Plug-and-play learned proximal trajectory for 3D sparse-view X-ray computed tomography

Romain Vo, Julie Escoda, Caroline Vienne, Etienne Decencière

Plug-and-Play Learned Proximal Trajectory for 3D Sparse-View X-Ray Computed Tomography

Romain Vo[1,2], Julie Escoda[1], Caroline Vienne[1], and Étienne Decencière[2]

[1] Université Paris-Saclay, CEA, List, F-91190, Palaiseau, France
[2] Mines Paris, PSL University, Centre for Mathematical Morphology (CMM), 77300 Fontainebleau, France
firstname.lastname@cea.fr, etienne.decenciere@minesparis.psl.eu

Abstract. Plug-and-Play algorithms (PnP) have recently emerged as a powerful framework for solving inverse problems in imaging. They leverage the power of Gaussian denoising algorithms to solve complex optimization problems. This work focuses on the challenging task of 3D sparse-view X-ray computed tomography (CT). We propose to replace the Gaussian denoising network in Plug-and-Play with a restoration network, *i.e.* a network trained to remove arbitrary artifacts. We show that using a restoration prior tailored to the specific inverse problem improves the performances of Plug-and-Play algorithms. Besides, we show that plugging a basic restoration network into a PnP scheme is not sufficient to obtain good results. Thus, we propose a procedure to train the restoration network to be a robust approximation of a proximal operator along a pre-defined optimization trajectory. We demonstrate the effectiveness and scalability of our approach on two 3D Cone-Beam CT datasets and outperform state-of-the-art methods in terms of PSNR. Code is available at https://github.com/romainvo/pnp-learned-proximal-trajectory.

Keywords: Deep learning · Sparse-View Computed Tomography · Inverse Problem · Regularization · Plug-and-Play

1 Introduction

X-ray computed tomography (CT) is a non-destructive technique for examining and visualizing the internal structure of objects without causing damage. It is highly valued for its ability to generate detailed 3D images, making it widely used in medical diagnostics, industrial inspection, and materials science [26, 39, 49]. Current issues with CT imaging include the inversion of low-dose (LDCT) or sparse-view data (SVCT). The former arises from the necessity to minimize the radiation dose to which the patient is exposed, while the latter stems from the need to decrease the number of projections taken, thereby reducing the time required for data acquisition. Both of these issues result in a limited number of measurements, leading to a loss of information and the appearance of artifacts in the reconstructed image. The reconstruction task poses further problems in 3D, as the evaluation of the forward operator becomes computationally expensive, especially in the case of cone-beam geometry setups [46], such as in industrial CT [26]. Apart from the scale of the problem, which naturally increases in 3D, the Cone-Beam

geometry introduces a significant challenge; that is, the inverse problem is not separable in axial 2D smaller problems as is typically done for medical datasets [33, 34].

A straightforward approach to solve SVCT is to train a post-processing network [10, 25], or restoration network, to learn a one-pass mapping between a low-quality reconstruction and a ground-truth reconstruction. This simple approach maximizes PSNR and is computationally efficient, though it does not leverage knowledge of the forward model during the reconstruction. As emphasized in [41], post-processing networks do not minimize a data-fidelity term and may generate artifacts unrelated to the measurement data.

Common strategies to address ill-posed imaging inverse problems combine variational regularization [8, 20] with learned priors [1, 40, 58]. The idea is to combine the knowledge of the forward operator in the form of a data-fidelity term to minimize coupled with regularization through a penalty term that encodes *a priori* information on the unknown solution. Plug-and-Play approaches (PnP) [44, 55] in particular, allow the combination of these two components in a modular fashion. They solve ill-posed inverse problems using iterative proximal splitting algorithms [16], where a denoiser replaces the proximal operator of the regularization term. This denoiser includes any trainable neural network, which allows leveraging the power of deep learning to learn a prior from data while still being able to use the model-based iterative reconstruction framework [3, 47, 52, 57]. Recent works have also focused on using other operators such as deblurring or super-resolution operators, as opposed to denoising, as implicit priors in Plug-and-Play approaches and have reached comparable performances in several imaging inverse problems [29, 36]. Despite successful performances in various applications, most existing PnP approaches rely on contractive fixed-point iterations to get convergence guarantees [43, 52]. It poses several problems regarding stability as, in practice, contraction is never guaranteed for neural networks [42, 54]. Other works require strong assumptions, such as strong convexity of the fidelity term [47], which limit the method's applicability to a specific class of problems that do not include CT. Different works have also proposed to use architectural constraints such as gradient-step denoisers [14, 30] to obtain convergence guarantees. Besides the previously mentioned works, diffusion models [28, 51] have attracted attention in the context of PnP [60] and inverse problems solving [11, 12, 35], as they provide a principled way to learn a prior from data. Nonetheless, these methods are either sampling-intensive and require many iterations to converge or rely on the proximal operator of the data-fidelity term, which is computationally intensive for CT.

Another class of works, the so-called deep unrolled approaches, tries to circumvent these issues by learning the entire iterative procedure [1, 2, 4, 22, 53]. Based on proximal splitting algorithms [16], unrolled methods propose to learn a prior, tailored to the specific inverse problem at hand, by training the denoiser end-to-end with the reconstruction procedure. The difference with PnP is that the denoiser is not pre-trained but instead trained jointly with the iterative procedure to minimize a reconstruction loss. These methods result in improved performances within a reduced number of steps. However, a significant drawback comes with training time and memory complexity, which scales linearly with the number of iterations and the complexity of the forward operator. Despite recent works on invertible unrolling [6, 46], constant memory train-

ing with Deep Equilibrium Models [5, 24, 27, 37, 61], unrolled techniques remain prohibitively expensive for large-scale problems such as Cone-Beam 3D CT [18, 34] and there exist only a few applications for downscaled data [6, 46].

This work proposes a novel approach to integrate restoration priors into the PnP framework. Inspired by Zhang *et al*. [57], we observe that Gaussian denoising networks, used in PnP, are less powerful than post-processing networks [10, 25] in removing task-specific artifacts (Fig. 1). We aim to learn a prior tailored to a specific inverse problem without relying on unrolled training. We show that naively plugging a post-processing network in a PnP framework is not satisfactory, and we propose an approach to train this prior as a robust approximation of a proximal operator along a stored and pre-defined optimization trajectory. Using a pre-defined optimization trajectory allows us to include knowledge of the forward problem during optimization of the prior while maintaining an offline procedure, *i.e*. no call to the forward model. Once learned, the restoration prior is used as a drop-in replacement for the proximal operator in classic PnP algorithms [52, 55].
The contributions of this paper can be summarized as follows:

- We propose a novel approach, Learned Proximal Trajectory (LPT), to parametrize and learn the prior in a PnP framework, which is modular and tailored to the specific inverse problem. First, training of the restoration operator is done offline and does not rely on unrolling. Second, we use the reconstruction loss as a regularizer in a composite problem to define a target optimization trajectory. Sampling from this trajectory, the restoration network is trained to predict the ground-truth image from any intermediate reconstruction of the optimization trajectory. This contrasts with Gaussian denoising networks, which are only trained to denoise Gaussian corrupted ground-truth images.
- We augment the inputs of the restoration networks with iteration step conditioning. This conditioning allows the network to adapt the restoration strength to the current state of the optimization trajectory. We show that this conditioning technique consistently improves the results of our experiments.
- Based on the convergence analysis of Cohen *et al*. [15], we show that our method converges without constraining the Lipschitz constant of the restoration network. In this work, we propose a simple condition for the regularization step size to ensure convergence to a fixed point.
- Experimentally, we show that this approach outperforms state-of-the-art PnP methods and supervised approaches in PSNR on the challenging task of 3D sparse-view X-ray computed tomography. Unlike learning-only approaches such as deep post-processing [25], our reconstructed images maintain high PSNR values while being geometrically more accurate and do not contain artifacts.

2 Background and related work

2.1 CT reconstruction as an inverse problem

The CT reconstruction problem can be framed as a generic linear inverse problem:

$$b = Ax^* + \epsilon, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ is the forward operator and $\epsilon \in \mathbb{R}^m$ is the measurement noise. The goal is to recover the true image $x^* \in \mathbb{R}^n$ from a set of measurements $b \in \mathbb{R}^m$. This problem poses a significant challenge due to its inherently *ill-posed* nature. In low-dose CT (LDCT), the limited radiation dose results in an unstable solution with respect to measurement noise. In sparse-view CT (SVCT), the system is underdetermined due to the limited number of projections $m \ll n$. In both cases, the solution is not unique, and the reconstruction is sensitive to noise.

The usual approach to compute an estimation $\hat{x}$ of the true image involves solving a composite optimization problem of the form:

$$\hat{x} \in \underset{x\in\mathbb{R}^n}{\arg\min}\ \ell(x) + \lambda\mathcal{R}(x),\ \lambda > 0, \tag{2}$$

where $\ell(x) = \frac{1}{2}\|Ax-b\|_2^2$ is the data-fidelity objective and $\mathcal{R}$ is a suitable regularization term. Popular choices of regularization include the Total Variation (TV) minimization [23,50], which promotes piecewise smoothness of the solution. Generally, optimization algorithms to solve Eq. (2) are built on proximal splitting methods [16], which are particularly suited for handling composite objectives and non-differentiable regularizers such as the ones based on the $L1$–norm. The proximity operator of a function $\phi : \mathbb{R}^n \to \mathbb{R}$ is defined as

$$\text{prox}_{\gamma\phi}(x) = \underset{y\in\mathbb{R}^n}{\arg\min}\ \gamma\phi(y) + \frac{1}{2}\|x-y\|_2^2,\ \ \gamma > 0. \tag{3}$$

As an example, the forward-backward splitting method (FBS), also known as the proximal gradient descent algorithm (PGD) [16], alternates between a gradient descent step $\text{Id} - \tau\nabla\ell$, $\tau > 0$, and the evaluation of the proximity operator of $\mathcal{R}$ with regularization parameter $\lambda > 0$, *i.e.*

$$\begin{cases} x_{k+\frac{1}{2}} = x_k - \tau\nabla\ell(x_k), \\ x_{k+1} = \text{prox}_{\tau\lambda\mathcal{R}}(x_{k+\frac{1}{2}}). \end{cases} \tag{4}$$

2.2 Plug-and-Play priors

Instead of handcrafting a regularization term, data-driven approaches [40, 58] learn a suitable regularization term from a set of training data. The objective is to obtain a more expressive function that can better capture the structure of the true image. Recent approaches rely on Plug-and-Play priors [55], which leverage the power of denoising algorithms as prior models into iterative optimization algorithms.

The PnP framework stems from the observation that the proximal operator in Eq. (3) takes the form of a regularized image denoiser for additive white Gaussian noise and, as such, could be replaced by a denoising operator in proximal splitting algorithms. Initially developed for computed tomography [55], this approach has been applied to many other inverse problems, such as super-resolution [57], magnetic resonance imaging [3], or inpainting [9]. Besides, the PnP framework has been extended to a wide range of proximal splitting algorithms including proximal gradient descent with PnP-PGD [15, 47, 52, 58]. In recent instances of PnP-PGD, the proximal step is replaced by a pre-trained denoising network D_σ with trainable parameters θ, and parameter $\sigma > 0$ to control its denoising strength, and $k > 0$ the iteration step:

$$\begin{cases} x_{k+\frac{1}{2}} = x_k - \tau \nabla \ell(x_k), \\ x_{k+1} = D_\sigma(x_{k+\frac{1}{2}}; \theta). \end{cases} \tag{5}$$

The denoiser D_σ is trained to minimize a denoising error given a dataset $\mathcal{D}$ of clean images x^* and their corrupted version $x^* + \xi_\sigma$ with additive white Gaussian noise $\xi_\sigma \sim \mathcal{N}(0, \sigma^2 I)$

$$\min_\theta \ \mathbb{E}_{x^* \sim p(x^*)} \ \|D_\sigma(x^* + \xi_\sigma; \theta) - x^*\|_2^2, \tag{6}$$

where $p(x^*)$ is the distribution of clean images in $\mathcal{D}$.

2.3 Deep unrolled optimization

Deep unrolled or deep unfolded optimization [1, 2, 19, 22] can be seen as a variant of data-driven regularization. Instead of learning a generic denoising prior, the goal is to learn an iterative optimization algorithm, which usually produces better results with fewer iterations than the PnP framework. Like Plug-and-Play, the idea is to replace the proximal step in Eq. (4) with a neural network G and to unroll the algorithm for a finite number of steps $K > 0$. Given an initial point x_0, which is a degraded version of x^*, we write

$$\mathbb{H}(x_0; \theta, K) = \left[\bigcirc_{k=0}^{K} G(\cdot\,; \theta) \circ (\mathrm{Id} - \tau \nabla \ell) \right] (x_0) \tag{7}$$

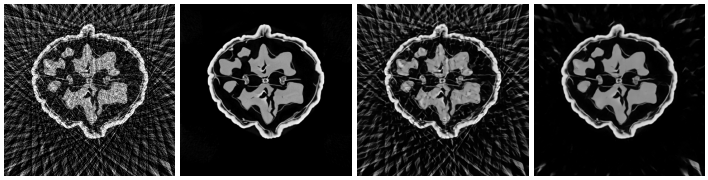
the unrolled computation resulting from K steps of optimization described in Eq. (4), where $\bigcirc$ is the compound composition operator. Once trained, the network G can be interpreted as the proximity operator of an *implicit* regularization function. In deep unfolding, the network G is trained end-to-end as part of the reconstruction process; thus, we optimize

$$\min_\theta \ \mathbb{E}_{x_0, x^* \sim p(x_0, x^*)} \ \|\mathbb{H}(x_0; \theta, K) - x^*\|_2^2. \tag{8}$$

While this approach improves the performance when compared to PnP priors, optimizing θ comes at a higher cost. Including the entire reconstruction process in the differentiation introduces an additional training cost. This cost scales with the number of iterations considered and, most importantly, with the size of the forward operator A, which is sampled at every iteration and represents the main bottleneck when training unrolled models [34].

3 Methodology

In this work, we propose a procedure to *explicitly* learn the proximity operator $\mathrm{prox}_{\mathcal{R}^*}$ of the reconstruction loss $\mathcal{R}^*$. We train a restoration operator $D(\cdot\,, k; \theta)$ to approximate the evaluation of $\mathrm{prox}_{\mathcal{R}^*}$ in every intermediate point x_k of the optimization path. This approach has three main advantages: **First**, the evaluation of the proximity operator in Eq. (4) only depends on the current iterate $x_{k+\frac{1}{2}}$, which allows to break down the training of D into independent subproblems, as opposed to unrolling Eq. (7). **Second**,



(a) Input **(b)** Ground-truth **(c)** Gaussian denoising network. **(d)** Post-processing network.

Fig. 1: Comparison of denoising and restoration priors on the Walnut-CBCT dataset. The *input* x_0 **(a)** is computed using the Feldkamp-Davis-Kress (FDK) algorithm [21] from the sparse-view projection data b (Eq. (2)). The *ground-truth* x^* **(b)** is computed using the FDK algorithm from the dense-view projection data. The *denoised* **(c)** and *restored* **(d)** images are obtained by applying one pass of a Gaussian denoising network (Eq. (6)) and one pass of a post-processing network (Eq. (10)), respectively.

as opposed to PnP, the network D is pre-trained on an entire reconstruction procedure, which allows us to obtain a robust *prior* in every point along the optimization path. **Third** the network D is a restoration operator, *i.e*. D is trained to remove artifacts from the specific degradation process, as opposed to denoising operators in classic PnP schemes [55]; in this work, we show that this choice leads to better reconstruction performance.

3.1 Reconstruction loss as regularizer

The objective of our work is to learn a prior tailored to the specific inverse problem without relying on unrolled optimization, *i.e*. without training D end-to-end.

In this work, we choose the reconstruction loss Eq. (9) as the regularization function $\mathcal{R}^*$ in a composite optimization problem Eq. (2); we then train a network D to approximate its proximity operator at inference. We motivate this choice using the following reasoning: ideally, the proximity operator $\text{prox}_{\mathcal{R}^*}(\cdot)$ in Eq. (4) should compute a preferred update direction such that the next iterate, x_{k+1}, is closer to the ground-truth, than the current iterate x_k. In other words, the proximity operator $\text{prox}_{\gamma\mathcal{R}^*}(\cdot)$ should be the projection onto the set of images that are closer to the ground-truth x^*. Hence, the regularizer can be defined as the squared Euclidean distance between an image x and its associated ground-truth x^*:

$$\mathcal{R}^*(x) = \frac{1}{2}\|x - x^*\|_2^2. \tag{9}$$

Inspired by Zhang *et al*. [57], we argue that while the prior used in PnP is trained for Gaussian denoising, this does not imply that the difference of the inputs from the ground-truth follows a Gaussian distribution. We build from this observation and show that our formulation brings a significant advantage compared to classic PnP schemes (Sec. 4). Using the reconstruction loss as the regularizer, we make sure that the prior is trained to remove artifacts from the specific degradation process (Fig. 1), which is more relevant to the reconstruction task and, in spirit, very similar to unrolled approaches.

As seen in Fig. 1, one pass of a Gaussian denoising network **(c)** is not able to remove the artifacts from the *sparse-view* degradation process **(a)**. At the same time, a post-processing network [25] **(d)** can remove most of the artifacts. We define a post-processing network as a network trained to predict the ground-truth x^* from an input image x_0

$$\min_{\theta} \ \mathbb{E}_{x_0,x^* \sim p(x_0,x^*)} \ \|D(x_0;\theta) - x^*\|_2^2. \tag{10}$$

3.2 Learned proximal operator

The regularizer $\mathcal{R}^*$ requires the ground-truth x^*, which is never available at inference. The objective is thus to train D to approximate the proximity operator of $\mathcal{R}^*$, namely D should remove the artifacts of x_0 in small steps rather than just one pass (Eq. (10)). Using the closed-form solution of the proximity operator of $\mathcal{R}^*$ [16], we define a regression target for D to approximate, *i.e*.

$$\text{prox}_{\gamma\mathcal{R}^*}(x) = \frac{x + \gamma x^*}{1+\gamma}, \tag{11}$$

with $\gamma > 0$ balancing how close the projection is from x^*. The training problem, thus, takes the form of

$$\min_{\theta} \mathbb{E}_{x\sim p(x)} \|D(x;\ \theta) - \text{prox}_{\lambda\tau\mathcal{R}^*}(x)\|_2^2. \tag{12}$$

Lastly, the main objective is to obtain a robust approximation of the proximity operator for every intermediate point x_k along the optimization trajectory. We define the optimization trajectory as the set $\{x_k\}_{k\in\mathbb{N}}$ generated by the PGD algorithm Eq. (4) with regularization function $\mathcal{R}^*$. Such trajectory (Fig. 2) can be built for every pair of initial point, ground-truth image (x_0, x^*) in the training dataset $\mathcal{D}$. Finally, we obtain the following optimization problem

$$\min_{\theta} \mathbb{E}_{x_0,x^*\sim p(x_0,x^*)} \mathbb{E}_{k\sim p(k)} \|D(x_{k+\frac{1}{2}}, k;\ \theta) - \text{prox}_{\lambda\tau\mathcal{R}^*}(x_{k+\frac{1}{2}})\|_2^2. \tag{13}$$

where $p(k)$ is a pre-defined distribution which samples $k \in [0, K]$.

In Eq. (13), we condition the output of D on the iteration step k. Similar to Gaussian denoising networks that take a noise level map as input [57], our intuition here is that D can adapt its restoration strength to the current state of the optimization trajectory. This conditioning approach is identical to the one used in ODE-based models such as Diffusion Models [28] or the recent Inversion by Direct Iteration [17].

3.3 Parametrization of the proximal operator

Equation (11) reveals that D must predict $\frac{x_{k+\frac{1}{2}} + \gamma x^*}{1+\gamma}$ given $x_{k+\frac{1}{2}}$. Since $x_{k+\frac{1}{2}}$ is available as input to the model, and γ is a pre-defined hyperparameter, we may choose another parametrization for D, *i.e*. the function that maps the current iterate $x_{k+\frac{1}{2}}$ to the ground-truth x^*

$$\min_{\theta} \mathbb{E}_{x_0,x^*\sim p(x_0,x^*)} \mathbb{E}_{k\sim p(k)} \|D(x_{k+\frac{1}{2}}, k;\ \theta) - x^*\|_2^2. \tag{14}$$

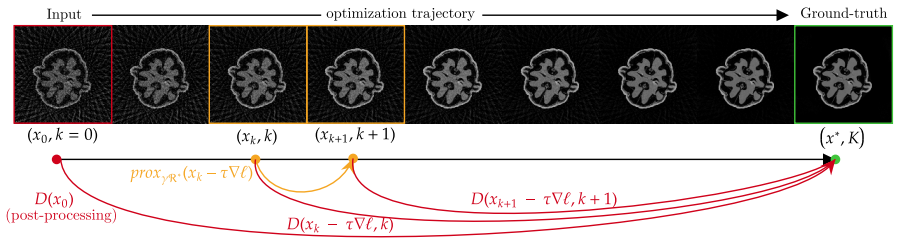


Fig. 2: Illustration of the pre-defined optimization trajectory. In Eq. (14), we train D to map each $x_{k+\frac{1}{2}} = x_k - \tau\nabla\ell$ to x^*. The proximity operator $\text{prox}_{\gamma\mathcal{R}^*}$ is defined as a small interpolation step between the current iterate $x_{k+\frac{1}{2}}$ and the ground-truth x^* (Eq. (15)).

Using this parametrization, the problem of learning the proximity operator of the reconstruction loss $\mathcal{R}^*$ reverts to learning to predict the ground-truth x^* from any point along the optimization trajectory. The proximity operator $\mathcal{R}^*$ is simply obtained as the linear combination of the current iterate $x_{k+\frac{1}{2}}$ and $\hat{x}_{k+\frac{1}{2}} = D(x_{k+\frac{1}{2}}, k; \theta)$:

$$x_{k+1} = \text{prox}_{\gamma\mathcal{R}^*}(x_{k+\frac{1}{2}}) \approx \frac{x_{k+\frac{1}{2}} + \gamma\hat{x}_{k+\frac{1}{2}}}{1+\gamma}. \tag{15}$$

In practice, we use the accelerated version of proximal gradient descent [7] that introduces an inertial step, Eq. (16), to the procedure

$$t_0 = 1, t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}, \quad q_k = \frac{t_k - 1}{t_{k+1}}. \tag{16}$$

We coin our method Learned Proximal Trajectory (LPT), the complete inference algorithm is detailed in Algorithm 1.

Algorithm 1 Plug-and-Play Learned Proximal Trajectory (LPT)

1: **Input:** $x_0 \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $k = 0$, $K > 0$, $\varepsilon > 0$, step size $\tau > 0$, regularization weight $\lambda > 0$, $\{q_k\}_{k\in\mathbb{N}}$ and network D.
2: **while** $k < K$ and $c_k > \varepsilon$ **do**:
- $x_{k+\frac{1}{2}} = x_k - \tau\nabla\ell(x_k)$ ▷ gradient step on fidelity-term
- $z_{k+1} = \dfrac{x_{k+\frac{1}{2}} + \gamma D(x_{k+\frac{1}{2}}, k)}{1+\gamma}$, with $\gamma = \tau\lambda$ ▷ learned proximal step
- $x_{k+1} = z_{k+1} + q_k(z_{k+1} - z_k)$ ▷ inertial step
- $c_k = \|x_{k+1} - x_k\|_2 / \|x_0\|_2$; $k = k+1$

3: **Output:** x_{k+1}.

3.4 Sampling intermediate points

To minimize the expectation in Eq. (14), one needs to efficiently and independently sample the intermediate reconstructions $x_{k+\frac{1}{2}} = x_k - \tau\nabla\ell(x_k)$ resulting from the

proximal gradient descent procedure. The PGD algorithm is iterative; therefore, no efficient way to sample those exists. Instead, we propose to use knowledge of the proximal operator $\text{prox}_{\gamma\mathcal{R}^*}$ during training to run a *target* optimization procedure Eq. (14) and save the intermediate reconstructions x_k on the disk. Furthermore, we choose to implement D as a 2D axial-wise operator to mitigate the memory requirements regarding storage and GPU complexity. This parametrization of D allows us to save only random slices of the intermediate reconstructions x_k instead of the whole volume. We argue that parametrizing the problem in that fashion is also one of the benefits of our approach, as it allows us to include knowledge of the forward operator A in the training process while maintaining an offline procedure, *i.e*. no call to A during the optimization of the parameters θ.

Parametrizing D as a 2D network introduces an axial bias during the reconstruction, but overall, it does not impede the 3D nature of the problem, as the forward and adjoint operators $A, A^\top$ remain 3D.

3.5 Convergence analysis

Our proposed method is adapted from the proximal gradient descent algorithm (Eq. (4)). Thus, we can apply the convergence analysis of PnP-PGD to our method. Most work on PnP convergence rely on *nonexpansiveness* of D [42, 52]; here we use the analysis of Cohen *et al*. [15] to show that our method converges for a chosen step size γ.

Let us define $D_\alpha = \alpha D + (1-\alpha)\text{Id}$, the relaxed operator stemming from D with $\alpha = \frac{\gamma}{1+\gamma}$. We see that one estimation of $\text{prox}_{\gamma\mathcal{R}^*}$ is exactly the computation of D_α. Thus, in compact form, the inference operator can be defined as

$$T(x) := D_\alpha(x - \tau\nabla\ell(x)), \quad \text{with } \alpha = \tfrac{\gamma}{1+\gamma}. \tag{17}$$

d-demicontractive operator. Cohen *et al*. [15] proved the convergence of PnP-PGD and its relaxed version PnP-αPGD, for a broad range of functions by considering d-demicontractive operators. Given that ℓ is convex and differentiable with a Lipchitz continuous gradient $L > 0$ and that a fixed point of T exists, they show that for a continuous d-demicontractive operator D, possibly *expansive*, if $\alpha \in (0, \frac{1-d}{2})$ and $\tau \in (0, \frac{2}{L})$, the sequence of iterates $\{x_k\}_{k\in\mathbb{N}}$ generated by $x_{k+1} = T(x_k)$ converges to a fixed point $x^* \in \mathbb{R}^n$ of T.

Furthermore, assuming that D is a β-Lipschitz continuous function, we show that D is also d-demicontractive with $d = 1 - \frac{2}{\beta+1}$. We derive a simple condition on the regularization step size $\gamma = \tau\lambda$, that is, $\gamma\beta < 1$. Hence, the regularization step size γ is limited by β, the Lipschitz constant of D (see *Supplementary Materials* for full development and details on *demicontractivity*).

This result is crucial as it provides the means to plug a broad range of relaxed restoration operators D_α in the PnP-PGD algorithm. In our case, we could estimate a lower bound on the Lipschitz constant of D by computing $\|J_D\|$, the maximal spectral norm of its Jacobian, over a dataset of images $\mathcal{D}$ [42].

In Sec. 4 and Fig. 5, we show that an appropriately relaxed operator D_α converges to a fixed-point while obtaining state-of-the-art performances. Furthermore, in Sec. 4.4, we empirically show that constraining the Lipschitz constant of D diminishes the performance of our Learned Proximal Trajectory scheme.

4 Experiments

In this section, we describe the training procedure of our Learned Proximal Trajectory approach and compare it with the state-of-the-art methods on the task of 3D sparse-view cone-beam computed tomography. We also provide a detailed ablation study to analyze the effect of different components of our approach.

4.1 Datasets

We conduct experiments on the **Walnut-CBCT** dataset [18] and on a private **Cork-CBCT** dataset as well. Both datasets contain experimental 3D Cone-Beam CT measurements, as opposed to simulation-based datasets [13, 33, 38] which simulate X-ray measurement from ground-truth images, and thus are limited in terms of accurately modeling the physics. The Walnut-CBCT dataset contains acquisitions with 1200 projections of size 972×768 and ground-truth reconstructions of size 501^3. The Cork-CBCT dataset contains acquisitions with 720 projections of size 1024^2 and reconstructions of size 1024^3. The ground-truth x^* is obtained for both datasets by running a proximal gradient descent with all measurements. We regularly sample $30, 50$ and 100 projections for each dataset and use the sparse FDK reconstructions as input images x_0.

For each dataset and each method, we evaluate the quality of the reconstructions using the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [56]. The detailed procedure is given in the *Supplementary Materials*.

4.2 Learned proximal trajectory training details.

Similar to recent PnP and imaging inverse problem approaches, we choose to parametrize D with a Deep Residual U-Net [22, 45, 57, 60]. During training, the input images are randomly cropped to 256^2 patches. The network is trained to minimize the L^2 loss in Eq. (14). On both datasets, we use the Adam optimizer [32], a learning rate of 10^{-4} and a batch size of 32. We train the network for 200k iterations with cosine annealing and a warm-up of 500 iterations. We set $\tau = \frac{1}{L}$ with L the Lipschitz constant of the data-fidelity gradient $\nabla \ell$ for both datasets. Algorithm 1 is run for $K = 500$ iterations and $\varepsilon = 10^{-4}$. We give more details in the *Supplementary Materials*.

4.3 3D sparse-view CT

Baselines. As traditional baselines, we evaluate the performance of the analytical reconstruction method, the Feldkamp Davis Kress (FDK) algorithm [21], and the Total-Variation (TV) regularized iterative reconstruction [48].

State-of-the-art methods. These experiments compare our method with traditional iterative and learned approaches for sparse-view CT reconstruction. First, we evaluate the performance of FDK-UNet, a supervised post-processing approach that computes a one-step restoration given a sparse-view FDK reconstruction [25, 59]. We evaluate the PnP-PGD algorithm, the standard proximal-gradient formulation of the PnP algorithm [15, 52, 55] using a learned Gaussian denoiser. Finally, we compare our method

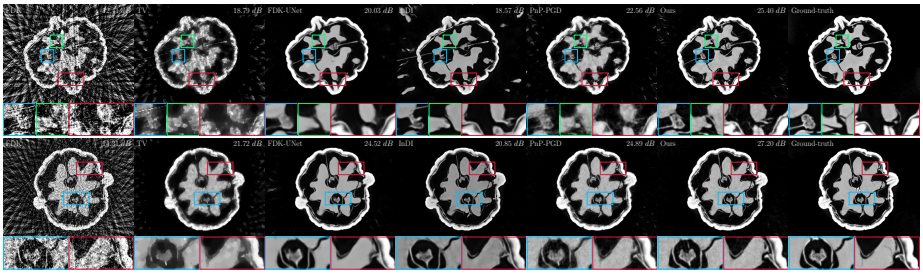


Fig. 3: Illustrations of sparse view reconstructions on the Walnut-CBCT [18] dataset using the methods compared in Tab. 1. *First row* (30/1200), *second row* (50/1200).

with the recent Inversion by Direct Iteration (InDI) [17], also proposing a supervised iterative restoration procedure. We use the same DRUNet backbone architecture for fair comparisons across all experiments; detailed parameters are given in the *Supplementary Materials*.

While PnP algorithms come in many different forms [31, 40], we focus the comparison on the most standard formulation, PnP-PGD. For efficient and fair comparison, we avoid approaches that add layers of memory or computational complexity that scale poorly with high-resolution 3D computed tomography, which includes gradient-step denoisers [14,30,31], iterative algorithms that require solving a data proximal subproblem [35, 57, 60], or diffusion models [11, 12]. As an indication, on the Cork-CBCT dataset, the inference time for 500 iterations of PnP-PGD or LPT is approximately 10 hours with 2 NVIDIA V100 GPUs.

Table 1: Reconstruction performances on the **Walnut-CBCT** dataset, with *best* and *second-best* results highlighted. We compare state-of-the-art approaches against our method and vary the number of views. $\ell/m = \frac{1}{2m}\|Ax-b\|_2^2$ is the normalized data-fidelity (Eq. (2)).

Walnut-CBCT - **3D**	**SSIM** ↑			**PSNR** ↑			ℓ/m ↓		
Views $(\cdot/1200)$	30	50	100	30	50	100	30	50	100
FDK [21]	0.153	0.206	0.280	12.83	15.40	18.75	1.06e−1	6.13e−2	2.84e−2
TV [48]	0.555	0.646	0.783	19.27	21.77	25.29	2.99e−4	1.93e−4	2.35e−4
FDK-UNet	0.767	0.865	0.838	22.24	26.56	25.28	7.82e−4	3.73e−4	1.34e−3
InDI [17]	0.696	0.829	0.816	20.04	21.68	21.85	3.92e−3	6.68e−3	4.39e−3
PnP-FBS [47]	0.693	0.764	0.829	22.94	25.88	28.17	7.56e−5	7.78e−5	9.12e−5
Ours: LPT	0.763	0.806	0.838	25.26	27.98	29.63	6.69e−5	7.77e−5	9.04e−5

Results. In Tab. 1, we provide a detailed comparison with the state-of-the-art and historical approaches on the 3D Walnut-CBCT dataset. Our method outperforms all the baselines and state-of-the-art methods regarding PSNR, while FDK-UNet ranks first in SSIM. The qualitative results in Fig. 3 suggest that methods with no feedback from the data-fidelity term, such as FDK-UNet or InDI, lack geometrical consistency and are more prone to modify the reconstructed surface of the object. The TV baseline pro-

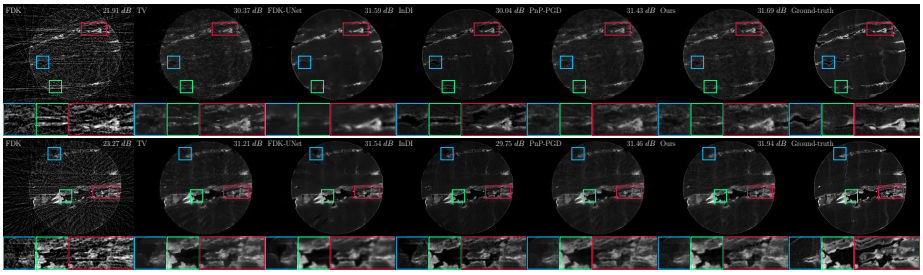


Fig. 4: Illustrations of sparse view reconstructions on the Cork-CBCT dataset using the methods compared in Tab. 2. *First row* (30/720), *second row* (50/720).

vides a more geometrically consistent reconstruction, although it removes important details. On the other hand, LPT and PnP-PGD provide the best results in geometrical consistency and detail preservation, with LPT providing sharper details and a better PSNR. We validate this observation using results on data-fidelity ℓ. We see in Tab. 1 that variational methods are the most data-consistent, *i.e*. they optimize a data-fidelity objective ℓ and are *a fortiori* the most faithful to the measurements. On the other hand, learned-only methods such as FDK-UNet maximize PSNR but do not respect experimental observations, which poses severe issues in critical applications such as medical imaging or non-destructive testing.

Table 2: Reconstruction performances on the **Cork-CBCT** dataset, with *best* and *second-best* results highlighted. We compare state-of-the-art approaches against our method and vary the number of views. $\ell/m = \frac{1}{2m}\|Ax - b\|_2^2$ is the normalized data-fidelity (Eq. (2)).

Cork-CBCT - **3D**	**SSIM** ↑			**PSNR** ↑			ℓ/m ↓		
Views $(\cdot/720)$	30	50	100	30	50	100	30	50	100
FDK [21]	0.164	0.258	0.321	22.02	24.11	27.15	4.51e−2	6.13e−2	1.17e−2
TV [48]	0.775	0.783	0.801	30.76	31.32	32.15	2.98e−4	3.23e−4	3.40e−4
FDK-UNet	0.823	0.858	0.858	32.38	33.23	33.93	1.23e−3	1.20e−3	1.21e−3
InDI [17]	0.816	0.820	0.836	30.54	31.01	31.61	1.79e−3	2.18e−3	1.92e−3
PnP-FBS [47]	0.822	0.842	0.855	32.01	33.01	33.73	2.88e−4	3.19e−4	3.27e−4
Ours: LPT	0.810	0.844	0.850	32.11	33.60	34.11	2.70e−4	3.11e−4	3.24e−4

Table 2 also provides results on the Cork-CBCT dataset. Overall, we observe the same trends as for the Walnut-CBCT dataset. Our Learned Proximal Trajectory approach outperforms the compared methods and ranks first in PSNR, while FDK-UNet ranks first in SSIM. Visually, we see in Fig. 4 that the InDI approach seems to provide sharper results than the other methods. However, similarly to the illustrations on the Walnut-CBCT dataset, FDK-UNet and InDI also create structure that are not present in the ground-truth. On the other hand, the PnP-PGD approach and our LPT method create less sharp reconstructions, though they maintain more accurate structures.

Table 3: Ablation study on the **Walnut-CBCT** dataset and the **Cork-CBCT** dataset. For each line we run the inference procedure given in Algorithm 1 with different networks D. The evaluations are done on the validation set. Each study builds up from the previous experience, with *best* results highlighted. $\|J_D\|$ denotes the maximum value of the Jacobian spectral norm over the optimization trajectory $\{x_k\}_{k\in\mathbb{N}}$.

	Walnut-CBCT - 3D			Cork-CBCT - 3D		
Optimization of D	**SSIM** ↑	**PSNR** ↑	$\|J_D\|$	**SSIM** ↑	**PSNR** ↑	$\|J_D\|$
A Baseline (FDK-UNet) - Eq. (10)	0.722	25.67	19.28	0.828	33.69	3.75
B + Trained on optimization trajectory (LPT)	0.816	28.30	8.82	0.837	33.74	13.0
C + Iteration step conditioning	0.818	28.44	10.9	0.843	34.41	10.2
D + Lipschitz constraint.	0.767	26.18	1.24	0.823	33.06	0.969

4.4 Ablation studies

In this section, we analyze the convergence of our method and conduct ablation studies on the validation sets to emphasize the effect of different components of our approach. In the experiments Tab. 3, λ is set as in Sec. 4, we give more details in the *Supplementary Materials*.

Training inputs from optimization trajectory. As emphasized in the closed-form expression of $\mathrm{prox}_{\lambda\mathcal{R}^*}$, one only needs an approximation $\hat{x}$ of the ground-truth x^* to compute a learned proximity update. In that regard, any network D trained to predict x^* from a degraded version x_0 is a good candidate network. However, we show that the choice of the training procedure significantly impacts the performance of the learned proximity operator. Specifically, we emphasize in Tab. 3 (B) the importance of sampling training inputs from the optimization trajectory during training rather than just sampling the initial degraded image x_0 (A) to obtain a robust approximation of x^*.

Iteration step conditioning. In Eq. (13), we see that the iteration step k is also given as input to the learned proximity operator. We show in Tab. 3 (C) that conditioning the learned proximity operator on the iteration step k consistently improves the results on both datasets. The results in Tabs. 1 and 2 are produced with the (C) configuration.

Lipschitz constraint. Using the convergence analysis of Cohen *et al.*, we show in Sec. 3.5 that a Lipschitz-constrained network D is not necessary to obtain convergence. In Tab. 3 (D), we emphasize that constraining the network D significantly reduces the

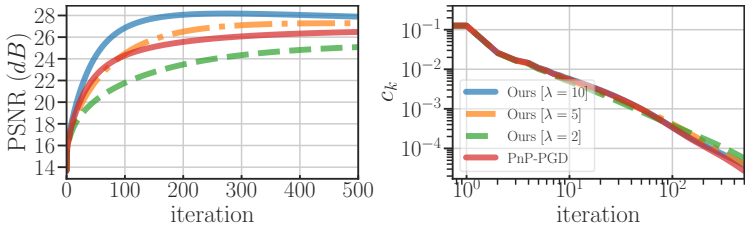


Fig. 5: Illustration of the convergence of our method on the Walnut-CBCT dataset [50/1200]. PSNR evolution (*left*) during the optimization. Fixed-point criterion c_k (*right*) during the optimization.

expressivity of the network D, and, thus, the reconstruction performances of our PnP scheme. We use the same procedure as Pesquet *et al*. [42] and constrain the Lipschitz constant of D by regularizing the spectral normal of its Jacobian.

Convergence analysis. We also provide convergence figures in Fig. 5. We observe that for an appropriately relaxed operator D, all the iterations converge to a fixed point with c_k decreasing log-linearly. In practice, the Lipschitz constant of $\nabla \ell$ being large, it forces $\tau \in (0, \frac{2}{L})$ to be very small and allows for relatively loose settings on λ as we only need to respect $\tau\lambda\beta < 1$. Although we see in Fig. 5 that up to a value of $\lambda = 10$, the increase in PSNR is not monotone, suggesting that we reach a limit in the quality of the reconstruction attainable.

5 Limitations and future work

In this work, we sample from a pre-defined optimization trajectory to train the restoration network. In Section 3.4, we first run a target optimization algorithm to generate the set of intermediate images x_k and store them on the disk. We argue that this procedure limits scalability and trades off memory usage for sampling speed during training. In future works, we plan to investigate the possibility of defining a simpler optimization trajectory that would not require storing intermediate images on the disk. Consequently, the choice of optimization trajectory is also crucial and can significantly impact the performance of the learned restoration operator. Hence, a principled way to select a regularization weight λ that balances data fidelity and regularization is also essential before training. Finally, the framework of our method is general and only considers a linear inverse problem; in that fashion, this work could also be applied to intensive computational imaging modalities such as Magnetic Resonance Imaging (MRI).

6 Conclusion

This paper presents a novel approach to train the learned operator D in a Plug-and-Play framework. First, we observe that Gaussian denoising networks are not the most effective operators for removing task-specific artifacts. Building on this observation, we propose to replace the Gaussian denoising network with a restoration network, *i.e*. a network trained to remove arbitrary artifacts. Second, to avoid the computational cost of unrolling optimization, we propose a procedure to train the restoration network as a robust approximation of a proximal operator along a pre-defined optimization trajectory. We confirm that sampling from a target optimization trajectory to train the network is crucial to learning a robust restoration network. Finally, we verify that using a PnP scheme with our restoration prior converges to a fixed point without constraining its Lipschitz constant. We demonstrate the effectiveness of our approach on the challenging task of 3D sparse-view X-ray computed tomography and outperform state-of-the-art methods in terms of PSNR. Our method provides experimentally grounded reconstructions with better geometrical consistency than FDK-UNet. Our approach maximizes data fidelity while maintaining a high level of quantitative reconstruction.

Acknowledgments

The work of R. Vo was supported by the FOCUS-EJN grant from CEA and made possible by the use of the FactoryIA supercomputer, financially supported by the Ile-de-France Regional Council. We thank Hoang Trieu Vy Le and Stéphane Lathuilière for their valuable feedback and discussions on this work.

References

1. Adler, J., Öktem, O.: Solving ill-posed inverse problems using iterative deep neural networks. Inverse Problems (2017). https://doi.org/10.1088/1361-6420/aa9581
2. Adler, J., Öktem, O.: Learned primal-dual reconstruction. IEEE Transactions on Medical Imaging (2018). https://doi.org/10.1109/TMI.2018.2799231
3. Ahmad, R., Bouman, C.A., Buzzard, G.T., Chan, S., Liu, S., Reehorst, E.T., Schniter, P.: Plug-and-play methods for magnetic resonance imaging: Using denoisers for image recovery. IEEE Signal Processing Magazine (2020). https://doi.org/10.1109/MSP.2019.2949470
4. Andrychowicz, M., Denil, M., Gómez, S., Hoffman, M.W., Pfau, D., Schaul, T., Shillingford, B., de Freitas, N.: Learning to learn by gradient descent by gradient descent. In: Adv. Neural Inform. Process. Syst. (2016)
5. Bai, S., Kolter, J.Z., Koltun, V.: Deep equilibrium models. In: Adv. Neural Inform. Process. Syst. (2019)
6. Bajić, B., Öktem, O., Rudzusika, J.: 3d helical ct reconstruction with memory efficient invertible learned primal-dual method. arXiv preprint arXiv:2205.11952 (2022). https://doi.org/10.48550/arXiv.2205.11952
7. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences (2009). https://doi.org/10.1137/080716542
8. Benning, M., Burger, M.: Modern regularization methods for inverse problems. Acta Numerica (2018). https://doi.org/10.1017/S0962492918000016
9. Chan, S.H.: Performance analysis of plug-and-play admm: A graph signal processing perspective. IEEE Transactions on Computational Imaging (2019). https://doi.org/10.1109/TCI.2019.2892123
10. Chen, H., Zhang, Y., Zhang, W., Liao, P., Li, K., Zhou, J., Wang, G.: Low-dose CT via convolutional neural network. Biomedical Optics Express (2017). https://doi.org/10.1364/BOE.8.000679
11. Chung, H., Kim, J., Mccann, M.T., Klasky, M.L., Ye, J.C.: Diffusion Posterior Sampling for General Noisy Inverse Problems. In: Int. Conf. Learn. Represent. (2023)
12. Chung, H., Sim, B., Ryu, D., Ye, J.C.: Improving Diffusion Models for Inverse Problems using Manifold Constraints. In: Adv. Neural Inform. Process. Syst. (2022)
13. Coban, S.B., Andriiashen, V., Ganguly, P.S., van Eijnatten, M., Batenburg, K.J.: Parallel-beam X-ray CT datasets of apples with internal defects and label balancing for machine learning. arXiv preprint arXiv:2012.13346 (2020)
14. Cohen, R., Blau, Y., Freedman, D., Rivlin, E.: It Has Potential: Gradient-Driven Denoisers for Convergent Solutions to Inverse Problems. In: Adv. Neural Inform. Process. Syst. (2021)
15. Cohen, R., Elad, M., Milanfar, P.: Regularization by Denoising via Fixed-Point Projection (RED-PRO). SIAM Journal on Imaging Sciences (Jan 2021). https://doi.org/10.1137/20M1337168

16. Combettes, P.L., Pesquet, J.C.: Proximal Splitting Methods in Signal Processing. In: Fixed-Point Algorithms for Inverse Problems in Science and Engineering, pp. 185–212. Springer Optimization and Its Applications, Springer (2011). https://doi.org/10.1007/978-1-4419-9569-8_10
17. Delbracio, M., Milanfar, P.: Inversion by direct iteration: An alternative to denoising diffusion for image restoration. Trans. Mach. Learn Res. (2023)
18. Der Sarkissian, H., Lucka, F., van Eijnatten, M., Colacicco, G., Coban, S.B., Batenburg, K.J.: A cone-beam X-ray computed tomography data collection designed for machine learning. Scientific Data (2019). https://doi.org/10.1038/s41597-019-0235-y
19. Ding, Q., Chen, G., Zhang, X., Huang, Q., Gao, H.J.H.: Low-Dose CT with Deep Learning Regularization via Proximal Forward Backward Splitting. Physics in Medicine & Biology (2020). https://doi.org/10.1088/1361-6560/ab831a
20. Engl, H.W., Hanke-Bourgeois, M., Neubauer, A.: Regularization of Inverse Problems. Mathematics and Its Applications, Kluwer Acad. Publ (2000)
21. Feldkamp, L.A., Davis, L.C., Kress, J.W.: Practical cone-beam algorithm. J. Opt. Soc. Am. A (1984). https://doi.org/10.1364/JOSAA.1.000612
22. Genzel, M., Gühring, I., Macdonald, J., März, M.: Near-Exact Recovery for Tomographic Inverse Problems via Deep Learning. In: Int. Conf. Mach. Learn. (2022)
23. Getreuer, P.: Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman. Image Processing On Line (2012). https://doi.org/10.5201/ipol.2012.g-tvd
24. Gilton, D., Ongie, G., Willett, R.: Deep Equilibrium Architectures for Inverse Problems in Imaging. IEEE Transactions on Computational Imaging (2021). https://doi.org/10.1109/TCI.2021.3118944
25. Han, Y.S., Yoo, J., Ye, J.C.: Deep Residual Learning for Compressed Sensing CT Reconstruction via Persistent Homology Analysis. arXiv preprint arXiv:1611.06391 (2016)
26. Hanke, R., Fuchs, T., Uhlmann, N.: X-ray based methods for non-destructive testing and material characterization. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment (2008). https://doi.org/10.1016/j.nima.2008.03.016
27. Heaton, H., Wu Fung, S., Gibali, A., Yin, W.: Feasibility-based fixed point networks. Fixed Point Theory and Algorithms for Sciences and Engineering (2021). https://doi.org/10.1186/s13663-021-00706-3
28. Ho, J., Jain, A., Abbeel, P.: Denoising Diffusion Probabilistic Models. In: Adv. Neural Inform. Process. Syst. (2020)
29. Hu, Y., Delbracio, M., Milanfar, P., Kamilov, U.: A restoration network as an implicit prior. In: Int. Conf. Learn. Represent. (2024)
30. Hurault, S., Leclaire, A., Papadakis, N.: Gradient Step Denoiser for convergent Plug-and-Play. In: Int. Conf. Learn. Represent. (2022)
31. Hurault, S., Leclaire, A., Papadakis, N.: Proximal Denoiser for Convergent Plug-and-Play Optimization with Nonconvex Regularization. In: Int. Conf. Mach. Learn. PMLR (2022)
32. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
33. Leuschner, J., Schmidt, M., Baguer, D.O., Maaß, P.: The LoDoPaB-CT Dataset: A Benchmark Dataset for Low-Dose CT Reconstruction Methods. Scientific Data (2021). https://doi.org/10.1038/s41597-021-00893-z
34. Leuschner, J., Schmidt, M., Ganguly, P.S., Andriiashen, V., Coban, S.B., Denker, A., Bauer, D., Hadjifaradji, A., Batenburg, K.J., Maass, P., van Eijnatten, M.: Quantitative Comparison of Deep Learning-Based Image Reconstruction Methods for Low-Dose and Sparse-Angle CT Applications. Journal of Imaging (2021). https://doi.org/10.3390/jimaging7030044

35. Liu, J., Anirudh, R., Thiagarajan, J.J., He, S., Mohan, K.A., Kamilov, U.S., Kim, H.: DOLCE: A Model-Based Probabilistic Diffusion Framework for Limited-Angle CT Reconstruction. In: Int. Conf. Comput. Vis. (2023)
36. Liu, J., Sun, Y., Eldeniz, C., Gan, W., An, H., Kamilov, U.S.: RARE: Image Reconstruction using Deep Priors Learned without Ground Truth. IEEE Journal of Selected Topics in Signal Processing (2020). https://doi.org/10.1109/JSTSP.2020.2998402
37. Liu, J., Xu, X., Gan, W., Shoushtari, S., Kamilov, U.: Online Deep Equilibrium Learning for Regularization by Denoising. In: Adv. Neural Inform. Process. Syst. (2022)
38. McCollough, C., Chen, B., Holmes III, D.R., Duan, X., Yu, Z., Yu, L., Leng, S., Fletcher, J.: Low Dose CT Image and Projection Data (LDCT-and-Projection-data) (2020). https://doi.org/10.7937/9NPB-2637
39. McCollough, C.H., Bartley, A.C., Carter, R.E., Chen, B., Drees, T.A., Edwards, P., Holmes III, D.R., Huang, A.E., Khan, F., Leng, S., McMillan, K.L., Michalak, G.J., Nunez, K.M., Yu, L., Fletcher, J.G.: Low-dose CT for the detection and classification of metastatic liver lesions: Results of the 2016 Low Dose CT Grand Challenge. Medical Physics (2017). https://doi.org/10.1002/mp.12345
40. Meinhardt, T., Moeller, M., Hazirbas, C., Cremers, D.: Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems. In: Int. Conf. Comput. Vis. (2017). https://doi.org/10.1109/ICCV.2017.198
41. Ongie, G., Jalal, A., Metzler, C.A., Baraniuk, R.G., Dimakis, A.G., Willett, R.: Deep learning techniques for inverse problems in imaging. IEEE Journal on Selected Areas in Information Theory (2020). https://doi.org/10.1109/JSAIT.2020.2991563
42. Pesquet, J.C., Repetti, A., Terris, M., Wiaux, Y.: Learning Maximally Monotone Operators for Image Recovery. SIAM Journal on Imaging Sciences (Jan 2021). https://doi.org/10.1137/20M1387961
43. Reehorst, E.T., Schniter, P.: Regularization by Denoising: Clarifications and New Interpretations. IEEE Transactions on Computational Imaging (2019). https://doi.org/10.1109/TCI.2018.2880326
44. Romano, Y., Elad, M., Milanfar, P.: The Little Engine That Could: Regularization by Denoising (RED). SIAM Journal on Imaging Sciences (2017). https://doi.org/10.1137/16M1102884
45. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI (2015). https://doi.org/10.1007/978-3-319-24574-4_28
46. Rudzusika, J., Bajic, B., Öktem, O., Schönlieb, C.B., Etmann, C.: Invertible learned primal-dual. In: NeurIPS 2021 Workshop on Deep Learning and Inverse Problems (2021)
47. Ryu, E., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-Play Methods Provably Converge with Properly Trained Denoisers. In: Int. Conf. Mach. Learn. PMLR (2019)
48. Sidky, E.Y., Jørgensen, J.H., Pan, X.: Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm. Physics in Medicine and Biology (2012). https://doi.org/10.1088/0031-9155/57/10/3065
49. Sidky, E.Y., Lorente, I., Brankov, J.G., Pan, X.: AAPM 2021 DL-sparse-view CT Grand Challenge. arXiv preprint arXiv:2005.10755 (2020)
50. Sidky, E.Y., Pan, X.: Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. Physics in Medicine and Biology (2008)
51. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep Unsupervised Learning using Nonequilibrium Thermodynamics. In: Int. Conf. Mach. Learn. PMLR (2015)
52. Sun, Y., Wohlberg, B., Kamilov, U.S.: An Online Plug-and-Play Algorithm for Regularized Image Reconstruction. IEEE Transactions on Computational Imaging (2019). https://doi.org/10.1109/TCI.2019.2893568

53. Tang, J., Mukherjee, S., Schönlieb, C.B.: Accelerating Deep Unrolling Networks via Dimensionality Reduction (2022)
54. Terris, M., Repetti, A., Pesquet, J.C., Wiaux, Y.: Building Firmly Nonexpansive Convolutional Neural Networks. In: ICASSP (2020). https://doi.org/10.1109/ICASSP40776.2020.9054731
55. Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-Play priors for model based reconstruction. In: IEEE Global Conference on Signal and Information Processing (2013). https://doi.org/10.1109/GlobalSIP.2013.6737048
56. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: From error visibility to structural similarity. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society (2004). https://doi.org/10.1109/tip.2003.819861
57. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE Trans. Pattern Anal. Mach. Intell. (2022). https://doi.org/10.1109/TPAMI.2021.3088914
58. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: IEEE Conf. Comput. Vis. Pattern Recog. (2017). https://doi.org/10.1109/CVPR.2017.300
59. Zhang, Z., Liang, X., Dong, X., Xie, Y., Cao, G.: A Sparse-View CT Reconstruction Method Based on Combination of DenseNet and Deconvolution. IEEE Transactions on Medical Imaging (2018). https://doi.org/10.1109/TMI.2018.2823338
60. Zhu, Y., Zhang, K., Liang, J., Cao, J., Wen, B., Timofte, R., Van Gool, L.: Denoising Diffusion Models for Plug-and-Play Image Restoration. In: IEEE Conf. Comput. Vis. Pattern Recog. (2023)
61. Zou, Z., Liu, J., Wohlberg, B., Kamilov, U.S.: Deep Equilibrium Learning of Explicit Regularizers for Imaging Inverse Problems. arXiv preprint arXiv:2303.05386 (2023). https://doi.org/10.48550/arXiv.2303.05386