Comparing Plug-and-Play and Unrolled networks

Minh Hai Nguyen, Pierre Weiss

Comparing Plug-and-Play and Unrolled networks

Minh Hai Nguyen and Pierre Weiss

***Abstract*—Plug-and-play and unrolled methods are among the most popular and efficient approaches for solving inverse problems in imaging. In this paper, we review their similarities and differences. We adopt a Bayesian framework to explain their statistical properties. This allows us to relate plug-and-play methods to Maximum A Posteriori (MAP) estimators and unrolled networks to Minimum Mean Square Error (MMSE) estimators. We clarify some of their properties including their stability and numerical efficiency. Overall, this paper provides a concise review of their respective advantages and limitations.**

***Index Terms*—MAP, MMSE, plug-and-play, unrolled methods, inverse problems, Bayesian imaging.**

I. Introduction

Since the 2010s, a revolution has occurred in solving inverse problems in imaging. Regularization terms have evolved from being conceived through reflection and analysis to using deep neural network trained from data. The characterization of solution properties has arguably lost in subtlety, but the reconstruction performance improved significantly. In major reconstruction challenges, the current state-of-the-art methods incorporate the physics of the problem [1], [2], [3]. In particular, the Plug-and-Play (PnP) methods [4], [5] and the unrolled networks [6], [7] are informed by forward models through data fidelity terms. These two algorithms are widely used, but there seems to be some confusion about their similarities and differences. Here, we aim to contribute to improving the discernment between these methods from a statistical perspective by showing their relationship to the Maximum A Posteriori (MAP) estimator and to the Minimum Mean Square Error (MMSE) estimator. After a brief description, we provide a concise review of their respective strengths and limitations. Most of the results presented here are already known, and we refer the interested readers to the excellent review articles [8], [9], [10] for additional insights. This paper is intended as a self-contained and concise review, enabling researchers and practitioners to quickly understand the key features of PnP and unrolled algorithms for image reconstruction, and to select the most suitable approach for their specific application.

II. Preliminaries and definitions

Most inverse problems consist in estimating an unknown signal $x \in \mathcal{X}$, given an observation $y \in \mathcal{Y}$ under the forward model $y = \mathcal{P}(Ax)$, where $\mathcal{P}$ is perturbation. The operator $A : \mathcal{X} \to \mathcal{Y}$ models the acquisition device. We assume that it is known in this paper. The perturbation $\mathcal{P}$ can be deterministic (e.g. quantization) or stochastic (e.g. additive white Gaussian noise). The most frequent application case is the finite dimensional setting, with $\mathcal{X} = \mathbb{R}^N$ and $\mathcal{Y} = \mathbb{R}^M$.

Minh Hai Nguyen is with Institut de Recherche en Informatique de Toulouse and Centre de Biologie Intégrative, Université de Toulouse, France, (e-mail: minh-hai.nguyen@univ-tlse3.fr).

Pierre Weiss is with Institut de Recherche en Informatique de Toulouse and Centre de Biologie Intégrative, Laboratoire MCD, CNRS & Université de Toulouse, France (e-mail: pierre.armand.weiss@gmail.com).

A. MAP and MMSE definitions

In ill-posed inverse problems, the measurement y does not carry all the information on x: some information is lost through the sensing operator A, as clearly exemplified by the inpainting problem in Fig. 1. Incorporating information on x can be achieved using a Bayesian formalism: we assume that the vector x is a realization of a random vector $\boldsymbol{x}$ with *prior* probability distribution $p_{\boldsymbol{x}}$. We can introduce the random vector $\boldsymbol{y}$ that relates to $\boldsymbol{x}$ by the forward model $\boldsymbol{y} = \mathcal{P}(A\boldsymbol{x})$. Under this assumption, the *posterior distribution* $p_{\boldsymbol{x}|\boldsymbol{y}}$ is obtained by the Bayes's formula $p_{\boldsymbol{x}|\boldsymbol{y}}(x|y) = \frac{p_{\boldsymbol{y}|\boldsymbol{x}}(y|x)p_{\boldsymbol{x}}(x)}{p_{\boldsymbol{y}}(y)}$. In the particular case of additive white Gaussian noise, $\mathcal{P}(\boldsymbol{z}) = \boldsymbol{z} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \sigma^2 \mathrm{Id}\right)$ independent of $\boldsymbol{x}$, we obtain:

$$p_{\boldsymbol{x}|\boldsymbol{y}}(x|y) \propto \exp\left(-\frac{1}{2\sigma^2}\left\|Ax - y\right\|^2\right) \cdot p_{\boldsymbol{x}}(x)$$

Two estimators based on the posterior distribution play a central role in understanding PnP and unrolled methods: the Maximum A Posteriori (MAP) and the Minimum Mean Square Error (MMSE). They are both defined based on different optimality criteria: the MAP is the best point estimator, meanwhile the MMSE is the best estimator in average.

Definition 1 (MAP estimator)**.** *The MAP estimator is the mapping that yields the* most likely *signal given the observation:* $\hat{x}_{\mathit{MAP}} : y \in \mathcal{Y} \mapsto \operatorname*{argmax}_{x \in \mathcal{X}} p_{\boldsymbol{x}|\boldsymbol{y}}(x|y)$*. For additive white Gaussian noise, taking the negative log-posterior yields:*

$$\hat{x}_{\mathit{MAP}}(y) \overset{\text{def}}{=} \operatorname*{argmin}_{x \in \mathcal{X}} f(x) + g(x) \tag{1}$$

with $f(x) = \frac{1}{2}\left\|Ax - y\right\|^2$ *and* $g(x) = -\sigma^2 \log p_{\boldsymbol{x}}(x)$*.*

Definition 2 (MMSE estimator)**.** *The MMSE estimator is the mapping that yields the best signal* in average *given the observation* y*. It is defined as:*

$$\hat{x}_{\mathit{MMSE}} \overset{\text{def}}{=} \operatorname*{argmin}_{\phi:\mathcal{Y}\to\mathcal{X}\ \mathit{measurable}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[\left\|\phi(\boldsymbol{y}) - \boldsymbol{x}\right\|_2^2\right] \tag{2}$$

Proposition 1 (MMSE and posterior mean)**.** *For almost all* $y \in \mathcal{Y}$*, the MMSE is equal to the posterior mean [13]:*

$$\hat{x}_{\mathit{MMSE}}(y) = \mathbb{E}\left[\boldsymbol{x}|\boldsymbol{y} = y\right] = \int_{\mathcal{X}} x \cdot p_{\boldsymbol{x}|\boldsymbol{y}}(x|y)dx \tag{3}$$

B. PnP and Unrolled networks

Understanding PnP and unrolled methods requires some basic notions of first-order optimization recalled below.

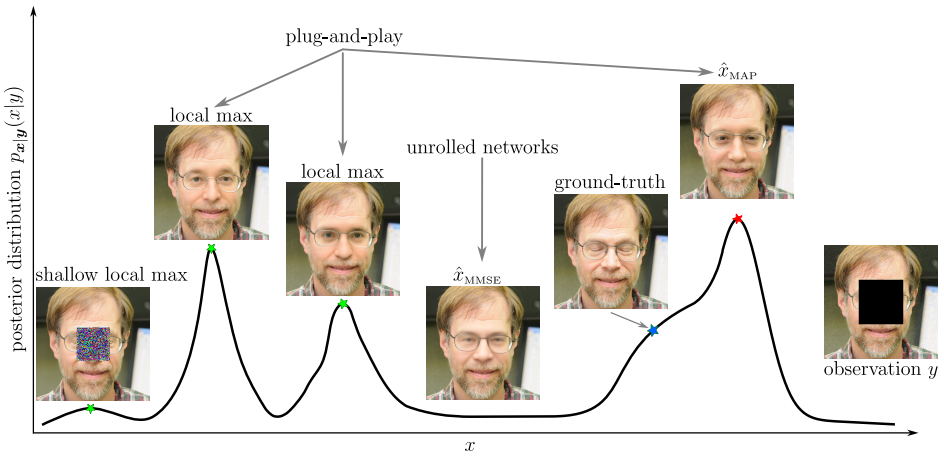

Fig. 1: Example of inpainting on FFHQ dataset. PnP algorithms yield local maximizers of the posterior, which are nice looking if well initialized, but potentially misleading. Unrolled networks are MMSE estimators approximations, which are blurry where uncertain and do not necessarily have a high posterior probability. PnP algorithms can be trapped in shallow local maxima. To overcome this, a stochastic version of the PnP algorithm using diffusion models as priors [11], [12] is used to initialize posterior samples and avoid these traps. Subsequently, 500 gradient steps are performed to minimize the negative log-posterior (1).

1) First order splitting methods: A simple way to find the MAP solution is to use an explicit gradient descent : $x_{k+1} = x_k - \tau_k \left(\nabla f(x_k) + \nabla g(x_k)\right)$. This solution often imposes severe restrictions on the step-size τ_k, resulting in slow algorithms. To counter this, it is often beneficial to consider implicit methods which can be defined through *proximal operators*. Given a function $h : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$, it is defined for $x_0 \in \mathcal{X}$ by:

$$\operatorname{prox}_h(x_0) \overset{\text{def}}{=} \operatorname*{argmin}_{x \in \mathcal{X}} h(x) + \frac{1}{2}\|x - x_0\|_2^2.$$

Proximal operators play a pivotal role in optimization, because they enable implicit gradient descent methods. This is shown by the following equivalence, valid for any lower semi-continuous function h and step-size $\tau > 0$:

$$x_{k+1} = x_k - \tau \nabla h(x_{k+1}) \Leftrightarrow x_{k+1} = \operatorname{prox}_{\tau h}(x_k).$$

It is also valid for non-differentiable functions by using the sub-differential. For composite optimization problems involving the sum of two terms as in (1), mixing explicit and implicit gradient steps yields a variety of optimization algorithms [14]. We illustrate this point in Tab. I, where we detail the Gradient Descent (GD), the Proximal Gradient Descent (PGD) and the Douglas-Rachford algorithm. The following proposition summarizes the main conditions for convergence.

Proposition 2 (Informal convergence results, [15], [16])**.** *The methods in Tab. I converge to critical points of $f + g$ if f and g are sufficiently "regular" and τ_k is sufficiently small.*

2) Denoisers as implicit priors: Assume that we wish to minimize $f + g$ in (1), where $g = -\sigma^2 \log p_{\boldsymbol{x}}$. Applying the previous optimization schemes requires computing ∇g or $\operatorname{prox}_{\tau g}$. The main idea of PnP methods is to relate these mappings to trained image denoisers. To this end, consider a denoising problem $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \eta^2 \mathrm{Id}\right)$. The following propositions are key for understanding PnP methods.

Proposition 3 (MAP denoiser and the proximal operator)**.** *The MAP denoiser D_η^{MAP} is* equal *to the proximal operator of the negative-log-prior:*

$$\begin{aligned} D_\eta^{MAP}(y) &\overset{\text{def}}{=} \operatorname*{argmin}_{x \in \mathcal{X}} \frac{1}{2\eta^2}\|x - y\|_2^2 - \log p_{\boldsymbol{x}}(x) \\ &= \operatorname{prox}_{-\eta^2 \log p_{\boldsymbol{x}}}(y). \end{aligned}$$

The relationship between the prior and the MMSE denoiser is more subtle. Let D_η^{MMSE} denote the MMSE denoiser and $p_\eta = p_{\boldsymbol{x}} * G_\eta$ denote a Gaussian smoothed version of the log-prior, where G_η is a centered Gaussian function of variance η^2. Observe that p_η converges weakly to $p_{\boldsymbol{x}}$ as $\eta \to 0$ since G_η tends to a Dirac mass. The following formula is often referred to as Tweedie's formula [17], [18].

Proposition 4 (MMSE denoiser and the prior's gradient)**.** *Let $g_\eta \overset{\text{def}}{=} -\sigma^2 \log\left(p_\eta\right)$. We have $\nabla g_\eta = \frac{\sigma^2}{\eta^2}\left(\mathrm{Id} - D_\eta^{MMSE}\right)$.*

Denoiser as implicit prior: The MMSE denoiser enables explicit gradient descent steps on a smoothed version of $-\log p_{\boldsymbol{x}}$, meanwhile the MAP denoiser performs an implicit gradient descent step on $-\log p_{\boldsymbol{x}}$.

3) PnP definition: PnP methods [4], [5], [19], [20], [21], [22] can be defined as first order methods. They perform implicit or explicit gradient steps (see Tab. I) on the prior term using an off-the-shelf denoisers $D : \mathcal{X} \to \mathcal{X}$. That is, $\operatorname{prox}_{\tau g}$ is replaced by D, or ∇g is replaced by a term of the form $\frac{\sigma^2}{\eta^2}\left(\mathrm{Id} - D\right)$, following Prop. 4.

Remark II.1. *PnP methods that perform explicit gradient steps on the prior were first referred to as Regularization-by-Denoising (RED) methods [5], [19], [20].*

Example 1. *Following Tab. I, an implicit PnP PGD method is $x_{k+1} = D_{\tau_k}^{MAP}\left(x_k - \tau_k \nabla f(x_k)\right)$. An explicit PnP PGD method writes $x_{k+1} = \operatorname{prox}_{\tau_k f}\left(x_k - \tau_k \nabla g_\eta(x_k)\right)$, see Prop. 4 for evaluating ∇g_η.*

4) Unrolled network definition: The unrolling method [23], [9] has exactly the same structure as PnP methods. The difference lies in the way the denoisers are trained. Consider any of the iterative schemes in Tab. I. The term $\nabla g : \mathcal{X} \to \mathcal{X}$ or $\operatorname{prox}_{\tau g} : \mathcal{X} \to \mathcal{X}$ arising at the k-th iteration can be

TABLE I: Examples of first order algorithms.

		Explicit in the prior		*Implicit in the prior*
Explicit in f	(GD)	$x_{k+1} = x_k - \tau_k(\nabla f(x_k) + \nabla g(x_k))$	(PGD)	$x_{k+1} = \operatorname{prox}_{\tau_k g}(x_k - \tau_k \nabla f(x_k))$
Implicit in f	(PGD)	$x_{k+1} = \operatorname{prox}_{\tau_k f}(x_k - \tau_k \nabla g(x_k))$	(Douglas-Rachford)	$x_{k+1} = x_k + \operatorname{prox}_g(2\operatorname{prox}_f(x_k) - x_k) - \operatorname{prox}_f(x_k)$

replaced by a denoiser $D_{\theta_k} : \mathcal{X} \to \mathcal{X}$, where θ_k denotes trainable weights. Running the scheme for $K \in \mathbb{N}^*$ iterations results in a reconstruction mapping of the form:

$$T_\theta(y) = T_K \circ T_{K-1} \circ \cdots \circ T_1(x_0(y)), \tag{4}$$

where $T_k : \mathcal{X} \to \mathcal{X}$ denotes the $k-$th iterate operator. It depends on the operator A through the data-term f and on the denoiser D_{θ_k}. The initial guess $x_0(y)$ can be chosen as a regularized inverse for instance. Note that the denoisers D_{θ_k} can vary from one iteration to the next. Thus, unrolled methods define a complex neural network architecture T_θ specifically designed for a given operator A.

III. PnP and the MAP

Assume an ideal situation where both D_τ^{MAP} and D_η^{MMSE} can be computed exactly. By construction, PnP methods are designed to maximize the posterior distribution $p_{\boldsymbol{x}|\boldsymbol{y}}$, or rather to minimize $-\log p_{\boldsymbol{x}|\boldsymbol{y}}$. Gathering Prop. 2, 3 and 4, we obtain the following guarantees [24], [20], [21].

Proposition 5 (Limit points of PnP methods)**.** *Under regularity and step-size assumptions not precised here:*

- *Implicit PnP methods with a MAP denoiser converge to critical points of the posterior.*
- *Explicit PnP methods with a MMSE denoiser converge to critical points of the posterior, with a smooth prior* $p_{\boldsymbol{x}} \star G_\eta$.
- *Implicit PnP methods with an MMSE denoiser still converge to critical points of a posterior with a different prior.*
- *The convergence is global whenever* f *and* g *are convex.*

A. Designing approximate denoisers

In practice, except for few analytic priors, neither of the theoretical denoisers D_τ^{MAP} nor D_η^{MMSE} are explicitly available. Before the advent of deep-learning, computing a MAP estimator (optimization) was considered easier than an MMSE estimator (integration), at least for convex priors. The situation is arguably reversed when dealing with nonconvex priors learned through data.

a) MAP denoiser: If the prior is log-concave, it is possible to evaluate D_τ^{MAP} using iterative methods. This is the case for handcrafted priors such as total variation, or some learned priors [25]. In general however, computing the global minimizer of a nonconvex prior is not possible, making D_τ^{MAP} inaccessible.

b) MMSE denoiser: Given a dataset of images $(x_i)_{1 \le i \le I}$, we can synthesize noisy images $y_i = x_i + \epsilon_i$, where the ϵ_i's are independent realizations of the random vector $\boldsymbol{\epsilon} \sim \mathcal{N}\left(0, \eta^2 \mathrm{Id}\right)$. Consider a parameterized family of estimators $(D_\theta)_{\theta \in \Theta}$, where θ is a parameter and $D_\theta : \mathcal{X} \to \mathcal{X}$ is a denoiser. In Fig. 1, D_θ was chosen as a convolutional neural network. Finding the best estimator in the family can be achieved by minimizing an empirical risk with a stochastic gradient algorithm [26]. This yields an optimal estimator $D_{\theta^\star}$, approximating the MMSE [27].

Claim 1 (MMSE approximation)**.** *Under some conditions, the trained neural network approximates the MMSE denoiser:*

$$D_{\theta^\star} \approx D_\eta^{\text{MMSE}}.$$

This can be shown by observing that:

$$\begin{aligned} D_{\theta^\star} &\approx \operatorname*{argmin}_{\theta, \phi = D_\theta} \frac{1}{I} \sum_{i=1}^{I} \|\phi(y_i) - x_i\|_2^2 \\ &\approx \operatorname*{argmin}_{\phi \text{ measurable}} \frac{1}{I} \sum_{i=1}^{I} \|\phi(y_i) - x_i\|_2^2 \\ &\approx \operatorname*{argmin}_{\phi \text{ measurable}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \left[\|\phi(\boldsymbol{y}) - \boldsymbol{x}\|_2^2\right] \overset{\text{def}}{=} D_\eta^{\text{MMSE}}. \end{aligned}$$

Here, we successively used the facts that i) we can solve the empirical risk minimization problem with sufficient precision, ii) the parameterized family of functions $(D_\theta)_{\theta \in \Theta}$ is sufficiently expressive to approximate the MMSE mapping, iii) the empirical risk converges to the average risk for a large dataset and iv) the mapping T_θ is able to generalize to unseen data. All those points may seem highly questionable, but empirical evidence and recent advances [28] indicate that they may hold true to some extent. Therefore, with learning techniques, data-driven MMSE denoisers can be effectively approximated.

c) MMSE denoiser for implicit methods: The MMSE denoiser actually corresponds to the MAP denoiser of a prior that is dependent on, but distinct from $p_{\boldsymbol{x}}$ and possibly nonconvex [29], [30]. Implicit PnP methods are widely used and item 3 of Prop. 5 is therefore important to justify convergence. To the best of our knowledge, further theoretical progress are required for a better understanding of the induced prior.

Finally, for an arbitrary denoiser, the convergence of PnP methods is not granted and instability can be observed when iterating the gradient steps.

Computability of MAP& MMSE denoisers:

- MAP denoisers, which should be used for implicit PnP methods are accessible only for simple priors (e.g., log-concave), which exhibit lower performance.
- MMSE denoisers, which should be used in explicit PnP methods can be approximated by training.
- MMSE denoisers can be used in implicit methods. Their statistical interpretation requires further clarification.

B. The problem of local maxima

PnP methods only converge to local maxima, which are often unsatisfactory. For instance, in Fig. 1, a basic PnP implementation yields the shallow local maximum. To avoid this, a multiscale method consists in using a sequence η_k of

regularization parameters for the prior. With a large value, the posterior distribution is smooth and unimodal, ensuring global convergence. We can then progressively get finer and finer prior approximation, avoiding traps [31]. It is also possible to use stochastic Langevin diffusion models by adding noise to the iterates [32], [33], [34]. They enable the generation of samples from the posterior distribution $p_{\boldsymbol{x}|\boldsymbol{y}}$. We show a few samples obtained in this way in Fig. 1.

IV. Unrolled methods and the MMSE

Training the unrolled network in (4) consists in optimizing the parameters $\theta = (\theta_1, \dots, \theta_K)$ of the denoisers of each block T_k. This is done by minimizing the empirical risk:

$$\theta^\star \approx \underset{\theta}{\operatorname{argmin}} \frac{1}{I} \sum_{i=1}^{I} \|T_\theta(y_i) - x_i\|^2$$

Following Claim 1, we therefore have $T_{\theta^\star} \approx \hat{x}_{\text{MMSE}}$ under the 4 conditions described after the claim. Empirical evidence suggests that the unrolled architecture is an excellent candidate for points ii) and iv).

Of importance, notice that the denoisers D_{θ_k} should not be considered as white Gaussian noise denoisers as in PnP algorithm. They are optimized to remove operator specific artifacts, since A appears in each block T_k.

Key differences between PnP and unrolled networks: PnP and unrolled methods possess exactly the same architecture, but are trained differently. PnP therefore yield MAP-like estimates, while well-trained unrolled networks approximate MMSE estimators.

V. Advantages and limitations of each method

Tab. II summarizes the main differences between PnP and unrolled methods. We justify each cell in what follows.

TABLE II: Comparison between plug-and-play and unrolled method

	PnP methods	Unrolled networks
Stat. interpretation	MAP (local.)	MMSE
Architecture	Identical, but smaller K for unrolled	
Training objective	Learn to denoise	Learn to reconstruct
Training cost	Rather lightweight	Rather expensive
Adaptivity	Any inverse problem	Problem-dependent
Inference time	Long if many iterations	Fast once trained
Convergence	Local minimizers	Ongoing research
Computation	High dim. optimization	High dim. integral
Stability	Low for nonsmooth priors	More stable
Appearance	Best looking	Best in average
Performance	Unrolled $>$ vanilla-PnP by up to 5 dB [35]	
Properties	Nice looking but uncertain	Blurry where unfaithful
	Trapped in local minimizers	Can be unlikely
	Improved with continuation	

Stability: The stability of both estimators to noise in the measurements is fundamental. The MAP stability heavily relies on the prior $p_{\boldsymbol{x}}$. The mapping $\hat{x}_{\text{MAP}}$ can become discontinuous if the posterior distribution $p_{\boldsymbol{x}|\boldsymbol{y}}$ has multiple global minimizers. This can happen even with log-concave priors, indicating that the MAP is generally unstable. In contrast, the MMSE estimator, defined as an integral of the joint probability $p_{\boldsymbol{x},\boldsymbol{y}}$, is continuous whenever the joint distribution $p_{(\boldsymbol{x},\boldsymbol{y})}$ is sufficiently smooth. It is therefore typically more stable than the MAP, as shown in [36]. A rigorous proof of this requires further investigation, which is out the scope of this paper.

Posterior property: The MAP estimator is designed to have a high posterior probability. In contrast, the MMSE, as the average optimal solution, can have an arbitrarily low posterior probability. In imaging, this often manifests as blurry regions in areas of uncertainty, as illustrated in Fig. 1.

Computational complexity: Contrarily to well-designed PnP methods, which can be run for an arbitrary number of iterations, unrolled methods are limited to a fixed number, typically between 1 and 16. The reason is that the training requires using back-propagation through K iterations, which are computationally and memory-intensive. This feature is also beneficial: since only a few iterations are needed, they typically run faster than PnP methods once trained.

Performance: Unrolled networks are designed optimally for a given operator. This results in performance gain varying from 0 to 5dBs in terms of Peak Signal to Noise Ratio [35].

Adaptivity: PnP methods are universal: a single denoiser can be used for any inverse problem and operator A. In contrast, unrolled methods are less adaptive; weights optimized for a specific operator A often perform poorly with a different operator A'. This adaptivity issue can be addressed by training the unrolled network on a family of operators [35], [37].

VI. Conclusion

We established links between PnP methods and the MAP estimator, as well as between unrolled networks and the MMSE estimator. This connection allowed us to list their distinct advantages and limitations for inverse problems solving.

PnP methods stand out for their ability to address arbitrary inverse problems without requiring network retraining. However, this flexibility comes at a cost, particularly for highly ill-posed problems, where issues such as getting trapped in poor local extrema, instability, and suboptimal performance can arise. Recent advances in stochastic PnP methods and diffusion processes [11], [12] demonstrate that these techniques can efficiently explore different modes of the posterior through sampling, as illustrated in Fig. 1. Uncertainty quantification remains a significant challenge in inverse problems, making this capability especially valuable.

Unrolled methods, in contrast, are tailored to a specific operator or a family of operators [35]. Although they can be time-consuming to train, they generally deliver optimal performance on average. For inherently uncertain inverse problems, this often results in blurry regions. This characteristic can be advantageous in scientific applications; instead of producing incorrect sharp details, the method generates a blur that a trained eye can interpret, offering a form of uncertainty quantification. Additionally, these methods tend to be faster and exhibit greater stability to noise, a crucial advantage in fields like medical imaging.

Acknowledgment

The authors acknowledge a support from the ANR Micro-Blind ANR-21-CE48-0008 and from the HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R3).

REFERENCES

[1] M. J. Muckley, B. Riemenschneider, A. Radmanesh, S. Kim, G. Jeong, J. Ko, Y. Jun, H. Shin, D. Hwang, M. Mostapha *et al.*, "Results of the 2020 fastmri challenge for machine learning mr image reconstruction," *IEEE transactions on medical imaging*, vol. 40, no. 9, pp. 2306–2317, 2021.

[2] F. S. de Moura, S. Siltanen, and M. Juvonen, "Helsinki deblur challenge 2021 (hdc20201) ipi special issue preface," *Inverse Problems and Imaging*, vol. 17, no. 5, pp. i–iii, 2023.

[3] A. Meaney, F. S. de Moura, M. Juvonen, and S. Siltanen, "Helsinki tomography challenge 2022: Description of the competition and dataset," *Applied Mathematics for Modern Challenges*, vol. 1, no. 2, pp. 170–201, 2023.

[4] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *2013 IEEE global conference on signal and information processing*. IEEE, 2013, pp. 945–948.

[5] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[6] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on international conference on machine learning*, 2010, pp. 399–406.

[7] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[8] S. Mukherjee, A. Hauptmann, O. Öktem, M. Pereyra, and C.-B. Schönlieb, "Learned reconstruction methods with convergence guarantees: A survey of concepts and applications," *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 164–182, 2023.

[9] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[10] M. Elad, B. Kawar, and G. Vaksman, "Image denoising: The deep learning revolution and beyond—a survey paper," *SIAM Journal on Imaging Sciences*, vol. 16, no. 3, pp. 1594–1654, 2023.

[11] A. Graikos, N. Malkin, N. Jojic, and D. Samaras, "Diffusion models as plug-and-play priors," *Advances in Neural Information Processing Systems*, vol. 35, pp. 14 715–14 728, 2022.

[12] Y. Zhu, K. Zhang, J. Liang, J. Cao, B. Wen, R. Timofte, and L. Van Gool, "Denoising diffusion models for plug-and-play image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1219–1229.

[13] A. Klenke, *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013.

[14] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," *Fixed-point algorithms for inverse problems in science and engineering*, pp. 185–212, 2011.

[15] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods," *Mathematical Programming*, vol. 137, no. 1, pp. 91–129, 2013.

[16] A. Themelis and P. Patrinos, "Douglas–rachford splitting and admm for nonconvex optimization: Tight convergence results," *SIAM Journal on Optimization*, vol. 30, no. 1, pp. 149–181, 2020.

[17] B. Efron, "Tweedie's formula and selection bias," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1602–1614, 2011.

[18] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching." *Journal of Machine Learning Research*, vol. 6, no. 4, 2005.

[19] E. T. Reehorst and P. Schniter, "Regularization by denoising: Clarifications and new interpretations," *IEEE transactions on computational imaging*, vol. 5, no. 1, pp. 52–67, 2018.

[20] S. Hurault, A. Leclaire, and N. Papadakis, "Gradient step denoiser for convergent plug-and-play," in *International Conference on Learning Representations (ICLR'22)*, 2022.

[21] ——, "Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9483–9505.

[22] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.

[23] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, p. 124007, 2017.

[24] X. Xu, Y. Sun, J. Liu, B. Wohlberg, and U. S. Kamilov, "Provable convergence of plug-and-play priors with mmse denoisers," *IEEE Signal Processing Letters*, vol. 27, pp. 1280–1284, 2020.

[25] A. Goujon, S. Neumayer, P. Bohra, S. Ducotterd, and M. Unser, "A neural-network-based convex regularizer for inverse problems," *IEEE Transactions on Computational Imaging*, 2023.

[26] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern recognition*, vol. 77, pp. 354–377, 2018.

[27] J. Adler and O. Öktem, "Deep bayesian inversion," 2018. [Online]. Available: https://arxiv.org/abs/1811.05910

[28] M. Belkin, "Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation," *Acta Numerica*, vol. 30, pp. 203–248, 2021.

[29] R. Gribonval, "Should penalized least squares regression be interpreted as maximum a posteriori estimation?" *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2405–2410, 2011.

[30] R. Gribonval and M. Nikolova, "A characterization of proximity operators," *Journal of Mathematical Imaging and Vision*, vol. 62, no. 6, pp. 773–789, 2020.

[31] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in neural information processing systems*, vol. 32, 2019.

[32] M. Girolami and B. Calderhead, "Riemann manifold langevin and hamiltonian monte carlo methods," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 73, no. 2, pp. 123–214, 2011.

[33] R. Laumont, V. D. Bortoli, A. Almansa, J. Delon, A. Durmus, and M. Pereyra, "Bayesian imaging using plug & play priors: when langevin meets tweedie," *SIAM Journal on Imaging Sciences*, vol. 15, no. 2, pp. 701–737, 2022.

[34] J. Song, A. Vahdat, M. Mardani, and J. Kautz, "Pseudoinverse-guided diffusion models for inverse problems," in *International Conference on Learning Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=9_gsMA8MRKQ

[35] A. Gossard and P. Weiss, "Training adaptive reconstruction networks for blind inverse problems," *SIAM Journal on Imaging Sciences*, vol. 17, no. 2, pp. 1314–1346, 2024. [Online]. Available: https://doi.org/10.1137/23M1545628

[36] M. Genzel, J. Macdonald, and M. März, "Solving inverse problems with deep neural networks–robustness included?" *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 1119–1134, 2022.

[37] M. Terris and T. Moreau, "Meta-prior: Meta learning for adaptive inverse problem solvers," 2023.