Fixed point method for PET reconstruction with plug-and-play regularization

Marion Savanier, Claude Comtat, Florent Sureau

▶ **To cite this version:**

Marion Savanier, Claude Comtat, Florent Sureau. Fixed point method for PET reconstruction with plug-and-play regularization. 2025. hal-04951474

HAL Id: hal-04951474
https://hal.science/hal-04951474v1

Preprint submitted on 17 Feb 2025

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.

Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

Fixed point method for PET reconstruction with plug-and-play regularization

Marion Savanier, Claude Comtat, Florent Sureau

***Abstract*—Deep learning has shown great promise for improving medical image reconstruction, often surpassing traditional model-based iterative methods. However, concerns remain about the stability and robustness of these approaches, particularly when trained on limited data. The Plug-and-Play framework offers a promising solution, showing that a convergent and robust reconstruction can be ensured, provided conditions on the plugged network. Yet, it has been underexplored in PET reconstruction. This paper introduces a convergent PnP algorithm for low-count PET reconstruction, leveraging the Douglas-Rachford splitting method and various networks trained on the fixed point conditions. We evaluate bias-standard deviation tradeoffs across multiple regions including an unseen pathological case and compared to model-based iterative reconstruction, post-reconstruction processing, and PnP with a Gaussian denoiser. Our findings emphasize the importance of how convergence conditions are imposed on the PnP networks. While spectral normalization underperformed, our deep equilibrium model remained competitive with convolutional architectures and generalized better on our unseen pathology. Our method achieved lower bias than post-reconstruction processing and reduced standard deviation at matched bias compared to model-based iterative reconstruction. Our results demonstrate PnP's potential to improve image quality and quantification accuracy in PET reconstruction.**

***Index Terms*—Low-count PET reconstruction, plug-and-play, convergence, Douglas-Rachford splitting, deep equilibrium model, constrained architectures**

I. Introduction

Positron emission tomography (PET) is widely used in oncology, neurology, and cardiology but reconstruction remains challenging due to its low spatial resolution and the presence of Poisson noise, compromising accurate quantification, especially in low-count imaging scenarios.

Over the past decade, model-based iterative methods combining a Poisson likelihood with handcrafted priors [1] have been increasingly replaced by deep learning approaches [2]. Recent advances focus on learning priors from databases of high-quality PET images, capturing underlying features specific to a given protocol. This approach is at the basis of hybrid methods that integrate the representation power of modern deep learning with the generalization capabilities of model-based iterative reconstruction thanks to their feedback mechanisms enforcing data consistency [3], [4].

One of such hybrid methods consists of unfolding a low number of iterations of a reconstruction algorithm [5] and learning parameters related to the regularization decoupled across iterations - now layers. Unfolding networks (or variational networks) thus integrate the learning of regularizers with the reconstruction task. While the original algorithm might compute a Maximum *a posteriori* (MAP) estimate, the unfolded iterations do not. Unfolding networks have demonstrated high performance and reduced susceptibility to hallucinations compared to standard deep learning post-processing for some inverse problems [6]. Yet, with iteration-dependent parameters, the connection between unfolding networks and the iterative algorithm from which it is derived no longer exists, and the network may exhibit discontinuity with respect to the data. Moreover, these networks are not easily amenable to 3D imaging because of the high memory load for backpropagation, limiting the number of unfolded iterations and, thus, the solution search space. This highlights the necessity of using a good initial input image. Given these drawbacks, unfolding networks perform best when the inverse problem is not overly ill-posed or when a preconditioner is employed [7].

A more scalable hybrid technique is the *plug-and-play* (PnP) approach [8], [9], where the proximity operator or the gradient of the regularization[1] is replaced by a pre-trained neural network within the iterations of an optimization algorithm. Under algorithm-dependent conditions on the network, classical optimization algorithms can be shown to converge to either a MAP estimate using results from nonconvex theory [11]–[14] or to a fixed point by leveraging results from monotone operators [15]–[17].

Various PnP methods exist, some aiming to learn smooth approximations to the score function - the gradient of the log of the "true" prior of PET images - using (non-blind) Gaussian denoisers [18] or, more recently, inverse gamma denoisers to replace gradients in a Bregman geometry [19]. These score-based approaches involve an additional noise level hyper-parameter, which ideally should depend on the size of the database [20]. However, in practice, this parameter requires tuning for each application and patient, making it unclear how well the score function is approximated, especially with limited training data, as in medical imaging. Score-based denoisers have also been used to replace (Bregman [19]) proximity operators, albeit at the cost of losing the theoretical interpretation of the learned prior [21]. Another class of PnP methods leverages the monotone operator theory, generalizing iterative convex algorithms [22]. These methods use learned monotone operators [23] or learned resolvents of monotone operators, requiring the neural network to be an averaged

M. Savanier, C. Comtat and F. Sureau are with BioMaps, Université Paris-Saclay, CEA, CNRS, Inserm, SHFJ, 91401 Orsay, France (e-mail: firstname.lastname@cea.fr).

[1]These methods are often referred to as RED (Regularization by Denoising) [10]

operator [15], [17], [25].

The choice of the training task for the plugged denoiser is often overlooked, with a primary focus on architectural constraints to ensure convergence. Most studies use Gaussian denoisers [15], [25], but alternatives exist, particularly in medical imaging, where more general artifact removal approaches are employed. These artifacts are related to the target inverse problem and are closer to those seen on images along the PnP iterations, such as subsampling streak artifacts in MRI [26]. The PnP paradigm has seen applications in MRI imaging [26]–[29], but there has been limited exploration in PET imaging.

In this work, we extend our previous results [30] to present a PnP method for PET reconstruction. Our goal is to reconstruct images with the quality of high-count images from low-count data to reduce the injected activity/acquisition time without degrading quantification. We focus on both the training task and the neural network's architecture. Drawing inspiration from deep unfolding, we do not use a task-independent denoiser but learn an operator related to our reconstruction task in a PnP manner. Unlike deep unfolding, the network is not trained within the iterations of the optimization algorithm, but it is trained to solve the variational inclusion that arises from our reconstruction problem. The PnP iterates are guaranteed to converge to a unique fixed point under conditions on the learned operator that are enforced either during training or by design.

The paper is organized as follows. Section II formulates the regularized PET reconstruction problem and presents approaches for implicitly regularizing the reconstruction using deep learning. Section III introduces the PnP method, its convergence conditions, and comments on how to implement them. Section IV describes the simulations and data used in the evaluation. Experimental results are shown in section V, followed by discussions in section VI. Finally, conclusions are drawn in section VII.

II. Background

A. Regularized PET reconstruction

Let $\mathbf{x} \in [0, +\infty[^N$ be the spatial distribution of an injected radiotracer we aim to estimate from measurements counts $\mathbf{y} = (y_m)_{m=0}^{M-1} \in [0, +\infty[^M$ in M lines of responses. Each y_m correspond to a Poisson random variable.

The maximum likelihood solution is computed by minimizing the negative logarithm of the Poisson likelihood or the following data fidelity term f obtained by neglecting terms independent of $\mathbf{x} = (x_n)_{n=0}^{N-1}$:

$$f(\mathbf{x}; \mathbf{y}, \mathbf{b}) = \sum_{m=0}^{M-1} \mathcal{KL}(y_m, p_m) \tag{1}$$

where

$$\mathbf{p} = \mathbf{H}\mathbf{x} + \mathbf{b}, \tag{2}$$

$$\mathcal{KL}(y, p) = \begin{cases} -y \log(p) + p & \text{if } y > 0 \text{ and } p > 0 \\ p & \text{if } p \geqslant 0 \text{ and } y = 0 \\ +\infty & \text{else.} \end{cases} \tag{3}$$

where $\mathbf{H} \in \mathbb{R}^{M \times N}$ is the system matrix accounting for normalization, geometric projection, resolution effects and including attenuation of photons, $\mathbf{b} \in]0, +\infty[^M$ is the expectation of the background counts (scatter, randoms), and $\mathbf{p}$ is the model of the expectation of the measured counts.

The inclusion of prior information on the PET image can be achieved by computing a MAP estimate, reading as

$$\overline{\mathbf{x}} \in \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} \ \lambda f(\mathbf{x}; \mathbf{y}, \mathbf{b}) + r(\mathbf{x}) + \iota_{[0, +\infty[^N}(\mathbf{x}), \tag{4}$$

or equivalently

$$0 \in \lambda \nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b}) + \partial r(\overline{\mathbf{x}}) + \partial \iota_{[0, +\infty[^N}(\overline{\mathbf{x}}), \tag{5}$$

where r is the regularizer, $\iota_{[0, +\infty[^N}$ the indicator function over the convex set $[0, +\infty[^N$, and

$$\nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b}) = \mathbf{H}^\top 1 - \mathbf{H}^\top \frac{\mathbf{y}}{\mathbf{H}\overline{\mathbf{x}} + \mathbf{b}}. \tag{6}$$

For convex and non necessarily smooth r, the Douglas-Rachford/ADMM algorithm is an efficient way of solving (4).

B. Learning the regularization using neural networks

The choice of regularization operators is tedious yet crucial. In this work, we consider learning the regularization from a database of PET images while ensuring convergence of the reconstruction scheme. Despite this constraint, there are many ways of learning such operators.

The function r can be learned directly using input (weakly) convex neural networks trained as adversarial regularizers [32], [33]. Alternatively, the regularization can be learned indirectly by replacing ∇r with a neural network $\mathbf{D}_\Theta$. In [11], the authors propose a score-based approach for learning a network parameterized as a gradient of a nonconvex r and plugged into the gradient step of the proximal gradient algorithm.

A third and most popular choice is to learn a surrogate to prox_r in proximal splitting algorithms. For instance, [12] leverage results from [34] to build from the gradient of a convex C^1 function a proximity operator of a related nonconvex function. They provide convergence guarantees for a panel of proximal algorithms using such proximity operators. A limitation of their results is that the range of the regularization parameter is often constrained, or the image of the network is assumed to be convex, which is difficult to verify in practice. The convex-nonconvex approach of [35] provides another way of creating proximity operators of nonconvex regularizers. This framework considers regularizers that are structured enough to guarantee the convexity of the overall objective that appears when computing its proximity operator [35]. Works employing nonconvex regularizers prove convergence of the iterates to one critical point, as it is the best case in most nonconvex settings. Other extensions of the use of convex regularizers consist of replacing proximity operators with more general resolvents of (maximally) monotone operators. With such surrogates, the optimization problem can be analyzed as a monotone inclusion problem similar to (5) handled by many classical splitting algorithms. Provided constraints on the network and not on the hyperparameters, the algorithms converge to a fixed point that can be made unique.

Our preliminary work [47] did not demonstrate a significant difference between learning an explicit gradient of a

nonconvex function compared to learning the resolvent of a maximally monotone operator for our application. As we will show later, a PnP method with a learned resolvent can be made convergent to a unique fixed point whose characterization will be used for learning. Our method relies on the Douglas-Rachford (DR) splitting technique to address an inclusion similar to (5).

III. Method

A. Douglas-Rachford splitting

We first introduce the DR splitting in the context of monotone operators and rely on definitions and notation from [22]. DR aims at finding the zeros of a sum of two operators $\mathcal{A}$ and $\mathcal{B}$ that are respectively α- and β-maximally monotone ($\alpha, \beta \in [0, +\infty[$) on a closed subset C of $\mathbb{R}^N$. Hereinafter, we assume that there exists at least one zero $\overline{\mathbf{x}} \in C$ of $\mathcal{A}+\mathcal{B}$, which thus writes as

$$0 \in \mathcal{A}\overline{\mathbf{x}} + \mathcal{B}\overline{\mathbf{x}}. \tag{7}$$

Let T_{DR} be the underlying DR operator defined by

$$T_{\mathrm{DR}} = \frac{1}{2}\,\mathrm{Id} + \frac{1}{2} R_{\mathcal{A}} R_{\mathcal{B}},$$

where $R_{\mathcal{A}} = 2J_{\gamma\mathcal{A}} - \mathrm{Id}$ and $R_{\mathcal{B}} = 2J_{\gamma\mathcal{B}} - \mathrm{Id}$ are the reflections of $J_{\gamma\mathcal{A}}$ and $J_{\gamma\mathcal{B}}$ and $\gamma \in \mathbb{R}$.
When $\gamma \in \,]0, +\infty[$ $J_{\gamma\mathcal{A}}$ and $J_{\gamma\mathcal{B}}$ are firmly nonexpansive (FNE) on C, the composition $R_{\mathcal{A}}R_{\mathcal{B}}$ is 1-Lipschitz and thus T_{DR} is also FNE on C. It follows from the Krasnoselskii-Mann theorem that the DR sequence $(\mathbf{v}^n)_{n\in\mathbb{N}} \in C^N$ defined as

$$(\forall n \in \mathbb{N}) \quad (\mathbf{v}^{n+1}, \mathbf{x}^n) = (T_{\mathrm{DR}}(\mathbf{v}^n), J_{\gamma\mathcal{B}}(\mathbf{v}^n)), \tag{8}$$

converges to a fixed point $\overline{\mathbf{v}} \in \operatorname{Fix} T_{\mathrm{DR}}$ and $\overline{\mathbf{x}} = J_{\gamma\mathcal{B}}\overline{\mathbf{v}} \in \operatorname{Zer}(\mathcal{A}+\mathcal{B})$.

Remark 1 *A key property of the DR sequences $(\mathbf{v}_n)_{n\in\mathbb{N}}$ is their Fejér monotonicity with respect to* $\operatorname{Fix}(T_{\mathrm{DR}})$*:*

$$(\forall n \in \mathbb{N}) \qquad \begin{aligned}\|\mathbf{v}^{n+1} - \overline{\mathbf{v}}\|^2 &= \|T_{\mathrm{DR}}(\mathbf{v}^n) - T_{\mathrm{DR}}(\overline{\mathbf{v}})\|^2 \\ &\leqslant \|\mathbf{v}^n - \overline{\mathbf{v}}\|^2 \leqslant \|\mathbf{v}^0 - \overline{\mathbf{v}}\|^2.\end{aligned}$$

We have that $(\forall n \in \mathbb{N})$, $\mathbf{v}^n \in B_{\|\mathbf{v}^0 - \overline{\mathbf{v}}\|}(\overline{\mathbf{v}})$, where $B_\sigma(\overline{\mathbf{v}})$ is the N-dimensional closed ball of radius σ centered on $\overline{\mathbf{v}}$.

It is well known that DR is equivalent to ADMM when initialized with $(\mathbf{x}^0, \mathbf{u}^0) = (J_{\gamma\mathcal{B}}(\mathbf{v}^0), J_{\gamma\mathcal{B}}(\mathbf{v}^0) - \mathbf{v}^0)$ and $(\mathbf{v}^n)_{n\geqslant 1} = (\mathbf{z}^n - \mathbf{u}^{n-1})_{n\geqslant 1}$. In this case, Iteration (8) reads as

$$\begin{aligned}\mathbf{z}^{n+1} &= J_{\gamma\mathcal{A}}(\mathbf{x}^n + \mathbf{u}^n) \\ \mathbf{x}^{n+1} &= J_{\gamma\mathcal{B}}(\mathbf{z}^{n+1} - \mathbf{u}^n) \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \mathbf{x}^{n+1} - \mathbf{z}^{n+1}.\end{aligned} \tag{9}$$

B. Variational case

In Problem (7), when $\mathcal{A} = \lambda\partial f$ and $\mathcal{B} = \partial(r + \iota_{[0,+\infty[^N})$, $\overline{\mathbf{x}}$ solves (4) i.e. it is the minimizer of an explicit objective function.
Then, (9) simplifies as

$$\begin{aligned}\mathbf{z}^{n+1} &= \operatorname{prox}_{\gamma\lambda f}(\mathbf{x}^n + \mathbf{u}^n) \\ \mathbf{x}^{n+1} &= \operatorname{prox}_{\gamma r + \iota_{[0,+\infty[^N}}(\mathbf{z}^{n+1} - \mathbf{u}^{n+1}) \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \mathbf{x}^{n+1} - \mathbf{z}^{n+1}.\end{aligned} \tag{10}$$

Iteration (10) is the classical variational form of ADMM. PET reconstruction algorithms frequently handle the positivity constraint together with f by considering $\mathcal{A} = \partial(\lambda f + \iota_{[0,+\infty[^N})$ and $\mathcal{B} = \partial r$ to use multiplicative EM-based algorithms for computing $\operatorname{prox}_{\gamma\lambda f + \iota_{[0,+\infty[^N}}$. Still, other convergent algorithms can handle $\operatorname{prox}_{\gamma\lambda f}$ even with subsets acceleration [36], [48].

C. Plug-and-play case

Our PnP approach is an instance of the DR iteration; it consists in solving Problem (7) with a variational monotone operator $\mathcal{A} = \partial(\lambda f + \frac{\zeta}{2}\|\cdot - \mathbf{x}_{\mathrm{EM}}\|^2)$ ($\zeta \in [0, +\infty[$) and a learned maximal monotone operator $\mathcal{B}$ which is implicitly defined through its resolvent $J_{\mathcal{B}} = \mathbf{D}_\Theta$. $\mathbf{D}_\Theta$ is a differentiable neural network that embeds the positivity constraint. Altogether, with our choice of $\mathcal{A}$ and $\mathcal{B}$ and given that f is differentiable on $[0, +\infty[^N$, Problem (7) becomes

$$0 \in \lambda\nabla f(\overline{\mathbf{x}}) + \zeta(\overline{\mathbf{x}} - \mathbf{x}_{\mathrm{EM}}) + (\mathbf{D}_\Theta^{-1} - \mathrm{Id}\,)\overline{\mathbf{x}}, \tag{11}$$

for $\overline{\mathbf{x}} \in [0, +\infty[^N$.
Note that operator $\mathbf{D}_\Theta^{-1} - \mathrm{Id}$ is a properly defined set-valued operator that might not reduce to a singleton. When $\zeta = 0$ and $\mathbf{D}_\Theta = \operatorname{prox}_{r + \iota_{[0,+\infty[^N}}$, (11) reduces to (5); our PnP approach thus generalizes the previous variational case.
When $\gamma \in \,]0, +\infty[$, the DR sequence $(\mathbf{v}^n)_{n\in\mathbb{N}}$ converges to $\overline{\mathbf{v}} = \overline{\mathbf{x}} + \lambda\nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b}) + \zeta(\overline{\mathbf{x}} - \mathbf{x}_{\mathrm{EM}})$, $(\mathbf{z}^n - \mathbf{x}^n)_{n\in\mathbb{N}}$ to 0, $(\mathbf{x}^n)_{n\in\mathbb{N}}$ and $(\mathbf{z}^n)_{n\in\mathbb{N}}$ to

$$[\mathrm{FP}] \quad \overline{\mathbf{x}} = \mathbf{D}_\Theta(\overline{\mathbf{v}}) = \mathbf{D}_\Theta(\overline{\mathbf{x}} + \lambda\nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b}) + \zeta(\overline{\mathbf{x}} - \mathbf{x}_{\mathrm{EM}})).$$

As the function $f + \frac{\zeta}{2}\|\cdot - \mathbf{x}_{\mathrm{EM}}\|^2$ is ζ strongly convex, $\mathcal{A} + \mathcal{B}$ is ζ strongly monotone. For $\zeta > 0$, there exists a unique solution to (11). Since $\mathcal{B}$ is 0-monotone, $\mathbf{D}_\Theta$ is 0.5-averaged i.e. FNE. The learned resolvent only allows for using $\gamma = 1$ in DR. Indeed, given a maximally monotone operator $\mathcal{B}$ and $\gamma \in \,]0, +\infty[$, no simple formula is known to express $J_{\gamma\mathcal{B}}$ from $J_{\mathcal{B}}$.
The ADMM version of the PnP DR algorithm applied to (11) is given in Algorithm 1.

Remark 2 *In general, Problem* (11) *does not compute a MAP solution. When $\lambda = 0$, $(\mathbf{x}_n)_{n\in\mathbb{N}}$ converges to a fixed point of*

Algorithm 1 PnP ADMM iteration ($\mathbf{D}_\Theta$ FNE)

$$\begin{aligned}\mathbf{z}^{n+1} &= \operatorname{prox}_{\lambda f + \zeta\|\cdot - \mathbf{x}_{\mathrm{EM}}\|^2}(\mathbf{x}^n + \mathbf{u}^n) \\ \mathbf{x}^{n+1} &= \mathbf{D}_\Theta(\mathbf{z}^{n+1} - \mathbf{u}^n) \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \mathbf{x}^{n+1} - \tilde{\mathbf{z}}^{n+1}\end{aligned}$$

$\mathbf{D}_\Theta$ *if* $\zeta = 0$ *or the post-processing solution* $\mathbf{D}_\Theta(\mathbf{x}_{\mathrm{EM}})$ *if* $\zeta = 1$. *When* $\lambda \longrightarrow +\infty$, *the sequence* $(\mathbf{x}_n)_{n\in\mathbb{N}}$ *converges very slowly away from the maximum likelihood solution to* $\overline{\mathbf{x}}$. *Contrary to the MAP approach where* $\lambda \longrightarrow +\infty$ *leads to a recovery of the maximum likelihood solution for all* r, *the latter does not satisfy* (11) *in general, except if* $\mathbf{D}_\Theta = \mathrm{Id}$.

D. Training $\mathbf{D}_\theta$

a) Learning a fixed point mapping: Our goal is to reconstruct low-count data through Algorithm 1 to achieve the image quality of high-count reconstructions. We thus train the parameters of $\mathbf{D}_\Theta$ such that it satisfies [FP] for low-count sinograms $\mathbf{y}$ and scaled high-count EM images $\overline{\mathbf{x}} = \mathbf{x}_{\mathrm{HC}}$.

For training, we used the following loss

$$\ell(\mathbf{x}_{\mathrm{in}}, \mathbf{x}_{\mathrm{HC}}) = \frac{\|\mathbf{x}_{\mathrm{HC}} - \mathbf{D}_\Theta(\mathbf{x}_{\mathrm{in}}(\lambda_{\mathrm{Train}}, \mathbf{y}, \mathbf{x}_{\mathrm{HC}}))\|_2}{\|\mathbf{x}_{\mathrm{HC}}\|_2}, \quad (12)$$

where

$$\mathbf{x}_{\mathrm{in}} = \mathbf{x}_{\mathrm{HC}} + \lambda_{\mathrm{Train}} \nabla f(\mathbf{x}_{\mathrm{HC}}; \mathbf{y}, \mathbf{b}) + \zeta(\mathbf{x}_{\mathrm{HC}} - \mathbf{x}_{\mathrm{EM}}).$$

For a fixed subject, $\mathrm{Var}(\nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b}))$ is inversely proportional to the dose injected. We thus explicitly modeled this dependence by setting

$$\lambda_{\mathrm{Train}} = \alpha_{\mathrm{Train}} \times \sqrt{\|\mathbf{H}\overline{\mathbf{x}} + \mathbf{b}\|_1}$$

so that dose effects are mitigated. In addition to controlling the amplitude of $\nabla f(\overline{\mathbf{x}}; \mathbf{y})$, λ_{Train} also controls the weight of the likelihood compared to the prior in Problem (11). As the introduction suggests, reliance on hybrid deep learning/model-based iterative methods on the data likelihood is crucial for generalization. This pushes for using high λ_{Train} values. On the other hand, for large values of λ_{Train}, $\mathbf{x}_{\mathrm{in}}(\lambda_{\mathrm{Train}}, \mathbf{y}, \mathbf{x}_{\mathrm{HC}})$ is close to $\lambda_{\mathrm{Train}} \nabla f(\overline{\mathbf{x}}; \mathbf{y}, \mathbf{b})$ whose dynamic will be very different from $\mathbf{x}_{\mathrm{HC}}$, making training more challenging, and thus convergence of the training optimizer slower.

b) Constraining the neural network to ensure convergence: Enforcing the FNE property on $\mathbf{D}_\Theta$ can be done by design of the architecture (in that case, the constrained holds on the whole space $\mathbb{R}^N$) or enforced locally. Indeed Remark 1 shows that a local FNE property on $C = B_{\|\mathbf{v}_0 - \overline{\mathbf{v}}\|}(\overline{\mathbf{v}})$ is sufficient for convergence. The choice of $\mathbf{v}_0$ is important; the further it is from $\overline{\mathbf{v}}$, the larger the region where the constraint must hold. Here we choose $\mathbf{v}_0 = \mathbf{x}_{\mathrm{EM}}$.

Architectures averaged by design include feed-forward architectures using spectral normalization. A drawback is that this strategy precludes architectures with skip connections and is a conservative strategy that may over-contain the Lipschitz constant when the number of layers is high. Architectures FNE by design also include proximity operators of convex functions. Instead of handcrafting them, they can be learned as Deep Equilibrium networks (DEQ) [37], [38].

More generally, any architecture can be trained to be locally FNE by adding a regularization term to the training loss

$$\ell(\mathbf{x}_{\mathrm{in}}, \mathbf{x}_{\mathrm{HC}}) + \beta \max\{|||J_{2\mathbf{D}_\Theta - \mathrm{Id}}(\tilde{\mathbf{x}})||| + \epsilon - 1, 0\}^{1+\sigma}, \quad (13)$$

where $|||J_{2\mathbf{D}_\Theta - \mathrm{Id}}(\tilde{\mathbf{x}})|||$ is the spectral norm of the jacobian of $2\mathbf{D}_\Theta - \mathrm{Id}$ at $\tilde{\mathbf{x}} \in B_{\|\mathbf{x}_{\mathrm{EM}} - \mathbf{x}_{\mathrm{in}}\|}(\mathbf{x}_{\mathrm{in}})$, and hyperparameters $\epsilon \geqslant 0$ and $\sigma > 0$ are respectively a margin and a smoothness parameter. This Lipschitz regularization promotes the smoothness of $2\mathbf{D}_\Theta - \mathrm{Id}$ in a data-dependent and task-dependent way since the Jacobian is evaluated on a point derived from the fixed point mapping.

If CNNs are natural candidates for the architecture of $\mathbf{D}_\Theta$, alternatives exist, such as networks unfolding the iterations of an optimization algorithm. Deep unfolding models usually have fewer parameters than traditional CNNs and tend to generalize better, suggesting they could be smoother models. In [39], the authors reported that such networks are associated with a lower Lipschitz constant compared to a DRUnet, making them prime candidates for learning locally constrained architectures in a limited number of epochs.

IV. Experimental setup

Training a data-driven resolvent to map $\mathbf{x}_{\mathrm{in}}$ to $\mathbf{x}_{\mathrm{HC}}$ given our FNE constraint requires a database of triplets of high-count images, low-count images, and low-count data.

A. Simulation studies

The database used for learning and evaluation is generated synthetically from real brain [^{18}F]-FDG PET scans.

Fourteen 3D PET phantoms were constructed based on PET reconstructions from healthy subjects, along with their T1-weighted MRI images. The T1 images were segmented into 100 regions using FreeSurfer[2] and the corresponding PET values were then measured in a frame between 30 and 60 minutes after injection using PETSurfer to generate the 14 piecewise constant phantoms with anatomical and functional variability [30]. Eleven phantoms were used for training, and the others were used for evaluation.

Paired high-count and low-count sinograms (with a dose reduction of 5) were simulated using an analytical simulator [40] for a Biograph 6 TruePoint TrueV PET system. We simulated normalization and attenuation but also scatter and random effects and added Poisson noise to the sinograms. Resolution degradation effects, e.g., positron range and finite crystal width, were simulated in image space using a Gaussian point spread function with FWHM 4 mm. This was repeated for ten realizations per phantom for different averaged total number of prompts in the range observed in the original FDG exams, yielding high-count sinograms with between 135M and 430M prompts and low-count sinograms with between 27M and 86M prompts. The sinograms were reconstructed with CASToR [41] on a grid of size $128 \times 128 \times 109$ with a voxel size of 2.09 × 2.09 × 2.03 mm^3, with resolution modeling included in the reconstruction (OSEM with eight iterations of 14 subsets) to yield paired high-count and low-count reconstructions.

For evaluation of bias/variance trade-offs, we simulated 50 low-count replicates with the same averaged number of events. We also inserted three hyper-intense spherical lesions in one

[2]https://surfer.nmr.mgh.harvard.edu

	Architecture	FNE by design	# parameters	Approx. time/epoch	# epochs with jac	One operator call
$\mathbf{D}_\Theta$	U-net	No	1 079 000	2h	20	Fast
$\mathbf{D}_\Theta^{\mathrm{SN}}$	DnCNN with spectral normalization	Yes	167 620	1min	-	Fast
$\mathbf{D}_\Theta^{\mathrm{DU}}$	Unfolding	No	59 689	3h	3	$\propto N_{\mathrm{layers}} = 10$
$\mathbf{D}_\Theta^{\mathrm{DEQ}}$	Deep equilibrium	Yes	18 298	2h	-	$\propto N_{\mathrm{iter}} = 1000$

TABLE I: Summary of the types of architectures used in Algorithm 1.

test phantom (two with a diameter of 8 pixels and the last one of 4 pixels). The lesion activity concentration values were four times the mean value of the area where they were inserted. We simulated two doses (50 replicates each) for this phantom (one with 22M prompts and one with 46M prompts). These two simulations are used to assess the empirical generalization of the methods evaluated on out-of-distribution examples with unseen pathologies.

B. Architecture choice

We investigated four architectures for learning our data-driven resolvent ($\mathbf{D}_\Theta$, $\mathbf{D}_\Theta^{\mathrm{DU}}$, $\mathbf{D}_\Theta^{\mathrm{SN}}$, and $\mathbf{D}_\Theta^{\mathrm{DEQ}}$). The different architectures are summarized in Table I.

a) DRUnet: $\mathbf{D}_\Theta$ is a standard DRUnet [9] with three levels, a single residual block, 32 channels, ELU activations, 3D strided convolutions, upsampling layers, and without any normalization layers and biases. We added an outer ReLU activation enforcing positivity.

b) Unfolding network: Our second network $\mathbf{D}_\Theta^{\mathrm{DU}}$ unfolds 10 iterations of an unmatched version of the Combettes-Pesquet algorithm proposed in [42]. This algorithm has more degrees of freedom thanks to unmatched linear operators (the transposes of linear operators are replaced by surrogate operators) while keeping convergence and thus stability guarantees. The algorithm solves the optimization problem

$$\mathbf{x}_{\mathrm{in}} - \mathbf{x} \in \partial \iota_{[0,+\infty[^N}(\mathbf{x}) + \tilde{\mathbf{L}}^\top(\mathbf{L}\mathbf{x} - \mathrm{prox}_{\|\operatorname{Diag}(\Lambda_\Theta)\cdot\|_1}(\mathbf{L}\mathbf{x})),$$

where $\mathbf{L} \in \mathbb{R}^{NC\times N}$ and $\tilde{\mathbf{L}} \in \mathbb{R}^{NC\times N}$. For each iteration / layer n, we set $\mathbf{L}_n = \operatorname{Diag}(\Lambda_\theta(\mathbf{x}_{\mathrm{EM}}))\mathbf{C}_n$ and $\tilde{\mathbf{L}}_n = \operatorname{Diag}(\Lambda_\theta(\mathbf{x}_{\mathrm{EM}}))\tilde{\mathbf{C}}_n$. We learn parameters $\{(\tilde{\mathbf{C}}_n)_{n\in[1,10]}, (\mathbf{C}_n)_{n\in[1,10]}, \Lambda_\theta\}$ as well as the algorithm's step-sizes. $\mathbf{C}_n$ and $\tilde{\mathbf{C}}_n$ are convolution layers with kernel size 7 and $C = 16$ channels. $\Lambda_\theta : \mathbb{R}^N \mapsto ([0,1]^N)^C$ is chosen as in [43]: it is an RFDN with one input channel, 40 features, C output channels, and superresolution factor one with a final sigmoid. This promotes spatial adaptivity of the regularization, which has been found to boost the performance of data-driven regularizers [43]. Details on the architecture of a layer of $\mathbf{D}_\Theta^{\mathrm{DU}}$ can be found in the supplemental materials.

c) DnCNN with spectral normalization: $\mathbf{D}_\Theta^{\mathrm{SN}} = \frac{1}{2}\operatorname{Id} + \frac{1}{2}\mathbf{S}_\Theta$, where $\mathbf{S}_\Theta$ is a DnCNN with 8 nonexpansive layers, convolutions with a kernel size $3\times3\times3$ and 32 channels.

d) Deep equilibrium:

$$\mathbf{D}_\Theta^{\mathrm{DEQ}}(\mathbf{x}_{\mathrm{in}}) = \underset{\mathbf{x}\geqslant 0}{\operatorname{argmin}} \ \frac{1}{2}\|\mathbf{x}-\mathbf{x}_{\mathrm{in}}\|^2 + \|\operatorname{Diag}(\Lambda_\theta(\mathbf{x}_{\mathrm{EM}}))\mathbf{C}_\theta\mathbf{x}\|_1,$$

where Λ_θ has the same architecture as for $\mathbf{D}_\Theta^{\mathrm{DU}}$ and $\mathbf{C}_\theta$ is also a convolution layer with kernel size $7\times7\times7$ and $C = 16$ channels. $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ is thus the proximity operator of an explicit convex nonsmooth spatially adaptive regularization function.

Only $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ and $\mathbf{D}_\Theta^{\mathrm{SN}}$ are FNE by design. $\mathbf{D}_\Theta$ and $\mathbf{D}_\Theta^{\mathrm{DU}}$ were enforced to be locally FNE during training through the Lipschitz regularization (13).

C. Competing methods

1) MAP with fair regularization: The first competing method is a classical MAP reconstruction with the fair regularization on the difference between the first-order neighboring pixels [1]:

$$R_{\mathrm{fair}}(\mathbf{x}) = \sum_{i=0}^{N} \sigma\left(\frac{|[\mathbf{D}\mathbf{x}]_i|}{\sigma} - \log(1 + \frac{|[\mathbf{D}\mathbf{x}]_i|}{\sigma})\right).$$

The threshold σ is tuned as advocated in [44]. For the sake of simplicity, we used Algorithm 1, where we replaced an application of $\mathbf{D}_\Theta$ with $\mathrm{prox}_{R_{\mathrm{fair}}}$ computed iteratively using FISTA with warm restart and 100 iterations [45]. In this setup, several values for hyperparameter λ were tested ($\lambda \in \{15, 52, 89, 126, 163, 200\}$).

2) Post processing: The second competing method performs a maximum-likelihood reconstruction with early stopping (computed using OSEM with 16 iterations of 14 subsets) followed by a post-processing denoising operation by a DRUnet with the same architecture as above but trained to map low- to high-count PET images without any Lipschitz regularization.

3) PnP ADMM with Gaussian deep denoiser: The third method uses a Gaussian denoiser $\mathbf{D}_\Theta^{\mathrm{G}}$ in Algorithm 1 rather than our reconstruction-driven operators, as classically done in computer vision. $\mathbf{D}_\Theta^{\mathrm{G}}$ has the same architecture as $\mathbf{D}_\Theta$ and is trained with the Lipschitz regularization to remove white Gaussian noise with a standard deviation in $[0, 3500]$ on high-count targets.

D. Implementation details

All networks ($\mathbf{D}_\Theta$, $\mathbf{D}_\Theta^{\mathrm{DU}}$, $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ and $\mathbf{D}_\Theta^{\mathrm{SN}}$) were implemented using Pytorch 2.1 and trained on [FP] with $\zeta = 10^{-6}$ and $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$ (see supplemental materials for the validation of this range). The Adam algorithm was used as the optimizer with default parameters except for training $\mathbf{D}_\Theta^{\mathrm{SN}}$ for which reducing the momentum parameter β_1 to 0.5 was found

beneficial. The learning rate was set to 10^{-3} for all networks. In the Jacobian regularization (13), the spectral norm is computed iteratively with the power method and autodifferentiation is used to compute products with the Jacobian matrix. This leads to a potentially very large memory load. To overcome these issues, similar to [24], we turn off the tracking of the gradients in the power method except for the last iteration to compute the maximum eigenvalue. We also increase the number of iterations along epochs from 5 to 50. Furthermore, we first pre-trained the networks without regularization and then added the regularization for 20 extra epochs. We set $\beta = 0.01$, $\epsilon = 0.05$, $\sigma = 0.1$ so as to ensure that the FNE property is satisfied in our budget of epochs. Without Jacobian regularization, the batch size was set to 8; with Jacobian regularization, it was set to 4 to fit the VRAM capacity during learning.

Optimization of (12) with our DEQ operator requires solving an inner optimization problem. The forward pass is conducted using 1000 iterations of the Condat-Vũ algorithm [46], and the backward pass is computed using the Anderson algorithm (with a maximum of 1000 iterations, regularization parameter of 10^{-6} for the inner inversion).

The networks were evaluated in Algorithm 1 with $\lambda = \alpha\sqrt{\|\mathbf{y}\|_1}$ - serving as a proxy for $\|\mathbf{H}\overline{\mathbf{x}} + \mathbf{b}\|_1$ - and $\alpha \in \{50, 160, 275, 380, 500\}$. Computation of the proximity operator in Algorithm 1 is efficiently performed using standard algorithms for solving penalized PET reconstruction problems [31]. Algorithm 1 was run until $\|\mathbf{z}^{n+1} - \mathbf{x}^n\|/\|\mathbf{x}^n\|$ and $\|\mathbf{x}^{n+1} - \mathbf{x}^n\|/\|\mathbf{x}^n\|$ are below 5×10^{-4}.

E. Evaluation metrics

The reconstruction methods were assessed using a multiscale error analysis based on the Coiflet 3 wavelet over four levels of resolution. The levels were chosen to capture a broad range of spatial frequencies while minimizing boundary effects. For each replicate, we computed the standard deviation of multiscale error reflecting the variability of performance across replicates and the mean image multiscale error obtained by averaging the replicates for each method to highlight systematic biases and artifacts that persist across replicates.

Additionally, the voxel-wise metrics of mean, bias with respect to the scaled phantom and standard deviation are calculated across the multiple noise realization and on different anatomically relevant regions of interest (ROI). Our phantoms are flat, so as to avoid favoring methods that naturally produce smoother results, we also apply a Gaussian filter to all reconstructed images and adjust the FWHM to 3mm. Voxel-wise residual maps are displayed in the supplemental materials for a more qualitative inspection.

V. Results

A. Comparison of FNE architectures

We first compare different FNE architectures ($\mathbf{D}_\Theta$, $\mathbf{D}_\Theta^{\mathrm{DU}}$, $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ and $\mathbf{D}_\Theta^{\mathrm{SN}}$) all trained on [FP] with $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$ and evaluated in Algorithm 1 with $\lambda = \alpha\sqrt{\|\mathbf{y}\|_1}$ and $\alpha \in \{50, 160, 275, 380, 500\}$ also on the data from Phantom A.

Figure 1 shows the mean bias and standard deviation of the multiscale error across replicates and the multiscale error of the mean image for each FNE architecture. We see that $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ and $\mathbf{D}_\Theta^{\mathrm{DU}}$ achieved the lowest errors across all scales, indicating superior preservation of both coarse and fine image structures. However, at finer scales, $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ demonstrated higher STD between replicates compared to $\mathbf{D}_\Theta^{\mathrm{DU}}$. In contrast, $\mathbf{D}_\Theta$ exhibited the lowest STD and slightly higher error at finer scales, suggesting it struggles with preserving high-frequency details. $\mathbf{D}_\Theta^{\mathrm{SN}}$ exhibited higher mean error, especially at coarser scales, suggesting a trade-off between preserving global structure and details in the image. Moreover, it was found to be the most sensitive to the value λ as shown by the span of the values in figure 1.

Figure 2 shows slices of the reconstructed images for one noisy replicate without Gaussian smoothing at a similar bias over the thalamus. The fact that the extracerebral area, as well as the white matter, are flatter with $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ is because $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ is the proximity operator of an ℓ_1 norm composed with a learned operator. Even though it does not compute a proximity operator, $\mathbf{D}_\Theta^{\mathrm{DU}}$ model seems to inherit some of the properties of the original ℓ_1-based minimization algorithm it unfolds because it also provides flatter reconstructed images. Both $\mathbf{D}_\Theta^{\mathrm{SN}}$ and $\mathbf{D}_\Theta$ slightly oversmooth the frontal area; with $\mathbf{D}_\Theta^{\mathrm{DU}}$, it most resembles the high-count reference. $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ appears as an intermediary between the low-count EM reconstruction and the high-count reference.

Both quantitatively and qualitatively, $\mathbf{D}_\Theta^{\mathrm{SN}}$ was demonstrated to underperform the other FNE networks.

B. ROI-based comparison with state-of-the-art methods

Second, we compare our PnP approach using our architectures - except for $\mathbf{D}_\Theta^{\mathrm{SN}}$ - with MBIR with fair regularization, post-reconstruction processing (PP), and PnP with a classical Gaussian denoiser ($\mathbf{D}_\Theta^{\mathrm{G}}$) on the data simulated from Phantom A.

In figure 3, additional multiscale errors and STD metrics are reported. Notably, we see that $\mathbf{D}_\Theta^{\mathrm{G}}$ yields higher bias compared to the other PnP approaches and that MBIR exhibits a higher STD, especially at finer scales, indicating it is either prone to keep noise or it introduced artifacts. The post-processing approach provides slightly degraded trade-offs compared to the PnP reconstruction with $\mathbf{D}_\Theta$.

To further evaluate the methods, we provide bias versus standard deviation plots for different ROIs (the whole brain, caudate, cerebellum, thalamus and the frontal area) in figure 4 with all previous methods except for $\mathbf{D}_\Theta^{\mathrm{SN}}$. For all regions, the use of the Gaussian denoiser $\mathbf{D}_\Theta^{\mathrm{G}}$ provides the highest bias values at matched STD compared to using other networks in Algorithm 1. The post-processing approach systematically yields the lowest STD but a higher bias on all ROIs. Our PnP approach provided a lower STD than MBIR at matched bias values on all metrics, although the improvement is less apparent on the frontal area for $\mathbf{D}_\Theta^{\mathrm{DEQ}}$. It also achieved biases similar to those obtained with the high-count reference for a relatively higher STD. Regarding the differences between our three PnP methods, we see that $\mathbf{D}_\Theta^{\mathrm{DEQ}}$ achieves the best tradeoffs relative to the other architectures over the whold

brain, the white matter and the cerebellum, closely followed by $\mathbf{D}_{\Theta}^{\mathrm{DU}}$ and then $\mathbf{D}_{\Theta}$. On all regions, $\mathbf{D}_{\Theta}$ provided lower STD values than $\mathbf{D}_{\Theta}^{\mathrm{DEQ}}$ and $\mathbf{D}_{\Theta}^{\mathrm{DU}}$ but increased bias, especially in the frontal area.

Reconstructed images are shown in Figure 5 for another slice and noise replicate than Figure 2. The MBIR reconstruction appears patchy while $\mathbf{D}_{\Theta}^{\mathrm{DEQ}}$ provided a more structure-dependent smoothing but preserved some structured noise also present in the original EM low-count solution. Reconstructions with $\mathbf{D}_{\Theta}$ and $\mathbf{D}_{\Theta}^{\mathrm{DU}}$ are difficult to distinguish over hot regions. Overall, post-processing provides the smoothest solution. Moreover, we see that the image obtained with $\mathbf{D}_{\Theta}^{\mathrm{G}}$ looks very similar to the post-processing image except in the extracerebral area, thus highlighting the over-regularization effect of this network, especially in hot regions.

C. Evaluation on two out-of-distribution test cases

Third, we evaluate the previous methods on the two synthetic test cases that differ from the training set. In the first scenario, we introduced hyper-intense lesions while keeping the same values for the different ROIs as before. In the second scenario, we simulated an extra dose reduction by a factor of two, including for hyper-intense lesions. Improved reconstruction methods should, therefore, lead to a fixed tumor mean relative to unsmoothed EM at high dose and reduced white matter standard deviation relative to unsmoothed EM at low dose.

Figure 7 shows plots of tumor mean across replicates versus mean STD in the tumors for the two doses. The figure shows close behavior between the MBIR and $\mathbf{D}_{\Theta}^{\mathrm{DEQ}}$ for both doses. For the two largest tumors, $\mathbf{D}_{\Theta}^{\mathrm{DEQ}}$ is able to achieve the flattest curve: the activity of tumors 2 and 3 remains very close to the value obtained with EM while the noise over the tumor is reduced. $\mathbf{D}_{\Theta}^{\mathrm{DU}}$ better preserves the smallest tumor. Meanwhile, the post-processing solution and PnP reconstruction with $\mathbf{D}_{\Theta}$ reduce the tumors' STD more than the others but underestimate the tumors' activity the most, especially for the smallest tumor for the post-processing method.

VI. Discussion

Our results first pinpoint that one can design a convergent PnP method that uses deep learning models trained to enforce the fixed-point condition of the reconstruction algorithm on target images. By doing so, we depart from standard PnP methods that consider separately the learning of the prior from the reconstruction of the image. Precisely, we showed that, in PnP ADMM, such models lead to better reconstructions than using a Gaussian denoiser, which struggled to preserve edges in the images. This is likely due to a mismatch between the training denoising task and input images of the network across the reconstruction iterations that should ultimately satisfy the fixed-point conditions. This underscores the importance of training networks on data representative of the reconstruction task, especially in contexts with little training data. As a consequence, our method is inherently designed for a specific acquisition model and scanner type. Changes in scanner and acquisition protocols may alter the statistical properties of the data, requiring retraining of the models to maintain optimal performance—just as is the case with deep unfolding reconstruction methods.

Enforcing the convergence conditions was performed either through training or by design. Notably, the use of spectral normalization did not meet expectations. Although we limited the number of stacked layers to avoid excessive downscaling of the Lipschitz constant, the approach still fell short in terms of performance. The likely cause is the difficulty of keeping the Lipschitz constant sufficiently close to 1 for the network while increasing the number of layers to obtain an efficient regularization. On the contrary, we have shown that PnP ADMM with locally regularized FNE models properly regularized during learning (based on the Fejér monotonicity of the DR sequences) converge and lead to improved results compared to spectral normalization. Yet, the Jacobian regularization needed for training was computationally heavy, often competing with the scaled similarity loss. Although the unfolding architecture resembles a fixed-point network operator, it did not eliminate the need for the Jacobian regularization. However, one should note that after pre-training, the DRUnet was associated with a spectral norm of about 10 on the training set, while the unfolding network was associated with a spectral norm of around 1.6. Given that the chosen Jacobian regularization strongly penalizes large spectral norms, the scaled similarity loss was more strongly degraded in the first epochs of training with the DRUnet compared to the unfolding model, leading to different dynamics between denoising and Jacobian regularization across epochs for the two types of architectures. The relationship between optimization and Jacobian regularization is a key area that warrants further investigation.

Despite being FNE by design, our DEQ proved to be at least as competitive in terms of performance as the other networks. Such an approach highlighted that global FNE property does not necessarily hurt performance. Furthermore, thanks to a spatially-dependent regularization strength, the network or learned proximity operator still maintains genuine high-intensity features (tumor regions) present and enables smoothing (reducing noise) across uniform areas. The images produced using the DEQ are sharper compared to the deep unfolding model and, above all, the DRUnet. It however sometimes retained high-intensity noisy structures from the low-count data. Finally, it generalized well on lower dose and real data.

Compared to post-reconstruction processing, the PnP reconstructions are less smoothed and are associated with a reduced bias. This highlights the value of embedding the network into iterative reconstruction rather than applying it after reconstruction, as already noted in [44]. Overall, the PnP methods consistently outperformed the traditional MBIR approach, and with a DEQ or an unfolded architecture, they succeeded in matching MBIR's performance on lesion recovery. The relatively small size of our training set may have influenced how networks handled smoothing. This may explain why more constrained networks like the unfolding network and the DEQ, which incorporate prior knowledge into their architecture, performed better on out-of-distribution test cases.

Future efforts should focus on designing effective FNE archi-

tectures. Architectures FNE by design are expected to best generalize and be more robust to the diversity of input images encountered by the network during optimization. DEQ is a promising direction and regularized unfolding architecture will perform better than correcting pre-trained architectures by incorporating pre- or post-smoothing steps or combining a network with a known Lipschitz constant and the identity operator. We still acknowledge that applying a DEQ is not as fast as applying an unfolding network or a DRUnet since it requires solving a minimization problem. Yet, for PET reconstruction, the cost of the reconstruction is dominated by the application of the proximity operator of the data fidelity (more precisely, application of the projection and backprojection), so this does not hinder the method's practicality.

VII. Conclusion

In this paper, we introduced a PnP method for PET reconstruction with convergence guarantees. The method leverages a data-driven regularization that was learned using fixed point conditions involving low-count images and data, as well as high-count images. Although the type of regularization was constrained by the limited size of our training data, the method showed competitive quantification metrics against several baselines - from MBIR to deep learning methods - even in out-of-distribution and real-world cases, establishing it as a reliable candidate for low-dose PET reconstruction where robustness to real-world degradation is important. While our work only touched upon the interaction between network architecture, optimization landscapes, and Jacobian regularization, we view these interactions as promising avenues for future research. Further work will explore alternative architectures incorporating anatomical information and extend evaluations to larger clinical datasets.

References

[1] K. Lange. Convergence of EM image reconstruction algorithms with Gibbs smoothing. *IEEE Transactions on Medical Imaging*, 9(4):439–446, December 1990.

[2] A. J. Reader et al. Deep Learning for PET Image Reconstruction. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 5(1):1–25, January 2021.

[3] T. Meinhardt et al. Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1799–1808, Venice, October 2017.

[4] J. Xu and F. Noo. Convex optimization algorithms in medical image reconstruction – in the age of AI. *Physics in Medicine & Biology*, 67(7), March 2022.

[5] G. Corda-D'Incan et al. Memory-Efficient Training for Fully Unrolled Deep Learned PET Image Reconstruction with Iteration-Dependent Targets. *IEEE transactions on radiation and plasma medical sciences*, 6(5):552–563, May 2022.

[6] Z. Ramzi et al. NC-PDNet: A Density-Compensated Unrolled Network for 2D and 3D Non-Cartesian MRI Reconstruction. *IEEE Transactions on Medical Imaging*, 41(7):1625–1638, July 2022.

[7] M. Savanier et al. Deep Unfolding of the DBFB Algorithm With Application to ROI CT Imaging With Limited Angular Density. *IEEE Transactions on Computational Imaging*, 9:502–516, 2023.

[8] S. V. Venkatakrishnan et al. Plug-and-Play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948, December 2013.

[9] K. Zhang et al. Plug-and-Play Image Restoration With Deep Denoiser Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6360–6376, October 2022.

[10] R. Cohen et al. Regularization by Denoising via Fixed-Point Projection (RED-PRO), October 2020. arXiv:2008.00226.

[11] S. Hurault et al. Gradient Step Denoiser for convergent Plug-and-Play, February 2022.

[12] S. Hurault et al. Proximal denoiser for convergent plug-and-play optimization with nonconvex regularization. In *International Conference on Machine Learning*, pp. 9483–9505. PMLR, 2022.

[13] R. Cohen et al. It Has Potential: Gradient-Driven Denoisers for Convergent Solutions to Inverse Problems. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18152–18164, 2021.

[14] Z. Wu et al. Extrapolated plug-and-play three-operator splitting methods for nonconvex optimization with applications to image restoration. *SIAM Journal on Imaging Sciences*, 17(2):1145–1181, 2024.

[15] J.-C. Pesquet et al. Learning Maximally Monotone Operators for Image Recovery. *SIAM Journal on Imaging Sciences*, 14(3):1206–1237, January 2021.

[16] Y. Sun et al. Scalable Plug-and-Play ADMM with Convergence Guarantees, January 2021. arXiv:2006.03224.

[17] Y. Suzuki et al. Convergent Primal-Dual Plug-and-Play Image Restoration: A General Algorithm and Applications, January 2025. arXiv:2501.03780.

[18] C. Y. Park et al. Plug-and-Play Priors as a Score-Based Method, December 2024. arXiv:2412.11108.

[19] S. Hurault et al. Convergent Bregman Plug-and-Play Image Restoration for Poisson Inverse Problems, June 2023.

[20] Y. Song and S. Ermon. Improved Techniques for Training Score-Based Generative Models, October 2020. arXiv:2006.09011.

[21] R. Fermanian et al. PnP-ReG: Learned Regularizing Gradient for Plug-and-Play Gradient Descent. *SIAM Journal on Imaging Sciences*, 16(2):585–613, June 2023.

[22] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer International Publishing, 2017.

[23] A. Pramanik et al. Memory-Efficient Model-Based Deep Learning With Convergence and Robustness Guarantees. *IEEE Transactions on Computational Imaging*, 9:260–275, 2023.

[24] Y. Belkouchi et al. Learning truly monotone operators with applications to nonlinear inverse problems, March 2024. arXiv:2404.00390.

[25] E. K. Ryu et al. Plug-and-Play Methods Provably Converge with Properly Trained Denoisers, May 2019. arXiv:1905.05406.

[26] J. Liu et al. RARE: Image Reconstruction using Deep Priors Learned without Ground Truth. *IEEE Journal of Selected Topics in Signal Processing*, 14(6):1088–1099, October 2020.

[27] J. Li et al. Plug-and-Play ADMM for MRI Reconstruction With Convex Nonconvex Sparse Regularization. *IEEE Access*, 9:148315–148324, 2021.

[28] A. Rasti-Meymandi et al. Plug and play augmented HQS: Convergence analysis and its application in MRI reconstruction. *Neurocomputing*, 518:1–14, January 2023.

[29] K. Wei et al. TFPnP: Tuning-free Plug-and-Play Proximal Algorithm with Applications to Inverse Imaging Problems. *J. Mach. Learn. Res.*, November 2020.

[30] F. Sureau et al. Convergent ADMM Plug and Play PET Image Reconstruction. In *Proceedings of the 17th International Meeting on Fully 3D Image Reconstruction in Radiology and Nuclear Medicine*, 2023.

[31] A. R. De Pierro. A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography. *IEEE transactions on medical imaging*, 14(1):132–137, 1995.

[32] S. Mukherjee et al. Data-Driven Convex Regularizers for Inverse Problems. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 13386–13390, April 2024. ISSN: 2379-190X.

[33] Z. Shumaylov et al. Provably Convergent Data-Driven Convex-Nonconvex Regularization, November 2023. arXiv:2310.05812.

[34] R. Gribonval and M. Nikolova. A Characterization of Proximity Operators. *Journal of Mathematical Imaging and Vision*, 62(6):773–789, July 2020.

[35] A. Goujon et al. Learning weakly convex regularizers for convergent image-reconstruction algorithms. *SIAM Journal on Imaging Sciences*, 17(1):91–115, 2024.

[36] F. Abboud et al. Dual Block-Coordinate Forward-Backward Algorithm with Application to Deconvolution and Deinterlacing of Video Sequences. *Journal of Mathematical Imaging and Vision*, 59(3):415–431, November 2017.

[37] E. Winston and J. Z. Kolter. Monotone operator equilibrium networks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pp. 10718–10728, Red Hook, NY, USA, December 2020. Curran Associates Inc.

[38] Z. Zou et al. Deep Equilibrium Learning of Explicit Regularization Functionals for Imaging Inverse Problems. *IEEE Open Journal of Signal Processing*, 4:390–398, 2023.

[39] H. T. V. Le et al. PNN: From proximal algorithms to robust unfolded image denoising networks and Plug-and-Play methods, August 2023. arXiv:2308.03139.

[40] S. Stute et al. Analytical simulations of dynamic PET scans with realistic count rates properties. In *2015 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pp. 1–3, October 2015.

[41] T. Merlin et al. CASToR: a generic data organization and processing code framework for multi-modal and multi-dimensional tomographic reconstruction. *Physics in Medicine & Biology*, 63(18):185005, September 2018.

[42] E. Chouzenoux et al. Convergence Results for Primal-Dual Algorithms in the Presence of Adjoint Mismatch. *SIAM Journal on Imaging Sciences*, January 2023. Publisher: Society for Industrial and Applied Mathematics University City, Philadelphia.

[43] S. Neumayer et al. Boosting weakly convex ridge regularizers with spatial adaptivity. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*, 2023.

[44] K. Gong et al. Iterative PET Image Reconstruction Using Convolutional Neural Network Representation. *IEEE Transactions on Medical Imaging*, 38(3):675–685, March 2019.

[45] A. Chambolle and C. Dossal. On the Convergence of the Iterates of the "Fast Iterative Shrinkage/Thresholding Algorithm". *Journal of Optimization Theory and Applications*, 166(3):968–982, September 2015.

[46] L. Condat. A Primal–Dual Splitting Method for Convex Optimization Involving Lipschitzian, Proximable and Linear Composite Terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, August 2013.

[47] M. Savanier et al. Learning with fixed point condition for convergent PnP PET reconstruction. In *ISBI 2024 - 21st IEEE International Symposium on Biomedical Imaging*, Athenes, Greece, May 2024.

[48] M. J. Ehrhardt et al. Faster PET reconstruction with non-smooth priors by randomization and preconditioning. *Physics in Medicine & Biology*, 64(22):225019, November 2019

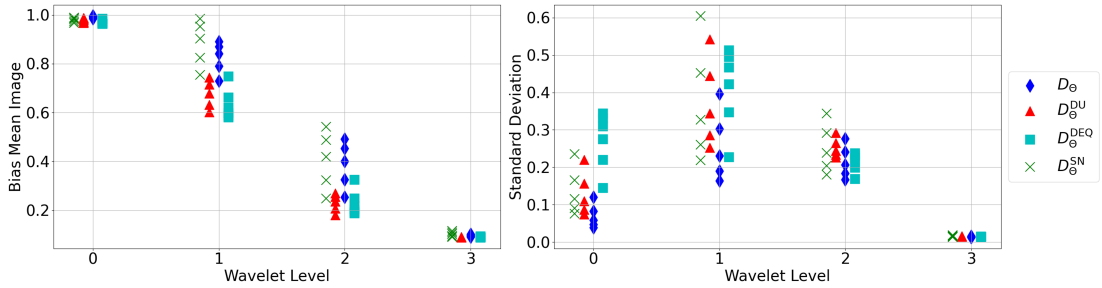


Fig. 1: Multiscale analysis of Algorithm 1 on Phantom A with different values of λ and four FNE architectures trained for $\lambda_{\text{Train}} = \alpha_{\text{Train}} \times \sqrt{\|\mathbf{H}\bar{\mathbf{x}} + \mathbf{b}\|_1}$ where $\alpha_{\text{Train}} \sim \mathcal{U}(50, 500)$.

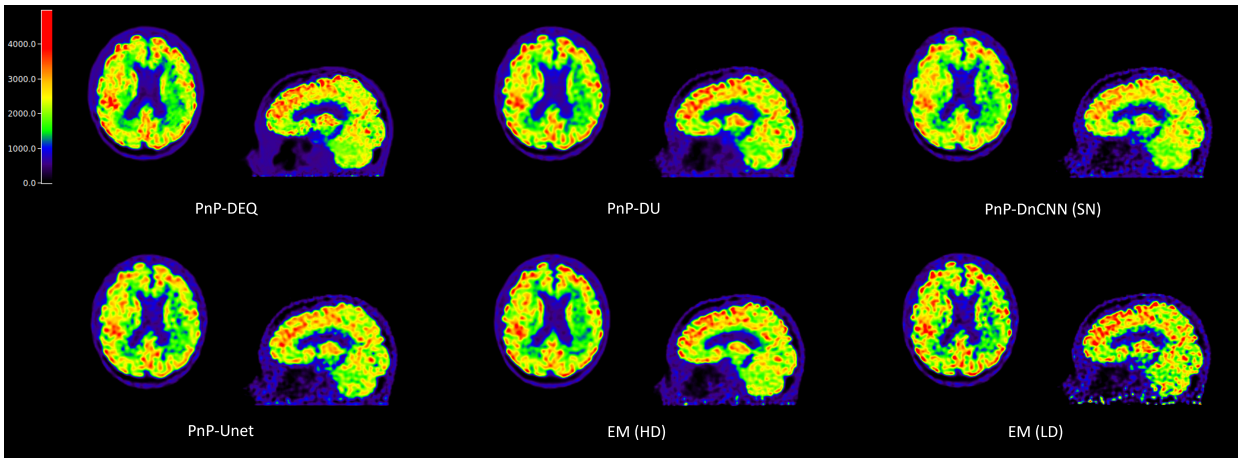


Fig. 2: Axial and sagittal slices of reconstructed images using the four FNE architectures for a first data realization from Phantom A with optimal λ.

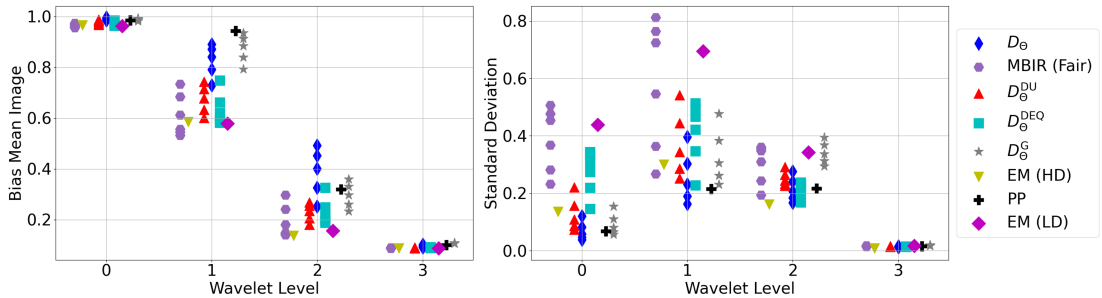


Fig. 3: Multiscale analysis of Algorithm 1 on Phantom A with different regularization parameter values for our PnP methods and state-of-the-art reconstruction methods

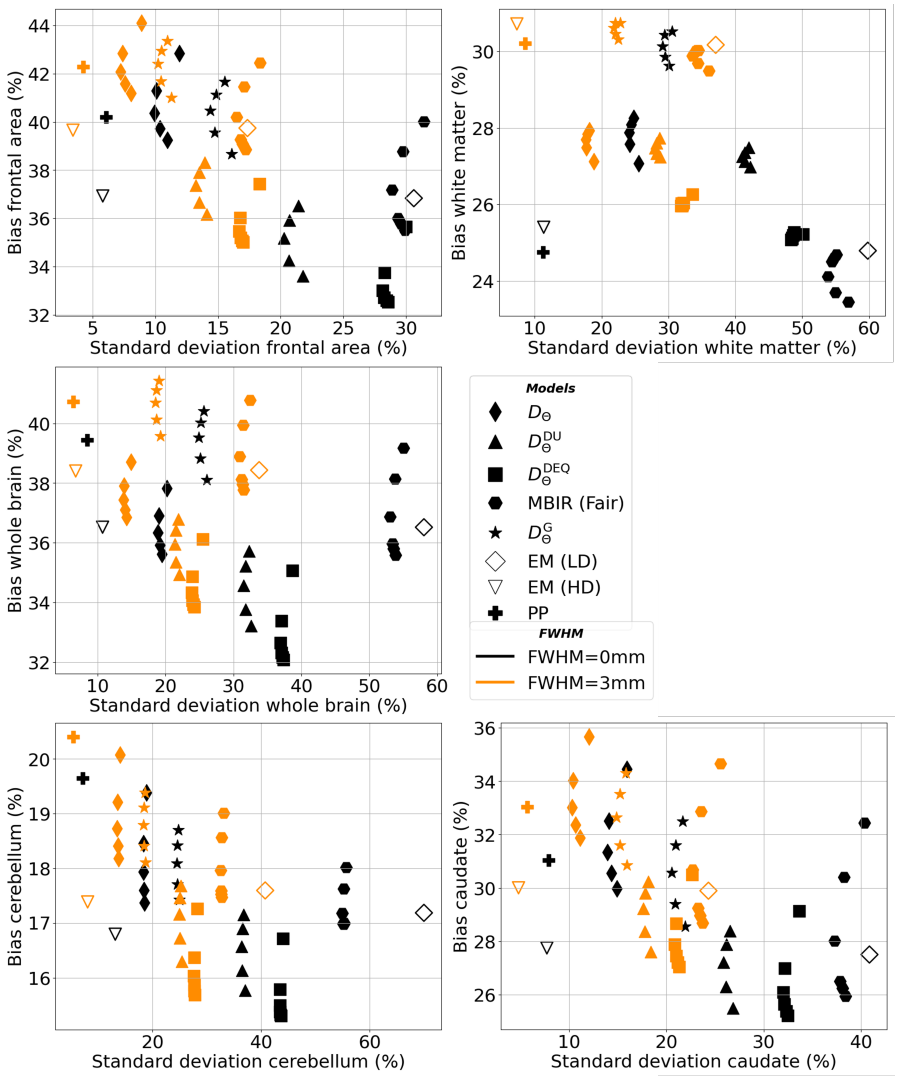


Fig. 4: Comparison of Algorithm 1 applied on data from Phantom A with learned resolvent trained on [FP] for $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$ or as a Gaussian denoiser with MBIR and post-EM-reconstruction processing (PP). Bias vs standard deviation in several ROIs. The mean and standard deviation values were calculated within the specified ROIs and then averaged across the multiple noise realizations for different values of the regularization parameter.

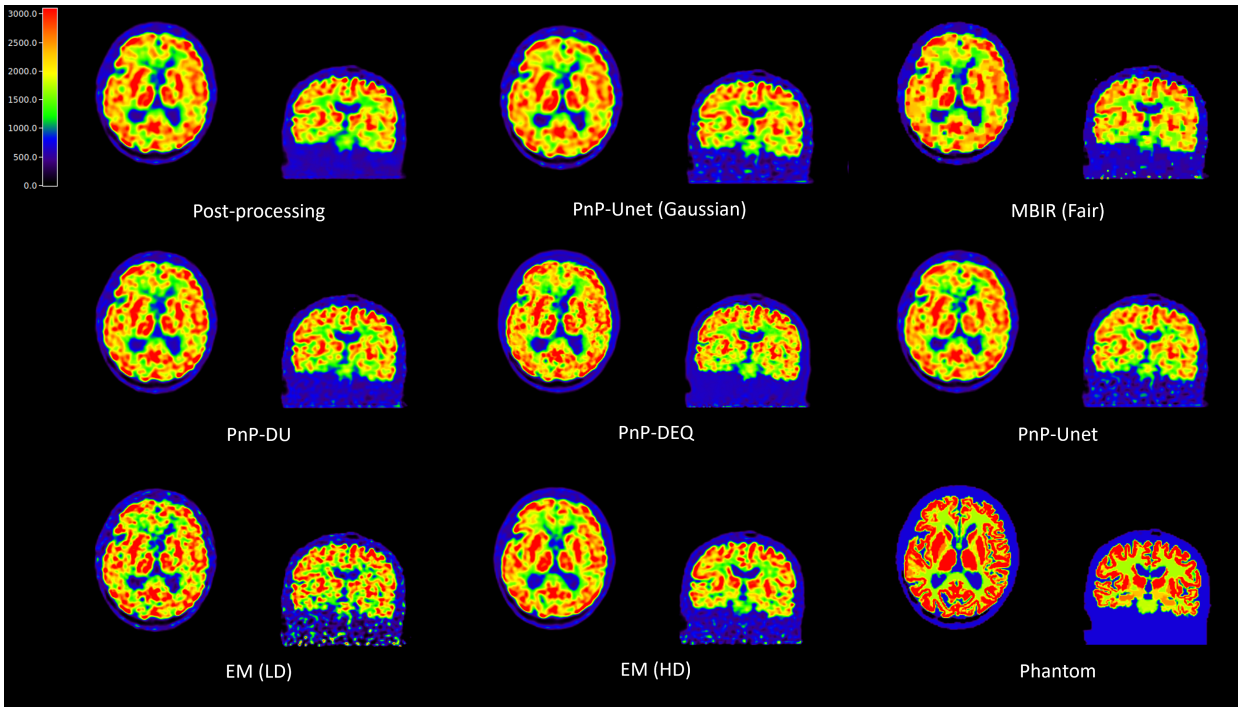


Fig. 5: Axial and coronal slices of reconstructed images for a second data realization obtained from Phantom A with optimal λ.

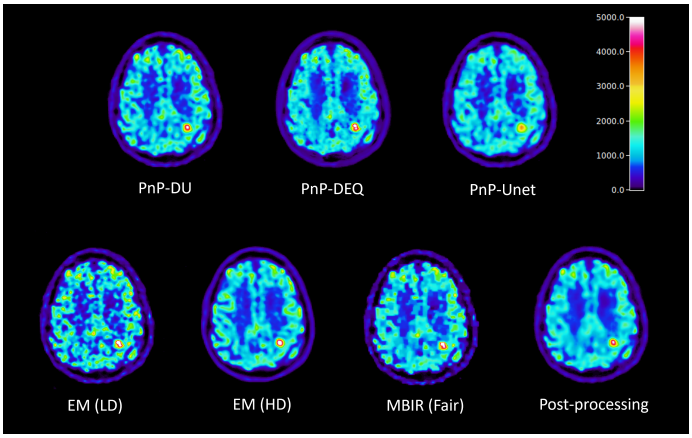
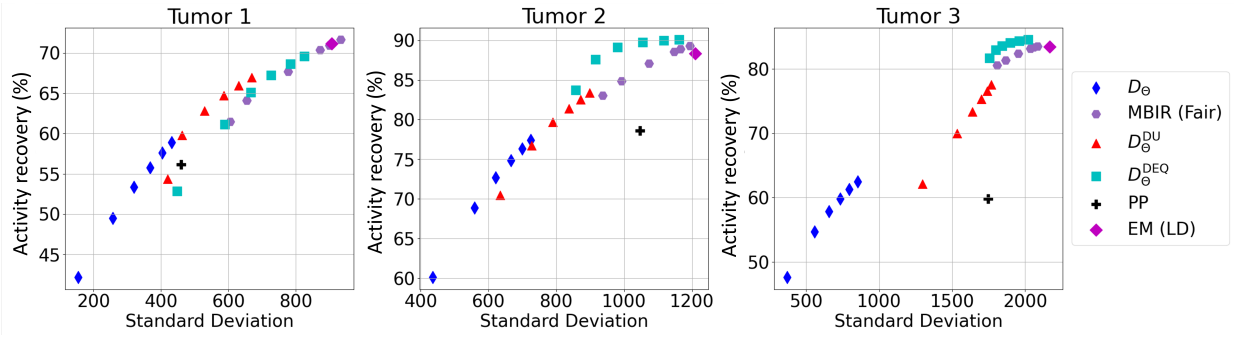
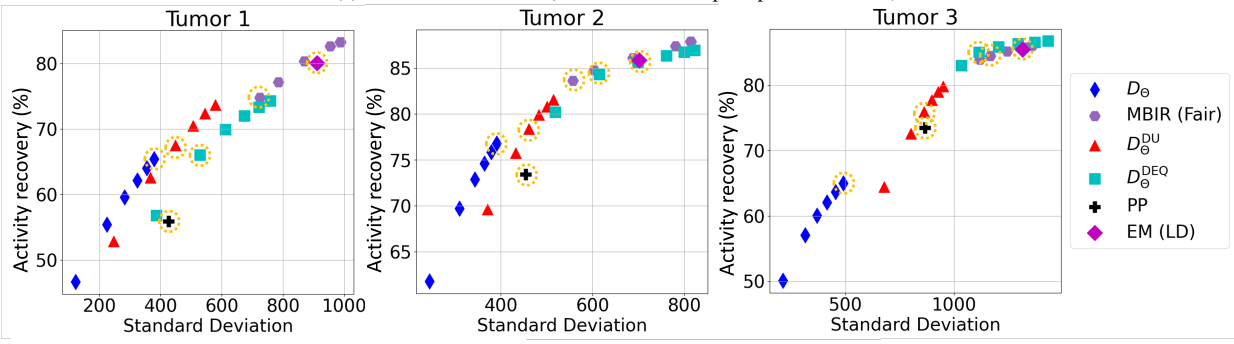


Fig. 6: Axial slices of the reconstructions of one replicate (dose reduction of 10, Phantom B, Tumor 3) with regularization parameters indicated by yellow circles in fig. 7.



(a) Dose reduction of 5 (Phantom B, 46M prompt coincidences)



(b) Dose reduction of 10 (Phantom B, 22M prompt coincidences)

Fig. 7: Mean tumor activity versus tumor standard deviation for two-dose reduction factors, and all reconstruction methods. Tumor 1 has a diameter of 4 pixels, while tumor 2 and tumor 3 have a diameter of 8 pixels.

Fixed point method for PET reconstruction with learned plug-and-play regularization - Supplemental Materials

I. Implementation details on $\mathbf{D}_{\Theta}^{\mathrm{DU}}$

Network $\mathbf{D}_{\Theta}^{\mathrm{DU}}$ unfolds ten iterations of the unmatched Combettes-Pesquet algorithm; it reads as

$$\mathbf{D}_{\Theta}^{\mathrm{DU}} = \mathcal{L}_{\Theta_{10}}^{\mathrm{UCP}} \circ \cdots \circ \mathcal{L}_{\Theta_1}^{\mathrm{UCP}}$$

where

$$\begin{aligned}
&\mathcal{L}_{\Theta_n}^{\mathrm{UCP}}(\cdot, \mathbf{x}_{\mathrm{in}}) \\
&\quad \Theta_n = \{\mathbf{L}_n, \tilde{\mathbf{L}}_n, \rho_n\} \\
&\quad {}^{\varepsilon}\vartheta_1 = 1 + \max\{|||\mathbf{L}_n|||, |||\tilde{\mathbf{L}}_n|||\} \\
&\quad {}^{\varepsilon}\vartheta_2 = \sqrt{1 + |||\mathbf{L}_n|||^2 + |||\tilde{\mathbf{L}}_n|||^2} \\
&\quad {}^{\varepsilon}\vartheta = \min\{{}^{\varepsilon}\vartheta_1, {}^{\varepsilon}\vartheta_2\} \\
&\quad \gamma = 0.99/{}^{\varepsilon}\vartheta \\
&\quad \mathbf{v}_{1,n} = \mathbf{x}_n - \gamma(\mathbf{x}_n - \mathbf{z} + \tilde{\mathbf{L}}_n \mathbf{u}_n) \\
&\quad \mathbf{p}_{1,n} = P_C(\mathbf{v}_{1,n}) \\
&\quad \mathbf{v}_{2,n} = \mathbf{u}_n + \gamma(\mathbf{L}_n \mathbf{x}_n + \mathbf{u}_n) \\
&\quad \mathbf{p}_{2,n} = \mathrm{prox}_{\gamma(\|\,\mathrm{Diag}(\Lambda_\theta)\cdot\|_1)^*}(\mathbf{v}_{2,n}) \\
&\quad \mathbf{q}_{2,n} = \mathbf{p}_{2,n} + \gamma(\mathbf{L}_n \mathbf{p}_{1,n} + \mathbf{p}_{2,n}) \\
&\quad \mathbf{q}_{1,n} = \mathbf{p}_{1,n} - \gamma(\mathbf{p}_{1,n} - \mathbf{z} + \tilde{\mathbf{L}}_n \mathbf{p}_{2,n}) \\
&\quad \mathbf{x}_{n+1} = \rho_n(\mathbf{x}_n - \mathbf{v}_{1,n} + \mathbf{q}_{1,n}) + (1 - \rho_n)\mathbf{x}_n \\
&\quad \mathbf{u}_{n+1} = \rho_n(\mathbf{u}_n - \mathbf{v}_{2,n} + \mathbf{q}_{2,n}) + (1 - \rho_n)\mathbf{u}_n
\end{aligned} \tag{1}$$

Note that $|||\mathbf{L}|||$ denotes the spectral norm of the linear operator $\mathbf{L}$.
We set $\mathbf{x}_0$ and $\mathbf{u}_0$ to zero. The learned parameters are $\Theta = \{(\rho_n)_{n\in[1,10]}, (\tilde{\mathbf{C}}_n)_{n\in[1,10]}, (\mathbf{C}_n)_{n\in[1,10]}, \Lambda_\theta\}$.

II. Evaluation metrics

For each replicate, we computed the error between the wavelet coefficients of the reconstructed replicate $\mathbf{x}_r^\lambda$ and the scaled phantom $\mathbf{x}^{\mathrm{Ph}}$ at each scale: for a wavelet decomposition across L levels with approximation coefficient, $A_L(\mathbf{x})$, and detail coefficients at scale $l < L$ for sub-band $b \in B$, $D_l^{(b)}(\mathbf{x})$,

$$\begin{aligned}
\overline{\mathbf{x}}^\lambda &= \frac{1}{N_{\mathrm{real}}} \sum_{r=1}^{N_{\mathrm{real}}} \mathbf{x}_r^\lambda \\
\mathrm{err}^{L,\lambda}(\overline{\mathbf{x}}^\lambda, \mathbf{x}_r^\lambda) &= \frac{\|A_L(\overline{\mathbf{x}}^\lambda) - A_L(\mathbf{x}_r^\lambda)\|_2}{\|A_L(\mathbf{x}^{\mathrm{Ph}})\|_2} \\
\mathrm{err}^{\mathrm{detail},l,b,\lambda}(\overline{\mathbf{x}}^\lambda, \mathbf{x}_r^\lambda) &= \frac{\|D_l^b(\overline{\mathbf{x}}^\lambda) - D_l^b(\mathbf{x}_r^\lambda)\|_2}{\|D_l^b(\mathbf{x}^{\mathrm{Ph}})\|_2} \\
\mathrm{err}^{l,\lambda}(\overline{\mathbf{x}}^\lambda, \mathbf{x}_r^\lambda) &= \frac{1}{|B|} \sum_{b\in B} \mathrm{err}^{\mathrm{detail},l,b,\lambda}(\overline{\mathbf{x}}^\lambda).
\end{aligned}$$

Errors were aggregated into the standard deviation of multiscale error $\mathrm{std}^{l,\lambda}$, for l in $[0, L]$,

$$\mathrm{std}^{l,\lambda} = \frac{\sum_{r=1}^{N_{\mathrm{real}}} \mathrm{err}^{l,\lambda}(\overline{\mathbf{x}}^\lambda, \mathbf{x}_r^\lambda)}{N_{\mathrm{real}}}, \tag{2}$$

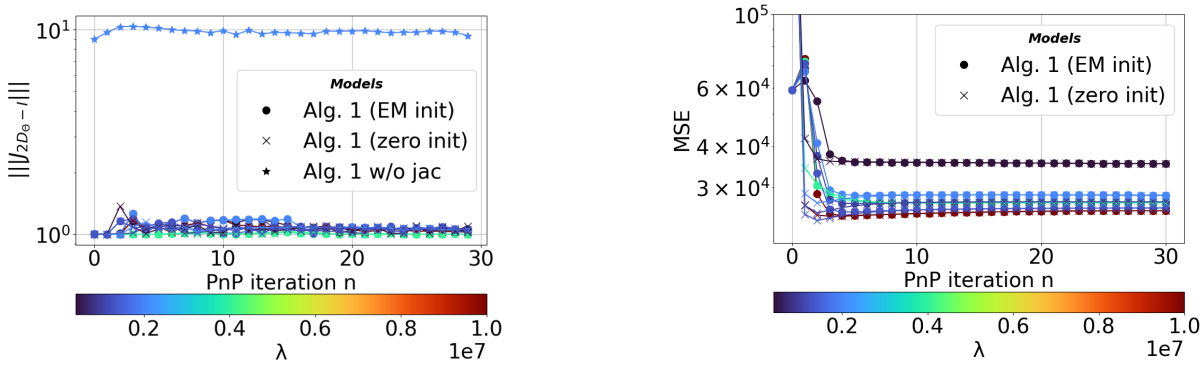
and the mean image multiscale error $\mathrm{bias}^{l,\lambda}$ to the phantom

$$\mathrm{bias}^{l,\lambda} = \mathrm{err}^{l,\lambda}(\overline{\mathbf{x}}^\lambda, \mathbf{x}^{\mathrm{Ph}}). \tag{3}$$

Our ROI-based metrics are computed as

$$\mathrm{bias}^{\mathrm{ROI},\lambda} = 100 \times \sqrt{\frac{\sum_j^{J_{\mathrm{ROI}}} (\overline{\mathbf{x}}_j^\lambda - \mathbf{x}_j^{\mathrm{Ph}})^2}{\sum_j^{J_{\mathrm{ROI}}} (\mathbf{x}_j^{\mathrm{Ph}})^2}} \tag{4}$$

$$\mathrm{std}^{\mathrm{ROI},\lambda} = 100 \times \sqrt{\frac{\sum_{r=1}^{N_{\mathrm{real}}} \sum_j^{J_{\mathrm{ROI}}} (\overline{\mathbf{x}}_j^\lambda - \mathbf{x}_{j,r}^\lambda)^2}{N_{\mathrm{real}} \sum_j^{J_{\mathrm{ROI}}} (\mathbf{x}_j^{\mathrm{Ph}})^2}}. \tag{5}$$



(a) Evolution of $J_{2\mathbf{D}_\Theta}(\mathbf{x}^n + \mathbf{u}^n)$ along PnP iterations

(b) Evolution of the MSE between $\mathbf{x}^n$ and $\mathbf{x}^{\mathrm{Ph}}$ along PnP iterations

Fig. 1: Comparison of Algorithm 1 with $\mathbf{D}_\Theta$ with two different initial estimates (EM or zero). Results are shown for one realization with different values of λ.

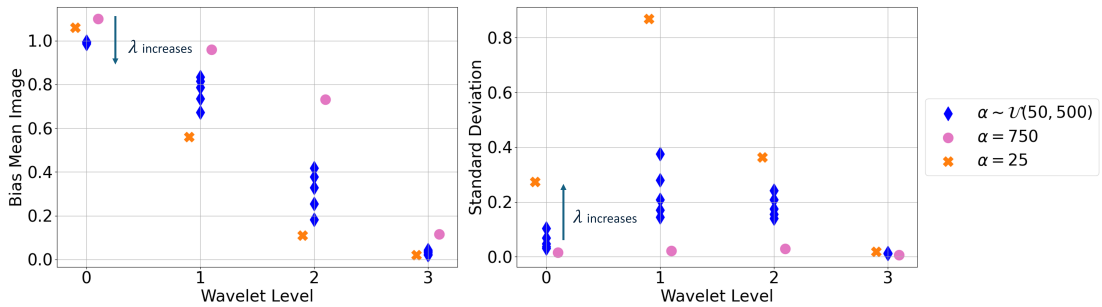


Fig. 2: Multiscale analysis of Algorithm 1 with different values of λ and three networks $\mathbf{D}_\Theta$ trained for $\lambda_{\mathrm{Train}} = \alpha_{\mathrm{Train}} \times \sqrt{\|\mathbf{H}\overline{\mathbf{x}} + \mathbf{b}\|_1}$ with $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$, $\alpha_{\mathrm{Train}} = 25$, $\alpha_{\mathrm{Train}} = 750$.

III. Learning with a local Jacobian regularization

Figure 1a shows the evolution of the spectral norm of the Jacobian of $2\mathbf{D}_\Theta - \mathrm{Id}$ applied on the iterates $\mathbf{x}_n + \mathbf{u}_n$ in Algorithm 1 run for different values of λ and two initial points ($(\mathbf{x}_0, \mathbf{z}_0) = (\mathbf{x}_{\mathrm{EM}}, \mathbf{D}_\Theta(\mathbf{x}_{\mathrm{EM}}))$ and $(\mathbf{x}_0, \mathbf{z}_0) = (0, 0)$). As a comparison, we also reported the value of the spectral norm associated with our DRUnet after our pre-training stage without Jacobian regularization for a fixed value of λ. We see that the spectral norm remains close to 1 for all values of λ and both initial values with Jacobian regularization, whereas, without it, the spectral norm is close to 10. Figure 1b shows the evolution of the MSE between $\mathbf{x}_n$ and $\mathbf{x}^{\mathrm{Ph}}$ along the iteration of Algorithm 1 with the same settings as Fig. 1a. The MSE converges to the same value for a given likelihood weight λ. This indicates that our local Jacobian regularization, when coupled with the nonzero quadratic term of our reconstruction, leads to convergence to the unique fixed point.

IV. Setting α for learning

Figure 2 illustrates the mean and standard deviation of the multiscale errors on the data from Phantom B for three FNE DRUnets plugged in Algorithm 1. The networks are trained with different values of α_{Train}: $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$, $\alpha_{\mathrm{Train}} = 25$ and $\alpha_{\mathrm{Train}} = 750$. Algorithm 1 is run with the same range of values for $\alpha = \alpha_{\mathrm{Train}}$.

Using $\alpha_{\mathrm{Train}} = 25$ yields the highest STD values, meaning that the replicate reconstructions are noisy. If training with $\alpha_{\mathrm{Train}} = 750$ strongly reduces the STD values across all scales, the reconstructions exhibited a significant degradation at the coarse scales in terms of bias. Using $\alpha_{\mathrm{Train}} \sim \mathcal{U}(50, 500)$ yields the lowest bias at the finest scale. At coarser scales (levels 1, 2 and 3), it strongly reduces the STD compared to $\alpha_{\mathrm{Train}} = 25$ while only slightly degrading the bias.

This experiment shows that choosing α_{Train} and thus λ_{Train} is crucial for generalization in PnP ADMM. Mitigating this effect can be achieved by training on a range of values for α_{Train}, which leads to the best compromise in terms of better bias/STD tradeoff.

V. Residual images

Figures 3 shows residual maps for our out-of-distribution cases (synthetics tumors with lower dose). This confirms visually that PnP-DEQ and PnP-DU are the most robust learning-based methods.

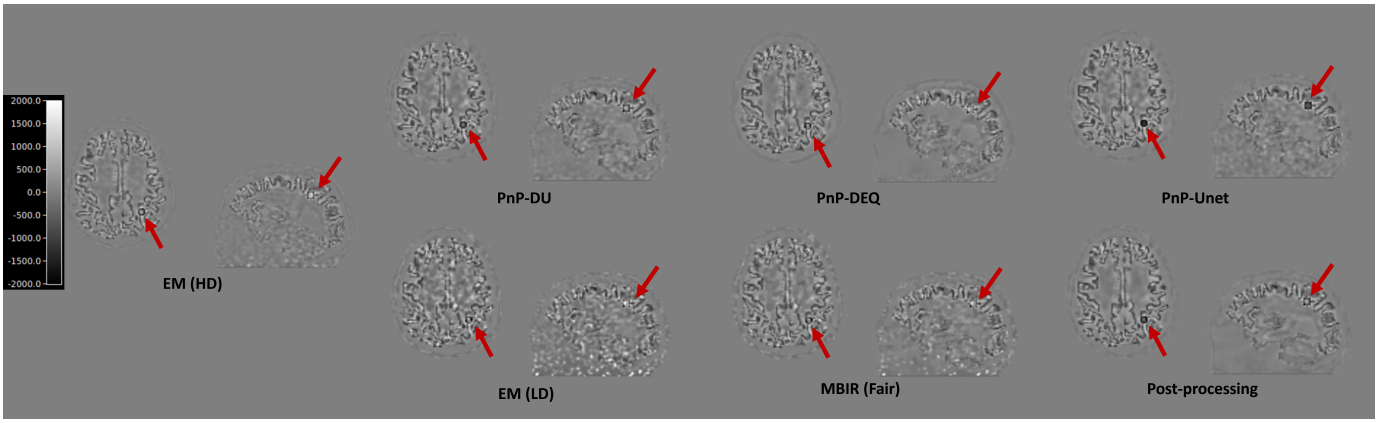


Fig. 3: Residual images to Phantom B for one replicate (dose reduction of 10) and regularization parameters indicated by yellow circles in the main document. The red arrow points to Tumor 3.