size μ_k must be set properly to reach convergence. Typically, one only has access to the denoiser itself, hence the step size is tuned empirically which may lead either to divergence or to slow convergence due to an overly small step size. By contrast, we have complete access to the objective function, thus, we can properly set the step size to ensure convergence. To that end we rely on the Descent Lemma [6] in the Supplemental, which implies that

$$F(\mathbf{x}_{k+1}) \leq F(\mathbf{x}_k) - \frac{\mu_k}{2} \|\nabla F(\mathbf{x}_k)\|_2^2, \forall \mu_k : 0 \leq \mu_k \leq \frac{1}{L}. \quad (14)$$

When $\nabla F(\mathbf{x}_k) \neq 0$ we obtain that $F(\mathbf{x}_{k+1}) < F(\mathbf{x}_k)$; otherwise, we reach a critical point. The challenge that remains is setting the step size according the Lipschitz constant L which is typically unknown. To address this, we set a backtracking step size defined as $\mu_k \triangleq 1/\hat{L}$ where \hat{L} is an approximation of the true constant L . We start with some initial guess $\hat{L} = L_0$ and at each iteration update \hat{L} and μ_k accordingly until inequality (14) is satisfied, ensuring we perform a descent step.

Algorithm 1: Regularization by Potential-Driven Denoising

Input: initial point $\mathbf{x}_0 \in \mathbb{R}^{d_x}$, $L_0 > 0$, $\beta > 1$, $\lambda > 0$, $f(\cdot)$, $g_\theta(\cdot)$ and the related denoiser $\mathcal{D}_\theta(\cdot)$.

- 1 Set $F(\cdot) \triangleq f(\cdot) + \lambda g_\theta(\cdot)$ and $\hat{L} = L_0$.
 - 2 **for** $k=0,1,2,\dots$ **do**
 - 3 $\mu_k = 1/\hat{L}$.
 - 4 $\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \left(\nabla f(\mathbf{x}_k) + \lambda(\mathbf{x}_k - \mathcal{D}_\theta(\mathbf{x}_k)) \right)$.
 - 5 **while** $F(\mathbf{x}_{k+1}) > F(\mathbf{x}_k) - \frac{\mu_k}{2} \|\nabla F(\mathbf{x}_k)\|_2^2$ **do**
 - 6 $\hat{L} = \beta \cdot \hat{L}$, $\mu_k = 1/\hat{L}$.
 - 7 $\mathbf{x}_{k+1} = \mathbf{x}_k - \mu_k \left(\nabla f(\mathbf{x}_k) + \lambda(\mathbf{x}_k - \mathcal{D}_\theta(\mathbf{x}_k)) \right)$.
-

The resulting inversion technique, outlined in Algorithm 1, can be viewed as RED-SD with backtracking step size which is enabled by the existence of an available prior function $g_\theta(\cdot)$. In addition, note that throughout the iterations it holds that $\hat{L} \leq \max(L_0, \beta L)$ where L is the true unknown Lipschitz constant. Also note that the total number of updates of \hat{L} is bounded by $\log_\beta(L/L_0) \approx \mathcal{O}(\log_2 L)$.

Convergence Guarantees We remark that other more sophisticated optimization techniques can be considered for solving (11), including acceleration schemes. However, rather than providing the most efficient recovery method, we focus here on simplicity to highlight the benefits of learned potential-driven denoisers. Moreover, there is a vast literature which provides convergence guarantees for Algorithm 1 under various conditions. In the following theorem we present a standard convergence analysis based on the smoothness of the objective function. The proof is given in the Supplemental.

Theorem 1. *Assume that $f(\cdot)$ and $g_\theta(\cdot)$ are continuously differentiable and smooth functions. Define $F(\cdot) \triangleq f(\cdot) + \lambda g_\theta(\cdot)$ and assume it is bounded below by F^* . Let $\{\mathbf{x}_k\}_{k \geq 0}$ be the sequence generated by Algorithm 1. Then,*

$$\min_{m=0,1,\dots,k-1} \|\nabla F(\mathbf{x}_m)\|_2^2 \leq \frac{2 \max(L_0, \beta L)}{k} (F(\mathbf{x}_0) - F^*).$$

The above theorem implies that $\|\nabla F(\mathbf{x}_k)\|_2^2 \rightarrow 0$ as $k \rightarrow \infty$ and that all limit points of the sequence $\{\mathbf{x}_k\}_{k \geq 0}$ are stationary points of the objective function $F(\cdot)$. Notice this result applies to all the three denoisers (or potentials) presented in Section 3.1. This emphasizes the importance of the smoothness of the potential functions which ensures the convergence of the recovery, regardless to the highly non-convex nature of the objective function.

Commonly, the function $f(\cdot)$ is a convex function, e.g. when it is the log-likelihood. In such scenarios, the use of convex potential networks leads to a special case where the recovery problem is convex and we obtain global convergence guarantees:

Corollary 1. *Assume that $f(\cdot)$ and $g_\theta(\cdot)$ are convex in addition to the conditions of Theorem 1. Then, Algorithm 1 converges to global minima of the objective function $F(\cdot)$.*

The proof is straightforward since according to Theorem 1 the sequence generated by Algorithm 1 converges to stationary points of the objective function, and any stationary point of a convex function

is a global minimizer. Thus, in the context of inverse problems, our convex potential networks connect the well-known world of convex optimization and with the highly non-linear non-convex nature of deep neural networks.

Finally, for the completeness of our work, we study the case where the recovery problem includes additional constraints on the solution, leading to the following formulation

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{d_x}} f(\mathbf{x}) + \lambda g_\theta(\mathbf{x}) + h(\mathbf{x}), \quad (15)$$

where the function $h(\cdot)$ is convex yet possibly non-smooth. We provide the full analysis consisting of algorithmic and theoretical results in the Supplemental.

4 Numerical Experiments

Here we study the performance of Algorithm 1 using our three proposed potential-driven denoisers: GraDnCNN, DnICNN, DnDICNN. We compare ourselves to PnP-PGD and RED-SD, applied with the popular DnCNN denoiser [54], for the tasks of Gaussian deblurring and image super resolution.

Datasets For training the denoising networks for blind Gaussian denoising we use the public DIV2K dataset [2] that consists of a total of 900 high resolution images, 800 for training and 100 for validation. We perform data augmentation by randomly cropping 100 patches of size 50×50 from each image of the DIV2K training set. We then contaminate each patch with an additive i.i.d. white Gaussian noise (WGN) with a noise level σ_n chosen uniformly at random from the range $[1, 55]$, creating our 80,000 sample training set. For our validation set, we replicate each image of the DIV2K validation set 10 times, and we add WGN to each copy with random noise level in the specified range. As our test images for evaluating the inverse methods for deblurring and super resolution, we use two different datasets. The first set, provided by the authors of [39], consists of 11 common color images (butterfly, starfish, etc), and we refer to it as Set11. For a more thorough assessment, we utilize all 500 natural images from the public Berkeley segmentation dataset (BSD500) [32]. All pixel values of the images described above are in the range $[0, 255]$.

Training Denoisers We experiment with four different denoisers. The first is a variant of the DnCNN-B architecture [54] with 7 layers, 128 feature maps, and 3×3 convolution kernels. For GraDnCNN we use the architecture shown in Fig. 1, consisting of $M = 6$ layers, ELU activation functions, 128 feature maps, and 3×3 bias-free convolutions where the final two convolution layers have a single output channel, and the last activation is global average pooling to produce a scalar. A similar architecture is used for DnICNN, where we set certain weights to be non-negative. The latter architecture is then used to construct our DnDICNN as the difference of two DnICNNs.

Given the datasets detailed above, we train each of the networks using an Adam optimizer for 100 epochs with a constant learning rate of 10^{-3} . For the training loss, we use a modified version of mean squared error (MSE) cost function: $\sum_n \frac{1}{\sigma_n^2} \text{MSE}(\mathbf{x}_n - \mathbf{x}_n^*, \mathcal{R}_\theta(\mathbf{x}_n))$ where \mathbf{x}_n^* denotes a clean patch, \mathbf{x}_n is the corresponding noisy patch contaminated with noise of level σ_n , and $\mathcal{R}_\theta(\mathbf{x}_n)$ is the output of the network. Since samples with larger noise levels typically lead to larger training errors, we find that this modification of dividing the MSE by the noise level is required to prevent the denoisers from over-smoothing. All experiments are performed in Tensorflow [1] where each model is trained on a single NVIDIA Tesla 32GB V100 GPU.

Inverse Problems We focus here on two common inverse problems: Gaussian deblurring and single image super resolution. For the task of deblurring we perform the following process on the test datasets (Set11, BSD500) described above. Each image is convolved with a 2D Gaussian function 25×25 with a standard deviation of 1.6. Then, we add to the result an i.i.d. WGN of level $\hat{\sigma} = \sqrt{2}$. For super resolution we follow the same procedure except we add an intermediate step, before noise contamination, in which we scale-down the image by a factor of 3 at each axis.

Recovery Given the degraded test images and the trained denoisers, we compare five different inverse methods: PnP-PGD and RED-SD, both with DnCNN, as well as Algorithm 1 employed with our three potential-driven denoisers. For Algorithm 1 we use the constrained update rule, described in the Supplemental, where the proximal operator is the projection of pixel values of the solution into the range $[0, 255]$. We apply the same projection on each intermediate solution of PnP-PGD

and RED-SD for improved stability. For properly setting the hyperparameters of the different tested methods we use Set11 to perform an extensive hyperparameter search, provided in the Supplemental.

We first study the convergence by running all the techniques for a fixed number of 400 iterations for both inverse tasks, where we measure convergence by $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|/\mu_k$. The results, averaged over each dataset separately, are given in Figures 2 and 3. The foremost observation is that in all the scenarios PnP-PGD diverges at a certain point. We conjecture the reason of this instability of PnP-PGD lies in the fact that unlike the other methods PnP-PGD includes no hyperparameters for constraining the denoiser, thus, powerful denoisers may lead to divergence by analogy with a too large learning rate. This can be prevented by considering more sophisticated version of PnP or by performing early stopping, however, it will introduce additional hyperparameter which need to be properly tuned for each data instance, making the technique less robust. Moreover, notice that the convergence of Algorithm 1 does not depend on the values of λ and μ , regardless of the potential-driven denoiser in use. This is a desired property that most RED and PnP methods do not exhibit. We note that the convex potential leads to fastest convergence which is expected, since in this case the objective function is smooth and convex, facilitating the optimization process significantly.

Next, we quantitatively compare the inverse algorithms by computing the average peak signal to noise ratio (PSNR) and structural similarity index (SSIM) of their solutions with respect to the ground truth images. As seen in Table1, Algorithm 1 with GraDnCNN produces the best performance in both tasks and for both test sets. Algorithm 1 with DnICNN yields the worst results among our potential-driven denoisers, yet comparable to PnP-PGD and RED-SD with DnCNN. The performance of Algorithm 1 with DnDICNN is consistent with the fact that DnDICNN presents a compromise between GraDnCNN and DnICNN in terms of the expressivity of the network. To visually support our claims, we provide selected recovered images in Fig. 4, while additional results are given in the Supplemental. An important remark is that in contrast to PnP and RED methods, presented here and many others, the solutions of Algorithm 1 and its performance are completely independent of the choice of the step-size μ . Hence, Algorithm 1 is more robust.

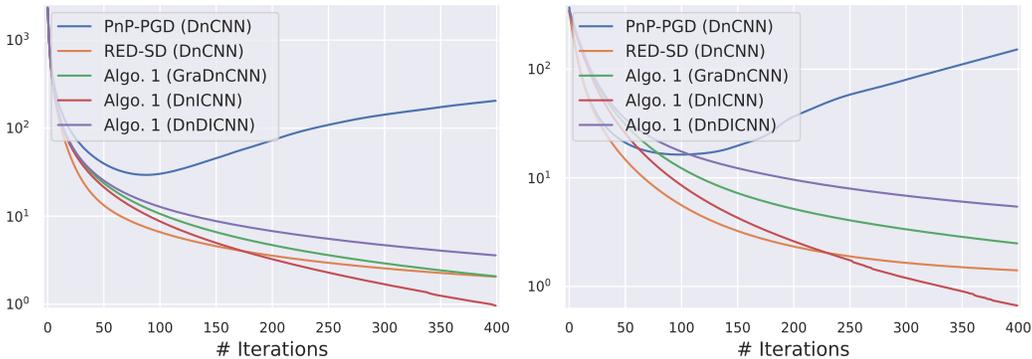


Figure 2: Average Convergence for Set11 - (Left) Gaussian deblurring (Right) Super resolution

Table 1: Average PSNR / SSIM obtained for different methods and datasets.

Datasets	PnP-PGD DnCNN	RED-SD DnCNN	Algorithm 1		
			GraDnCNN	DnICNN	DnDICNN
Gaussian Deblurring					
Set11	28.81 / 0.86	28.69 / 0.82	29.42 / 0.86	28.91 / 0.85	29.07 / 0.84
BSD500	28.79 / 0.84	28.32 / 0.79	28.91 / 0.83	28.63 / 0.82	28.41 / 0.80
Super Resolution					
Set11	26.20 / 0.77	26.14 / 0.76	26.69 / 0.79	25.95 / 0.76	26.20 / 0.76
BSD500	26.36 / 0.75	26.24 / 0.73	26.57 / 0.74	26.15 / 0.73	25.95 / 0.71

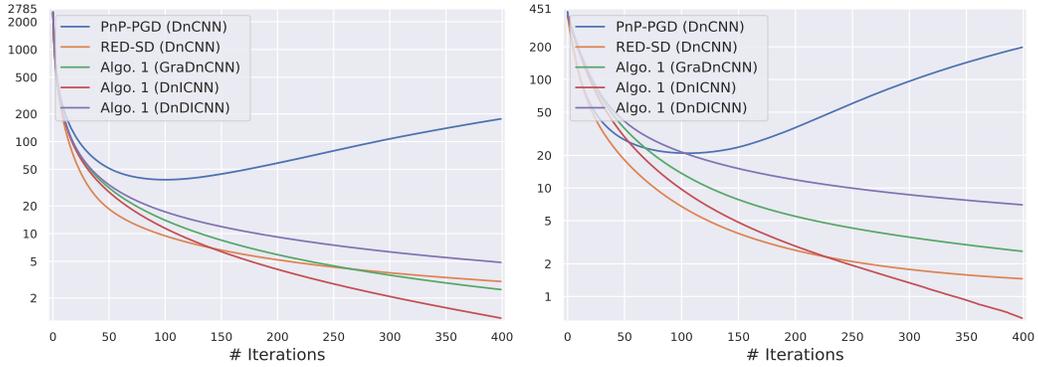


Figure 3: Average Convergence for BSD500 - (Left) Gaussian deblurring (Right) Super resolution.

5 Conclusion

In this paper we developed a provably convergent method with explicit denoising regularization for solving inverse problems. To that end, we first introduced the notion of potential-driven denoisers, i.e. denoisers which are based on the gradient of learned scalar-valued networks. We then utilized our denoisers along with their potential functions to derive an inverse gradient descent method with backtracking step size, ensuring convergence. We demonstrate experimentally that the resulting methods provide improved performance compared to standard RED and PnP methods, while offering provable stability.

The developed framework complements RED and PnP as our potential-driven denoisers can be used within any RED or PnP techniques to guarantee convergence, regardless of the data being used. Furthermore, while our work focus on learning denoisers, the concept of potential-driven networks may be generalized to benefit other image-to-image operations.

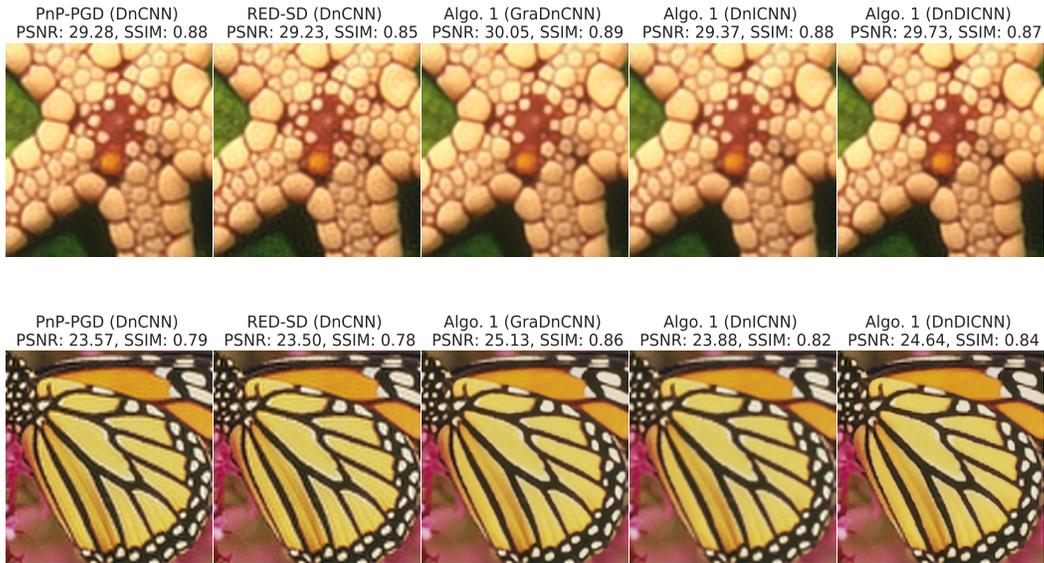


Figure 4: Recovery results of the tasks of (top) Gaussian deblurring and (bottom) super resolution.

References

- [1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: Tensorflow: A system for large-scale machine learning. In: Symposium on Operating Systems Design and Implementation. pp. 265–283 (2016)
- [2] Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 126–135 (2017)
- [3] Ahmad, R., Bouman, C.A., Buzzard, G.T., Chan, S., Liu, S., Reehorst, E.T., Schniter, P.: Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery. *IEEE Signal Processing Magazine* **37**(1), 105–116 (2020)
- [4] Amos, B., Xu, L., Kolter, J.Z.: Input convex neural networks. In: International Conference on Machine Learning. pp. 146–155. PMLR (2017)
- [5] Bayati, M., Montanari, A.: The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory* **57**(2), 764–785 (2011)
- [6] Beck, A., Teboulle, M.: Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications* pp. 42–88 (2009)
- [7] Berthier, R., Montanari, A., Nguyen, P.M.: State evolution for approximate message passing with non-separable functions. *Information and Inference: A Journal of the IMA* **9**(1), 33–79 (2020)
- [8] Buzzard, G.T., Chan, S.H., Sreehari, S., Bouman, C.A.: Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. *SIAM Journal on Imaging Sciences* **11**(3), 2001–2020 (2018)
- [9] Chan, S.H.: Performance analysis of plug-and-play ADMM: A graph signal processing perspective. *IEEE Transactions on Computational Imaging* **5**(2), 274–286 (2019)
- [10] Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging* **3**(1), 84–98 (2016)
- [11] Chatterjee, P., Milanfar, P.: Is denoising dead? *IEEE Transactions on Image Processing* **19**(4), 895–911 (2009)
- [12] Cohen, R., Elad, M., Milanfar, P.: Regularization by denoising via fixed-point projection (RED-PRO). arXiv preprint arXiv:2008.00226 (2020)
- [13] Combettes, P.L., Pesquet, J.C.: Fixed point strategies in data science. *IEEE Transactions on Signal Processing* (2021)
- [14] Danielyan, A., Katkovnik, V., Egiazarian, K.: Image deblurring by augmented Lagrangian with BM3D frame prior. In: Workshop on Information Theoretic Methods in Science and Engineering (WITMSE), Tampere, Finland. pp. 16–18 (2010)
- [15] Donoho, D.L., Maleki, A., Montanari, A.: Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences* **106**(45), 18914–18919 (2009)
- [16] Efron, B.: Tweedie’s formula and selection bias. *Journal of the American Statistical Association* **106**(496), 1602–1614 (2011)
- [17] Egiazarian, K., Katkovnik, V.: Single image super-resolution via BM3D sparse coding. In: IEEE European Signal Processing Conference (EUSIPCO). pp. 2849–2853 (2015)
- [18] Fletcher, A.K., Pandit, P., Rangan, S., Sarkar, S., Schniter, P.: Plug-in estimation in high-dimensional linear inverse problems: A rigorous analysis. In: Advances in Neural Information Processing Systems. pp. 7440–7449 (2018)
- [19] Gribonval, R.: Should penalized least squares regression be interpreted as maximum a posteriori estimation? *IEEE Transactions on Signal Processing* **59**(5), 2405–2410 (2011)
- [20] Gribonval, R., Nikolova, M.: On Bayesian estimation and proximity operators. *Applied and Computational Harmonic Analysis* (2019)
- [21] Gribonval, R., Nikolova, M.: A characterization of proximity operators. *Journal of Mathematical Imaging and Vision* **62**(6), 773–789 (2020)

- [22] Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters* **24**(12), 1872–1876 (2017)
- [23] Kawar, B., Vaksman, G., Elad, M.: SNIPS: Solving noisy inverse problems stochastically. arXiv preprint arXiv:2105.14951 (2021)
- [24] Kawar, B., Vaksman, G., Elad, M.: Stochastic image denoising by sampling from the posterior distribution. arXiv preprint arXiv:2101.09552 (2021)
- [25] Koh, J., Lee, J., Yoon, S.: Single-image deblurring with neural networks: A comparative survey. *Computer Vision and Image Understanding* **203**, 103134 (2021)
- [26] Lai, W.S., Huang, J.B., Hu, Z., Ahuja, N., Yang, M.H.: A comparative study for single image blind deblurring. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1701–1709 (2016)
- [27] Laumont, R., De Bortoli, V., Almansa, A., Delon, J., Durmus, A., Pereyra, M.: Bayesian imaging using plug & play priors: when Langevin meets Tweedie. arXiv preprint arXiv:2103.04715 (2021)
- [28] Le Thi, H.A., Dinh, T.P.: DC programming and DCA: thirty years of developments. *Mathematical Programming* **169**(1), 5–68 (2018)
- [29] Levin, A., Nadler, B.: Natural image denoising: Optimality and inherent bounds. In: *IEEE CVPR*. pp. 2833–2840 (2011)
- [30] Levin, A., Nadler, B., Durand, F., Freeman, W.T.: Patch complexity, finite pixel correlations and optimal denoising. In: *European Conference on Computer Vision*. pp. 73–86. Springer (2012)
- [31] Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pp. 136–144 (2017)
- [32] Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proc. 8th Int’l Conf. Computer Vision*. vol. 2, pp. 416–423 (July 2001)
- [33] Metzler, C.A., Maleki, A., Baraniuk, R.G.: From denoising to compressed sensing. *IEEE Transactions on Information Theory* **62**(9), 5117–5144 (2016)
- [34] Milanfar, P.: Symmetrizing smoothing filters. *SIAM Journal on Imaging Sciences* **6**(1), 263–284 (2013)
- [35] Nair, P., Gavaskar, R.G., Chaudhury, K.N.: Fixed-point and objective convergence of plug-and-play algorithms. *IEEE Transactions on Computational Imaging* **7**, 337–348 (2021)
- [36] Pesquet, J.C., Repetti, A., Terris, M., Wiaux, Y.: Learning maximally monotone operators for image recovery. arXiv preprint arXiv:2012.13247 (2020)
- [37] Reehorst, E.T., Schniter, P.: Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on computational imaging* **5**(1), 52–67 (2018)
- [38] Rick Chang, J., Li, C.L., Póczos, B., Vijaya Kumar, B., Sankaranarayanan, A.C.: One network to solve them all: Solving linear inverse problems using deep projection models. In: *Proceedings of the IEEE International Conference on Computer Vision*. pp. 5888–5897 (2017)
- [39] Romano, Y., Elad, M., Milanfar, P.: The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences* **10**(4), 1804–1844 (2017)
- [40] Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-play methods provably converge with properly trained denoisers. In: *International Conference on Machine Learning*. pp. 5546–5557. PMLR (2019)
- [41] Sreehari, S., Venkatakrishnan, S.V., Wohlberg, B., Buzzard, G.T., Drummy, L.F., Simmons, J.P., Bouman, C.A.: Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging* **2**(4), 408–423 (2016)
- [42] Sun, Y., Liu, J., Kamilov, U.S.: Block coordinate regularization by denoising. In: *Advances in Neural Information Processing Systems*. pp. 380–390 (2019)
- [43] Sun, Y., Wohlberg, B., Kamilov, U.S.: An online plug-and-play algorithm for regularized image reconstruction. *IEEE Transactions on Computational Imaging* **5**(3), 395–408 (2019)

- [44] Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.: Scene-adapted plug-and-play algorithm with convergence guarantees. In: IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2017)
- [45] Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.: A convergent image fusion algorithm using scene-adapted Gaussian-mixture-based denoising. *IEEE Transactions on Image Processing* **28**(1), 451–463 (2018)
- [46] Terris, M., Repetti, A., Pesquet, J.C., Wiaux, Y.: Building firmly nonexpansive convolutional neural networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8658–8662 (2020)
- [47] Ttirer, T., Giryes, R.: Image restoration by iterative denoising and backward projections. *IEEE Transactions on Image Processing* **28**(3), 1220–1234 (2018)
- [48] Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: IEEE Global Conference on Signal and Information Processing. pp. 945–948 (2013)
- [49] Wei, K., Aviles-Rivero, A., Liang, J., Fu, Y., Schönlieb, C.B., Huang, H.: Tuning-free plug-and-play proximal algorithm for inverse imaging problems. In: International Conference on Machine Learning. pp. 10158–10169. PMLR (2020)
- [50] Wu, Z., Sun, Y., Liu, J., Kamilov, U.S.: Online regularization by denoising with applications to phase retrieval. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
- [51] Xing, J., Kamilov, U., Wu, W., Wang, Y., Zhang, M.: Plug-and-play priors for reconstruction-based placental image registration. In: Smart Ultrasound Imaging and Perinatal, Preterm and Paediatric Image Analysis, pp. 133–142. Springer (2019)
- [52] Xu, X., Liu, J., Sun, Y., Wohlberg, B., Kamilov, U.S.: Boosting the performance of Plug-and-Play Priors via denoiser scaling. arXiv preprint arXiv:2002.11546 (2020)
- [53] Xu, X., Sun, Y., Liu, J., Wohlberg, B., Kamilov, U.S.: Provable convergence of plug-and-play priors with MMSE denoisers. *IEEE Signal Processing Letters* **27**, 1280–1284 (2020)
- [54] Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing* **26**(7), 3142–3155 (2017)
- [55] Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In: Proceedings of the IEEE CVPR. pp. 3929–3938 (2017)
- [56] Zhang, K., Zuo, W., Zhang, L.: Deep plug-and-play super-resolution for arbitrary blur kernels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1671–1681 (2019)