



HAL
open science

SPyRiT: an open source package for single-pixel imaging based on deep learning

Juan F P J Abascal, Thomas Baudier, Romain Phan, Audrey Repetti, Nicolas Ducros

► To cite this version:

Juan F P J Abascal, Thomas Baudier, Romain Phan, Audrey Repetti, Nicolas Ducros. SPyRiT: an open source package for single-pixel imaging based on deep learning. 2024. hal-04662876v3

HAL Id: hal-04662876

<https://hal.science/hal-04662876v3>

Preprint submitted on 16 Oct 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SPyRiT: an open source package for single-pixel imaging based on deep learning

Juan FJP Abascal,[†] T. Baudier,[†] R. Phan,[†] A. Repetti,[‡] and N. Ducros^{*,†,¶}

[†]*Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM-Saint Etienne, CNRS,
Inserm, CREATIS UMR 5220, U1294, F-69621, LYON, France*

[‡]*MACS & EPS, Heriot-Watt University, Maxwell Institute for Mathematical Sciences,
Edinburgh, UK*

[¶]*Institut universitaire de France (IUF), France*

E-mail: nicolas.ducros@creatis.insa-lyon.fr

Abstract

Single-pixel imaging is able to acquire an image from a few point measurements thanks to dedicated reconstruction algorithms. In recent years, reconstruction approaches based on deep learning have outperformed most alternatives. However, computational experiments and data-driven methods have become difficult, if not impossible, to reproduce. The development of tools enabling reproducibility and benchmarking is therefore now essential. This paper describes SPyRiT, an open source PyTorch-based toolbox capable of handling various simulation configurations and reconstruction methods based on deep learning. In particular, we compare several supervised and plug-and-play methods, including post-processing and iterative strategies. Our results demonstrate that supervised methods trained on simulated data can be successfully applied to experimental data when the signal-to-noise ratio of the measurements is higher or equal to that of the training phase. On the other hand, the hyperparameter of the plug-and-play methods can be tuned to manage lower signal-to-noise ratios. The

16 modularity of SPyRiT enables the evaluation of various configurations and the rigorous
17 benchmarking of reconstructions based on deep learning in single-pixel imaging, as well
18 as in most computational optics problems.

19 **Introduction**

20 Single-pixel imaging is a technique that enables an image to be acquired from a set of point
21 measurements using dedicated reconstruction algorithms.^{1,2} The technique can be traced
22 back to the concept of Hadamard spectroscopy,³ that introduced Hadamard-modulated mea-
23 surements to enhance the signal-to-noise ratio (SNR) of images reconstructed using the least
24 squares method. Single-pixel imaging has gained renewed interest with the advent of com-
25 pressed sensing theory, providing a theoretical framework to ensure perfect reconstruction
26 of an image acquired from few measurements obtained with random modulations.⁴ The
27 many different applications of the technique include imaging in the visible and infrared do-
28 mains,⁵ in addition to the mid-infrared,⁶ ultraviolet,⁷ terahertz⁸ and X-ray⁹ ranges, as well
29 as through scattering media,¹⁰ fluorescence lifetime imaging,¹¹ 3D time-of-flight imaging,¹²
30 time-of-flight LiDAR¹³ and light-field microscopy.¹⁴

31 Deep learning has revolutionized the field of image reconstruction¹⁵ and greatly ad-
32 vanced computational optics.¹⁶ Since the first application of deep learning to single-pixel
33 imaging,¹⁷ a great variety of data-driven reconstruction methods have been designed for
34 the field. Supervised methods train the free parameters of a reconstruction algorithm from
35 image-measurement pairs. Simple yet efficient reconstruction algorithms are obtained by
36 post-processing a first (e.g., linear) reconstruction by an image-domain neural network (e.g.,
37 a U-Net¹⁸ or an attention network¹⁹). The first reconstruction step can be learned from the
38 data in an end-to-end¹⁷ or two-step²⁰ manner, chosen as the pseudo-inverse solution,^{21,22} or
39 the minimum mean squares error solution.²³ Following the trend of algorithm unrolling,^{24,25}
40 an empirical expectation maximisation network was designed in²⁶ while²⁷ considered un-
41 rolling the iterative hard thresholding for expanders. Adversarial training has also been

42 considered.^{28,29} To improve the generalization ability of supervised methods, one strategy
43 involves fine-tuning using a physics-based loss.³⁰ An alternative is to consider plug-and-play
44 (PnP) methods that combine a (deep) denoiser prior with a model-based reconstruction
45 algorithm. Multiple configurations (e.g., sampling ratio, noise level, type of modulation)
46 can, therefore, be considered using the same denoiser, whereas supervised methods require
47 retraining. Among the possible model-based reconstruction algorithms that can be used for
48 PnP, the proximal gradient descent³¹ or the alternating direction method of multipliers al-
49 gorithm³² have been widely used in imaging. Different types of denoisers can be considered,
50 such as traditional (e.g., BM3D³³) or U-Net based denoisers (e.g., U-Net,¹⁸ DR-UNet³⁴),
51 or auto-regressive models.³⁵ More recently, untrained networks have emerged that do not
52 require pretraining.^{36,37}

53 In this new data-driven deep learning era, reproducibility is a major challenge. Indeed,
54 despite huge progress in both the practical and theoretical aspects, most computational
55 experiments and data-driven methods have become difficult, if not impossible, to reproduce
56 by an independent researcher,³⁸ an issue referred to as the *reproducibility crisis*. Open source
57 platforms with reference datasets and reconstruction algorithms are therefore required in
58 order to ensure the reproducibility of results and fair benchmarking needed to establish
59 reliable conclusions.

60 This work introduces SPyRiT, the first, to the best of our knowledge, package dedicated
61 to single-pixel imaging reconstruction based on deep learning. Though packages for solv-
62 ing general inverse problems using deep learning are available (e.g., Pixu³⁹ or DeepInv⁴⁰),
63 dedicated packages are usually preferred when targeting specific applications (e.g., RTK,⁴¹
64 ASTRA⁴² or TIGRE⁴³ in computed tomography; BART⁴⁴ in magnetic resonance imaging).
65 SPyRiT, on the other hand, provides a platform for the simulation and reconstruction of
66 single-pixel measurements, integrated within the OpenSpyrit ecosystem that provides data
67 and hardware control software.⁴⁵ The modular structure of the Python-based package al-
68 lows users to simulate measurements from images, simulate corrupted noisy measurements,

69 process noisy measurements prior to reconstruction, define data-driven reconstruction algo-
 70 rithms and train the associated networks.

71 To illustrate the capabilities of SPyRiT, we implement and compare six different data-
 72 driven reconstruction methods, belonging to the supervised and PnP families of algorithms.
 73 All methods are assessed in a controlled manner by evaluation of the images reconstructed
 74 by SPyRiT from the ImageNet dataset and SPIHIM experimental dataset. While SPyRiT
 75 has been designed for single-pixel imaging, it may also benefit other computational optical
 76 imaging modalities, e.g., ghost imaging which is based on the same imaging principle.

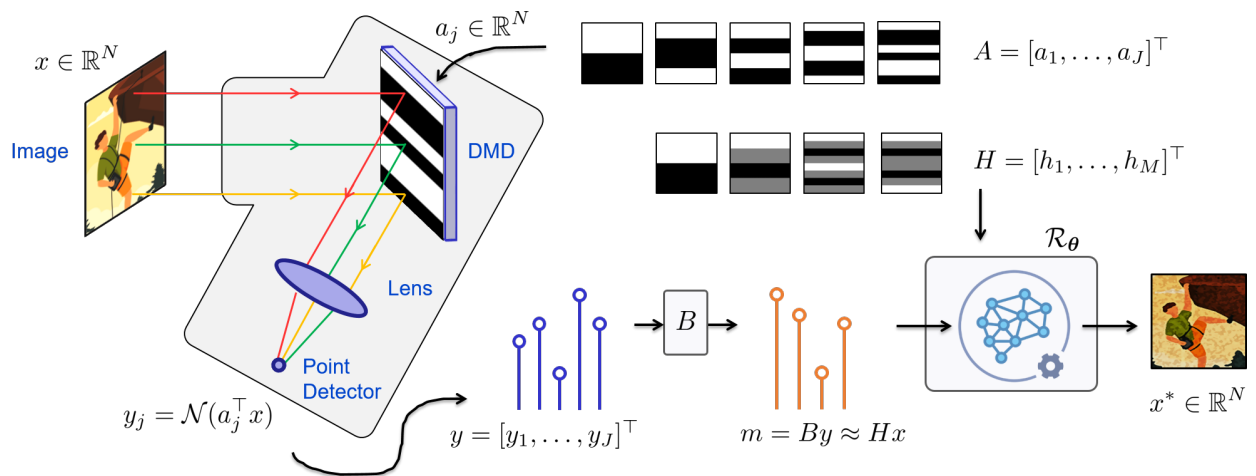


Figure 1: Principle of single-pixel imaging. A single-pixel camera measures a noisy version of the scalar product of some image $x \in \mathbb{R}^N$ and a DMD pattern $a_j \in \mathbb{R}^N$. The measurement vector $y \in \mathbb{R}^J$, that collects all measurements obtained for all DMD patterns, may be preprocessed, leading to the preprocessed measurement vector $m \in \mathbb{R}^M$. A reconstruction algorithm \mathcal{R}_θ is necessary to estimate the unknown image x from the preprocessed measurements.

77 Theory

78 Single-pixel imaging forward model

79 Single-pixel imaging aims to recover an unknown image $x \in \mathbb{R}^N$ from a few noisy observations

$$m \approx Hx, \tag{1}$$

80 where $H: \mathbb{R}^N \rightarrow \mathbb{R}^M$ is a linear measurement operator. In the case when $M = N$, it may be
 81 desirable to work with orthogonal bases, such as Hadamard or Fourier^{46,47} to benefit from
 82 Fellgett’s advantage for improved SNR.³ This implies patterns with negative values¹. In
 83 practice, however, measurements are obtained by uploading a set of light patterns onto a
 84 spatial light modulator (e.g., a digital micromirror device (DMD), see Fig. 1). Therefore,
 85 only positive patterns can be implemented. We model the actual acquisition process as

$$y = \mathcal{N}(Ax), \quad (2)$$

86 where $\mathcal{N}: \mathbb{R}^J \rightarrow \mathbb{R}^J$ represents a (not necessarily linear) noise operator (e.g., Poisson or
 87 Poisson-Gaussian), $A: \mathbb{R}^N \rightarrow \mathbb{R}^J$ is the actual acquisition operator that models the (positive)
 88 DMD patterns, and J is the number of DMD patterns.

89 **Handling non negativity with preprocessing**

90 We may consider preprocessing techniques that transform the actual model Eq. (2) into the
 91 target model Eq. (1)

$$m = By \approx Hx, \quad (3)$$

92 where $B: \mathbb{R}^J \rightarrow \mathbb{R}^M$ is the (linear) preprocessing operator chosen such that $BA = H$. Note
 93 that the noise of the preprocessed measurements $m = By$ is not the same as that of the
 94 actual measurements y .

95 In the simple case when H has only non-negative entries, we can choose $A = H$ and
 96 B is the identity matrix. There exist multiple variants to build the preprocessing operator
 97 such that $BA = H$ when H has negative entries (e.g., see⁴⁸). A simple workaround consists
 98 in splitting the positive and negative entries of the matrix. Alternatively, one can add an

¹An orthogonal matrix with non-negative entries is a permutation matrix, i.e., a matrix that has only one non zero entry (equal to one) in each row. This case corresponds to measuring each pixel in the scene independently

99 appropriate constant to the patterns, which usually leads to increased noise.

100 **Data-driven reconstruction**

101 In the presence of noise and when fewer measurements than unknowns are available (i.e.,
 102 $M < N$), regularized approaches must be adopted (see Appendix). Data-driven methods
 103 based on deep learning are a modern regularized approach. They aim to find an estimate
 104 $x^* \in \mathbb{R}^N$ of the unknown image x from the preprocessed measurements By (see Eq. (3)),
 105 using a reconstruction operator $\mathcal{R}_{\theta^*}: \mathbb{R}^M \rightarrow \mathbb{R}^N$. Specifically,

$$\mathcal{R}_{\theta^*}(m) = x^* \approx x, \quad (4)$$

106 where θ^* represents the parameters learned during a training procedure.

107 In the case of supervised learning, it is assumed that a training dataset $\{x_i, y_i\}_{1 \leq i \leq I}$ of
 108 pairs of ground truth images and measurements is available. θ^* is then obtained by solving

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^I \mathcal{L}(x_i, \mathcal{R}_{\theta}(y_i)), \quad (5)$$

109 where \mathcal{L} is the training loss (e.g., squared error). It should be noted that if only a dataset
 110 of ground truth images $\{x_i\}_{1 \leq i \leq I}$ is available, the associated set of measurements can be
 111 obtained by defining $y_i = \mathcal{N}(Ax_i)$, $1 \leq i \leq I$.

112 The design of the reconstruction operator \mathcal{R}_{θ} has been the topic of extensive research
 113 over the last decade. A simple yet efficient method consists in correcting a traditional (e.g.
 114 linear) reconstruction by a data-driven nonlinear step.⁴⁹ In this context,

$$\mathcal{R}_{\theta} = \mathcal{G}_{\theta} \circ \mathcal{R}, \quad (6)$$

115 where $\mathcal{R}: \mathbb{R}^M \rightarrow \mathbb{R}^N$ is a traditional hand-crafted (e.g., regularized) reconstruction operator
 116 and $\mathcal{G}_{\theta}: \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a nonlinear neural network that acts in the image domain. Algorithm

117 unfolding can be seen as a generalization of Eq. (6), which consists in defining \mathcal{R}_θ from an
 118 iterative scheme

$$\mathcal{R}_\theta = \mathcal{R}_{\theta_K} \circ \dots \circ \mathcal{R}_{\theta_1}, \quad (7)$$

119 where \mathcal{R}_{θ_k} can be interpreted as the computation of the k -th iteration of the iterative scheme
 120 and $\theta = \bigcup_k \theta_k$. Many variants of unfolded algorithms have been proposed in the literature.²⁵
 121 The choice of \mathcal{R}_{θ_k} depends on the choice of the algorithm that is unrolled. It generally involves
 122 one or several analytical steps that rely on A or H , their adjoint operators (or pseudo-inverse)
 123 as well as one or several neural networks. Some standard architectures for \mathcal{R}_{θ_k} are described
 124 in Section . In particular, the formulation of Eq. (7) encompasses PnP and regularization
 125 by denoising methods, for which the neural networks/denoisers involved in \mathcal{R}_{θ_k} are trained
 126 beforehand, independently of A or H .

127 **Standard data-driven architectures**

128 A simple choice for the first step of the post-processing approach defined by Eq. (6) is to
 129 use the pseudo-inverse of H , i.e.,

$$\mathcal{R}(m) = H^\dagger m. \quad (8)$$

130 An alternative is to use the minimum mean square estimator,⁵⁰ i.e.,

$$\mathcal{R}(m) = \Sigma H^\top (H \Sigma H^\top + \Gamma)^{-1} m, \quad (9)$$

131 where $\Sigma \in \mathbb{R}^{N \times N}$ represents the image covariance and $\Gamma \in \mathbb{R}^{M \times M}$ the noise covariance.
 132 These two methods are further detailed in Appendix and Appendix .

133 A simple choice for the iterative method described in Eq. (7) is the proximal gradient
 134 descent⁵¹ for solving the regularized least-squares problem (see Appendix),

$$x_{k+1} = \mathcal{R}_{\theta_k}(x_k) = \mathcal{G}_{\theta_k}(x_k - \gamma_k H^\top (H x_k - m)), \quad (10)$$

135 where the proximal denoiser has been replaced by a neural network $\mathcal{G}_{\theta_k} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ acting
136 in the image domain and $\gamma_k > 0$ is the step size.

137 Method

138 Overview of the SPyRiT package

139 SPyRiT has a modular structure with the core functionalities organized in six submodules
140 (see Fig. 2) that provide classes that all inherit from PyTorch `nn.Module`. The module
141 `spyrit.core.torch` provides additional auxiliary PyTorch-based functions used throughout
142 the SPyRiT package.

- 143 • `spyrit.core.meas` provides measurement operators that compute linear measure-
144 ments corresponding to A in Eq. (2). It also provides the adjoint and pseudo-inverse
145 of A , which are at the basis of any reconstruction algorithm.
- 146 • `spyrit.core.noise` provides noise operators corresponding to \mathcal{N} in Eq. (2).
- 147 • `spyrit.core.prep` provides preprocessing operators for the operator B introduced in
148 Eq. (3). These preprocessing operators can also implement additional processing steps,
149 such as the normalization of Poisson-corrupted images that vary across a wide range
150 of intensities.
- 151 • `spyrit.core.nnet` provides neural network operators that correspond to \mathcal{G} in Eq. (6)
152 or Eq. (10).
- 153 • `spyrit.core.recon` provides the reconstruction operator that correspond to \mathcal{R}_θ .
- 154 • `spyrit.core.train` provides the functionalities to solve the minimization problem of
155 Eq. (5).

156 **Poisson-corrupted split Hadamard measurements**

157 To illustrate the capabilities of SPyRiT, we consider Poisson-corrupted split Hadamard mea-
 158 surements

$$y_\alpha \sim \mathcal{P}(\alpha Ax), \quad (11)$$

159 where α (in photons) represents the intensity of the image $x \in [0, 1]^N$. The higher the
 160 intensity, the higher the SNR of the measurements. We chose the acquisition matrix $A \in$
 161 $\mathbb{R}_+^{2M \times N}$, $M \leq N$, such that the positive and negative components of the target matrix
 162 $H \in \mathbb{R}^{M \times N}$ are measured independently

$$A = \begin{bmatrix} H_+ \\ H_- \end{bmatrix}, \quad (12)$$

163 where $H_\pm = \max(0, \mp H) \in \mathbb{R}_+^{M \times N}$. Choosing $B = [I_N, -I_N]$, we recover $BA = H$. In
 164 the case of accelerated acquisitions for which $M < N$, we choose H as a subsampled (i.e.,
 165 row-decimated) Hadamard matrix

$$H = [I_M, 0]QH_\uparrow, \quad (13)$$

166 where $H_\uparrow \in \mathbb{R}^{N \times N}$ represents the full Hadamard matrix, $I_M \in \mathbb{R}^{M \times M}$ the identity matrix,
 167 and $Q \in \{0, 1\}^{N \times N}$ a permutation matrix that ranks the Hadamard coefficients by signif-
 168 icance. The entire process is illustrated in Fig. 2d and Fig. 2e. The acquisition matrix
 169 can be defined using the `meas.HadamSplit` class, while Poisson noise is handled via the
 170 `noise.Poisson` class.

171 Reconstruction algorithms

172 We consider algorithms that fall into different reconstruction strategies (e.g., optimization-
173 based, post-processing and iterative methods, supervised or PnP methods). Given the modu-
174 larity of SPyRiT, all methods can be easily defined by exploiting the acquisition and denoising
175 operators. The selected methods are:

- 176 • Pseudo-inverse (Pinv), which corresponds to Eq. (8) with \mathcal{G}_θ being the identity.
- 177 • Pseudo-inverse with supervised denoising (Pinv-Net),²¹ which corresponds to Eq. (8)
178 where \mathcal{G}_θ is trained using Eq. (5).
- 179 • Pseudo-inverse with PnP denoising (Pinv-PnP), which corresponds to Eq. (8) where
180 \mathcal{G}_θ is trained independently as a denoiser.
- 181 • Supervised denoised completion (DC-Net),⁵⁰ which corresponds to Eq. (9) where \mathcal{G}_θ
182 is trained using Eq. (5).
- 183 • Learned proximal gradient descent (LPGD), which corresponds to Eq. (10) where \mathcal{G}_θ
184 is trained using Eq. (5).
- 185 • Proximal gradient descent with PnP denoising (DPGD-PnP), which corresponds to
186 Eq. (10), where \mathcal{G}_θ is trained independently as a denoiser.

187 The Pinv, Pinv-Net and Pinv-PnP methods are all instances of the same class `core.recon.PinvNet`,
188 where the only difference is the denoising layer (the identity for Pinv, UNet¹⁸ trained in a
189 supervised manner for Pinv-Net, and pretrained DR-UNet³⁴ for Pinv-PnP). The LPGD
190 and DPGD-PnP are also two instances of the same class `recon.LearnedPGD`. All algorithm
191 details are provided in Section . Note that both PnP methods (i.e., Pinv-PnP, DPGD-
192 PnP) rely on a hyperparameter (ν for Pinv-PnP and μ for DPGD-PnP) that allows the
193 denoising/regularization level to be manually tuned (see Appendix and Sec. 1.1 and 1.2 of
194 Supplement 1).

195 Example code

196 The code required to run both the simulation and reconstruction with the `pinvNet` network
197 is shown below.

```
198 # Data simulation
199 import math
200 import torch
201 from spyrit.core.meas import HadamSplit
202 from spyrit.core.noise import Poisson
203
204
205 alpha = 100.0
206 h = math.sqrt(N)
207 meas_op = HadamSplit(M, h, Ord_rec)
208 noise_op = Poisson(meas_op, alpha)
209 torch.manual_seed(0)
210 y = noise_op(x) # x is vectorized image
211
212 # Image reconstruction
213 from spyrit.core.recon import PinvNet
214 from spyrit.core.nnet import Unet
215
216 prep_op = SplitPoisson(alpha, meas_op)
217 denoi_net = Unet()
218 full_op = PinvNet(noise_op, prep_op, denoi_net)
219 x_rec = full_op.reconstruct(y)
220
221 # Simulation and reconstruction
222 torch.manual_seed(0)
223 x_rec_2 = full_op(x)
224
225 (x_rec == x_rec_2).all() # True
226
```

227 The code that generates the masks, acquisition matrices and images in Fig. 2 is provided

228 in the last section of Supplement 1. Multiple examples are provided in the tutorials accessible
229 in.⁵²

230 **Training**

231 All supervised methods (i.e., Pinv-Net, DC-Net, LPGD) are trained by minimizing the
232 mean squared error loss of Eq. (5) over the test dataset (100k images) of the ImageNet
233 ILSVRC2012 database.⁵³ Color images are subject to grayscale transformation, with random
234 crops of size 128×128 pixels and normalization to $[-1, 1]$ applied to all images. Measurements
235 are simulated according to Eq. (11) for an image intensity $\alpha = 10$ photons. The minimization
236 problem is solved using the Adam optimizer⁵⁴ with a learning rate of 10^{-3} , a batch size of 128
237 for Pinv-Net and LPGD and 256 for DC-Net, and 30 epochs. The learning rate is decreased
238 by a factor of 0.5 every 10 epochs. The weight decay regularization parameter is set to
239 10^{-7} . The parameters of the optimizer were optimized previously for DC-Net and Pinv-
240 Net.²³ For the PnP methods (i.e., Pinv-PnP and DPGD-PnP), the data driven layers are
241 trained independently as denoisers, see³⁴ for Pinv-PnP and⁵⁵ for DPGD-PnP. The methods
242 adapt to different noise levels by adjustment of a hyperparameter that controls the amount
243 of regularization/denoising (see Sec. 1.1 and 1.2 of Supplement 1).

244 **Results**

245 **Simulated data**

246 We simulate Hadamard split measurements corrupted by Poisson noise according to Eq. (11)
247 and Eq. (12) for three different image intensities α equal to 2, 10 and 50 photons, which
248 correspond to increasing SNR. Methods are evaluated using the entire evaluation set (50k
249 images) of the ImageNet ILSVRC2012 database.⁵³ All images are cropped to $N = 128 \times 128$
250 pixels. The number of measurements is set to $M = 4096$ and a $\times 4$ square subsampling
251 strategy is used (i.e., the measurements correspond to a full acquisition for 64×64 Hadamard

252 patterns). We compare all methods in terms of root mean square error (RMSE), structural
253 similarity index measure (SSIM), and by visual inspection. The hyperparameter of the PnP
254 methods is chosen for each noise level to minimize RMSE and maximize SSIM across a subset
255 of 384 images of the training set of the ImageNet ILSVRC2012 database. When RMSE and
256 SSIM disagree, the hyperparameter was selected by visual inspection, as described in Sections
257 1.1 and 1.2 of Supplement 1.

258 Table 1 reports RMSE and SSIM for the three noise levels for all methods. For the noise
259 level used for training ($\alpha = 10$ photons), all supervised methods provide similar results in
260 terms of RMSE and SSIM, which are slightly better than for PnP methods. Among PnP
261 methods, Pinv-PnP yields better metrics than DPGD-PnP. Deep learning methods reduce
262 RMSE by up to half compared to the traditional method Pinv. Comparing the results for
263 higher SNR ($\alpha = 50$ photons), slightly improved metrics (lower RMSE and higher SSIM) are
264 achieved by deep learning methods with respect to $\alpha = 10$ photons. For lower SNR ($\alpha = 2$
265 photons), supervised methods (except for DC-Net) approximately double RMSE compared
266 to PnP methods.

267 Figure 3 presents a number of ground truth and reconstructed images. For $\alpha = 10$
268 photons, all deep learning methods are able to remove noise quite efficiently. Supervised
269 methods lead to sharper images than PnP methods. Pinv-Net and LPGD are more consistent
270 across images than DC-Net, which yields sharper images in some cases and more blurred
271 results in others. LPGD leads to sharper results than Pinv-Net for some images. Pinv-PnP
272 provides a natural texture with low noise but images are less sharp. DC-Net provides the
273 best results in terms of sharpness. For other image intensities, the best results are obtained
274 using PnP methods and DC-Net, with more detail or less noise. For $\alpha = 2$ photons (i.e.,
275 lower SNR than training), supervised methods result in noisy images equivalent to Pinv,
276 except for DC-Net which achieves similar results to PnP methods. For $\alpha = 50$ photons (i.e.,
277 higher SNR than training) DC-Net performs well. More results can be found in Sec. 1.3 of
278 Supplement 1, including measurements simulated from the human brain image dataset.⁵⁶

279 **Experimental data**

280 Experimental data is used from the SPIHIM dataset,⁴⁵ an open-access collection of single-
281 pixel acquisitions that fulfil the FAIR principles.⁵⁷ In particular, we consider the Star Sector
282 resolution target (Thorlabs, R1L1S2P, see top row of Fig. 4) and a tomato slice (see bot-
283 tom row of Fig. 4), which has a smother texture and no symmetry. As in Section , the
284 measurements correspond to a full acquisition with $M = 4096$ split Hadamard patterns of
285 size 64×64 , which correspond to a $\times 4$ square subsampling strategy for a reconstruction
286 with $N = 128 \times 128$ pixels. Figure 4 shows the reconstruction of the Star Sector resolution
287 target and tomato slice. As for the simulated data, Pinv results in noisy images while the
288 data-driven methods efficiently remove noise. For the resolution target, Pinv-Net, LPGD
289 and DPGD-PnP effectively remove noise but lead to artefacts that are especially visible
290 close to the central white disk. DC-Net leads to a more natural texture but slightly blurred.
291 Pinv-PnP yields almost perfect reconstruction. For the tomato slice, supervised methods
292 Pinv-Net and LPGD result in the sharpest images, while the other methods yield images
293 that are excessively blurred. These findings are corroborated by the reconstruction results
294 of other images, which are reported in Section 2 of Supplement 1.

295 **Conclusion**

296 To the best of our knowledge, this is the first time a PyTorch package has been presented to
297 enable reproducible research and benchmarking in the field of single-pixel imaging. The pack-
298 age includes a number of supervised and plug-and-play methods, including post-processing
299 and iterative strategies.

300 We observe that all data-driven methods perform well provided that the image intensity
301 is similar or higher to the training phase. However, most of the data-driven methods are not
302 robust if the intensity is lower than the training phase, which is often the case in practice.
303 This suggests that comparative studies should carefully address this kind of discrepancy.

Table 1: Reconstruction metrics obtained for measurements simulated using the evaluation set of the ImageNet ILSVRC2012 database for three image intensities α . All data-driven reconstruction methods were trained considering $\alpha = 10$ photons. Values correspond to mean (standard deviations) across images. For DPGD-PnP, standard deviations are not shown because metrics were obtained from only 384 images, given its higher computational cost. Pinv leads to highly corrupted images for some cases, resulting in NaN values for SSIM.

Methods	RMSE	SSIM
$\alpha = 2$		
Pinv	1.09 (0.64)	NaN
Pinv-Net	0.81 (0.51)	0.19 (0.14)
LPGD	0.84 (0.51)	0.18 (0.12)
DC-Net	0.41 (0.19)	0.41 (0.19)
Pinv-PnP ($\nu = 115$)	0.34 (0.16)	0.42 (0.19)
DPGD-PnP ($\mu = 6000$)	0.45 (-)	0.26 (-)
$\alpha = 10$ (as for training)		
Pinv	0.53 (0.29)	NaN
Pinv-Net	0.25 (0.14)	0.61 (0.16)
LPGD	0.26 (0.14)	0.61 (0.16)
DC-Net	0.26 (0.14)	0.61 (0.17)
Pinv-PnP ($\nu = 50$)	0.27 (0.14)	0.56 (0.18)
DPGD-PnP ($\mu = 3500$)	0.30 (-)	0.46 (-)
$\alpha = 50$		
Pinv	0.31 (0.15)	NaN
Pinv-Net	0.22 (0.14)	0.68 (0.17)
LPGD	0.23 (0.14)	0.66 (0.18)
DC-Net	0.25 (0.15)	0.71 (0.14)
Pinv-PnP ($\nu = 20$)	0.23 (0.13)	0.68 (0.15)
DPGD-PnP ($\mu = 1500$)	0.22 (-)	0.67 (-)

304 In this regard, the versatility of PnP methods is an advantage via manual tuning of the
 305 hyperparameter with respect to the noise level. The hyperparameter, however, has a signif-
 306 icant impact on reconstruction quality and therefore requires careful selection. Among the
 307 supervised methods, DC-Net appears to be robust to deviations in the noise level, almost
 308 comparable to PnP methods and without hyperparameter selection. This is due to the fact
 309 that DC-Net includes a denoising step that removes noise prior to reconstruction. We also
 310 observe that the data-driven methods trained on simulations perform very well on experi-
 311 mental data, provided that the noise level corresponds to that of the training phase, which
 312 indicates that there is not a large model mismatch between simulations and experiments.

313 The modularity and versatility of SPyRiT make it suitable for further studies beyond
 314 this work, such as the comparison or learning of acquisition matrices, noise models, or re-
 315 construction algorithms. Current developments include hyperspectral imaging and dynamic
 316 imaging.⁵⁸ While SPyRiT has been designed for single-pixel imaging, it may also benefit
 317 other modalities, in particular in the field of computational optics where the formation mod-
 318 els are often in the form of Eq. (2). Computational ghost imaging⁵⁹ is of particular interest
 319 due to its equivalence, from an optical perspective, to single-pixel imaging.

320 **Appendix**

321 **Algorithm details**

322 **Reconstruction problem**

323 Regularized methods consist in defining the estimate as a solution to the minimization re-
 324 construction problem

$$\min_x \mathcal{D}(m, Hx) + g_\mu(x), \quad (14)$$

325 where \mathcal{D} is the data fidelity term and g_μ is a regularization term that depends on the
 326 hyperparameter (e.g., regularization parameter) $\mu > 0$. We choose the weighted least-squares
 327 data fidelity term

$$\mathcal{D}(m, Hx) = \frac{1}{2} \|m - Hx\|_{\Gamma^{-1}}^2, \quad (15)$$

328 where $\Gamma \in \mathbb{R}^{M \times M}$ represents the noise covariance, which is assumed to be diagonal for
 329 independent measurements, i.e., $\Gamma = \text{diag}(\sigma_1^2, \dots, \sigma_M^2)$ with $\sigma_i^2 > 0$ the variance of the i -
 330 th measurement. For the Poisson-corrupted split measurements described in Section , the
 331 variance can be approximated as

$$\sigma_i^2 = \frac{1}{4}(m_i^+ + m_i^-), \quad (16)$$

332 where m_i^+ and m_i^- represent, respectively, the raw measurements obtained from the positive
 333 and negative component of the i -th target pattern.²³

334 **Closed-forms: pseudo-inverse and Tikhonov regularization**

335 Without regularization (i.e., when $g_\mu(x) = 0$) and in the absence of noise model (i.e., choosing
 336 $\Gamma = I$), the solution to the problem in Eq. (14) is given by the pseudo-inverse approach
 337 defined by Eq. (8).

338 When $g_\mu(x) = \|x\|_{\Sigma^{-1}}^2$, where Σ represents the image covariance, the solution to the
 339 problem in Eq. (14) admits the closed form solution given by Eq. (9). The DC-Net method
 340 is of the form of Eq. (6) with \mathcal{R} given in Eq. (9). In this approach, the image covariance Σ ,
 341 which has been computed on the ImageNet ILSVRC2012 database (see⁶⁰), and the learnable
 342 parameters of \mathcal{G}_θ have been downloaded from.⁶¹

343 **Proximal gradient descent**

344 A common iterative method used to solve the minimization problem in Eq. (14) with data-
 345 fidelity term given in Eq. (15) is the proximal gradient descent algorithm (a.k.a. forward-

backward splitting algorithm).⁶² For this problem, iterations are given by

$$x_{k+1} = \text{prox}_{\gamma g_\mu}(x_k - \gamma H^\top \Gamma^{-1}(Hx_k - m)), \quad (17)$$

where $\text{prox}_{\gamma g}$ is the proximity operator of g (see⁶²) and γ is a step size. The sequence $(x_k)_{k \in \mathbb{N}}$ generated by Eq. (17) converges with rate $O(1/k)$ using a fixed step size $\gamma \in (0, \beta^{-1})$, where

$$\beta = \|\Gamma^{-1/2} H H^\top \Gamma^{-1/2}\|_2 = \|H^\top \Gamma^{-1} H\|_2, \quad (18)$$

with $\|\cdot\|_2$ the spectral norm (i.e., the largest singular value of its argument).

Step size for a Hadamard matrix

To compute the step-size γ in Eq. (17), it is necessary to compute β . In the case when H is the Hadamard matrix, we have $H^\top H = N I_N$. Hence, using Eq. (18), we have $\beta = N \|\Gamma^{-1}\|_S = N \max_i \sigma_i^{-1}$, so

$$\beta^{-1} = \frac{\min_i \sigma_i^2}{N}. \quad (19)$$

Learned proximal gradient descent

The LPGD method considers the proximal iteration of Eq. (17), where the proximity operator of g_μ is replaced by a neural network \mathcal{G}_{θ_k} ,⁵¹ i.e.,

$$x_{k+1} = \mathcal{G}_{\theta_k}(x_k - \gamma H^\top (Hx_k - m)). \quad (20)$$

To limit the number of parameters, the weights of the neural network are shared across iterations, i.e., for every k , $\mathcal{G}_{\theta_k} \equiv \mathcal{G}_\theta$. Since the neural networks of LPGD are learned end-to-end, we only unroll Eq. (20) over a fixed number of iterations K .

In this context, two hyperparameters need to be tuned: the number of iterations K and the step size γ . The number of iterations is commonly chosen between 3 and 10, as a trade-

362 off between performance and computational cost. We have tested both fixed and decreasing
 363 step sizes, i.e., $\gamma_k = \tau\gamma_{k-1}$ with a decaying multiplicative factor $0 < \tau \leq 1$. The step size
 364 has been initialized to the inverse of the Lipschitz constant given in Eq. (19), as suggested
 365 by optimization theory (see Appendix). We have tested different values for the number of
 366 iterations (3, 6 and 10) and different decaying factors (0.1, 0.5, 0.9, and 1). The best results
 367 were obtained for $\tau = 0.9$, and as little difference was found in terms of number of iterations,
 368 we selected $K = 3$ (results not shown).

369 Dual PGD network

370 We consider the LPGD iteration given in Eq. (20), where \mathcal{G}_{θ_k} is trained as a denoiser
 371 independent from the measurement model and is itself an unfolded algorithm that mimics
 372 the proximity operator of g_μ . Specifically, we consider the case when $g_\mu = \iota_{[-1,1]} + \lambda\|L \cdot\|_1$
 373 with $L: \mathbb{R}^N \rightarrow \mathbb{R}^S$ a linear sparsifying operator. In this case the proximity operator of g_μ
 374 can be computed using the dual PGD algorithm.⁶² Within a deep learning framework, only
 375 a fixed number of iterations of the dual PGD algorithm are unrolled, and the operator L can
 376 be learned (and changed for each layer), leading to a Dual PGD network dubbed DPGD-PnP
 377 (see^{55,63}).

378 The DPGD-PnP network is composed of 20 layers, and the linearity associated with
 379 each layer corresponds to a convolution with 64 features. We trained the network using the
 380 ImageNet test dataset.

381 Since this approach is based on the LPGD iteration from Eq. (20), the step size is chosen
 382 as per Appendix . The main advantage compared to the supervised approaches is that the
 383 training is independent of the measurement model. However, a large number of iterations
 384 must be considered. We choose $K_{\max} = 101$ with a stopping criterion $\|x_{k+1} - x_k\| < 10^{-4}\|x_k\|$
 385 in case convergence is reached before K_{\max} iterations.

Acknowledgement

Institut Universitaire de France; Agence Nationale de la Recherche (ANR-11-LABX-0063, ANR-22-CE19-0030-01).

Supporting Information Available

Additional simulation and experimental details, methods comparisons, choice of hyper-parameters, evaluation metrics. For reproducibility, the code and data are provided in https://github.com/openspyrit/spyrit-examples/tree/master/2024_spyrit.

References

- (1) Edgar, M. P.; Gibson, G. M.; Padgett, M. J. Principles and Prospects for Single-Pixel Imaging. *Nature Photonics* **2019**, *13*, 13–20.
- (2) Gibson, G. M.; Gibson, G. M.; Johnson, S. D.; Johnson, S. D.; Padgett, M. J.; Padgett, M. J. Single-Pixel Imaging 12 Years on: A Review. *Optics Express* **2020**, *28*, 28190–28208.
- (3) Harwit, M.; Sloane, N. J. A. *Hadamard Transform Optics*; Academic Press, 1979.
- (4) Duarte, M.; Davenport, M.; Takhar, D.; Laska, J.; Sun, T.; Kelly, K.; Baraniuk, R. Single-Pixel Imaging via Compressive Sampling. *Signal Processing Magazine, IEEE* **2008**, *25*, 83–91.
- (5) Radwell, N.; Mitchell, K. J.; Gibson, G. M.; Edgar, M. P.; Bowman, R.; Padgett, M. J. Single-Pixel Infrared and Visible Microscope. *Optica* **2014**, *1*, 285–289.
- (6) Wang, Y.; Huang, K.; Fang, J.; Yan, M.; Wu, E.; Zeng, H. Mid-Infrared Single-Pixel Imaging at the Single-Photon Level. *Nature Communications* **2023**, *14*, 1073.

- 407 (7) Ye, J.-T.; Yu, C.; Li, W.; Li, Z.-P.; Lu, H.; Zhang, R.; Zhang, J.; Xu, F.; Pan, J.-W.
408 Ultraviolet Photon-Counting Single-Pixel Imaging. *Applied Physics Letters* **2023**, *123*,
409 024005.
- 410 (8) Chan, W. L.; Charan, K.; Takhar, D.; Kelly, K. F.; Baraniuk, R. G.; Mittleman, D. M. A
411 Single-Pixel Terahertz Imaging System Based on Compressed Sensing. *Applied Physics*
412 *Letters* **2008**, *93*, 121105.
- 413 (9) Klein, Y.; Schori, A.; Dolbnya, I. P.; Sawhney, K.; Schwartz, S. X-Ray Computational
414 Ghost Imaging with Single-Pixel Detector. *Optics Express* **2019**, *27*, 3284–3293.
- 415 (10) Jauregui-Sánchez, Y.; Clemente, P.; Lancis, J.; Tajahuerce, E. Single-Pixel Imaging
416 with Fourier Filtering: Application to Vision through Scattering Media. *Optics Letters*
417 **2019**, *44*, 679–682.
- 418 (11) Rousset, F.; Ducros, N.; Peyrin, F.; Valentini, G.; D’Andrea, C.; Farina, A. Time-
419 Resolved Multispectral Imaging Based on an Adaptive Single-Pixel Camera. *Opt. Ex-*
420 *press* **2018**, *26*, 10550–10558.
- 421 (12) Sun, M.-J.; Zhang, J.-M. Single-Pixel Imaging and Its Application in Three-
422 Dimensional Reconstruction: A Brief Review. *Sensors* **2019**, *19*, 732.
- 423 (13) Radwell, N.; Johnson, S. D.; Edgar, M. P.; Higham, C. F.; Murray-Smith, R.; Pad-
424 gett, M. J. Deep Learning Optimized Single-Pixel LiDAR. *Applied Physics Letters*
425 **2019**, *115*, 231101.
- 426 (14) Yao, M.; Cheng, J.; Huang, Z.; Zhang, Z.; Li, S.; Peng, J.; Zhong, J. Reflection Light-
427 Field Microscope with a Digitally Tunable Aperture by Single-Pixel Imaging. *Optics*
428 *Express* **2019**, *27*, 33040–33050.
- 429 (15) Arridge, S.; Maass, P.; Öktem, O.; Schönlieb, C.-B. Solving Inverse Problems Using
430 Data-Driven Models. *Acta Numerica* **2019**, *28*, 1–174.

- 431 (16) Barbastathis, G.; Ozcan, A.; Situ, G. On the Use of Deep Learning for Computational
432 Imaging. *Optica* **2019**, *6*, 921–943.
- 433 (17) Higham, C. F.; Murray-Smith, R.; Padgett, M. J.; Edgar, M. P. Deep Learning for
434 Real-Time Single-Pixel Video. *Scientific Reports* **2018**, *8*, 2369.
- 435 (18) Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomed-
436 ical Image Segmentation. International Conference on Medical Image Computing and
437 Computer-Assisted Intervention. 2015; pp 234–241.
- 438 (19) Yang, S.; Qin, H.; Yan, X.; Yuan, S.; Yang, T. Deep Spatial-Spectral Prior with an
439 Adaptive Dual Attention Network for Single-Pixel Hyperspectral Reconstruction. *Op-
440 tics Express* **2022**, *30*, 29621–29638.
- 441 (20) Shang, R.; Hoffer-Hawlik, K.; Wang, F.; Situ, G.; Luke, G. P. Two-Step Training Deep
442 Learning Framework for Computational Imaging without Physics Priors. *Optics Express*
443 **2021**, *29*, 15239–15254.
- 444 (21) Rizvi, S.; Cao, J.; Zhang, K.; Hao, Q. Improving Imaging Quality of Real-time Fourier
445 Single-pixel Imaging via Deep Learning. *Sensors* **2019**, *19*, 4190.
- 446 (22) Ducros, N.; Lorente Mur, A.; Peyrin, F. A Completion Network for Reconstruction from
447 Compressed Acquisition. IEEE 17th International Symposium on Biomedical Imaging
448 (ISBI). Iowa City, United States, 2020; pp 619–623.
- 449 (23) Lorente Mur, A.; Leclerc, P.; Peyrin, F.; Ducros, N. Single-Pixel Image Reconstruction
450 from Experimental Data Using Neural Networks. *Optics Express* **2021**, *29*, 17097–
451 17110.
- 452 (24) Monga, V.; Li, Y.; Eldar, Y. C. Algorithm Unrolling: Interpretable, Efficient Deep
453 Learning for Signal and Image Processing. *IEEE Signal Processing Magazine* **2021**, *38*,
454 18–44.

- 455 (25) Zhang, J.; Chen, B.; Xiong, R.; Zhang, Y. Physics-Inspired Compressive Sensing: Beyond Deep Unrolling. *IEEE Signal Processing Magazine* **2023**, *40*, 58–72.
456
- 457 (26) Lorente Mur, A.; Peyrin, F.; Ducros, N. Deep Expectation-Maximization for Single-
458 Pixel Image Reconstruction With Signal-Dependent Noise. *IEEE Transactions on Computational Imaging* **2022**, *8*, 759–769.
459
- 460 (27) Sauder, J.; Genzel, M.; Jung, P. Learning Structured Sparse Matrices for Signal Recovery via Unrolled Optimization. NeurIPS 2021 Workshop on Deep Learning and Inverse
461 Problems. 2021.
462
- 463 (28) Li, J.; Li, Y.; Li, J.; Zhang, Q.; Li, J. Single-Pixel Compressive Optical Image Hiding Based on Conditional Generative Adversarial Network. *Optics Express* **2020**, *28*,
464 22992–23002.
465
- 466 (29) Jiang, P.; Liu, J.; Wu, L.; Xu, L.; Hu, J.; Zhang, J.; Zhang, Y.; Yang, X. Fourier Single
467 Pixel Imaging Reconstruction Method Based on the U-net and Attention Mechanism
468 at a Low Sampling Rate. *Optics Express* **2022**, *30*, 18638–18654.
- 469 (30) Wang, F.; Wang, C.; Deng, C.; Han, S.; Situ, G. Single-Pixel Imaging Using Physics
470 Enhanced Deep Learning. *Photonics Research* **2022**, *10*, 104.
- 471 (31) Qu, G.; Wang, P.; Yuan, X. Dual-Scale Transformer for Large-Scale Single-Pixel Imaging. IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR). 2024.
472
- 473 (32) Tian, Y.; Fu, Y.; Zhang, J. Plug-and-Play Algorithms for Single-Pixel Imaging. *Optics and Lasers in Engineering* **2022**, *154*, 106970.
474
- 475 (33) Mäkinen, Y.; Azzari, L.; Foi, A. Collaborative Filtering of Correlated Noise: Exact
476 Transform-Domain Variance for Improved Shrinkage and Patch Matching. *IEEE Transactions on Image Processing* **2020**, *29*, 8339–8354.
477

- 478 (34) Zhang, K.; Li, Y.; Zuo, W.; Zhang, L.; Van Gool, L.; Timofte, R. Plug-and-Play Image
479 Restoration With Deep Denoiser Prior. *IEEE Transactions on Pattern Analysis and*
480 *Machine Intelligence* **2022**, *44*, 6360–6376.
- 481 (35) Dave, A.; Vadathya, A. K.; Subramanyam, R.; Baburajan, R.; Mitra, K. Solving Inverse
482 Computational Imaging Problems Using Deep Pixel-Level Prior. *IEEE Transactions on*
483 *Computational Imaging* **2019**, *5*, 37–51.
- 484 (36) Wang, F.; Wang, C.; Chen, M.; Gong, W.; Zhang, Y.; Han, S.; Situ, G. Far-Field Super-
485 Resolution Ghost Imaging with a Deep Neural Network Constraint. *Light: Science &*
486 *Applications* **2022**, *11*, 1.
- 487 (37) Li, J.; Wu, B.; Liu, T.; Zhang, Q. URNet: High-quality Single-Pixel Imaging with Un-
488 trained Reconstruction Network. *Optics and Lasers in Engineering* **2023**, *166*, 107580.
- 489 (38) Shenouda, J.; Bajwa, W. U. A Guide to Computational Reproducibility in Signal Pro-
490 cessing and Machine Learning [Tips & Tricks]. *IEEE Signal Processing Magazine* **2023**,
491 *40*, 141–151.
- 492 (39) Simeoni, M.; Kashani, S.; Rué-Queralt, J.; Developers, P. Pyxu-Org/Pyxu: Pyxu.
493 Zenodo.
- 494 (40) Tachella, J. Deepinv/Deepinv: PyTorch Library for Solving Imaging Inverse Problems
495 Using Deep Learning.
- 496 (41) Rit, S.; Oliva, M. V.; Brousmiche, S.; Labarbe, R.; Sarrut, D.; Sharp, G. C. The
497 Reconstruction Toolkit (RTK), an Open-Source Cone-Beam CT Reconstruction Toolkit
498 Based on the Insight Toolkit (ITK). *Journal of Physics: Conference Series* **2014**, *489*,
499 012079.
- 500 (42) van Aarle, W.; Palenstijn, W. J.; Cant, J.; Janssens, E.; Bleichrodt, F.; Dabrovolski, A.;

- 501 Beenhouwer, J. D.; Batenburg, K. J.; Sijbers, J. Fast and Flexible X-ray Tomography
502 Using the ASTRA Toolbox. *Optics Express* **2016**, *24*, 25129–25147.
- 503 (43) Biguri, A.; Dosanjh, M.; Hancock, S.; Soleimani, M. TIGRE: A MATLAB-GPU Tool-
504 box for CBCT Image Reconstruction. *Biomedical Physics & Engineering Express* **2016**,
505 *2*, 055010.
- 506 (44) BART Toolbox for Computational Magnetic Resonance Imaging.
- 507 (45) Beneti Martins, G.; Mahieu-Williame, L.; Baudier, T.; Ducros, N. OpenSpyrit: An
508 Ecosystem for Open Single-Pixel Hyperspectral Imaging. *Optics Express* **2023**, *31*,
509 15599.
- 510 (46) Zhang, Z.; Ma, X.; Zhong, J. Single-Pixel Imaging by Means of Fourier Spectrum
511 Acquisition. *Nature communications* **2015**, *6*.
- 512 (47) Zhang, Z.; Wang, X.; Zheng, G.; Zhong, J. Hadamard Single-Pixel Imaging versus
513 Fourier Single-Pixel Imaging. *Opt. Express* **2017**, *25*, 19619–19639.
- 514 (48) Lorente Mur, A.; Ochoa, M.; Cohen, J. E.; Intes, X.; Ducros, N. Handling Negative
515 Patterns for Fast Single-Pixel Lifetime Imaging. Proc. SPIE 10862. 2019.
- 516 (49) Jin, K. H.; McCann, M. T.; Froustey, E.; Unser, M. Deep Convolutional Neural Network
517 for Inverse Problems in Imaging. *IEEE Transactions on Image Processing* **2017**, *26*,
518 4509–4522.
- 519 (50) Mur, A. L.; Montcel, B.; Peyrin, F.; Ducros, N. Deep Neural Networks for Single-
520 Pixel Compressive Video Reconstruction. *Unconventional Optical Imaging II*. 2020; p
521 113510S.
- 522 (51) Mardani, M.; Sun, Q.; Vasawanala, S.; Pappyan, V.; Monajemi, H.; Pauly, J.; Donoho, D.
523 Neural Proximal Gradient Descent for Compressive Imaging. Proceedings of the 32nd

- 524 International Conference on Neural Information Processing Systems. Red Hook, NY,
525 USA, 2018; pp 9596–9606.
- 526 (52) <https://spyrit.readthedocs.io/en/master/gallery/index.html>.
- 527 (53) Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpa-
528 thy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L. ImageNet Large Scale Visual
529 Recognition Challenge. *International Journal of Computer Vision* **2015**, *115*, 211–252.
- 530 (54) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. 3rd International
531 Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9,
532 2015, Conference Track Proceedings. 2015.
- 533 (55) Le, H. T. V.; Repetti, A.; Pustelnik, N. PNN: From Proximal Algorithms to Robust
534 Unfolded Image Denoising Networks and Plug-and-Play Methods. 2023.
- 535 (56) Fabelo, H. et al. Deep Learning-Based Framework for In Vivo Identification of Glioblas-
536 toma Tumor Using Hyperspectral Images of Human Brain. *Sensors* **2019**, *19*, 920.
- 537 (57) Wilkinson, M. D. et al. The FAIR Guiding Principles for Scientific Data Management
538 and Stewardship. *Scientific Data* **2016**, *3*, 160018.
- 539 (58) Maitre, T.; Bretin, E.; Mahieu-William, L.; Sdika, M.; Ducros, N.
- 540 (59) Shapiro, J. H. Computational Ghost Imaging. *Physical Review A* **2008**, *78*, 061802.
- 541 (60) Ref. 45, Sec. 3.3.
- 542 (61) https://github.com/openspyrit/spyrit-examples/tree/master/2022_OE_spyrit2.
- 543 (62) Combettes, P. L.; Pesquet, J.-C. In *Fixed-Point Algorithms for Inverse Problems in*
544 *Science and Engineering*; Bauschke, H. H., Burachik, R. S., Combettes, P. L., Elser, V.,
545 Luke, D. R., Wolkowicz, H., Eds.; Springer: New York, NY, 2011; pp 185–212.

546 (63) Repetti, A.; Terris, M.; Wiaux, Y.; Pesquet, J.-C. Dual Forward-Backward Unfolded
547 Network for Flexible Plug-and-Play. 2022 30th European Signal Processing Conference
548 (EUSIPCO). 2022; pp 957–961.

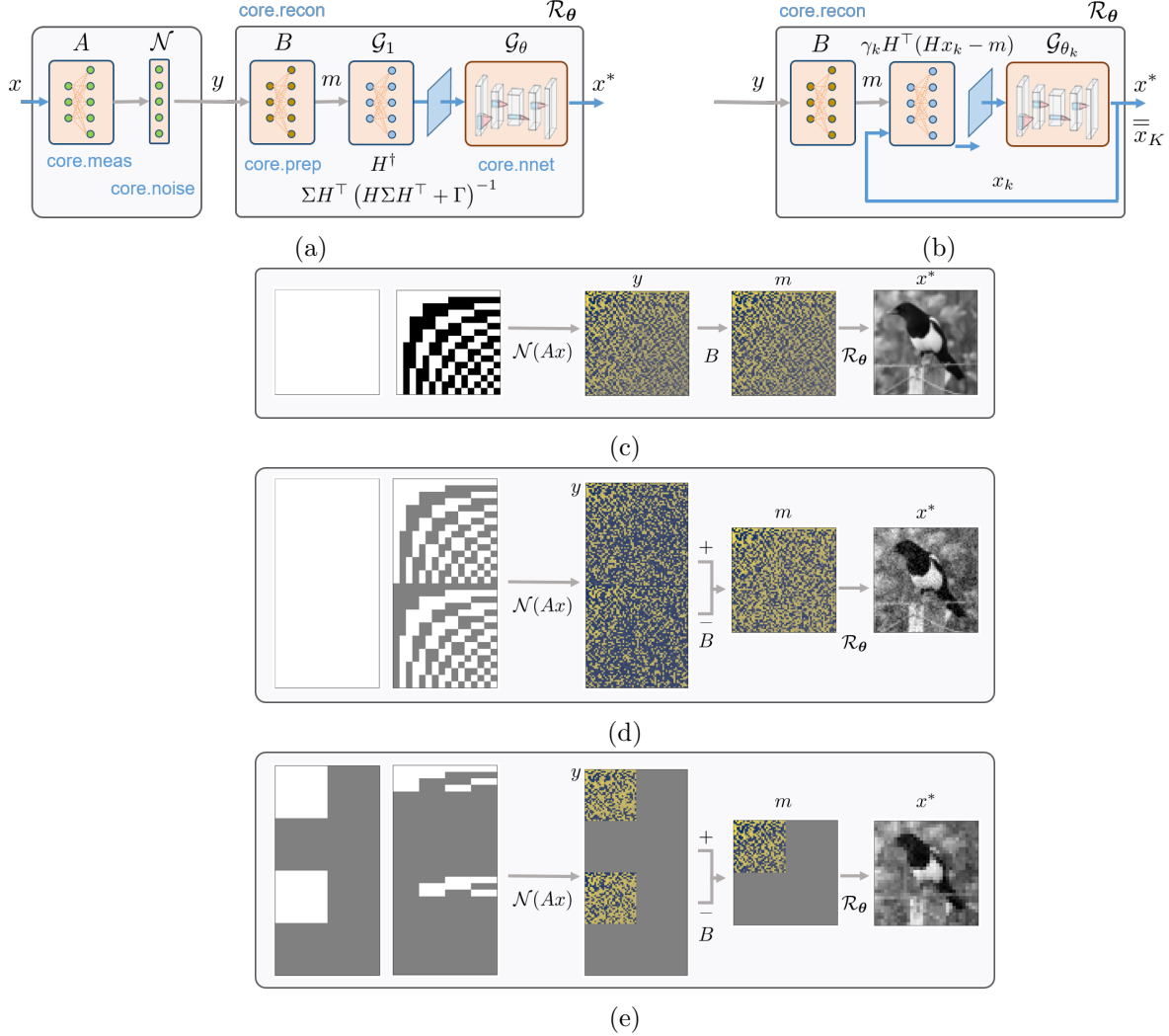


Figure 2: Typical SPyRiT pipelines involving input of an image $x \in \mathbb{R}^N$, simulation of modulated measurements Ax , application of a noise model \mathcal{N} which leads to the raw measurements $y \in \mathbb{R}^J$. From these, the reconstruction operator \mathcal{R}_θ outputs an estimate x^* of x . **(a)** Post-processing reconstruction approaches corresponding to Eq. (6). All steps are clearly separated and defined independently in the submodules `core.meas`, `core.noise`, and `core.recon`. The reconstruction algorithm can integrate preprocessing from `core.prep`, analytical steps and neural networks from `core.nnet`. **(b)** Iterative reconstruction approaches corresponding to Eq. (7). **(c)** Hadamard acquisitions which cannot be implemented in practice due to negative values in the target acquisition matrix. **(d)** Split Hadamard patterns defined by Eq. (12) with $M = N$ and the corresponding preprocessing $B = [I_N, -I_N]$. Poisson noise is simulated with $\alpha = 100$. **(e)** Square subsampling for split Hadamard acquisition, i.e., Eq. (12) with $M < N$ and the corresponding preprocessing $B = [I_M, -I_M]$. No noise is simulated. **(c-e)** The images on the left correspond to sampling masks. White masks correspond to the case $M = N$ where all measurements are kept, while grey pixels appear in the case of accelerated acquisitions where $M < N$. A sampling mask is a graphical representation of the permutation matrix Q in Eq. (13). Each white pixel of a sampling mask corresponds to the nonzero entry of each of the first M rows of Q . For simplicity, the $N = 64 \times 64$ acquisition matrices used are presented here as $N = 16 \times 16$.

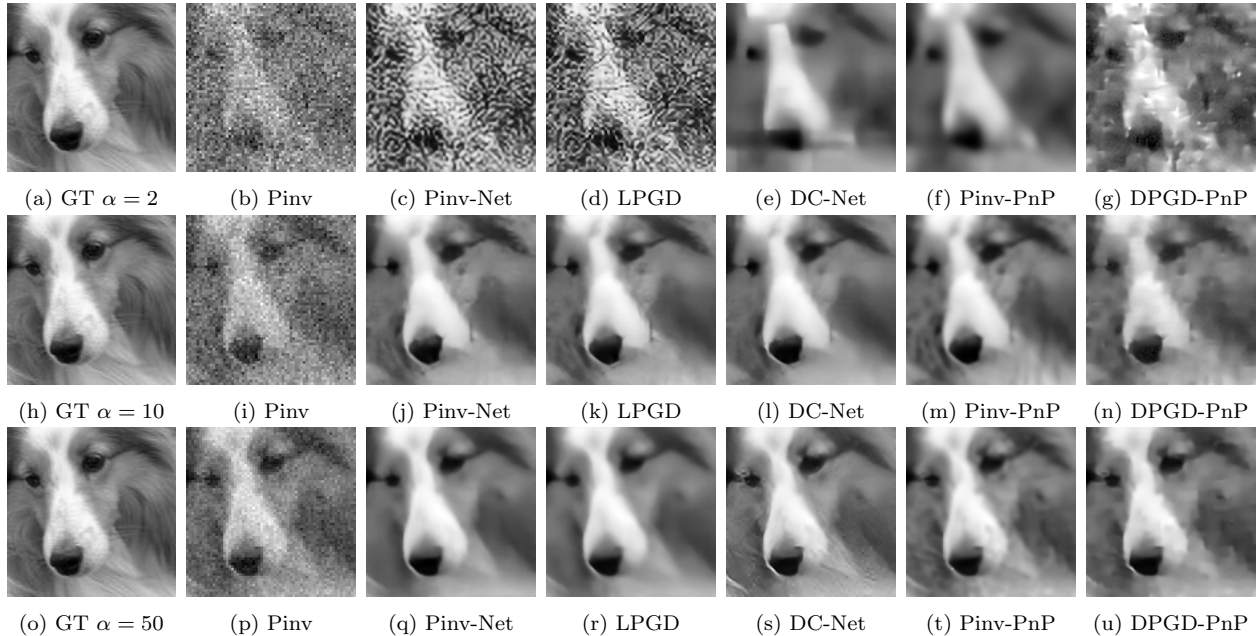


Figure 3: Ground truth (GT) and reconstructed images obtained from measurements simulated from the ImageNet ILSVRC2012 evaluation set. All supervised methods are trained with an image intensity $\alpha = 10$ photons and tested for $\alpha = 2$ (first row), $\alpha = 10$ (second row), and $\alpha = 50$ (third row) photons. The measurements correspond to Hadamard split acquisitions with a $\times 4$ square subsampling strategy. The images are reconstructed from $M = 4096$ measurements, with the number of pixels set to $N = 128 \times 128$. The Pinv-PnP hyperparameter is set to $\nu = 115$ for $\alpha = 2$; $\nu = 45$ for $\alpha = 10$; and $\nu = 20$ for $\alpha = 50$. The DPGD-PnP hyperparameter is set to $\mu = 6000$ for $\alpha = 2$; $\mu = 3500$ for $\alpha = 10$; and $\mu = 1500$ for $\alpha = 50$.

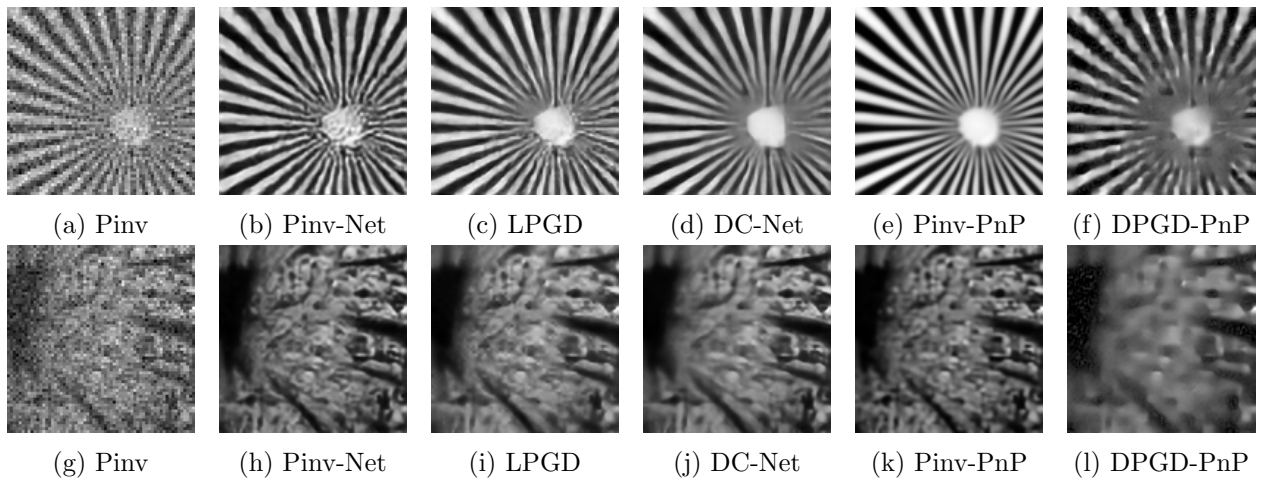


Figure 4: Experimental reconstruction results for the Star Sector resolution target and tomato slice. The measurements correspond to Hadamard split acquisitions using a $\times 4$ square subsampling strategy. The images are reconstructed from $M = 4096$ measurements, with the number of pixels set to $N = 128 \times 128$. The Pinv-PnP hyperparameter is set to $\nu = 55$ for the resolution target; and $\nu = 35$ for the tomato slice. The DPGD-PnP hyperparameter is set to $\mu = 4000$ for the resolution target; and $\mu = 4000$ for the tomato slice.