On Maximum-a-Posteriori estimation with Plug & Play priors and stochastic gradient descent

Rémi Laumont, Valentin de Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, Marcelo Pereyra

jmiv manuscript No.
(will be inserted by the editor)

On Maximum-a-Posteriori estimation with Plug & Play priors and stochastic gradient descent

Rémi Laumont · Valentin De Bortoli ⋆ · Andrés Almansa · Julie Delon · Alain Durmus · Marcelo Pereyra

Received: date / Accepted: date

Abstract Bayesian methods to solve imaging inverse problems usually combine an explicit data likelihood function with a prior distribution that explicitly models expected properties of the solution. Many kinds of priors have been explored in the literature, from simple ones expressing local properties to more involved ones exploiting image redundancy at a non-local scale. In a departure from explicit modelling, several recent works have proposed and studied the use of implicit priors defined by an image denoising algorithm. This approach, commonly known as Plug & Play (PnP) regularisation, can deliver remarkably accurate results, particularly when combined with state-of-the-art denoisers based on convolutional neural networks. However, the theoretical analysis of PnP Bayesian models and algorithms is difficult and works on the topic often rely on unrealistic assumptions on the properties of the image denoiser. This papers studies maximum-a-posteriori (MAP) estimation for Bayesian models with PnP priors. We first consider questions related to existence, stability and well-posedness, and then present a convergence proof for MAP computation by PnP stochastic gradient descent (PnP-SGD) under realistic assumptions on the denoiser used. We report a range of imaging experiments demonstrating PnP-SGD as well as comparisons with other PnP schemes.

Keywords Plug & Play · Bayesian imaging · Stochastic Gradient Descent · Inverse Problems · Deblurring · Inpainting · Denoising

Mathematics Subject Classification (2020) 65K10 · 65K05 · 62F15 · 62C10 · 68Q25 · 68U10 · 90C26

⋆ Corresponding author

R. Laumont, A. Almansa and J. Delon
Université Paris Cité, CNRS, MAP5 UMR 8145, F-75006 Paris, France

V. De Bortoli
Department of Statistics, University of Oxford, 24-29 St Giles OX1 3LB, Oxford U.K.
E-mail: valentin.debortoli@gmail.com

A. Durmus
Centre Borelli, UMR 9010, École Normale Supérieure Paris-Saclay, France.

Marcelo Pereyra
Heriot-Watt University & Maxwell Institute for Mathematical Sciences, Edinburgh, U.K.

1 Introduction

Many inverse problems in imaging sciences consider the estimation of an unknown image $x \in \mathbb{R}^d$ from an observation y, related to x as follows

$$y = \mathbf{A}(x) + n \,, \tag{1}$$

where $\mathbf{A}$ is an observation operator and n is additive noise. Equation (1) is commonly referred to as the forward model.

Recovering x from the observation y by inverting the observation model is usually an ill-posed or ill-conditioned problem, in the sense that the the solution is not unique, or it is not stable w.r.t. perturbations of the observation y. To reduce estimation uncertainty and provide meaningful solutions it is necessary to use additional information about the unknown image x so that the estimation problem becomes well-posed[1].

The Bayesian statistical framework provides a powerful paradigm to formulate well-posed solutions to such imaging inverse problems. In this framework, the likelihood of the observation y given the unknown x is described by a statistical model with probability density function $p(y|x)$, and assumptions on the unknown x take the form of a marginal or prior density $p(x)$. These densities are usually specified explicitly, either directly or via their potentials $U(x) = -\log p(x)$ and $F(x, y) = -\log p(y|x)$. Observed and prior information are then combined by using Bayes' theorem to derive the posterior distribution of x given y, with probability density function given by

$$p(x|y) = \frac{p(y|x)p(x)}{\int p(y|\tilde{x})p(\tilde{x})\mathrm{d}\tilde{x}} \,. \tag{2}$$

This model underpins our inferences about $x|y$ and provides the basis for deriving Bayesian estimates. While different Bayesian estimators can then be considered, the Bayesian imaging literature predominantly relies on the maximum-a-posteriori (MAP) estimator

$$\hat{x}_{\mathrm{MAP}} = \underset{x\in\mathbb{R}^d}{\arg\max}\, p(x|y) = \underset{x\in\mathbb{R}^d}{\arg\min}\, \{F(x, y) + U(x)\} \,, \tag{3}$$

which is usually computationally cheaper than other estimators that require computing expectations w.r.t $x|y$, such as the the Minimum Mean Square Error estimator (MMSE) $\hat{x}_{\mathrm{MMSE}} = \mathbb{E}[x|y] = \int_{\mathbb{R}^d} \tilde{x}p(\tilde{x}|y)\mathrm{d}\tilde{x}$.

Until recently, most approaches in Bayesian imaging relied on explicit priors such as Markov random fields [1] (with the fields based on the total-variation pseudo-norm and its approximations being particularly prominent examples [59, 14, 42]), priors expressing sparsity in a transformed domain [26], or learning-based priors like patch-based Gaussian mixture models [75, 71, 67]. Among these priors, log-concave models have been particularly favored for both computational and analytical reasons. With regards to computation, log-concavity leads to formulations for MAP estimation and uncertainty quantification which benefit from the full arsenal of convex optimization tools, scaling efficiently to high-dimensions, with strong and well-known convergence guarantees [15, 50, 12, 5, 51, 56]. Similarly, log-concavity also enables the use of state-of-the-art Monte Carlo sampling algorithms (see, e.g., [27, 53]). Moreover, from an

[1] A problem is said to be well-posed in the sense of Hadamard when a solution exists, is unique, and depends in a Lipschitz continuous manner w.r.t. the observed data y.

analytical viewpoint, log-concavity guarantees the well-posedness of $p(x|y)$, and that $\hat{x}_{\text{MAP}}$ is formally a Bayesian estimator (as opposed to simply being the point with greatest density w.r.t. the Lebesgue measure, which is a significantly weaker result, see [52] for details).

Computation of the MAP solution. When the posterior density $p(.|y)$ is proper and differentiable, with $\nabla \log p(.|y)$ Lipschitz continuous, it is possible to use first-order optimisation methods to compute maximisers of $p(.|y)$, *i.e.* MAP estimators. The simplest first order optimisation scheme to compute $\hat{x}_{\text{MAP}}$ is arguably the gradient descent algorithm, given by an initial state $X_0 \in \mathbb{R}^d$ and the following recursion for all $k \in \mathbb{N}$

$$X_{k+1} = X_k - \delta_k \nabla F(X_k, y) - \delta_k \nabla U(X_k) \ , \tag{4}$$

where $(\delta_k)_{k\in\mathbb{N}} \in (\mathbb{R}_+)^{\mathbb{N}}$ is a sequence of step-sizes. The sequence $(X_k)_{k\in\mathbb{N}}$ converges to critical points of $p(.|y)$ under mild assumptions on the sequence $(\delta_k)_{k\in\mathbb{N}}$ [49] and $p(.|y)$. Alternatively, the stochastic gradient descent (SGD) variant

$$X_{k+1} = X_k - \delta_k \nabla F(X_k, y) - \delta_k \nabla U(X_k) + \delta_k Z_{k+1} \ , \tag{5}$$

where $\{Z_k \ : \ k \in \mathbb{N}\}$ is a family of i.i.d Gaussian random variables with zero mean and identity covariance matrix, is more robust to local minima and saddle points and hence more suitable when $x \mapsto p(x|y)$ is not log-concave on $\mathbb{R}^d$ [9,7]. Of course, there are many optimisation schemes with better convergence properties than SGD (see, e.g., [49] in the convex case), as well as other dynamics to construct efficient optimisation algorithms [37,76]. Nevertheless, SGD is straightforward to apply, robust, and has a detailed convergence theory, making it a valuable algorithm in the imaging scientist's toolbox.

Agnostic Deep learning approaches. In the last few years, deep neural networks have become ubiquitous to solve inverse problems in imaging, showing unmatched performances for point estimation for some specific problems like image denoising. Deep networks can be trained end-to-end as agnostic regressors, *i.e.* without explicitly using the knowledge of the forward model (1) [25,72,74,30,61,29] or on the contrary can use this model explicitly via unrolled optimization techniques [32,17,24,31]. One disadvantage of using neural networks as agnostic regressors to solve imaging inverse problems is that in order to achieve state-of-the-art performance it is usually necessary to train the network for a specific problem configuration - the network must be retrained if the forward model or any model parameters change significantly. Even some model-based and unrolling approaches that are not completely agnostic may require retraining to be used in cases where the degradation model is very different from the models considered during training [21]. Also, imaging approaches based on neural networks struggle to support more advanced inferences by comparison to a Bayesian treatment by Monte Carlo sampling, which can support a wide breadth of statistical analyses beyond point estimation, particularly Bayesian decision-theoretic approaches to deal with advanced forms of uncertainty quantification (e.g., hypothesis tests, p-values, model misspecification tests), as well as approaches to deal with automatic calibration of partially unknown models and objective model comparison [57].

Plug & Play (PnP) approaches. PnP approaches strike a balance between an explicit and fully modular modelling paradigm that represents the likelihood $p(y|x)$ and the prior $p(x)$ explicitly (or the data-fidelity and regularisation terms in a variational formulation), and a purely data-driven approach that seeks to infer the model from data. More precisely, these methods usually combine an explicit likelihood density or data-fidelity term with a prior or regularisation term that is implicitly defined by an image denoising algorithm D_ε [3]. This construction takes place at the algorithmic level, as opposed to at an explicit modelling level, by using the denoiser D_ε in lieu of the gradient ∇U (also called score in the literature) or the proximal operator prox_U within an iterative optimisation scheme to compute $\hat{x}_{\mathrm{MAP}}$ [43,73,16,36,60]. Unlike the end-to-end deep learning approaches mentioned in the previous paragraph, this strategy benefits from the modularity of the Bayesian and the variational approaches, which decouple the regularisation model and data observation model, while still taking advantage of neural networks. Indeed the regularisation can be represented by a neural network denoiser D_ε that was learnt from training data. This strategy has been shown to deliver remarkably accurate results for a wide range of inverse problems, particularly when D_ε is chosen carefully. The question of the convergence of these PnP algorithms has been the focus of several papers in the recent years [60, 70,64], but it has not been satisfactorily answered yet, especially with regards to the strong assumptions made on D_ε, on F, and on the algorithm parameters. All of these limitations will be detailed in Section 2.

Lastly, many other fundamental questions related to inference with PnP schemes remain largely unexplored, particularly in the context of the Bayesian paradigm. For example, questions related to the correct definition of the Bayesian models, to the existence and well-posedness of the estimators that the PnP scheme seeks to compute, and whether these are proper Bayesian estimators in the sense of decision theory.

Contributions. The aim of this paper is to significantly improve our theoretical understanding of MAP estimation with *Plug & Play* priors. We first study some fundamental questions related to the posterior density that are essential for meaningful MAP estimation. We establish easily verifiable conditions such that the PnP posterior density is proper, well-posed, Lipschitz continously differentiable, and stable w.r.t. the parameter ε defining the strengh of the denoiser D_ε. Following on from this, we investigate in detail the convergence of the Plug-and-Play Stochastic Gradient Descent (PnP-SGD) for MAP estimation. This iterative algorithm takes the following form: for $X_0 \in \mathbb{R}^d$ and any $k \in \mathbb{N}$

$$X_{k+1} = X_k - \delta_k \nabla F(X_k, y) - (\delta_k/\varepsilon)(X_k - D_\varepsilon(X_k)) + \delta_k Z_{k+1} \;, \qquad \text{(PnP-SGD)}$$

where $\{Z_k \;:\; k \in \mathbb{N}\}$ is a family of independent Gaussian random variables with zero mean and identity covariance matrix, $D_\varepsilon : \mathbb{R}^d \to \mathbb{R}^d$ is a denoiser operator and $(\delta_k)_{k\in\mathbb{N}}$ is a sequence of step-sizes. We establish convergence results for PnP-SGD under mild and realistic assumptions on D_ε that hold for several well-known denoisers, including state-of-the-art denoisers based on convolutional neural networks. We also provide extensive experiments on several canonical inverse problems, implementing PnP-SGD with the denoising neural network presented in [60], which satisfies our convergence guarantees. Using the same denoising network, we also implement other state-of-the-art Plug-and-Play methods, and show that all schemes provide close results (in terms of image quality) when they converge. Although PnP-SGD is often

slower than schemes using D_ε to approximate a proximal operator, the conditions of convergence provided in this paper for PnP-SGD are less restrictive than those provided in the literature for other PnP schemes, and convergence is possible for any regularization parameter balancing the weights of the data and prior terms.

The paper is organized as follows. Section 2 presents an overview of previous works on *Plug & Play* approaches for MAP estimation. Following on from this, Section 3 describes the proposed theoretical framework for analysing MAP estimation with PnP priors, as well as a detailed convergence theory for MAP computation by PnP-SGD. Section 4 illustrates the behavior of PnP-SGD, along with other PnP schemes, on several classical imaging problems.

2 A survey of Plug & Play methods in imaging

In the context of imaging inverse problems, *Plug & Play* methods aim at using a carefully chosen denoiser $D_\varepsilon : \mathbb{R}^d \to \mathbb{R}^d$ to implicitly define an image prior. This is achieved by relating D_ε to a proximal operator or a gradient associated with the prior density. In the first case, D_ε replaces a MAP estimator for a denoising problem. In the second case, D_ε replaces a Minimum Mean Square Error (MMSE) estimator for a denoising problem, related to the gradient of a log-prior via Tweedie's identity [2][28].

In what follows, we describe how these approaches have been widely used to compute solutions to inverse problems. In our discussion, we pay particular attention to questions related to algorithmic convergence, and to the interpretation of the computed solutions, as this has been an important focus of the literature.

2.1 Plug & Play MAP estimators using proximal splitting

Let $D_\varepsilon^\dagger$ denote the MAP estimator to recover x from a noisy observation $x_\varepsilon \sim \mathcal{N}(x, \varepsilon\,\mathrm{Id})$ under the assumption that x has marginal density $p(x) \propto \exp[-U(x)]$; that is, $D_\varepsilon^\dagger(x_\varepsilon) = \arg\min_{x\in\mathbb{R}^d}\{\frac{1}{2}\|x_\varepsilon - x\|^2 + \varepsilon U(x)\} = \mathrm{prox}_{\varepsilon U}(x_\varepsilon)$. When we set the PnP denoiser D_ε such that $D_\varepsilon = D_\varepsilon^\dagger$, any optimization scheme making use of a proximal descent on the prior can be used to solve (3) via D_ε.

For instance, the alternating direction method of multipliers (ADMM) [8] writes the augmented Lagrangian of (3) as

$$E_\varepsilon(x, z, v) = F(x, y) + \|x - z\|^2/(2\varepsilon) + v^\top(x - z) + U(z)\ .$$

The joint optimization of the augmented Lagrangian is given by

$$(\hat{x}_{\mathrm{MAP}}, \hat{z}_{\mathrm{MAP}}) = \arg\min_{x,z\in\mathbb{R}^d} \max_{v\in\mathbb{R}^d} E_\varepsilon(x, z, v).$$

[2] Notice that although it is conceptually helpful to distinguish these two cases (in order to make a historical and practical survey of the subject), there are clear theoretical connections between the two approaches. Indeed, under regularity conditions on the Bayesian modelinvolved, MAP denoisers can be expressed as MMSE denoisers under an alternative (albeit often unknown) Bayesian model [33]. However this equivalence can not always be exploited in practice and has been mostly ignored in the literature on *Plug & Play* methods until very recently with the work of Xu *et al.* [70] to be presented later.

This provides the solution $\hat{x}_{\text{MAP}} = \hat{z}_{\text{MAP}}$ of (3) when $\varepsilon \to 0$. In practice, the joint optimization is solved by an alternate minimization scheme on x and z and a gradient ascent on $\boldsymbol{u} = \varepsilon v$,

$$x_{k+1} = \arg\min_x E_\varepsilon(x, z_k, \boldsymbol{u}_k/\varepsilon) = \operatorname{prox}_{\varepsilon F(\cdot,y)}(z_k - \boldsymbol{u}_k) , \tag{6}$$

$$z_{k+1} = \arg\min_z E_\varepsilon(x_{k+1}, z, \boldsymbol{u}_k/\varepsilon) = \operatorname{prox}_{\varepsilon U}(x_{k+1} + \boldsymbol{u}_k) = D_\varepsilon(x_{k+1} + \boldsymbol{u}_k) , \tag{7}$$

$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + x_{k+1} - z_{k+1} . \tag{8}$$

Similarly, when $F(., y)$ is differentiable, the simpler Forward-backward splitting (FBS) scheme [19], which only requires to compute ∇F, can be written in a Plug-and-Play fashion as

$$x_{k+1} = \operatorname{prox}_{\varepsilon U}(x_k - \varepsilon \nabla F(x_k, y)) = D_\varepsilon(x_k - \varepsilon \nabla F(x_k, y)) . \tag{9}$$

A fully proximal version of this algorithm, called Backward-backward splitting (BBS) [19], writes

$$x_{k+1} = \operatorname{prox}_{\varepsilon U}(\operatorname{prox}_{\varepsilon F}(x_k)) = D_\varepsilon(\operatorname{prox}_{\varepsilon F}(x_k)) . \tag{10}$$

BBS aims at solving a slightly modified version of (3) where F is replaced by its Moreau envelope with parameter ε. The same algorithm can be derived using half-quadratic splitting to solve (3).

When U is convex, such splitting schemes and many variants (including primal-dual methods, ISTA or FISTA, etc.) are well understood and proved to converge to the global optimum [8]. They have also been successfully used for non-convex U like patch-based Gaussian mixture models (GMM) as pioneered for external learning by Zoran & Weiss in [75]. The use of splitting schemes with non-convex GMM priors was later refined with convergence guarantees for scene-adapted learning [66].

Following the seminal work [68], this kind of splitting schemes have become ubiquitous in cases where U (and hence $D_\varepsilon^\dagger$) are unknown and unspecified, but a denoiser D_ε is available and assumed to be a good approximation of $D_\varepsilon^\dagger = \operatorname{prox}_{\varepsilon U}$. As popular and efficient these methods have become, their convergence properties have remained largely unknown. Indeed, for most denoisers D_ε, there is no guarantee that there exists a potential U such that $D_\varepsilon = \operatorname{prox}_{\varepsilon U}$. In [62], Sreehari *et al.* establish some sufficient conditions for this to happen: D_ε must be differentiable, and its Jacobian J_{D_ε} should be symmetric with eigenvalues within the $[0, 1]$ interval to ensure non expansiveness. These assumptions hold for transform-domain thresholding denoisers and for variants of Non Local means [10] where symmetry is explicitly enforced [62]. More recently, it has also been shown [48] that a special class of linear denoisers (including Non Local Means [10]) are proximal operators of some closed, proper functions. This approach necessitates to work with a non standard inner product though. The previous proofs do not hold for most popular denoisers, including BM3D [20], Non Local Bayes [39] and neural networks denoisers like DnCNN [72], as observed in [55].

Consensus equilibirum / fixed point interpretation. Since it remains difficult to show that PnP schemes converge to the MAP or even a critical point of (3), several authors have proposed to analyse these schemes from a consensus equilibrium point of view [13,2], or similarly to consider and analyse these approches as fixed-point algorithms [63,60]. The fixed points attained by these algorithms cannot be interpreted as MAP estimators, but should be seen as solving a set of equilibrium equations

involving both the denoiser and the data term. For instance, for PnP-FBS, the idea is to show convergence to the set of points x satisfying $x = D_\varepsilon(x - \varepsilon\nabla F(x, y))$. It can easily be shown that the fixed points of several of these PnP algorithms (in particular PnP-ADMM and PnP-FBS) coincide [43,63].

In [63], assuming that such fixed points exist, Sun et al. show convergence of PnP-ISTA (which is equivalent to PnP-FBS above) under the assumptions that ∇F is $\mathtt{L}_y$-Lipschtitz, $\varepsilon\mathtt{L}_y \leqslant 1$ and D_ε is θ-averaged, see [5, Definition 4.33] for a definition. This assumption on the denoiser is probably too strong, since most denoisers cannot be considered as averaged operators. In [64], Sun et al. reformulate PnP-ADMM with different convergence conditions, and still assume quite restrictive conditions on the denoiser D_ε [3].

In [60], Ryu *et al.* propose a convergence analysis of PnP-ADMM, PnP-FBS and PnP-DRS (PnP Douglas-Rachford Splitting), based on the weaker assumption that the residual operator $D_\varepsilon - \mathrm{Id}$ is $\mathtt{L}$-Lipschitz with a Lipschitz constant which depends both on the data fitting term F and the denoiser D_ε. The proof also requires F to be μ-strongly convex (which excludes all cases where $\mathbf{A}$ is not full rank and de facto excludes some of the applications considered in [60]) and it imposes quite restrictive assumptions on relative values of μ, ε and $\mathtt{L}$.

In a similar direction, Xu et al. [70] very recently proposed a convergence study for PnP-ISTA, with the assumption that ∇F is $\mathtt{L}_y$-Lipschitz with $\varepsilon\mathtt{L}_y \leqslant 1$. However, they assume that D_ε is an exact MMSE denoiser, *i.e.* $D_\varepsilon(x_\varepsilon) = \mathbb{E}[x|x_\varepsilon]$, where $x \sim p$ and $x_\varepsilon \sim \mathcal{N}(x, \varepsilon\,\mathrm{Id})$ conditionally to x. Therefore their theoretical results do not carry to many classical denoisers, such that those learned from training data and implemented by neural networks.

Assumptions on algorithm parameters. Most of the convergence proofs for PnP algorithms impose restrictive assumptions on the choice of parameters used in the iterative schemes. This may exclude interesting ranges of parameters for several inverse problems. For instance, for PnP-FBS, the parameter ε (which can be interpreted as the step of the proximal or gradient descents) and the Lipschitz parameter $\mathtt{L}_y$ of ∇F must typically be chosen such that $\mathtt{L}_y\varepsilon \leqslant C$ with $C \in [1, 2]$ (see [60,70], the exact value of C depends on the convergence proof). If $F(x, y) = \frac{1}{2\alpha\sigma^2}\|\mathbf{A}x - y\|^2$, with $\|\mathbf{A}\| \leq 1$, it implies that $\frac{1}{\alpha} \leqslant \frac{\sigma^2}{\varepsilon}$. The parameter ε is imposed by the denoiser D_ε (the denoiser is trained for a noise of variance ε), and σ is given by the quantity of noise in the forward model. If, for instance, the forward model involves a noise standard deviation σ which is 5 times smaller than the one used for the denoiser D_ε, it means that the penalty α (which balances the respective weights of the data and prior terms) should be chosen larger than 25, which implies that the algorithm will only converges for huge regularizations. We will see in the experimental section that for this kind of reason the PnP-FBS algorithm often fails to converge for classical imaging inverse problems, or converges only for values of α which are not interesting in practice. Fully proximal algorithms such as PnP-ADMM or PnP-BBS are much more robust in practice, even when the conditions of their theoretical convergence are not fully met. The PnP-SGD algorithm that we will introduce in the following does not suffer from the same convergence limitations.

[3] In [64], the residual $\mathrm{Id} - D_\varepsilon$ is assumed to be firmly non expansive, which is equivalent to say that D_ε is firmly non expansive, see [5, Proposition 4.4].

AMP algorithms. It is worth mentioning at this point that the Plug-and-Play framework has also been shown to be very efficient with Approximate Message Passing algorithms [2]. These algorithms have excellent convergence properties for data terms of the form $\|\mathbf{A}x-y\|^2$ with $\mathbf{A}$ belonging to specific classes of random matrices. This restriction on $\mathbf{A}$ does not hold for the inverse problems considered in the current paper so we focus instead on classical optimization scheme such as the ones described above.

2.2 Plug & Play MAP estimators using gradient descent

Now, assume that $D_\varepsilon = D_\varepsilon^\star$, where $D_\varepsilon^\star$ is the MMSE estimator to recover x from the noisy observation $x_\varepsilon \sim \mathcal{N}(x, \varepsilon\,\mathrm{Id})$ when x has marginal density $p(x)$; that is,

$$D_\varepsilon^\star(x_\varepsilon) = \mathbb{E}[x|x_\varepsilon] = \textstyle\int_{\mathbb{R}^d} z p(z) G_\varepsilon(x_\varepsilon - z)\mathrm{d}z \,/\, \int_{\mathbb{R}^d} p(z) G_\varepsilon(x_\varepsilon - z)\mathrm{d}z\ , \tag{11}$$

where G_ε is a Gaussian kernel with variance ε, meaning that for all $x \in \mathbb{R}^d$,

$$G_\varepsilon(x) = (2\pi\varepsilon)^{-d/2} \exp[-\|x\|^2/(2\varepsilon)]\ .$$

We introduce the following class of smooth approximations of $p(x)$, defined for any $x \in \mathbb{R}^d$ by

$$p_\varepsilon(x) = \textstyle\int_{\mathbb{R}^d} p(\tilde{x}) G_\varepsilon(x - \tilde{x})\mathrm{d}\tilde{x}\ . \tag{12}$$

In this case, Tweedie's identity [28] establishes the following relationship between the MMSE denoiser $D_\varepsilon^\star$ and (12), for any $x \in \mathbb{R}^d$

$$\nabla U_\varepsilon(x) = -\nabla \log p_\varepsilon(x) = (x - D_\varepsilon^\star(x))/\varepsilon\ , \tag{13}$$

where $U_\varepsilon = -\log(p_\varepsilon)$. This relation can be used to plug the MMSE denoiser $D_\varepsilon^\star$ in any gradient descent scheme involving ∇U_ε and it is at the core of the algorithm PnP-SGD presented in this paper. Similarly to the MAP denoiser $D_\varepsilon^\dagger$, the MMSE denoiser $D_\varepsilon^\star$ is usually not known, so PnP methods rely on other denoisers D_ε that are believed to be good approximations of $D_\varepsilon^\star$. Observe that CNN denoisers are usually trained to minimize an empirical quadratic risk on a large database of natural images. As a consequence, they naturally produce good approximations of MMSE denoisers $D_\varepsilon^\star$ for realistic image priors. This makes approaches based on Tweedie's identity particularly attractive. On the other hand, learning mechanisms to produce good approximations of MAP denoisers $D_\varepsilon^\dagger$ are much less widespread, although under some conditions, MMSE denoisers can be shown to be MAP denoisers on a different prior (see [33, 70, 35]).

A similar relation is derived by Romano *et al.* in [58] where they present the Regularization by Denosing (RED) method, which proposes an insightful Bayesian formulation of denoiser-based priors as image-adaptive Laplacian regularizations. Instead of using Tweedie's identity, the RED method solves equation (3) via different optimization algorithms (including gradient descent and ADMM) with explicit regularization $U_\varepsilon(x) = (1/2)\langle x,\, x - D_\varepsilon(x)\rangle$. As shown in [55], under the assumptions that D_ε is locally homogeneous and has symmetric Jacobian, this implies that for any $x \in \mathbb{R}^d$, $\nabla U_\varepsilon(x) = x - D_\varepsilon(x)$, which is (up to a scaling factor $1/\varepsilon$) the same expression as Tweedie's identity in (13). Unfortunately, as pointed out before, these assumptions on D_ε are not strictly satisfied by most commonly used denoisers [55],

although we note that Jacobian symmetry can be explicitely enforced [45]. The convergence of the RED algorithms for denoisers that do not verify the above-mentioned assumption remains unproven. As an alternative interpretation the RED algorithm can be seen as a way to approximate the score ∇U_ε by $(x - D_\varepsilon(x))/\varepsilon$ in the optimality equation $\nabla F + \nabla U_\varepsilon = 0$. Here the optimal MMSE denoiser $D^\star_\varepsilon$ is again replaced by some other denoiser.

More recently, [18] studies a projected RED estimator which seeks to minimise a data fidelity term subject to the constraint that the solution belongs to the set of fixed points $\{x \in \mathbb{R}^d : x = D_\varepsilon(x)\}$, thus sharing strong link with the consensus equilibrium interpretation of proximal-based PnP estimators. It is reported in [18] that when D_ε is a demi-contractive mapping, its fixed points define a convex set, which allows the construction of provably convergent algorithms for this alternative RED estimator. However, as pointed out in [54], verifying that a given denoising operator is demi-contractive is not easy and, to be the best of our knowledge, it is not yet clear what denoisers verify this property. Furthermore, from a Bayesian inference viewpoint, additional studies would be required in order to determine when this projected RED estimator defines or approximates a MAP estimator for a suitable Bayesian model - we leave this as a perspective for future work.

In a similar direction, two very recent works [34,35] show how to train efficiently a denoiser that explicitly satisfies $D_\varepsilon(x) = x - \nabla g_\varepsilon(x)$ for some functional g_ε. Plugging this denoiser in appropriate PnP schemes, they are able to prove convergence to stationary points of an explicit cost function.

The PnP-SGD optimisation algorithm that will be presented in the next section is very close to the gradient descent version of RED presented in [58]. We will show that it converges to the vicinity of the solution of (3) under much milder conditions than previously assumed. In particular, the convergence guarantees hold even when D_ε is not an exact MAP or MMSE denoiser, which is often the case in practice. Importantly, our convergence guarantees hold for the neural network denoiser used in [60] (a variant of DnCNN [72] with a contractive residual) and also for the native Non Local Means [11].

3 PnP maximum-a-posteriori estimation: analysis and computation

3.1 Analysis of maximum-a-posteriori estimation with PnP priors

We are interested in MAP estimation for Bayesian models involving PnP priors that are defined implicitly by an image denoising algorithm D_ε. We pay special attention to the highly practically relevant case in which D_ε approximates the optimal MMSE denoiser $D^\star_\varepsilon$ associated to p, i.e., $D^\star_\varepsilon = \mathbb{E}[x|x_\varepsilon]$ for $x_\varepsilon \sim \mathcal{N}(x, \varepsilon\,\mathrm{Id})$ when x has marginal density p. As mentioned previously, state-of-the-art denoisers based on neural networks are often trained to approximate $D^\star_\varepsilon$ by using a sample of clean images $\{x_i\}_{i=1}^N$ from p, corresponding noisy samples $\{x'_i\}_{i=1}^N$ with $x'_i \sim \mathcal{N}(x_i, \varepsilon\,\mathrm{Id})$, and choosing D_ε to approximately minimize the empirical MSE loss $\sum_{i=1}^N \|D_\varepsilon(x'_i) - x_i\|^2$. Similarly, many state-of-the-art patch-based image denoisers are also designed to approximate $D^\star_\varepsilon$.

The fact that D_ε is only an approximation of $D^\star_\varepsilon$ leads to several complications in the analysis and computation of MAP solutions. For example, unlike $D^\star_\varepsilon$, D_ε does not define a gradient mapping in general, and key results such as Tweedie's identity [28]

do not hold. Moreover, in the case of neural network denoisers trained from samples $\{x_i\}_{i=1}^N$ from p, the model is unknown as it is only available through $\{x_i\}_{i=1}^N$, making it difficult to check that basic regularity properties required for MAP estimation are satisfied.

Rather than imposing strong assumptions on D_ε, we address these difficulties by formulating our analysis in the *M-complete* Bayesian framework, in which we assume that the posterior $p(x|y)$ associated with the true prior $p(x)$ exists but remains largely unknown, and all inference on $x|y$ are conducted by using operational approximations of this true model [6]. In particular, we focus on the class of smooth approximations of $p(x|y)$ given for any $\varepsilon > 0$ and $x \in \mathbb{R}^d$ by

$$p_\varepsilon(x|y) = \frac{p_\varepsilon(x)p(y|x)}{\int_{\mathbb{R}^d} p_\varepsilon(\tilde{x})p(y|\tilde{x})\mathrm{d}\tilde{x}} \,, \tag{14}$$

where $p_\varepsilon(x)$ is the smooth approximation of the prior $p(x)$ defined in (12). We will study MAP estimation for $p_\varepsilon(x|y)$ to establish that the procedure is well defined, well posed, amenable to efficient computation, and that it provides a useful approximation to MAP estimation with the true posterior $p(x|y)$. Following on from this, Section 3.2 will study the computation of MAP solutions for $p_\varepsilon(x|y)$ by using PnP SGD with a generic denoiser D_ε that approximates $D_\varepsilon^\star$, where we will pay particular attention to the conditions on D_ε required to ensure convergence, as well as to the bias introduced by using D_ε instead of $D_\varepsilon^\star$.

It is established in [38] that, under basic assumptions on the likelihood function $p(y|x)$ detailed in H1 below, the posterior approximation $p_\varepsilon(x|y)$ is well defined, proper, and can be made as close to the true posterior $p(x|y)$ as desired by reducing the value of ε, with the approximation error vanishing as $\varepsilon \to 0$. Crucially, [38] also establishes that, under H1 and mild assumptions on the optimal MMSE denoiser $D_\varepsilon^\star$ (essentially, that the denoising problem underlying $D_\varepsilon^\star$ is well posed in the sense of Hadamard), then $x \mapsto p_\varepsilon(x|y)$ is differentiable with $x \mapsto \nabla \log p_\varepsilon(x|y)$ Lipschitz continuous. We conclude that the approximation $p_\varepsilon(x|y)$ is well defined and amenable to computation by first-order schemes, such as SGD to compute critical points of $p_\varepsilon(x|y)$ and perform MAP estimation.

H1 *For any $y \in \mathbb{R}^m$, $\sup_{x\in\mathbb{R}^d} p(y|x) < +\infty$, $p(y|\cdot) \in \mathrm{C}^1(\mathbb{R}^d, (0, +\infty))$. In addition, there exists $\mathsf{L}_y > 0$ such that $\nabla \log p(y|\cdot)$ is L_y Lipschitz continuous.*

With the above-mentioned properties of $p_\varepsilon(x|y)$ in mind, we wonder if computing a MAP solution for $p_\varepsilon(x|y)$ provides useful information about a MAP solution for $p(x|y)$. More precisely, we study if critical points for $p_\varepsilon(x|y)$ are stable w.r.t. variations in ε, and if they converge to critical points of $p(x|y)$ as $\varepsilon \to 0$. Proposition 1 below establishes that this is indeed the case. In words, MAP solutions computed with $p_\varepsilon(x|y)$ are in the neighbourhood of MAP solutions for $p(x|y)$, with ε controlling a trade-off between the computational efficiency of first-order schemes and the accuracy of the delivered solutions w.r.t. $p(x|y)$. When ε is large, the approximation of the posterior is smoother so gradient descent can be used with larger steps to improve convergence speed (as the gradients have a smaller Lipschitz constant).

Formally, we investigate the stability of the set of stationary points $\mathsf{S}_{\varepsilon,\mathsf{K}} = \{x \in \mathsf{K} \,:\, \nabla \log p_\varepsilon(x|y) = 0\}$ w.r.t. $\varepsilon > 0$, where K is a compact set. We show that for any sequence $(\varepsilon_n, x_n)_{n\in\mathbb{N}}$ such that $\lim_{n\to+\infty} \varepsilon_n = 0$ and for any $n \in \mathbb{N}$, $x_n \in \mathsf{S}_{\varepsilon_n,\mathsf{K}}$ every cluster point of $(x_n)_{n\in\mathbb{N}}$ belongs to $\mathsf{S}_\mathsf{K} = \{x \in \mathsf{K} \,:\, \nabla \log p(x|y) = 0\}$. In other

words, we show that the stationary points of the approximate posterior are close to the ones of the true posterior.

Proposition 1 *Assume* H1 *and that* $p \in \mathrm{C}^1(\mathbb{R}^d, (0, +\infty))$ *with* $\|p\|_\infty + \|\nabla p\|_\infty < +\infty$. *Then for any compact set* $\mathsf{K} \subset \mathbb{R}^d$ *and* $(x_{\varepsilon_n})_{n\in\mathbb{N}}$ *such that* $\lim_{n\to+\infty} \varepsilon_n = 0$ *and for any* $n \in \mathbb{N}$, $x_{\varepsilon_n} \in \mathsf{S}_{\varepsilon_n,\mathsf{K}}$, *we have that any cluster point* $x^\star$ *of* x_{ε_n} *satisfies* $x^\star \in \mathsf{S}_{\mathsf{K}}$.

Proof Let $\mathsf{K} \subset \mathbb{R}^d$ be a compact set and $(x_n, \varepsilon_n)_{n\in\mathbb{N}}$ such that $\lim_{n\to+\infty} \varepsilon_n = 0$ and for any $n \in \mathbb{N}$, $x_n \in \mathsf{S}_{\varepsilon_n,\mathsf{K}}$. Let $x^\star \in \mathsf{S}$ a cluster point of $(x_n)_{n\in\mathbb{N}}$. Hence, for any $n \in \mathbb{N}^*$ there exist an increasing sequence $(k_n)_{n\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ such that $\lim_{n\to+\infty} x_{k_n} = x^\star$.

In what follows, we show that $\lim_{n\to+\infty} \nabla \log(p_{\varepsilon_{k_n}}(x_{k_n})) = \nabla \log p(x^\star)$. First, we show that

$$\lim_{n\to+\infty} \max(|p - p_{\varepsilon_{k_n}}|_{\infty,\mathsf{K}}, \|\nabla p - \nabla p_{\varepsilon_{k_n}}\|_{\infty,\mathsf{K}}) = 0 \,. \tag{15}$$

Indeed, let $f \in \mathrm{C}(\mathbb{R}^d, \mathbb{R}^m)$ with $m \in \mathbb{N}$ such that $\|f\|_\infty < +\infty$ and denote $f_\varepsilon \in \mathrm{C}(\mathbb{R}^d, \mathbb{R}^m)$ given for any $x \in \mathbb{R}^d$ by

$$f_\varepsilon(x) = \textstyle\int_{\mathbb{R}^d} f(\tilde{x}) G_\varepsilon(x - \tilde{x}) \mathrm{d}\tilde{x} \,, \tag{16}$$

where we recall that for any $x \in \mathbb{R}^d$, $G_\varepsilon(x)$ is a Gaussian kernel with variance ε. For ease of notation, we define $G = G_1$. Let $\eta > 0$. Then, there exists $R > 0$ such that for any $\varepsilon > 0$ we have

$$\textstyle\int_{\|\tilde{x}\|>R} \|f(x - \varepsilon^{1/2}\tilde{x}) - f(x)\| G(\tilde{x}) \mathrm{d}\tilde{x} \leqslant 2\|f\|_\infty \int_{\|\tilde{x}\|>R} G(\tilde{x}) \mathrm{d}\tilde{x} < \eta/2 \,. \tag{17}$$

Let $\mathsf{K}' = \mathsf{K} + \overline{\mathrm{B}}(0, R)$. We have that K' is compact and therefore f is uniformly continuous on K'. Hence there exists $\xi > 0$ such that for any $x \in \mathsf{K}$, $\varepsilon \in (0, \xi]$ and $y \in \overline{\mathrm{B}}(0, R)$ we have

$$|f(x - \varepsilon^{1/2} y) - f(x)| \leqslant \eta/2 \,. \tag{18}$$

Hence, combining (17) and (18) we get that for any $x \in \mathsf{K}$ and $\varepsilon \in (0, \xi]$

$$\begin{aligned}
\|f_\varepsilon(x) - f(x)\| &\leqslant \textstyle\int_{\mathbb{R}^d} \|f(x - \tilde{x}) - f(x)\| G_\varepsilon(\tilde{x}) \mathrm{d}\tilde{x} && (19)\\
&\leqslant \textstyle\int_{\mathbb{R}^d} \|f(x - \varepsilon^{1/2}\tilde{x}) - f(x)\| G(\tilde{x}) \mathrm{d}\tilde{x} && (20)\\
&\leqslant \textstyle\int_{\overline{\mathrm{B}}(0,R)} \|f(x - \varepsilon^{1/2}\tilde{x}) - f(x)\| G(\tilde{x}) \mathrm{d}\tilde{x} && (21)\\
&\quad + \textstyle\int_{\overline{\mathrm{B}}(0,R)^{\mathrm{c}}} \|f(x - \varepsilon^{1/2}\tilde{x}) - f(x)\| G(\tilde{x}) \mathrm{d}\tilde{x} \leqslant \eta \,. && (22)
\end{aligned}$$

Hence $\lim_{\varepsilon\to 0} \|f - f_\varepsilon\|_{\infty,\mathsf{K}} = 0$. Therefore using this result and that $p \in \mathrm{C}^1(\mathbb{R}^d, \mathbb{R})$ with $\|p\|_\infty + \|\nabla p\|_\infty < +\infty$ we get that

$$\lim_{n\to+\infty} \max(|p - p_{\varepsilon_{k_n}}|_{\infty,\mathsf{K}}, \|\nabla p - \nabla p_{\varepsilon_{k_n}}\|_{\infty,\mathsf{K}}) = 0 \,. \tag{23}$$

Combining this result, the fact that $\lim_{n\to+\infty} x_{k_n} = x^\star$ and that $p > 0$, we get that $\lim_{n\to+\infty} \nabla \log(p_{\varepsilon_{k_n}})(x_{k_n}) = \nabla \log p(x^\star)$. Indeed, we have that for any $n \in \mathbb{N}$

$$\begin{aligned}
&\|\nabla \log(p_{\varepsilon_{k_n}}(x_{k_n})) - \nabla \log p(x^\star)\| && (24)\\
&\quad \leqslant \|\nabla \log(p_{\varepsilon_{k_n}}(x_{k_n})) - \nabla \log p(x_{k_n})\| + \|\nabla \log p(x_{k_n}) - \nabla \log p(x^\star)\| \,. && (25)
\end{aligned}$$

We conclude using (23) and that $\log p \in \mathrm{C}(\mathbb{R}^d, \mathbb{R})$. Finally, we obtain that

$$0 = \lim_{n\to+\infty} \left\{ \nabla \log p(y|x_{k_n}) + \nabla \log p_{\varepsilon_{k_n}}(x_{k_n}) \right\} = \nabla \log p(y|x^\star) + \nabla \log p(x^\star) \,. \tag{26}$$

Hence, $x^\star \in \mathsf{S}_\mathsf{K}$. □

Note that the above result can be strengthened to show the convergence at the levels of sets. More precisely, we can show that any cluster point of $\{\mathsf{S}_{\mathsf{K},\varepsilon}\}_{\varepsilon>0}$ is a subset of S_K in the sense of the Haussdorff distance, see [47] for a definition.

As a third and final point in our analysis, we study if MAP estimation for $p_\varepsilon(x|y)$ is a well-posed estimation procedure, which is an essential requirement for meaningful inference. One would ideally seek to establish the existence of a unique global maximiser that is Lipschitz continuous w.r.t. perturbations of the observed data y. Unfortunately, this is not possible without imposing very strong assumptions on the model. Instead, Proposition 2 below shows that, under some assumptions on the likelihood $p(y|x)$, the set of critical points of $p_\varepsilon(x|y)$ is locally Lipschitz continuous w.r.t. perturbations of y, which is a weaker form of well-posedness. Notice that the assumptions on the likelihood can be relaxed when $D_\varepsilon^\star$ is contractive, but this is usually unrealistic. This highlights a limitation of MAP estimation by comparison to other Bayesian estimators, namely MMSE estimation, which is shown in [38] to be well-posed under significantly weaker assumptions.

Proposition 2 *Assume* H1 *and that* $(x, y) \mapsto p(y|x) \in \mathrm{C}^2(\mathbb{R}^d \times \mathbb{R}^m, \mathbb{R})$. *Let* $\varepsilon > 0$, *we have that* $(x, y) \mapsto p_\varepsilon(x|y) \in \mathrm{C}^2(\mathbb{R}^d \times \mathbb{R}^m, \mathbb{R}_+)$. *Let* $y_0 \in \mathbb{R}^m$ *denote some observed data and* $x^\star_{y_0} \in \mathbb{R}^d$ *a local maximiser of the posterior* $x \mapsto p_\varepsilon(x|y_0)$ *with* $\nabla^2 \log p_\varepsilon(x^\star_{y_0}|y_0)$ *negative. Then there exists an open set* $\mathsf{V}_0 \subset \mathbb{R}^m$ *and a function* $x^\star(y) \in \mathrm{C}^1(\mathsf{V}_0, \mathbb{R}^d)$ *such that* $y_0 \in \mathsf{V}_0$ *and for any* $y \in \mathsf{V}_0$, $x^\star(y)$ *is a strict local maximizer of* $x \mapsto p_\varepsilon(x|y)$.

Proof First, using that $p \in \mathrm{C}(\mathbb{R}^d, \mathbb{R}_+)$ we have that for any $v \in \mathbb{R}^d$ and $c \in \mathbb{R}$ there exists $\mathsf{A} \in \mathcal{B}(\mathbb{R}^d)$ such that $\int_\mathsf{A} |\langle x, v\rangle - c|\, p(x)\mathrm{d}x > 0$, meaning that there is no lower-dimensional affine space of $\mathbb{R}^d$ to which x belongs almost surely. Hence, we can apply [33, Lemma II.1] and $D_\varepsilon^\star \in \mathrm{C}^\infty(\mathbb{R}^d, \mathbb{R}^d)$.

We have that for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$, $\nabla \log p_\varepsilon(x|y) = \nabla \log p(y|x) + D_\varepsilon^\star(x)$. Hence, we get that $(x, y) \mapsto p_\varepsilon(x|y) \in \mathrm{C}^2(\mathbb{R}^d \times \mathbb{R}^m, \mathbb{R}_+)$. Since $-\nabla^2 \log p(x^\star_{y_0}|y_0)$ is positive there exist $\mathsf{U}_1 \subset \mathbb{R}^d$ open and $\mathsf{V}_1 \subset \mathbb{R}^m$ open such that for any $x \in \mathsf{U}_1$ and $y \in \mathsf{V}_1$, $-\nabla^2 \log p_\varepsilon(x|y)$ is positive. Hence, for any $y \in \mathsf{V}_1$, $x \in \mathsf{U}_1$ is a strict local maximizer if and only $\nabla \log p_\varepsilon(x|y) = 0$.

We have that $\nabla_x(\nabla_x \log p_\varepsilon)(x^\star_{y_0}|y_0)$ is invertible. Therefore using the implicit function theorem, there exist $\mathsf{V}_0 \subset \mathbb{R}^m$ open and $x^\star \in \mathrm{C}^1(\mathsf{V}_0, \mathsf{U}_1)$ such that for any $y \in \mathsf{V}_0$, $\nabla \log p_\varepsilon(x^\star(y)|y) = 0$, *i.e.* $x^\star(y)$ is a strict local maximizer of $x \mapsto \log p_\varepsilon(x|y)$, since $-\nabla^2 \log p_\varepsilon(x^\star(y)|y)$ is positive, which concludes the proof. □

To conclude, a major challenge in understanding Bayesian inference with PnP priors and providing guarantees for the delivered solutions is that the underlying prior and posterior densities $p(x)$ and $p(x|y)$ are unknown. Also, the image denoiser D_ε used to construct PnP schemes is not usually directly related to the model. Instead, when it approximates the optimal MMSE denoiser $D_\varepsilon^\star$, it is indirectly related to the model via Tweedie's identity and the smooth approximations $p_\varepsilon(x)$ and $p_\varepsilon(x|y)$. We establish that these operational approximations are useful for MAP inference for

$x|y$, in the sense that they are well defined, proper, and MAP solutions for $p_\varepsilon(x|y)$ can be made arbitrarily close to the true MAP solutions through the choice of ε. Importantly, under some assumptions, MAP solutions for $p_\varepsilon(x|y)$ are well posed and amenable to efficient computation by first order optimisation methodology.

3.2 PnP-SGD and convergence

We are now ready to study the computation of MAP solutions for $p_\varepsilon(x|y)$ by using PnP SGD with a generic denoiser D_ε that approximates $D^\star_\varepsilon$. We pay particular attention to the conditions on D_ε required to ensure convergence, and to the bias introduced by using D_ε instead of $D^\star_\varepsilon$.

We begin by using Tweedie's identity to express SGD to compute critical points of $p_\varepsilon(x|y)$ as the following sequence: $X_0 \in \mathbb{R}^d$ and for any $k \in \mathbb{N}$

$$X_{k+1} = X_k - \delta_k \nabla F(X_k, y) - \delta_k/\varepsilon (X_k - D^\star_\varepsilon(X_k)) + \delta_k Z_{k+1} \,, \tag{27}$$

where $(\delta_k)_{k\in\mathbb{N}} \in (\mathbb{R}_+)^{\mathbb{N}}$ is a sequence of step-sizes, $\varepsilon > 0$, and $\{Z_k \,:\, k \in \mathbb{N}\}$ a family of i.i.d. Gaussian random variables with zero mean and identity covariance matrix. We recall that the sequences $(X_k)_{k\in\mathbb{N}}$ and $(Z_k)_{k\in\mathbb{N}}$ are defined on an underlying probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

As mentioned previously, in most practically relevant cases $D^\star_\varepsilon$ is an abstract quantity that cannot be computed. Instead, we have a different denoiser D_ε that can be assumed to be a good approximation of $D^\star_\varepsilon$. For example, when we have access to samples $\{x_i\}_{i=1}^N$ from p we can consider a noisy version of these samples $\{x'_i\}_{i=1}^N$ with level $\varepsilon > 0$ and train a neural network based denoiser D_ε to minimize the loss $\sum_{i=1}^N \|D_\varepsilon(x'_i) - x_i\|^2$. This loss corresponds to the empirical version of $\mathbb{E}[\|D_\varepsilon(x_\varepsilon) - x\|^2]$ (with $x \sim p$ and $x_\varepsilon \sim \mathcal{N}(x, \varepsilon\,\mathrm{Id})$ conditionally to x) whose minimizer is the MMSE $D^\star_\varepsilon$.

Using a generic denoiser D_ε in our SGD scheme in lieu of $D^\star_\varepsilon$ we obtain the Plug & Play SGD algorithm associated with following recursion: $X_0 \in \mathbb{R}^d$ and for any $k \in \mathbb{N}$

$$X_{k+1} = X_k + \delta_k (b_\varepsilon(X_k) + Z_{k+1}) \,, \tag{28}$$

$$b_\varepsilon(x) = \nabla \log(p(y|x)) + \alpha(D_\varepsilon(x) - x)/\varepsilon \,, \tag{29}$$

where we note that we have introduced a regularization parameter $\alpha > 0$ that controls the amount of regularisation enforced by D_ε. The original SGD algorithm is recovered by setting $\alpha = 1$ and $D_\varepsilon = D^\star_\varepsilon$.

We now turn to the proof of convergence of PnP-SGD. The asymptotic estimates we derive in this work are only valid for sequences which remain in a compact set K, which is a classical assumption in stochastic approximation [65, 23, 22, 44]. Under tighter conditions on $x \mapsto \log p_\varepsilon(x|y)$ this limitation can be circumvented using the global asymptotic results of [65, Theorem A1.1]. Another way to remove this restriction would be to consider an additive term of the form $x \mapsto (x - \mathit{\Pi}_{\mathsf{C}}(x))/\lambda$ in b_ε (where $\mathit{\Pi}_{\mathsf{C}}$ is the projection onto some compact convex set C and $\lambda > 0$ some hyperparameter) which ensures the stability of the numerical scheme. We leave this analysis for future work. In practice, we have not observed any stability issues for PnP-SGD provided that the stepsize is chosen appropriately see Section 4.3.

In what follows, we show that the bias of PnP-SGD depends on the distance between D_ε and the MMSE estimator $D^\star_\varepsilon$, using recent results from [65].

Algorithm 1 PnP-SGD

Require: $n, n_{\texttt{burnin}} \in \mathbb{N}$, $y \in \mathbb{R}^m$, $\varepsilon, \alpha, \delta > 0$
  Initialization: Set $X_0 = \tilde{x}$ and $k = 0$.
  for $k = 0 : N$ **do**
    $Z_{k+1} \sim \mathcal{N}(0, \mathrm{Id})$
    if $k \leqslant n_{\texttt{burnin}}$ **then**
      $X_{k+1} = X_k + \delta_0 \nabla \log(p(y|X_k)) + (\delta_0 \alpha/\varepsilon)(D_\varepsilon(X_k) - X_k) + \delta_0 Z_{k+1}$
    end if
    if $k > n_{\texttt{burnin}}$ **then**
      $X_{k+1} = X_k + \delta_k \nabla \log(p(y|X_k)) + (\delta_k \alpha/\varepsilon)(D_\varepsilon(X_k) - X_k) + \delta_k Z_{k+1}$
      $\delta_{k+1} = \delta_0 (k + 1 - n_{\texttt{burnin}})^{-0.8}$
    end if
  end for
  return X_N

H2 *Assume that there exist* $\varepsilon_0 > 0$, $\mathtt{L} \geqslant 0$ *and a function* $\mathtt{M} : \mathbb{R}^+ \to \mathbb{R}^+$ *such that for any* $\varepsilon \in (0, \varepsilon_0]$, $R \geqslant 0$, $x_1, x_2 \in \mathbb{R}^d$ *and* $x \in \overline{\mathrm{B}}(0, R)$ *we have*

$$\|D_\varepsilon(x_1) - D_\varepsilon(x_2)\| \leqslant \mathtt{L} \|x_1 - x_2\| \ , \qquad \left\|D_\varepsilon(x) - D^\star_\varepsilon(x)\right\| \leqslant \mathtt{M}(R) \ , \tag{30}$$

where we recall that

$$D^\star_\varepsilon(x_1) = \textstyle\int_{\mathbb{R}^d} \tilde{x} \ g_\varepsilon(\tilde{x}|x_1) \mathrm{d}\tilde{x} \ , \tag{31}$$

with $\tilde{x} \mapsto g_\varepsilon(\tilde{x}|x)$ *the probability density of* X *given* $X_\varepsilon = x$ *where* $X_\varepsilon \sim \mathcal{N}(X, \varepsilon \, \mathrm{Id})$ *conditionally to* X *and* $X \sim p$.

The first part of (30) regarding the smoothness property of the denoiser can be explicitly verified for a certain class of neural networks by adding a spectral regularization term for each layer of the neural network, see [60,46]. The second condition follows from carefully selecting the loss of the neural network as in the Noise2Noise network introduced in [40] and controlling the population error. We refer the reader to [38] for more details regarding the role of the bounding function $\mathtt{M}(R)$. In particular, for neural network denoisers, [38, Proposition 3.1] explains how to promote low values of $\mathtt{M}(R)$ during training by using a particular loss function. In addition, [38] makes connections with universal approximation results (see e.g., [4, Section 4.7]).

We are now ready to state Proposition 3 which ensures that stable PnP-SGD sequences are close to the set of stationary points of $x \mapsto \log p_\varepsilon(x|y)$ where $x \mapsto \log p_\varepsilon(x|y)$ is given in (14). The distance to this set of stationary points is controlled by the approximation error of the D_ε.

H3 *For any* $y \in \mathbb{R}^m$, $x \mapsto -\log p(y|x)$ *is real-analytic* [4] [5] [6].

[4] A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be real-analytic if for any $x_0 = (x_0^1, \ldots, x_0^d) \in \mathbb{R}^d$ there exists $(a_{n_1,\ldots,n_d})_{n_1,\ldots,n_d \in \mathbb{N}} \in \mathbb{R}^{\mathbb{N}^d}$ and $r > 0$ such that for any $x = (x^1, \ldots, x^d) \in \mathrm{B}(x_0, r)$

$$f(x) = \textstyle\sum_{n_1 \in \mathbb{N}} \cdots \sum_{n_d \in \mathbb{N}} a_{n_1,\ldots,n_d} \prod_{j=1}^d (x^j - x_0^j)^{n_j} \ .$$

[5] The assumption that $x \mapsto \log(p(y|x))$ is real-analytic is satisfied in all of our experiments since there exists $\mathbf{A} \in \mathbb{R}^{p \times d}$ and $\sigma > 0$ such that for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$, $\log p(y|x) = \|\mathbf{A}x - y\|^2/(2\sigma^2)$.

[6] From Liouville's theorem one could think that the simultaneously verifying that $\nabla \log p(y|\cdot)$ is Lipschitz continuous and that $x \mapsto \log(p(y|x))$ is real-analytic restricts our analysis to models

In the following, d stands for the distance induced by the Euclidean norm.

Proposition 3 *Assume* H1*,* H2 *and* H3*. Let* $\alpha > 0$ *and* $\varepsilon \in (0, \varepsilon_0]$*. Assume that* $\lim_{k\to+\infty} \delta_k = 0$*,* $\sum_{k\in\mathbb{N}} \delta_k = +\infty$ *and* $\sum_{k\in\mathbb{N}} \delta_k^2 < +\infty$*. Let* $R > 0$*,* $\mathsf{K} \subset \overline{\mathrm{B}}(0, R)$ *be a compact set,* $X_0 \in \mathbb{R}^d$ *and* $\mathsf{A}_{\varepsilon,\mathsf{K}} \in \mathcal{F}$ *given by*

$$\mathsf{A}_{\varepsilon,\mathsf{K}} = \{\omega \in \Omega : \textit{ there exists } k_0 \in \mathbb{N} \textit{ such that for any } k \geqslant k_0,\ X_k(\omega) \in \mathsf{K}.\}\ , \quad (32)$$

where $(X_k)_{k\in\mathbb{N}}$ *is given by* (28)*. Then there exist* $C_{\varepsilon,\mathsf{K}} \geqslant 0$ *and* $r_{\varepsilon,\mathsf{K}} \in (0, 1)$ *such that* $\limsup_{k\to+\infty} d(X_k(\omega), \mathsf{S}_{\varepsilon,\mathsf{K}}) \leqslant C_{\varepsilon,\mathsf{K}} \mathtt{M}(R)^{r_{\varepsilon,\mathsf{K}}}$ *for any* $\omega \in \mathsf{A}_{\varepsilon,\mathsf{K}}$*, with*

$$\mathsf{S}_{\varepsilon,\mathsf{K}} = \{x \in \mathsf{K} : \nabla \log p_\varepsilon(x|y) = 0\}\ , \quad (33)$$

where $x \mapsto p_\varepsilon(x|y)$ *is given in* (14)*.*

Proof In order to prove this theorem we are going to apply [65, Theorem 2.1]. In particular, in order to follow the notation of [65, Theorem 2.1], we define for $k \in \mathbb{N}$, $\zeta_k = Z_{k+1}$ and $\eta_k = b_\varepsilon(X_k) - \nabla \log p(y|X_k) - \nabla \log p_\varepsilon(X_k)$. Let $\varepsilon > 0$ and $\omega \in \mathsf{A}_{\varepsilon,\mathsf{K}}$. Using H2 we have for any $k \in \mathbb{N}$,

$$\|b_\varepsilon(X_k) - \nabla \log p(y|X_k) - \nabla \log p_\varepsilon(X_k)\| = \varepsilon^{-1} \|D_\varepsilon(X_k) - D_\varepsilon^\star(X_k)\| \leqslant \mathtt{M}(R)/\varepsilon\ . \quad (34)$$

Hence, we obtain that [65, Assumption 2.1, Assumption 2.2] are satisfied. In what follows, we show that [65, Assumption 2.3.c] holds. We have that for any $x \in \mathbb{R}^d$, $p_\varepsilon(x) = (p * G_\varepsilon)(x)$, where $*$ denotes the convolution product. Since $p, G_\varepsilon \in \mathrm{L}^1(\mathbb{R}^d)$ we get that for any $\xi \in \mathbb{R}^d$, $\widehat{p * G_\varepsilon}(\xi) = \hat{p}(\xi)\hat{G}_\varepsilon(\xi)$. Since $p \in \mathrm{L}^1(\mathbb{R}^d)$, $\|\hat{p}\|_\infty < +\infty$ using Riemann-Lebesgue theorem and in addition $\hat{G}_\varepsilon(\xi) = \exp[-\varepsilon \|\xi\|^2 /2]$. Hence, $\widehat{p * G_\varepsilon} \in \mathrm{L}^1(\mathbb{R}^d)$ and we obtain that for almost every $x \in \mathbb{R}^d$

$$p_\varepsilon(x) = \textstyle\int_{\mathbb{R}^d} \hat{p}(\xi)\hat{G}_\varepsilon(\xi) \exp[\mathrm{i}\langle x, \xi\rangle] \mathrm{d}\xi\ . \quad (35)$$

In the rest of the proof, we denote $\bar{p}_\varepsilon : \mathbb{C}^d \to \mathbb{C}$ given for any $z = (z^1, \ldots, z^d) \in \mathbb{C}^d$ by $\bar{p}_\varepsilon(z) = \int_{\mathbb{R}^d} \hat{p}(\xi)\hat{G}_\varepsilon(\xi) \exp[\mathrm{i}\langle z, \xi\rangle] \mathrm{d}\xi$ where for any $z_1, z_2 \in \mathbb{C}^d$ we have $\langle z_1, z_2\rangle = \sum_{j=1}^d z_1^j \bar{z}_2^j$. We have that $\bar{p}_\varepsilon$ is analytic using the dominated convergence theorem. Since for any $x \in \mathbb{R}^d$, $p_\varepsilon(x) > 0$ and $\bar{p}_\varepsilon \in \mathrm{C}(\mathbb{C}^d, \mathbb{C})$, there exists an open set $\mathsf{U} \subset \mathbb{C}^d$ such that for any $z \in \mathsf{U}$, $\Re(\bar{p}_\varepsilon(z)) > 0$. Since $\log : \mathbb{C}\backslash(\{t \in \mathbb{C} : \Re(t) \leqslant 0\}) \to \mathbb{C}$ is analytic we obtain that $z \mapsto \log \bar{p}_\varepsilon(z)$ is analytic on U. Hence, $x \mapsto \log p(y|x) + \log p_\varepsilon(x)$ is real-analytic on $\mathbb{R}^d$. We conclude using [65, Theorem 2.1]. □

The proof can be extended to the case where $Z_k = 0$ using [65, Theorem 2.1]. In this case the assumption that $\sum_{k\in\mathbb{N}} \delta_k^2 < +\infty$ can be replaced by $\lim_{k\to+\infty} \delta_k = 0$.

The following experimental section demonstrates the PnP-SGD algorithm on three canonical imaging inverse problems, namely image deblurring, inpainting, and denoising, along with other standard PnP algorithms.

for which $\nabla^2 \log p(y|\cdot)$ is constant (i.e., Gaussian models), but this is not the case because Liouville's theorem applies entire functions, which are a subclass of the real-analytic class.

4 Experimental study

In this section, we study the behaviour of several PnP algorithms for three classical inverse problems: denoising, deblurring and inpainting. We recall that in each of these problems we consider a prior model $p(x) \propto \exp[-U(x)]$ which is unknown and that the inference $x|y$ is obtained by approximation of this model. For the deblurring and denoising problems, the log-posterior of the degradation model can be written for any $x, y \in \mathbb{R}^d$ as

$$-\log p(x|y) = \|\mathbf{A}x - y\|^2/(2\sigma^2) + \alpha U(x) + C \;, \tag{36}$$

where $\mathbf{A}$ is a $d \times d$ matrix, $C \geqslant 0$ is a constant and the parameter $\alpha \geqslant 0$ balances the weights of the log-likelihood $F(x, y)$ and the log-prior U. In this case, we have for any $x, y \in \mathbb{R}^d$, $F(x, y) = \|\mathbf{A}x - y\|^2/(2\sigma^2)$. In our inpainting experiments, we change the likelihood so that pixels are either visible or hidden. In this case the log-posterior can be written for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$ as

$$-\log p(x|y) = \iota_{\mathbf{Q}x=y} + \alpha U(x) + C \;, \text{ with } \iota_{\mathsf{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathsf{C} \\ +\infty & \text{otherwise,} \end{cases} \tag{37}$$

with $\mathbf{Q}$ a $m \times d$ matrix consisting of m random lines from the $d \times d$ identity matrix.

4.1 Image dataset

In Figures 1 and 2 we present the 6 original images used in the experiments. These images contain both geometric structures, constant areas and textured regions. On the same figures, we display degraded versions of each image for each set of experiments. For the denoising experiment, the level of the Gaussian noise is fixed to $\sigma^2 = (30/255)^2$. In the case of deblurring, the operator $\mathbf{A}$ correponds to a 9×9 uniform blur operator, and we add Gaussian noise with variance $\sigma^2 = (1/255)^2$. Finally, in the context of inpainting, we hide 80% of the pixels. The dataset analysed during the current study is available from the corresponding author on reasonable request.

4.2 Algorithms

In this section, we evaluate PnP-SGD (Algorithm 1) along with three other classical PnP algorithms: PnP-ADMM (Algorithm 2), PnP-FBS (Algorithm 3) and PnP-BBS (Algorithm 4). Note that in the case of inpainting the log-likelihood is not differentiable, since ι_C is not differentiable. In Section 4.6 we will present an extension of these PnP algorithms to this setting using proximal operators.

In order to take into account the parameter $\alpha > 0$ into Algorithms 2-3-4, we slightly modify the target function. Instead of minimizing $x \mapsto -\log p(x|y)$ we aim at minimizing $x \mapsto -\log p(x|y)/\alpha$. Doing so the parameter $\alpha > 0$ can be included in the parameters of the log-likelihood which becomes $(x, y) \mapsto F(x, y)/\alpha$. All algorithms are implemented using Python and the PyTorch library. Our experiments are run on an Intel Xeon CPU E5-2609 server with an Nvidia Titan XP graphic card.

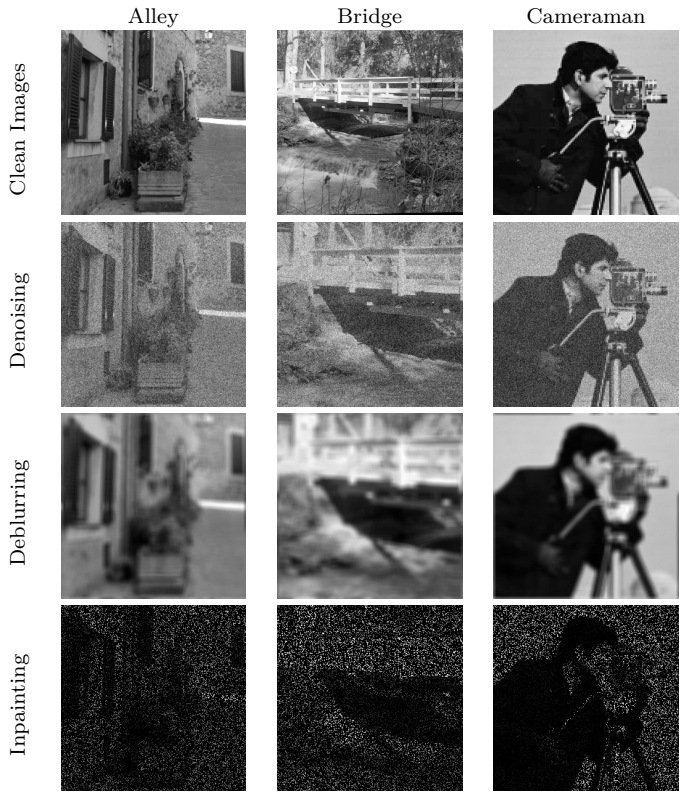


Fig. 1 *Dataset (part 1):* First three images in our dataset, and examples of degraded images for the three inverse problems considered in this paper. For denoising, we add a Gaussian noise with variance $\sigma^2 = (30/255)^2$. For deblurring, the operator $\mathbf{A}$ correponds to a 9×9 uniform blur operator, and we add Gaussian noise with variance $\sigma^2 = (1/255)^2$. For inpainting, we hide 80% of the pixels.

Algorithm 2 PnP-ADMM

Require: $n \in \mathbb{N}$, $y \in \mathbb{R}^m$, $\varepsilon > 0, \alpha > 0$, $x_0 \in \mathbb{R}^d$
  Initialization: Set $x_0 = z_0$, and $u_k = 0$.
  for $k = 0 : N$ **do**
    $x_{k+1} = \text{prox}_{(\varepsilon/\alpha)F(\cdot,y)}(z_k - u_k)$
    $z_{k+1} = D_\varepsilon(x_{k+1} + u_k)$
    $u_{k+1} = u_k + (x_{k+1} - z_{k+1})$
  end for
  return x_{N+1}

Algorithm 3 PnP-FBS

Require: $n \in \mathbb{N}$, $y \in \mathbb{R}^m$, $\varepsilon > 0$, $\alpha > 0$, $x_0 \in \mathbb{R}^d$
  for $k = 0 : N$ **do**
    $x_{k+1} = D_\varepsilon\left(x_k - (\varepsilon/\alpha)\nabla F(x_k, y)\right)$
  end for
  return x_{N+1}

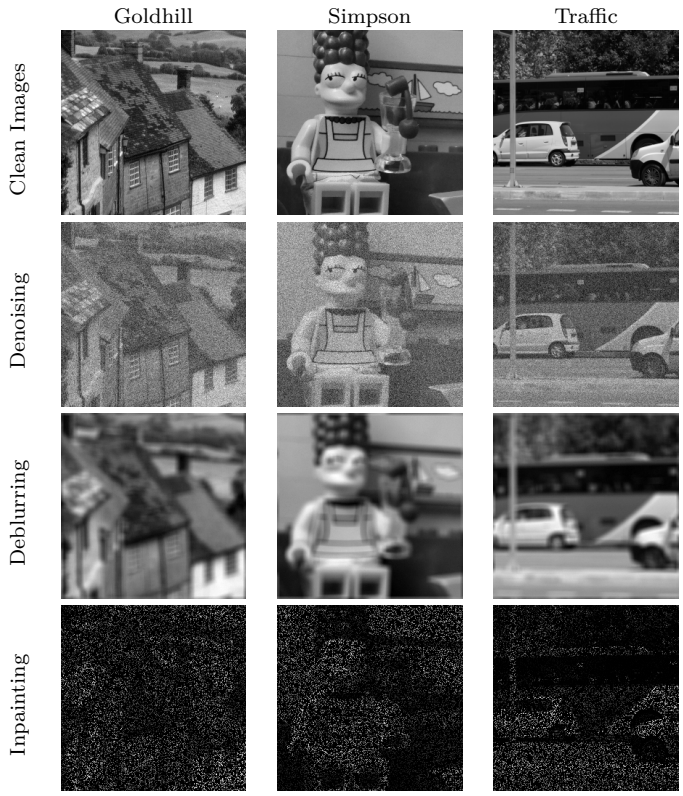


Fig. 2 *Dataset (part 2):* Last three images in our dataset, and examples of degraded images for the three inverse problems considered in this paper. For denoising, we add a Gaussian noise with variance $\sigma^2 = (30/255)^2$. For deblurring, the operator $\mathbf{A}$ correponds to a 9×9 uniform blur operator, and we add Gaussian noise with variance $\sigma^2 = (1/255)^2$. For inpainting, we hide 80% of the pixels.

Algorithm 4 PnP-BBS

Require: $n \in \mathbb{N}$, $y \in \mathbb{R}^m$, $\varepsilon > 0$, $\alpha > 0$, $x_0 \in \mathbb{R}^d$
  for $k = 0 : N$ **do**
    $x_{k+1} = D_\varepsilon(\mathrm{prox}_{(\varepsilon/\alpha)F(\cdot,y)}(x_k))$
  end for
  return x_{N+1}

4.3 Parameters settings and convergence conditions

In this section, we recall and discuss the choice of the different parameters, as well as the convergence conditions for PnP-SGD. We also discuss the convergence properties of PnP-ADMM and PnP-FBS following the guidelines of [60,70].

Recall that from (14), we denote $\mathtt{L}_y$ the Lipschitz constant of the log-likelihood gradient $x \mapsto \nabla F(., y)$. For $F(x, y) = \|\mathbf{A}x - y\|^2/(2\sigma^2)$, $\mathtt{L}_y = \|\mathbf{A}^\star\mathbf{A}\|/\sigma^2$, with $\mathbf{A}^\star$ the adjoint of $\mathbf{A}$. F is μ-strongly convex if and only if $\mathbf{A}$ is invertible, in which case $\mu = \lambda_{\min}(\mathbf{A})^2/\sigma^2$, where $\lambda_{\min}(\mathbf{A})$ is the smallest singular value of $\mathbf{A}$. In our experiments we have $\lambda_{\min} = 1$ for denoising and $\lambda_{\min} = 0$ for deblurring and inpainting. In our

experiments, the operator $\mathbf{A}$ is always chosen such that $\|\mathbf{A}^\star\mathbf{A}\| = 1$. Note that if F is replaced by F/α, as it the case in Algorithms 2-3-4, we have that $\mathtt{L}_y$ and μ are replaced by $\mathtt{L}_y/\alpha$ and μ/α.

Denoiser. In all experiments, the denoising operator D_ε is chosen as the pretrained denoising neural network introduced in [60]. This denoiser is trained so that $\mathrm{Id} - D_\varepsilon$ is $\mathtt{L}$-Lipschitz with $\mathtt{L} < 1$. Note that this corresponds to the first part of (30) in H2. In [60] three pretrained denoisers, at noise level $\varepsilon = (5/255)^2, (15/255)^2, (40/255)^2$ are proposed. In this work, we only use the first one in our denoising and deblurring experiments. The inpainting problem requires a more subtle strategy relying on a coarse to fine approach, described in Section 4.6.

PnP-SGD. In Algorithm 1, we consider a burn-in regime with a constant step δ_0 until some iteration n_{burnin}. After this initial phase, we set $(\delta_k)_{k\in\mathbb{N}}$ to be a decreasing sequence satisfying the conditions of Proposition 3. In the case of denoising or deblurring, δ_0 is given by

$$\delta_0 = \delta_{\mathtt{stable}}/6, \text{ where } \delta_{\mathtt{stable}} := 2/\mathtt{L}_{\mathrm{tot}} , \quad \mathtt{L}_{\mathrm{tot}} = \alpha\mathtt{L}/\varepsilon + \|\mathbf{A}^*\mathbf{A}\|/\sigma^2 , \tag{38}$$

where $\mathtt{L}_{\mathrm{tot}}$ is the Lipschitz constant of $\nabla \log p(.|y)$. Note that setting $\delta_0 = \delta_{\mathtt{stable}}$ ensures that the deterministic scheme: $x_0 \in \mathbb{R}^d$ and for any $k \in \mathbb{N}$, $x_{k+1} = x_k + \delta_0 \nabla \log p(x_k|y)$, satisfies that $(\log p(x_k|y))_{k\in\mathbb{N}}$ is non-decreasing. After the burn-in, we use a decreasing sequence of step-sizes $(\delta_k)_{k\in\mathbb{N}}$ such that for any $k \in \mathbb{N}$ we have

$$\delta_k := \delta_0 \times (k - n_{\mathtt{burnin}})^{-0.8} , \tag{39}$$

which satisfies the conditions required in Proposition 3 for convergence. Note that contrary to existing work, any value of $\alpha > 0$ can be used in Algorithm 1 provided that δ_0 is defined accordingly using (38).

PnP-ADMM. The convergence results of [60] for PnP-ADMM require the strong convexity of F. In our experiments, this condition is met for denoising experiments (since $\mathbf{A} = \mathrm{Id}$), but not for inpainting nor deblurring if the blur operator is not invertible (which is the case for a 9×9 uniform blur). In the denoising case, following [60], PnP-ADMM converges to a fixed point if $\mathtt{L} \in [0, 1)$ and $\mathtt{L}/(1 + \mathtt{L}(1 - 2\mathtt{L})) < \varepsilon/(\alpha\sigma^2)$. In practice, $\mathtt{L}$ and ε being set, this condition can only be satisfied for small values of the regularization parameter α, which often lead to poor-quality results. However, Algorithm 1 experimentally converges to a fixed point with interesting visual properties. This suggests that it might be possible to prove the convergence of PnP-ADMM under weaker conditions than the ones of [60].

PnP-FBS. Similarly to PnP-ADMM the convergence results obtained by [60] for PnP-FBS are only valid in a strongly convex setting. In our case this corresponds to the denoising experiment here. The condition on the Lipschitz constant of the denoiser D_ε is $\mathtt{L}/(1 + \mathtt{L}) < \varepsilon/(\alpha\sigma^2) < (\mathtt{L} + 2)/(\mathtt{L} + 1)$. In Section 4.4, we show that these conditions are not met in our experiments. In practice, we still observe convergence of the algorithm for the denoising experiments. This is no longer case in non-strongly convex problems, see Section 4.5 and Section 4.6. In [70], convergence towards the set of stationary points of the log-posterior is established for PnP-FBS provided that $D_\varepsilon = D_\varepsilon^\star$, *i.e.* D_ε is the optimal MMSE. In addition, [70] requires that

$\varepsilon \mathtt{L}_y \leqslant 1$. This condition implies that $\varepsilon \|\mathbf{A}^\star \mathbf{A}\| \leqslant \alpha\sigma^2$. Since $\|\mathbf{A}^\star \mathbf{A}\| = 1$ for all our experiments, this implies $\alpha \geqslant \varepsilon/\sigma^2$. In experiments with large noise level (as it it the case for our denoising setting), this leads to acceptable values of α. However, when σ is small in comparison to ε (which is the case for deblurring), the regularization parameter α for which the convergence is ensured is too highlighted in Section 4.3.

4.4 Denoising

We open our experimental section with an illustrative numerical experiment related to image denoising in a scenario of additive white Gaussian noise. This toy problem allows us to illustrate some differences between the Algorithms 1-4. Of course, PnP methods are not required to compute an accurate solution for the image denoising problem. A more accurate and significantly more computationally efficient approach for this problem is to scale the denoiser, as described in [70]. For completeness, we also report comparisons with that approach.

For these denoising experiments, we add a Gaussian noise of variance $\sigma^2 = (30/255)^2$ (see the second row of Figures 1 and 2 for examples of degraded images). In this experiment we use a denoiser D_ε trained for a noise level $\varepsilon = (5/255)^2$ on a dataset $\{x_i, x_i'\}_{i=1}^N$ with $x_i \sim p$ and $x_i' \sim \mathcal{N}(x_i, \varepsilon\,\mathrm{Id})$ for any $i \in \{1, \dots, N\}$. Using this denoiser in Algorithms 1-4, we aim at denoising y with noise level σ^2.

We run all algorithms for several values of the regularization parameter α and for two different initializations: first a TV-L_2 initialization, *i.e.* applying a simple TV-L_2 restoration to the noisy image following [59,15], and second an oracle initialization (using the original image without degradation). Although the noisy observation y is a natural initialization, we observed that initializing at y usually leads to unsatisfying results for all PnP schemes with a small value of ε. We believe that this arises from the high non-convexity of the problem. Our goal here is to assess the dependency of the algorithm on initialization, since the log-posterior we study is highly non-convex.

For PnP-SGD, the initial step-size δ_0 and the sequence $(\delta_k)_{k\in\mathbb{N}}$ are defined as explained in Section 4.3. For these denoising experiments, the resulting value of δ_0 is already quite small, such that decreasing δ_k after the burn-in phase effectively stops the search for a better optimum and does not change the result. The number of iterations $n_{\mathtt{burnin}}$ for the burn-in was set between 5000 and 25000 for SGD. Within that range, we stop this phase as soon as $|\mathrm{PSNR}(X_{k+1}) - \mathrm{PSNR}(X_k)| < 0.1 \times \delta_0$. This conservative choice allows to make sure that the algorithm reaches its steady state, so that the oracle initialization (starting from an overestimated value of PSNR) does not overestimate the global maximum and the non-oracle initializations (starting from an underestimated value of PSNR) do not under-estimate it. In practice, convergence is reached after a few hundreds of iterations in most cases and only rarely did the algorithm iterate beyond 5000. Increasing δ_0 to $\delta_0 = 0.9 \times \delta_{\mathtt{stable}}$ also permits to achieve faster convergence, but in this case adding a decreasing phase for $(\delta_k)_{k\in\mathbb{N}}$ after the burn-in regime is important to achieve the same asymptotic results.

For the splitting-based algorithms (ADMM, BBS, FBS), practical convergence is very fast and 100 iterations are largely sufficient in all cases. Observe that since we use a denoiser trained for a noise level $\varepsilon = (5/255)^2$, and our denoising experiments are run for $\sigma^2 = (30/255)^2$, theoretical convergence of PnP-ADMM following [60] requires that $\alpha < (1+\mathtt{L}(1-2\mathtt{L}))/36\mathtt{L}$. The exact value of $\mathtt{L}$ for the denoising considered in [60] is not available, but our experiments suggest that $\mathtt{L} \approx 1$. This implies that

only drastically small values of α meet the previous condition. As a result, this condition is not satisfied with the choices of α that are experimentally optimal but does not prevent the algorithm to converge in practice. In the same way, provided that $\mathtt{L} \in [0, 1)$, convergence of PnP-FBS following [60] implies that α is at least larger than 18, see Section 4.3. Yet, interesting values of α for this denoising experiment are far smaller The condition provided in [70], $\alpha \geqslant \varepsilon/\sigma^2 = 1/3$ gives more realistic values for α but we remind that in this case we must assume that $D_\varepsilon = D_\varepsilon^\star$.

Figure 3 summarizes the results of this denoising experiment on 10 independent random noise realizations on each of the 6 images in the dataset, for PnP-SGD, PnP-ADMM and PnP-BBS (PnP-FBS is not shown here for the sake of clarity, but it shows a very similar behavior). We first observe that initialization seems to play a very minor role for all the algorithms considered in this problem. A TV-L_2 initialization is sufficient to reach virtually the same reconstruction quality as the oracle initialization. This might be explained by the fact that denoising is a relatively simple inverse problem. Second, all algorithms produce very similar results, with an optimal value of α around 0.25, see Figure 3. Table 1 summarizes the denoising results of all algorithms (including PnP-FBS) obtained for this nearly optimal setting of $\alpha = 0.25$. In Figure 4 we display the results of the different algorithms for this denoising experiment. If the PSNR values are quite close, it seems that the algorithms make different compromises in terms of visual results. For example, the estimator obtained with PnP-ADMM seems to exhibit sharper edges. However, it also seems to hallucinate more false structures than other algorithms. This difference in estimation results is expected, as these algorithms do not actually minimize the same objective function $G(x) = F(x, y) + U(x)$. For example, for PnP-SGD, $U(x) = -\log p_\varepsilon(x)$, where p_ε results from the convolution between the true prior p and a Gaussian kernel g_ε. For PnP-ADMM, the prior potential U is related to the prior p via an inverse denoiser as shown in [33,35].

Denoising $\sigma^2 = (30/255)^2$, $\varepsilon = (5/255)^2$, TV-L_2 init, $\alpha = 0.25$					
	PnP-SGD	PnP-ADMM	PnP-BBS	PnP-FBS	Denoiser Scaling
Overall PSNR	27.65	27.37	27.65	27.56	28.04
Simpson	30.04	30.10	30.41	30.35	30.39
Traffic	27.36	27.09	27.31	27.27	27.57
Cameraman	28.54	28.21	28.74	28.48	29.17
Alley	27.16	26.82	26.98	26.96	27.49
Bridge	26.28	25.83	26.18	26.03	26.73
Goldhill	26.55	26.18	26.30	26.30	26.9

Table 1 Plug & Play denoising for $\sigma^2 = (30/255)^2$ with the prior implicit in D_ε for $\varepsilon = (5/255)^2$. D_ε is also used for the denoiser scaling experiments. This table shows mean PSNR values over K=10 independent noise realizations for each of the six images. The regularization parameter $\alpha = 0.25$ is nearly optimal for all algorithms.

Figure 5 delivers a first convergence diagnosis of PnP-SGD for the denoising task. The evolution of the average PSNR computed for each image over the 10 different experiments suggests that only a thousand of iterations seem to be needed to reach the stationary regime. This impression is confirmed by the evolution of the average gradient norm of the log-posterior, as after 1000 iterations, a plateau around 0.4 is reached. The decay after 5000 iterations is due to the forced decay of the δ_k as the

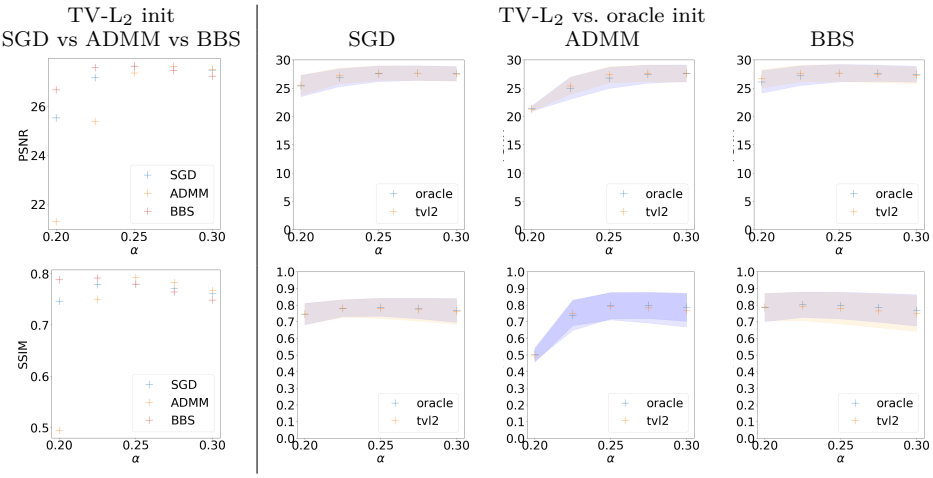


Fig. 3 Plug & Play denoising for $\sigma^2 = (30/255)^2$ with the prior implicit in D_ε for $\varepsilon = (5/255)^2$ and different values of the regularization parameter α. This table shows means and standard deviations for PSNR and SSIM values over K=10 independent noise realizations for each of the six images and different values of the regularization parameter α. The averaging is performed over the different realizations and the different images. Initialization plays a very minor role in this case and all algorithms achieve similar (nearly optimal) performance for $\alpha = 0.25$.

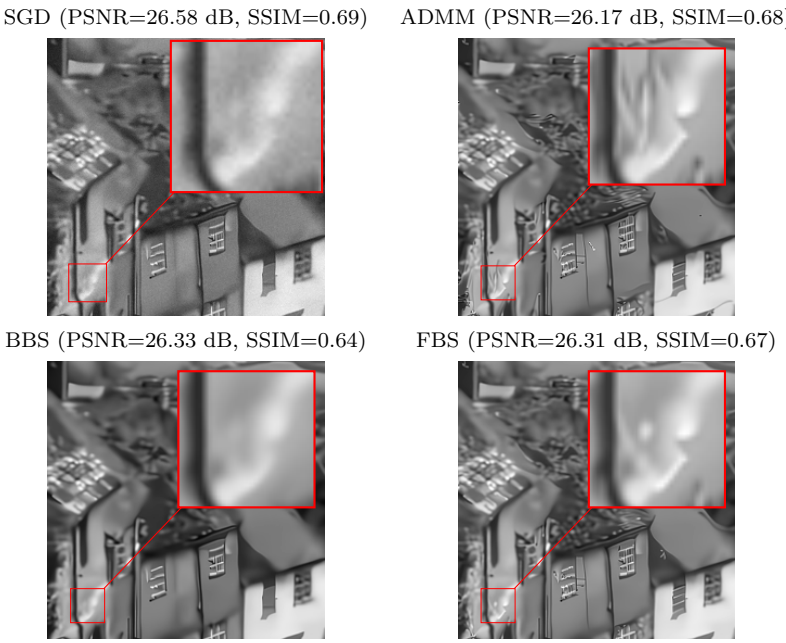
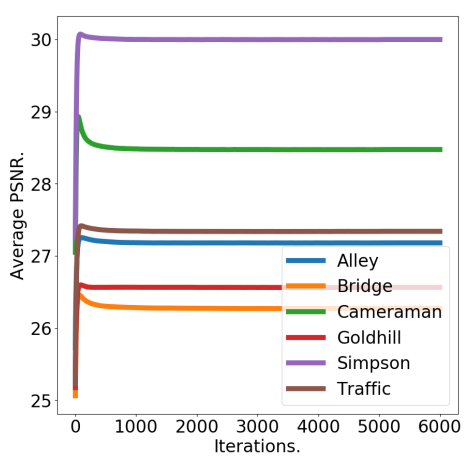
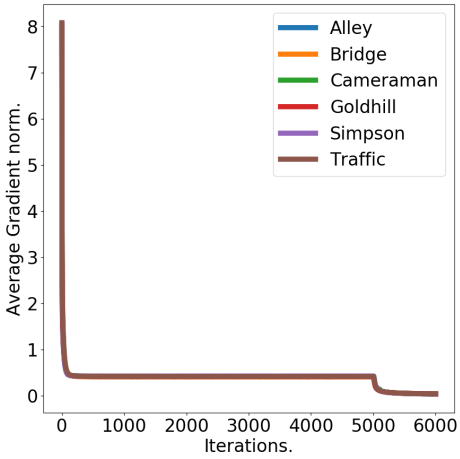


Fig. 4 Plug & Play denoising for $\sigma^2 = (30/255)^2$, $\varepsilon = (5/255)^2$ and with $\alpha = 0.25$. Although the results obtained by the different methods are close from a quantitative point of view, they look for different compromises. For example, PnP-ADMM looks for sharper edges than PnP-SGD but tends to hallucinate structures.

algorithm has left the burn-in phase during which the discretization step-size was held constant.



Evolution of the PSNR.



Evolution of the gradient norm.

Fig. 5 Convergence diagnosis for Plug & Play denoising for $\sigma^2 = (30/255)^2$, $\varepsilon = (5/255)^2$ with $\alpha = 0.25$ and $\mathrm{TV} - \mathrm{L}_2$ initialization. Left: Evolution of the average PSNR computed for $K = 10$ independent noise realizations for each image. A thousand of iterations seem to be sufficient to leave the burn-in phase and enter the stationary phase. The decay of the discretization step-size δ_k does not alter the results, which suggests that the algorithm has converged. Right: Evolution of the average gradient norm of the log-posterior computed over the 10 experiments for each image. In less than 500 iterations, it stabilizes around 0.4 for each image. These plots suggest that the algorithm has converged. The decrease observed after 5000 iterations is explained by the decay of the discretization step-size δ_k and does not alter the final result.

4.5 Deblurring

We now turn to the deblurring problem. We first present a comparison between Algorithms 1-4 for a 9×9 uniform blur in the same spirit as in Section 4.4. Following on from this, we illustrate the performance of the PnP-SGD algorithm for different blur kernels. In all the deblurring experiments, we use Gaussian noise with standard deviation $\sigma = 1/255$.

Experiments with PnP-SGD follow the same rules as for the denoising problem and the same observations are valid. When running PnP-ADMM we use approximately 200 iterations to ensure the convergence whereas for PnP-FBS and PnP-BBS, we use approximately 500 iterations. Except for PnP-SGD (using Proposition 3), these PnP algorithms are not guaranteed to converge according to [60] since $\mathbf{A}$ is not invertible. In practice PnP-FBS indeed converges only for very large values of the regularization parameter α, whereas other PnP algorithms converge for all our experiments. As highlighted in Section 4.3 this suggests that convergence for PnP-ADMM and PnP-FBS occur under weaker conditions than the ones prescribed in [60].

Figure 6 summarizes the results of deblurring on 10 independent random noise realizations on each of the 6 images in the dataset, for PnP-SGD, PnP-ADMM and PnP-BBS (PnP-FBS is not shown here because it does not converge most of the time), for TV-L_2 and oracle initializations. Again, initialization appears to play a minor role in the final results.

Observe that all algorithms show very similar performances (when they converge) for these deblurring experiments. While PnP-SGD is slower to converge, it is ensured to approximate the MAP theoretically. Table 2 summarizes the deblurring results of all algorithms (including PnP-FBS) obtained for the nearly optimal setting of $\alpha = 0.3$. In Figure 7 we display the results of the different algorithms for this deblurring experiment. Interestingly, we note that visual results for this deblurring problem are much more similar to each other than for denoising experiments.

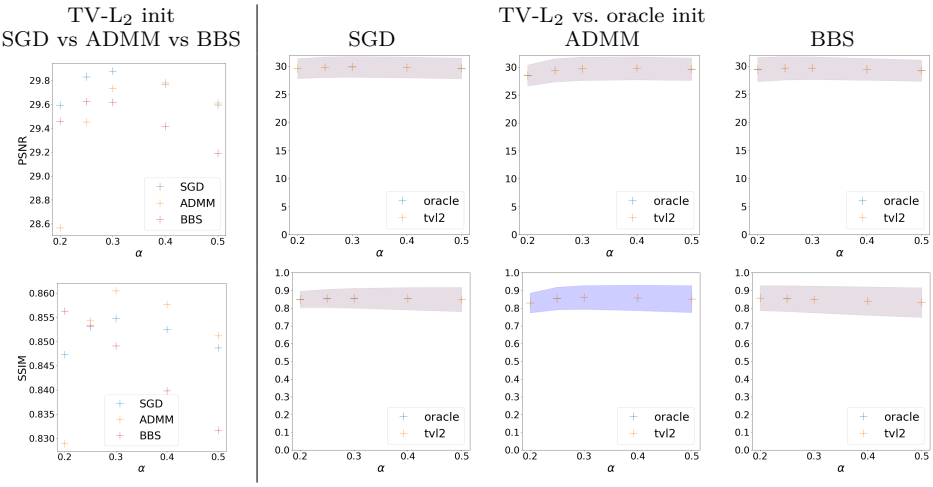


Fig. 6 Plug & Play deblurring. Image are blurred with a 9×9 uniform kernel, a Gaussian noise of standard deviation $\sigma^2 = (1/255)^2$ is added. The denoiser D_ε is trained at $\varepsilon = (5/255)^2$. The plots shows mean and standard deviation values of PSNR and SSIM over K=10 independent noise realizations for each of the six images and different values of the regularization parameter α. The averaging is performed over the different realizations and the different images. Initialization plays a very minor role in this case and all algorithms achieve similar (nearly optimal) performance for $\alpha = 0.3$, except for FBS which requires a larger (sub-optimal) α to converge.

In a manner akin to the denoising problem, we study the convergence of PnP-SGD for deblurring by computing the evolution of the average PSNR and gradient norm of the log-posterior over the iterations - see Figure 8. We observe that for most images, the algorithm requires in the order of 4 000 iterations to produce a solution with stable PSNR. Similarly, the average gradient norm of the log-posterior decreases quickly and requires in the order of 6 000 iterations to reach a plateau of approximately 0.2. The experiments with the `Cameraman` image exhibit a different behaviour, where the algorithm does not attain a stable PSNR within the first 10 000 iterations. Note that for this image, the regularization effect of stopping the algorithm early marginally improves the estimation results. This is related to the fact that, for this image, early stopping of the algorithm attenuates the mild artefact that are hallucinated by the image denoiser.

Deblurring a 9×9 kernel with $\sigma^2 = (1/255)^2$, $\varepsilon = (5/255)^2$, TV-L_2 init, $\alpha = 0.3$					
	PnP-SGD	PnP-SGD (burn-in)	PnP-ADMM	PnP-BBS	PnP-FBS
Overall PSNR	29.88	29.88	29.73	29.62	NaN
Simpsons	33.51	33.51	33.93	33.70	NaN
Traffic	29.41	29.41	29.27	29.10	NaN
Cameraman	30.68	30.68	30.43	30.39	NaN
Alley	29.26	29.27	28.99	28.90	NaN
Bridge	28.08	28.08	27.77	27.65	NaN
Goldhill	28.33	28.33	28.01	27.97	NaN

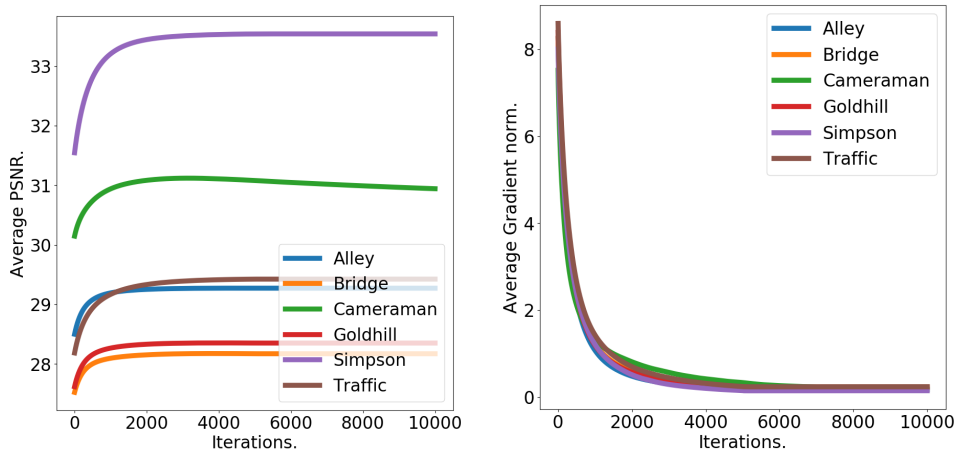
Table 2 Plug & Play deblurring. Images are blurred with a 9×9 uniform kernel, a Gaussian noise of standard deviation $\sigma = 1/255$ is added. The denoiser D_ε is trained at $\varepsilon^2 = (5/255)^2$. This table shows mean PSNR values over K=10 independent noise realizations for each of the six images for the different PnP schemes. The regularization parameter $\alpha = 0.30$ is nearly optimal for all algorithms. The post burn-in phase involving a decaying step-size does not significantly influence the results produced by using PnP-SGD.

Fig. 7 Plug & Play deblurring, for a 9×9 kernel, an additive Gaussian noise of standard deviation $\sigma^2 = (1/255)^2$, for $\varepsilon = (5/255)^2$ and for the nearly optimal value of $\alpha = 0.3$.

In addition to the previous experiments, we illustrate the performance of PnP-SGD with other blur kernels, namely the 8 real-world camera shake kernels of [41]. As previously, we consider additive white Gaussian noise of standard deviation $\sigma = 1/255$. Table 3 shows the average PSNR scores achieved by Algorithms 1-4 over K=10 independent noise realizations and over the six images introduced in Figures 1-2. The algorithms are initialized with the solution given by TV $-$ L_2 and the regularization parameter α is chosen to optimise the performance of PnP-SGD. We observe that PnP-SGD performs strongly with all blur kernels, as reflected by high PSNR scores. Similarly to denoising, we observe differences with the solution produced by using PnP-ADMM and the other PnP schemes, which do not seek to minimize the same objective function.

4.6 Inpainting

The inpainting problem consists in trying to recover $x \in \mathbb{R}^d$ from a small proportion of its pixels, namely from the measurements vector $y = \mathbf{Q}x$, where $\mathbf{Q}$ is a $m \times d$ matrix consisting of m random lines from the $d \times d$ identity matrix, and $m = qd \ll d$. In our experiments we set $q = 20\%$. In this case, since measurements are not affected by noise, the data-fitting term takes the form of a hard constraint, *i.e.* for any $x \in \mathbb{R}^d$



Evolution of the PSNR. Evolution of the gradient norm.

Fig. 8 Convergence diagnosis for Plug & Play deblurring with $\sigma^2 = (1/255)^2$, $\varepsilon = (5/255)^2$, $\alpha = 0.3$ and TV $-$ L$_2$ initialization. Left: Evolution of the average PSNR computed for K $= 10$ independent noise realizations for each of the 6 images. As expected, the convergence is slower for the deblurring problem, which has a worse conditioning than denoising. For most images, the algorithm requires in the order of 4000 iterations to produce a stable solution. The image `Cameraman` requires a larger number of iterations. Also note that, for this image, the additional regularization introduced by early stopping the algorithm marginally improves the estimation results. Right: Evolution of the average gradient norm of the log-posterior computed over the 10 experiments for each image. For all images, the gradient norm stabilizes around 0.2.

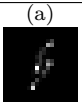
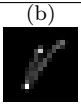
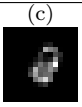
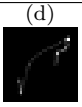
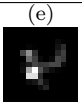
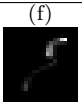
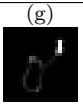
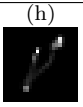
Methods	(a) 	(b) 	(c) 	(d) 	(e) 	(f) 	(g) 	(h) 
PnP-SGD	34.72	34.39	33.81	34.24	34.48	35.22	33.91	33.46
PnP-ADMM	35.59	35.18	34.38	34.95	35.11	35.93	34.47	34.12
PnP-BBS	33.9	33.79	33.37	34.01	34.03	34.62	33.40	32.98
PnP-FBS	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 3 Plug & Play deblurring. Images are blurred with the 8 real-world camera shake kernels of [41]. A Gaussian noise of standard deviation $\sigma = 1/255$ is added. The denoiser D_ε is trained at $\varepsilon^2 = (5/255)^2$. The algorithms are initialized with TV $-$ L$_2$. This table shows mean PSNR values over K=10 independent noise realizations and for the six images. The regularization parameter α is chosen to be nearly optimal for PnP-SGD and for each blur kernel.

and $y \in \mathbb{R}^m$ we have

$$F(x, y) = \iota_{\mathsf{C}_y}(x), \quad \text{where } \mathsf{C}_y = \{x : y = \mathbf{Q}x\} .$$

The non-differentiability of F is not problem when using ADMM and BBS since in this case the proximal operator of $\gamma F(\cdot, y)$ is not only defined but admits a closed-form (which is independent of $\gamma = \varepsilon/\alpha$). More precisely, we have for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$, $\operatorname{prox}_{\gamma\iota_C}(x) = \mathbf{P}^\star\mathbf{P}x + \mathbf{Q}^* y$ in terms of the $(d - m) \times d$ matrix $\mathbf{P}$ containing all the lines of the identity matrix which are not contained in $\mathbf{Q}$. However, SGD and FBS cannot be directly applied to this problem because they require F to be differentiable. Nevertheless we can apply these algorithms to an equivalent

formulation in the reduced space $\mathbb{R}^{d-m}$ of unknown pixels, as shown in the following subsection.

4.6.1 Adapting SGD to the non-differentiable inpainting problem

In what follows, we denote by $\tilde{x} := \mathbf{P}x \in \mathbb{R}^n$ the vector of $n = d - m$ unknown pixels in x. Given the unknown pixels $\tilde{x} = \mathbf{P}x$ and the measurements $y = \mathbf{Q}x$ we can reconstruct x via the affine mapping $f_y : \mathbb{R}^n \to \mathbb{R}^d$ defined for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$ by $f_y(\tilde{x}) = \mathbf{P}^*\tilde{x} + \mathbf{Q}^*y$.

The solution of the original problem $x_{\text{MAP}} = \arg\min_x F(x, y) + U(x)$ can then be written as

$$x_{\text{MAP}} = \arg\min_{x \in \mathsf{C}_y} U(x) = f_y(\arg\min_{\tilde{x}} U(f_y(\tilde{x}))) \;, \quad \tilde{x}_{\text{MAP}} = \arg\min_{\tilde{x}} U(f_y(\tilde{x})) \;, \tag{40}$$

and $\tilde{x}_{\text{MAP}}$ can be found by gradient descent on $\tilde{U} = U \circ f_y$. Using the chain rule and Tweedie's formula, we have that the gradient of $\tilde{U}$ is given for any $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$ by

$$\nabla\tilde{U}(\tilde{x}) = \mathbf{P}\nabla U(f_y(\tilde{x})) = (1/\varepsilon)\mathbf{P}(\mathrm{Id} - D_\varepsilon) \circ f_y(\tilde{x}) \;. \tag{41}$$

Finally, since the affine operators $\mathbf{P}$ and f_y are 1-Lipschitz we have that $\tilde{\mathtt{L}} \leqslant (1/\varepsilon)$, where $\tilde{\mathtt{L}}$ is the Lipschitz constant of $\nabla\tilde{U}$.

4.6.2 Parameter settings and results

The inpainting problem we consider is extremely ill-posed since 80% of the pixels are only constrained by the image prior. Since our implicit prior $p_\varepsilon(x)$ is most likely far from log-concave, the posterior shows a particularly large number of local optima. For this reason all methods are extremely sensitive to the initial condition. The initial conditions used in the previous experiments may misguide both ADMM and SGD to a wrong local optimum.

To deal with this more difficult case, we consider a different approach, combining:

- A coarse to fine scheme where we start by solving the MAP problem for large values of ε, and then use the result of this coarse MAP as an initialization for the next smaller value of ε. In our experiments we used $\varepsilon = (40/255)^2, (15/255)^2, (5/255)^2$, both for ADMM and for SGD;
- For each value of ε, a burn-in phase of 2000 iterations with $\delta_0 = 2.5\delta_{\mathtt{stable}}$, followed by a phase of 1000 decreasing steps, as defined in (39).

Table 4 summarizes the results of different algorithmic strategies to solve our inpainting problem, on our set of 6 images with $K = 4$ random realizations for each image, and Figure 9 shows an example of results on the *Simpsons* image.

We can observe in Table 4 that the coarse-to-fine scheme is beneficial to both SGD and ADMM, allowing to reach a reconstruction quality which comes very close to the oracle initialization. This benefit is also clear on the visual results shown on Figure 9. In the case of a random initialization, the coarse to fine strategy is needed to avoid the apparition of spurious geometric structure in the background. In the case of the $\mathrm{TV} - \mathrm{L}_2$ initialization, it yields better continuity in the fine black lines of the image. This holds both for ADMM and SGD.

In these inpainting experiments, we also observed that using larger initial step-sizes at the beginning and using the stochastic gradient descent instead of a simple

Method	PSNR		SSIM	
	mean	std dev	mean	std dev
Random initialization				
SGD $\varepsilon = (5/255)^2$	23.43	2.75	0.7715	0.0517
SGD $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	**26.32**	1.76	0.8074	0.0702
ADMM $\varepsilon = (5/255)^2$	19.34	3.09	0.6787	0.0629
ADMM $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	25.94	2.19	**0.8292**	0.0745
TV-L_2 initialization				
SGD $\varepsilon = (5/255)^2$	26.01	1.53	0.8042	0.0684
SGD $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	**26.34**	1.80	0.8074	0.0699
ADMM $\varepsilon = (5/255)^2$	25.38	1.74	0.8216	0.0754
ADMM $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	25.87	2.13	**0.8266**	0.0764
Oracle initialization				
SGD $\varepsilon = (5/255)^2$	**26.67**	1.66	0.8116	0.0700
SGD $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	26.36	1.76	0.8079	0.0702
ADMM $\varepsilon = (5/255)^2$	26.16	2.18	**0.8330**	0.0742
ADMM $\varepsilon = (40/255)^2$, $(15/255)^2$, $(5/255)^2$	25.93	2.14	0.8269	0.0768

Table 4 Inpainting with $p = 0.8$, $\sigma = 0$ with random, TV-L_2 and oracle initialization. Mean and standard deviation of PSNR and SSIM measures computed on K=4 random tests for each of the 6 images. Note the effectiveness of the coarse-to-fine scheme with either random or TV-L_2 initialization: Coarse to fine SGD is only 0.33 dB away from the solution obtained with oracle init, which should be quite close to the global optimum. ADMM is only 0.22 dB away from the solution obtained with oracle init.

gradient descent are important to obtain good MAP estimates. This could be explained by the non-convex nature of this problem: the stochastic term and the larger step sizes are required to avoid getting trapped in spurious local optima.

5 Conclusion

This paper studied MAP estimation in Bayesian imaging models with PnP priors defined by image denoising algorithms. We established that, under mild conditions, MAP solutions are well-posed and in a neighbourhood of the MAP solutions of the decision-theoretically optimal Bayesian model to solve the problem. For computation, we proposed a PnP-SGD optimisation method that is provably convergent under mild and realistic assumptions on the denoiser used. The proposed approach was then illustrated on a range of imaging inverse problems by using a deep network denoiser that satisfied our conditions for convergence.

In future work, we would like to continue our theoretical and empirical investigation of Bayesian PnP models, methods and algorithms. Two main priorities are to develop provably convergent accelerated algorithms, and to develop methods to automatically adjust the regularisation parameter α directly from the observed data y, in a manner akin to [69]. It would also be interesting to extend the decision-theoretic foundation of MAP estimation in log-concave models of [52] to encompass MAP estimation in (not log-concave) PnP Bayesian models.

Acknowledgements VDB was partially supported by EPSRC grant EP/R034710/1. RL was partially supported by grants from Région Ile-De-France. AD acknowledges support of the Lagrange Mathematical and Computing Research Center. MP was partially supported by the EPSRC grants EP/T007346/1 and EP/W007681/1. JD and AA acknowledge support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01).

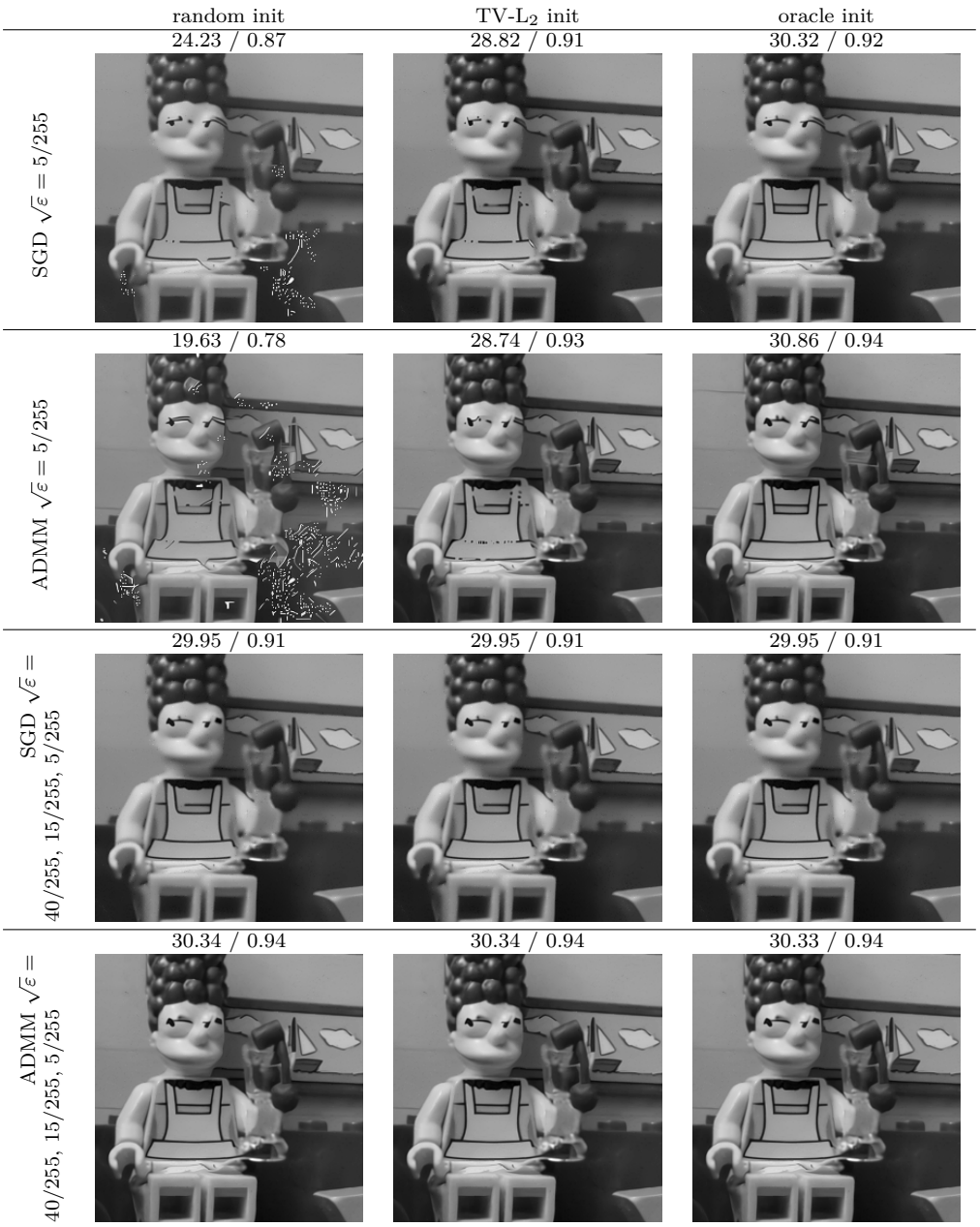


Fig. 9 Inpainting results for the Simpson's image with $p = 0.8$, $\sigma = 0$ each column corresponds to a different initial condition.

Computer experiments for this work ran on a Titan Xp GPU donated by NVIDIA, as well as on HPC resources from GENCI-IDRIS (Grants 2020-AD011011641 and 2021-AD011011641R1).

References

1. Introduction to markov random fields. In: Markov Random Fields for Vision and Image Processing. The MIT Press (2011). DOI 10.7551/mitpress/8579.003.0001. URL https://doi.org/10.7551/mitpress/8579.003.0001
2. Ahmad, R., Bouman, C.A., Buzzard, G.T., Chan, S., Liu, S., Reehorst, E.T., Schniter, P.: Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery. IEEE signal processing magazine **37**(1), 105–116 (2020)
3. Arridge, S., Maass, P., Öktem, O., Schönlieb, C.B.: Solving inverse problems using data-driven models. Acta Numerica **28**, 1–174 (2019). DOI 10.1017/S0962492919000059
4. Bach, F.: Breaking the curse of dimensionality with convex neural networks. The Journal of Machine Learning Research **18**(1), 629–681 (2017)
5. Bauschke, H.H., Combettes, P.L., et al.: Convex analysis and monotone operator theory in Hilbert spaces, vol. 408. Springer (2011)
6. Bernardo, J., Smith, A.: Bayesian Theory, vol. 15 (2000). DOI 10.2307/2983298
7. Bottou, L., Curtis, F.E., Nocedal, J.: Optimization methods for large-scale machine learning. Siam Review **60**(2), 223–311 (2018)
8. Boyd, S., Parikh, N., Chu, E.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Now Publishers Inc (2011)
9. Brandière, O., Duflo, M.: Les algorithmes stochastiques contournent-ils les pièges? Ann. Inst. H. Poincaré Probab. Statist. **32**(3), 395–427 (1996)
10. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, pp. 60–65. IEEE (2005)
11. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation **4**(2), 490–530 (2005)
12. Bubeck, S.: Convex optimization: Algorithms and complexity. arXiv preprint arXiv:1405.4980 (2014)
13. Buzzard, G.T., Chan, S.H., Sreehari, S., Bouman, C.A.: Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium. SIAM Journal on Imaging Sciences **11**(3), 2001–2020 (2018)
14. Chambolle, A.: An algorithm for total variation minimization and applications. Journal of Mathematical Imaging and Vision **20**, 89–97 (2004). DOI 10.1023/B:JMIV.0000011325.36760.1e
15. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of mathematical imaging and vision **40**(1), 120–145 (2011)
16. Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play admm for image restoration: Fixed-point convergence and applications. IEEE Transactions on Computational Imaging **3**(1), 84–98 (2017). DOI 10.1109/TCI.2016.2629286
17. Chen, Y., Pock, T.: Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(6), 1256–1272 (2017). DOI 10.1109/TPAMI.2016.2596743
18. Cohen, R., Elad, M., Milanfar, P.: Regularization by denoising via fixed-point projection (red-pro) (2020)
19. Combettes, P.L., Pesquet, J.C.: Proximal splitting methods in signal processing. In: Fixed-point algorithms for inverse problems in science and engineering, pp. 185–212. Springer (2011)
20. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3d filtering. In: Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, vol. 6064, p. 606414. International Society for Optics and Photonics (2006)
21. Debarnot, V., Weiss, P.: Deepblur: Blind identification of space variant psf. In: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), pp. 1544–1547 (2021). DOI 10.1109/ISBI48211.2021.9433857

22. Delyon, B.: General results on the convergence of stochastic algorithms. IEEE Transactions on Automatic Control **41**(9), 1245–1255 (1996)
23. Delyon, B., Lavielle, M., Moulines, E.: Convergence of a stochastic approximation version of the em algorithm. Annals of statistics pp. 94–128 (1999)
24. Diamond, S., Sitzmann, V., Heide, F., Wetzstein, G.: Unrolled optimization with deep priors (2017)
25. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: European conference on computer vision, pp. 184–199. Springer (2014)
26. Donoho, D.L.: De-noising by soft-thresholding. IEEE transactions on information theory **41**(3), 613–627 (1995)
27. Durmus, A., Moulines, E., Pereyra, M.: Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau. SIAM Journal on Imaging Sciences **11**(1), 473–506 (2018)
28. Efron, B.: Tweedie's formula and selection bias. Journal of the American Statistical Association **106**(496), 1602–1614 (2011)
29. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3848–3856 (2019)
30. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG) **35**(6), 191 (2016)
31. Gilton, D., Ongie, G., Willett, R.: Neumann networks for inverse problems in imaging (2019)
32. Gregor, K., LeCun, Y.: Learning fast approximations of sparse coding. In: Proceedings of the 27th International Conference on International Conference on Machine Learning, pp. 399–406. Omnipress (2010)
33. Gribonval, R.: Should penalized least squares regression be interpreted as maximum a posteriori estimation? IEEE Transactions on Signal Processing **59**(5), 2405–2410 (2011)
34. Hurault, S., Leclaire, A., Papadakis, N.: Gradient Step Denoiser for convergent Plug-and-Play. In: (ICLR) International Conference on Learning Representations (2022). URL `http://arxiv.org/abs/2110.03220`
35. Hurault, S., Leclaire, A., Papadakis, N.: Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization (2) (2022). URL `http://arxiv.org/abs/2201.13256`
36. Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. IEEE Signal Processing Letters **24**(12), 1872–1876 (2017)
37. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
38. Laumont, R., de Bortoli, V., Almansa, A., Delon, J., Durmus, A., Pereyra, M.: Bayesian imaging using plug & play priors: when langevin meets tweedie (2021)
39. Lebrun, M., Buades, A., Morel, J.M.: A nonlocal bayesian image denoising algorithm. SIAM Journal on Imaging Sciences **6**(3), 1665–1688 (2013)
40. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189 (2018)
41. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1964–1971. IEEE (2009)
42. Louchet, C., Moisan, L.: Posterior expectation of the total variation model: Properties and experiments. SIAM Journal on Imaging Sciences **6**(4), 2640–2684 (2013). DOI 10.1137/120902276
43. Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: (ICCV) International Conference on Computer Vision, pp. 1781–1790 (2017). DOI 10.1109/ICCV.2017.198
44. Metivier, M., Priouret, P.: Applications of a kushner and clark lemma to general classes of stochastic algorithms. IEEE Transactions on Information Theory **30**(2), 140–151 (1984)
45. Milanfar, P.: Symmetrizing Smoothing Filters. SIAM Journal on Imaging Sciences **6**(1), 263–284 (2013). DOI 10.1137/120875843
46. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)

47. Munkres, J.R.: Topology (2000)
48. Nair, P., Gavaskar, R.G., Chaudhury, K.N.: Fixed-point and objective convergence of plug-and-play algorithms. IEEE Transactions on Computational Imaging **7**, 337–348 (2021)
49. Nesterov, Y.: Lectures on convex optimization, *Springer Optimization and Its Applications*, vol. 137. Springer, Cham (2018). DOI 10.1007/978-3-319-91578-4. URL https://doi.org/10.1007/978-3-319-91578-4. Second edition of [MR2142598]
50. Parikh, N., Boyd, S.: Proximal algorithms. Foundations and Trends in optimization **1**(3), 127–239 (2014)
51. Pereyra, M.: Maximum-a-posteriori estimation with bayesian confidence regions. SIAM Journal on Imaging Sciences **10**(1), 285–302 (2017). DOI 10.1137/16M1071249. URL https://doi.org/10.1137/16M1071249
52. Pereyra, M.: Revisiting Maximum-a-Posteriori estimation in log-concave models. SIAM Journal on Imaging Sciences **12**(1), 650–670 (2019)
53. Pereyra, M., Vargas Mieles, L., Zygalakis, K.C.: Accelerating proximal Markov chain Monte Carlo by using an explicit stabilized method. SIAM J. Imaging Sci. **13**(2), 905–935 (2020). DOI 10.1137/19M1283719. URL https://doi.org/10.1137/19M1283719
54. Pesquet, J.C., Repetti, A., Terris, M., Wiaux, Y.: Learning maximally monotone operators for image recovery (2020)
55. Reehorst, E.T., Schniter, P.: Regularization by denoising: Clarifications and new interpretations. IEEE Transactions on Computational Imaging **5**(1), 52–67 (2018). DOI 10.1109/TCI.2018.2880326
56. Repetti, A., Pereyra, M., Wiaux, Y.: Scalable bayesian uncertainty quantification in imaging inverse problems via convex optimization. SIAM Journal on Imaging Sciences **12**(1), 87–118 (2019). DOI 10.1137/18M1173629
57. Robert, C.: The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation. Springer Texts in Statistics. Springer New York (2007). URL https://books.google.fr/books?id=6oQ4s8Pq9pYC
58. Romano, Y., Elad, M., Milanfar, P.: The little engine that could: Regularization by denoising (red). SIAM Journal on Imaging Sciences **10**(4), 1804–1844 (2017)
59. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena **60**(1-4), 259–268 (1992). DOI 10.1016/0167-2789(92)90242-F
60. Ryu, E.K., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-play methods provably converge with properly trained denoisers. In: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pp. 5546–5557 (2019). URL http://proceedings.mlr.press/v97/ryu19a.html
61. Schwartz, E., Giryes, R., Bronstein, A.M.: Deepisp: Toward learning an end-to-end image processing pipeline. IEEE Transactions on Image Processing **28**(2), 912–923 (2018)
62. Sreehari, S., Venkatakrishnan, S.V., Wohlberg, B., Buzzard, G.T., Drummy, L.F., Simmons, J.P., Bouman, C.A.: Plug-and-Play Priors for Bright Field Electron Tomography and Sparse Interpolation. IEEE Transactions on Computational Imaging **2**(4), 1–1 (2016). DOI 10.1109/TCI.2016.2599778
63. Sun, Y., Wohlberg, B., Kamilov, U.S.: An online plug-and-play algorithm for regularized image reconstruction. IEEE Transactions on Computational Imaging (2019)
64. Sun, Y., Wu, Z., Wohlberg, B., Kamilov, U.S.: Scalable plug-and-play admm with convergence guarantees. arXiv preprint arXiv:2006.03224 (2020)
65. Tadić, V.B., Doucet, A., et al.: Asymptotic bias of stochastic gradient search. The Annals of Applied Probability **27**(6), 3255–3304 (2017)
66. Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.: A convergent image fusion algorithm using scene-adapted gaussian-mixture-based denoising. IEEE Transactions on Image Processing **28**(1), 451–463 (2018)
67. Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.T.: Scene-Adapted Plug-and-Play Algorithm with Guaranteed Convergence: Applications to Data Fusion in Imaging (2018)
68. Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing, pp. 945–948. IEEE (2013)
69. Vidal, A.F., De Bortoli, V., Pereyra, M., Durmus, A.: Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach part i: Methodology and experiments. SIAM Journal on Imaging Sciences **13**(4), 1945–1989 (2020)

70. Xu, X., Sun, Y., Liu, J., Wohlberg, B., Kamilov, U.S.: Provable Convergence of Plug-and-Play Priors with MMSE denoisers (4), 1–10 (2020). URL http://arxiv.org/abs/2005.07685
71. Yu, G., Sapiro, G., Mallat, S.: Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. IEEE Transactions on Image Processing **21**(5), 2481–2499 (2011)
72. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
73. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning Deep CNN Denoiser Prior for Image Restoration. In: (CVPR) IEEE Conference on Computer Vision and Pattern Recognition, pp. 2808–2817. IEEE (2017). DOI 10.1109/CVPR.2017.300
74. Zhang, K., Zuo, W., Zhang, L.: Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018)
75. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: 2011 International Conference on Computer Vision, pp. 479–486. IEEE (2011). DOI 10.1109/ICCV.2011.6126278. URL http://people.csail.mit.edu/danielzoran/EPLLICCVCameraReady.pdf
76. Zou, F., Shen, L., Jie, Z., Zhang, W., Liu, W.: A sufficient condition for convergences of adam and rmsprop. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11127–11135 (2019)