CNN-Based Projected Gradient Descent for Consistent CT Image Reconstruction

Harshit Gupta, Kyong Hwan Jin, Ha Q. Nguyen, Michael T. McCann, *Member, IEEE*, and Michael Unser, *Fellow, IEEE*

***Abstract*— We present a new image reconstruction method that replaces the projector in a projected gradient descent (PGD) with a convolutional neural network (CNN). Recently, CNNs trained as image-to-image regressors have been successfully used to solve inverse problems in imaging. However, unlike existing iterative image reconstruction algorithms, these CNN-based approaches usually lack a feedback mechanism to enforce that the reconstructed image is consistent with the measurements. We propose a relaxed version of PGD wherein gradient descent enforces measurement consistency, while a CNN recursively projects the solution closer to the space of desired reconstruction images. We show that this algorithm is guaranteed to converge and, under certain conditions, converges to a local minimum of a non-convex inverse problem. Finally, we propose a simple scheme to train the CNN to act like a projector. Our experiments on sparse-view computed-tomography reconstruction show an improvement over total variation-based regularization, dictionary learning, and a state-of-the-art deep learning-based direct reconstruction technique.**

***Index Terms*— Deep learning, inverse problems, biomedical image reconstruction, low-dose computed tomography.**

I. Introduction

WHILE medical imaging is a fairly mature area, there is recent evidence that it may still be possible to reduce the radiation dose and/or speedup the acquisition process without compromising image quality. This can be accomplished with the help of sophisticated reconstruction algorithms that incorporate some prior knowledge (*e.g.*, sparsity) on the class of underlying images [1]. The reconstruction task is usually formulated as an inverse problem where the image-formation physics are modeled by an operator $\mathbf{H} : \mathbb{R}^N \to \mathbb{R}^M$ (called the *forward model*). The measurement equation is $\mathbf{y} = \mathbf{Hx} + \mathbf{n} \in \mathbb{R}^M$, where $\mathbf{x} \in \mathbb{R}^N$ is the space-domain image that we are interested in recovering and $\mathbf{n} \in \mathbb{R}^M$ is the noise intrinsic to the acquisition process.

In the case of *extreme imaging*, the number of measurements is reduced as much as possible to decrease either the radiation dose in computed tomography (CT) or the scanning time in MRI. Moreover, the measurements are typically very noisy due to short integration times, which calls for some form of denoising. Indeed, there may be significantly fewer measurements than the number of unknowns ($M \ll N$). This gives rise to an ill-posed problem in the sense that there may be an infinity of consistent images that map to the same measurements $\mathbf{y}$. Thus, one challenge of the reconstruction algorithm is to select the best solution among a multitude of potential candidates.

The available reconstruction algorithms can be broadly arranged in three categories (or generations), which represent the continued efforts of the research community to address the aforementioned challenges.

1) *Classical Algorithms:* Here, the reconstruction is performed directly by applying a suitable linear operator. In the case where $\mathbf{H}$ is unitary (as in a simple MRI model), the operator is simply the backprojection (BP) $\mathbf{H}^{\mathrm{T}}\mathbf{y}$. In general, the reconstruction operator should approximate a pseudoinverse of $\mathbf{H}$. For example, the filtered backprojection (FBP) for x-ray CT involves applying a linear filter to the measurements and back projecting them, *i.e.* $\mathbf{H}^{\mathrm{T}}\mathbf{F}\mathbf{y}$ where $\mathbf{F} : \mathbb{R}^M \to \mathbb{R}^M$. Though its expression is usually derived in the continuous domain [2], the filter $\mathbf{F}$ can be viewed as an approximate version of $(\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}$. Classical algorithms are fast and provide excellent results when the number of measurements is large and the noise is small [3]. However, they are not suitable for extreme imaging because they introduce artifacts that are intimately connected to the inversion step.
2) *Iterative Algorithms:* These algorithms avoid the shortcomings of the classical ones by solving

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathbb{R}^N} \left(E(\mathbf{Hx}, \mathbf{y}) + \lambda R(\mathbf{x})\right), \tag{1}$$

where $E : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}^+$ is a data-fidelity term that favors solutions that are consistent with the mea-

Manuscript received February 13, 2018; revised April 24, 2018; accepted April 25, 2018. Date of publication May 3, 2018; date of current version May 31, 2018. This work was supported in part by the European Research Council (H2020-ERC Project GlobalBioIm) under Grant 692726 and in part by the European Union's Horizon 2020 Framework Programme for Research and Innovation (call 2015) under Grant 665667. *(Corresponding author: Harshit Gupta.)*

H. Gupta, K. H. Jin, and M. Unser are with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: harshit.gupta.cor@gmail.com).

H. Q. Nguyen was with the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. He is now with the Viettel Research and Development Institute, Hanoi VN-100000, Vietnam.

M. T. McCann is with the Center for Biomedical Imaging, Signal Processing Core and the Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMI.2018.2832656

surements, $R : \mathbb{R}^N \to \mathbb{R}^+$ is a suitable regularizer that encodes prior knowledge about the image $\mathbf{x}$ to be reconstructed, and $\lambda \in \mathbb{R}^+$ is a tradeoff parameter. For example, in CT reconstruction, E could be weighted least-squares and R could be an indicator function that enforces non-negativity. Under the assumption that the functionals E and R are convex, the solution of (1) also satisfies

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{S}_R} E(\mathbf{H}\mathbf{x}, \mathbf{y}) \tag{2}$$

with $\mathcal{S}_R = \{\mathbf{x} \in \mathbb{R}^N : R(\mathbf{x}) \leq \tau\}$ for some unique τ that depends on the regularizaton parameter λ. Therefore, the solution has the best data fidelity among all images in the set $\mathcal{S}_R$ which is implicitly defined by R. This shows that the quality of the reconstruction depends heavily on the prior encoder R. Generally, these priors are either hand-picked (*e.g.*, total variation (TV) or the ℓ_1-norm of the wavelet coefficients of the image [1], [4]–[7]) or learned through a dictionary [8]–[10]. However, in either case, they are restricted to well-behaved functionals that can be minimized via a convex routine [11]–[14]. This limits the type of prior knowledge that can be injected into the algorithm.

3) *Learning-Based Algorithms:* Recently, a surge in using deep learning to solve inverse problems in imaging [15]–[19], has established new state-of-the-art results for tasks such as sparse-view CT reconstruction [16]. Rather than reconstructing the image from the measurements $\mathbf{y}$ directly, the most successful strategies have been to train the CNN as a regressor between a rough initial reconstruction $\mathbf{A}\mathbf{y}$, where $\mathbf{A} : \mathbb{R}^M \to \mathbb{R}^N$, and the final, desired reconstruction [16], [17]. This initial reconstruction could be obtained using classical algorithms (*e.g.*, FBP, BP) or by some other linear operation. Once the training is complete, the reconstruction for a new measurement $\mathbf{y}$ is given by $\mathbf{x}^* = CNN_{\boldsymbol{\theta}^*}(\mathbf{A}\mathbf{y})$, where $CNN_{\boldsymbol{\theta}} : \mathbb{R}^N \to \mathbb{R}^N$ denotes the CNN as a function and $\boldsymbol{\theta}^*$ denotes the internal parameters of the CNN after training. These schemes exploit the fact that the structure of images can be learned from representative examples. CNNs are favored because of the way they encode the data in their hidden layers. In this sense, a CNN can be seen as a good prior encoder.

Although the results reported so far are remarkable in terms of image quality, there is still some concern as to whether or not they can be trusted, especially in the context of diagnostic imaging. The main limitation of direct algorithms such as [16] is that they do not provide any guarantee on the worst-case performance. Moreover, even in the case of noiseless (or low-noise) measurements, there is no insurance that the reconstructed image is consistent with the measurements because, unlike for the iterative schemes, there is no feedback mechanism that imposes this consistency.

A. Overview of Proposed Method

In this paper, we present a simple yet effective iterative scheme (see Figure 1), which tries to incorporate the advantages of the existing algorithms and side-steps their disadvantages. Specifically:

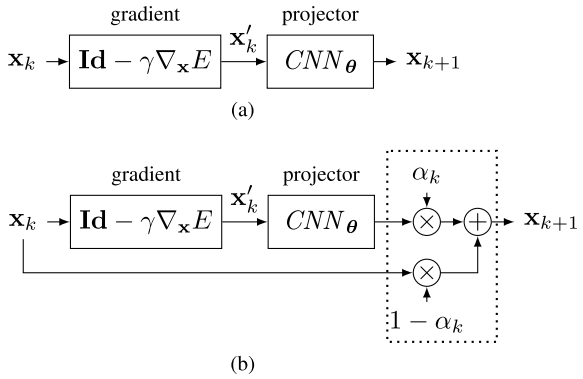

Fig. 1. (a) Block diagram of projected gradient descent using a CNN as the projector and E as the data-fidelity term. The gradient step promotes consistency with the measurements and the projector forces the solution to belong to the set of desired solutions. If the CNN is only an approximate projector, the scheme may diverge. (b) Block diagram of the proposed relaxed projected gradient descent. The α_ks are updated in such a way that the algorithm always converges (see Algorithm 1 for more details).

- We first propose to learn a CNN that acts as a projector onto a set $\mathcal{S}$ which can be intuitively thought of as the manifold of the data (*e.g.*, biomedical images). In this sense, our CNN encodes the prior knowledge of the data. Its purpose is to map an input image to an output image that is more similar to the training data.
- Given a measurement $\mathbf{y}$, we initialize our reconstruction using a classical algorithm.
- We then iteratively alternate between minimizing the data-fidelity term and projecting the result onto the set S by applying a suitable variant of the projected gradient descent (PGD) which ensures convergence.

Besides the design of the implementation, our contribution is in the proposal of the relaxed form of PGD that is guaranteed to converge and under certain conditions can also find a local minima of a nonconvex inverse problem. Moreover, as we shall see later, this method outperforms existing algorithms on low-dose x-ray CT reconstructions.

B. Related and Prior Work

Deep learning has already shown promising results in image denoising, superresolution, and deconvolution. Recently, it has also been used to solve inverse problems in imaging using limited data [16]–[19], and in compressed sensing [20]. However, as discussed earlier, these regression-based approaches lack a feedback mechanism that could be beneficial in solving inverse problems.

Another usage of deep learning is to complement iterative algorithms. This includes learning a CNN as an unrolled version of the iterative shrinkage-thresholding algorithm (ISTA) [21] or ADMM [22]. In [23], inverse problems involving non-linear forward models are solved by partially learning the gradient descent. In [24], the iterative algorithm is replaced by a recurrent neural network (RNN). Recently, in [25], a cascade of CNNs is used to reconstruct images.

Within this cascade the data-fidelity is enforced at multiple steps. However, in all of these approaches the training is performed end-to-end, meaning that the network parameters are dependent on the iterative scheme chosen.

These approaches differ from plug-and-play ADMM [26]–[28], where an independent off-the-shelf denoiser or a trained operator is plugged into the iterative scheme of the alternating-direction method of multipliers (ADMM) [14]. ADMM is an iterative optimization technique that alternates between (i) a linear solver that reinforces consistency with respect to the measurements; and (ii) a nonlinear operation that re-injects the prior. The idea of plug-and-play ADMM is to replace (ii), which resembles denoising, with an off-the-shelf denoiser. Plug-and-play ADMM is more general than the optimization framework (1) but still lacks theoretical justifications. In fact, there is little understanding yet of the connection between the use of a given denoiser and the regularization it imposes (though this link has recently been explored in [29]).

In [30], a generative adversarial network (GAN) trained as a projector onto a set, has been used with the plug-and-play ADMM. Similarly, in [31], the inverse problem is solved over a set parameterised by a generative model. However, it requires a precise initialization of the parameters. In [32], similarly to us, the projector in PGD is replaced with a neural network. However, the scheme lacks convergence guarantee and a rigorous theoretical analysis.

Our scheme is similar in spirit to plug-and-play ADMM, but is simpler to analyze. Although our methodology is generic and can be applied in principle to any inverse problem, our experiments here involve sparse-view x-ray CT reconstruction. For a recent overview of the field, see [33]. Current approaches to sparse-view CT reconstruction follow the formulation (1), e.g., using a penalized weighted least-squares data term and sparsity-promoting regularizer [34], dictionary learning-based regularizer [35], or generalized total variation regularizer [36]. There are also prior works on the direct application of CNNs to CT reconstruction. These methods generally use the CNN to denoise the sinogram [37] or the reconstruction obtained from a standard technique [16], [38]–[40]; as such, they do not perform the reconstruction directly.

C. Roadmap

The paper is organized as follows: In Section II, we discuss the mathematical framework that motivates our approach and justify the use of a projector onto a set as an effective strategy to solve inverse problems. In Section III, we present our algorithm, which is a relaxed version of PGD. It has been modified so as to converge in practical cases where the projection property is only approximate. We discuss in Section IV a novel technique to train the CNN as a projector onto a set, especially when the training data is small. This is followed by experiments (Section V), results and discussions (Section VI and Section VII), and conclusions (Section VIII).

II. Theoretical Framework

Our goal is to use a trained CNN iteratively inside PGD to solve an inverse problem. To understand why this scheme will be effective, we first analyze how using a projector onto a set, combined with gradient descent, can be helpful in solving inverse problems. Properties of PGD using an orthogonal projector onto a convex set are known [41]. Here, we extend these results for any projector onto a nonconvex set. This extension is required because there is no guarantee that the set of desirable reconstruction images is convex. Proofs of all the results in this section can be found in the supplementary material.

A. Notation

We consider the finite-dimensional Hilbert space $\mathbb{R}^N$ equipped with the scalar product $\langle \cdot, \cdot \rangle$ that induces the ℓ_2 norm $\|\cdot\|_2$. The spectral norm of the matrix $\mathbf{H}$, denoted by $\|\mathbf{H}\|_2$, is equal to its largest singular value. For $\mathbf{x} \in \mathbb{R}^N$ and $\varepsilon > 0$, we denote by $\mathcal{B}_\varepsilon(\mathbf{x})$ the ℓ_2-ball centered at $\mathbf{x}$ with radius ε, *i.e.*,

$$\mathcal{B}_\varepsilon(\mathbf{x}) = \left\{ \mathbf{z} \in \mathbb{R}^N : \|\mathbf{z} - \mathbf{x}\|_2 \leq \varepsilon \right\}.$$

The operator $T : \mathbb{R}^N \to \mathbb{R}^N$ is Lipschitz-continuous with constant L if

$$\|T(\mathbf{x}) - T(\mathbf{z})\|_2 \leq L \|\mathbf{x} - \mathbf{z}\|_2, \quad \forall \mathbf{x}, \mathbf{z} \in \mathbb{R}^N.$$

It is contractive if it is Lipschitz-continuous with constant $L < 1$ and non-expansive if $L = 1$. A fixed point $\mathbf{x}^*$ of T (if any) satisfies $T(\mathbf{x}^*) = \mathbf{x}^*$.

Given the set $\mathcal{S} \subset \mathbb{R}^N$, the mapping $P_\mathcal{S} : \mathbb{R}^N \to \mathcal{S}$ is called a projector if it satisfies the idempotent property $P_\mathcal{S} P_\mathcal{S} = P_\mathcal{S}$. It is called an orthogonal projector if

$$P_\mathcal{S}(\mathbf{x}) = \inf_{\mathbf{z} \in \mathcal{S}} \|\mathbf{x} - \mathbf{z}\|_2, \quad \forall \mathbf{x} \in \mathbb{R}^N.$$

B. Constrained Least Squares

Consider the problem of the reconstruction of the image $\mathbf{x} \in \mathbb{R}^N$ from its noisy measurements $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where $\mathbf{H} \in \mathbb{R}^{M \times N}$ is the linear forward model and $\mathbf{n} \in \mathbb{R}^M$ is additive white Gaussian noise. The framework is also applicable to Poisson noise model-based CT via a suitable transformation, as shown in Appendix B.

Our reconstruction incorporates a strong form of prior knowledge about the original image: We assume that $\mathbf{x}$ must lie in a set $\mathcal{S} \subset \mathbb{R}^N$ that contains all objects of interest. The proposed way to make the reconstruction consistent with the measurements as well as with the prior knowledge is to solve the constrained least-squares problem

$$\min_{\mathbf{x} \in \mathcal{S}} \frac{1}{2} \|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2. \tag{3}$$

The condition $\mathbf{x} \in \mathcal{S}$ in (3) plays the role of a regularizer. If no two points in $\mathcal{S}$ have the same measurements and in case $\mathbf{y}$ is noiseless, then out of all the points in $\mathbb{R}^N$ that are consistent with the measurement $\mathbf{y}$, (3) selects a unique point $\mathbf{x}^* \in \mathcal{S}$. In this way, the ill-posedness of the inverse problem is bypassed. When the measurements are noisy, (3) returns a point $\mathbf{x}^* \in \mathcal{S}$ such that $\mathbf{y}^* = \mathbf{H}\mathbf{x}^*$ is as close as possible to $\mathbf{y}$. Thus, it also denoises the measurement, where

the quantity $\mathbf{y}^*$ can be regarded as the denoised version of $\mathbf{y}$. Note that formulation (3) is similar to (2) for the case when E is least-squares, with the difference that the search space is the data manifold $\mathcal{S}$ instead of a set defined by the regularizer $\mathcal{S}_R$.

The point $\mathbf{x}^* \in \mathcal{S}$ is called a local minimizer of (3) if

$$\exists \varepsilon > 0 : \left\|\mathbf{H}\mathbf{x}^* - \mathbf{y}\right\|_2 \leq \left\|\mathbf{H}\mathbf{x} - \mathbf{y}\right\|_2, \quad \forall \mathbf{x} \in \mathcal{S} \cap \mathcal{B}_\varepsilon(\mathbf{x}^*).$$

C. Projected Gradient Descent

When $\mathcal{S}$ is a closed convex set, it is well known [41] that a solution of (3) can be found by PGD

$$\mathbf{x}_{k+1} = P_{\mathcal{S}}(\mathbf{x}_k - \gamma\, \mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{x}_k + \gamma\, \mathbf{H}^{\mathrm{T}}\mathbf{y}), \tag{4}$$

where γ is a step size chosen such that $\gamma < 2/\left\|\mathbf{H}^{\mathrm{T}}\mathbf{H}\right\|_2$. This algorithm combines the orthogonal projection onto $\mathcal{S}$ with the gradient descent with respect to the quadratic objective function, also called the Landweber update [42]. PGD [43, Sec. 2.3] is a subclass of the forward-backward splitting [44], [45], which is known in the ℓ_1-minimization literature as iterative shrinkage/thresholding algorithms (ISTA) [11], [12], [46].

In our problem, $\mathcal{S}$ is presumably non-convex, but we propose to still use the update (4) with some projector $P_{\mathcal{S}}$ that may not be orthogonal. In the rest of this section, we provide sufficient conditions on the projector $P_{\mathcal{S}}$ (not on $\mathcal{S}$ itself) under which (4) leads to a local minimizer of (3). Similarly to the convex case, we characterize the local minimizers of (3) by the fixed points of the combined operator

$$G_\gamma(\mathbf{x}) = P_{\mathcal{S}}(\mathbf{x} - \gamma\, \mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{x} + \gamma\, \mathbf{H}^{\mathrm{T}}\mathbf{y}) \tag{5}$$

and then show that some fixed point of that operator must be reached by the iteration $\mathbf{x}_{k+1} = G_\gamma(\mathbf{x}_k)$ as $k \to \infty$, regardless of the initial point $\mathbf{x}_0$. We first state a sufficient condition for each fixed point of G_γ to become a local minimizer of (3).

Proposition 1: Let $\gamma > 0$ and $P_{\mathcal{S}}$ be such that, for all $\mathbf{x} \in \mathbb{R}^N$,

$$\langle \mathbf{z} - P_{\mathcal{S}}\mathbf{x}\,,\, \mathbf{x} - P_{\mathcal{S}}\mathbf{x} \rangle \leq 0, \quad \forall \mathbf{z} \in \mathcal{S} \cap \mathcal{B}_\varepsilon(P_{\mathcal{S}}\mathbf{x}), \tag{6}$$

for some $\varepsilon > 0$. Then, any fixed point of the operator G_γ in (5) is a local minimizer of (3). Furthermore, if (6) is satisfied globally, in the sense that

$$\langle \mathbf{z} - P_{\mathcal{S}}\mathbf{x}\,,\, \mathbf{x} - P_{\mathcal{S}}\mathbf{x} \rangle \leq 0, \quad \forall \mathbf{x} \in \mathbb{R}^N, \mathbf{z} \in \mathcal{S}, \tag{7}$$

then any fixed point of G_γ is a solution of (3).

Two remarks are in order. First, (7) is a well-known property of orthogonal projections onto closed convex sets. It actually implies the convexity of $\mathcal{S}$ (see Proposition 2). Second, (6) is much more relaxed and easily achievable, for example, as stated in Proposition 3, by orthogonal projections onto unions of closed convex sets. (Special cases are unions of subspaces, which have found some applications in data modeling and clustering [47]).

Proposition 2: If $P_{\mathcal{S}}$ is a projector onto $\mathcal{S} \subset \mathbb{R}^N$ that satisfies (7), then $\mathcal{S}$ must be convex.

Proposition 3: If $\mathcal{S}$ is a union of a finite number of closed convex sets in $\mathbb{R}^N$, then the orthogonal projector $P_{\mathcal{S}}$ onto $\mathcal{S}$ satisfies (6).

Propositions 1-3 suggest that, when $\mathcal{S}$ is non-convex, the best we can hope for is to find a local minimizer of (3) through a fixed point of G_γ. Theorem 1 provides a sufficient condition for PGD to converge to a unique fixed point of G_γ.

Theorem 1: Let $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and smallest eigenvalues of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$, respectively. If $P_{\mathcal{S}}$ satisfies (6) and is Lipschitz-continuous with constant $L < (\lambda_{\max}+\lambda_{\min})/(\lambda_{\max}-\lambda_{\min})$, then, for $\gamma = 2/(\lambda_{\max}+\lambda_{\min})$, the sequence $\{\mathbf{x}_k\}$ generated by (4) converges to a local minimizer of (3), regardless of the initialization $\mathbf{x}_0$.

It is important to note that the projector $P_{\mathcal{S}}$ can never be contractive since it preserves the distance between any two points on $\mathcal{S}$. Therefore, when $\mathbf{H}$ has a nontrivial null space, the condition $L < (\lambda_{\max}+\lambda_{\min})/(\lambda_{\max}-\lambda_{\min})$ of Theorem 1 is not feasible. The smallest possible Lipschitz constant of $P_{\mathcal{S}}$ is $L = 1$, which means that $P_{\mathcal{S}}$ is non-expansive. Even with this condition, it is not guaranteed that the combined operator F_γ has a fixed point. This limitation can be overcome when F_γ is assumed to have a nonempty set of fixed points. Indeed, we state in Theorem 2 that one of them must be reached by iterating the averaged operator $\alpha\,\mathrm{Id} + (1-\alpha)G_\gamma$, where $\alpha \in (0,1)$ and Id is the identity operator. We call this scheme averaged PGD (APGD).

Theorem 2: Let $\lambda_{\max}$ be the largest eigenvalue of $\mathbf{H}^{\mathrm{T}}\mathbf{H}$. If $P_{\mathcal{S}}$ satisfies (6) and is a non-expansive operator such that G_γ in (5) has a fixed point for some $\gamma < 2/\lambda_{\max}$, then the sequence $\{\mathbf{x}_k\}$ generated by APGD, with

$$\mathbf{x}_{k+1} = (1-\alpha)\mathbf{x}_k + \alpha G_\gamma(\mathbf{x}_k) \tag{8}$$

for any $\alpha \in (0,1)$, converges to a local minimizer of (3), regardless of the initialization $\mathbf{x}_0$.

III. Relaxation with Guaranteed Convergence

Despite their elegance, Theorems 1 and 2 are not directly productive when we construct the projector $P_{\mathcal{S}}$ by training a CNN because it is unclear how to enforce the Lipschitz continuity of $P_{\mathcal{S}}$ on the CNN architecture. Without putting any constraints on the CNN, however, we can still achieve the convergence of the reconstruction sequence by modifying PGD as described in Algorithm 1; we name it relaxed projected gradient descent (RPGD). In Algorithm 1, the projector $P_{\mathcal{S}}$ is replaced by the general nonlinear operator F. We also introduce a sequence $\{c_k\}$ that governs the rate of convergence of the algorithm and a sequence $\{\alpha_k\}$ of relaxation parameters that evolves with the algorithm. The convergence of RPGD is guaranteed by Theorem 3. More importantly, if the nonlinear operator F is actually a projector and the relaxation parameters do not go all the way to 0, then RPGD converges to a meaningful point.

Theorem 3: Let the input sequence $\{c_k\}$ of Algorithm 1 be asymptotically upper-bounded by $C < 1$. Then, the following statements hold true for the reconstruction sequence $\{\mathbf{x}_k\}$:

(i) *$\mathbf{x}_k \to \mathbf{x}^*$ as $k \to \infty$, for all choices of F;*
(ii) *if F is continuous and the relaxation parameters $\{\alpha_k\}$ are lower-bounded by $\varepsilon > 0$, then $\mathbf{x}^*$ is a fixed*

Algorithm 1 Relaxed Projected Gradient Descent (RPGD)

Input: $\mathbf{H}$, $\mathbf{y}$, $\mathbf{A}$, nonlinear operator F, step size $\gamma > 0$, positive sequence $\{c_n\}_{n\geq 1}$, $\mathbf{x}_0 = \mathbf{A}\mathbf{y} \in \mathbb{R}^N$, $\alpha_0 \in (0, 1]$.
Output: reconstructions $\{\mathbf{x}_k\}$, relaxation parameters $\{\alpha_k\}$.

$k \leftarrow 0$
while not converged **do**
  $\mathbf{z}_k = F(\mathbf{x}_k - \gamma\,\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{x}_k + \gamma\,\mathbf{H}^{\mathrm{T}}\mathbf{y})$
  if $k \geq 1$ **then**
    if $\|\mathbf{z}_k - \mathbf{x}_k\|_2 > c_k\,\|\mathbf{z}_{k-1} - \mathbf{x}_{k-1}\|_2$ **then**
      $\alpha_k = c_k\|\mathbf{z}_{k-1} - \mathbf{x}_{k-1}\|_2 / \|\mathbf{z}_k - \mathbf{x}_k\|_2\,\alpha_{k-1}$
    else
      $\alpha_k = \alpha_{k-1}$
    end if
  end if
  $\mathbf{x}_{k+1} = (1 - \alpha_k)\mathbf{x}_k + \alpha_k\mathbf{z}_k$
  $k \leftarrow k + 1$
end while

point of

$$G_\gamma(\mathbf{x}) = F(\mathbf{x} - \gamma\,\mathbf{H}^{\mathrm{T}}\mathbf{H}\mathbf{x} + \gamma\,\mathbf{H}^{\mathrm{T}}\mathbf{y}); \tag{9}$$

(iii) *if, in addition to* (ii), *F is indeed a projector onto $\mathcal{S}$ that satisfies* (6), *then $\mathbf{x}^*$ is a local minimizer of* (3).

We prove Theorem 3 in Appendix A. Note that the weakest statement here is (i); it guarantees that RPGD always converges, albeit not necessarily to a fixed point of G_γ. Moreover, the assumption about the continuity of F in (ii) is automatically satisfied when F is a CNN.

In summary, we have described three algorithms: PGD, APGD, and RPGD. PGD is a standard algorithm which, in the event of convergence, finds a local minima of (3); however, it does not always converge. APGD ensures convergence under the broader set of conditions given in Theorem 2; but, in order to have these properties, both PGD and APGD necessarily need a projector. While, we shall train our CNN to act like a projector, it may not exactly fulfill the required conditions. This is the motivation for RPGD, which, unlike PGD and APGD, is guaranteed to converge. It also retains the desirable properties of PGD and APGD: it finds a local minima of (3), given that the conditions (ii) and (iii) of Theorem 3 are satisfied. Note, however, that when the set $\mathcal{S}$ is nonconvex, this local minimum may not be a global minimum. The results of Section II and III are summarized in Table IV given in the supplementary material.

IV. Training a CNN as a Projector

For any point $\mathbf{x} \in \mathcal{S}$, a projector onto $\mathcal{S}$ should satisfy $P_{\mathcal{S}}\mathbf{x} = \mathbf{x}$. Moreover, we want that

$$\mathbf{x} = P_{\mathcal{S}}(\tilde{\mathbf{x}}), \tag{10}$$

where $\tilde{\mathbf{x}}$ is any perturbed version of $\mathbf{x}$. Given the training set, $\{\mathbf{x}^1, \ldots, \mathbf{x}^Q\}$ of Q points drawn from the set $\mathcal{S}$, we generate the ensemble $\{\{\tilde{\mathbf{x}}^{1,1}, \ldots, \tilde{\mathbf{x}}^{Q,1}\}, \ldots, \{\tilde{\mathbf{x}}^{1,N} \ldots, \tilde{\mathbf{x}}^{Q,N}\}\}$ of $N \times Q$ perturbed points and train the CNN by minimizing the loss function

$$J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \underbrace{\sum_{q=1}^{Q} \left\|\mathbf{x}^q - CNN_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^{q,n})\right\|_2^2}_{J_n(\boldsymbol{\theta})}. \tag{11}$$

The optimization proceeds by stochastic gradient descent for T epochs, where an epoch is defined as one pass though the training data.

It remains to select the perturbations that generate the $\mathbf{x}^{q,n}$. Our goal here is to create a diverse set of perturbations so that the CNN does not overfit one specific type. In our experiments, while training for the tth epoch, we chose

$$\tilde{\mathbf{x}}^{q,1} = \mathbf{x}^q \tag{12}$$

$$\tilde{\mathbf{x}}^{q,2} = \mathbf{A}\mathbf{H}\mathbf{x}^q \tag{13}$$

$$\tilde{\mathbf{x}}^{q,3} = CNN_{\boldsymbol{\theta}_{t-1}}(\tilde{\mathbf{x}}^{q,2}), \tag{14}$$

where $\mathbf{A}$ is a classical linear reconstruction algorithm (FBP in our experiments), and $\boldsymbol{\theta}_t$ are the CNN parameters after t epochs. Equations (12), (13), and (14) correspond to no perturbation, a linear perturbation, and a dynamic nonlinear perturbation, respectively. We now comment on each perturbation in detail.

Keeping $\tilde{\mathbf{x}}^{q,1}$ in the training ensemble will train the CNN with the defining property of the projector: the projector maps a point in the set $\mathcal{S}$ onto itself. If the CNN were trained only with (12), it would be an autoencoder [48].

To understand the perturbation $\tilde{\mathbf{x}}^{q,2}$ in (13), recall that $\mathbf{A}\mathbf{H}\mathbf{x}^q$ is the classical linear reconstruction of $\mathbf{x}^q$ from its measurement $\mathbf{y} = \mathbf{H}\mathbf{x}^q$. Perturbation (13) is indeed useful because we initialize RPGD with $\mathbf{A}\mathbf{H}\mathbf{x}^q$. Using only (13) for training would return the same CNN as in [16].

The perturbation $\tilde{\mathbf{x}}^{q,3}$ in (14) is the output of the CNN whose parameters $\boldsymbol{\theta}_t$ change with every epoch t; thus, it is a nonlinear and dynamic (epoch-dependent) perturbation of $\mathbf{x}^q$. The rationale for using (14) is that it greatly increases the training diversity by allowing the network to see T new perturbations of each training point, without greatly increasing the total training size since it only requires Q additional gradient computations per epoch. Moreover, (14) is in sync with the iterative scheme of RPGD, where the output of the CNN is processed with a gradient descent and is again fed back into itself.

A. Architecture

Our CNN architecture is the same as in [16], which is a U-net [49] with intrinsic skip connections among its layers and an extrinsic skip connection between the input and the output. The intrinsic skip connections help to eliminate singularities during the training [50]. The extrinsic skip connections make this network a residual net; *i.e.*, $CNN = \mathrm{Id} + Unet$, where Id denotes the identity operator and $Unet : \mathbb{R}^N \to \mathbb{R}^N$ denotes U-net as a function. Therefore, U-net actually provides the projection error (negative perturbation) that should be added to the input to get the projection.

Residual nets have been shown to be effective for image recognition [51] and for solving inverse problems [16]. While

the residual-net architecture does not increase the capacity or the approximation power of the CNN, it does help in learning functions that are close to an identity operator, as is the case in our setting.

B. Sequential Training Strategy

We train the CNN in three stages. In Stage 1, we train it for T_1 epochs with respect to the partial-loss function J_2 in (11) which only uses the ensemble $\{\tilde{\mathbf{x}}^{q,2}\}$ generated by (13). In Stage 2, we add the ensemble $\{\tilde{\mathbf{x}}^{q,3}\}$ according to (14) at every epoch and then train the CNN with respect to the loss function $J_2 + J_3$; we repeat this procedure for T_2 epochs. Finally, in Stage 3, we train the CNN for T_3 epochs with all three ensembles $\{\tilde{\mathbf{x}}^{q,1}, \tilde{\mathbf{x}}^{q,2}, \tilde{\mathbf{x}}^{q,3}\}$ to minimize the original loss function $J = J_1 + J_2 + J_3$ from (11).

We shall see in Section VII-B that this sequential procedure speeds up the training without compromising the performance. The parameters of *Unet* are initialized by a normal distribution with a very low variance. Since $CNN = \mathrm{Id} + Unet$, this function acts close to an identity operator in the initial epochs and makes it redundant to use $\{\tilde{\mathbf{x}}^{q,1}\}$ for the initial training stages. Therefore, $\{\tilde{\mathbf{x}}^{q,1}\}$ is only added at the last stage when the *CNN* is no longer close to an identity operator. After training with only $\{\tilde{\mathbf{x}}^{q,2}\}$ in Stage 1, $\tilde{\mathbf{x}}^{q,3}$ will be close to $\mathbf{x}^q$ since it is the output of the CNN for the input $\tilde{\mathbf{x}}^{q,2}$. This eases the training for $\{\tilde{\mathbf{x}}^{q,3}\}$ in the second and third stage.

V. Experiments

We validate the proposed method on the challenging case of sparse-view CT reconstruction. Conventionally, CT imaging requires many views to obtain good quality reconstruction. We call this scenario full-dose reconstruction. Our main aim in these experiments is to reduce the number of views (or dose) for CT imaging while retaining the quality of full-dose reconstructions. We denote a k-times reduction in views by $\times k$.

The measurement operator $\mathbf{H}$ for our experiments is the Radon transform. It maps an image to the values of its integrals along a known set of lines [2]. In 2D, the measurements are indexed by the angle and offset of each lines and arranged in a 2D sinogram. We implemented $\mathbf{H}$ and $\mathbf{H}^{\mathrm{T}}$ with Matlab's `radon` and `iradon` (normalized to satisfy the adjoint property), respectively. The Matlab code for the RPGD and the sequential-strategy-based training are made publically available[1].

A. Datasets

We use two datasets for our experiments.

1) Mayo Clinic Dataset. It consists of 500 clinically realistic, (512×512) CT images from the lower lungs to the lower abdomen of 10 patients. Those were obtained from the Mayo clinic AAPM Low Dose CT Grand Challenge [52].

2) Rat Brain Dataset. We use a real (1493 px $\times$ 720 view $\times$ 377 slice) sinogram from a CT scan of a single rat brain. The data acquisition was performed at the Paul Scherrer Institute in Villigen, Switzerland at the TOMCAT beam line of the Swiss Light Source. During pre-processing, we split this sinogram slice-by-slice and downsampled it to create a dataset of 377 (729 px $\times$ 720 view) sinograms. CT images of size (512×512) were then generated from these full-dose sinograms (using the FBP, see Section V-C). For the qth z-slice, we denote the corresponding image $\mathbf{x}^q_{\mathrm{FD}}$. For experiments based on this dataset, the first 327 and the last 25 slices are used for training and testing, respectively. This left a gap of 25 slices in between the training and testing data.

[1] https://github.com/harshit-gupta-epfl/CNN-RPGD

B. Experimental Setups

We now describe three experimental setups. We use the first dataset for the first experiment and the second for the last two.

1) Experiment 1: We split the Mayo dataset into 475 images from 9 patients for training and 25 images from the remaining patient for testing. We assume these images to be the ground truth. From the qth image $\mathbf{x}^q$, we generated the sparse-view sinogram $\mathbf{y}^q = \mathbf{H}\mathbf{x}^q$ using several different experimental conditions. Our task is to reconstruct the image from the sinogram.

The sinograms always have 729 offsets per view, but we varied the number of views and the level of measurement noise for different cases. We took 144 views and 45 views, which corresponds to $\times 5$ and $\times 16$ dosage reductions (assuming a full-view sinogram has 720 views). We added Gaussian noise to the sinograms to make the SNR equal to 35, 40, 45, 70, and infinity dB, where we refer to the first three as *high measurement noise* and the last two as *low measurement noise*. The SNR of the sinogram $\mathbf{y} + \mathbf{n}$ is defined as

$$\mathrm{SNR}(\mathbf{y} + \mathbf{n}, \mathbf{y}) = 20 \log_{10} \left(\|\mathbf{y}\|_2 / \|\mathbf{n}\|_2 \right). \tag{15}$$

For testing with the low and high measurement noise, we trained the CNNs without noise and at the 40-dB level of noise, respectively (see Section V-D for details).

To make the experiments more realistic and to reduce the inverse crime, the sinograms were generated by slightly perturbing the angles of the views by a zero-mean additive white Gaussian noise (AWGN) with standard deviation of 0.05 degrees. This creates a deliberate mismatch between the actual measurement process and the forward model.

2) Experiment 2: We used images $\mathbf{x}^q_{\mathrm{FD}}$ from the rat-brain dataset to generate Poisson-noise-corrupted sinograms $\mathbf{y}^q$ with 144 views. Just as in Experiment 1, the task is to reconstruct $\mathbf{x}^q_{\mathrm{FD}}$ back from $\mathbf{y}^q$. Sinograms were generated with 25, 30, and 35 dB SNR with respect to $\mathbf{H}\mathbf{x}^q_{\mathrm{FD}}$. To achieve this, in (26) and (27), we assume the readout noise to be zero and $\{b_1, \ldots, b_m\} = b_0 = 1.66 \times 10^5, 5.24 \times 10^5$, and 1.66×10^6, respectively. More details about this process is given in Appendix B. The CNNs were trained at only the 30-dB level of noise. Again, our task is to reconstruct the images from the sinograms.

3) Experiment 3. We downsampled the views of the original, (729×720) rat-brain sinograms by 5 to obtain sparse-view sinograms of size (729×144). For the qth z-slice, we denote the corresponding sparse-view sinograms $\mathbf{y}^q_{\mathrm{Real}}$. Note that, unlike in Experiments 1 and 2, the sinogram was not generated from an image but was obtained experimentally.

C. Comparison Methods

Given the ground truth $\mathbf{x}$, our figure of merit for the reconstructed $\mathbf{x}^*$ is the regressed SNR given by

$$\mathrm{SNR}(\mathbf{x}^*, \mathbf{x}) = \arg\max_{a,b} \mathrm{SNR}(a\mathbf{x}^* + b, \mathbf{x}), \tag{16}$$

where the purpose of a and b is to adjust for contrast and offset. We also evaluate the performance using the structural similarity index (SSIM) [53]. We compare five reconstruction methods.

*1) **FBP**.* FBP is the classical direct inversion of the Radon transform $\mathbf{H}$, here implemented in Matlab by the `iradon` command with the `ram-lak` filter and `linear` interpolation as options.

*2) **Total-Variation** Reconstruction.* TV solves

$$\mathbf{x}_{\mathrm{TV}} = \min_{\mathbf{x}} \left(\frac{1}{2} \|\mathbf{H}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_{\mathrm{TV}} \right) \text{ s.t. } \mathbf{x} \geq 0, \tag{17}$$

where

$$\|\mathbf{x}\|_{\mathrm{TV}} = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \sqrt{(\mathrm{D}_{\mathrm{h};i,j}(\mathbf{x}))^2 + (\mathrm{D}_{\mathrm{v};i,j}(\mathbf{x}))^2},$$

$\mathrm{D}_{\mathrm{h};i,j}(\mathbf{x}) = [\mathbf{x}]_{i,j+1} - [\mathbf{x}]_{i,j}$, and $\mathrm{D}_{\mathrm{v};i,j}(\mathbf{x}) = [\mathbf{x}]_{i,j+1} - [\mathbf{x}]_{i,j}$. The optimization is carried out via ADMM [14].

*3) **Dictionary Learning (DL)**.* DL [35] solves

$$\mathbf{x}_{\mathrm{DL}} = \arg\min_{\mathbf{x},\alpha} \left(\|\mathbf{H}\mathbf{x} - \mathbf{y}\|^2 + \lambda \sum_{j=1}^{J} \|\mathbf{E}_j\mathbf{x} - \mathbf{D}\alpha_j\|^2 + \lambda \nu_j \|\alpha_j\|_0 \right), \tag{18}$$

where $\mathbf{E}_j : \mathbb{R}^{N\times N} \to \mathbb{R}^{L^2}$ extracts and vectorizes the jth patch of size $(L \times L)$ from the image $\mathbf{x}$, $\mathbf{D} \in \mathbb{R}^{L^2\times 256}$ is the dictionary, α_j is the jth column of $\alpha \in \mathbb{R}^{256\times R}$, and $R = (N-L+1)^2$. Note that the patches are extracted with a sliding distance of one pixel.

For a given $\mathbf{y}$, the dictionary $\mathbf{D}$ is learned from the corresponding ground truth using the procedure described in [54]. The objective (18) is then solved iteratively by first minimizing it with respect to $\mathbf{x}$ using gradient descent as described in [35] and then with respect to α using orthogonal matching pursuit (OMP) [55]. Since $\mathbf{D}$ is learned from the testing ground truth itself, the performance that we report here is an upper bound to the one that would be achieved by learning it using the training images.

*4) **FBPconv**.* FBPconv [16] is a state-of-the-art deep-learning technique, in which a residual CNN with U-net architecture is trained to directly denoise the FBP . It has been shown to outperform other deep-learning-based direct reconstruction methods for sparse-view CT. In our proposed method, we use a CNN with the same architecture as in FBPconv. As a result, in our framework, FBPconv corresponds to training with only the ensemble in (13). In the testing phase, the FBP of the measurements is fed into the trained CNN to output the reconstruction image.

*5) **RPGD**.* RPGD is our proposed method. It is described in Algorithm 1. There the nonlinear operator F is the CNN trained as a projector (as discussed in Section IV). For experiments with Poisson noise, we use the slightly modified RPGD described in Appendix B. For all the experiments, FBP is used for the operator $\mathbf{A}$.

D. Training and Selection of Parameters

1) Experiment 1: For TV, the regularization parameter λ is selected via a golden-section search over 20 values so as to maximize the SNR of $\mathbf{x}_{\mathrm{TV}}$ with respect to the ground truth. We set the additional penalty parameter inside ADMM (see [14, eq. (2.6)]) equal to λ. The rationale for this heuristic is that it puts the soft-threshold parameter in the same order of magnitude as the image gradients. We set the number of iterations to 100, which was enough to show good empirical convergence.

For DL, the parameters are selected via a parameter sweep, roughly following the approach described in [35, Table 1]. Specifically: The patch size is $L = 8$.

During dictionary learning, the sparsity level is set to 5 and 10. During reconstruction, the sparsity level for OMP is set to 5, 8, 10, 12, 20, and 25, while the tolerance level is taken to be 10, 100, and 1000. This, in effect, is the same as sweeping over ν_j in (18). For each of these $2 \times 6 \times 3 = 36$ parameter settings, λ in (18) is chosen by a golden-section search over 7 values.

As discussed earlier, the CNNs for both the $\times 5$ and $\times 16$ cases are trained separately for high and low measurement noise.

a) Training with noiseless measurements: The training of the projector for RPGD follows the sequential procedure described in Section IV, with the configurations

- $\times 5$, no noise: $T_1 = 80$, $T_2 = 49$, $T_3 = 5$;
- $\times 16$, no noise: $T_1 = 71$, $T_2 = 41$, $T_3 = 11$.

We use the CNN obtained right after the first stage for FBPconv, since during this stage, only the training ensemble in (13) is taken into account. We empirically found that the training error J_2 converged in T_1 epochs of Stage 1, yielding an optimal performance for FBPconv.

b) Training with 40-dB measurement noise: This includes replacing the ensemble in (13) with $\{\mathbf{A}\mathbf{y}^q\}$ where $\mathbf{y}^q = \mathbf{H}\mathbf{x}^q + \mathbf{n}$, has a 40-dB SNR with respect to $\mathbf{H}\mathbf{x}^q$. With 20% probability, we also perturb the views of the measurements with an AWGN of 0.05 standard deviation so as to enforce robustness to model mismatch. These CNNs are initialized with the ones obtained after the first stage of the noiseless training and are then trained with the configurations

- $\times 5$, 40-dB noise: $T_1 = 35$, $T_2 = 49$, $T_3 = 5$;
- $\times 16$, 40-dB noise: $T_1 = 32$, $T_2 = 41$, $T_3 = 11$.

Similarly to the previous case, the CNNs obtained after the first and the third training stage are used in FBPconv and RPGD, respectively. For clarity, these variants will be referred to as **FBPconv40** and **RPGD40**.

The learning rate is decreased in a geometric progression from 10^{-2} to 10^{-3} in Stage 1 and kept at 10^{-3} for Stages 2 and 3. Recall that the last two stages contain the ensemble with dynamic perturbation (14) which changes in every epoch. The lower learning rate, therefore, avoids drastic changes in parameters between the epochs. The batch size is fixed to 2. The

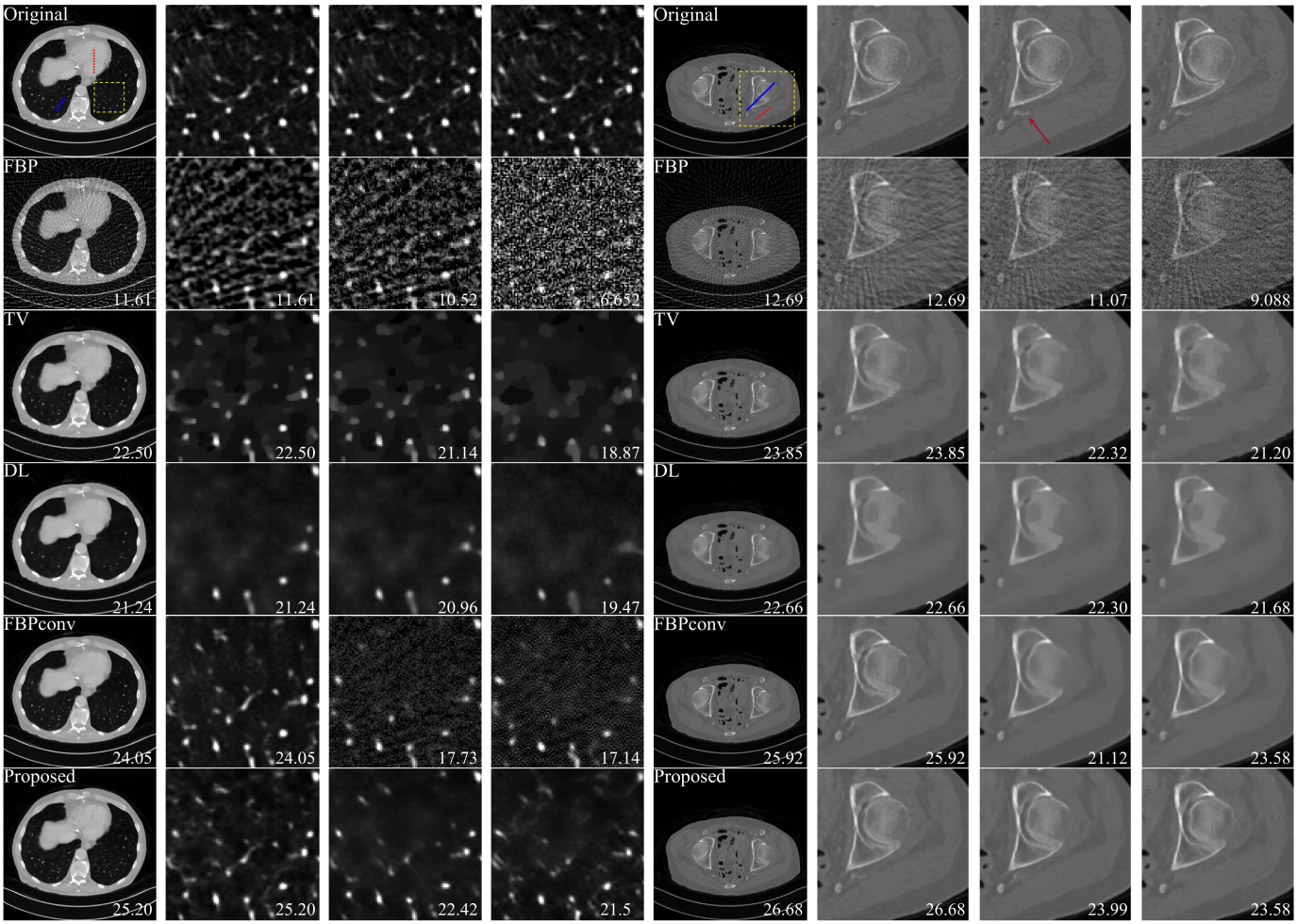

Fig. 2. Comparison of reconstructions using different methods for the $\times 16$ case in Experiment 1. First column: reconstruction from noiseless measurements of a lung image. Second column: zoomed version of the area marked by the box in the original in the first column. Third and fourth columns: zoomed version for the case of 45 and 35 dB, respectively. Fifth to eighth columns: corresponding results for an abdomen image. Seventh and eighth column correspond to 45 and 40 dB, respectively. (a) Results(∞-dB). (b) zoom(∞-dB). (c) zoom(45-dB). (d) zoom(35-dB). (e) Results(∞-dB). (f) zoom(∞-dB). (g) zoom(45-dB). (h) zoom(40-dB).

TABLE I

RECONSTRUCTION RESULTS FOR EXPERIMENT 1 WITH LOW MEASUREMENT NOISE (GAUSSIAN). GRAY CELLS INDICATE THAT THE METHOD WAS TUNED/TRAINED FOR THE CORRESPONDING NOISE LEVEL

Case	Measurement SNR (dB)	Quality Index	Method				
			FBP	TV	DL	FBPconv	RPGD
$\times 16$	∞	SNR	12.74	24.21	23.11	26.19	**27.02**
		SSIM	0.178	0.277	0.231	0.323	**0.374**
	70	SNR	12.73	24.20	23.10	26.18	**26.94**
		SSIM	0.178	0.277	0.231	0.324	**0.325**
$\times 5$	∞	SNR	24.19	30.80	29.36	32.09	**32.62**
		SSIM	0.434	0.511	0.424	0.480	**0.554**
	70	SNR	24.15	30.74	29.24	32.08	**32.56**
		SSIM	0.432	0.507	0.422	0.483	**0.553**

other hyper-parameters follow [16]. For stability, gradients above 10^{-2} are clipped and the momentum is set to 0.99. The total training time for the noiseless case is around 21.5 hours on a Titan X GPU (Pascal architecture).

The hyper-parameters for RPGD are chosen as follows: The relaxation parameter α_0 is initialized with 1, the sequence $\{c_k\}$ is set to the constant $C = 0.99$ for RPGD and $C = 0.8$ for RPGD40. For each noise level and views number, the only free parameter γ is swept over 20 values geometrically spaced between 10^{-2} and 10^{-5}. We pick the γ which gives the best average SNR over the 25 test images. Note that, for TV and DL, the value of the optimum λ generally increases as

TABLE II

RECONSTRUCTION RESULTS FOR EXPERIMENT 1 WITH HIGH MEASUREMENT NOISE (GAUSSIAN). GRAY CELLS INDICATE THAT THE METHOD WAS TUNED/TRAINED FOR THE CORRESPONDING NOISE LEVEL

Case	Measurement SNR (dB)	Quality Index	Method				
			FBP	TV	DL	FBPconv40	RPGD40
×16	40+5	SNR	11.08	22.59	22.74	20.87	**24.16**
		SSIM	0.127	0.238	0.222	0.161	**0.262**
	40	SNR	9.09	21.40	22.13	23.26	**23.73**
		SSIM	0.096	0.210	0.209	0.205	**0.252**
	40-5	SNR	6.51	20.01	20.93	16.20	**22.59**
		SSIM	0.066	0.179	0.187	0.128	**0.221**
×5	40+5	SNR	18.85	27.18	**27.82**	22.56	27.17
		SSIM	0.241	0.367	0.364	0.201	**0.384**
	40	SNR	14.96	25.46	26.26	**28.24**	27.61
		SSIM	0.167	0.314	0.315	0.324	**0.361**
	40-5	SNR	10.76	23.44	22.24	18.90	**24.58**
		SSIM	0.110	0.261	0.263	0.193	**0.300**

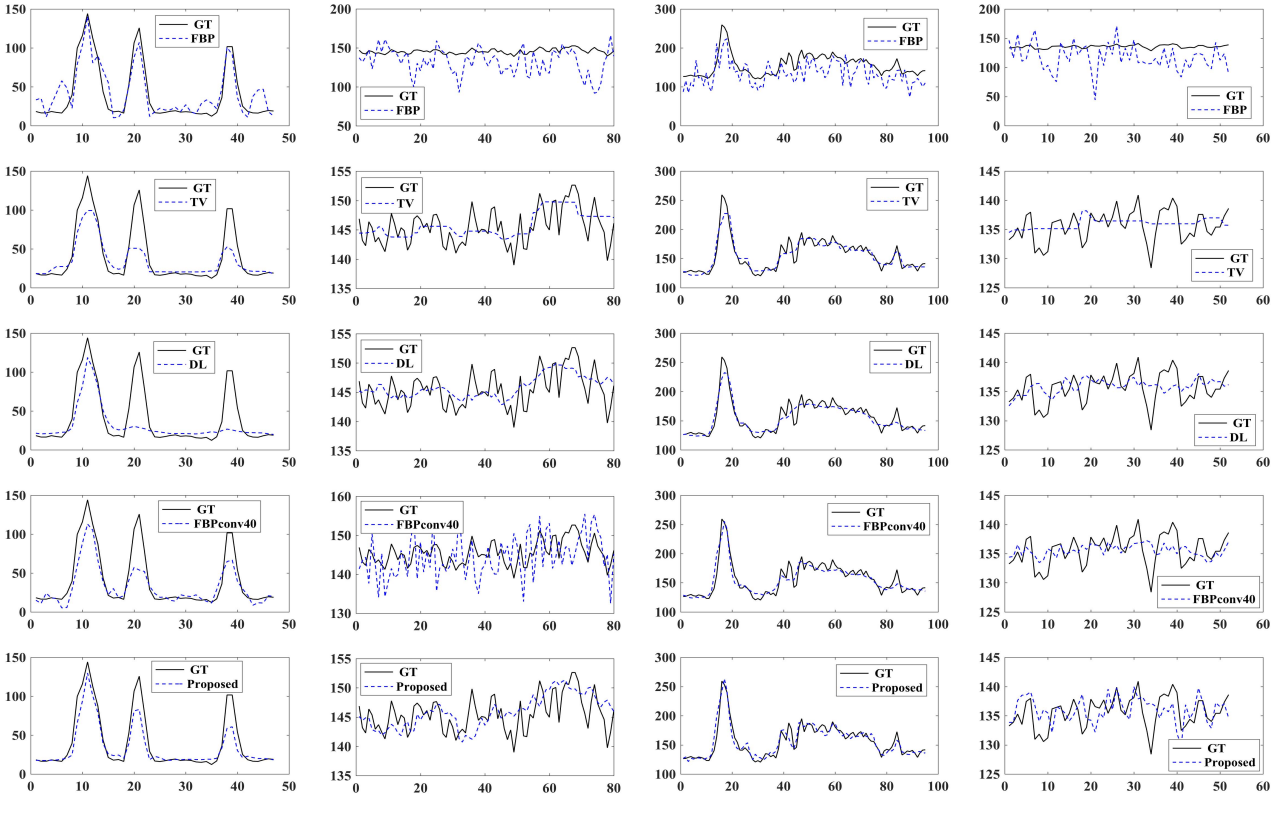

Fig. 3. Profile of the high- and low-contrast regions marked in the first and fifth columns of Figure 2 by solid and dashed line segments, respectively. First and second columns: ×16, 45-dB noise case for the lung image. Third and fourth columns: ×16, 40-dB noise case for the abdomen image. (a) High-contrast profile. (b) Low-contrast profile. (c) High-contrast profile. (d) Low-contrast profile.

the measurement noise increases; however, no such obvious relation exists for γ. This is mainly because it is the step size of the gradient descent in RPGD and not a regularization parameter. In all experiments, the gradient step is skipped during the first iteration.

On the GPU, one iteration of RPGD takes less than 1 second. The algorithm is stopped when the residual $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2$ reaches a value less than 1, which is sufficiently small compared to the dynamic range [0,350] of the image. It takes around 1-2 minutes to reconstruct an image with RPGD.

2) Experiment 2: For this case the CNNs are trained similarly to the CNN for RPGD40 in Experiment 1. Perturbations (12)-(14) are used with the replacement of $\mathbf{AHx}^q_{\text{FD}}$ in (13) by $\mathbf{Ay}^q$, where $\mathbf{y}^q$ had 30 dB Poisson noise. The $\mathbf{x}^q_{\text{FD}}$ and $\mathbf{Ay}^q_{\text{Real}}$ are multiplied with a constant so that their maximum pixel value is 480.

The CNN obtained after the first stage is used as FBPconv. While testing, we keep $C = 0.4$. Other training hyperparameters and testing parameters of the RPGD are kept the same as the RPGD40 for ×5 case in Experiment 1.

3) Experiment 3: The CNNs are trained using the perturbations (12)-(14) with two modifications: (i) $\mathbf{x}^q$ is replaced with $\mathbf{x}^q_{\text{FD}}$ because the actual ground truth was unavailable; and (ii) $\mathbf{AHx}^q$ in (13) is replaced with $\mathbf{Ay}^q_{\text{Real}}$ because we have now access to the actual sinogram.

All other training hyper-parameters and testing parameters are kept the same as RPGD for the $\times 5$ case in Experiment 1. Similar to Experiment 1, the CNN obtained after the first stage of the sequential training is used as the FBPconv.

VI. Results and Discussions

A. Experiment 1

We report in Tables I and II the results for low and high measurement noise, respectively. FBPconv and RPGD are used for low noise, while FBPconv40 and RPGD40 are used for high noise. The reconstruction SNRs and SSIMs are averaged over the 25 test images. The gray cells indicate that the method was optimized for that level of noise. As discussed earlier, adjusting λ for TV and DL indirectly implies tuning for the measurement noise; therefore, all of the cells in these columns are gray. This is different for the learning methods, where tuning for the measurement noise requires retraining.

1) Low Measurement Noise: In the low-noise cases (Table I), the proposed RPGD method outperforms all the others for both $\times 5$ and $\times 16$ reductions in terms of SNR and SSIM indices. FBP performs the worst but is able to retain enough information to be utilized by FBPConv and RPGD. Due to the convexity of the iterative scheme, TV is able to perform well but tends to smooth textures and edges. DL performs worse than TV for $\times 16$ case but is equivalent to it for $\times 5$ case. On one hand, FBPConv outperforms both TV and DL. but it is surpassed by RPGD. This is mainly due to the feedback mechanism in RPGD which lets RPGD use the information in the given measurements to increase the quality of the reconstruction. In fact, for the $\times 16$, no noise, case, the SNRs of the sinogram of the reconstructed images for TV, FBPconv, and RPGD are around 47 dB, 57 dB, and 62 dB, respectively. This means that reconstruction using RPGD has both better image quality and more reliability since it is consistent with the given noiseless measurement.

2) High Measurement Noise: In the noisier cases (Table II), RPGD40 yields a better SNR than other methods in the low-view cases ($\times 16$) and is more consistent in performance than the others in the high-view ($\times 5$) cases. In terms of the SSIM index, it outperforms all of them. The performance of DL and TV are robust to the noise level with DL performing better than others in terms of SNR for the 45-dB, $\times 5$, case. FBPconv40 substantially outperforms DL and TV in the two scenarios with 40-dB noise measurement, over which it was actually trained. For this noise level and $\times 5$ case, it even performs slightly better than RPGD40 but only in terms of SNR. However, as the level of noise deviates from 40 dB, the performance of FBPconv40 degrades significantly. Surprisingly, its performances in the 45-dB cases are much worse than those in the corresponding 40-dB cases. In fact, its SSIM index for the 45-dB, $\times 5$, case is even worse than FBP. This implies that FBPConv40 is highly sensitive to the difference between the training and testing conditions. By contrast, RPGD40 is more robust to this difference due to its iterative correction. In the $\times 16$ case with 45-dB and 35-dB noise level, it outperforms FBPconv40 by around 3.5 dB and 6 dB, respectively.

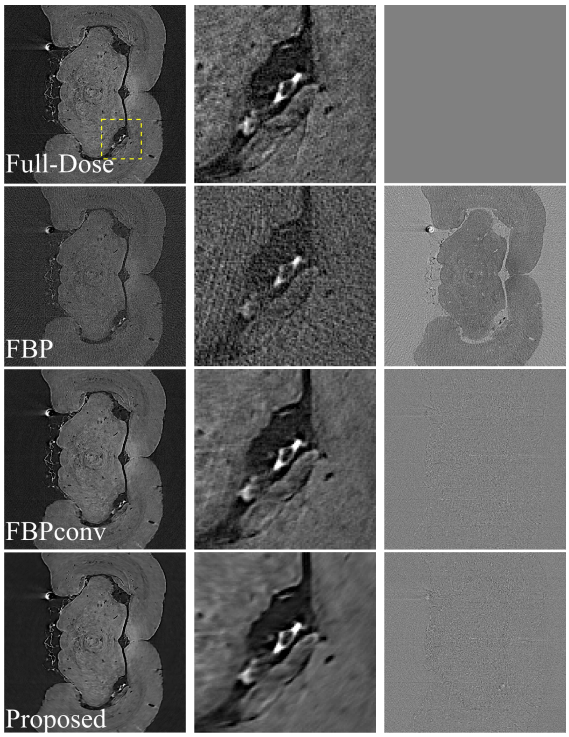

Fig. 4. Reconstruction results for a test slice in Experiment 3. Full-dose image is obtained by taking FBP of the full-view sinogram. The rest of the reconstructions are obtained from the sparse-view ($\times 5$) sinogram. The last column shows the difference between the reconstruction and the full-dose image. (a) Results (∞-dB). (b) zoom (∞-dB). (c) diff (∞-dB).

3) Case Study: The reconstructions of lung and abdomen images for the case of $\times 16$ downsampling and noiseless measurements are illustrated in Figure 2 (first and fifth columns). FBP is dominated by line artifacts, while TV and DL satisfactorily removes those but blurs the fine structures. FBPConv and RPGD are able to reconstruct these details. The zoomed version (second and sixth columns) suggests that RPGD is able to reconstruct the fine details better than the other methods. This observation remains the same when the measurement quality degrades. The remaining columns, contain the reconstructions for different noise levels. For the abdomen image it is noticeable that only TV is able to retain the small bone structure marked by an arrow in the zoomed version of the lung image (seventh column). Possible reason for this could be that the structure similar to this were rare in the training set. Increasing the training data size with suitable images could be a solution.

Figure 3 contains the profiles of high- and low-contrast regions of the reconstructions for the two images. These regions are marked by line segments inside the original image in the first column of Figure 2. The FBP profile is highly noisy and the TV and DL profiles overly smooth the details. FBPconv40 is able to accommodate the sudden transitions

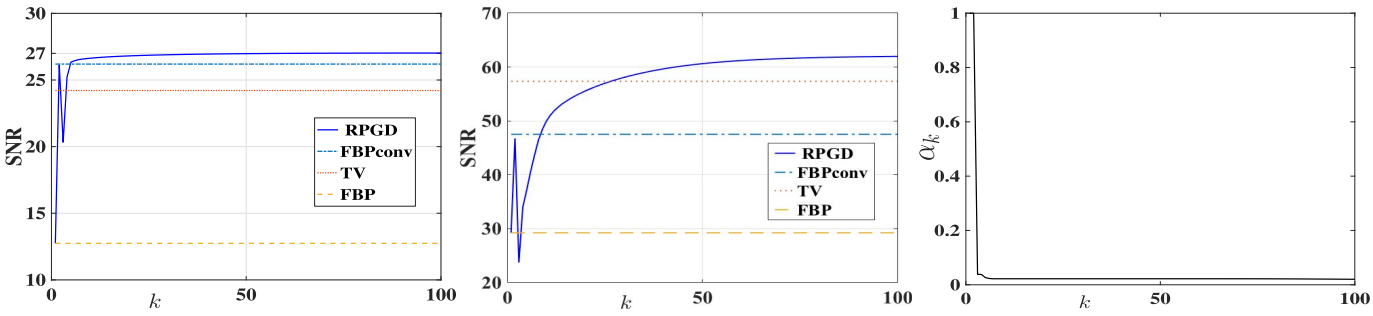

Fig. 5. Convergence with iteration k of RPGD for the Experiment 1, $\times 16$, no-noise case when $C = 0.99$. Results are averaged over 25 test images. (a) SNRs of $\mathrm{x_k}$ with respect to the ground-truth image. (b) SNRs of $\mathrm{Hx_k}$ with respect to the ground-truth sinogram. (c) Evolution of the relaxation parameters α_k. In (a) and (b), the FBP, FBPconv, and TV results are independent of the RPGD iteration k but have been shown for the sake of comparison.

TABLE III
RECONSTRUCTION RESULTS FOR EXPERIMENT 2 WITH POISSON NOISE AND $\times 5$ VIEWS REDUCTION. GREY CELL INDICATE THAT THE METHOD WAS TRAINED FOR THE CORRESPONDING NOISE LEVEL

Measurement SNR (dB)	Quality Index	Method		
		FBP	FBPconv	RPGD
30-5	SNR	4.61	7.45	**8.21**
	SSIM	0.112	0.134	**0.154**
30	SNR	5.96	9.18	**9.22**
	SSIM	0.200	0.174	**0.246**
30+5	SNR	7.75	9.50	**9.75**
	SSIM	0.305	0.132	**0.332**

in the high-contrast case. RPGD40 is slightly better in this regard. For the low-contrast case, RPGD40 is able to follow the structures of the original (GT) profile better than the others. A similar analysis holds for the $\times 5$ case (Figure 7, supplementary material).

B. Experiment 2

We show in Table III the regressed SNR and SSIM indices averaged over the 25 reconstructed slices. RPGD outperforms both FBP and FBPconv in terms of SNR and SSIM. Similar to the Experiment 1, its performance is also more robust with respect to noise mismatch. Fig. 9 in the supplementary material compares the reconstructions for a given test slice.

C. Experiment 3

In Figure 4, we show the reconstruction result for one slice for $\gamma = 10^{-5}$. Since the ground truth is unavailable, we show the reconstructions without a quantitative comparison. It can be seen that the proposed method is able to reconstruct images with reasonable perceptual quality.

VII. BEHAVIOR OF ALGORITHMS

We now explore the behavior of the proposed method in more details, including its empirical convergence and the effect of sequential training.

A. Convergence of RPGD

In Figure 5, we show the behavior of RPGD with respect to the iteration number k for Experiment 1. The evolution of the SNR of images $\mathbf{x}_k$ and their measurements $\mathbf{H}\mathbf{x}_k$ computed with respect to the ground truth image and the ground-truth measurement are shown in Figures 5 (a) and (b), respectively. We give α_k with respect to the iteration k in Figure 5 (c). The results are averaged over 25 test images for $\times 16$, no noise, case and $C = 0.99$. RPGD outperforms all the other methods in the context of both image quality and measurement consistency.

Due to the high value of the step size ($\gamma = 2 \times 10^{-3}$) and the large difference ($\mathbf{H}\mathbf{x}_k - \mathbf{y}$), the initial few iterations have large gradients and result in the instability of the algorithm. The reason is that the CNN is fed with ($\mathbf{x}_k - \gamma \mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{x}_k - \mathbf{y})$), which is drastically different from the perturbations on which it was trained. In this situation, α_k decreases steeply and stabilizes the algorithm. At convergence, $\alpha_k \neq 0$; therefore, according to Theorem 3, $\mathbf{x}_{100}$ is the fixed point of (9) where $F = CNN$.

B. Advantages of Sequential Training

Here, we experimentally verify the advantages of the sequential-training strategy discussed in Section V. Using the setup of Experiment 1, we compare the training time and performance of the CNNs trained with and without this strategy for the $\times 16$ downsampling and no noise case. For the gold standard (systematic training of CNN), we train a CNN as a projector with the 3 types of perturbation in every epoch. We use 135 epochs for training which is roughly equal to $\{T_1 + T_2 + T_3\}$ used during training for the corresponding sequential-training-based CNN. This number was sufficient for the convergence of the training error. The reconstruction performance of RPGD using this gold standard CNN is 26.86 dB, compared to 27.02 dB for RPGD using the sequentially trained CNN. The total training times are 48 and 22 hours, respectively. This demonstrates that the sequential strategy reduces the training time (in this case more than 50%), while preserving (or even slightly increasing) the reconstruction performance.

VIII. Conclusion

We have proposed a simple yet effective iterative scheme (RPGD) where one step of enforcing measurement consistency is followed by a CNN that tries to project the solution onto the set of desired reconstruction images. The whole scheme is ensured to be convergent. We also introduced a novel method to train a CNN that acts like a projector using a reasonably small dataset (475 images). For sparse-view CT reconstruction, our method outperforms the previous techniques for both noiseless and noisy measurements.

The proposed framework is generic and can be used to solve a variety of inverse problems including superresolution, deconvolution, accelerated MRI, *etc*. This can bring more robustness and reliability to the current deep-learning-based techniques.

Appendix

A. Proof of Theorem 3

(i) Set $\mathbf{r}_k = (\mathbf{x}_{k+1} - \mathbf{x}_k)$. On one hand, it is clear that

$$\mathbf{r}_k = (1-\alpha_k)\mathbf{x}_k + \alpha_k \mathbf{z}_k - \mathbf{x}_k = \alpha_k (\mathbf{z}_k - \mathbf{x}_k). \tag{19}$$

On the other hand, from the construction of $\{\alpha_k\}$,

$$\begin{aligned} \alpha_k \|\mathbf{z}_k - \mathbf{x}_k\|_2 &\leq c_k \alpha_{k-1} \|\mathbf{z}_{k-1} - \mathbf{x}_{k-1}\|_2 \\ \Leftrightarrow \|\mathbf{r}_k\|_2 &\leq c_k \|\mathbf{r}_{k-1}\|_2 . \end{aligned} \tag{20}$$

Iterating (20) gives

$$\|\mathbf{r}_k\|_2 \leq \|\mathbf{r}_0\|_2 \prod_{i=1}^{k} c_i, \quad \forall k \geq 1. \tag{21}$$

We now show that $\{\mathbf{x}_k\}$ is a Cauchy sequence. Since $\{c_k\}$ is asymptotically upper-bounded by $C < 1$, there exists K such that $c_k \leq C, \forall k > K$. Let m, n be two integers such that $m > n > K$. By using (21) and the triangle inequality,

$$\begin{aligned} \|\mathbf{x}_m - \mathbf{x}_n\|_2 &\leq \sum_{k=n}^{m-1} \|\mathbf{r}_k\|_2 \leq \|\mathbf{r}_0\|_2 \prod_{i=1}^{K} c_i \sum_{k=n-K}^{m-1-K} C^k \\ &\leq \left(\|\mathbf{r}_0\|_2 \prod_{i=1}^{K} c_i \right) \frac{C^{n-K} - C^{m-K}}{1-C}. \end{aligned} \tag{22}$$

The last inequality proves that $\|\mathbf{x}_m - \mathbf{x}_n\|_2 \to 0$ as $m \to \infty, n \to \infty$, or $\{\mathbf{x}_k\}$ is a Cauchy sequence in the complete metric space $\mathbb{R}^N$. As a consequence, $\{\mathbf{x}_k\}$ must converge to some point $\mathbf{x}^* \in \mathbb{R}^N$.

(ii) Assume from now on that $\{\alpha_k\}$ is lower-bounded by $\varepsilon > 0$. By definition, $\{\alpha_k\}$ is also non-increasing and, thus, convergent to $\alpha^* > 0$. Next, we rewrite the update of $\mathbf{x}_k$ in Algorithm 1 as

$$\mathbf{x}_{k+1} = (1-\alpha_k)\mathbf{x}_k + \alpha_k G_\gamma(\mathbf{x}_k), \tag{23}$$

where G_γ is defined by (9). Taking the limit of both sides of (23) leads to

$$\mathbf{x}^* = (1-\alpha^*)\mathbf{x}^* + \alpha^* \lim_{k\to\infty} G_\gamma(\mathbf{x}_k). \tag{24}$$

Moreover, since the nonlinear operator F is continuous, G_γ is also continuous. Hence,

$$\lim_{k\to\infty} G_\gamma(\mathbf{x}_k) = G_\gamma\left(\lim_{k\to\infty} \mathbf{x}_k\right) = G_\gamma(\mathbf{x}^*). \tag{25}$$

By plugging (25) into (24), we get that $\mathbf{x}^* = G_\gamma(\mathbf{x}^*)$, which means that $\mathbf{x}^*$ is a fixed point of the operator G_γ.

(iii) Now that $F = P_{\mathcal{S}}$ satisfies (6), we invoke Proposition 1 to infer that $\mathbf{x}^*$ is a local minimizer of (3), thus completing the proof.

B. RPGD for Poisson Noise in CT

In the case where the CT measurements are corrupted by Poisson noise, the data-fidelity term in (3) should be replaced by weighted least squares [35], [56], [57]. For the sake of completeness, we show a sketch of the derivation. Let $\mathbf{x}$ represent the distribution of linear attenuation coefficient of an object and $[\mathbf{Hx}]_m$ represents their line integral. The mth CT measurement, y_m, is a Poisson random variable with parameters

$$p_m \sim \text{Poisson}\left(b_m e^{-[\mathbf{Hx}]_m} + r_m\right) \tag{26}$$

$$y_m = -\log\left(\frac{p_m}{b_m}\right) \tag{27}$$

where b_m is the blank scan factor and r_m is the readout noise. Since logarithm is bijective, the negative log-likelihood of $\mathbf{y}$ given $\mathbf{x}$ is equal to the one of $\mathbf{p}$ given $\mathbf{x}$. After removing the constants, we use this negative log-likelihood as the data-fidelity term

$$E(\mathbf{Hx}, \mathbf{y}) = \sum_{m=1}^{M} \left(\hat{p}_m - p_m \log \hat{p}_m\right), \tag{28}$$

where $\hat{p}_m = b_m e^{-[\mathbf{Hx}]_m} + r_m$ is the expected value of p_m. We then perform a quadratic approximation of E with respect to $\mathbf{Hx}$ around the point $(-\ln(\frac{\hat{p}_m - r_m}{b_m}))$ using a Taylor expansion. After ignoring the higher-order terms, this yields

$$E(\mathbf{Hx}, \mathbf{y}) = \sum_{m=1}^{M} \frac{w_m}{2} \left(\mathbf{Hx} - \log\left(\frac{b_m}{p_m - r_m}\right)\right)^2, \tag{29}$$

where $w_m = \frac{(p_m - r_m)^2}{p_m}$.

In the case when the readout noise r_m is insignificant, (29) can be written as

$$E(\mathbf{Hx}, \mathbf{y}) = \sum_{m=1}^{M} \frac{w_m}{2} ([\mathbf{Hx}]_m - y_m)^2 \tag{30}$$

$$= \frac{1}{2} \|\mathbf{W}^{\frac{1}{2}}\mathbf{Hx} - \mathbf{W}^{\frac{1}{2}}\mathbf{y}\|^2 \tag{31}$$

$$= \frac{1}{2} \|\mathbf{H}'\mathbf{x} - \mathbf{y}'\|^2, \tag{32}$$

where $\mathbf{W} \in \mathbb{R}^{M\times M}$ is a diagonal matrix with $[\mathbf{diag}(\mathbf{W})]_m = w_m = p_m$, $\mathbf{H}' = \mathbf{W}^{\frac{1}{2}}\mathbf{H}$, and $\mathbf{y}' = \mathbf{W}^{\frac{1}{2}}\mathbf{y}$.

Imposing the data manifold prior, we get the equivalent of Problem (3) as

$$\min_{\mathbf{x}\in\mathcal{S}} \frac{1}{2} \|\mathbf{H}'\mathbf{x} - \mathbf{y}'\|^2. \tag{33}$$

Note that all the results discussed in Section II and III apply to Problem (33). As a consequence, we use Algorithm 1 to solve the problem with the following small change in the gradient step:

$$\mathbf{z}_k = F(\mathbf{x}_k - \gamma\, \mathbf{H}'^T \mathbf{H}' \mathbf{x}_k + \gamma\, \mathbf{H}'^T \mathbf{y}'). \tag{34}$$

ACKNOWLEDGMENT

The authors thank Emmanuel Soubies for his helpful suggestions on training the CNN and Dr. Cynthia McCollough, the Mayo Clinic, the American Association of Physicists in Medicine, and the National Institute of Biomedical Imaging and Bioengineering for the Mayo-clinic dataset. They also thank Dr. Marco Stampanoni, Swiss Light Source, Paul Scherrer Institute, Villigen, Switzerland, for the rat-brain dataset. They also thankfully acknowledge the support of the NVIDIA Corporation, in providing the Titan X GPU for this research.

REFERENCES

[1] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[2] A. C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging* (Classics in Applied Mathematics). New York, NY, USA: SIAM, 2001.

[3] X. C. Pan, E. Y. Sidky, and M. Vannier, "Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?" *Inverse Problems*, vol. 25, no. 12, p. 123009, 2009.

[4] C. Bouman and K. Sauer, "A generalized Gaussian image model for edge-preserving MAP estimation," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 296–310, Jul. 1993.

[5] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, Feb. 1997.

[6] E. Candès and J. Romberg, "Sparsity and incoherence in compressive sampling," *Inverse Probl.*, vol. 23, no. 3, pp. 969–985, 2007.

[7] S. Ramani and J. A. Fessler, "Parallel MR image reconstruction using augmented Lagrangian methods," *IEEE Trans. Med. Imag.*, vol. 30, no. 3, pp. 694–706, Mar. 2011.

[8] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[9] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 1, pp. 59–73, Jul. 2011.

[10] S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Efficient sum of outer products dictionary learning (SOUP-DIL) and its application to inverse problems," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 694–709, Dec. 2017.

[11] M. A. T. Figueiredo and R. D. Nowak, "An EM algorithm for wavelet-based image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 906–916, Aug. 2003.

[12] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[13] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[15] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 85–95, Nov. 2017.

[16] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[17] Y. S. Han, J. Yoo, and J. C. Ye. (2017). "Deep learning with domain adaptation for accelerated projection-reconstruction MR." [Online]. Available: https://arxiv.org/abs/1703.01135

[18] S. Antholzer, M. Haltmeier, and J. Schwab. (2017). "Deep learning for photoacoustic tomography from sparse data." [Online]. Available: https://arxiv.org/abs/1704.04587

[19] S. Wang *et al.*, "Accelerating magnetic resonance imaging via deep learning," in *Proc. IEEE Int. Symp. Biomed. Imag. (ISBI)*, Apr. 2016, pp. 514–517.

[20] A. Mousavi and R. G. Baraniuk. (2017). "Learning to invert: Signal recovery via deep convolutional networks." [Online]. Available: https://arxiv.org/abs/1701.03891

[21] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 399–406.

[22] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 10–18.

[23] J. Adler and O. Öktem. (2017). "Solving ill-posed inverse problems using iterative deep neural networks." [Online]. Available: https://arxiv.org/abs/1704.04058

[24] P. Putzky and M. Welling. (2017). "Recurrent inference machines for solving inverse problems." [Online]. Available: https://arxiv.org/abs/1706.04008

[25] J. Schlemper, J. Caballero, J. V. Hajnal, A. Price, and D. Rueckert, "A deep cascade of convolutional neural networks for MR image reconstruction," in *Proc. Int. Conf. Inf. Process. Med. Imag.*, 2017, pp. 647–658.

[26] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Dec. 2013, pp. 945–948.

[27] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-play ADMM for image restoration: Fixed-point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, Jan. 2017.

[28] S. Sreehari *et al.*, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, Dec. 2016.

[29] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (RED)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[30] J. H. R. Chang, C.-L. Li, B. Póczos, B. V. K. V. Kumar, and A. C. Sankaranarayanan. (2017). "One network to solve them all—Solving linear inverse problems using deep projection models." [Online]. Available: https://arxiv.org/abs/1703.09912

[31] A. Bora, A. Jalal, E. Price, and A. G. Dimakis. (2017). "Compressed sensing using generative models." [Online]. Available: https://arxiv.org/abs/1703.03208

[32] B. Kelly, T. P. Matthews, and M. A. Anastasio. (2017). "Deep learning-guided image reconstruction from incomplete data." [Online]. Available: https://arxiv.org/abs/1709.00584

[33] J. Z. Liang, P. J. La Riviere, G. El Fakhri, S. J. Glick, and J. Siewerdsen, "Guest editorial low-dose CT: What has been done, and what challenges remain?" *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2409–2416, Dec. 2017.

[34] S. Ramani and J. A. Fessler, "A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction," *IEEE Trans. Med. Imag.*, vol. 31, no. 3, pp. 677–688, Mar. 2012.

[35] Q. Xu, H. Yu, X. Mou, L. Zhang, J. Hsieh, and G. Wang, "Low-dose X-ray CT reconstruction via dictionary learning," *IEEE Trans. Med. Imag.*, vol. 31, no. 9, pp. 1682–1697, Sep. 2012.

[36] S. Niu *et al.*, "Sparse-view X-ray CT reconstruction via total generalized variation regularization," *Phys. Med. Biol.*, vol. 59, no. 12, pp. 2997–3017, 2014.

[37] L. Gjesteby, Q. Yang, Y. Xi, Y. Zhou, J. Zhang, and G. Wang, "Deep learning methods to guide CT image reconstruction and reduce metal artifacts," *Proc. SPIE*, vol. 10132, p. 101322W, Mar. 2017.

[38] H. Chen *et al.*, "Low-dose CT with a residual encoder-decoder convolutional neural network," *IEEE Trans. Image Process.*, vol. 36, no. 12, pp. 2524–2535, Dec. 2017.

[39] E. Kang, J. Min, and J. C. Ye, "A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction," *Med. Phys.*, vol. 44, no. 10, pp. e360–e375, Oct. 2017.

[40] Y. S. Han, J. Yoo, and J. C. Ye. (2016). "Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis." [Online]. Available: https://arxiv.org/abs/1611.06391

[41] B. Eicke, "Iteration methods for convexly constrained ill-posed problems in Hilbert space," *Numer. Funct. Anal. Optim.*, vol. 13, nos. 5–6, pp. 413–429, 1992.
[42] L. Landweber, "An iteration formula for fredholm integral equations of the first kind," *Amer. J. Math.*, vol. 73, no. 3, pp. 615–624, Jul. 1951.
[43] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Cambridge, MA, USA: Athena Scientific, 1999.
[44] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.
[45] P. L. Combettes and J.-C. Pesquet, *Proximal Splitting Methods in Signal Processing*. New York, NY, USA: Springer, 2011, pp. 185–212.
[46] J. Bect, L. Blanc-Féraud, G. Aubert, and A. Chambolle, "A ℓ_1-unified variational framework for image restoration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2004, pp. 1–13.
[47] A. Aldroubi and R. Tessera, "On the existence of optimal unions of subspaces for data modeling and clustering," *Found. Comput. Math.*, vol. 11, no. 3, pp. 363–379, Jun. 2011.
[48] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[49] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image. Comput. Comput. Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
[50] A. E. Orhan and X. Pitkow. (2017). "Skip connections eliminate singularities." [Online]. Available: https://arxiv.org/abs/1701.09175
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[52] C. McCollough, "TU-FG-207A-04: Overview of the low dose CT grand challenge," *Med. Phys.*, vol. 43, no. 6, pp. 3759–3760, 2016.
[53] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, vol. 2. Nov. 2003, pp. 1398–1402.
[54] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.
[55] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[56] K. Sauer and C. Bouman, "A local update strategy for iterative reconstruction from projections," *IEEE Trans. Signal Process.*, vol. 41, no. 2, pp. 534–548, Feb. 1993.
[57] I. A. Elbakri and J. A. Fessler, "Statistical image reconstruction for polyenergetic X-ray computed tomography," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 89–99, Feb. 2002.
[58] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer, 2011.