

Provably Convergent Plug-and-Play Quasi-Newton Methods

Tan, Hong Ye; Mukherjee, Subhadip; Tang, Junqi; Schönlieb, Carola-Bibiane

License:
Creative Commons: Attribution (CC BY)

Document Version
Peer reviewed version

Citation for published version (Harvard):
Tan, HY, Mukherjee, S, Tang, J & Schönlieb, C-B 2023, 'Provably Convergent Plug-and-Play Quasi-Newton Methods', *SIAM Journal on Imaging Sciences*.

[Link to publication on Research at Birmingham portal](#)

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

Provably Convergent Plug-and-Play Quasi-Newton Methods

Hong Ye Tan*, Subhadip Mukherjee*[†], Junqi Tang*[‡], and Carola-Bibiane Schönlieb*

Abstract. Plug-and-Play (PnP) methods are a class of efficient iterative methods that aim to combine data fidelity terms and deep denoisers using classical optimization algorithms, such as ISTA or ADMM, with applications in inverse problems and imaging. Provable PnP methods are a subclass of PnP methods with convergence guarantees, such as fixed point convergence or convergence to critical points of some energy function. Many existing provable PnP methods impose heavy restrictions on the denoiser or fidelity function, such as *nonexpansiveness* or *strict convexity*, respectively. In this work, we propose a novel algorithmic approach incorporating quasi-Newton steps into a provable PnP framework based on proximal denoisers, resulting in greatly accelerated convergence while retaining light assumptions on the denoiser. By characterizing the denoiser as the proximal operator of a weakly convex function, we show that the fixed points of the proposed quasi-Newton PnP algorithm are critical points of a weakly convex function. Numerical experiments on image deblurring and super-resolution demonstrate 2–8x faster convergence as compared to other provable PnP methods with similar reconstruction quality.

Key words. Plug-and-Play, inverse problems, quasi-Newton methods, image reconstruction

MSC codes. 49M15, 49J52, 65K15

1. Introduction. Many image restoration problems can be formulated as reconstructing data $x \in \mathbb{R}^n$ from a noisy measurement $y = Ax + \varepsilon \in \mathbb{R}^m$, where A is a linear forward operator, and ε is some measurement noise. One common way to solve this is the variational formulation

$$(1.1) \quad \arg \min_{x \in \mathbb{R}^n} \varphi(x) = f(x) + g(x),$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is typically a continuously differentiable data fidelity term, and $g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is a regularization term that controls the prior. In many cases, the fidelity term incorporates a forward operator $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$, which may correspond to physical operators such as blurring operators or Radon transforms [28]. For a noisy measurement $y = Ax + \varepsilon$ with additive white noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, the fidelity term takes the form of the negative log likelihood $f(x) = \|Ax - y\|^2 / (2\sigma^2)$. For many physical forward operators, such as blurring or down-sampling, the optimization problem $\min_x f(x)$ is ill-posed, thus a regularization term is needed [36]. Classical examples for regularization include using Fourier spectra (spectral regularization) or total variation (TV) regularization on natural images [62, 63], whereas recent works aim to learn a neural network regularizer [44, 49].

Fully data-driven approaches have been shown to outperform explicitly defined regularizers [77, 76, 49]. However, the outputs of these learned schemes often do not correspond to

*Department of Applied Mathematics and Theoretical Physics, University of Cambridge, UK (hyt35@cam.ac.uk, sm2467@cam.ac.uk, cbs31@cam.ac.uk).

[†]Department of Computer Science, University of Bath, UK. Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India (smukherjee@ece.iitkgp.ac.in).

[‡]School of Mathematics, University of Birmingham, UK (j.tang.2@bham.ac.uk).

36 closed-form minimization problems of the form (1.1). This is particularly limiting in sensitive
 37 applications such as medical imaging, where interpretability is necessary [73, 72]. Recent
 38 lines of work consider combining iterative algorithms with generic denoisers, with notable
 39 examples including regularization by denoising (RED) [17, 58], consensus equilibrium [12],
 40 and deep mean-shift priors [2]. In this work, we will focus on the line of Plug-and-Play (PnP)
 41 methods, which arise from replacing proximal steps with denoisers. Under certain conditions
 42 on the fidelity and denoisers as detailed in Section 1.2, fixed point convergence of certain PnP
 43 methods can be established, characterized by critical points of a corresponding functional.

44 The PnP framework of replacing the regularization proximal step with a denoiser is flexible
 45 in the choice of denoiser. In particular, it allows for the use of both classical denoisers such as
 46 NLM or BM3D [11, 18], as well as data-driven denoisers [78, 77, 64]. This allows for extending
 47 the use of Gaussian denoisers to other image reconstruction tasks, such as super-resolution or
 48 image deblurring. Recently, PnP methods based on the half-quadratic splitting were able to
 49 achieve state-of-the-art performance for image reconstruction using a variable-strength Gauss-
 50 ian denoiser called DRUNet [78]. Named the deep Plug-and-Play image restoration (DPIR)
 51 method, DPIR outperforms or is competitive with fully learned methods for applications such
 52 as image deblurring, super-resolution, and demosaicing while using only a single denoiser prior
 53 [77]. This work demonstrates the flexibility of PnP, using one prior for multiple reconstruction
 54 tasks.

55 While PnP methods can be used to achieve excellent performance, empirical convergence
 56 does not equate to traditional notions of convergence. Indeed, while DPIR is able to achieve
 57 state-of-the-art results in as few as eight PnP iterations, there are no associated theoretical
 58 results. Moreover, DPIR can diverge when more PnP iterations are applied [32]. This can
 59 be empirically alleviated using various stopping criteria, but this raises an additional issue
 60 for defining a notion of “best reconstruction”. In this work, we sidestep this by considering
 61 provable PnP methods. We use the term “provable PnP” to refer to PnP methods equipped
 62 with some notion of convergence, such as fixed-point convergence, or the stronger notion of
 63 convergence to critical points of a function.

64 Various approaches for accelerating PnP methods have been proposed, including using
 65 classical accelerated optimization algorithms, block-coordinate methods, parallelization, and
 66 dimensionality reduction [38, 23, 71, 37, 68]. In the context of convergence to fixed points
 67 of a functional, theoretical results for PnP based on accelerated classical methods such as
 68 FISTA have not arisen in the literature. This work proposes to extend the work on provable
 69 PnP methods by introducing a quasi-Newton step to accelerate convergence, while retaining
 70 a corresponding closed-form minimization problem with relatively weak constraints.

71 **1.1. Definitions and Notations.** We begin with some definitions and notation. Let $\overline{\mathbb{R}} =$
 72 $\mathbb{R} \cup \{+\infty\}$ be the extended real line. Recall that a function $g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is *proper* if the effective
 73 domain $\text{dom } g = \{x \in \mathbb{R}^n \mid g(x) < +\infty\}$ is nonempty, and *closed* (or *lower-semicontinuous*) if
 74 for every sequence $x^k \rightarrow x$ in \mathbb{R}^n , we have $g(x) \leq \liminf_k g(x^k)$.

75 **Definition 1.1.** For a scalar $\gamma > 0$ and a proper closed convex function $g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$, the
 76 proximal map is

$$77 \quad (1.2) \quad \text{prox}_{\gamma g}(x) = \arg \min_{u \in \mathbb{R}^n} \left\{ g(u) + \frac{1}{2\gamma} \|u - x\|^2 \right\}.$$

78 The Moreau envelope is the value function of the proximal map, defined as

79 (1.3)
$$g^\gamma(x) = \min_{u \in \mathbb{R}^n} \left\{ g(u) + \frac{1}{2\gamma} \|u - x\|^2 \right\}.$$

80 Properties of the Moreau envelope and proximal operators are well documented in classical
 81 literature [59, 7, 48, 27]. In particular, for proper closed convex g , the proximal operator is
 82 single-valued and nonexpansive, and the envelope function g^γ is convex and \mathcal{C}^1 with derivative

83
$$\nabla g^\gamma(x) = \gamma^{-1}(x - \text{prox}_{\gamma g}(x)).$$

84 **1.2. Plug-and-Play Methods.** The Plug-and-Play (PnP) framework was first introduced
 85 by Venkatakrisnan et al. in 2013 for model-based image reconstruction [74]. PnP meth-
 86 ods arise from composite convex optimization algorithms, wherein a prior regularization step
 87 is associated with a denoising step. The first composite optimization algorithm considered
 88 was Alternating Directions Method of Multipliers (ADMM), a classical proximal splitting
 89 algorithm used for minimizing composite functions. In the case of image reconstruction, a
 90 maximum likelihood estimation model can be decomposed into a composite problem. For a
 91 noisy measurement y and unknown data x , let $p(y|x)$ be the conditional likelihood, and $p(x)$
 92 the prior of the unknown x . The maximum a-posteriori (MAP) estimate \hat{x} is given as follows:

93
$$\begin{aligned} \hat{x} &= \arg \max_x \{p(y|x) + p(x)\} \\ &= \arg \min_x \{f(x; y) + g(x)\}, \end{aligned}$$

94
95

96 where f is the likelihood/fidelity term, and g is the prior/regularization term. A classical
 97 example would be TV regularization for additive Gaussian noise, where the fidelity term is
 98 $f(x; y) = \|Ax - y\|_2^2 / 2\sigma^2$, and the prior term is $g(x) = \lambda \|\nabla x\|_1$ [63]. To solve the minimization
 99 problem for general convex f, g , proximal splitting algorithms such as ADMM consider alter-
 100 nating applications of the individual proximal operators $\text{prox}_f, \text{prox}_g$ or subgradients $\partial f, \partial g$.
 101 The key observation of PnP is that the prior regularization step can also be interpreted as a
 102 denoising operation [64].

103 More generally, the PnP framework can be applied to monotone operator splitting meth-
 104 ods. Under light conditions, the composite convex optimization problem of minimizing $f + g$
 105 can be reformulated as the monotone inclusion problem $0 \in \partial f(x) + \partial g(x)$ [59, 7]. For convex
 106 f and g , the operators ∂f and ∂g are monotone operators. Monotone operator splitting meth-
 107 ods aim to solve the inclusion $0 \in \partial f(x) + \partial g(x)$, using only the resolvents of the individual
 108 operators $\partial f, \partial g$, and/or the individual operators $\text{prox}_f, \text{prox}_g$ themselves [7]. In convex analysis
 109 terms, this corresponds to splitting the proximal operator prox_{f+g} in terms of the simpler
 110 proximals prox_f and prox_g or gradients ∇f and ∇g . Two common splitting algorithms are
 111 the forward-backward splitting (FBS) and the Douglas-Rachford splitting (DRS), given as
 112 follows [7, 21]:

113 (FBS)
$$x^{k+1} = \text{prox}_g(I - \nabla f)(x^k);$$

114

115 (DRS)
$$\begin{cases} x^{k+1} = \text{prox}_f(y^k), \\ y^{k+1} = y^k + \text{prox}_g(2x^{k+1} - y^k) - x^{k+1}. \end{cases}$$

116 One classical application of a splitting algorithm is the iterative thresholding and shrinkage
 117 algorithm (ISTA) for LASSO problems, where the fidelity f is quadratic, and the prior term is
 118 the ℓ_1 norm $g(x) = \|x\|_1$ [19, 9]. Applying the PnP framework to FBS and DRS, by replacing
 119 the prior proximal terms prox_g with a denoiser D_σ , gives the PnP-FBS and PnP-DRS methods.

$$120 \text{ (PnP-FBS)} \quad x^{k+1} = D_\sigma(I - \nabla f)(x^k);$$

121

$$122 \text{ (PnP-DRS)} \quad \begin{cases} x^{k+1} = \text{prox}_f(y^k), \\ y^{k+1} = y^k + D_\sigma(2x^{k+1} - y^k) - x^{k+1}. \end{cases}$$

123 Provable PnP results first arose by Chan et al. for the PnP-ADMM scheme, demonstrating
 124 fixed-point convergence under a bounded denoiser assumption $\|D_\sigma(x) - x\| \leq C\sigma^2$ [15]. Ryu
 125 et al. demonstrate convergence of the PnP-FBS algorithm when f is strongly convex and the
 126 denoiser residual $D_\sigma - I$ is Lipschitz with [sufficiently small Lipschitz constant, as well as for](#)
 127 [PnP-DRS and PnP-ADMM in the case where \$D_\sigma - I\$ is Lipschitz with Lipschitz constant less](#)
 128 [than 1](#) [64]. Various works show fixed-point convergence of PnP-ADMM and PnP-FBS when
 129 f has Lipschitz gradient under an ‘‘averaged denoiser’’ assumption, where $(1 - \theta)I + \theta D_\sigma$ is
 130 nonexpansive for some $\theta \in (0, 1)$, mainly using monotone operator theory [69, 70, 29]. Cohen
 131 et al. show fixed-point convergence of a relaxed PnP-FBS scheme when f has Lipschitz
 132 gradient under a demicontractive denoiser assumption, which is a strictly weaker condition
 133 than nonexpansiveness [17]. Sreehari et al. show convergence of PnP-ADMM to an implicitly
 134 defined convex function when the denoiser is nonexpansive and has symmetric gradient, by
 135 utilizing Moreau’s theorem to characterize the denoiser as a proximal map of a convex function
 136 [66, 48]. In the case of nonexpansive linear denoisers, PnP-FBS and PnP-ADMM converge to
 137 fixed points of a closed-form convex optimization problem [51].

138 While plentiful, many of these convergence results impose restrictive or difficult-to-verify
 139 conditions on the denoisers D_σ . Instead of replacing the regularizing proximal operator prox_g
 140 with a denoiser, Hurault et al. and Cohen et al. instead consider applying FBS with the
 141 proximal operator on the fidelity term and a gradient step on the regularization, $x^{k+1} =$
 142 $\text{prox}_f(I - \nabla g)(x^k)$ [32, 16]. Replacing the regularization step with a denoiser $D_\sigma = I - \nabla g_\sigma$
 143 results in the Gradient Step PnP (GS-PnP) algorithm $x^{k+1} = (\text{prox}_f \circ D_\sigma)(x^k)$. Using this
 144 parameterization, they show further that the fixed points of GS-PnP are stationary points of
 145 a particular (non-convex) function. Moreover, a follow-up work shows that a gradient-step
 146 denoiser of the form $D_\sigma = I - \nabla g_\sigma$ can be interpreted as a proximal step $D_\sigma = \text{prox}_{\phi_\sigma}$
 147 [33]. Using this, they are able to achieve iterate convergence under KL-type conditions to a
 148 stationary point of a (non-convex) closed-form functional of the form (1.1).

149 The GS-PnP style schemes require that the gradient of the potential ∇g_σ is Lipschitz with
 150 Lipschitz constant $L < 1$. Methods of training neural networks with Lipschitz constraints
 151 include spectral regularization, adversarial training against Lipschitz bounds during training,
 152 or spline based architectures [64, 46, 22, 52]. Hurault et al. consider fine-tuning the DRUNET
 153 denoiser by using spectral regularization to enforce the Lipschitz gradient condition [33]. While
 154 it can be shown empirically that the Lipschitz constant is less than one locally, there is no
 155 theoretical guarantee, which can lead to occasional divergence. One possible way of remedying

156 this is by averaging the denoiser with the identity operator, as remarked in [33]. This consists
 157 of replacing the denoiser $D_\sigma = I - \nabla g_\sigma$ with the relaxed $D_\sigma^\alpha := (1 - \alpha)I + \alpha D_\sigma = I - \alpha \nabla g_\sigma$
 158 for some $\alpha \in (0, 1)$. We can rewrite the relaxed denoiser as $D_\sigma^\alpha = I - \nabla g_\sigma^\alpha$, where $g_\sigma^\alpha = \alpha g_\sigma$
 159 has αL -Lipschitz gradient. Taking $\alpha < 1/L$ gives the appropriate contraction condition on g_σ^α
 160 and thus convergence of the associated PnP schemes [33, 31].

161 **1.3. Quasi-Newton Methods.** For minimizing a twice continuously differentiable function
 162 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, a classical second-order method is Newton's method [54]:

$$163 \quad (1.4) \quad x^{k+1} = x^k - (\nabla^2 f)^{-1} \nabla f(x^k),$$

164 where $\nabla^2 f$ is the Hessian of f . This can be interpreted as minimizing a local quadratic
 165 approximation

$$166 \quad (1.5a) \quad \hat{f}_k(y) = f(x^k) + \nabla f(x^k)^\top (y - x^k) + \frac{1}{2} (y - x^k)^\top \nabla^2 f(x^k) (y - x^k),$$

$$167 \quad (1.5b) \quad x^{k+1} = \arg \min_y \hat{f}_k(y).$$

169 Newton's method is able to achieve quadratic convergence rates with appropriate initialization
 170 and step-sizes [54]. However, the inverse of the Hessian may be computationally demanding,
 171 especially in high-dimensional applications such as image processing. Quasi-Newton (qN)
 172 methods propose to replace the inverse Hessian $(\nabla^2 f)^{-1}$ with (low-rank) approximations to
 173 the inverse Hessian, with notable examples including the Broyden-Goldfarb-Fletcher-Shanno
 174 (BFGS) algorithm, the David-Fletcher-Powell (DFP) formula, and the symmetric rank one
 175 method (SR1) [54].

176 Like Newton's method, quasi-Newton methods utilize the curvature information from the
 177 Hessian approximation to accelerate convergence, with applications in non-convex stochastic
 178 optimization, neural network training, and Riemannian optimization [13, 30, 75]. Classi-
 179 cal theory gives asymptotic superlinear convergence under the Dennis-Moré condition, which
 180 states that the Hessian approximation converges to the Hessian at the minimum [20]. Non-
 181 asymptotic convergence of quasi-Newton methods is still an active area of research. BFGS and
 182 DFP have only recently been shown to have non-asymptotic superlinear convergence rates of
 183 $\mathcal{O}((1/k)^{k/2})$ when the objective function is strongly convex with Lipschitz continuous gradi-
 184 ent, has Lipschitz continuous Hessian at the minimum, and satisfies a concordance condition
 185 [35, 61]. However, BFGS sees empirical success even when these conditions are not explic-
 186 itly verified, including in the non-convex setting [41, 42]. Interestingly, certain accelerated
 187 proximal gradient methods can be interpreted as a proximal quasi-Newton method [55].

188 Variants of BFGS include limited memory BFGS (L-BFGS), stochastic BFGS, greedy
 189 BFGS, and sharpened BFGS [43, 34, 47, 65, 60]. Of these variants, the limited memory
 190 version is most suited to repeated iteration. Standard quasi-Newton methods continually
 191 update the Hessian approximation using all the previous iterates, leading to a linear per-
 192 iteration computational cost increase. L-BFGS instead utilizes only the last m iterates, where
 193 $m > 1$ is a user-specified parameter, typically chosen to be less than 50. [Moreover, the](#)
 194 [Hessian need not be stored and/or computed at each iteration, as the method only relies on](#)
 195 [Hessian-vector products, which can be computed efficiently with two loop recursions \[54\].](#)

196 To relate quasi-Newton methods to the PnP framework described previously, we would like
 197 to consider applying Newton-type methods for convex composite optimization, by replacing
 198 a proximal operator with a denoiser. Lee et al. consider the problem of minimizing

$$199 \quad (1.6) \quad \varphi(x) = f(x) + g(x),$$

200 where $f(x)$ is a convex \mathcal{C}^1 function, and g is a possibly non-smooth convex regularizer [39].
 201 For a symmetric positive definite matrix $B_k \approx \nabla^2 f(x^k)$, the proximal Newton-type search
 202 direction Δx^k , satisfying $x^{k+1} = x^k + t_k \Delta x^k$, is given as the minimizer of a local quadratic
 203 approximation on the smooth component $\hat{f}_k(y)$:

$$204 \quad (1.7a) \quad \hat{f}_k(y) = f(x^k) + \nabla f(x^k)^\top (y - x^k) + \frac{1}{2} (y - x^k)^\top B_k (y - x^k),$$

$$205 \quad (1.7b) \quad \Delta x^k = \arg \min_d \hat{\varphi}_k(x^k + d) := \hat{f}_k(x^k + d) + g(x^k + d). \\ 206$$

207 Define the *scaled proximal map* for a positive definite matrix B as in [39]:

$$208 \quad (1.8) \quad \text{prox}_g^B(x) := \arg \min_{y \in \mathbb{R}^n} g(y) + \frac{1}{2} \|y - x\|_B^2,$$

209 where the B -norm is defined as $\|z\|_B^2 = z^\top B z$. For example, taking B to be the identity
 210 matrix results in the standard proximal map as defined in (1.2). The search direction (1.7b)
 211 has a closed form in terms of the scaled proximal map:

$$212 \quad (1.9) \quad \Delta x^k = \text{prox}_g^{B_k}(x - B_k^{-1} \nabla f(x^k)) - x^k.$$

213 With this search direction, appropriate step sizes and B_k , the proximal Newton-type methods
 214 are able to achieve similar convergence rates to Newton-type methods, achieving global con-
 215 vergence and local superlinear convergence. While the scaled proximal map allows for such
 216 analysis, it is not amenable to the PnP framework. For example, if we compute the Hessian
 217 approximation B_k using a BFGS-type approach, a naive approach of replacing $\text{prox}_g^{B_k}$ with
 218 a denoiser would require a careful analysis of the interaction of B_k on the resulting regu-
 219 larization, and possibly require the denoiser to depend on B_k . Instead, we seek a proximal
 220 Newton-type method that utilizes only the unscaled proximal map, with possibly a scalar con-
 221 stant which can be easily interpreted as a regularization parameter controlling the strength
 222 of regularization.

223 In Section 2, we will detail a classical composite minimization algorithm that uses only
 224 the unscaled proximal map prox_g , as well as arbitrary descent steps that allow for Newton-
 225 type steps. We further extend the classical analysis from convex to weakly convex functions,
 226 inspired by the GS-PnP characterization of denoisers as proximal maps of weakly convex
 227 functions. In Section 3, we use this extension to propose the PnP-quasi-Newton (PnP-qN)
 228 method, further convergence and characterizing cluster points of the algorithm. In Section 4,
 229 we evaluate the proposed PnP-qN method with the quasi-Newton method given by L-BFGS,
 230 and compare it with other [provable and non-provable PnP methods with comparable recon-](#)
 231 [struction quality](#).

232 **2. Proximal Quasi-Newton.** In this section, we will first describe a classical algorithm for
 233 optimizing composite sums of a (possibly non-convex) smooth function and a (possibly non-
 234 smooth) convex function. We will then extend the analysis to allow for *weak convexity* instead
 235 of *convexity*. By replacing proximal terms with deep denoisers corresponding to proximal
 236 operators of weakly convex maps, we construct a Plug-and-Play scheme with convergence
 237 properties of the classical algorithm.

238 Let us work on the Euclidean domain \mathbb{R}^n . Let $\mathcal{C}_{L_f}^{1,1}$ denote the class of \mathcal{C}^1 functions
 239 $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with L_f -Lipschitz gradient, and Γ_0 the class of proper, closed, and convex
 240 functions $g : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$. Consider a variational objective having the following form:

$$241 \quad (2.1) \quad \varphi = f + g, \quad f \in \mathcal{C}_{L_f}^{1,1}, \quad g \in \Gamma_0.$$

242 We can consider f as the fidelity term and g as a regularization term. A prominent example
 243 from inverse problems is the quadratic fidelity loss $f(x; y) = \frac{1}{2} \|Ax - y\|^2$ for some linear
 244 forward operator $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$ and observation $y \in \mathbb{R}^m$, where the norm is taken as the
 245 Euclidean norm.

246 **2.1. MINFBE: Minimizing Forward-Backward Envelope.** We first detail a classical com-
 247 posite optimization algorithm for minimizing (2.1), which will serve as the base of our proposed
 248 PnP scheme. Moreover, we describe some of its convergence properties that transfer to the
 249 PnP framework. By constructing a smooth convex envelope function around the original ob-
 250 jective φ , this envelope can be shown to have desirable properties such as sharing minimizers,
 251 smoothness, and being minorized and majorized by convex functions. By applying descent
 252 steps and proximal mappings in a particular fashion, the classical algorithm is able to obtain
 253 global objective convergence to critical points at a rate of $\mathcal{O}(1/k)$, local linear convergence if
 254 the function is locally strongly convex, and superlinear convergence when the descent steps
 255 are taken to be quasi-Newton with suitable assumptions [67].

256 For the problem (2.1), define the following expressions [67]:

$$257 \quad (2.2a) \quad l_\varphi(u, x) = f(x) + \langle \nabla f(x), u - x \rangle + g(u),$$

$$258 \quad (2.2b) \quad T_\gamma(x) = \arg \min_u \left\{ l_\varphi(u, x) + \frac{1}{2\gamma} \|u - x\|^2 \right\} = \text{prox}_{\gamma g}(x - \gamma \nabla f(x)),$$

$$259 \quad (2.2c) \quad R_\gamma(x) = \gamma^{-1}(x - T_\gamma(x)),$$

$$260 \quad (2.2d) \quad \varphi_\gamma(x) = \min_u \left\{ l_\varphi(u, x) + \frac{1}{2\gamma} \|u - x\|^2 \right\}.$$

262 Here, l_φ is a local linearized decoupling of φ , T_γ can be interpreted as an FBS step (with
 263 step-size γ for $f + g$) and R_γ is a scaled residual or “gradient direction”. Note that $x =$
 264 $T_\gamma(x) \Leftrightarrow x \in \text{zer } \partial \varphi$, i.e. fixed points of T_γ correspond to critical points of φ . φ_γ is defined as
 265 the *forward-backward envelope* of φ . We further explicitly write the Moreau envelope for g :

$$266 \quad (2.3a) \quad g^\gamma(x) = \min_u \left\{ g(u) + \frac{1}{2\gamma} \|u - x\|^2 \right\}$$

$$267 \quad (2.3b) \quad = g(\text{prox}_{\gamma g}(x)) + \frac{1}{2\gamma} \|\text{prox}_{\gamma g}(x) - x\|^2.$$

268

269 With the above definitions, we have the following closed-form expressions for the forward-
270 backward envelope:

$$271 \quad (2.4a) \quad \varphi_\gamma = f(x) + g(T_\gamma(x)) - \gamma \langle \nabla f(x), R_\gamma(x) \rangle + \frac{\gamma}{2} \|R_\gamma(x)\|^2$$

$$272 \quad (2.4b) \quad = f(x) - \frac{\gamma}{2} \|\nabla f(x)\|^2 + g^\gamma(x - \gamma \nabla f(x)).$$

274 In fact, φ_γ has many desirable properties, such as sharing minimizers with φ , and having an
275 easily computable derivative in terms of the Hessian of f .

276 **Proposition 2.1** ([67, Sec 2]). *The following holds:*

- 277 *i.* $\varphi(z) = \varphi_\gamma(z)$ for all $\gamma > 0$, $z \in \text{zer } \partial\varphi$;
- 278 *ii.* $\inf \varphi = \inf \varphi_\gamma$ and $\arg \min \varphi \subseteq \arg \min \varphi_\gamma$ for $\gamma \in (0, 1/L_f]$;
- 279 *iii.* $\arg \min \varphi = \arg \min \varphi_\gamma$ for all $\gamma \in (0, 1/L_f]$.

280 *Suppose additionally that f is \mathcal{C}^2 . Then φ_γ is \mathcal{C}^1 and the gradient of φ_γ can be written as*

$$281 \quad (2.5) \quad \nabla \varphi_\gamma(x) = (I - \gamma \nabla^2 f(x)) R_\gamma(x).$$

282 *Moreover, if $\gamma \in (0, 1/L_f)$, the set of stationary points of φ_γ equals $\text{zer } \partial\varphi$.*

283 Assuming that we are able to compute both φ_γ and φ , Proposition 2.1(i) allows us to
284 check whether we have converged to a stationary point of φ . Algorithm 2.1 is a classical
forward-backward algorithm for optimizing the nonsmooth composite objective (2.1).

Algorithm 2.1 MINFBE [67]

Require: $x^0, \gamma_0 > 0, \xi \in (0, 1), \beta \in [0, 1], k \leftarrow 0$

- 1: **if** $R_{\gamma_k}(x^k) = 0$ **then**
 - 2: stop
 - 3: **end if**
 - 4: Choose d^k s.t. $\langle d^k, \nabla \varphi_{\gamma_k}(x^k) \rangle \leq 0$
 - 5: Choose $\tau_k \geq 0$ and $w^k = x^k + \tau_k d^k$ s.t. $\varphi_{\gamma_k}(w^k) \leq \varphi_{\gamma_k}(x^k)$
 - 6: **if** $f(T_{\gamma_k}(w^k)) > f(w^k) - \gamma_k \langle \nabla f(w^k), R_{\gamma_k}(w^k) \rangle + \frac{(1-\beta)\gamma_k}{2} \|R_{\gamma_k}(w^k)\|^2$ **then**
 - 7: $\gamma_k \leftarrow \xi \gamma_k$, goto 1
 - 8: **end if**
 - 9: $x^{k+1} \leftarrow T_{\gamma_k}(w^k)$
 - 10: $\gamma_{k+1} \leftarrow \gamma_k$
 - 11: $k \leftarrow k + 1$, goto 1
-

285 In Algorithm 2.1, ξ is an Armijo backtracking parameter, while β is used to control the
286 strictness of the descent condition in Step 6. For appropriately chosen γ , the condition in Step
287 6 never holds, as stated in the next lemma. Moreover, the step-sizes γ_k are bounded below
288 by a constant in terms of σ , β and L_f . This guarantees that a step is always possible.
289

290 **Lemma 2.2** ([67, Lem 3.1]). *Let $(\gamma_k)_{k \in \mathbb{N}}$ be the sequence of step-size parameters in Algo-*
291 *gorithm 2.1, and let $\gamma_\infty = \min_{i \in \mathbb{N}} \gamma_i$. Then for all $k \geq 0$,*

$$292 \quad \gamma_k \geq \gamma_\infty \geq \min\{\gamma_0, \xi(1 - \beta)/L_f\}.$$

293 The MINFBE algorithm can be interpreted as a descent step (Step 5) followed by a FBS
 294 step (Step 9). In particular, note that the descent direction d^k does not have to be the
 295 direction of steepest descent, which allows for more flexibility in the algorithm. By combining
 296 the two of these steps together, the algorithm achieves global convergence as well as local
 297 linear convergence. This algorithm enjoys the following convergence guarantees.

298 **Definition 2.3 (Linear and Superlinear Convergence).** *We say a sequence $(x^k)_{k \in \mathbb{N}}$ converges*
 299 *to x_* ;*

300 *i. Q-linearly with factor $\omega \in [0, 1)$ if $\|x^{k+1} - x_*\| \leq \omega \|x^k - x_*\|$ for all $k \geq 0$;*

301 *ii. Q-superlinearly if $\|x^{k+1} - x_*\| / \|x^k - x_*\| \rightarrow 0$.*

302 *The convergence is R-linear (R-superlinear) if $\|x^k - x_*\| \leq a_k$ for some sequence $(a_k)_{k \in \mathbb{N}}$ s.t.*
 303 *$a_k \rightarrow 0$ Q-linearly (Q-superlinearly).*

304 **Theorem 2.4 ([67, Thm 3.6, 3.7]).** *Suppose that f is convex and that φ is coercive. In*
 305 *particular, suppose that the level set $\{x \in \mathbb{R}^n \mid \varphi(x) \leq \varphi(x^0)\}$ has diameter R , $0 < R < \infty$.*
 306 *Then for the sequences generated by Algorithm 2.1, either $\varphi(x^0) - \inf \varphi \geq R^2 / \gamma_0$ and*

$$307 \quad (2.6) \quad \varphi(x^1) - \inf \varphi \leq \frac{R^2}{2\gamma_0},$$

308 *or for any $k \in \mathbb{N}$, it holds that*

$$309 \quad (2.7) \quad \varphi(x^k) - \inf \varphi \leq \frac{2R^2}{k \min\{\gamma_0, \xi(1 - \beta)/L_f\}}.$$

310 *Suppose in addition that x_* is a strong minimizer of φ , i.e. there exists a neighborhood N*
 311 *of x_* and $c > 0$ such that for any $x \in N$,*

$$312 \quad \varphi(x) - \varphi(x_*) \geq \frac{c}{2} \|x - x_*\|^2.$$

313 *Then for sufficiently large k , $(\varphi(x^k))_{k \in \mathbb{N}}$ and $(\varphi_{\gamma_k}(w^k))_{k \in \mathbb{N}}$ converge Q-linearly to $\varphi(x_*)$ with*
 314 *factor ω , where*

$$315 \quad \omega \leq \max \left\{ \frac{1}{2}, 1 - \frac{c}{4} \min\{\gamma_0, \xi(1 - \beta)/L_f\} \right\} \in [1/2, 1),$$

316 *and $(x^k)_{k \in \mathbb{N}}$ converges R-linearly to x_* . If x_* is also a strong minimizer of φ_{γ_∞} where γ_∞ is*
 317 *defined as in Lemma 2.2, then $(\varphi(w^k))_{k \in \mathbb{N}}$ also converges R-linearly to x_* .*

318 In MINFBE, the initial descent step w^k can be chosen arbitrarily as long as the objective
 319 function decreases. Suppose now that the descent direction is chosen using a quasi-Newton
 320 method:

$$321 \quad d^k = -B_k^{-1} \nabla \varphi_\gamma(x^k).$$

322 If B_k are positive definite, then d^k are valid search directions. Assuming that B_k satisfy the
 323 Dennis-Moré condition [54, 20], we can get superlinear convergence of the iterates.

324 **Theorem 2.5** ([67, Thm 4.1]). Fix $\gamma > 0$. Suppose that $\nabla\varphi_\gamma$ is strictly differentiable at a
 325 stationary point $x_* \in \text{zer } \partial\varphi$, and that $\nabla^2\varphi_\gamma(x_*)$ is nonsingular. Let $(B_k)_{k \in \mathbb{N}}$ be a sequence of
 326 nonsingular $\mathbb{R}^{n \times n}$ matrices, and suppose the sequences

$$327 \quad (2.8) \quad w^k = x^k - B_k^{-1} \nabla\varphi_\gamma(x^k), \quad x^{k+1} = T_\gamma(w^k)$$

328 converge to x_* . If $x^k, w^k \notin \text{zer } \partial\varphi$ for all $k \geq 0$ and the Dennis-Moré condition

$$329 \quad (2.9) \quad \lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2\varphi_\gamma(x^k))(w^k - x^k)\|}{\|w^k - x^k\|} = 0$$

330 holds, then $(x^k)_{k \in \mathbb{N}}$ and $(w^k)_{k \in \mathbb{N}}$ converge Q -superlinearly to x_* .

331 If B_k are updated accordingly to the BFGS update step, then the updates as given in the
 332 previous theorem converge superlinearly to the minimum, under some additional assumptions
 333 on φ such as being convex with strong local minimum x_* , or satisfying a stronger Kurdyka-
 334 Lojasiewicz property at cluster points $\omega(x^0)$ [67, Thm 4.3]. Moreover, it can be shown that
 335 $\tau_k = 1$ is a valid step-size for sufficiently large k . For completeness, the BFGS update steps
 336 are given as below. Note that it is usually more practical to update the inverse Hessian
 337 approximation $H_k = B_k^{-1}$ [54].

$$338 \quad (2.10a) \quad s^k = w^k - x^k, \quad y^k = \nabla\varphi_\gamma(w^k) - \nabla\varphi_\gamma(x^k),$$

$$339 \quad (2.10b) \quad B_{k+1} = \begin{cases} B_k + \frac{y^k y^{k\top}}{y^{k\top} s^k} - \frac{B_k s^k (B_k s^k)^\top}{s^{k\top} B_k s^k} & \text{if } \langle s^k, y^k \rangle > 0, \\ B_k & \text{otherwise} \end{cases}.$$

$$340 \quad (2.10c) \quad H_{k+1} = \begin{cases} \left(I - \frac{s^k y^{k\top}}{y^{k\top} s^k}\right) H_k \left(I - \frac{y^k s^{k\top}}{y^{k\top} s^k}\right) + \frac{s^k s^{k\top}}{y^{k\top} s^k} & \text{if } \langle s^k, y^k \rangle > 0, \\ H_k & \text{otherwise} \end{cases}.$$

342 **2.2. Weakly-Convex Extension.** Suppose now that g is not convex, but instead is M -
 343 weakly convex. Recall that a function $g(x)$ is M -weakly convex if $g + M\|x\|^2/2$ is convex.
 344 For a M -weakly convex function g , we have for all x, y and $z \in \partial g(y)$ (where ∂g denotes the
 345 Clarke subdifferential of g),

$$346 \quad (2.11a) \quad g(x) \geq g(y) + \langle z, x - y \rangle - \frac{M}{2} \|x - y\|^2,$$

$$347 \quad (2.11b) \quad g(tx + (1-t)y) \leq tg(x) + (1-t)g(y) + \frac{M}{2} t(1-t) \|x - y\|^2.$$

349 In the following Section 3, we will model the proposed denoiser $D_\sigma = \text{prox}_g$ as the **proximal**
 350 **operator** of a weakly convex function. In particular, a gradient step denoiser $D_\sigma = I - \nabla g_\sigma$
 351 with contractive ∇g_σ is the **proximal operator** of a weakly convex function [31]. We can extend
 352 the classical convex analysis to this case as well, albeit with a smaller allowed γ .

353 To transfer the results from the previous section to the case where g is weakly convex, we
 354 are required to check that the function values at the MINFBE iterates are non-increasing. As
 355 we will show in the following proposition, this is still the case for sufficiently small γ . Many
 356 properties of the forward-backward envelope still hold, and we are still able to attain global
 357 convergence and superlinear local convergence, subject to the Dennis-Moré condition (2.9).

358 **Proposition 2.6.** For all $x \in \mathbb{R}^n$, $\gamma > 0$,

359 *i.* $\varphi_\gamma(x) \leq \varphi(x) - \frac{\gamma - M\gamma^2}{2} \|R_\gamma(x)\|^2$;

360 *ii.* $\varphi(T_\gamma(x)) \leq \varphi_\gamma(x) - \frac{\gamma}{2}(1 - \gamma L_f) \|R_\gamma(x)\|^2$ for all $\gamma > 0$;

361 *iii.* $\varphi(T_\gamma(x)) \leq \varphi_\gamma(x)$ for all $\gamma \in (0, 1/L_f]$.

362 **Proof. (i).** By the optimality condition in (2.2b), we have

$$363 \quad R_\gamma(x) - \nabla f(x) \in \partial g(T_\gamma(x)).$$

364 By (2.11a), we have

$$365 \quad g(x) \geq g(T_\gamma(x)) + \langle R_\gamma(x) - \nabla f(x), x - T_\gamma(x) \rangle - \frac{M}{2} \|x - T_\gamma(x)\|^2$$

$$366 \quad = g(T_\gamma(x)) - \gamma \langle \nabla f(x), R_\gamma(x) \rangle + \gamma \|R_\gamma(x)\|^2 - \frac{M\gamma^2}{2} \|R_\gamma(x)\|^2.$$

367

368 Adding $f(x)$ to both sides and applying (2.4a) gives the result.

369 **(ii), (iii).** The proof is identical to that in [67, Prop 2.2], requiring only the Lipschitz
370 convexity of ∇f . ■

371 **Proposition 2.7.** Suppose $\gamma - M\gamma^2 \geq 0$, or equivalently $\gamma \in [0, 1/M]$. Then the following
372 hold:

373 *i.* $\varphi_\gamma(z) = \varphi(z)$ for all $z \in \text{zer } \partial \varphi$;

374 *ii.* $\inf \varphi = \inf \varphi_\gamma$ and $\arg \min \varphi \subseteq \arg \min \varphi_\gamma$ for $\gamma \in (0, 1/L_f]$;

375 *iii.* $\arg \min \varphi = \arg \min \varphi_\gamma$ for $\gamma \in (0, 1/L_f]$.

376 **Proof. (i).** Proposition 2.6(i) combined with the condition $\gamma - M\gamma^2 \geq 0$ shows $\varphi_\gamma(x) \leq$
377 $\varphi(x)$. If $z \in \text{zer } \partial \varphi$, then $z = T_\gamma(z)$, and Proposition 2.6(ii) reads $\varphi(z) \leq \varphi_\gamma(z)$.

378 **(ii), (iii).** Identical to [67, Prop 2.3]. ■

379 With weakly convex functions, we are still able to provide a lower bound on the γ such
380 that the condition in Step 6 of Algorithm 2.1 does not hold, [removing the need to reduce step-](#)
381 [sizes](#). The proof relies only on the Lipschitz constant of ∇f and does not require convexity of
382 g . However, we require that $\gamma - M\gamma^2 \geq 0$. In practice, the denoisers we use have $M < 1/2$,
383 which allows for any $\gamma \in (0, 1)$.

384 **Lemma 2.8.** Suppose g is weakly convex. If $0 < \gamma < \min\{(1 - \beta)/L_f, 1/M\}$, then the
385 condition in Step 6 in Algorithm 2.1 never holds. Moreover, this implies MINFBE iterations
386 satisfy $\gamma_k \geq \gamma_\infty \geq \min\{\gamma_0, \xi(1 - \beta)/L_f, 1/M\} > 0$ for all k .

387 **Proof.** Suppose $0 < \gamma < \min\{(1 - \beta)/L_f, 1/M\}$, and for contradiction that the condition
388 in Step 6 holds. Then there exists some w such that

$$389 \quad f(T_\gamma(w)) > f(w) - \gamma \langle \nabla f(w), R_\gamma(w) \rangle + \frac{(1 - \beta)\gamma}{2} \|R_\gamma(w^k)\|^2.$$

390 Adding $g(T_\gamma(w))$ to both sides and considering (2.4a), this becomes

$$391 \quad \varphi(T_\gamma(w)) > \varphi_\gamma(w) - \frac{\beta\gamma}{2} \|R_\gamma(w)\|^2.$$

392 But from Proposition 2.6(ii), we also have

$$\begin{aligned}
 393 \quad \varphi(T_\gamma(w)) &\leq \varphi_\gamma(w) - \frac{\gamma}{2}(1 - \gamma L_f) \|R_\gamma(w)\|^2 \\
 394 \quad &\leq \varphi_\gamma(w) - \frac{\beta\gamma}{2} \|R_\gamma(w)\|^2, \\
 395
 \end{aligned}$$

396 where the second inequality follows from $\gamma < (1 - \beta)/L_f$, giving a contradiction. The second
 397 part holds since $(\gamma_k)_{k \in \mathbb{N}}$ is a non-increasing sequence. \blacksquare

398 *Remark 2.9.* While $\gamma < 1/M$ is not strictly needed for the proof of the above lemma, this
 399 requirement is needed for convergence in future results.

400 The following theorem characterizes the convergence of the functional φ , which relies on
 401 the non-increasing condition of Step 5 in Algorithm 2.1. This is an analogue of [67, Prop 3.4].

402 **Theorem 2.10.** *Suppose $0 < \gamma_0 < 1/M$. Then the MINFBE iterations satisfy the following:*

- 403 i. $\varphi(x^{k+1}) \leq \varphi(x^k) - \frac{\beta\gamma_k}{2} \|R_{\gamma_k}(w^k)\|^2 - \frac{\gamma_k - M\gamma_k^2}{2} \|R_{\gamma_k}(x^k)\|^2$;
- 404 ii. *Either the sequence $\|R_{\gamma_k}(x^k)\|$ is square-summable, or $\varphi(x^k) \rightarrow \inf \varphi = -\infty$ and the*
 405 *set $\omega(x^0)$ of cluster points of the sequence $(x^k)_{k \in \mathbb{N}}$ is empty.*
- 406 iii. $\omega(x^0) \subseteq \text{zer } \partial\varphi$;
- 407 iv. *If $\beta > 0$, then either the sequence $\|R_{\gamma_k}(w^k)\|$ is square-summable and every cluster*
 408 *point of $(w^k)_{k \in \mathbb{N}}$ is critical, or $\varphi_{\gamma_k}(w^k) \rightarrow \inf \varphi = -\infty$ and $(w^k)_{k \in \mathbb{N}}$ has no cluster*
 409 *points.*

410 *Proof. (i).* Recalling $x^{k+1} = T_{\gamma_k}(w^k)$,

$$\begin{aligned}
 411 \quad \varphi(x^{k+1}) &\leq \varphi_{\gamma_k}(w^k) - \frac{\beta\gamma_k}{2} \|R_{\gamma_k}(w^k)\|^2 \\
 412 \quad (2.12) \quad &\leq \varphi_{\gamma_k}(x^k) - \frac{\beta\gamma_k}{2} \|R_{\gamma_k}(w^k)\|^2
 \end{aligned}$$

$$\begin{aligned}
 413 \quad (2.13) \quad &\leq \varphi(x^k) - \frac{\beta\gamma_k}{2} \|R_{\gamma_k}(w^k)\|^2 - \frac{\gamma_k - M\gamma_k^2}{2} \|R_{\gamma_k}(x^k)\|^2, \\
 414
 \end{aligned}$$

415 where the first and second inequalities come from Step 6 and 5 in Algorithm 2.1 respectively,
 416 and the final inequality is Proposition 2.6(i).

417 **(ii)-(iv).** We follow [67] with minor modifications. Let $\varphi_* = \lim_{k \rightarrow \infty} \varphi(x^k)$, which exists
 418 as $(\varphi(x^k))_{k \in \mathbb{N}}$ is monotone by (i) and $\gamma_k - M\gamma_k^2 \geq 0$. If $\varphi_* = -\infty$, then $\inf \varphi = -\infty$. By
 419 properness and lower semi-continuity of φ , as well as the monotonicity of $\varphi(x^k)$, no cluster
 420 points of $(x^k)_{k \in \mathbb{N}}$ exist. If instead $\varphi_* > -\infty$, by telescoping (2.13),

$$421 \quad (2.14) \quad \frac{1}{2} \sum_{i=0}^k \gamma_i (\beta \|R_{\gamma_i}(w^i)\|^2 + (1 - \gamma_i M) \|R_{\gamma_i}(x^i)\|^2) \leq \varphi(x^0) - \varphi(x^{k+1}) \leq \varphi(x^0) - \varphi_*.$$

422 Since γ_k is uniformly bounded below by Lemma 2.8, we have square summability of $\|R_{\gamma_k}(x^k)\|$,
 423 showing (ii).

424 By square summability, $R_{\gamma_k}(x^k) \rightarrow 0$. Moreover, the functions $R_{\gamma_k} = R_{\gamma_\infty}$ are constant for
 425 sufficiently large k , and R_{γ_∞} is continuous by continuity of the proximal operator and of ∇f .

426 Therefore, any cluster point $z \in \omega(x^k)$ has $R_{\gamma_\infty}(x^{k_j}) \rightarrow R_{\gamma_\infty}(z) = 0$ for some subsequence
 427 $x^{k_j} \rightarrow z$. Thus $z = T_{\gamma_\infty}(z) \Rightarrow z \in \text{zer } \partial\varphi$, showing (iii).

428 If $\beta > 0$, for sufficiently large k such that $\gamma_k = \gamma_\infty$, the following chain of inequalities
 429 holds:

$$430 \quad (2.15) \quad \varphi_{\gamma_k}(w^{k+1}) \leq \varphi_{\gamma_k}(x^{k+1}) = \varphi_{\gamma_k}(T_k(w^k)) \leq \varphi_{\gamma_k}(w^k).$$

431 The first inequality comes from Step 5, the equality from Step 9, and the final inequality
 432 from Proposition 2.6. The monotonicity of $\varphi_{\gamma_k}(w^k)$ for sufficiently large k allows for a similar
 433 argument to hold for the w^k sequence, giving (iv). ■

434 Convergence results can also be extended to the weakly convex case. In particular, the fol-
 435 lowing theorem shows the convergence of the residuals between each step.

436 **Theorem 2.11 (Global Residual Convergence).** *Suppose $0 < \gamma_0 \leq 1/(2M)$, and let $c =$
 437 $\min\{\gamma_0, \xi(1 - \beta)/L_f, 1/M\} > 0$ be the lower bound for γ_∞ . The MINFBE iterations satisfy*

$$438 \quad (2.16) \quad \min_{i \leq k} \|R_{\gamma_i}(x^i)\|^2 \leq \frac{2}{k+1} \frac{\varphi(x^0) - \inf \varphi}{c - Mc^2}.$$

439 *If in addition $\beta > 0$, then we also have*

$$440 \quad (2.17) \quad \min_{i \leq k} \|R_{\gamma_i}(w^i)\|^2 \leq \frac{2}{k+1} \frac{\varphi(x^0) - \inf \varphi}{\beta c}.$$

441 *Proof.* As in [67, Thm 3.5]. If $\inf \varphi = -\infty$, there is nothing to prove, so suppose otherwise
 442 that $\inf \varphi > -\infty$. Considering (2.14) along with $(\gamma_k)_{k \in \mathbb{N}}$ being nonincreasing implies

$$443 \quad (2.18) \quad \frac{(k+1)(\gamma_k - M\gamma_k^2)}{2} \min_{i \leq k} \|R_{\gamma_i}(x^i)\|^2 + \frac{(k+1)\beta\gamma_k}{2} \min_{i \leq k} \|R_{\gamma_i}(w^i)\|^2 \leq \varphi(x^0) - \inf \varphi.$$

444 Now note that $\gamma - M\gamma^2$ is increasing for $\gamma < 1/(2M)$, so $\gamma_k - M\gamma_k^2$ is lower bounded by
 445 $c - Mc^2 > 0$. Rearranging yields both inequalities. ■

446 To obtain convergence of the objective similar to Theorem 2.4, it is insufficient for g
 447 to be weakly convex. We can alternatively utilize the KL property, which is a useful and
 448 general property satisfied by a large class of functions, including semialgebraic functions [4].
 449 Moreover, it can be used to show convergence in the absence of other regularity conditions
 450 such as convexity [5, 10, 33].

451 **Definition 2.12 (KL Property [5, 10]).** *Suppose $\varphi : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$ is proper and lower semi-
 452 continuous. φ satisfies the Kurdyka-Łojasiewicz (KL) property at a point x_* in $\text{dom } \partial\varphi$ if
 453 there exists $\eta \in (0, +\infty]$, a neighborhood U of x_* and a continuous concave function $\Psi :$
 454 $[0, \eta) \rightarrow [0, +\infty)$ such that:*

- 455 1. $\Psi(0) = 0$;
- 456 2. Ψ is \mathcal{C}^1 on $(0, \eta)$;
- 457 3. $\Psi'(s) > 0$ for $s \in (0, \eta)$;
- 458 4. For all $u \in U \cap \{\varphi(x_*) < \varphi(u) < \varphi(x_*) + \eta\}$, we have

$$459 \quad \varphi'(\varphi(u) - \varphi(x_*)) \text{dist}(0, \partial\varphi(u)) \geq 1.$$

460 We say that φ is a KL function if the KL property is satisfied at every point of $\text{dom } \partial\varphi$.

461 Utilizing the KL property, we are able to show that the iterates generated by MINFBE are
 462 sufficiently well-behaved, and hence converge. Moreover, from Theorem 2.10, we have that the
 463 iterates converge to critical points of the non-convex objective φ . Under the PnP scheme, this
 464 will correspond to convergence to critical points of some function determined by the denoiser.

465 **Theorem 2.13.** *Suppose that f satisfies the KL condition and g is semialgebraic, and both
 466 f and g are bounded from below. Suppose further that there exist constants $\bar{\tau}, c > 0$ such that
 467 $\tau_k < \bar{\tau}$ and $\|d^k\| \leq c\|R_{\gamma_k}(x^k)\|$, $\beta > 0$, and that φ is coercive or has compact level sets. Then
 468 the sequence of iterates $(x^k)_{k \in \mathbb{N}}$ is either finite and ends with $R_{\gamma_k}(x^k) = 0$, or converges to a
 469 critical point of φ .*

470 *Proof.* Deferred to the supplementary material. The proof is very similar to that in [67,
 471 Thm 3.9, Appendix 4]. ■

472 The crux of using the MINFBE method is that we are able to incorporate Newton-type
 473 steps into the iterations. Since we are able to get convergence to a critical point from the pre-
 474 vious theorem, we are in a position to apply the next theorem to show superlinear convergence
 475 in a neighborhood of a minimizer.

476 **Theorem 2.14.** *Suppose that f is continuously differentiable with L_f -Lipschitz gradient and
 477 g is M -weakly convex. Let $\gamma = \gamma_\infty$ as in Lemma 2.8. Suppose the search directions are chosen
 478 as*

$$479 \quad d^k = -B_k^{-1} \nabla \varphi_\gamma(x^k),$$

480 *the step-sizes in Step 5 are chosen with $\tau_k = 1$ tried first, and B_k satisfy the Dennis-Moré
 481 condition (2.8). Suppose further that the iterates $(x^k)_{k \in \mathbb{N}}$, $(w^k)_{k \in \mathbb{N}}$ converge to a critical point
 482 x_* at which $\nabla \varphi_\gamma$ is continuously differentiable with $\nabla^2 \varphi_\gamma(x_*) \succ 0$. Then $(x^k)_{k \in \mathbb{N}}$ and $(w^k)_{k \in \mathbb{N}}$
 483 converge Q -superlinearly to x_* .*

484 *Proof.* The proof is nearly identical to [67, Thm 4.1]. If γ_g is M -weakly convex, then for
 485 $\gamma < 1/M$, $u \mapsto \left(g(u) + \frac{1}{2\gamma}\|u - x\|^2\right)$ is strongly convex. Thus $\text{prox}_{\gamma g}$ is 1-Lipschitz [59]. The
 486 rest of the proofs of Thm 4.1 and 4.2 of [67] follows as usual. ■

487 This shows superlinear convergence instead of linear convergence in the case where the critical
 488 point is a strong local minimum, i.e. it is locally strongly convex. Note the differentiability
 489 condition in the second part can be dropped if f and g are both \mathcal{C}^2 . Moreover, assuming
 490 either φ is convex and x_* is a strong local minimum, or φ satisfies a stronger KL inequality,
 491 these conditions indeed hold [if \$B_k\$ is updated according to the BFGS scheme](#) [67, Thm 4.3].

492 **3. PnP-qN: Deep Denoiser Extension.** To convert Algorithm 2.1 to the PnP framework,
 493 we consider replacing the proximal step in (2.2b) with a denoiser. In particular, we consider
 494 the gradient-step denoiser setup in [33]. Let the denoiser D_σ be given by

$$495 \quad (3.1a) \quad D_\sigma = I - \nabla g_\sigma,$$

$$496 \quad (3.1b) \quad g_\sigma = \frac{1}{2}\|x - N_\sigma(x)\|^2,$$

497

498 where g_σ is a \mathcal{C}^2 function with L -Lipschitz gradient with $L < 1$. Note the subscript in g_σ
 499 represents a denoising strength, as opposed to the forward-backward envelope of g as we will
 500 define for our problem later. The mapping $N_\sigma(x)$ takes the form of a \mathcal{C}^2 neural network,
 501 allowing for the computation of g_σ explicitly. Under these assumptions, the denoiser D_σ takes
 502 the form of a proximal mapping of a weakly convex function, as stated in the next proposition.

503 **Proposition 3.1** ([31, Prop 1]). $D_\sigma(x) = \text{prox}_{\phi_\sigma}(x)$, where ϕ_σ is defined by

$$504 \quad (3.2) \quad \phi_\sigma(x) = g_\sigma(D_\sigma^{-1}(x)) - \frac{1}{2}\|D_\sigma^{-1}(x) - x\|^2$$

505 if $x \in \text{Im}(D_\sigma)$, and $\phi_\sigma(x) = +\infty$ otherwise. Moreover, ϕ_σ is $\frac{L}{L+1}$ -weakly convex.

506 This proposition allows us to take the weak convexity constant required in the previous section
 507 as $M = L/(L + 1)$. Since $L < 1$, we have $M < 1/2$. This result can be thought of a slight
 508 extension of the fact that a function f is a proximal operator of some proper convex l.s.c.
 509 function φ , if and only if it is a subgradient of a convex l.s.c. function ψ and f is nonexpansive
 510 [27, 48].

511 Suppose that $\gamma_k = \gamma > 0$ is fixed in the MINFBE iterations, satisfying the conditions in
 512 Lemma 2.8. Consider making the substitution with ϕ_σ defined as in Proposition 3.1, targeting
 513 $\varphi = f + g$:

$$514 \quad (3.3) \quad \gamma g = \phi_\sigma.$$

515 The FBS step $T_\gamma(x) = \text{prox}_{\gamma g}(x - \gamma \nabla f(x))$ thus becomes, using $D_\sigma = \text{prox}_{\phi_\sigma}$,

$$516 \quad (3.4) \quad T_\gamma(x) = D_\sigma(x - \gamma \nabla f(x)).$$

517 This will target the objective function $\varphi(x) = f(x) + g(x) = f(x) + \phi_\sigma(x)/\gamma$. To iterate
 518 Algorithm 2.1 with this substitution, we need to evaluate φ_γ . Recalling (2.4b), we can instead
 519 evaluate the Moreau envelope g^γ . By definition (2.3b) and the substitution (3.3), we have:

$$\begin{aligned} 520 \quad g^\gamma(y) &\stackrel{(2.3b)}{=} g(\text{prox}_{\gamma g}(y)) + \frac{1}{2\gamma}\|\text{prox}_{\gamma g}(y) - y\|^2 \\ 521 \quad &\stackrel{(3.3)}{=} \frac{1}{\gamma}\phi_\sigma(D_\sigma(y)) + \frac{1}{2\gamma}\|D_\sigma(y) - y\|^2 \\ 522 \quad &\stackrel{(3.2)}{=} \frac{1}{\gamma}g_\sigma(D_\sigma^{-1}(D_\sigma(y))) - \frac{1}{2\gamma}\|D_\sigma^{-1}(D_\sigma(y)) - D_\sigma(y)\|^2 + \frac{1}{2\gamma}\|D_\sigma(y) - y\|^2 \\ 523 \quad &= \frac{1}{\gamma}g_\sigma(y). \\ 524 \end{aligned}$$

525 Using this substitution, we obtain the Plug-and-Play scheme PnP-MINFBE, detailed in Al-
 526 gorithm 3.1. We have a closed form for the forward-backward envelope of φ , as well as some

527 other expressions essential for iterating MINFBE, given by:

$$528 \quad (3.5a) \quad \varphi(x) = f(x) + \frac{1}{\gamma}\phi_\sigma(x),$$

$$529 \quad (3.5b) \quad \varphi_\gamma(x) = f(x) - \frac{\gamma}{2}\|\nabla f(x)\|^2 + \frac{1}{\gamma}g_\sigma(x - \gamma\nabla f(x)),$$

$$530 \quad (3.5c) \quad \nabla\varphi_\gamma(x) = (I - \gamma\nabla^2 f)R_\gamma(x),$$

$$531 \quad (3.5d) \quad \varphi(x^{k+1}) = f(x^{k+1}) + \frac{1}{\gamma} \left(g_\sigma(w^k - \gamma\nabla f(w^k)) - \|w^k - \gamma\nabla f(w^k) - T_\gamma(w^k)\|^2/2 \right).$$

532

Algorithm 3.1 PnP-MINFBE

Require: $x^0, \gamma < \min\{\gamma_0, (1 - \beta)/L_f, 1/M\}, \beta \in [0, 1], k \leftarrow 0$

- 1: **if** $R_{\gamma_k}(x^k) = 0$ **then**
 - 2: stop
 - 3: **end if**
 - 4: Choose d^k s.t. $\langle d^k, \nabla\varphi_\gamma(x^k) \rangle \leq 0$
 - 5: Choose $\tau_k \geq 0$ and $w^k = x^k + \tau_k d^k$ s.t. $\varphi_\gamma(w^k) \leq \varphi_\gamma(x^k)$
 - 6: $x^{k+1} \leftarrow D_\sigma(w^k - \gamma\nabla f(w^k))$
 - 7: $k \leftarrow k + 1$, goto 1
-

533 To compute the search direction d^k at each step, we can use a quasi-Newton method
 534 to approximate the inverse Hessian of φ_γ . While a closed form exists for $\nabla^2\varphi_\gamma$, [such as in](#)
 535 [\[67, Thm 2.10\]](#), it requires the Jacobian of the denoiser D_σ , rendering methods requiring the
 536 Hessian computationally intractable due to the dimensionality of our problems. Therefore,
 537 we resort to a BFGS-like algorithm using the differences and secants

$$538 \quad s^k = w^k - x^k, \quad y^k = \nabla\varphi_\gamma(w^k) - \nabla\varphi_\gamma(x^k).$$

539 In particular, we will use the L-BFGS method due to the memory restrictions imposed by using
 540 images for our experiments. This can be implemented using a two-loop recursion, using only
 541 the last m secants computed [54]. We additionally impose a safeguard to reject updating the
 542 Hessian approximation if the secant condition $\langle s^k, y^k \rangle > 0$ is not satisfied. For completeness,
 543 we write the two-loop recursion for L-BFGS in Algorithm 3.2. The initial (inverse) Hessian
 544 approximations are chosen as $H_0^k = c_k I$ as in [54], given by

$$545 \quad c_k = \frac{\langle s^{k-1}, y^{k-1} \rangle}{\langle y^{k-1}, y^{k-1} \rangle}.$$

546 Utilizing the results from the previous section, we can show the following convergence
 547 results for [PnP-MINFBE \(Algorithm 3.1\)](#) and [PnP-LBFGS \(Algorithm 3.3\)](#).

548 **Corollary 3.2.** *Suppose that f is \mathcal{C}^1 and KL with L_f -Lipschitz gradient, g_σ is \mathcal{C}^2 and semi-*
 549 *algebraic with L_g -Lipschitz gradient with $L_g < 1$. Assume further that $\gamma < 1/(2M)$ is chosen*
 550 *as in Lemma 2.8 such that $\gamma = \gamma_\infty$, and there exist $\bar{\tau}, c > 0$ such that $\tau_k \leq \bar{\tau}$ and $\|d^k\| \leq$
 551 $c\|R_\gamma(x^k)\|$. Then the PnP-MINFBE iterations of Algorithm 3.1 satisfy the following:*

Algorithm 3.2 L-BFGS [54]

Require: $m > 0$, secants $(s^i)_{i=k-m}^{k-1}$, differences $(y^i)_{i=k-m}^{k-1}$, initial Hessian guesses $(H_0^k)_{k \in \mathbb{N}}$

- 1: $q \leftarrow \nabla \varphi_\gamma(x^k)$
- 2: $\rho_i \leftarrow 1/\langle y^i, s^i \rangle$ for $i = k-1, k-2, \dots, k-m$
- 3: **for** $i = k-1, k-2, \dots, k-m$ **do**
- 4: $\alpha_i \leftarrow \rho_i \langle s^i, q \rangle$
- 5: $q \leftarrow q - \alpha_i y^i$
- 6: **end for**
- 7: $r \leftarrow H_0^k q$
- 8: **for** $i = k-m, k-m+1, \dots, k-1$ **do**
- 9: $\beta \leftarrow \rho_i \langle y^i, r \rangle$
- 10: $r \leftarrow r + (\alpha_i - \beta) s^i$
- 11: **end for**
- 12: **stop with** $B_k^{-1} \nabla \varphi_\gamma(x^k) = H^k \nabla \varphi_\gamma(x^k) = r$

Algorithm 3.3 PnP-LBFGS

Require: $x^0, \gamma < \min\{(1-\beta)/L_f, 1/M\}, \beta \in [0, 1], k \leftarrow 0$

- 1: **if** $R_{\gamma_k}(x^k) = 0$ **then**
- 2: stop
- 3: **end if**
- 4: Compute $d^k \leftarrow -B_k^{-1} \nabla \varphi_\gamma(x^k)$ using L-BFGS (c.f. Algorithm 3.2) with differences and secants $(s^i, y^i)_{i=k-m}^{k-1}$.
- 5: Choose $\tau_k \in [0, 1]$ and $w^k = x^k + \tau_k d^k$ s.t. $\varphi_\gamma(w^k) \leq \varphi_\gamma(x^k)$
- 6: $x^{k+1} \leftarrow D_\sigma(w^k - \gamma \nabla f(w^k))$
- 7: $s^k \leftarrow w^k - x^k, y^k \leftarrow \nabla \varphi_\gamma(w^k) - \nabla \varphi_\gamma(x^k)$
- 8: $k \leftarrow k+1$, goto 1

- 552 *i.* $\varphi(x^k)$ decreases monotonically;
- 553 *ii.* The residuals $R_\gamma(x^k)$ converge to zero at a rate $\mathcal{O}(1/\sqrt{k})$;
- 554 *iii.* If the iterates are bounded, then the iterates are either finite or converge to a critical
- 555 point of $\varphi = f + \frac{1}{\gamma} \phi_\sigma$. Moreover, $\varphi = \varphi_\gamma$ at these critical points.
- 556 *iv.* If furthermore $d^k = -B_k^{-1} \nabla \varphi_\gamma(x^k)$ and the B_k satisfy the Dennis-Moré condition
- 557 (2.8), then the x^k and w^k converge superlinearly to x_* .

558 *Proof.* (i), (ii). Follows from Theorems 2.10 and 2.11. (iii). By the Tarski-Siedenber

559 theorem [5], compositions and inverses of semi-algebraic mappings are semi-algebraic. There-

560 fore D_σ and D_σ^{-1} are semi-algebraic (on their domain), and hence so is ϕ_σ . Therefore,

561
$$\varphi = f + \frac{1}{\gamma} \phi_\sigma$$

562 is a KL function. Moreover, φ_γ is also a KL function. So we have convergence by Theorem 2.13.

563 The final part follows from Proposition 2.7. (iv). Follows from Theorem 2.14. ■

Table 1: Hyperparameters for PnP-LBFGS.

	Deblur			SR		
	2.55	7.65	12.75	2.55	7.65	12.75
σ						
α	0.5	0.5	0.7	0.5	0.5	0.5
γ			1			
β			0.01			
λ	1	1	1	4	1.5	1
σ_d/σ	1	0.75	0.75	2	1	0.75

Table 2: Hyperparameters for PnP- $\hat{\alpha}$ PGD.

	Deblur			SR		
	2.55	7.65	12.75	2.55	7.65	12.75
σ						
α	0.6	0.8	0.85	1	1	1
L_f		1			0.25	
λ			$(\alpha + 1)/(\alpha L_f)$			
$\hat{\alpha}$			$1/(\lambda L_f)$			
σ_d/σ	1.5	1	1	2	2	2

564 *Remark 3.3.* An essential part of the classical proof relies on the fact that $\tau = 1$ will
565 eventually always be accepted in MINFBE, under a Newton-type descent direction choice.
566 During numerical testing, we observed that the Armijo search for τ was only occasionally
567 necessary when the image is being optimized, with at most 10 line searches required before
568 converging.

569 In our case, f will be a quadratic fidelity term of the form $f(x) = \|Ax - y\|^2/2$ for some linear
570 operator A and measurement y . This is semi-algebraic and hence KL, and moreover trivially
571 bounded below. From (3.1b), we additionally have that g_σ is bounded below. Since N_σ will
572 take the form of a neural network which is a composition of semi-algebraic operations and
573 arithmetic operations, g_σ will also be semi-algebraic. Therefore, we can apply Corollary 3.2
574 and get convergence to critical points of the associated function $\varphi = f + \frac{1}{\gamma}\phi_\sigma$.

575 **4. Experiments.** In this section, we consider the application of the proposed PnP-LBFGS
576 method, given by Algorithm 3.3, with a pre-trained denoiser to image deblurring and super-
577 resolution. We use the pretrained Lipschitz-constrained proximal denoiser given in [33]. The
578 (gradient-step) denoiser takes the form (3.1), where N_σ is a neural network based on the
579 DRUNet architecture [77]. The Lipschitz constraint on ∇g_σ is enforced by applying a penalty
580 on the spectral norm of $\nabla^2 g_\sigma$ during training. While this spectral constraint affects the
581 performance of the end-to-end denoiser, it provides sufficient conditions for convergence in
582 the context of PnP, in particular, convergence to a critical point of a closed-form functional.

583 The datasets we consider for image reconstruction are the CBSD68, CBSD10 and set3c
 584 datasets¹, containing images of size 256×256 with three color channels and pixel intensity
 585 values in $[0, 255]$ [45]. The forward operators corresponding to [the considered reconstruction](#)
 586 [problems of deblurring and super-resolution](#) are linear, and we can write the fidelity term as
 587 $f(x) = \lambda \|Ax - y\|^2/2$, where A is the degradation operator, y is the degraded image, and λ is
 588 a regularization parameter. For reconstruction, y will be taken as $y = Ax_{\text{true}} + \varepsilon$, where x_{true}
 589 is the ground-truth image and the noise ε is pixel-wise Gaussian with standard deviations
 590 $\sigma \in \{2.55, 7.65, 12.75\}$ corresponding to 1%, 3%, and 5% noise (relative to the maximum pixel
 591 intensity value), respectively. The underlying optimization problems corresponding to fixed
 592 points of PnP-MINFBE thus take the form (as in (3.5a)):

$$593 \quad (4.1) \quad \min_x \varphi(x) = \frac{\lambda}{2} \|Ax - y\|^2 + \frac{1}{\gamma} \phi_\sigma,$$

594 where $\gamma \leq \min\{(1 - \beta)/L_f, 1/2M\}$ as in Lemma 2.8 and Theorem 2.11. In this case, f is \mathcal{C}^2 ,
 595 and we can easily compute the derivative [of the forward-backward envelope using](#) (3.5c).

596 The methods we compare against are PnP methods with similar convergence guarantees,
 597 namely $\mathcal{O}(1/\sqrt{k})$ residual convergence and a KL-type iterate convergence [33]. [Our analysis](#)
 598 [additionally shows](#) superlinear convergence to minima with positive-definite Hessian using
 599 Newton’s directions. Although we can not verify whether the Hessian approximation B_k
 600 obtained via L-BFGS satisfies the Dennis-Moré condition for superlinear convergence, we
 601 will empirically demonstrate faster convergence in terms of both time and iteration count
 602 compared to the competing methods.

603 The PnP methods that we will compare against are the PnP-PGD, PnP-DRS, PnP-
 604 DRSDiff and PnP- $\hat{\alpha}$ PGD methods [33, 31]. Here PGD stands for proximal gradient descent,
 605 DRS for Douglas-Rachford splitting, DRSDiff for DRS with differentiable fidelity terms, and
 606 $\hat{\alpha}$ PGD for $\hat{\alpha}$ -relaxed PGD. The update rules corresponding to the chosen PnP methods for
 607 comparison are as follows:

$$608 \quad (\text{PnP-PGD}) \quad \begin{cases} z^{k+1} = x^k - \lambda \nabla f(x^k) \\ x^{k+1} = D_\sigma(z^{k+1}) \end{cases}$$

$$609 \quad (\text{PnP-DRSDiff}) \quad \begin{cases} y^{k+1} = \text{prox}_{\lambda f}(x^k) \\ z^{k+1} = D_\sigma(2y^{k+1} - x^k) \\ x^{k+1} = x^k + (z^{k+1} - y^{k+1}) \end{cases}$$

$$610 \quad (\text{PnP-DRS}) \quad \begin{cases} y^{k+1} = D_\sigma(x^k) \\ z^{k+1} = \text{prox}_{\lambda f}(2y^{k+1} - x^k) \\ x^{k+1} = x^k + (z^{k+1} - y^{k+1}) \end{cases}$$

$$611 \quad (\text{PnP-}\hat{\alpha}\text{PGD}) \quad \begin{cases} q^{k+1} = (1 - \hat{\alpha})y^k + \hat{\alpha}x^k \\ x^{k+1} = D_\sigma(x^k - \lambda \nabla f(q^{k+1})) \\ y^{k+1} = (1 - \hat{\alpha})y^k + \hat{\alpha}x^{k+1} \end{cases}$$

612
613

¹<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>

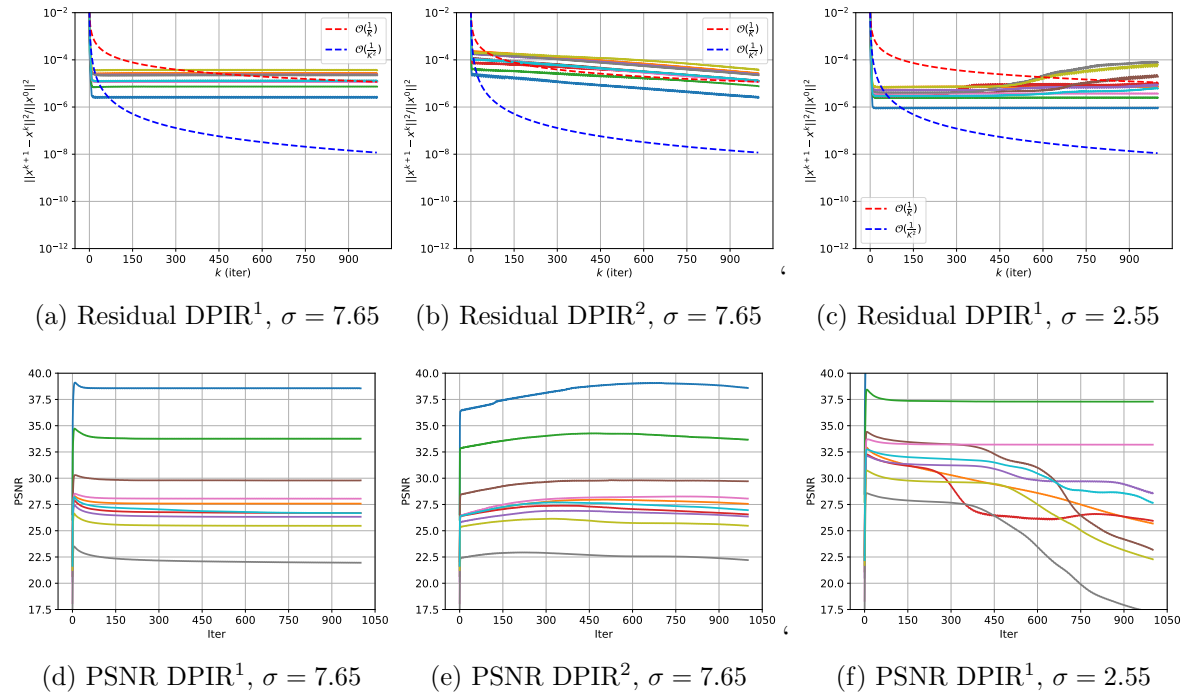


Figure 1: Performance of DPIR measured in terms of residual $\|x^{k+1} - x^k\|^2 / \|x^0\|^2$ and PSNR for deblurring with noise levels $\sigma = 2.55, 7.65$, applied with two different denoiser strength regimes. Each curve corresponds to one of the 10 images from the CBSD10 dataset. DPIR¹ has denoiser strength decreased from 49 to σ over 8 iterations for deblurring, and extended with $\sigma_d = \sigma$ for following iterations. DPIR² has denoiser strength decreased from 49 to σ over 1000 iterations. We observe that both methods have decreasing PSNR at later iterations and non-converging residual, and further that DPIR diverges for small noise levels.

614 **4.1. Hyperparameter and Denoiser Choices.** The hyperparameters for the proposed
 615 PnP-LBFGS and the existing PnP- $\hat{\alpha}$ PGD methods are as in Tables 1 and 2, respectively,
 616 chosen via grid search to maximize the PSNR over the set3c dataset for the respective image
 617 reconstruction problems. The hyperparameter grid for PnP-LBFGS is given in the subsequent
 618 subsections, while the grid for PnP- $\hat{\alpha}$ PGD is given below. For the denoiser in our experiment,
 619 we use the pre-trained network N_σ as in [33].

620 The convergence conditions for PnP-PGD and PnP-DRSdiff are that g_σ has L -Lipschitz
 621 gradient for some $L < 1$, and directly using the denoiser D_σ maintains theoretical convergence.
 622 For PnP-DRS, the condition needs to be strengthened to $L < 1/2$. In this case, the denoiser is
 623 replaced with an averaged denoiser of the form $(I + D_\sigma)/2 = I - \frac{1}{2}\nabla g_\sigma$, which gives convergence
 624 results but changes the underlying optimization problem. For PnP-LBFGS and PnP- $\hat{\alpha}$ PGD,
 625 we use an averaged denoiser $D_\sigma^\alpha = I - \alpha\nabla g_\sigma$ which appears to have better performance, with
 626 the relaxation parameter α chosen as in Tables 1 and 2. As remarked in the introduction,
 627 adding the relaxation parameter α means that the effective Lipschitz constant of the potential

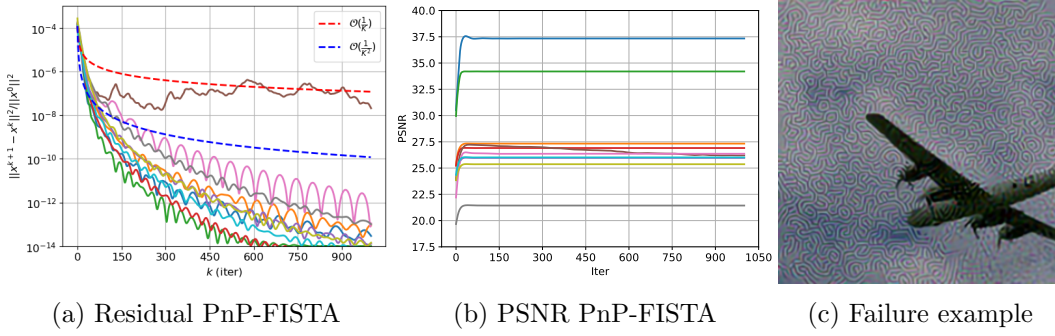


Figure 2: Residual $\|x^{k+1} - x^k\|^2 / \|x^0\|^2$ and PSNR for PnP-FISTA applied to super-resolution with noise level $\sigma = 7.65$. Each curve corresponds to one of the 10 images from the CBSD10 dataset. Using the parameters of PnP-LBFGS, which should resolve any Lipschitz constraint issues, has the same divergence issue. PnP-FISTA sometimes fails, leading to images with artifacts as seen in subfigure (c).

628 gradient $\alpha \nabla g_\sigma$ is αL , which alleviates divergence issues when $L > 1$. In this case, $D_\sigma^\alpha = \text{prox}_{\phi_\sigma^\alpha}$
 629 for some weakly convex ϕ_σ^α , and the previous computations hold with g_σ replaced with αg_σ .

630 For the parameters of the relaxed PnP- $\hat{\alpha}$ PGD algorithm, we perform a grid search as in
 631 [31]. To obtain the values of the denoiser averaging parameter α and the denoiser strength
 632 σ_d , we do a grid search for the set3c dataset with $\alpha \in \{0.6, 0.7, 0.8, 0.85, 0.9, 1.0\}$ and $\sigma_d / \sigma \in$
 633 $\{0.5, 0.75, 1.0, 1.5, 2.0\}$, where the noise level is $\sigma = 7.65$. The main difficulty in finding these
 634 hyperparameters is the dependence between α and σ_d , leading to poor reconstructions for
 635 many of these values. Given the denoiser averaging parameter α , the other hyperparameters
 636 of PnP- $\hat{\alpha}$ PGD are given by $\lambda = \frac{\alpha+1}{\alpha L_f}$, $\hat{\alpha} = \frac{1}{\lambda L_f}$.

637 For the Lipschitz constant, we take $L_f = 1$ for deblurring and $L_f = 1/4$ for super-
 638 resolution with $s_{sr} = 2, 3$, as in Subsections 4.3 and 4.4. It appears approximating $L_f = 1$ for
 639 super-resolution or $L_f = 1/9 = 1/s_{sr}^2$ for $s_{sr} = 3$ results in divergence, indicating sensitivity to
 640 their hyperparameters. We find the best values to be as in Table 2, with the grid search taken
 641 to maximize the PSNR over the set3c dataset. We additionally employ a stopping criterion
 642 based on the Lyapunov functional that PnP- $\hat{\alpha}$ PGD minimizes, with the same sensitivity as
 643 PnP-DRS and PnP-DRSdiff [31].

644 The regularization parameter λ for the underlying optimization problem is restricted for
 645 PnP-LBFGS in a manner similar to PnP-PGD and PnP-DRS (but not PnP-DRSdiff). For
 646 PnP-PGD and PnP-DRS, one condition for convergence is that $\lambda L_f < 1$ [33]. However, for
 647 PnP-LBFGS, Lemma 2.8 gives the condition that $\gamma < (1 - \beta) / (\lambda L_f)$, targeting stationary
 648 points of

649
$$\varphi(x) = \frac{\lambda}{2} \|Ax - y\|^2 + \frac{1}{\gamma} \phi_\sigma.$$

650 We note that as λ increases, the allowed γ decreases, which correspondingly increases the
 651 smallest allowed coefficient $1/\gamma$ of the prior ϕ_σ at the same rate as λ . This puts an upper

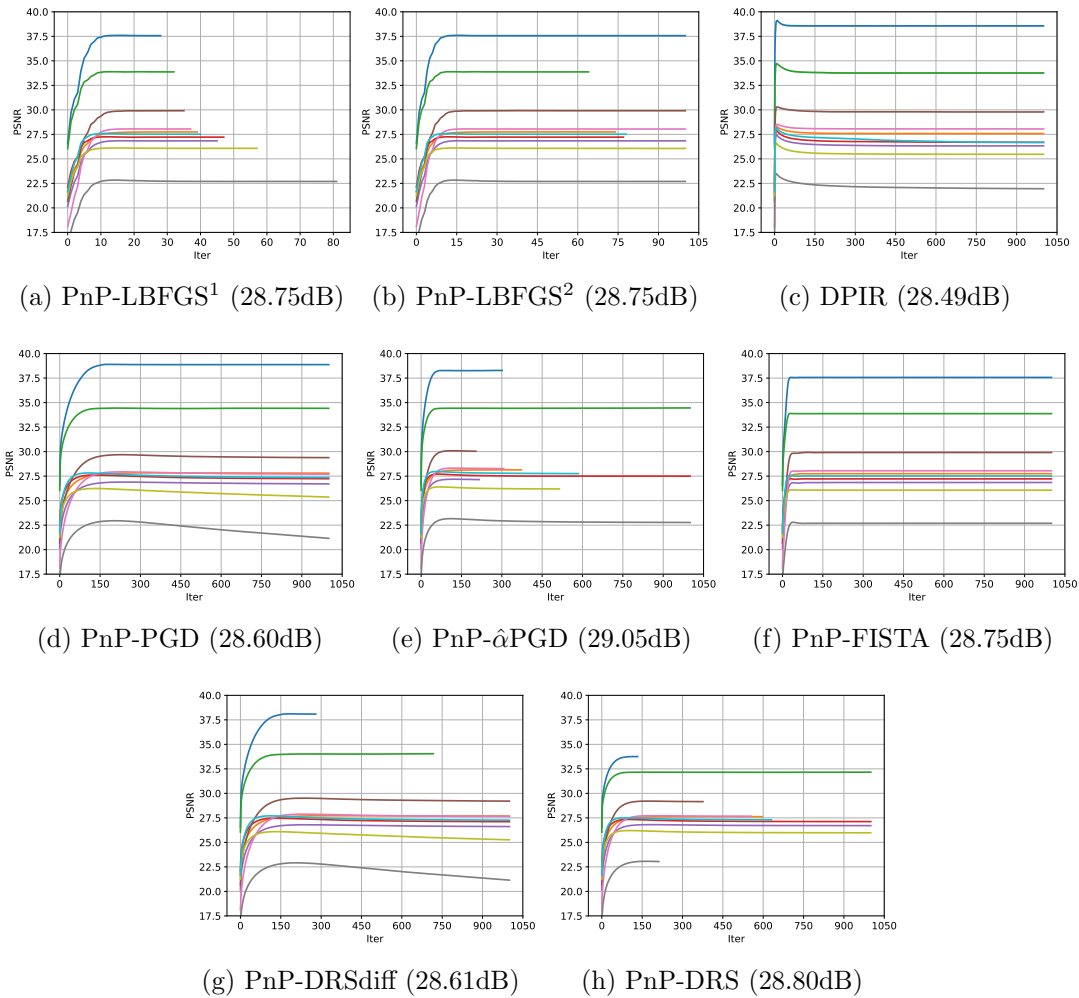


Figure 3: Convergence of the PSNRs for deblurring, with the average dB in brackets. Each curve corresponds to one of the 10 images from the CBSD10 dataset. Note that the scale of (a) is 10 times smaller than the other curves, terminating at 100 instead of 1000. PnP-LBFGS and PnP-DRS have generally more stable convergence, which can be attributed to the smaller Lipschitz constant of $I - D_\sigma$. PnP-LBFGS¹ also converges in much fewer iterations than the compared methods. The average PSNR between PnP-LBFGS with the two stopping criteria differ by only 0.0013dB.

652 bound on the ratio between the fidelity term and the regularization term, which may be
 653 restrictive for low-noise applications.

654 The memory length for LBFGS was chosen to be $m = 20$, with a maximum of 100
 655 iterations per image. The denoiser D_σ^α is chosen with denoising strength σ_d similar to that
 656 used for PnP-DRS as in [33]. By using different denoising strengths, we are able to further

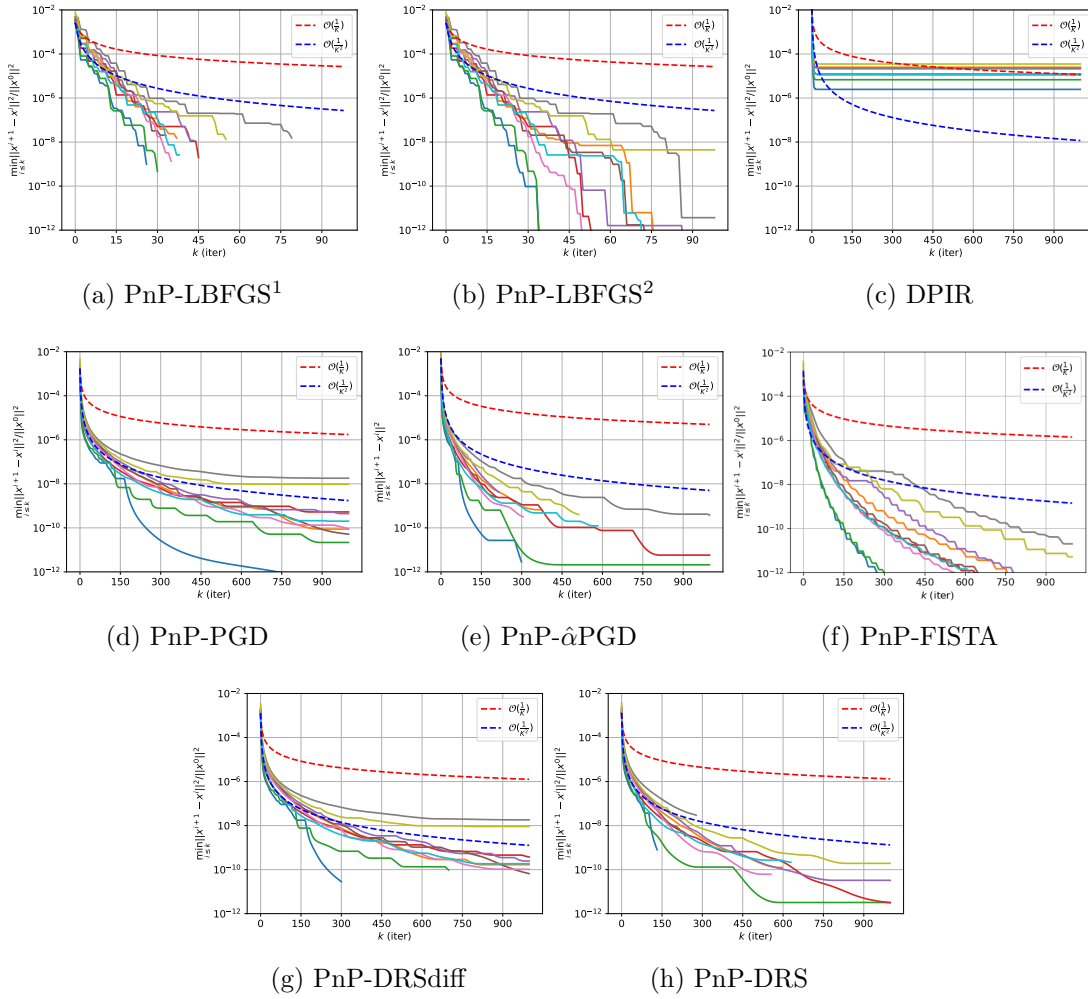


Figure 4: Convergence of the residuals $\min_{i \leq k} \|x^{i+1} - x^i\|^2 / \|x^0\|^2$ of the various methods for deblurring. Each curve corresponds to one of the 10 images from the CBSD10 dataset, evaluated with the first blur kernel and $\sigma = 7.65$. Note that the x-axis scale of (a) is 10 times smaller than the other curves, terminating at 100 instead of 1000.

657 control regularization along with the scaling parameter λ . The step-sizes τ_k are chosen using
 658 an Armijo line search starting from $\tau_k = 1$, and multiplying by 0.5 if the φ_γ decrease condition
 659 in Step 5 of Algorithm 3.3 is not met [3, 8].

660 We additionally introduce a stopping criterion based on the differences between consecu-
 661 tive iterates of the envelope $\varphi_\gamma(x^{k+1}) - \varphi_\gamma(x^k) < 10^{-5}$, as well as the envelope and objective
 662 $\varphi(x^k) - \varphi_\gamma(x^k) < 5 \times 10^{-5}$, where we stop if at least one criterion is met for 5 iterations
 663 in a row. We note that while the criteria can be strengthened, there is minimal change in
 664 the optimization result. We label PnP-LBFGS with the envelope-based stopping criterion as

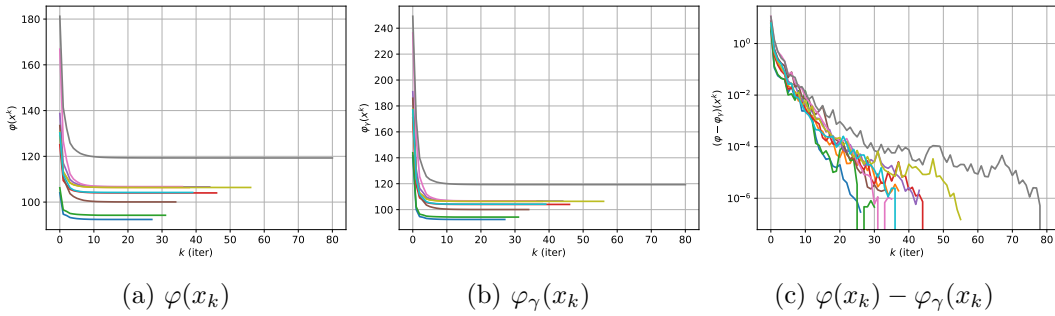


Figure 5: Evolution of the objective φ , forward-backward envelope φ_γ , and their difference $\varphi - \varphi_\gamma$ for deblurring with PnP-LBFGS¹. These values are equal at the true minima, i.e., $\varphi_\gamma(x_*) = \varphi(x_*)$. Each curve corresponds to one of the 10 images from the CBSD10 dataset, evaluated with the first blur kernel and $\sigma = 7.65$.

665 PnP-LBFGS¹. For completeness, we also consider the stopping criterion when the relative dif-
 666 ference between consecutive function values of φ is less than 10^{-8} . We label PnP-LBFGS with
 667 the objective change stopping criterion as PnP-LBFGS². The PnP-LBFGS algorithms with
 668 the two stopping criteria are labeled with superscripts, as PnP-LBFGS¹ and PnP-LBFGS²,
 669 respectively. We further use PnP-LBFGS without superscripts to refer to both methods to-
 670 gether, which share their parameters.

671 All implementations were done in PyTorch, and the experiments were performed on an
 672 AMD EPYC 7352 CPU and a Quadro RTX 6000 GPU with 24GB of memory [56]. The code
 673 for our experiments are publicly available².

674 **4.2. PnP Methods Without Convergence Guarantees.** For further comparison, we ad-
 675 ditionally consider two non-provable PnP methods, namely DPIR [77] and PnP-FISTA [38].
 676 DPIR is based on the half-quadratic splitting, which splits prox_{f+g} into alternating prox_f
 677 and prox_g steps, and further replaces prox_g with a denoising step D_{σ_k} in the spirit of PnP.
 678 PnP-FISTA is based on the fast iterative shrinkage-thresholding algorithm, which arises by
 679 applying a Nesterov-style acceleration to the forward-backward splitting [38, 37]. We note that
 680 neither of these methods correspond to critical points of functions in the existing literature.

$$\begin{aligned}
 681 \quad (\text{DPIR}) \quad & \begin{cases} \alpha_k = \hat{\lambda}\sigma^2/\sigma_k^2, \\ x_{k+1} = \text{prox}_{f/2\alpha_k}(z_k), \\ z_{k+1} = D_{\sigma_k}(x_k). \end{cases} \\
 682 \quad (\text{PnP-FISTA}) \quad & \begin{cases} x_k = D_\sigma(y_k - \lambda\nabla f(y_k)), \\ t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \\ y_{k+1} = x_k + \frac{t_k-1}{t_{k+1}}(x_k - x_{k-1}). \end{cases} \\
 683 \quad &
 \end{aligned}$$

²<https://github.com/hyt35/Prox-qN>

684 **4.2.1. DPIR.** To improve the performance, DPIR uses a decreasing noise regime as well
 685 as image transformations during iteration [77, Sec. 4.2]. To extend past eight iterations, we
 686 consider using the log-scale noise from $\sigma_d = 49$ to $\sigma_d = \sigma$ over 8 and 24 iterations for deblurring
 687 and super-resolution respectively, as recommended in the DPIR paper [77, Sec. 5.1.1, 5.2].
 688 The scaling for the proximal term is determined by a scaling parameter $\hat{\lambda}$, which was chosen
 689 to be $\hat{\lambda} = 0.23$ in the original work. Figure 1 shows that while DPIR achieves state-of-the-art
 690 performance in the low iteration regime, the PSNR begins to drop when HQS is extended
 691 past the number of iterations used in the original DPIR paper [32]. Moreover, DPIR appears
 692 to have poor performance in the low noise regime for the following image reconstruction
 693 experiments. **In the following experiments, we consider DPIR with the suggested 8 and 24**
 694 **iterations for deblurring and super-resolution respectively, as well as extending up to 1000**
 695 **iterations to check the convergence behavior.**

696 **4.2.2. PnP-FISTA.** The denoiser parameters for PnP-FISTA are considered to be either
 697 the parameters for PnP-LBFGS or PnP-PGD. **Proofs for PnP schemes** such as PnP-PGD
 698 or PnP-DRS **generally** rely on classical monotone operator theory, and showing that the
 699 denoiser satisfies the necessary assumptions. However, proofs of convergence of FISTA depend
 700 heavily on the convexity of the problem [9, 14], and non-convex proofs additionally require
 701 techniques or conditions such as adaptive backtracking [24, 55] or quadratic growth conditions
 702 [6]. These techniques and conditions are difficult to convert and verify in the PnP regime,
 703 which translates to difficulties in showing convergence of the associated PnP-FISTA schemes.

704 In the following experiments, we run the DPIR and PnP-FISTA methods for 1000 itera-
 705 tions unless stated otherwise to verify the convergence behavior. Figures 1 and 2 additionally
 706 demonstrate some common modes of divergence for DPIR and PnP-FISTA, with DPIR failing
 707 for low noise levels and PnP-FISTA failing with artifacts.

708 **4.3. Deblurring.** For deblurring, 10 blur kernels were used, including eight camera shake
 709 kernels, a 9×9 uniform kernel, and a 25×25 Gaussian kernel with standard deviation $\sigma_{\text{blur}} =$
 710 1.6 [40, 33]. Visualizations of the kernels can be found in the supplementary material. The
 711 blurring operator A corresponds to convolution with circular boundary conditions. In this
 712 case, the transpose A^\top can be easily implemented using a transposed convolution with circular
 713 boundary conditions. The blurring operator was previously scaled to have $\|A^\top A\|_{\text{op}} \approx 0.96$,
 714 which was verified using a power iteration. Thus, ∇f is approximately 0.96λ -Lipschitz.

715 We chose hyperparameters of PnP-LBFGS following a grid search maximizing the PSNR
 716 on the set3c dataset. The parameter grids are $\alpha \in \{0.5, 0.7, 0.9, 1.0\}$, $\lambda \in \{0.8, 0.9, 1.0\}$, $\gamma \in$
 717 $\{0.8, 0.85, 0.9, 1.0\}$, and $\sigma_d/\sigma \in \{0.5, 0.75, 1.0, 1.5, 2.0\}$. Note that this choice obeys $\gamma <$
 718 $\min\{(1 - \beta)/L_f, 1/(2M)\}$, since φ_σ is at most $1/2$ -weakly convex. We observe empirically
 719 that the step-size $\tau = 1$ is also a valid descent almost all of the time, verifying the claim that
 720 is required to prove the superlinear convergence as remarked in Remark 3.3. The underlying
 721 optimization problems are slightly different for PnP-LBFGS and PnP-PGD: for PnP-PGD,
 722 the fidelity regularization is chosen to be $\lambda = 0.99$, and the iterates converge to cluster points
 723 of $\varphi_{\text{PnP-PGD}}$:

$$724 \quad \varphi_{\text{PnP-LBFGS}} = \frac{1}{2}\|Ax - y\|^2 + \phi_\sigma^\alpha, \quad \varphi_{\text{PnP-PGD}} = \frac{0.99}{2}\|Ax - y\|^2 + \phi_\sigma.$$

725 We observe in Table 3 that the PnP-PGD and PnP-DRSdiff converge to very similar results

Table 3: Table of average PSNR (dB) comparing existing [provable and non-provable](#) PnP methods evaluated on the CBSD68 dataset compared to the proposed PnP-LBFGS methods. The time shown is the average reconstruction time per image. The PnP-LBFGS¹ method is significantly faster per image due to the faster convergence [compared to the other provable PnP methods](#).

σ	2.55	7.65	12.75	Time (s)
PnP-LBFGS ¹	31.19	27.95	26.61	5.80
PnP-LBFGS ²	31.17	27.78	26.61	9.55
PnP-PGD	30.57	27.80	26.61	25.93
PnP-DRSdiff	30.57	27.78	26.61	22.72
PnP-DRS	31.54	28.07	26.60	19.26
PnP- $\hat{\alpha}$ PGD	31.52	28.15	26.74	15.66
PnP-FISTA	30.24	27.15	26.60	24.32
DPIR (iter 10 ³)	27.40	27.58	26.46	19.62
DPIR (iter 8)	32.01	28.34	26.86	0.55

726 since they both minimize the same underlying functional. However, the PnP iterations some-
 727 times do not converge, as demonstrated by the steadily decreasing PSNR in subfigures (d)
 728 and (g) of Figure 3. This can be attributed to the Lipschitz constant of g_σ being greater
 729 than 1 at these iterates. [The use of the averaged denoiser \$D_\sigma^\alpha\$](#) in PnP-DRS and PnP-LBFGS
 730 [reduces divergence](#), where we see convergence for these images as well. We generally observe
 731 that PnP- $\hat{\alpha}$ PGD has the best performance in terms of PSNR, which can be attributed to
 732 the larger allowed value of λ . Nonetheless, we observe significantly faster convergence for
 733 PnP-LBFGS compared to the other methods to comparable PSNR values for each test image.

734 Comparing with the non-provable PnP methods, we observe in Figure 3 that PnP-FISTA
 735 converges to the same PSNR as PnP-LBFGS [on CBSD10](#), but has a worse performance [when](#)
 736 [averaged over](#) all CBSD68 images in Table 3. This can be attributed to divergence of the
 737 method for denoisers where the Lipschitz constant of ∇g_σ is greater than 1. DPIR instead
 738 reaches its peak in the first couple of iterations, before decreasing to the fixed point as iterated
 739 by the denoiser with the final denoising strength $\sigma_d = \sigma$. [This results in worse performance](#)
 740 [of DPIR at iteration 10³ as compared to iteration 8, demonstrating the non-convergence and](#)
 741 [the current gap in performance between provable PnP and non-provable PnP.](#)

742 Figure 3 and Figure 4 additionally demonstrate the difference between the stopping cri-
 743 teria. The stopping criteria of PnP-LBFGS¹ is sufficient for convergence to a reasonable
 744 PSNR, and allows for much earlier stopping. PnP-LBFGS² stops after more iterates and
 745 demonstrates the significantly faster convergence of the residuals compared to the other con-
 746 sidered PnP methods. Moreover, Figure 5 shows the convergence curves of the objective
 747 φ and forward-backward envelope φ_γ , which rapidly converge to the same value, verifying
 748 Proposition 2.1.

749 **4.4. Super-resolution.** For super-resolution, we consider the forward operator with scale
 750 $s_{sr} \in \{2, 3\}$ as $A = SK : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{\lfloor n/s_{sr} \rfloor \times \lfloor n/s_{sr} \rfloor}$, which is a composition of a downsam-

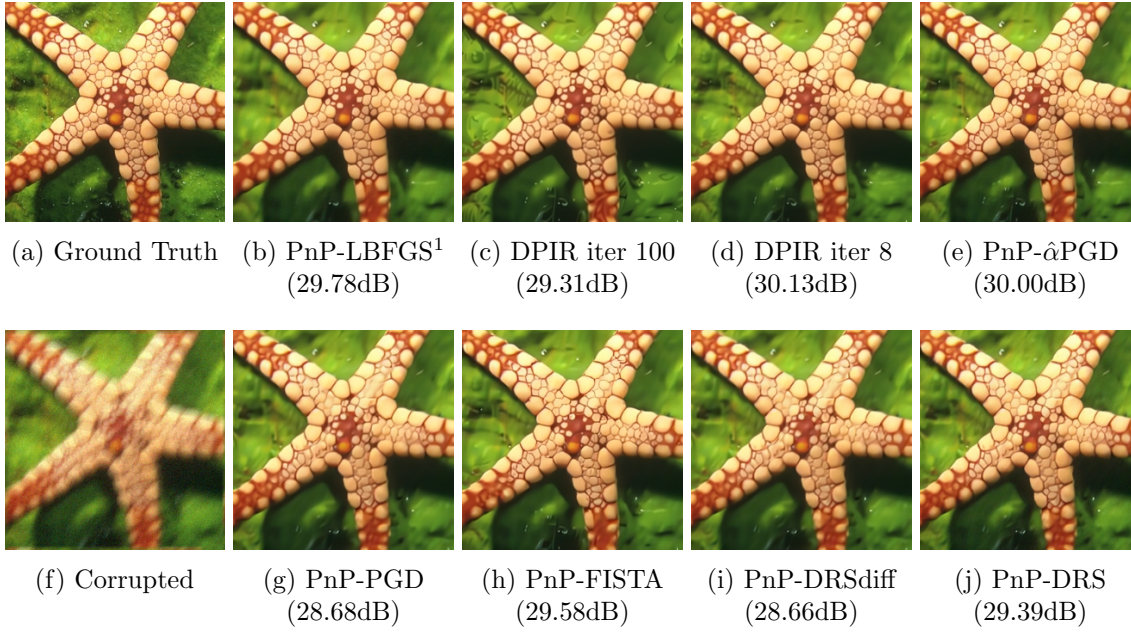


Figure 6: Deblurring visualization using starfish image, with each method limited to a maximum of 100 iterations. Experiments are run with additive Gaussian noise $\sigma = 7.65$. PnP-LBFGS¹ converges within the first 100 iterations, while the other PnP algorithms take longer to converge. Since the result of PnP-LBFGS¹ and PnP-LBFGS² are nearly identical, we show only PnP-LBFGS¹. DPIR starts to decrease in PSNR after 8 iterations, leading to slightly worse performance.

751 pling operator $S : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{\lfloor n/s_{sr} \rfloor \times \lfloor n/s_{sr} \rfloor}$ and a circular convolution $K : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$.
 752 The convolutions K are Gaussian blur kernels with blur strength given by standard devia-
 753 tions $\sigma_{\text{blur}} = \{0.7, 1.2, 1.6, 2.0\}$ as in [77, 33]. For the PnP-LBFGS parameters, we chose
 754 hyperparameters maximizing the PSNR using a grid search on the set3c dataset over the fol-
 755 lowing ranges: $\alpha \in \{0.5, 0.7, 0.9, 1.0\}$, $\lambda \in \{1.0, 2.0, 3.0, 4.0\}$, $\gamma \in \{0.8, 0.85, 0.9, 1.0\}$, and
 756 $\sigma_d/\sigma \in \{0.5, 0.75, 1.0, 1.5, 2.0\}$.

757 The Hessian $\nabla^2 f = \lambda A^\top A = \lambda K^\top S^\top S K$ is easily available, as $S^\top S : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n \times n}$ is
 758 a mask operator comprised of setting pixels with index not in $(s_{sr}\mathbb{Z})^2$ to zero, and K^\top is a
 759 transposed convolution with circular boundary conditions. Note that on the image manifold,
 760 $S^\top S$ is approximately $1/s_{sr}^2$ -Lipschitz, as we set $(s_{sr}^2 - 1)/s_{sr}^2$ of the pixels to zero. With K
 761 being approximately 1-Lipschitz, we have that $A^\top A$ is approximately $1/s_{sr}^2$ -Lipschitz.

762 The PnP-LBFGS parameters are $\beta = 0.01, \gamma = 1$, and $\lambda = 2, 1.5, 1$ for noise levels
 763 $\sigma = 2.55, 7.65, 12.75$ respectively. We can take [these values of \$\lambda\$](#) since $L_f \approx 1/s_{sr}^2 \leq 1/4$ and
 764 $\gamma = 1$ still obeys $\gamma < \min\{(1 - \beta)/L_f, 1/(2M)\}$. The underlying functionals are as follows:

$$765 \quad \varphi_{\text{PnP-LBFGS}} = \frac{\lambda_{\text{LBFGS}}}{2} \|Ax - y\|^2 + \phi_\sigma^\alpha, \quad \varphi_{\text{PnP-PGD}} = \frac{0.99}{2} \|Ax - y\|^2 + \phi_\sigma.$$

Table 4: Table of averaged PSNR (dB) corresponding to the competing PnP methods evaluated on the CBSD68 dataset for super-resolution, as compared with the proposed PnP-LBFGS method. The time is the average reconstruction time per image for $\sigma = 7.65$. The performance of PnP-LBFGS is almost identical to the compared provable PnP methods due to minimizing the same variational form, but with faster convergence.

Scale σ	$s = 2$				$s = 3$			
	2.55	7.65	12.75	Time (s)	2.55	7.65	12.75	Time (s)
PnP-LBFGS ¹	27.89	26.62	25.80	3.19	26.12	25.32	24.68	4.80
PnP-LBFGS ²	27.89	26.62	25.80	9.81	26.12	25.30	24.68	13.15
PnP-PGD	27.44	26.57	25.82	25.99	25.60	25.20	24.63	37.33
PnP-DRSdiff	27.44	26.58	25.82	18.24	25.60	25.19	24.63	32.83
PnP-DRS	27.93	26.61	25.79	15.74	26.13	25.29	24.67	27.00
PnP- $\hat{\alpha}$ PGD	27.94	26.62	25.72	4.24	26.11	25.32	24.69	8.78
PnP-FISTA	26.38	26.44	25.79	24.61	24.96	25.15	24.63	33.13
DPIR (iter 10 ³)	18.58	26.36	25.74	19.58	17.53	24.96	24.55	19.67
DPIR (iter 24)	27.82	26.60	25.85	0.98	26.06	25.29	24.67	0.97

766 We observe in Table 4 that the results for PnP-LBFGS are comparable to the other
767 provable PnP methods, with overall faster wall-clock times. In Figure 7 and Figure 8, we
768 are again able to see the difference between the stopping criteria. For the CBSD10 dataset,
769 PnP-LBFGS¹ converges on all images in under 40 iterations, while PnP-LBFGS² sometimes
770 requires all 100 iterations, and the other PnP methods take anywhere from 100 to 10³ iterations
771 to converge. Figure 8 shows again that the convergence of the residuals is significantly faster
772 than the compared PnP methods per iteration. Note that for PnP-LBFGS, PnP-DRS and
773 PnP- $\hat{\alpha}$ PGD, we are allowed to choose larger values of the fidelity regularization term λ , leading
774 to better reconstructions in the low noise regime compared to PnP-PGD and PnP-DRSdiff.

775 As seen in Figure 8c, DPIR does not converge for super-resolution, and we observe an
776 oscillating behavior of the residuals and PSNR. In contrast, PnP-FISTA is able to converge
777 slightly faster than PnP-PGD, but does not converge for some images as seen by the decreasing
778 PSNR for one curve in Figure 7. Both PnP-FISTA and DPIR are able to perform reasonably
779 for higher noise levels of $\sigma = 12.75$, but have more divergence issues for lower noise levels,
780 leading to reduced performance as seen in Table 4. We again observe the gap in performance
781 between DPIR at iteration 10³ and at iteration 24 as suggested in the original DPIR work.
782 The performance gap between DPIR and provable PnP methods is less apparent for super-
783 resolution as opposed to deblurring, as observed in [32].

784 **4.5. Computational Complexity.** While each iteration of PnP-LBFGS has increased com-
785 plexity, we observed convergence in much fewer iterations. In this section, we outline the
786 computational requirements for the number of neural network N_σ evaluations, denoising steps
787 D_σ , as well as computations of ∇f and $\nabla^2 f$ required per iteration. Note that if a closed form
788 for $\nabla^2 f$ is intractable, computations of (3.5c) can be replaced with Hessian-vector products,
789 available in many deep learning libraries.

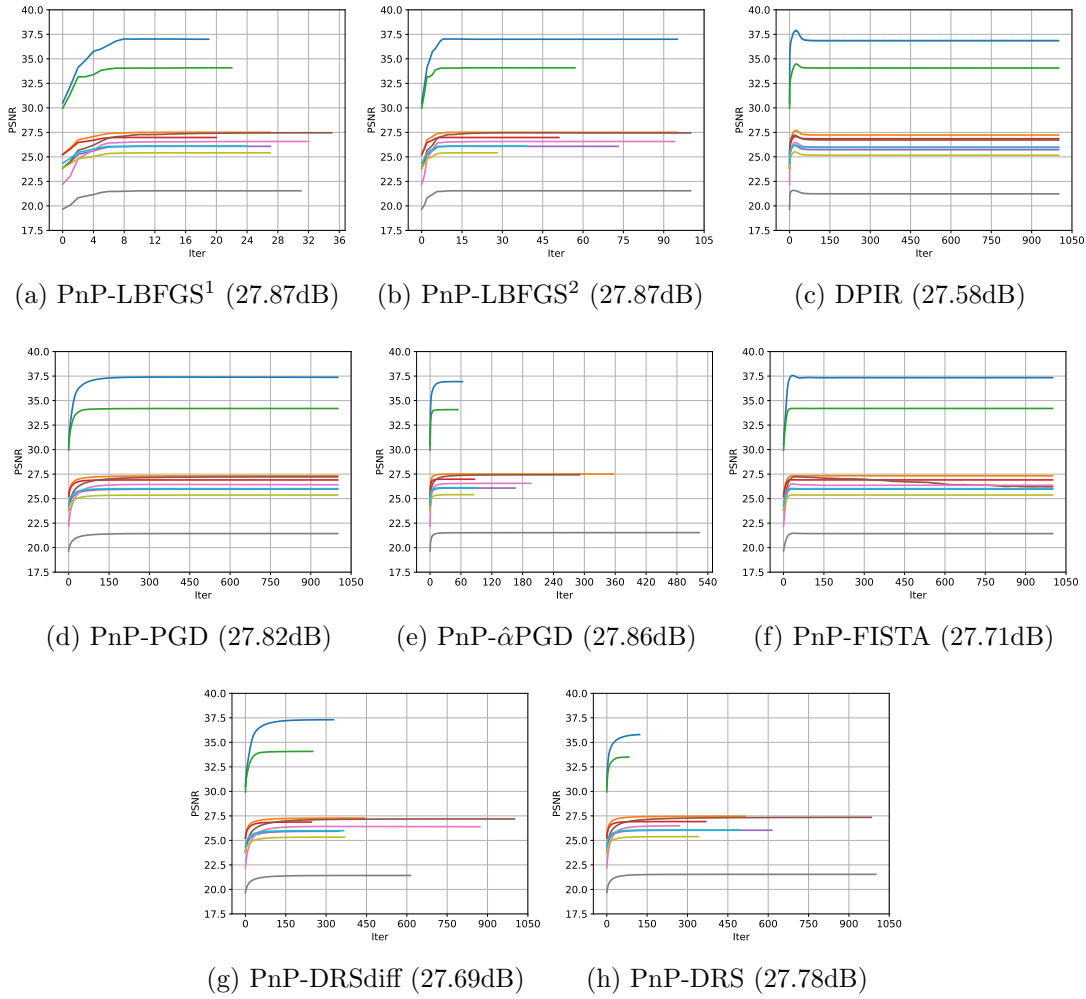


Figure 7: Convergence of the PSNR (dB) of the various curves for super-resolution, with the average dB in brackets. Each curve corresponds to one of the 10 images from the CBSD10 dataset, evaluated with the Gaussian blur kernel with standard deviation $\sigma_{\text{blur}} = 1.2$ and additive noise $\sigma = 7.65$, with scale $s_{sr} = 2$. We observe the convergence of PSNRs in under 40 iterations for PnP-LBFGS¹, much faster than the compared PnP methods.

790 We can calculate T_γ and R_γ together using one call each of ∇f and D_σ . From (3.5), φ_γ
 791 requires ∇f and g_σ , which in turn requires N_σ . $\nabla\varphi_\gamma$ has a closed form, which requires R_γ
 792 and an evaluation of $\nabla^2 f$.

793 Consider a single iteration of PnP-LBFGS. We first compute $\nabla\varphi_\gamma(x^k)$ and $\varphi_\gamma(x^k)$. Com-
 794 puting d^k using L-BFGS does not require any additional evaluations of D_σ , N_σ , ∇f or $\nabla^2 f$,
 795 as the secants and differences will have been computed in the previous iteration. For each test
 796 of w^k , we need to compute a single iteration of φ_γ , which takes one evaluation each of ∇f

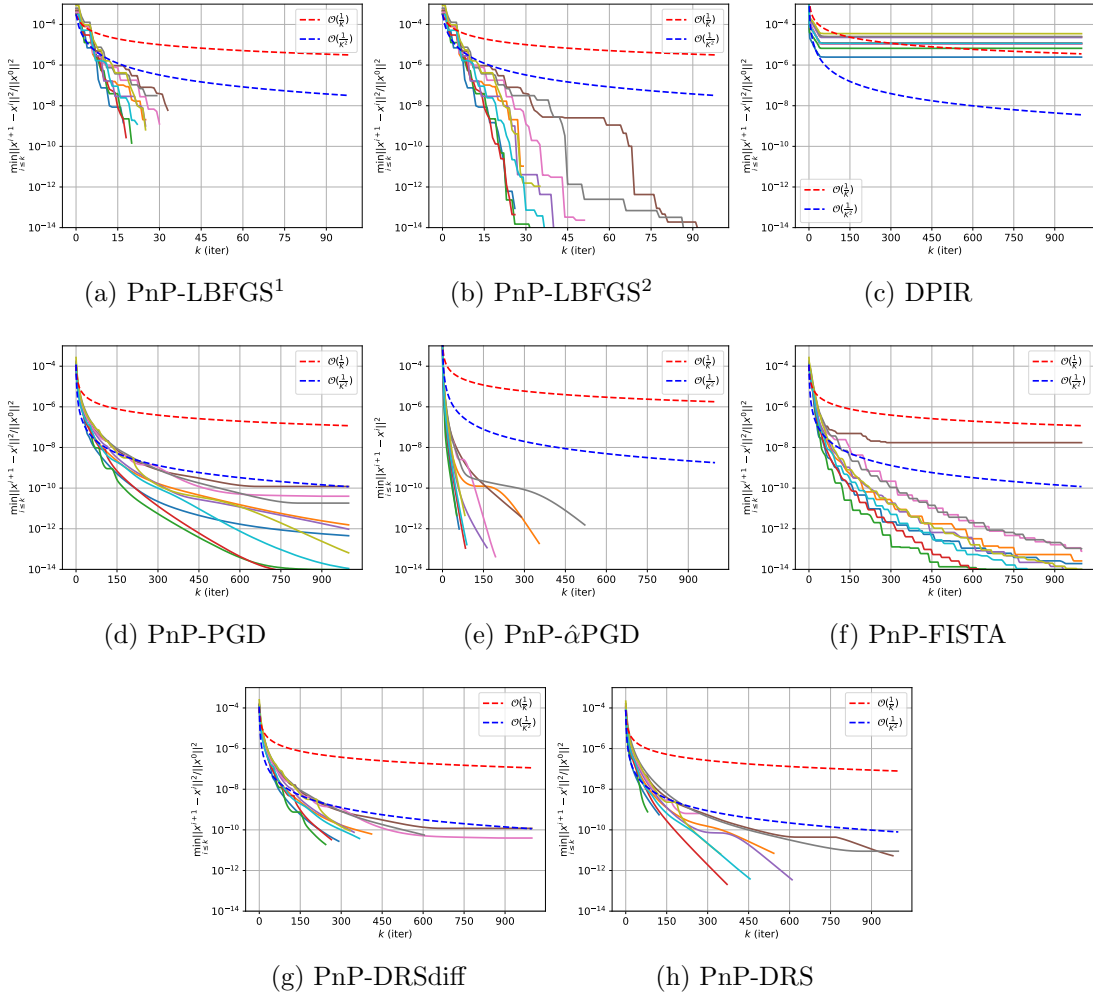


Figure 8: Convergence of the residuals $\min_{i \leq k} \|x^{i+1} - x^i\|^2 / \|x^0\|^2$ of the various methods for super-resolution. Each curve corresponds to one of the 10 images from the CBSD10 dataset, evaluated with the Gaussian blur kernel with standard deviation $\sigma_{\text{blur}} = 1.2$ and additive noise $\sigma/255 = 7.65$, with scale $s_{sr} = 2$. PnP-LBFGS² demonstrates significantly faster residual convergence of the proposed method.

797 and N_σ . Once a suitable w^k is found, we compute $T_\gamma(w^k)$ and $R_\gamma(w^k)$ together using the last
 798 stored $\nabla f(w^k)$, requiring only one additional D_σ operation. For the secant y^k , we require an
 799 evaluation of $\nabla \varphi_\gamma(w^k)$, which requires only one additional $\nabla^2 f$ evaluation. This concludes
 800 one iteration.

801 To evaluate the proposed stopping criteria for PnP-LBFGS¹, we are also required to
 802 compute $\varphi(x^{k+1})$ from (3.5d). Note we already have $g_\sigma(w^k - \gamma \nabla f(w^k))$ from computing
 803 $\varphi_\gamma(w^k)$, and $T_\gamma(w^k) = x^k$, hence we get $\varphi(x^{k+1})$ with no further evaluations needed.

804 In total, assuming we need T tests for τ_k , the per iteration-cost is

$$805 \quad (4.2) \quad \begin{pmatrix} \#N_\sigma \\ \#D_\sigma \\ \#\nabla f \\ \#\nabla^2 f \end{pmatrix}_{\text{PnP-LBFGS}} = \underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}}_{\substack{\nabla\varphi_\gamma(x^k), \\ \varphi_\gamma(x^k)}} + T \underbrace{\begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}}_{\text{test } w^k} + \underbrace{\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}}_{\substack{T_\gamma(w^k), \\ R_\gamma(w^k)}} + \underbrace{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_{\nabla\varphi_\gamma(w^k)} = \begin{pmatrix} T+1 \\ 2 \\ T+1 \\ 2 \end{pmatrix}.$$

806 At later iterations, the number of tests is only $T = 1$, since the step-size $\tau = 1$ is accepted
 807 almost always. Therefore, later iterations require two of $N_\sigma, D_\sigma, \nabla f$ and $\nabla^2 f$. For comparison,
 808 PnP-PGD requires one evaluation each of D_σ and ∇f , and the PnP-DRS methods require one
 809 evaluation each of D_σ and prox_f . Note that for these methods to test their stopping criteria
 810 by computing φ , they also require one evaluation of g_σ and hence of N_σ [33]. These methods
 811 thus have complexity

$$812 \quad \begin{pmatrix} \#N_\sigma \\ \#D_\sigma \\ \#\nabla f \end{pmatrix}_{\text{PnP-PGD}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} \#N_\sigma \\ \#D_\sigma \\ \#\text{prox}_f \end{pmatrix}_{\substack{\text{PnP-DRS;} \\ \text{PnP-DRSdiff}}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

813 To compute the asymptotic complexity of PnP-LBFGS, suppose the images have dimen-
 814 sion d , and that the denoisers have P parameters. From (4.2), we can read off the com-
 815 plexity of computing one iteration given d^k as $\mathcal{O}(d \times P \times T)$, with $\mathcal{O}(d)$ memory require-
 816 ment to hold the x^k, w^k and intermediate gradients. To compute d^k , the computational
 817 complexity of L-BFGS scales linearly with the input dimension and memory length m , and
 818 requires us to store m secants and differences. The asymptotic complexity per iteration is thus
 819 $\mathcal{O}(d \times P \times T + md)$, where the number of tests T is eventually always 1. The total memory
 820 requirement is $\mathcal{O}((m+1) \times d)$, where we store m differences and secants.

821 A similar complexity analysis can be applied to the PnP-PGD, PnP-DRSdiff and PnP-
 822 DRS methods to achieve a per-iteration computational complexity of $\mathcal{O}(d \times P)$ and mem-
 823 ory requirement of $\mathcal{O}(d)$. However, these three PnP methods do not come with improved
 824 convergence rates under additional smoothness assumptions, and come with residual conver-
 825 gence at a rate $\min_{i \leq k} \|x^{i+1} - x^i\| = \mathcal{O}(1/k)$. PnP-LBFGS achieves residual convergence
 826 $\min_{i \leq k} \|R_{\gamma_i}(x^i)\| = \mathcal{O}(1/k)$ from Theorem 2.11, as well as superlinear convergence under the
 827 assumptions of Theorem 2.14. This is summarized in Table 5.

828 The above complexity analysis shows that the main increase in computational burden for
 829 PnP-LBFGS is the requirement of two evaluations of $\nabla^2 f$ at each iteration, as well as at least
 830 double the number of neural network evaluations compared to the compared PnP methods.
 831 However, assuming only one test for w^k is needed, each iteration only requires one additional
 832 evaluation of the denoiser-related networks N_σ, D_σ and fidelity gradient ∇f (or prox_f) to the
 833 compared PnP methods. In our experiments, $\nabla^2 f$ has a low computational cost due to the
 834 closed form. This allows us to trade roughly 2–3× the per-iteration cost with nearly 10×
 835 fewer iterations required as shown in Figures 4 and 8, resulting in fewer total function calls,
 836 and thus the 4–5× faster reconstruction times as shown in Tables 3 and 4.

Table 5: Complexity to achieve an ϵ -optimal solution, in terms of the squared residual for PnP-PGD/DRS/DRSdiff, and in terms of the residual $R_{\gamma_i}(x^i)$ for PnP-LBFGS. Under the assumptions of Theorem 2.14 for superlinear convergence, the number of tests is eventually always $T = 1$, and we are able to achieve at least linear speedup.

Complexity	PnP-PGD/DRS/DRSdiff	PnP-LBFGS	PnP-LBFGS superlinear
Computation	$\mathcal{O}(dP\epsilon^{-1})$	$\mathcal{O}((dPT + md)\epsilon^{-1})$	$\mathcal{O}((dP + md) \log \epsilon)$
Memory	$\mathcal{O}(d)$	$\mathcal{O}((m + 1)d)$	$\mathcal{O}((m + 1)d)$

837 **5. Conclusion.** In this work, we propose a Plug-and-Play approach to image reconstruction that utilizes descent steps based on the forward-backward envelope. Using the descent
838 formulation, we are able to further incorporate quasi-Newton steps to accelerate convergence.
839 The resulting PnP scheme is provably convergent with a gradient-step assumption on the
840 denoiser by using the Kurdyka-Lojasiewicz property and theoretically achieves superlinear
841 convergence if a Hessian approximation satisfying the Dennis-Moré condition is available.
842 Moreover, properties of the forward-backward envelope allow for additional ways of checking
843 convergence. Our experiments demonstrate that it is able to converge significantly faster in
844 terms of both time and iteration count as well as having highly competitive performance when
845 compared with competing PnP methods with similar convergence guarantees.
846

847 For future works, one route is to consider alternative parameterizations of the denoiser
848 D_σ . For example, consider the objective $\varphi = f + \phi_\sigma$ and the task of learning the regularization
849 term ϕ_σ [49, 50]. By enforcing convexity of ϕ_σ through the neural network architecture, such
850 as using input-convex neural networks [1], (weakly-) convex ridge regularizers [25, 26], firm
851 nonexpansiveness [57], or parametric splines [53], results from [67] utilizing convexity such
852 as global sublinear convergence and local linear convergence can be applied. This may also
853 alleviate divergence problems caused when Lipschitz constraints on the denoisers are violated,
854 as sometimes arises using spectral regularization. One restriction of the proposed method lies
855 in the restriction of the regularization parameter, which imposes a bound on the minimum
856 amount of regularization. Future works could look to loosen this restriction, similarly to
857 [31]. In addition, while only simple forward operators such as image deblurring and super-
858 resolution are experimented on in this work, the accelerated convergence rate and model-based
859 interpretation may make this PnP scheme suitable for more complicated forward operators
860 such as CT ray transforms. Future works may explore these practical applications, with a
861 suitably trained “denoiser” for these domains.

862 **Acknowledgements.** Hong Ye Tan acknowledges support from GSK.ai and the Masa-
863 son Foundation. Professor Carola-Bibiane Schönlieb acknowledges support from the Philip
864 Leverhulme Prize, the Royal Society Wolfson Fellowship, the EPSRC advanced career fel-
865 lowship EP/V029428/1, EPSRC grants EP/S026045/1 and EP/T003553/1, EP/N014588/1,
866 EP/T017961/1, the Wellcome Innovator Awards 215733/Z/19/Z and 221633/Z/20/Z, the Eu-
867 ropean Union Horizon 2020 research and innovation programme under the Marie Skłodowska-
868 Curie grant agreement No. 777826 NoMADS, the Cantab Capital Institute for the Mathe-

869 matics of Information and the Alan Turing Institute.

870

REFERENCES

- 871 [1] B. AMOS, L. XU, AND J. Z. KOLTER, *Input convex neural networks*, in International Conference on
872 Machine Learning, PMLR, 2017, pp. 146–155.
- 873 [2] S. ARJOMAND BIGDELI, M. ZWICKER, P. FAVARO, AND M. JIN, *Deep mean-shift priors for image restora-*
874 *tion*, Advances in Neural Information Processing Systems, 30 (2017).
- 875 [3] L. ARMIJO, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific Journal
876 of mathematics, 16 (1966), pp. 1–3.
- 877 [4] H. ATTOUCH, J. BOLTE, P. REDONT, AND A. SOUBEYRAN, *Proximal alternating minimization and*
878 *projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality*,
879 Mathematics of operations research, 35 (2010), pp. 438–457.
- 880 [5] H. ATTOUCH, J. BOLTE, AND B. F. SVAITER, *Convergence of descent methods for semi-algebraic and*
881 *tame problems: proximal algorithms, forward-backward splitting, and regularized gauss-seidel meth-*
882 *ods*, Mathematical Programming, 137 (2013), pp. 91–129.
- 883 [6] J.-F. AUJOL, C. DOSSAL, AND A. RONDEPIERRE, *Fista is an automatic geometrically optimized algorithm*
884 *for strongly convex functions*, Mathematical Programming, (2023), pp. 1–43.
- 885 [7] H. BAUSCHKE AND P. COMBETTES, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*,
886 CMS Books in Mathematics, Springer New York, 2011.
- 887 [8] A. BECK, *First-order methods in optimization*, SIAM, 2017.
- 888 [9] A. BECK AND M. TEBoulLE, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*,
889 SIAM journal on imaging sciences, 2 (2009), pp. 183–202.
- 890 [10] J. BOLTE, S. SABACH, AND M. TEBoulLE, *Proximal alternating linearized minimization for nonconvex*
891 *and nonsmooth problems*, Mathematical Programming, 146 (2014), pp. 459–494.
- 892 [11] A. BUADES, B. COLL, AND J.-M. MOREL, *Non-local means denoising*, Image Processing On Line, 1
893 (2011), pp. 208–212.
- 894 [12] G. T. BUZZARD, S. H. CHAN, S. SREEHARI, AND C. A. BOUMAN, *Plug-and-play unplugged: Optimization-*
895 *free reconstruction using consensus equilibrium*, SIAM Journal on Imaging Sciences, 11 (2018),
896 pp. 2001–2020.
- 897 [13] R. H. BYRD, S. L. HANSEN, J. NOCEDAL, AND Y. SINGER, *A stochastic quasi-Newton method for large-*
898 *scale optimization*, SIAM Journal on Optimization, 26 (2016), pp. 1008–1031.
- 899 [14] A. CHAMBOLLE AND C. DOSSAL, *On the convergence of the iterates of the “fast iterative shrink-*
900 *age/thresholding algorithm”*, Journal of Optimization theory and Applications, 166 (2015), pp. 968–
901 982.
- 902 [15] S. H. CHAN, X. WANG, AND O. A. ELGENDY, *Plug-and-play admm for image restoration: Fixed-point*
903 *convergence and applications*, IEEE Transactions on Computational Imaging, 3 (2016), pp. 84–98.
- 904 [16] R. COHEN, Y. BLAU, D. FREEDMAN, AND E. RIVLIN, *It has potential: Gradient-driven denoisers for*
905 *convergent solutions to inverse problems*, Advances in Neural Information Processing Systems, 34
906 (2021), pp. 18152–18164.
- 907 [17] R. COHEN, M. ELAD, AND P. MILANFAR, *Regularization by denoising via fixed-point projection (red-pro)*,
908 SIAM Journal on Imaging Sciences, 14 (2021), pp. 1374–1406.
- 909 [18] K. DABOV, A. FOI, V. KATKOVNIK, AND K. EGAZARIAN, *Image denoising by sparse 3-d transform-*
910 *domain collaborative filtering*, IEEE Transactions on image processing, 16 (2007), pp. 2080–2095.
- 911 [19] I. DAUBECHIES, M. DEFRISE, AND C. DE MOL, *An iterative thresholding algorithm for linear inverse*
912 *problems with a sparsity constraint*, Communications on Pure and Applied Mathematics: A Journal
913 Issued by the Courant Institute of Mathematical Sciences, 57 (2004), pp. 1413–1457.
- 914 [20] J. E. DENNIS AND J. J. MORÉ, *A characterization of superlinear convergence and its application to*
915 *quasi-Newton methods*, Mathematics of computation, 28 (1974), pp. 549–560.
- 916 [21] J. DOUGLAS AND H. H. RACHFORD, *On the numerical solution of heat conduction problems in two and*
917 *three space variables*, Transactions of the American mathematical Society, 82 (1956), pp. 421–439.
- 918 [22] M. FAZLYAB, A. ROBey, H. HASSANI, M. MORARI, AND G. PAPPAS, *Efficient and accurate estimation*
919 *of Lipschitz constants for deep neural networks*, Advances in Neural Information Processing Systems,

- 920 32 (2019).
- 921 [23] W. GAN, S. SHOUSHARI, Y. HU, J. LIU, H. AN, AND U. S. KAMILOV, *Block coordinate plug-and-play*
922 *methods for blind inverse problems*, arXiv preprint arXiv:2305.12672, (2023).
- 923 [24] T. GOLDSTEIN, C. STUDER, AND R. BARANIUK, *A field guide to forward-backward splitting with a fast*
924 *implementation*, arXiv preprint arXiv:1411.3406, (2014).
- 925 [25] A. GOUJON, S. NEUMAYER, P. BOHRA, S. DUCOTTERD, AND M. UNSER, *A neural-network-based convex*
926 *regularizer for image reconstruction*, arXiv preprint arXiv:2211.12461, (2022).
- 927 [26] A. GOUJON, S. NEUMAYER, AND M. UNSER, *Learning weakly convex regularizers for convergent image-*
928 *reconstruction algorithms*, arXiv preprint arXiv:2308.10542, (2023).
- 929 [27] R. GRIBONVAL AND M. NIKOLOVA, *A characterization of proximity operators*, Journal of Mathematical
930 Imaging and Vision, 62 (2020), pp. 773–789.
- 931 [28] S. HELGASON AND S. HELGASON, *The radon transform*, vol. 2, Springer, 1980.
- 932 [29] J. HERTRICH, S. NEUMAYER, AND G. STEIDL, *Convolutional proximal neural networks and plug-and-play*
933 *algorithms*, Linear Algebra and its Applications, 631 (2021), pp. 203–234.
- 934 [30] W. HUANG, K. A. GALLIVAN, AND P.-A. ABSIL, *A Broyden class of quasi-Newton methods for Riemann-*
935 *ian optimization*, SIAM Journal on Optimization, 25 (2015), pp. 1660–1685.
- 936 [31] S. HURAUULT, A. CHAMBOLLE, A. LECLAIRE, AND N. PAPADAKIS, *A relaxed proximal gradient descent*
937 *algorithm for convergent plug-and-play with proximal denoiser*, 2023.
- 938 [32] S. HURAUULT, A. LECLAIRE, AND N. PAPADAKIS, *Gradient step denoiser for convergent plug-and-play*,
939 arXiv preprint arXiv:2110.03220, (2021).
- 940 [33] S. HURAUULT, A. LECLAIRE, AND N. PAPADAKIS, *Proximal denoiser for convergent plug-and-play opti-*
941 *mization with nonconvex regularization*, 2022.
- 942 [34] Q. JIN, A. KOPPEL, K. RAJAWAT, AND A. MOKHTARI, *Sharpened quasi-Newton methods: Faster superlin-*
943 *ear rate and larger local convergence neighborhood*, in International Conference on Machine Learning,
944 PMLR, 2022, pp. 10228–10250.
- 945 [35] Q. JIN AND A. MOKHTARI, *Non-asymptotic superlinear convergence of standard quasi-Newton methods*,
946 Mathematical Programming, 200 (2023), pp. 425–473.
- 947 [36] J. KAIPIO AND E. SOMERSALO, *Statistical and computational inverse problems*, vol. 160, Springer Science
948 & Business Media, 2006.
- 949 [37] U. S. KAMILOV, C. A. BOUMAN, G. T. BUZZARD, AND B. WOHLBERG, *Plug-and-play methods for inte-*
950 *grating physical and learned models in computational imaging: Theory, algorithms, and applications*,
951 IEEE Signal Processing Magazine, 40 (2023), pp. 85–97.
- 952 [38] U. S. KAMILOV, H. MANSOUR, AND B. WOHLBERG, *A plug-and-play priors approach for solving nonlinear*
953 *imaging inverse problems*, IEEE Signal Processing Letters, 24 (2017), pp. 1872–1876.
- 954 [39] J. D. LEE, Y. SUN, AND M. A. SAUNDERS, *Proximal Newton-type methods for minimizing composite*
955 *functions*, SIAM Journal on Optimization, 24 (2014), pp. 1420–1443.
- 956 [40] A. LEVIN, Y. WEISS, F. DURAND, AND W. T. FREEMAN, *Understanding and evaluating blind deconv-*
957 *olution algorithms*, in 2009 IEEE conference on computer vision and pattern recognition, IEEE, 2009,
958 pp. 1964–1971.
- 959 [41] D.-H. LI AND M. FUKUSHIMA, *A modified BFGS method and its global convergence in nonconvex mini-*
960 *mization*, Journal of Computational and Applied Mathematics, 129 (2001), pp. 15–35.
- 961 [42] D.-H. LI AND M. FUKUSHIMA, *On the global convergence of the BFGS method for nonconvex unconstrained*
962 *optimization problems*, SIAM Journal on Optimization, 11 (2001), pp. 1054–1064.
- 963 [43] D. C. LIU AND J. NOCEDAL, *On the limited memory BFGS method for large scale optimization*, Mathe-
964 matical programming, 45 (1989), pp. 503–528.
- 965 [44] S. LUNZ, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Adversarial regularizers in inverse problems*, Advances in
966 neural information processing systems, 31 (2018).
- 967 [45] D. MARTIN, C. FOWLKES, D. TAL, AND J. MALIK, *A database of human segmented natural images and its*
968 *application to evaluating segmentation algorithms and measuring ecological statistics*, in Proceedings
969 Eighth IEEE International Conference on Computer Vision. ICCV 2001, vol. 2, IEEE, 2001, pp. 416–
970 423.
- 971 [46] T. MIYATO, T. KATAOKA, M. KOYAMA, AND Y. YOSHIDA, *Spectral normalization for generative adver-*
972 *sarial networks*, arXiv preprint arXiv:1802.05957, (2018).
- 973 [47] A. MOKHTARI AND A. RIBEIRO, *Global convergence of online limited memory BFGS*, The Journal of

- 974 Machine Learning Research, 16 (2015), pp. 3151–3181.
- 975 [48] J.-J. MOREAU, *Proximité et dualité dans un espace hilbertien*, Bulletin de la Société mathématique de
976 France, 93 (1965), pp. 273–299.
- 977 [49] S. MUKHERJEE, S. DITTMER, Z. SHUMAYLOV, S. LUNZ, O. ÖKTEM, AND C.-B. SCHÖNLIEB, *Learned*
978 *convex regularizers for inverse problems*, 2020.
- 979 [50] S. MUKHERJEE, A. HAUPTMANN, O. ÖKTEM, M. PEREYRA, AND C.-B. SCHÖNLIEB, *Learned reconstruc-*
980 *tion methods with convergence guarantees*, arXiv preprint arXiv:2206.05431, (2022).
- 981 [51] P. NAIR, R. G. GAVASKAR, AND K. N. CHAUDHURY, *Fixed-point and objective convergence of plug-and-*
982 *play algorithms*, IEEE Transactions on Computational Imaging, 7 (2021), pp. 337–348.
- 983 [52] S. NEUMAYER, A. GOUJON, P. BOHRA, AND M. UNSER, *Approximation of Lipschitz functions using deep*
984 *spline neural networks*, SIAM Journal on Mathematics of Data Science, 5 (2023), pp. 306–322.
- 985 [53] H. Q. NGUYEN, E. BOSTAN, AND M. UNSER, *Learning convex regularizers for optimal Bayesian denoising*,
986 IEEE Transactions on Signal Processing, 66 (2017), pp. 1093–1105.
- 987 [54] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, NY, USA, 2e ed., 2006.
- 988 [55] P. OCHS AND T. POCK, *Adaptive fista for nonconvex optimization*, SIAM Journal on Optimization, 29
989 (2019), pp. 2482–2503.
- 990 [56] A. PASZKE, S. GROSS, F. MASSA, A. LERER, J. BRADBURY, G. CHANAN, T. KILLEEN, Z. LIN,
991 N. GIMELSHEIN, L. ANTIGA, A. DESMAISON, A. KOPF, E. YANG, Z. DEVITO, M. RAISON,
992 A. TEJANI, S. CHILAMKURTHY, B. STEINER, L. FANG, J. BAI, AND S. CHINTALA, *Pytorch: An*
993 *imperative style, high-performance deep learning library*, in Advances in Neural Information Process-
994 ing Systems 32, H. Wallach, H. Laroche, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and
995 R. Garnett, eds., Curran Associates, Inc., 2019, pp. 8024–8035, [http://papers.neurips.cc/paper/](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf)
996 [9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf](http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf).
- 997 [57] J.-C. PESQUET, A. REPETTI, M. TERRIS, AND Y. WIAUX, *Learning maximally monotone operators for*
998 *image recovery*, SIAM Journal on Imaging Sciences, 14 (2021), pp. 1206–1237.
- 999 [58] E. T. REEHORST AND P. SCHNITER, *Regularization by denoising: Clarifications and new interpretations*,
1000 IEEE transactions on computational imaging, 5 (2018), pp. 52–67.
- 1001 [59] R. ROCKAFELLAR, *Convex Analysis*, Princeton mathematical series ; 28, Princeton University Press,
1002 Princeton, NJ, 1972.
- 1003 [60] A. RODOMANOV AND Y. NESTEROV, *Greedy quasi-Newton methods with explicit superlinear convergence*,
1004 SIAM Journal on Optimization, 31 (2021), pp. 785–811.
- 1005 [61] A. RODOMANOV AND Y. NESTEROV, *Rates of superlinear convergence for classical quasi-Newton methods*,
1006 Mathematical Programming, (2021), pp. 1–32.
- 1007 [62] D. L. RUDERMAN, *The statistics of natural images*, Network: Computation in Neural Systems, 5 (1994),
1008 pp. 517–548.
- 1009 [63] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica
1010 D: nonlinear phenomena, 60 (1992), pp. 259–268.
- 1011 [64] E. RYU, J. LIU, S. WANG, X. CHEN, Z. WANG, AND W. YIN, *Plug-and-play methods provably converge*
1012 *with properly trained denoisers*, in International Conference on Machine Learning, PMLR, 2019,
1013 pp. 5546–5557.
- 1014 [65] N. N. SCHRAUDOLPH, J. YU, AND S. GÜNTER, *A stochastic quasi-Newton method for online convex*
1015 *optimization*, in Artificial intelligence and statistics, PMLR, 2007, pp. 436–443.
- 1016 [66] S. SREEHARI, S. V. VENKATAKRISHNAN, B. WOHLBERG, G. T. BUZZARD, L. F. DRUMMY, J. P. SIM-
1017 MONS, AND C. A. BOUMAN, *Plug-and-play priors for bright field electron tomography and sparse*
1018 *interpolation*, IEEE Transactions on Computational Imaging, 2 (2016), pp. 408–423.
- 1019 [67] L. STELLA, A. THEMELIS, AND P. PATRINOS, *Forward-backward quasi-Newton methods for nonsmooth*
1020 *optimization problems*, Computational Optimization and Applications, 67 (2017), pp. 443–487.
- 1021 [68] Y. SUN, J. LIU, Y. SUN, B. WOHLBERG, AND U. S. KAMILOV, *Async-red: A provably conver-*
1022 *gent asynchronous block parallel stochastic method using deep denoising priors*, arXiv preprint
1023 arXiv:2010.01446, (2020).
- 1024 [69] Y. SUN, B. WOHLBERG, AND U. S. KAMILOV, *An online plug-and-play algorithm for regularized image*
1025 *reconstruction*, IEEE Transactions on Computational Imaging, 5 (2019), pp. 395–408.
- 1026 [70] Y. SUN, Z. WU, X. XU, B. WOHLBERG, AND U. S. KAMILOV, *Scalable plug-and-play admm with con-*
1027 *vergence guarantees*, IEEE Transactions on Computational Imaging, 7 (2021), pp. 849–863.

- 1028 [71] J. TANG, *Accelerating plug-and-play image reconstruction via multi-stage sketched gradients*, arXiv pre-
1029 print arXiv:2203.07308, (2022).
- 1030 [72] E. TJOA AND C. GUAN, *A survey on explainable artificial intelligence (xai): Toward medical xai*, IEEE
1031 transactions on neural networks and learning systems, 32 (2020), pp. 4793–4813.
- 1032 [73] A. VELLIDO, *The importance of interpretability and visualization in machine learning for applications in*
1033 *medicine and health care*, Neural computing and applications, 32 (2020), pp. 18069–18083.
- 1034 [74] S. V. VENKATAKRISHNAN, C. A. BOUMAN, AND B. WOHLBERG, *Plug-and-play priors for model based*
1035 *reconstruction*, in 2013 IEEE Global Conference on Signal and Information Processing, IEEE, 2013,
1036 pp. 945–948.
- 1037 [75] X. WANG, S. MA, D. GOLDFARB, AND W. LIU, *Stochastic quasi-Newton methods for nonconvex stochastic*
1038 *optimization*, SIAM Journal on Optimization, 27 (2017), pp. 927–956.
- 1039 [76] F. WEN, L. CHU, P. LIU, AND R. C. QIU, *A survey on nonconvex regularization-based sparse and low-*
1040 *rank recovery in signal processing, statistics, and machine learning*, IEEE Access, 6 (2018), pp. 69883–
1041 69906.
- 1042 [77] K. ZHANG, Y. LI, W. ZUO, L. ZHANG, L. VAN GOOL, AND R. TIMOFTE, *Plug-and-play image restoration*
1043 *with deep denoiser prior*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021),
1044 pp. 6360–6376.
- 1045 [78] K. ZHANG, W. ZUO, S. GU, AND L. ZHANG, *Learning deep cnn denoiser prior for image restoration*, in
1046 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3929–3938.