

Efficient Bayesian Computation Using Plug-and-Play Priors for Poisson Inverse Problems

Teresa Klatzer^{1,4}

Savvas Melidonis²

Marcelo Pereyra^{3,4}

Konstantinos C. Zygalakis^{1,4}

March 21, 2025

Abstract

This paper introduces a novel plug-and-play (PnP) Langevin sampling methodology for Bayesian inference in low-photon Poisson imaging problems, a challenging class of problems with significant applications in astronomy, medicine, and biology. PnP Langevin sampling algorithms offer a powerful framework for Bayesian image restoration, enabling accurate point estimation as well as advanced inference tasks, including uncertainty quantification and visualization analyses, and empirical Bayesian inference for automatic model parameter tuning. However, existing PnP Langevin algorithms are not well-suited for low-photon Poisson imaging due to high solution uncertainty and poor regularity properties, such as exploding gradients and non-negativity constraints. To address these challenges, we propose two strategies for extending Langevin PnP sampling to Poisson imaging models: (i) an accelerated PnP Langevin method that incorporates boundary reflections and a Poisson likelihood approximation and (ii) a mirror sampling algorithm that leverages a Riemannian geometry to handle the constraints and the poor regularity of the likelihood without approximations. The effectiveness of these approaches is demonstrated through extensive numerical experiments and comparisons with state-of-the-art methods.

1 Introduction

Low-photon Poisson imaging problems are ubiquitous in scientific and engineering applications, particularly in scenarios involving low illumination or short acquisition times (see, e.g., [58, 5, 21] for excellent introductions to the topic). Poisson-distributed measurements arise for instance from the use of single-photon detectors that discriminate individual photons within a given time frame [2, 46, 56], as well as from standard CMOS cameras that operate under poorly illuminated conditions [13]. As a result, Poisson imaging problems play a crucial role in astronomy and remote sensing [26, 17, 47], where imaging systems often operate under limited illumination conditions, in biomedical microscopy, where photon counts are limited to minimize photo-toxicity [55], and in nuclear medical imaging [70, 57], where emission-based modalities such as PET and SPECT produce Poisson-distributed measurement data with statistics that are directly related to the amount of radiation used.

We herein consider Poisson imaging problems where one seeks to recover an unknown image of interest $x \in \mathbb{R}^n$ from a linear measurement $y \in \mathbb{R}^m$ corrupted by Poisson noise, with likelihood function given by

$$p(y|x) \propto \prod_{i=1}^m \exp \left\{ y_i \log(\alpha(Ax)_i) - \alpha(Ax)_i - \iota_{\mathbb{R}_0^+}(x_i) \right\}, \quad (1)$$

¹School of Mathematics, University of Edinburgh, Edinburgh, EH9 3FD, UK

²Forschungszentrum Jülich GmbH, 52425 Jülich, Germany

³School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, EH14 4AS, UK

⁴Maxwell Institute for Mathematical Sciences, Bayes Centre, 47 Potterrow, EH8 9BT, Edinburgh, UK

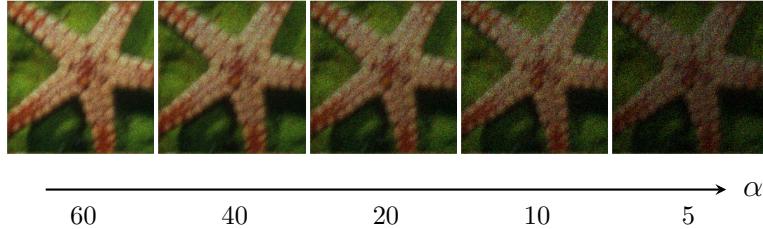


Figure 1: Illustration of different photon levels α on a blurred test image (kernel Fig.2b)

where $A \in \mathbb{R}_+^{m \times n}$ is a positive linear measurement operator representing deterministic aspects of the data acquisition process, $\alpha > 0$ is a scalar related to the level of shot noise contaminating the measurements (the smaller the value of α , the harder the problem, see Fig. 1 above), i denotes the i -th element of a vector, and $\iota_{\mathbb{R}_0^+}$ is the indicator function enforcing a non-negativity constraint on the elements of x . We assume that A is known (i.e., the problem is non-blind) and that $A^T A$ is rank deficient or exhibits a poor conditioning number, making the estimation problem challenging.

A main difficulty in Poisson imaging problems is the signal-dependent nature of Poisson noise, which leads to a signal-to-noise ratio that varies significantly across the image. In addition, in dark regions or low-illumination conditions, the noise distribution is strongly non-Gaussian. As a result, conventional image restoration techniques that rely on ℓ_2 -norm data-fidelity terms struggle with Poisson noise, especially in low-photon settings (for signals in the range $[0, 1]$, we see that conventional techniques break when $\alpha \leq 20$ [54]). Of course, the above difficulties are amplified in problems that are ill-posed or ill-conditioned, as considered in this paper. Such cases require introducing a significant amount of regularization to deliver accurate solutions.

Modern image reconstruction methods for Poisson imaging problems rely strongly on data-driven regularization techniques derived from machine learning (ML) (see, e.g., [54, 9, 60, 40]). We focus on so-called Plug-and-Play (PnP) strategies that encode the regularization function through an image denoising operator, which is embedded within an iterative computation algorithm in lieu of the regularizer's gradient or proximal operator, and used in combination with an explicit data-fidelity observation model that is specified during inference time (see [41] for a recent survey on PnP methods with theoretical guarantees). PnP approaches are predominantly derived from optimization schemes that deliver point estimators of x from y [41], or alternatively from stochastic sampling schemes that deliver a Monte Carlo approximation of a posterior distribution $p(x|y)$ defined implicitly by the image denoiser used [28, 11, 59, 40].

In this paper, we study novel PnP Markov chain Monte Carlo (MCMC) techniques suitable for Poisson imaging problems. Such techniques [28, 11, 59] are valuable because they allow forms of inference that are not accessible with PnP techniques based on optimization [23] or denoising diffusion models [40]. For example, PnP MCMC techniques can be embedded within empirical Bayesian machinery to tackle semi-blind imaging problems [38] or to automatically optimize regularization parameters [64, 61]. PnP MCMC techniques are also valuable for performing uncertainty quantification analyses [28, 34]. Unfortunately, to the best of our knowledge, the PnP MCMC techniques currently available are not suitable for Poisson imaging problems due to fundamental issues related to the poor regularity properties of (1). The aim of this paper is to address this important gap in Bayesian imaging methodology by proposing two extensions of the PnP framework [28] for Poisson imaging problems.

The remainder of this article is structured as follows: Section 2 reviews key concepts in Bayesian computation relevant to Bayesian imaging with PnP priors in the context of Gaussian observation models, and explains the main challenges in applying this approach for Poisson imaging problems. Section 3 presents two methodologies that address these difficulties and extend PnP Bayesian computation to Poisson problems: (i) the reflected PnP-SKROCK, an accelerated sampling scheme that leverages a variant of the Langevin diffusion tailored for constrained domains; and (ii) the PnP Mirror Langevin algorithm, which generalises PnP-ULA to a non-Euclidean geometry where the Poisson likelihood function is Lipschitz differentiable. In

Section 4, we present an extensive experimental analysis that thoroughly explores different neural network denoiser architectures and algorithmic choices for the considered Poisson problems, as well as comparisons with alternative strategies from the state-of-the-art. Conclusions and perspectives for future work are finally reported in Section 5.

2 Bayesian imaging with Plug-and-Play priors

2.1 The overdamped Langevin diffusion process

We consider performing Bayesian computation for Poisson imaging models of the form

$$p(x|y) = \frac{p(x)p(y|x)}{\int p(y|\tilde{x})p(\tilde{x})d\tilde{x}} \quad (2)$$

where $p(y|x)$ is given by (1) and where $p(x)$ is an image prior encoded by a ML model, which will be discussed in detail in Section 2.2. The canonical approach to computing posterior probabilities, expectations and estimators in Bayesian imaging problems is to employ a Markov Chain Monte Carlo (MCMC) method to draw samples from the posterior distribution of interest $p(x|y)$, followed by Monte Carlo integration [49]. In particular, Bayesian imaging methods often rely on discretizations of the overdamped Langevin diffusion process, which scales efficiently to high dimensional problems [15]. This diffusion is governed by the following stochastic differential equation (SDE)

$$dX_t = \nabla \log p(X_t)dt + \nabla \log p(y|X_t)dt + \sqrt{2}dW_t, \quad (3)$$

where W_t is a d -dimensional Brownian motion [15]. Several discrete-time approximations of (3) can be considered, but the most common choice is the Euler Maruyama (EM) approximation, leading to the so-called Unadjusted Langevin Algorithm (ULA)

$$X_{k+1} = X_k + \delta \nabla \log p(X_k) + \delta \nabla \log p(y|X_k) + \sqrt{2\delta}\xi_{k+1},$$

with a time step δ and ξ_k being i.i.d. standard Gaussian random variables. There exists a vast literature analyzing the properties of ULA under the assumption that $\nabla \log p(x|y)$ is Lipschitz, with detailed convergence guarantees in the log-concave [12, 50, 15, 31] and non log-concave settings [35, 16, 10]. Similar results have been established for variants of ULA suitable for non-smooth models [14, 8, 30, 18], as well as for models and ULAs using machine-learning priors under mild regularity conditions such as Lipschitz continuity and boundedness (see, e.g., [28, 7]). In the following section, we introduce the PnP Langevin framework [28] which we adopt in this paper.

2.2 Plug-and-Play priors

Bayesian imaging methods have traditionally relied on model-driven priors that enforce specific expected properties in the solution, such as smoothness, sparsity, or piecewise regularity. However, these are often not informative enough to deliver accurate reconstructions in challenging Poisson imaging problems. Therefore, we consider data-driven priors encoded by ML models, which can leverage large data sets of clean images to derive highly informative statistical image priors. In particular, herein we focus on priors encoded by ML-based image denoising operators, which can be embedded within iterative Monte Carlo sampling or optimization schemes in a PnP manner (see, e.g., [53, 44, 25, 20, 62] for examples of PnP optimization schemes and [28, 7, 48, 37] for examples of PnP sampling schemes). Here, we adopt the Bayesian PnP framework introduced in [28].

The main idea behind the Bayesian PnP approach [28] is to construct a Bayesian model $p(x|y)$ where the prior knowledge is represented in the form of an image denoising operator D_ϵ , rather than through an explicit prior distribution $p(x)$. This operator estimates the posterior expectation of x given a noisy observation $x' \sim \mathcal{N}(x, \epsilon \text{Id})$, with noise variance ϵ . In practice, D_ϵ is implemented by a deep neural network trained on a data set of clean and noisy image pairs

$\{x_i, x'_i\}_{i=1}^N$. The connection between D_ϵ and the prior $p(x)$ stems from Tweedie's identity, which states that if D_ϵ is close to the posterior expectation of x given $x' \sim \mathcal{N}(x, \epsilon \text{Id})$, then

$$\nabla \log p_\epsilon(x) \approx \frac{1}{\epsilon} (D_\epsilon(x) - x). \quad (4)$$

where $p_\epsilon(x)$ is a smooth approximation of $p(x)$ defined as $p_\epsilon(x) = \int k_\epsilon(x, \tilde{x}) p(\tilde{x}) d\tilde{x}$, obtained via convolution with a Gaussian smoothing kernel k_ϵ of bandwidth $\sqrt{\epsilon}$. As ϵ decreases, the approximation p_ϵ converges to p , at the expense of reduced smoothness.

In the context of PnP Langevin sampling methods, the gradient $\nabla \log p(x)$ in the Langevin SDE is first replaced by $\nabla \log p_\epsilon(x)$ and subsequently approximated by (4) by using a denoising operator D_ϵ that has been trained so that (4) holds. When combined with an ULA scheme, this leads to the PnP-ULA [28]

$$X_{k+1} = X_k + \delta \nabla \log p(y|X_k) + \frac{\delta}{\epsilon} (D_\epsilon(X_k) - X_k) + \sqrt{2\delta} \xi_{k+1}. \quad (5)$$

An alternative approach, known as projected PnP-ULA, modifies PnP-ULA by enforcing a hard projection onto a constraint set C . The projection ensures better geometric ergodicity properties for the generated Markov chain and also guarantees that the chain stays within the region of the solution space in which D_ϵ has been trained, avoiding out-of-distribution evaluations where D_ϵ can potentially behave erratically. In particular, we have the following recursion

$$X_{k+1} = \Pi_C(X_k + \delta \nabla \log p(y|X_k) + \frac{\delta}{\epsilon} (D_\epsilon(X_k) - X_k) + \sqrt{2\delta} \xi_{k+1}). \quad (6)$$

When dealing with Poisson noise, two key challenges hinder the direct application of PnP-ULA or PPnP-ULA. First, the continuous-time Langevin process is not well-defined because of the non-negativity constraint on x . Second, the gradient $x \mapsto \nabla \log p(y|x)$ for (1) is not globally Lipschitz, which is key for guaranteeing the convergence of the Langevin process to $p(x|y)$. In the next section, we discuss two approaches to deal with these challenges, each leading to new PnP algorithms.

3 Proposed methodology

We present two strategies to extend the conventional PnP Langevin approach to Poisson Bayesian imaging problems. Instead of directly modifying PnP algorithms, we introduce key adjustments at the level of the continuous-time SDE. These ensure that the algorithms resulting from time discretizations of the SDE are robust to the non-smoothness and constraints of (1). Note that, for presentation clarity, in a slight abuse of notation, we use $\nabla \log p_\epsilon(x)$ as a shorthand for the prior score function, although the algorithms are in practice implemented by using denoisers that do not verify Tweedie's identity exactly. This allows us to consider a wide range of denoiser architectures, including non-Euclidean denoisers, without having to redefine the algorithms.

3.1 Langevin diffusion for constrained domains

3.1.1 The reflected overdamped Langevin diffusion process

Building on [39], we first consider the following reflected Langevin SDE (RSDE) to sample approximately from $p(x|y)$

$$dX_t = \nabla \log p^\beta(y|X_t) dt + \nabla \log p_\epsilon(X_t) dt + \sqrt{2} dW_t + d\kappa_t \quad (7)$$

where $x \mapsto p^\beta(y|x)$ with $\beta > 0$ is a regularized approximation of the original likelihood (1), given by

$$p^\beta(y|x) \propto \prod_{i=1}^m \exp \left\{ y_i \log(\alpha(Ax)_i + \beta) - \alpha(Ax)_i - \beta - \nu_{\mathcal{R}_0^+}(x_i) \right\}, \quad (8)$$

where κ_t is a local time that increases only on the boundary $\partial\mathbb{R}_+^n$. The local time enforces the non-negativity constraint, as required by (1) [45, 39], while taking $\beta > 0$ ensures that $\nabla \log p^\beta(y|x)$ is Lipschitz-continuous on the positive orthant¹. Note that setting $\beta = 0$ recovers the original likelihood (1), and that the bias stemming from using $\beta > 0$ can be made arbitrarily small by reducing the value of β , at the expense of convergence speed (see [39, Section 4.4] for recommendations regarding setting β).

The RSDE (7) can be approximated by a standard EM scheme, leading to a constrained ULA. Two standard ways to discretize this RSDE ensuring that the samples are always in \mathbb{R}_{++}^n are using a reflected [39] or a projected Euler scheme [28]. The reflected EM scheme is defined by

$$X_{k+1} = \left| X_k - \delta \nabla \log p^\beta(y|X_k) - \delta \nabla \log p_\epsilon(X_k) + \sqrt{2\delta} \xi_{k+1} \right| \quad (9)$$

where δ and ξ_k remain as previously, and $|.|$ denotes the component-wise absolute value. The projected EM scheme is defined by

$$X_{k+1} = \left(X_k - \delta \nabla \log p^\beta(y|X_k) - \delta \nabla \log p_\epsilon(X_k) + \sqrt{2\delta} \xi_{k+1} \right)^+ \quad (10)$$

where $(.)^+$ is the projection to the positive orthant. Under mild regularity assumptions on p_ϵ , both approaches are by construction well-posed and converge exponentially fast to a neighborhood of the regularized target $p_{\beta,\epsilon}(x|y) \propto p^\beta(y|x)p_\epsilon(x)$, with the reflected EM schemes (9) typically exhibiting a smaller bias [39].

3.1.2 Reflected and Projected PnP Langevin algorithms

PnP priors encoded by neural network denoisers can become unreliable when evaluated on out-of-distribution data [28]. Therefore, to enable the reliable use of recursions (9) and (10) for sampling, we impose an additional constraint to ensure that the iterates remain within the domain of D_ϵ , which we henceforth denote by $C \subset \mathbb{R}^d$. Again, we can incorporate this constraint through reflection or projection onto C . This leads to two variants of PnP-ULA, namely PnP-ULA with reflection (RPnP-ULA, see Algorithm 1), and PnP-ULA with projection (PPnP-ULA, see Algorithm 2). For each algorithm we need to specify the number of iterations N , the noisy and blurry observation y , the photon level α (which appears within $\nabla \log p^\beta$), a noise level ϵ for the used denoiser, a hyper-parameter $\rho > 0$, a step size δ , and the set C , which for computational tractability we assume to be convex and compact. The choice of the step size δ for RPnP-ULA is governed by bounds given in [28], which requires $\delta < \frac{1}{3} \text{Lip}(\nabla \log p_{\beta,\epsilon}(x|y))^{-1}$ where $\text{Lip}(.)$ denotes the Lipschitz constant. PPnP-ULA admits a larger step size, at the expense of additional estimation bias [28].

Algorithm 1 PnP-ULA with Reflection (RPnP-ULA)

Require: $N \in \mathbb{N}, y \in \mathbb{R}^m, \epsilon, \delta > 0, \rho > 0, C \subset \mathbb{R}^n$ convex and compact

Initialization:

Set $X_0 \in \mathbb{R}_{++}^n$ and $k = 0$.

for $k = 0 : N$ **do**

$$Z_{k+1} \sim \mathcal{N}(0, \text{Id})$$

$$X_{k+1} = \mathcal{R}_C \left(X_k + \delta \nabla \log p^\beta(y|X_k) + \delta \rho \nabla \log p_\epsilon(X_k) + \sqrt{2\delta} Z_{k+1} \right)$$

end for

One potential drawback of EM-based algorithms such as Algorithms 1 and 2 is that they might converge slowly due to a step size restriction [43]. One algorithm that alleviates this step size restriction is the stochastic orthogonal Runge–Kutta–Chebyshev (SKROCK) method [43], which can be shown to behave similarly to “accelerated” optimization algorithms in terms of

¹The constant $\beta > 0$ can be interpreted as a constant background noise level.

Algorithm 2 PnP-ULA with Projection (PPnP-ULA)

Require: $N \in \mathbb{N}, y \in \mathbb{R}^m, \epsilon, \delta > 0, \rho > 0, C \subset \mathbb{R}^n$ convex and compact

Initialization:

Set $X_0 \in \mathbb{R}_{++}^n$ and $k = 0$.
for $k = 0 : N$ **do**

$$Z_{k+1} \sim \mathcal{N}(0, \text{Id})$$

$$X_{k+1} = \Pi_C \left(X_k + \delta \nabla \log p^\beta(y|X_k) + \delta \rho \nabla \log p_\epsilon(X_k) + \sqrt{2\delta} Z_{k+1} \right)$$

end for

convergence to equilibrium for Gaussian targets. A follow-up method called reflected SKROCK (RSKROCK) was proposed in [39] to deal with Poisson noise and analytical model-based priors by adding a reflection to SKROCK, and was shown to behave in an accelerated manner.

Following on from this, in this paper we propose a so-called ‘‘accelerated’’ variant of RPnP-ULA: PnP-SKROCK with reflection (RPnP-SKROCK, see Algorithm 3). Unlike ULAs that perform a single gradient evaluation per iteration, SKROCK requires $s \in \mathbb{N}$ gradient evaluations per iteration, allowing SKROCK to take much longer integration steps. While the cost per iterations increases by a factor s , in problems that are ill-conditioned or ill-posed, this can lead to an improvement in convergence speed of the order of s^2 . We refer the reader to [43] for more details. In all our experiments, we use $s = 10$ and Runge-Kutta expansion parameter $\eta = 0.05$.

Algorithm 3 Reflected PnP-SKROCK (RPnP-SKROCK)

Require: $N \in \mathbb{N}, y \in \mathbb{R}^m, \epsilon, \delta > 0, \rho > 0, \eta, C \subset \mathbb{R}^n$ convex and compact

Compute $l_s = (s - 0.5)^2(2 - 4/3\eta) - 1.5$

Compute $\omega_0 = 1 + \frac{\eta}{s^2}, \quad \omega_1 = \frac{T_s(\omega_0)}{T'_s(\omega_0)}, \quad \mu_1 = \frac{\omega_1}{\omega_0}, \quad \nu_1 = s\omega_1/2, \quad k_1 = s\omega_1/\omega_0$

Choose $\delta \in (0, \delta_s^{max}]$, where $\delta_s^{max} = l_s / \text{Lip}(\nabla \log p_{\beta,\epsilon}(x|y))$; $\rho > 0$

Initialization $X_0 \in \mathbb{R}_{++}^n$ and $k = 0$

for $k = 0 : (N - 1)$ **do**

$$Z_{k+1} \sim \mathcal{N}(0, \mathbb{I}_n)$$

$$K_0 = X_k$$

$$W_1 = \mathcal{R}_C \left(X_k + \nu_1 \sqrt{2\delta} Z_{k+1} \right)$$

$$Y_1 = X_k - \mu_1 \delta \nabla \log p^\beta(y|W_1) + \mu_1 \delta \rho \nabla \log p_\epsilon(W_1) + k_1 \sqrt{2\delta} Z_{k+1}$$

$$K_1 = \mathcal{R}_C(Y_1)$$

for $j = 2 : s$ **do**

Compute $\mu_j = \frac{2\omega_1 T_{j-1}(\omega_0)}{T_j(\omega_0)}, \quad \nu_j = \frac{2\omega_0 T_{j-1}(\omega_0)}{T_j(\omega_0)}, \quad k_j = -\frac{T_{j-2}(\omega_0)}{T_j(\omega_0)} = 1 - \nu_j$

$$K_j = \mathcal{R}_C(-\mu_j \delta \nabla \log p^\beta(y|K_{j-1}) + \mu_j \delta \rho \nabla \log p_\epsilon(K_{j-1}) \\ + \nu_j K_{j-1} + k_j K_{j-2})$$

end for

$$X_{k+1} = K_s$$

end for

return $\{X_k : k \in \{1, \dots, N\}\}$

3.2 A Riemannian overdamped Langevin diffusion process

An interesting alternative to the RSDE approach described previously is to tackle the constraints and non-Lipschitz gradients by modifying the geometry underpinning the Langevin SDE. This can be achieved by using the so-called mirror Langevin scheme [68], analogue to mirror descent

optimization algorithms [42]. Mirror optimization algorithms have been introduced by [42] and can be viewed as projected sub-gradient (or Bregman proximal gradient) methods, derived from using a Bregman divergence instead of the usual Euclidean squared distance to alter the geometry of the problem [4]. In mirror schemes, this modification from Euclidean to (Hessian) Riemannian geometry is encoded through the so-called mirror map [4]. Following [68], we consider the following (Riemannian) mirror Langevin SDE to sample from a generic distribution π

$$\begin{aligned} X_t &= \nabla\phi^*(Y_t) \\ dY_t &= \nabla \log \pi(X_t) dt + \sqrt{2[\nabla^2\phi(X_t)]^{1/2}} W_t \end{aligned} \quad (11)$$

where ϕ is the mirror map, which we assume to be $\mathcal{C}^2(\mathcal{X})$ Legendre-type convex where \mathcal{X} denotes the support of π , and where ϕ^* the Legendre-Fenchel conjugate of ϕ , i.e.,

$$\phi^*(y) = \sup_{x \in \mathcal{X}} \langle x, y \rangle - \phi(x),$$

and where π is assumed differentiable on \mathcal{X} . This mirror Langevin SDE stems from endowing \mathcal{X} with a Riemannian metric, derived from the Hessian $\nabla^2\phi(x)$ (see [68] for details).

For Poisson inverse problems, a natural choice for ϕ is the Burg's entropy (or maximum entropy) [6], given by

$$\phi(x) = \sum_{i=1}^n \log(x_i), \quad (12)$$

as the likelihood (1) is smooth w.r.t. to this choice of ϕ (see, e.g., [3] for details). Again, there are many ways to obtain a discrete sampling scheme from a mirror Langevin SDE [33, 27, 68, 1, 22, 19]. We choose the EM scheme studied in [68]

$$X_{k+1} := \nabla\phi^* \left(\nabla\phi(X_k) - \delta \nabla \log \pi(X_k) + \sqrt{2\delta[\nabla^2\phi(X_k)]} \xi_{k+1} \right). \quad (13)$$

Crucially, (13) can be applied directly to π without the need for smoothing or reflections, as the constraints and non-smoothness of (1) are being dealt with by (12).

Algorithm 4 below summarizes our proposed PnP mirror Langevin algorithm (PnP-MLA) to sample from $p(x|y)$. Because the positivity constraints and smoothness issues are addressed by using the Burg's entropy (12) as mirror map ϕ , PnP-MLA uses the correct likelihood $\nabla \log p(y|x)$, not the approximation $\nabla \log p^\beta(y|x)$ as previously.

Algorithm 4 PnP Mirror Langevin Algorithm (PnP-MLA)

Require: $N \in \mathbb{N}, y \in \mathbb{R}^m, \delta > 0, \rho > 0$

Initialization:

Set $X_0 \in \mathbb{R}^n$ and $k = 0$.

for $k = 0 : N$ **do**

$$Z_{k+1} \sim \mathcal{N}(0, Id)$$

$$X_{k+1} = \nabla\phi^* \left(\nabla\phi(X_k) + \delta \nabla \log p(y|X_k) + \delta \rho \nabla \log p_\epsilon(X_k) + \sqrt{2\delta[\nabla^2\phi(X_k)]} Z_{k+1} \right)$$

end for

4 Experiments

We analyze the proposed Bayesian imaging methodologies through a series of numerical experiments related to non-blind Poisson image deconvolution, focusing on the choice of denoiser architecture and Langevin sampling algorithm. We have chosen this deconvolution problem because it provides a flexible framework for comparing different denoiser and computation strategies under various degrees of problem ill-conditioning and levels of noise. More precisely, we consider

(1) in the specific case where A is a nearly singular blur operator leading to an ill-conditioned problem with highly noise-sensitive solutions; in the following experiments, we implement A using a range of blur kernels taken from [23, 32], depicted in Figure 2 below. We carry out tests using two open datasets of color images: the **set3c** dataset [24], and a subset of 10 images from the **CBSD68** validation set (images of size 256×256 pixels obtained by a centre crop) [36], which we henceforth refer to as the **CBSD10** set. The images are displayed in Figure 3 and Figure 4.

The remainder of this section is organized as follows:

- In Section 4.1, we discuss the different denoisers that we will use in our numerical experiments and their corresponding neural network architectures.
- In Section 4.2, we compare these different image denoiser architectures as PnP image priors. We assess their performance in terms of reconstruction accuracy, uncertainty visualization quality, and the convergence speed of the resulting PnP algorithm.
- In Section 4.3, we compare sampling algorithms when used with the most competitive image prior as identified in Section 4.2; we assess performance in terms of algorithm stability, convergence speed, and reconstruction quality.
- In Section 4.4, we compare the proposed algorithms to alternative strategies from the state-of-the-art (SOTA). To capture different aspects of the estimation error, we assess reconstruction accuracy by computing peak signal-to-noise ratio (PSNR)², the structural similarity index measure (SSIM) [65], and the learned perceptual image patch similarity (LPIPS) [69].

Experiments were conducted on a workstation with an Intel i9-9940X CPU (14 cores, 28 threads, 3.30 GHz base clock), 126 GB RAM, NVIDIA GeForce RTX 2080 Ti Rev. A (11 GB VRAM, CUDA 12.6) GPU, Rocky Linux 8.10 operating system and using Python 3.9.18 software with PyTorch 2.1.0 and Numpy 1.24.3 libraries.

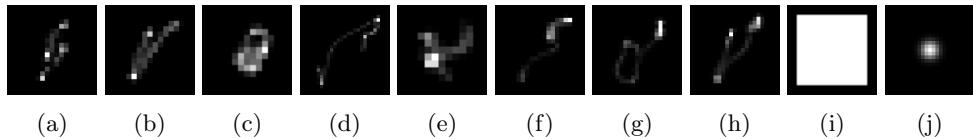


Figure 2: Examples of blur kernels (support 25×25 pixels): motion blur kernels [32] (2a)-(2h), a box blur (2i) and an isotropic Gaussian blur (2j) of bandwidth 1.6 pixels.

4.1 Choice of PnP prior for Poisson imaging problems

As mentioned previously, PnP imaging methods rely predominantly on denoisers that are specifically trained to approximate the posterior mean or MMSE denoiser, with some mild regularization to control the Lipschitz regularity of D_ϵ . These are predominantly denoisers encoded by convolutional neural networks (CNNs), which are trained in a supervised manner using clean and noisy image pairs corrupted by Gaussian noise with level ϵ . Noteworthy examples include the DnCNN architecture (Denoising CNNs) [67] or the DRUNet (Dilated-Residual U-Net) [66]. Patch-based denoising algorithms such as BM3D [29] are also widely within PnP optimization schemes, but they are too computationally expensive for Langevin PnP sampling. Special attention is paid in the PnP literature to denoisers that are non-expansive (have a Lipschitz constant smaller than 1), as well as denoisers that are equivalent to gradient steps or proximal steps for a specific potential function, which allow embedding the denoisers within optimization machinery with convergence guarantees [24, 25].

By contrast, the framework [28] for PnP-ULA can be applied with any Lipschitz continuous denoiser D_ϵ for which (4) holds approximately, and without the need for D_ϵ to be contractive

$$^2\text{PSNR}(\hat{x}, x) = 20 \log_{10} \left(\frac{\text{MAX}_x}{\sqrt{\text{MSE}(\hat{x}, x)}} \right), \text{MSE}(\hat{x}, x) = \frac{1}{n} \sum_{i=0}^{n-1} (x(i) - \hat{x}(i))^2$$



(a) Butterfly

(b) Leaves

(c) Starfish

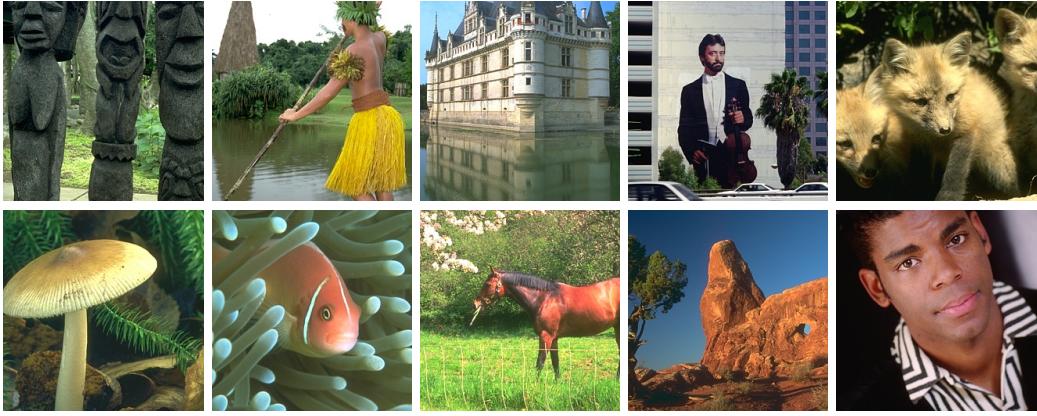
Figure 3: Ground truth images from **set3c**

Figure 4: CBSD10 data set

or define a gradient or proximal mapping. However, it is still useful to have some control of the Lipschitz constant of D_ϵ , as this leads to Bayesian models with better regularization properties and to PnP-ULA samplers that converge faster. We explore the following choices for D_ϵ :

1. The non-expansive denoiser (LMMO) from [44]³, based on a modified DnCNN architecture that is trained to provide a maximally monotone operator.
2. The GS-DRUNet denoiser [24]⁴, based on the DRUNet architecture and realized as a trained gradient descent step.
3. The Prox-DRUNet denoiser [25]⁵, also based on the DRUNet architecture and realized as a proximal descent step.
4. The B-DRUNet denoiser [23]⁶, a Bregman proximal step DRUNet architecture, with a proximal step defined via a (non-Euclidean) Bregman divergence.

The total number of parameters for the first denoiser is 668k, whereas the three DRUNet denoisers have 17M parameters. The DRUNet gradient-step denoisers have the form $D_\epsilon(x) = x - \nabla g_\epsilon(x) = N_\epsilon(x) + J_{N_\epsilon}^T(x)(x - N_\epsilon(x))$ where $N_\epsilon(x)$ is a DRUNet and $J_{N_\epsilon}(x)$ denotes its Jacobian. As a result, they have a higher computational complexity compared to the DnCNN-type architecture (due to the computation of the Jacobian J_{N_ϵ} and two applications of

³https://github.com/matthieutrs/LMMO_lightning

⁴<https://github.com/samuro95/GSPnP>

⁵<https://github.com/samuro95/Prox-PnP>

⁶<https://github.com/samuro95/BregmanPnP>

$N_\epsilon(x)$). Substituting this formulation into Tweedie's formula Eq. (4), we obtain $\nabla \log p_\epsilon(x) = \frac{1}{\epsilon}(x - \nabla g_\epsilon(x) - x) = -\frac{1}{\epsilon}\nabla g_\epsilon(x)$.

We also consider B-DRUNet [23], a Bregman generalization of the Euclidean proximal gradient step denoiser [25]. This denoiser is based a generalised form of Tweedie's identity, suitable for denoising problems with non-Gaussian noise distributions. More precisely, given a noisy observation z of an image x , then [23]

$$\hat{x}_{\text{MMSE}}(z) = \mathbb{E}[x|z] = z - \frac{1}{\gamma}(\nabla^2\phi(z))^{-1} \cdot \nabla(-\log p_\gamma)(z), \quad (14)$$

where ϕ is a C^2 convex potential of Legendre type, related to the distribution of the noise, and γ a scalar representing the noise level. Note that when ϕ is quadratic, (14) reduces to the conventional Tweedie identity (4) for Gaussian denoising. It is useful to set ϕ to match the algebraic form of the likelihood function. In our case, this leads to the Burg entropy (12) introduced previously, which is related to inverse gamma noise (see [23] for details). Given a trained Bregman denoiser $B_\gamma(z) \approx \hat{x}_{\text{MMSE}}(z)$, we obtain the required approximation to the score prior score function as $-\nabla \log p_\gamma(z) \approx \gamma \nabla^2 \phi(z)(z - B_\gamma(z))$. Note that in this case, we have $B_\gamma(z) = z - (\nabla \phi(z))^{-1} \cdot \nabla g_\gamma(z)$, which simplifies to $\nabla g_\gamma \approx -\frac{1}{\gamma} \nabla \log p_\gamma$. Thus, we can use our algorithms without any modification by setting $\epsilon = \frac{1}{\gamma}$ and $\nabla \log p_\epsilon(x) = -\frac{1}{\epsilon} \nabla g_{1/\epsilon}(x)$.

Moreover, a common practical issue with PnP optimization and sampling algorithms is that they can suffer from reconstruction artefacts which become more pronounced as iterations progress. Usual strategies to mitigate these issues in PnP optimization schemes are early stopping and iteration-dependent parameter fine-tuning [51, 53, 44]. However, these strategies are ineffective for PnP-ULAs, which aim to explore the posterior distribution to compute statistical estimates of interest, as this requires an algorithm that is ergodic and therefore stable. An interesting alternative, which we adopt herein, is to improve the stability of PnP-ULA by enforcing denoiser equivariance as recommended in [63]. Based on the intuition that image priors should be invariant to certain groups of transformations, such as rotations or reflections, we define transformations associated with a group \mathcal{G} as $\{T_g\}_{g \in \mathcal{G}}$ with $T_g \in \mathbb{R}^{n \times n}$ denoting unitary matrices to describe a transformation. The resulting averaged denoiser is $D_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} T_g^{-1} D_\epsilon(T_g x)$ (analogous for B_γ). For computational efficiency, rather than summing over all transformation, we approximate the sum by a simple one-sample Monte Carlo estimate by drawing $g \sim \mathcal{G}$ and setting $\tilde{D}_{\mathcal{G}}(x) = T_g^{-1} D_\epsilon(T_g x)$. This strategy effectively mitigates artefacts from training the denoiser imperfectly, and leads to PnP schemes with significantly better stability [63].

4.2 Comparison of PnP denoisers

4.2.1 Experimental set up

We compare the different denoisers from Section 4.1 focusing on their performance for Bayesian PnP Poisson image deconvolution. We evaluate them in terms of the accuracy (PSNR, SSIM, LPIPS) of the reconstruction \hat{x}_{MMSE} , the quality of the uncertainty estimates by visually contrasting the posterior standard deviations with the residual reconstruction errors for \hat{x}_{MMSE} , and by measuring the number of iterations required to compute the \hat{x}_{MMSE} (we track the PSNR and stop when this statistic is stable).

We use pre-trained checkpoints for GS-DRUNet, Prox-DRUNet and B-DRUNet, and retrain the LMMO denoiser based on the latest released training software⁷. To ensure fair comparisons, the same training strategy was used for all networks in a shared range of noise levels $\sigma_{\text{net}} \in [0, 50]$. We perform our comparisons with two different algorithms, PPnP-ULA as an example of a projected Euclidean algorithm⁸, and the non-Euclidean PnP mirror Langevin algorithm (PnP-MLA). In the case of PnP-MLA we use the mirror map associated with Burg's entropy to ensure that the generated samples remain strictly positive. In addition, for this experiment, we seek to evaluate the LMMO, GS-DRUNet and Prox-DRUNet denoisers in terms of image reconstruction

⁷https://github.com/matthieutrs/LMMO_lightning

⁸We have also conducted comparisons with other Euclidean algorithms such as RPnP-ULA (Algorithm 1) and R-SKROCK (Algorithm 3) -not reported here- and observed very similar results.

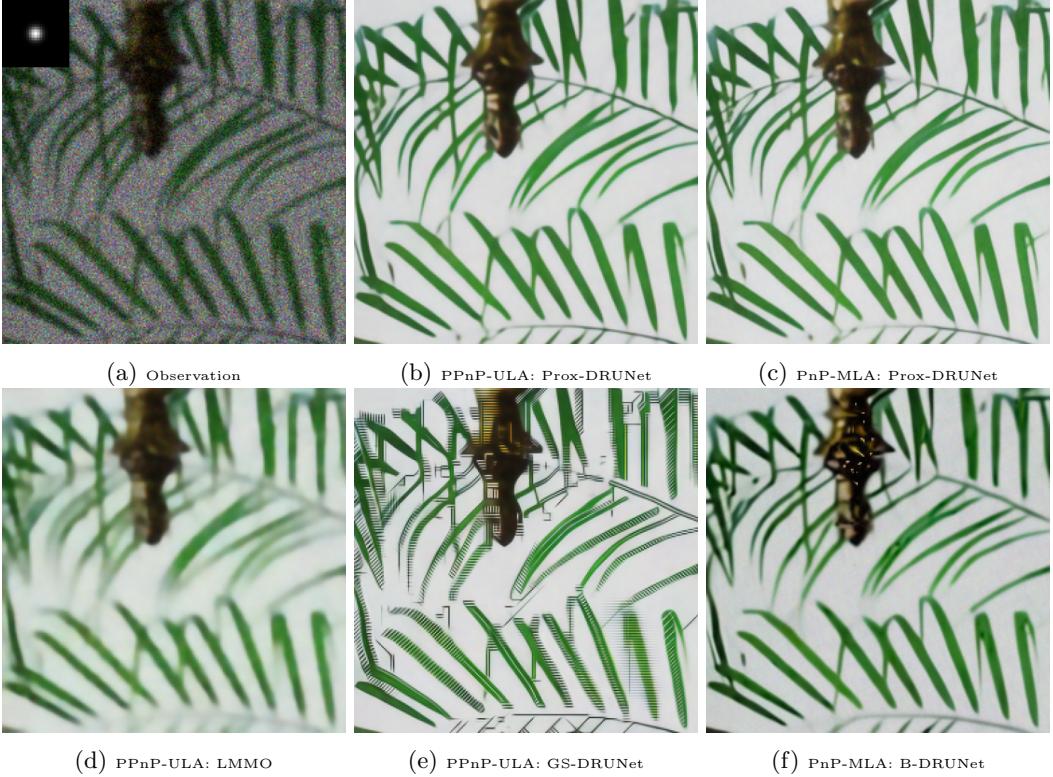


Figure 5: Poisson deconvolution problem for photon level $\alpha = 20$ and Gaussian blur of size 25×25 with standard deviation 1.6. using the `leaves` image. Noisy and blurry observation, and the MMSE reconstructions \hat{x}_{MMSE} for Prox-DRUNet, LMMO, GS-DRUNet and B-DRUNet using PPnP-ULA and PnP-MLA.

performance and UQ capabilities, without enforcing equivariance by randomization, as we seek to compare the denoiser architectures without this effect, which can be assessed separately. However, we do apply randomization to B-DRUNet, as the algorithm is unstable otherwise.

With regards to step size values, for PPnP-ULA, [28] suggests setting $\delta \in (0, \delta_L)$ with $\delta_L = 1/(L_{p^\beta} + L_\epsilon/\epsilon)$, where $L_{p^\beta} = \alpha^2 \cdot (\max(y)/\beta^2) \cdot \|AA^T\|$ is the Lipschitz constant of $\log p^\beta$ and L_ϵ is the Lipschitz constant of the denoiser; LMMO and Prox-DRUNet have been trained so that $L_\epsilon = 1$,⁹ while for GS-DRUNet $1 < L_\epsilon < 5$ [25]. However, while this choice is suitable for Gaussian imaging problems [28], we find this choice of step size overly conservative for Poisson problems and set $\delta = 0.2 \cdot 10^2 \cdot \delta_L$, which in our experience provides a good accuracy-speed trade-off. To the best of our knowledge, there is no theory for setting the step size of PnP-MLA (Algorithm 4) under a PnP prior or a non-convex setting. We set the step size in PnP-MLA as $\delta = 10^{-5} \simeq (2 \cdot 10^2) \cdot \delta_L$ by using grid search and optimizing for PSNR. We run the algorithms for 10^6 iterations to stress-test the stability of the algorithms. Using grid search, the parameter ϵ of the LMMO, GS-DRUNet, Prox-DRUNet denoisers D_ϵ is set to $\epsilon = (20/255)^2$ and $\rho = 1$. For the B-DRUNet denoiser B_γ , we set $\gamma = 30$ and $\rho = 2$. We set $C = \{x : 0 \leq x_i \leq 1\}$, as this corresponds to the training domain of the denoisers.

4.2.2 Experimental results

A set of representative results with the `leaves` image from `set3c` is presented in Figure 5 (additional results are available in Appendix A). Figure 5a depicts a realization y generated by the forward model Eq. (1) with the operator A modelling a Gaussian blur operator of size 25×25

⁹The training loss of LMMO and Prox-DRUNet includes a penalization term on the spectral norm of the Jacobian, for more details see [25, 44].

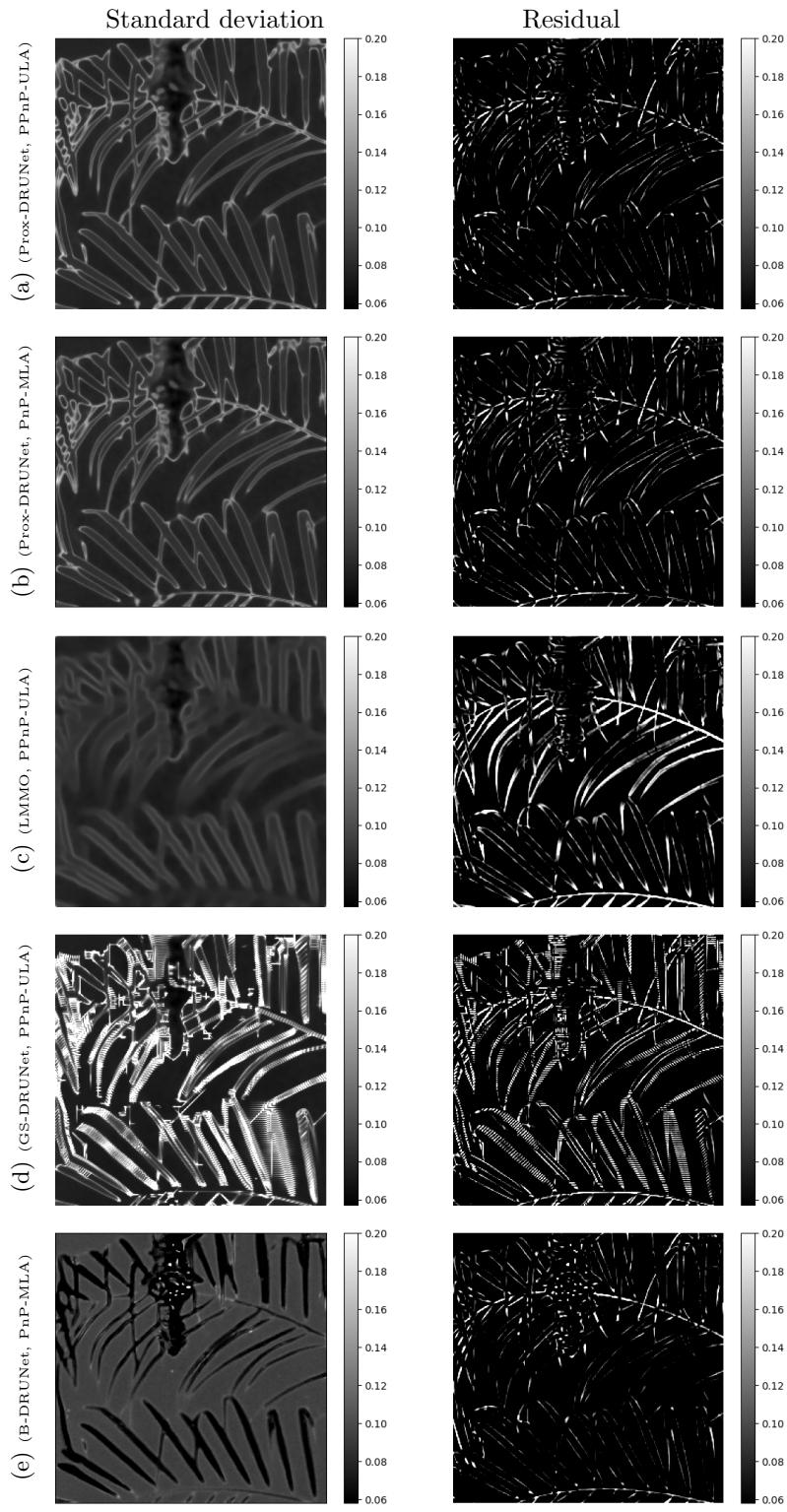


Figure 6: Poisson deconvolution problem as in Fig. 5. Pairwise: Pixelwise standard deviations (left column) and residuals (right column) for all tested denoisers using either PPnP-ULA or PnP-MLA.

Method	Denoiser	PSNR	SSIM	LPIPS	Iter. until 98% peak PSNR	sec./Iter.
PPnP-ULA	LMMO	17.20	0.63	0.38	4000	0.012
	GS-DRUNet	16.06	0.65	0.35	4000	0.025
	Prox-DRUNet	20.93	0.82	0.12	18000	0.025
PnP-MLA	Prox-DRUNet	21.29	0.82	0.13	12000	0.025
	B-DRUNet	17.83	0.75	0.23	136500	0.025

Table 1: Quantitative results for \hat{x}_{MMSE} calculated by PPnP-ULA and PnP-MLA for the `leaves` image. Last two columns: The number of iterations required to reach 98% PSNR and the time per iteration in seconds.

with standard deviation 1.6 pixels and a photon level set to $\alpha = 20$. Figures 5b to 5f show the posterior mean \hat{x}_{MMSE} , as calculated for the different denoisers and different algorithms. We observe that Prox-DRUNet outperforms the other denoisers in terms of reconstruction quality for both Euclidean and non-Euclidean algorithms and see that both algorithms deliver highly similar estimates, which capture fine detail without noticeable artefacts. In comparison, the LMMO denoiser produces an estimate with excessive smoothing, while the GS-DRUNet exhibits stripe-like artefacts. These artefacts are also present when respecting the step size bounds recommended in [28] (see Section 3.1.2, not shown here), and increase gradually with greater step size. The B-DRUNet also exhibits artefacts, amplifying some structures in the top central part of the image.

Table 1 summarizes the performance of the denoisers for the considered image. The LMMO prior leads to faster convergence but achieves low reconstruction quality, while GS-DRUNet is not stable without randomization, so the reconstruction quality decreases as the iterations progress and the artefacts become more pronounced. Similarly, B-DRUNet also leads to instability and reaches its top performance in around $1.3 \cdot 10^5$ iterations before artefacts start to amplify. Conversely, Prox-DRUNet outperforms the other networks in all the image quality metrics considered. To check for artefacts, it can be a helpful diagnostic tool to compare image metrics using the cumulative mean over all samples with the mean computed from a subset of samples (e.g., 100 samples seem to be enough to be representative). Scores will be comparable unless the denoiser produces artefacts, then the scores over the subset of samples will be significantly worse.

For completeness, Table 1 also reports the number of iterations that PPnP-ULA and PnP-MLA require to reach 98% of peak PSNR performance, as an indicator of convergence speed for the posterior mean. We observe that the convergence speed of PPnP-ULA and PnP-MLA are similar for the Prox-DRUNet denoiser. We also report the time in seconds per iteration, which is largely determined by the complexity of one denoiser application. Note that the Prox-DRUNet yields the best accuracy-speed trade-off, reaching almost top performance under $2 \cdot 10^4$ iterations for both PPnP-ULA and PnP-MLA. We point out that for PPnP-ULA, similar quantitative behavior was observed for the different denoisers even when randomization was used, as well as when using Algorithms 1 and 3.

Lastly, Figure 6 shows the pixel-wise posterior standard deviation, as calculated with the LMMO, Prox-DRUNet, GS-DRUNet and B-DRUNet denoisers. For reference, next to each posterior standard deviation plot, we also report the residual obtained by comparing \hat{x}_{MMSE} to the true image (note that these standard deviations represent the models' marginal predictions for these residuals, at the pixel level). We observe that LMMO, Prox-DRUNet, GS-DRUNet produce uncertainty plots that are broadly in agreement with their residuals. For Prox-DRUNet and LMMO, uncertainty concentrates around edges and contours, as expected for a deconvolution problem, whereas the uncertainty estimates of GS-DRUNet are aligned with its reconstruction artefacts. Conversely, B-DRUNet produces uncertainty plots that highlight homogenous regions, and which do not align well with its residual. We conclude that Prox-DRUNet is the most appropriate denoising architecture for Bayesian PnP inference in Poisson image deblurring problems and use this as the prior for the rest of our numerical experiments.

4.3 Langevin sampling algorithms for PnP Poisson deconvolution

4.3.1 Experimental set up

We now compare different PnP Langevin strategies with Prox-DRUNet as prior. We study how different choices of step size affects convergence speed, reconstruction quality, and numerical stability, with the goal to explore the trade-off that yields the best results for each algorithm. We interpret loss of stability through artefacts that appear in the reconstructions. We compare MMSE reconstructions computed with different sampling algorithms for a range of step sizes, both quantitatively and qualitatively, to obtain step size recommendations for each algorithm. Additionally, we compare the convergence speed between different sampling algorithms for the recommended step size over a set of images by looking at mean cumulative quality metrics. How image quality and convergence speed interrelate for different algorithms can guide the choice of sampling algorithm according to given priorities with regards to metric and computational budget.

For this experiment, we use images from the `set3c` data set, depicted in Figure 3, the blur kernel displayed Figure 2b and photon level $\alpha = 20$. We choose the `starfish` image to illustrate qualitative results. With regards to the algorithms and their step sizes, for the RPnP-ULA (Algorithm 1) and PPnP-ULA (Algorithm 2), we test a range of step sizes $\delta = c \cdot \delta_L$, with $c \in [1, 10^3]$. For RPnP-SKROCK (Algorithm 3), we test step sizes of the form $\delta = c \cdot \ell_s \delta_L$, with $c \in [0.1, 10^2]$, where $\ell_s = (s - 0.5)^2(2 - 4/3\eta) - 1.5$ with $s = 10$ and $\eta = 0.05$; for more details see [43]. For the PnP-MLA (Algorithm 4), we test step sizes of the form $\delta = c \cdot \delta_L$, where $c \in [10, 10^3]$. The Prox-DRUNet denoiser parameter ϵ is set to $\epsilon = (20/255)^2$ for RPnP-ULA, PPnP-ULA and RPnP-SKROCK, whereas we use $\epsilon = (25/255)^2$ for PnP-MLA (see Appendix B.2 for complementary results for PnP-MLA with $\epsilon = (20/255)^2$). In all cases, to improve numerical stability and reduce PnP artefacts, we enforce equivariance to horizontal and vertical flips as well as to rotations of multiples of 90 degrees by randomization.

4.3.2 Experimental results

We show qualitative results in Figure 7, comparing MMSE reconstructions \hat{x}_{MMSE} for a range of step sizes for algorithms RPnP-ULA, PPnP-ULA, RPnP-SKROCK, and PnP-MLA respectively; with quantitative results averaged over `set3c` in the captions. Note that we use a detail of the `starfish` image in this figure to illustrate the effects of the chosen step size; the full image for the best step size of each algorithm is shown in Figure 8. More detailed quantitative results are deferred to the Appendix B.1 in Tables 4-7.

Looking at Figures 7.1 and 7.2, we find that RPnP-ULA and PPnP-ULA both achieve the best quality at $c = 1.5 \cdot 10^2$ in all reported metrics (PSNR, LPIPS, SSIM). RPnP-ULA is stable up to $c = 5 \cdot 10^2$, PPnP-ULA is stable up to $c = 3 \cdot 10^2$. For larger c artefacts in the form of stripes gravely appear, see e.g. Figure 7.1(e) and 7.2(e). For both algorithms the performance starts to decrease for $c > 1.5 \cdot 10^2$.

In Figure 7.3, we observe that RPnP-SKROCK is stable as the step size increases and does not show artefacts in its reconstructions. For large step sizes, however, we see significant oversmoothing which is a sign of excessive bias, see e.g. Figure 7.3(d) and 7.3(e). In this case, PSNR and SSIM continue to improve, while LPIPS worsens, quantitatively confirming a loss of detail. The overall performance deteriorates for a step size for which $c > 0.3 \cdot 10^2$. Therefore, we recommend to choose a step size for which $c = 0.02 \cdot 10^2$ or $c = 0.3 \cdot 10^2$ depending on whether objective (PSNR, SSIM) or perceptive quality (LPIPS) is more important. In Figure 7.4, we observe that PnP-MLA recovers fine details without any oversmoothing effect for all the considered step sizes. Performance improves across all metrics as the step size increases, but only negligibly when $c > 5 \cdot 10^2$.

Finally, in Figure 9 we compare quality metrics (PSNR and LPIPS) against neural function evaluations (NFE) averaged over `set3c` using recommended step sizes to assess the convergence speed. We observe that RPnP-SKROCK outperforms all other algorithms in terms of PSNR in performance and convergence speed. In addition, RPnP-ULA and PPnP-ULA show similar performance and convergence speed while they outperform RPnP-SKROCK and PnP-MLA in

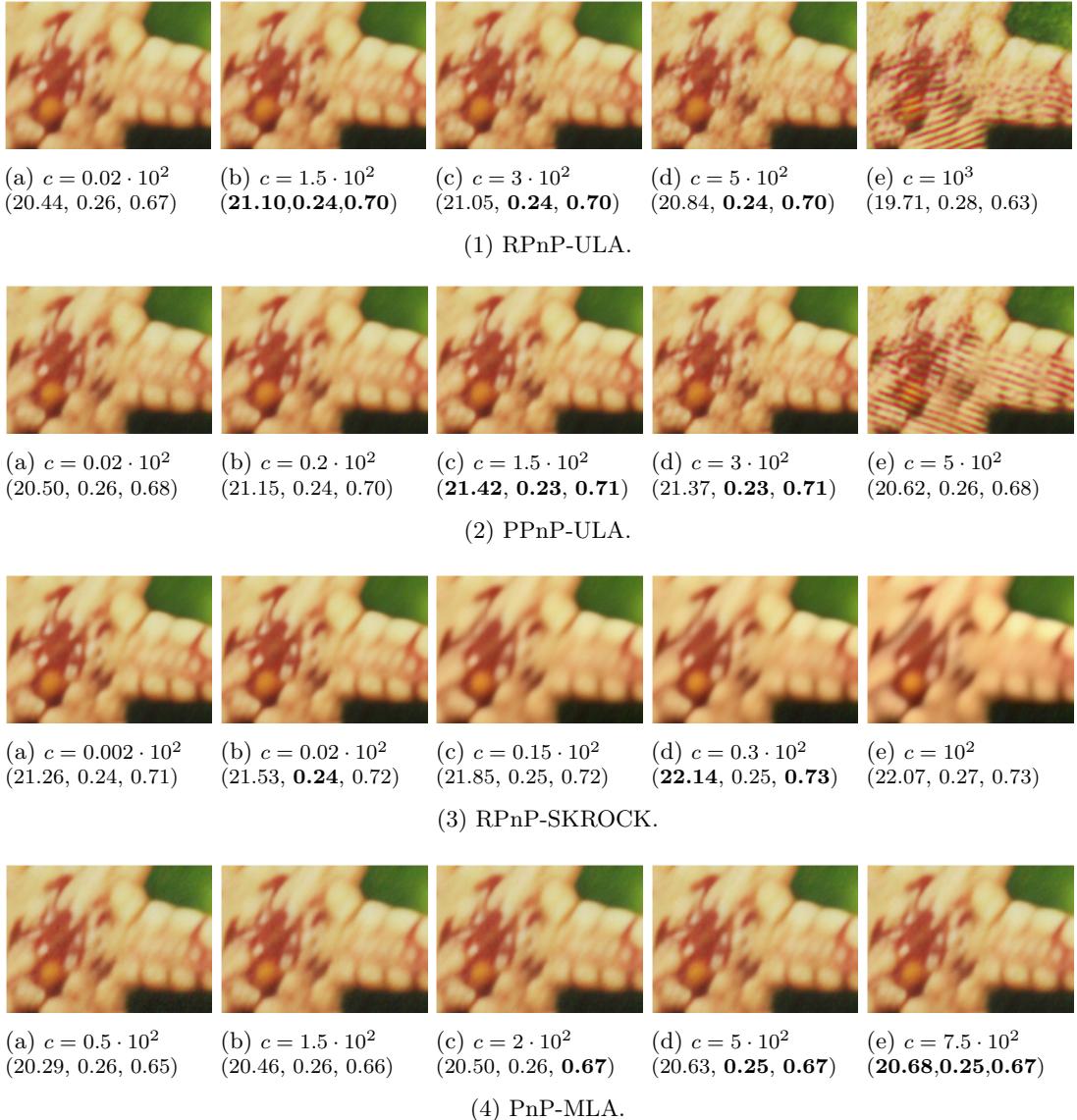


Figure 7: Comparison of MMSE results obtained with different sampling algorithms exemplified on the `starfish` image. The chosen step size affects the results. (PSNR, LPIPS, SSIM) values averaged over the whole `set3c` indicated below each image. Best of each algorithm marked in **bold**.

terms of LPIPS. This occurs due to RPnP-SKROCK and PnP-MLA introducing more bias by taking larger step sizes. PnP-MLA shows slower convergence in terms of PSNR but it performs similarly with the other algorithms in terms of LPIPS with sharp convergence in $\sim 10^4$ NFEs.

4.4 Comparisons with the state-of-the-art

4.4.1 Experimental set up

We are now ready to present comparisons with approaches for Poisson image restoration from the state-of-the-art. These alternative methodologies deliver a single point estimate, rather than a posterior distribution from which point estimates and other forms of inference can be computed. Based on the conclusions of Sections 4.2 and 4.3, we report Bayesian PnP inference

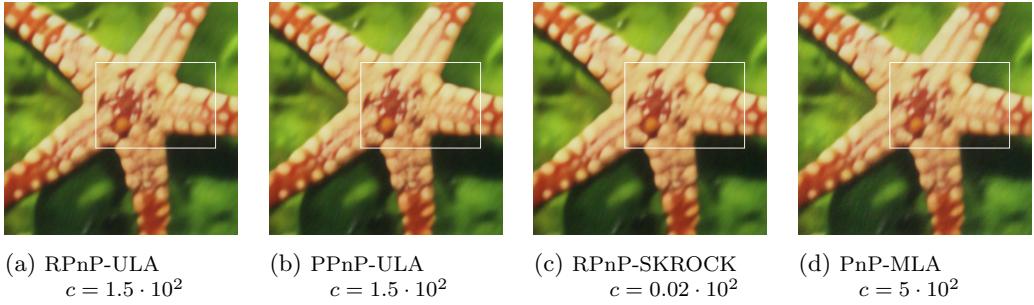


Figure 8: MMSE estimates under the most competitive step sizes for each algorithm.

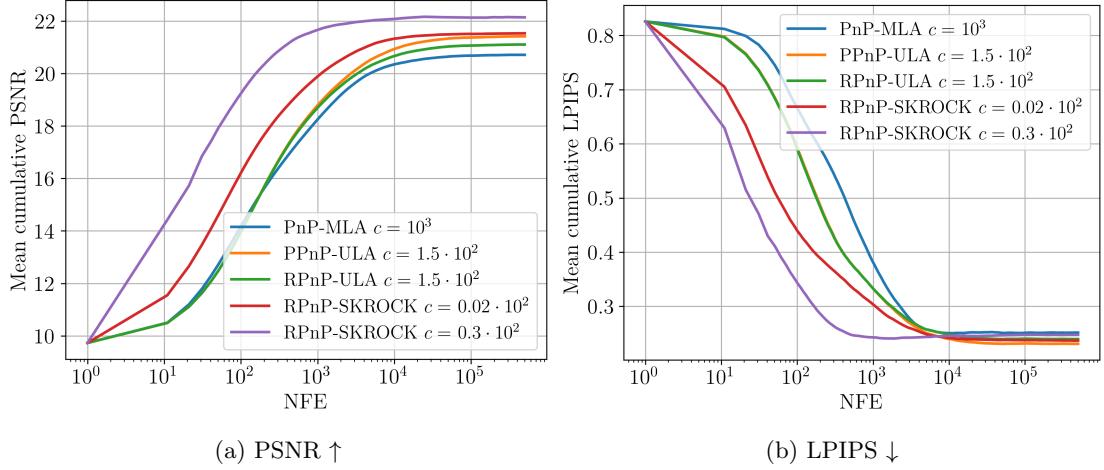


Figure 9: Mean cumulative metrics for the `set3c` data set using selected time steps, with x-axis in log scale to show the tail behaviour more clearly.

with the RPNP-SKROCK (Algorithm 3) and PnP-MLA (Algorithm 4), and we use ProxDRUNet as PnP prior in both cases with $\epsilon = (20/255)^2$ and $\epsilon = (25/255)^2$, respectively. We set $\delta = (0.02 \cdot 10^2) \cdot \ell_s \delta_L$ and $\delta = (5 \cdot 10^2) \cdot \delta_L$ for RPNP-SKROCK and PnP-MLA respectively.

For these comparisons, we consider different shot noise levels and use real-world camera shake kernels extracted from [32], see Figure 2. We present a selection of these experiments with the kernels from Figure 2e and 2j, which we henceforth denote as **Blur 1** and **Blur 2**, respectively, and three different levels of shot noise ($\alpha = 5, 10, 20$). We report comparisons with the PnP-ADMM scheme PIP [52] which uses a patch-based BM3D denoiser, the unrolled network PhD-Net [54], and the Bregman proximal gradient method PnP-BPG [23]. We run our experiments on the **CBSD10** set.

4.4.2 Experimental results

Table 2 summarizes the results for this experiment. We observe that the proposed RPNP-SKROCK and PnP-MLA Bayesian PnP approaches are very competitive in terms of reconstruction PSNR and SSIM, outperforming or performing similarly to the state-of-the-art in all cases. PnP-MLA consistently outperforms all other methods in perceptual quality, as measured by LPIPS, with RPNP-SKROCK following close in performance. With regards to the alternative approaches, as expected, PIP is the least competitive method, as it is based on a patch-based BM3D denoiser. PnP-BPG performs strongly in SSIM, while PhD-Net is competitive in PSNR. However, they are less accurate when errors are measured via LPIPS. Detailed quantitative results for each image of **CBSD10** are depicted in scatter plots in Fig. 17 in Appendix C.

Figure 10 displays a selection of results. We observe that RPNP-SKROCK and PnP-MLA achieve reconstructions with better detail and no colorization or color blocking artefacts; see the

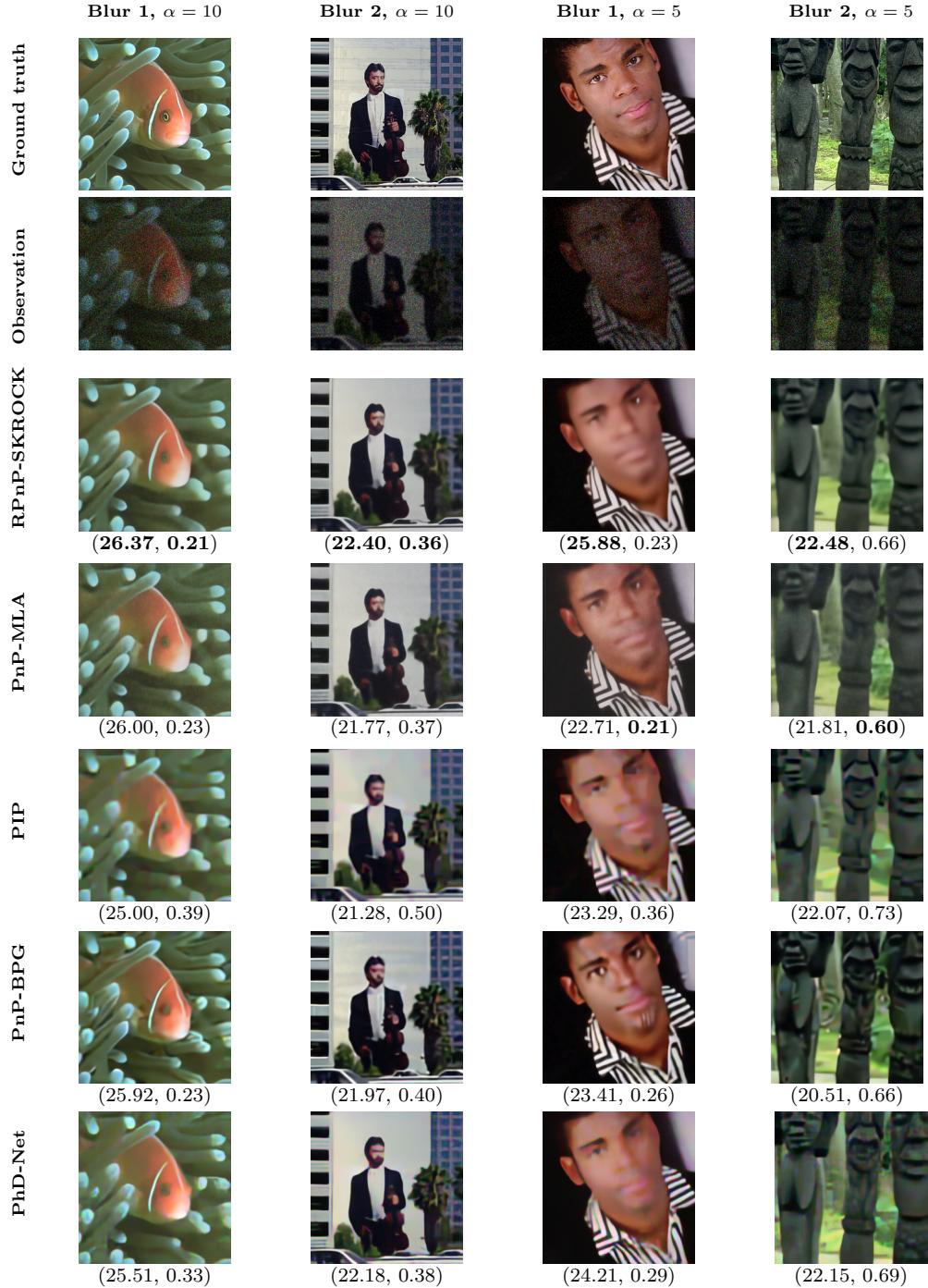


Figure 10: Visual results on CBSD10 dataset for different methods and simulation settings. Associated (PSNR, LPIPS) values are indicated below each image; best marked in **bold**.

second and fourth columns of Figure 10 as an example. Finally, in Figure 11 we present another example image from the CBSD10 dataset. Observe that the Bayesian PnP methods are able to better recover sharp details, and PnP-MLA even recovers some of the texture in the background, whereas the alternative methods struggle to recover perceptual details and textures.



Figure 11: Qualitative results for different algorithms for $\alpha = 10$ and Blur 1. In the second row, we zoomed in a specific area, to give more attention to details. Reported metrics: (PSNR, LPIPS). Best result marked in **bold**.

Level	Kernel	Metrics	RPNP-SKROCK	PnP-MLA	PIP	PnP-BPG	PhD-Net
$\alpha = 5$	Blur 1	PSNR	22.77	21.75	21.94	21.24	<u>22.37</u>
		SSIM	<u>0.57</u>	<u>0.54</u>	<u>0.54</u>	0.57	<u>0.54</u>
		LPIPS	<u>0.52</u>	0.45	0.61	<u>0.52</u>	0.55
	Blur 2	PSNR	22.82	21.81	22.08	21.23	22.60
		SSIM	0.58	0.55	0.55	<u>0.57</u>	<u>0.57</u>
		LPIPS	<u>0.53</u>	0.48	0.61	0.54	0.53
$\alpha = 10$	Blur 1	PSNR	23.42	22.87	22.86	23.38	23.12
		SSIM	<u>0.61</u>	0.58	0.58	0.63	0.58
		LPIPS	<u>0.43</u>	0.40	0.57	0.47	0.52
	Blur 2	PSNR	<u>23.35</u>	22.88	22.96	23.27	23.41
		SSIM	<u>0.61</u>	0.59	0.59	0.63	<u>0.61</u>
		LPIPS	<u>0.47</u>	0.44	0.57	0.50	0.50

Table 2: Quantitative results (averaged over the CBSD10 set). For each quality metric, the best result is shown in **bold** and the second best is underlined.

5 Conclusion

This paper presents two generalizations of the PnP-ULA framework of [28] for Poisson imaging problems: PnP Langevin methods derived from a reflected and regularized overdamped Langevin diffusion, and PnP mirror Langevin methods. We focus on two novel algorithms: RPNP-SKROCK, which is an accelerated PnP Langevin sampling method based on boundary reflections and an approximation of the Poisson likelihood function; and PnP-MLA, the first PnP mirror sampling algorithm in the literature, which exploits a change of geometry to simultaneously deal with the constraints and with the Poisson likelihood without approximations.

Moreover, using Poisson image deblurring as testbed, we compared a wide range of image denoising architectures and algorithmic choices, and concluded that the Prox-DRUNet denoiser is a robust image prior for the considered imaging problems. Extensive numerical experiments show that the proposed sampling methods are particularly relevant in low-photon settings (small α). In such cases, the proposed sampling algorithms outperform PnP optimization-based strategies in reconstruction accuracy, while also providing uncertainty quantification. Among these, PnP-MLA consistently achieves the highest perceptual quality (LPIPS), demonstrating superior detail recovery at low photon levels. This suggests strong potential for applications requiring faithful reconstruction of small structures in sparse photon-starved data. Alternatively, RPnP-SKROCK achieves remarkable MMSE performance (PSNR) with a comparatively low computational cost, requiring in the order of $5 \cdot 10^3$ NFEs or less, while offering a similarly detailed uncertainty quantification. As a general rule, we would recommend PnP-MLA when $\alpha < 20$ (for images with pixel values in the range $[0, 1]$ and assuming A is normalized), and otherwise RPnP-SKROCK.

As mentioned previously, a main reason for constructing Markov kernels such as PnP-MLA and RPnP-SKROCK is to embed them within empirical Bayesian strategies that automatically calibrate unknown model parameters in semi-blind imaging problems (see, e.g., [64, 38]). This remains a main direction for future work. In addition, while in this paper we have focused on relatively conventional PnP denoiser architectures, we see great potential in using a PnP prior derived from a denoising diffusion model (DM) within a Langevin sampling scheme, like in [37]. Extending the DM-based PnP-ULA of [37] to Poisson imaging problems by using either PnP-MLA or RPnP-SKROCK is also an important perspective for future work. Lastly, a highly promising but more challenging direction for further research would be to explore accelerated PnP-MLA sampling, combining the strengths of PnP-MLA and RPnP-SKROCK.

References

- [1] Kwangjun Ahn and Sinho Chewi. Efficient Constrained Sampling via the Mirror-Langevin Algorithm. In *Adv Neural Inf Process Syst*, volume 34, pages 28405–28418, 2021.
- [2] Yoann Altmann, Steve McLaughlin, and Miles Padgett. Unsupervised Restoration of Subsampled Images Constructed from Geometric and Binomial Data. In *IEEE 7th Int'l Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 1–5, 2017.
- [3] Heinz H Bauschke, Jérôme Bolte, and Marc Teboulle. A Descent Lemma Beyond Lipschitz Gradient Continuity: First-order Methods Revisited and Applications. *Math. Oper. Res.*, 42(2):330–348, 2017.
- [4] Amir Beck and Marc Teboulle. Mirror Descent and Nonlinear Projected Subgradient Methods for Convex Optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.
- [5] M Bertero, P Boccacci, G Desiderà, and G Vicidomini. Image Deblurring with Poisson Data: From Cells to Galaxies. *Inverse Probl.*, 25(12):123006, 2009.
- [6] J.P. Burg. *Maximum Entropy Spectral Analysis*. University Microfilms, 1981.
- [7] Ziruo Cai, Junqi Tang, Subhadip Mukherjee, Jinglai Li, Carola-Bibiane Schönlieb, and Xiaoqun Zhang. NF-ULA: Normalizing Flow-Based Unadjusted Langevin Algorithm for Imaging Inverse Problems. *SIAM J. Imaging Sci.*, 17(2):820–860, 2024.
- [8] Niladri Chatterji, Jelena Diakonikolas, Michael I Jordan, and Peter Bartlett. Langevin Monte Carlo Without Smoothness. In *AISTATS*, pages 1716–1726. PMLR, 2020.
- [9] Dongdong Chen, Julián Tachella, and Mike E. Davies. Robust equivariant imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5637–5646, 2022.
- [10] Xiang Cheng, Niladri S. Chatterji, Yasin Abbasi-Yadkori, Peter L. Bartlett, and Michael I. Jordan. Sharp Convergence Rates for Langevin Dynamics in the Nonconvex Setting. *arXiv e-prints*, 2018.
- [11] Florentin Coeurdoux, Nicolas Dobigeon, and Pierre Chainais. Plug-and-play split gibbs sampler: Embedding deep generative priors in bayesian inference. *IEEE Trans. Image Process.*, 33:3496–3507, June 2024.
- [12] A. S. Dalalyan. Theoretical Guarantees for Approximate Sampling from Smooth and Log-Concave Densities. *J. R. Stat. Soc. Ser. B Methodol.*, 79(3):651–676, 2017.
- [13] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk. Single-pixel Imaging via Compressive Sampling. *IEEE Signal Process. Mag.*, 25(2):83–91, 2008.
- [14] A. Durmus, E. Moulines, and M. Pereyra. Efficient Bayesian Computation by Proximal Markov Chain Monte Carlo: When Langevin Meets Moreau. *SIAM J. Imaging Sci.*, 11(1):473–506, 2018.
- [15] Alain Durmus and Éric Moulines. Nonasymptotic Convergence Analysis for the Unadjusted Langevin Algorithm. *Ann. Appl. Probab.*, 27(3):1551 – 1587, 2017.
- [16] Murat A Erdogdu and Rasa Hosseinzadeh. On the Convergence of Langevin Monte Carlo: The Interplay Between Tail Growth and Smoothness. In *COLT*, pages 1776–1822. PMLR, 2021.

- [17] Mário A. T. Figueiredo and José M. Bioucas-Dias. Restoration of Poissonian Images Using Alternating Direction Optimization. *IEEE Trans. Image Process.*, 19(12):3133–3145, 2010.
- [18] Susan Ghaderi, Masoud Ahookhosh, Adam Arany, Alexander Skupin, Panagiotis Patrinos, and Yves Moreau. Smoothing Unadjusted Langevin Algorithms for Nonsmooth Composite Potential Functions. *Appl Math Comput*, 464:128377, 2024.
- [19] Mark Girolami and Ben Calderhead. Riemann Manifold Langevin and Hamiltonian Monte Carlo Methods. *J. R. Stat. Soc. Ser. B Methodol.*, 73(2):123–214, 2011.
- [20] A Hauptmann, S Mukherjee, CB Schönlieb, and F Sherry. Convergent Regularization in Inverse Problems and Linear Plug-and-Play Denoisers. *FoCM*, 2024.
- [21] Thorsten Hohage and Frank Werner. Inverse problems with Poisson data: statistical regularization theory, applications and algorithms. *Inverse Problems*, 32(9):093001, 2016.
- [22] Ya-Ping Hsieh, Ali Kavis, Paul Rolland, and Volkan Cevher. Mirrored Langevin Dynamics. In *Adv Neural Inf Process Syst*, volume 31, 2018.
- [23] Samuel Hurault, Ulugbek Kamilov, Arthur Leclaire, and Nicolas Papadakis. Convergent Bregman Plug-and-Play Image Restoration for Poisson Inverse Problems. In *Adv Neural Inf Process Syst*, volume 36, pages 27251–27280, 2023.
- [24] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Gradient Step Denoiser for convergent Plug-and-Play. In *ICLR*, 2022.
- [25] Samuel Hurault, Arthur Leclaire, and Nicolas Papadakis. Proximal Denoiser for Convergent Plug-and-Play Optimization with Nonconvex Regularization. In *ICML*, volume 162, pages 9483–9505. PMLR, 2022.
- [26] H. Lantéri and C. Theys. Restoration of Astrophysical Images—The Case of Poisson Data with Additive Gaussian Noise. *EURASIP J. Adv. Signal Process.*, page 643143, 2005.
- [27] Tim Tsz-Kit Lau and Han Liu. Bregman Proximal Langevin Monte Carlo via Bregman-Moreau Envelopes. In *PMLR*, volume 162, pages 12049–12077, 2022.
- [28] Rémi Laumont, Valentin De Bortoli, Andrés Almansa, Julie Delon, Alain Durmus, and Marcelo Pereyra. Bayesian Imaging Using Plug & Play Priors: When Langevin Meets Tweedie. *SIAM J. Imaging Sci.*, 15(2):701–737, 2022.
- [29] Marc Lebrun. An Analysis and Implementation of the BM3D Image Denoising Method. *Image Process. Line*, 2:175–213, 2012.
- [30] Joseph Lehec. The Langevin Monte Carlo Algorithm in the Non-Smooth Log-Concave Case. *Ann. Appl. Probab.*, 33(6A):4858 – 4874, 2023.
- [31] B. Leimkuhler and C. Matthews. Rational Construction of Stochastic Numerical Methods for Molecular Sampling. *Appl Math Res Express*, 2013(1):34–56, 2012.
- [32] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and Evaluating Blind Deconvolution Algorithms. In *CVPR*, pages 1964–1971, 2009.
- [33] Ruilin Li, Molei Tao, Santosh S. Vempala, and Andre Wibisono. The Mirror Langevin Algorithm Converges with Vanishing Bias. volume 167, pages 718–742. PMLR, 2022.
- [34] Tobías I Liaudat, Matthijs Mars, Matthew A Price, Marcelo Pereyra, Marta M Betcke, and Jason D McEwen. Scalable Bayesian Uncertainty Quantification With Data-Driven Priors for Radio Interferometric Imaging. *RAS Techniques and Instruments*, 3(1):505–534, 2024.
- [35] Mateusz B. Majka, Aleksandar Mijatović, and Łukasz Szpruch. Nonasymptotic Bounds for Sampling Algorithms Without Log-Concavity. *Ann. Appl. Probab.*, 30(4):1534 – 1581, 2020.

- [36] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. In *ICCV*, volume 2, pages 416–423, 2001.
- [37] Charlesquin Kemajou Mbakam, Jean-Francois Giovannelli, and Marcelo Pereyra. Empirical Bayesian Image Restoration by Langevin Sampling With a Denoising Diffusion Implicit Prior, 2024.
- [38] Charlesquin Kemajou Mbakam, Marcelo Pereyra, and Jean-François Giovannelli. Marginal Likelihood Estimation in Semiblind Image Deconvolution: A Stochastic Approximation Approach. *SIAM J. Imaging Sci.*, 17(2):1206–1254, 2024.
- [39] Savvas Melidonis, Paul Dobson, Yoann Altmann, Marcelo Pereyra, and Konstantinos Zygalakis. Efficient Bayesian Computation for Low-Photon Imaging Problems. *SIAM J. Imaging Sci.*, 16(3):1195–1234, 2023.
- [40] Savvas Melidonis, Yiming Xi, Konstantinos C. Zygalakis, Yoann Altmann, and Marcelo Pereyra. Score-Based Denoising Diffusion Models for Photon-Starved Image Restoration Problems. *TMLR*, 2025.
- [41] Subhadip Mukherjee, Andreas Hauptmann, Ozan Öktem, Marcelo Pereyra, and Carola-Bibiane Schönlieb. Learned Reconstruction Methods With Convergence Guarantees: A Survey of Concepts and Applications. *IEEE Signal Process. Mag.*, 40(1):164–182, 2023.
- [42] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience, 1983.
- [43] M. Pereyra, L. A. Vargas-Mieles, and K. C. Zygalakis. Accelerating Proximal Markov Chain Monte Carlo by Using an Explicit Stabilized Method. *SIAM J. Imaging Sci.*, 13(2):905–935, 2020.
- [44] Jean-Christophe Pesquet, Audrey Repetti, Matthieu Terris, and Yves Wiaux. Learning Maximally Monotone Operators for Image Recovery. *SIAM J. Imaging Sci.*, 14:1206–1237, 2020.
- [45] Andrey Pilipenko. *An Introduction to Stochastic Differential Equations With Reflection*, volume 1. Universitätsverlag Potsdam, 2014.
- [46] Joshua Rapp and Vivek K Goyal. A Few Photons Among Many: Unmixing Signal and Noise for Photon-Efficient Active Imaging. *IEEE Trans. Comput. Imaging*, 3(3):445–459, 2017.
- [47] Behnoor Rasti, Paul Scheunders, Pedram Ghamisi, Giorgio Licciardi, and Jocelyn Chanussot. Noise Reduction in Hyperspectral Imagery: Overview and Application. *Remote Sensing*, 10:482, 03 2018.
- [48] Marien Renaud, Jiaming Liu, Valentin De Bortoli, Andres Almansa, and Ulugbek Kamilov. Plug-and-Play Posterior Sampling under Mismatched Measurement and Prior Models. In *ICLR*, 2024.
- [49] Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer texts in statistics. Springer, New York, NY, November 2010.
- [50] Gareth O. Roberts and Richard L. Tweedie. Exponential Convergence of Langevin Distributions and Their Discrete Approximations. *Bernoulli*, 2(4):341–363, 1996.
- [51] Yaniv Romano, Michael Elad, and Peyman Milanfar. The Little Engine That Could: Regularization by Denoising (RED). *SIAM J. Imaging Sci.*, 10(4):1804–1844, 2017.
- [52] Arie Rond, Raja Giryes, and Michael Elad. Poisson Inverse Problems by the Plug-and-Play Scheme. *J. Vis. Commun. Image Represent.*, 41:96–108, 2016.

- [53] Ernest Ryu, Jialin Liu, Sicheng Wang, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Plug-and-Play Methods Provably Converge with Properly Trained Denoisers. In *ICML*, volume 97, pages 5546–5557. PMLR, 2019.
- [54] Yash Sanghvi, Abhiram Gnanasambandam, and Stanley H. Chan. Photon Limited Non-Blind Deblurring Using Algorithm Unrolling. *IEEE Trans. Comput. Imaging*, 8:851–864, 2022.
- [55] P. Sarder and A. Nehorai. Deconvolution Methods for 3-D Fluorescence Microscopy Images. *IEEE Signal Process. Mag.*, 23(3):32–45, 2006.
- [56] Dongeek Shin, Ahmed Kirmani, Vivek K Goyal, and Jeffrey H. Shapiro. Photon-Efficient Computational 3-D and Reflectivity Imaging With Single-Photon Detectors. *IEEE Trans. Comput. Imaging*, 1(2):112–125, 2015.
- [57] Imraj RD Singh, Alexander Denker, Riccardo Barbano, Željko Kereta, Bangti Jin, Kris Thielemans, Peter Maass, and Simon Arridge. Score-Based Generative Models for PET Image Reconstruction. *MELBA*, 2:547–585, 2024.
- [58] J.-L. Starck and F. Murtagh. *Astronomical Image and Data Analysis (Astronomy and Astrophysics Library)*. Springer-Verlag, Berlin, Heidelberg, 2006.
- [59] Yu Sun, Zihui Wu, Yifan Chen, Berthy T. Feng, and Katherine L. Bouman. Provable Probabilistic Imaging Using Score-Based Generative Priors. *IEEE Trans. Comput. Imaging*, 10:1290–1305, 2024.
- [60] Julián Tachella, Mike Davies, and Laurent Jacques. UNSURE: Self-Supervised Learning With Unknown Noise Level and Stein’s Unbiased Risk Estimate. In *ICLR*, 2025.
- [61] Hong Ye Tan, Ziruo Cai, Marcelo Pereyra, Subhadip Mukherjee, Junqi Tang, and Carola-Bibiane Schönlieb. Unsupervised Training of Convex Regularizers using Maximum Likelihood Estimation. *TMLR*, 2024.
- [62] Hong Ye Tan, Subhadip Mukherjee, Junqi Tang, and Carola-Bibiane Schönlieb. Provably Convergent Plug-and-Play Quasi-Newton Methods. *SIAM J. Imaging Sci.*, 17(2):785–819, 2024.
- [63] Matthieu Terris, Thomas Moreau, Nelly Pustelnik, and Julian Tachella. Equivariant Plug-and-Play Image Reconstruction. In *CVPR*, pages 25255–25264, 2024.
- [64] Ana Fernandez Vidal, Valentin De Bortoli, Marcelo Pereyra, and Alain Durmus. Maximum Likelihood Estimation of Regularization Parameters in High-Dimensional Inverse Problems: An Empirical Bayesian Approach Part I: Methodology and Experiments. *SIAM J. Imaging Sci.*, 13(4):1945–1989, 2020.
- [65] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.
- [66] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-Play Image Restoration With Deep Denoiser Prior. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(10_Part_1):6360–6376, 2022.
- [67] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.*, 26(7):3142–3155, 2017.
- [68] Kelvin Shuangjian Zhang, Gabriel Peyré, Jalal Fadili, and Marcelo Pereyra. Wasserstein Control of Mirror Langevin Monte Carlo. In *COLT*, pages 3814–3841. PMLR, 2020.

- [69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, pages 586–595, 2018.
- [70] Qingping Zhou, Tengchao Yu, Xiaoqun Zhang, and Jinglai Li. Bayesian Inference and Uncertainty Quantification for Medical Image Reconstruction With Poisson Data. *SIAM J. Imaging Sci.*, 13(1):29–52, 2020.

A Additional information for Section 4.2

Additional results. We present results based on the `starfish` image from `set3c`, to complement the results for the `leaves` image presented in Section 4.2. Figure 3c presents the ground truth and Figure 12a presents a realization y generated by the forward model Eq. (1) with operator A modeling the motion blur kernel in Figure 2e, while the photon level of the Poisson noise is set to $\alpha = 20$.

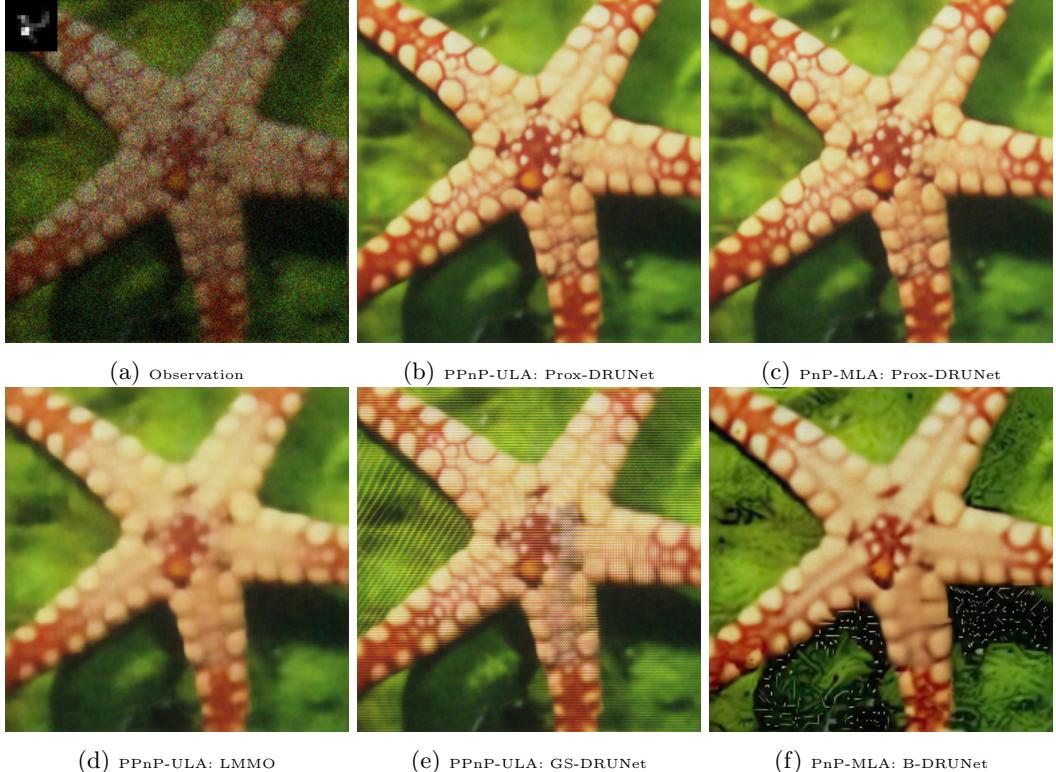


Figure 12: Poisson deconvolution problem for $\alpha = 20$ and Blur 1 using the `starfish` image. Noisy and blurry observation, and the MMSE reconstructions \hat{x}_{MMSE} for Prox-DRUNet, LMMD, GS-DRUNet and B-DRUNet using PPnP-ULA and PnP-MLA.

Figures 12b to 12f show the posterior mean \hat{x}_{MMSE} , as calculated for the different denoisers and different algorithms using 10^6 iterations to stress-test the stability of the algorithms. Prox-DRUNet proved to be the most competitive in terms of stability and performance among all the considered denoisers. More specifically, we observe that PPnP-ULA with the LMMD denoiser produces an estimate of the mean with excessive smoothing, while with the GS-DRUNet exhibits checkerboard-like and stripe-like reconstruction artefacts on large parts of the `starfish` and the background. The B-DRUNet denoiser progressively develops artefacts after approximately 10^5 iterations which deteriorate gravely over time, see Figure 12f.

Table 3 summarizes the performance of the denoisers for the considered image. The LMMD prior leads to faster convergence but achieves low reconstruction quality, while GS-DRUNet is not stable, so the reconstruction quality decreases as the iterations progress and the artefacts become more pronounced. Similarly, B-DRUNet also leads to instability and reaches its top performance in around 10^5 iterations before artefacts start to amplify. Conversely, Prox-DRUNet outperforms the other networks in all the image quality metrics considered. For completeness, Table 3 also reports the number of iterations that PPnP-ULA and PnP-MLA require to reach 98% of peak PSNR performance, as an indicator of convergence speed for the posterior mean. Convergence speed is comparable between PPnP-ULA and PnP-MLA using Prox-DRUNet. Note that the Prox-DRUNet yields the best accuracy-speed trade-off, reaching almost top performance at

Method	Denoiser	PSNR	SSIM	LPIPS	Iter. until 98% PSNR
PPnP-ULA	LMMO	21.63	0.60	0.41	11500
	GS-DRUNet	18.68	0.46	0.41	12000
	Prox-DRUNet	23.54	0.69	0.29	25000
PnP-MLA	Prox-DRUNet	23.68	0.69	0.29	33000
	B-DRUNet	19.53	0.49	0.47	111500

Table 3: Quantitative results for \hat{x}_{MMSE} (calculated by PPnP-ULA and PnP-MLA) for the **starfish** image. Last column: The number of iterations required such that 98% PSNR is reached.

approximately $3 \cdot 10^4$ iterations for PPnP-ULA and PnP-MLA. We point out that for PPnP-ULA similar quantitative behavior was observed for the different denoisers even when equivariance was used, as well as when using Algorithms 1 and 3. We do not report results for PnP-MLA without randomization as this technique was necessary to stabilize PnP-MLA when using B-DRUNet.

Figure 13 shows the pixel-wise posterior standard deviation, as calculated with the LMMO, Prox-DRUNet, GS-DRUNet and B-DRUNet denoisers. For reference, next to each posterior standard deviation plot, we also report the residual obtained by comparing \hat{x}_{MMSE} to the true image (note that these standard deviations represent the models' marginal predictions for these residuals, at the pixel level). We observe that LMMO and Prox-DRUNet produce uncertainty plots that are broadly in agreement with their residuals. For Prox-DRUNet and LMMO, uncertainty concentrates around edges and contours, as expected for a deconvolution problem, whereas the uncertainty estimates of GS-DRUNet are aligned with its reconstruction artefacts. Conversely, B-DRUNet produces uncertainty plots that highlight homogenous regions, and which do not align well with its residual, and has particularly high uncertainty in the areas where artefacts develop. We conclude that Prox-DRUNet is the most appropriate denoising architecture for Bayesian PnP inference in Poisson image deblurring problems regardless of the sampling algorithm.

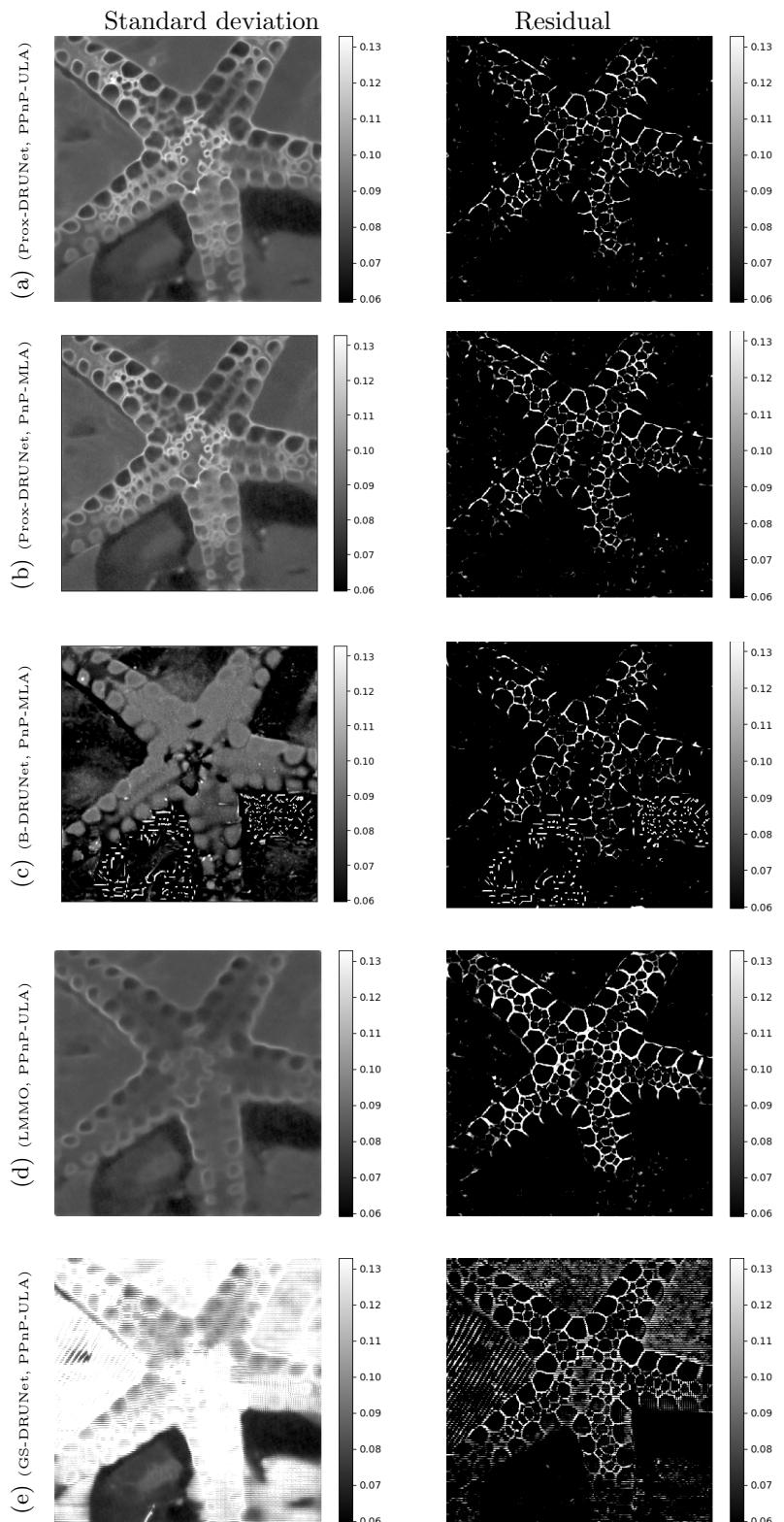


Figure 13: Poisson deconvolution problem as in Fig. 12. Pairwise: Pixelwise standard deviations (left column) and residuals (right column) for all tested denoisers using either PPnP-ULA or PnP-MLA.

B Additional information for Section 4.3

B.1 Additional details for MMSE reconstructions of `set3c`

To support the results in Figure 7, we include here the quantitative results averaged over `set3c` in Tables 4-7.

c	PSNR	LPIPS	SSIM	Stability
$0.02 \cdot 10^2$	20.44	0.26	0.67	✓
$0.1 \cdot 10^2$	20.95	0.25	0.70	✓
$0.2 \cdot 10^2$	21.02	0.25	0.70	✓
$1.5 \cdot 10^2$	21.10	0.24	0.70	✓
$3 \cdot 10^2$	21.05	0.24	0.70	✓
$5 \cdot 10^2$	20.84	0.24	0.70	✓
10^3	19.71	0.28	0.63	✗

Table 4: Quantitative results for RPnP-ULA under different step sizes on `set3c`.

c	PSNR	LPIPS	SSIM	Stability
$0.02 \cdot 10^2$	20.50	0.26	0.68	✓
$0.1 \cdot 10^2$	21.04	0.24	0.70	✓
$0.2 \cdot 10^2$	21.15	0.24	0.70	✓
$1.5 \cdot 10^2$	21.42	0.23	0.71	✓
$3 \cdot 10^2$	21.37	0.23	0.71	✓
$5 \cdot 10^2$	20.62	0.26	0.68	✗
10^3	19.36	0.30	0.61	✗

Table 5: Quantitative results for PPnP-ULA under different step sizes on `set3c`.

c	PSNR	LPIPS	SSIM	Stability	Smoothing
$0.002 \cdot 10^2$	21.26	0.24	0.71	✓	-
$0.01 \cdot 10^2$	21.44	0.24	0.71	✓	-
$0.02 \cdot 10^2$	21.53	0.24	0.72	✓	-
$0.15 \cdot 10^2$	21.85	0.25	0.72	✓	-
$0.3 \cdot 10^2$	22.14	0.25	0.73	✓	✓
$0.5 \cdot 10^2$	22.05	0.26	0.72	✓	✓
10^2	22.07	0.27	0.73	✓	✓

Table 6: Quantitative results for RPnP-SKROCK under different step sizes on `set3c`.

c	PSNR	LPIPS	SSIM	Stability
$0.1 \cdot 10^2$	19.74	0.26	0.60	✓
$0.5 \cdot 10^2$	20.29	0.26	0.65	✓
10^2	20.40	0.26	0.66	✓
$1.5 \cdot 10^2$	20.46	0.26	0.66	✓
$2 \cdot 10^2$	20.50	0.26	0.67	✓
$5 \cdot 10^2$	20.63	0.25	0.67	✓
$7.5 \cdot 10^2$	20.68	0.25	0.67	✓
10^3	20.71	0.25	0.68	✓

Table 7: Quantitative results for PnP-MLA under different step sizes for $\epsilon = 25$ on `set3c`.

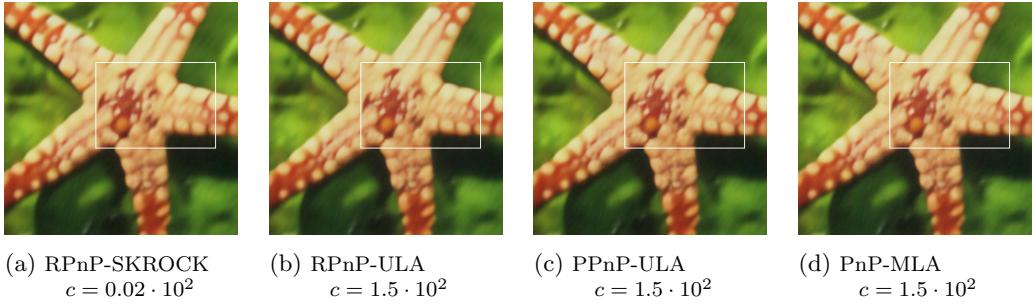


Figure 14: Overview of best qualitative MMSE results for all algorithms with $\epsilon = 20$.

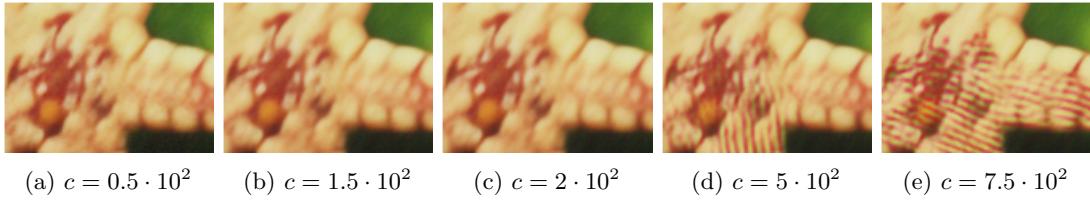


Figure 15: PnP-MLA with $\epsilon = 20$.

c	PSNR	LPIPS	SSIM	Stability
0.1 · 10²	20.22	0.23	0.62	✓
0.5 · 10²	21.06	0.23	0.69	✓
1 · 10²	21.20	0.24	0.70	✓
1.5 · 10²	21.27	0.24	0.70	✓
2 · 10²	21.31	0.24	0.70	✓
5 · 10²	21.36	0.24	0.71	~
7.5 · 10²	21.18	0.24	0.70	~

Table 8: PnP-MLA, mean over `set3c`, $\epsilon = 20$. Instability shows through artefacts (stripes) that appear in one of the images in the data set, which increase in strength as the step size increases.

B.2 Additional results for PnP-MLA using Prox-DRUNet with $\epsilon = 20$

Here, additional results to Section 4.3 are presented. We use a detail of the `starfish` image shown in Figure 14 to illustrate the effects of the chosen step size. Results for the PnP-MLA using $\epsilon = 20$ are summarized in Table 8, yielding similar results to PPnP-ULA in Table 5 in terms of best metric and step size, both quantitatively and qualitatively (compare Figures 7.2 and 15, with the best results in Figure 15b). With increasing step size the algorithm shows instability by introducing artefacts that only appear in the `starfish` image (not in any other image in `set3c` - not shown here), see Figure 15d and 15e. We find that by choosing $\epsilon = 25$ for the neural network denoiser (these results are described in the main text for Section 4.3), and thus increasing the denoising strength slightly, all artefacts can be mitigated.

We compare the convergence speed for selected step sizes in Figure 16. We compute the cumulative metrics (depicted here are PSNR and LPIPS) against neural function evaluations (NFE) and take the mean over the `set3c` data set. We observe that in terms of convergence speed RPnP-SKROCK outperforms RPnP-ULA and PPnP-ULA (which are fairly similar in speed) and the PnP-MLA for both chosen step sizes. Furthermore, the behaviour in terms of PSNR is rather monotonic as NFEs increase, but this is not necessarily the case for LPIPS with the most visible non-monotonic behaviour occurring for PnP-MLA with the best LPIPS obtained overall around 10^4 NFEs.

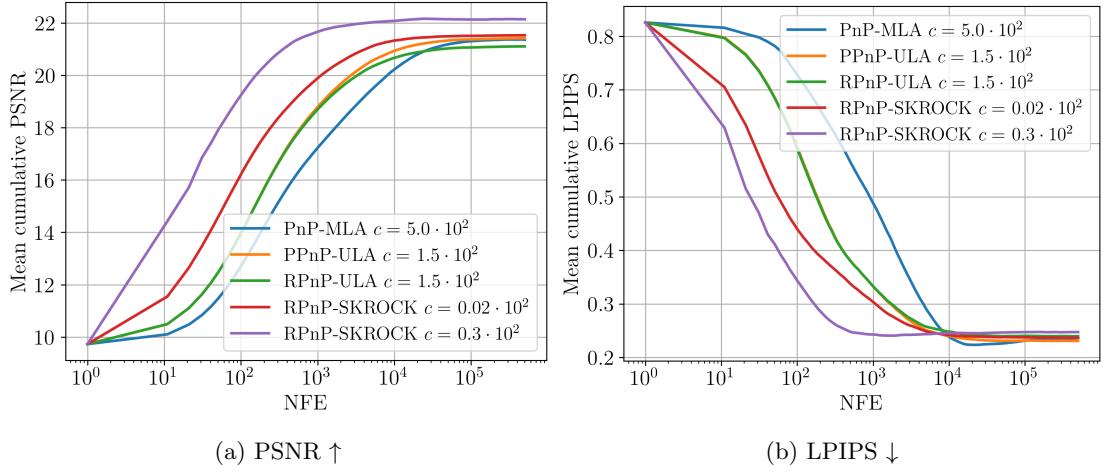


Figure 16: Mean cumulative metrics for the `set3c` data set using selected time steps ($\epsilon = 20$ for all algorithms).

C Additional information for Section 4.4

C.1 Scatter plots

The plots in Figure 17 depict detailed results of each image in the data set used for the experiments in Section 4.4. Each symbol denotes a different image, and each algorithm is depicted in a distinct color.

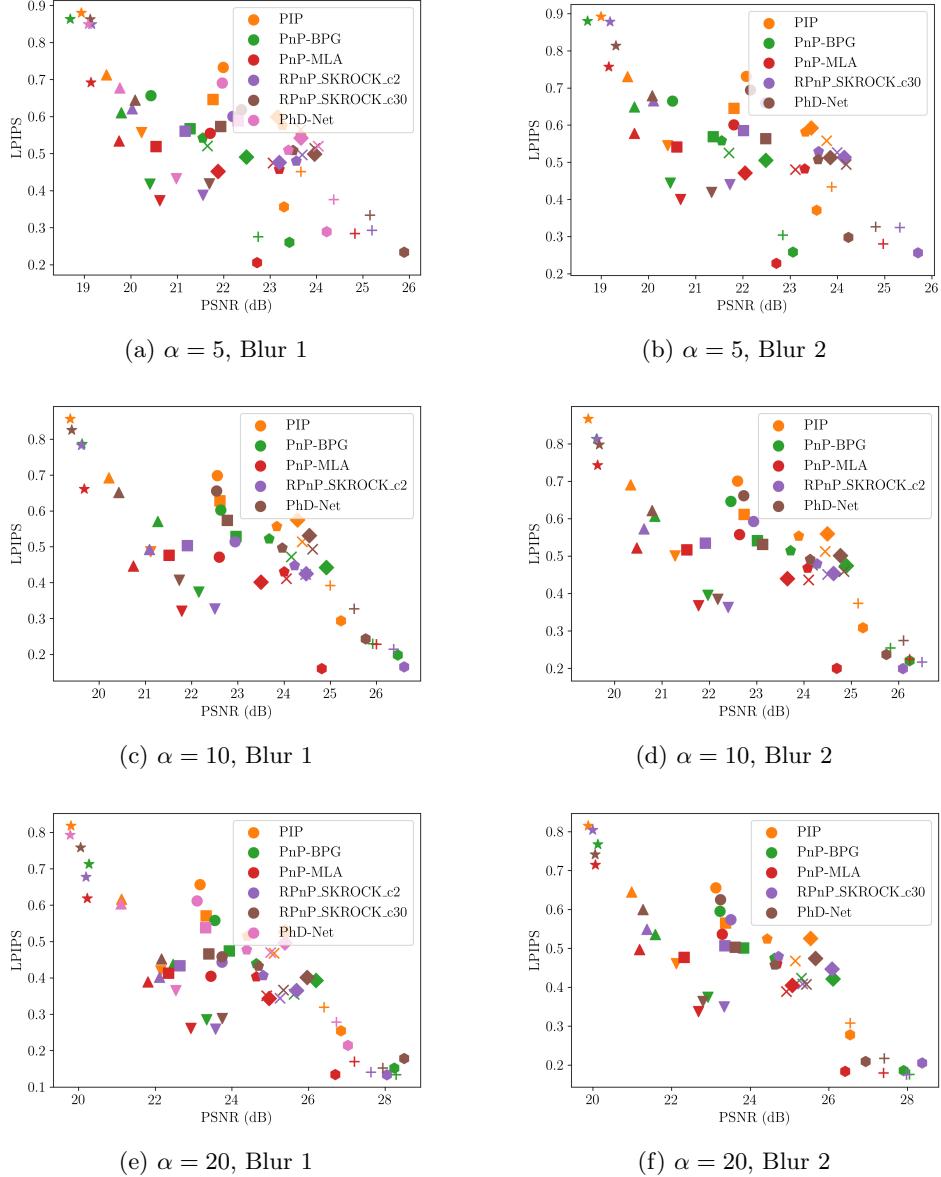


Figure 17: Scatter plots to support Section 4.4. Each marker marks a distinct image, the colors correspond to different methods. On the LPIPS axis, lower is better; on the PSNR axis, higher is better.