

Learned Primal-Dual Reconstruction

Jonas Adler^{ID} and Ozan Öktem^{ID}

Abstract—We propose the Learned Primal-Dual algorithm for tomographic reconstruction. The algorithm accounts for a (possibly non-linear) forward operator in a deep neural network by unrolling a proximal primal-dual optimization method, but where the proximal operators have been replaced with convolutional neural networks. The algorithm is trained end-to-end, working directly from raw measured data and it does not depend on any initial reconstruction such as filtered back-projection (FBP). We compare performance of the proposed method on low dose computed tomography reconstruction against FBP, total variation (TV), and deep learning based post-processing of FBP. For the Shepp-Logan phantom we obtain >6 dB peak signal to noise ratio improvement against all compared methods. For human phantoms the corresponding improvement is 6.6 dB over TV and 2.2 dB over learned post-processing along with a substantial improvement in the structural similarity index. Finally, our algorithm involves only ten forward-back-projection computations, making the method feasible for time critical clinical applications.

Index Terms—Inverse problems, tomography, deep learning, primal-dual, optimization.

I. INTRODUCTION

IN AN inverse problem, the goal is to reconstruct parameters characterizing the system under investigation from indirect observations. Such problems arise in several areas of science and engineering, like tomographic imaging where one seeks to visualize the interior structure of an object (2D/3D image) from indirect observations. Imaging technologies of this type, such as computed tomography (CT) and magnetic resonance imaging (MRI) imaging, are indispensable for contemporary medical diagnostics, intervention and monitoring.

There is by now a rich theory for *model driven* tomographic image reconstruction. A key component is knowledge about the forward model, which describes the data formation process in absence of noise. In many applications, an explicit forward model can be derived starting out from the underlying physical

Manuscript received November 24, 2017; revised January 11, 2018; accepted January 18, 2018. Date of publication January 29, 2018; date of current version May 31, 2018. This work was supported in part by the Swedish Foundation of Strategic Research under Grant AM13-0049 and Industrial Ph.D., under Grant ID14-0055, in part by Elekta, and in part by the National Institute of Biomedical Imaging and Bioengineering under Grant EB017095 and Grant EB017185. (Corresponding author: Jonas Adler.)

J. Adler is with the Department of Mathematics, KTH - Royal Institute of Technology, 100 44 Stockholm, Sweden, and also with Elekta Instrument AB, SE-103 93 Stockholm, Sweden (e-mail: jonasadl@kth.se).

O. Öktem is with the Department of Mathematics, KTH - Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: ozan@kth.se).

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.2018.2799231

principles that are utilized by the imaging modality. Another component accounts for the knowledge about the statistical properties of data and a priori information about the image to be recovered.

A parallel line of development in signal processing has been the usage of deep learning for solving a wide range of tasks that can be cast as supervised learning. The success of these *data driven* approaches are this far confined to tasks where knowledge about the forward model is not needed, or has little importance. As an example, an entirely data driven approach for tomographic image reconstruction applicable to clinical sized problems has yet to be demonstrated.

A central question is whether one can combine elements of *model and data driven approaches for solving ill-posed inverse problems*. In particular, is there a framework for incorporating the knowledge of a forward model when designing a neural network for reconstruction? The Learned Primal-Dual reconstruction method developed in this paper is such a framework. It applies to general inverse problems and it is best described in an abstract setting, so our starting point is to formalize the notion of an inverse problem.

Mathematically, an inverse problem can be formulated as reconstructing (estimating) a signal $f_{\text{true}} \in X$ from data $g \in Y$ where

$$g = \mathcal{T}(f_{\text{true}}) + \delta g. \quad (1)$$

Here, the reconstruction space X and the data space Y are typically Hilbert Spaces, $\mathcal{T}: X \rightarrow Y$ is the forward operator that models how a signal gives rise to data in absence of noise, and $\delta g \in Y$ is a single sample of a Y -valued random variable that represents the noise component of data.

A. Variational Regularization

A common model driven approach for solving (1) is to maximize the likelihood of the signal, or equivalently minimizing the negative data log-likelihood [1]:

$$\min_{f \in X} \mathcal{L}(\mathcal{T}(f), g). \quad (2)$$

This minimization is for typical choices of \mathcal{T} ill-posed, that is, a solution (if it exists) is unstable with respect to the data g in the sense that small changes to data results in large changes to a reconstruction. Hence, a maximum likelihood solution typically leads to over-fitting against data.

Variational regularization, also referred to as model based iterative reconstruction in medical image processing, avoids over-fitting by introducing a functional $\mathcal{S}: X \rightarrow \mathbb{R}$ (regularization functional) that encodes a priori information

about the true (unknown) f_{true} and penalizes unlikely solutions [2], [3]. Hence, instead of minimizing only the negative data log-likelihood as in (2), one now seeks to minimize a regularized objective functional by solving

$$\min_{f \in X} [\mathcal{L}(\mathcal{T}(f), g) + \lambda S(f)] \quad \text{for a fixed } \lambda \geq 0. \quad (3)$$

In the above, λ (regularization parameter) governs the influence of the a priori knowledge encoded by the regularization functional against the need to fit data.

B. Machine Learning in Inverse Problems

We here review results on learned iterative schemes, see [4] for a wider review on machine learning for medical imaging and [5] for usage of machine learning for solving inverse problems in general.

Machine learning is widely used for non-linear function approximation under weak assumptions and has recently emerged as the state of the art for several image processing tasks such as classification and segmentation. Applied to the inverse problem in (1), it can be phrased as the problem of finding a (non-linear) mapping $\mathcal{T}_\theta^\dagger: Y \rightarrow X$ satisfying the following *pseudo-inverse* property:

$$\mathcal{T}_\theta^\dagger(g) \approx f_{\text{true}} \quad \text{whenever data } g \text{ is related to } f_{\text{true}} \text{ as in (1).}$$

A key element in machine learning approaches is to parameterize the set of such pseudo-inverse operators by a parameter $\theta \in \Theta$ where Θ is some parameter space and the main algorithmic complication is to select an appropriate structure of $\mathcal{T}_\theta^\dagger$ such that, given appropriate training, the pseudo-inverse property is satisfied as well as possible.

In the context of tomographic reconstruction, three main research directions have been proposed. The first is so called learned post-processing or learned denoisers. Here, the learned reconstruction operator is of the form

$$\mathcal{T}_\theta^\dagger = \Lambda_\theta \circ \mathcal{T}^\dagger$$

where $\Lambda_\theta: X \rightarrow X$ is a learned post-processing operator and $\mathcal{T}^\dagger: Y \rightarrow X$ is some approximate pseudo-inverse, e.g. given by filtered back-projection (FBP) in CT reconstruction. This type of method is relatively easy to implement, given that the pseudo-inverse can be applied off-line, before the learning is performed, which reduces the learning to inferring an $X \rightarrow X$ transformation. This has been investigated in [6]–[8].

Another method is to learn a regularizer and use this regularizer in a classical variational reconstruction scheme according to (3). Examples of this include dictionary learning [9], but several alternative methods have been investigated, such as learning a variational auto-encoder [10] or using a cascade of wavelet transforms (scattering transform) [11].

Finally, some authors investigate learning the full reconstruction operator, going all the way from data to reconstruction. Doing this in one step is typically very computationally expensive and does not scale to the data sizes encountered in tomographic reconstruction. Instead, *learned iterative schemes* have been studied. These schemes resemble classical optimization methods used for tomographic reconstruction but use

machine learning to find the best update in each iteration given the last iterate and results of applying the forward operator and its adjoint as input.

One of the first works on learned iterative schemes is [12], which learns an ADMM-like scheme for MRI reconstruction. A further development along this lines is given in [13], which learns over a broader class of schemes instead of ADMM-type of schemes. The application is to finite dimensional inverse problems typically arising in image restoration. This approach was in [5] further extended to non-linear forward operators in to the infinite dimensional setting, which also applies learned iterative schemes to (non-linear, pre-log) CT. Similar approaches for MRI reconstruction have also been considered [14], [15]. Here, the situation is simpler than CT reconstruction since the forward operator is approximated by a Fourier transform, i.e. MRI reconstruction amounts to inverting the Fourier transform.

Given a structure of $\mathcal{T}_\theta^\dagger$, the “learning” part refers to choosing an “optimal” set of parameters θ given some *training data*, where the concept of optimality is typically quantified through a loss functional that measures the quality of a learned pseudo-inverse $\mathcal{T}_\theta^\dagger$.

To define this loss functional, consider a $(X \times Y)$ -valued random variable (\mathbf{f}, \mathbf{g}) with joint probability distribution μ . This could be e.g. the probability distribution of human bodies and corresponding noisy tomographic data. We define the optimal reconstruction operator as the one whose reconstructions have the lowest average mean squared distance to the true reconstructions, where the average is taken w.r.t. μ . Finding this reconstruction operator is then given by selecting the parameters $\theta \in \Theta$ so that the loss functional $L(\theta)$ is minimized:

$$L(\theta) := \mathbb{E}_{(\mathbf{f}, \mathbf{g}) \sim \mu} [\|\mathcal{T}_\theta^\dagger(\mathbf{g}) - \mathbf{f}\|_X^2]. \quad (4)$$

However, in practice we often do not have access to the probability distribution μ of the random variable (\mathbf{f}, \mathbf{g}) . Instead, we know a finite set of samples $(g_1, f_1), \dots, (g_N, f_N)$. In this setting, we replace μ in (4) with its empirical counterpart, so the loss function is replaced with the empirical loss

$$\widehat{L}(\theta) := \frac{1}{N} \sum_{i=1}^N [\|\mathcal{T}_\theta^\dagger(g_i) - f_i\|_X^2]. \quad (5)$$

Since our main goal is to minimize the loss functional, our practical goal is thus two-fold: we want to find a learned reconstruction scheme that minimizes the empirical loss, and we also want it to generalize to new, unseen data.

II. CONTRIBUTION AND OVERVIEW OF PAPER

This paper proposes the *Learned Primal-Dual* algorithm, a general framework for solving inverse problems that combines deep learning with model based reconstruction. The proposed learned iterative reconstruction scheme involves convolutional neural networks (CNNs) in both the reconstruction and data space, and these are connected by the forward operator and its adjoint. We train the networks to minimize the mean squared error of the reconstruction and demonstrate

that this achieves very high performance in CT reconstruction, surpassing recent learning based methods on both analytical and human data.

We emphasize that we learn the whole reconstruction operator, mapping data to reconstruction, and not just a post-processing nor only the prior in isolation.

In addition, we make all of our code and learned parameters open source so that the community can reproduce the results and apply the methods to other inverse problems [16].

III. THE LEARNED PRIMAL-DUAL ALGORITHM

We here introduce how primal-dual algorithms can be learned from data and how this can be used to solve inverse problems.

A. Primal-Dual Optimization Schemes

In imaging, the minimization in (3) is a large scale optimization problem, which traditionally has been addressed using gradient based methods such as gradient descent or its extensions to higher order derivatives, e.g. quasi-Newton or Newton methods. However, many regularizers of interest result in a non-differentiable objective functional, so gradient based methods are not applicable. A common approach to handle this difficulty is to consider a smooth approximation, which however introduces additional parameters and gives non-exact solutions.

An alternative approach is to use methods from non-smooth convex optimization. *Proximal* methods have been developed in order to directly work with non-smooth objective functionals. Here, a proximal step replaces the gradient step. The simplest example of such an algorithm is the proximal point algorithm for minimizing an objective functional $\mathcal{G} : X \rightarrow \mathbb{R}$. It can be seen as the proximal equivalent of the gradient descent scheme and is given by

$$f_{i+1} = \text{prox}_{\tau\mathcal{G}}(f_i) \quad (6)$$

where $\tau \in \mathbb{R}^+$ is a step size and the *proximal operator* is defined by

$$\text{prox}_{\tau\mathcal{G}}(f) = \arg \min_{f' \in X} \left[\mathcal{G}(f') + \frac{1}{2\tau} \|f' - f\|_X^2 \right] \quad (7)$$

While this algorithm could, in theory, be applied to solve (3) it is rarely used directly since (7) does not have a closed form solution. Proximal primal-dual schemes offer a work around. In these schemes, an auxiliary *dual* variable in the range of the operator is introduced and the *primal* ($f \in X$) and dual variables are updated in an alternating manner.

One well known primal-dual scheme is the primal dual hybrid gradient (PDHG) algorithm [17], also known as the Chambolle-Pock algorithm, with a recent extension to non-linear operators [18]. The scheme (algorithm 1) is adapted for minimization problems with the following structure:

$$\min_{f \in X} \left[\mathcal{F}(\mathcal{K}(f)) + \mathcal{G}(f) \right] \quad (8)$$

where $\mathcal{K} : X \rightarrow U$ is a (possibly non-linear) operator, U is a Hilbert space and $\mathcal{F} : U \rightarrow \mathbb{R}$ and $\mathcal{G} : X \rightarrow \mathbb{R}$ are functionals

Algorithm 1 Non-Linear primal dual hybrid gradient

- 1: Given: $\sigma, \tau > 0$ s.t. $\sigma\tau\|\mathcal{K}\|^2 < 1$, $\gamma \in [0, 1]$ and $f_0 \in X$, $h_0 \in U$.
 - 2: **for** $i = 1, \dots$ **do**
 - 3: $h_{i+1} \leftarrow \text{prox}_{\sigma\mathcal{F}^*}(h_i + \sigma\mathcal{K}(\bar{f}_i))$
 - 4: $f_{i+1} \leftarrow \text{prox}_{\tau\mathcal{G}}(f_i - \tau[\partial\mathcal{K}(f_i)]^*(h_{i+1}))$
 - 5: $\bar{f}_{i+1} \leftarrow f_{i+1} + \gamma(f_{i+1} - f_i)$
-

on the dual/primal spaces. Note that (3) is a special case of (8) if we set $\mathcal{F} := \mathcal{L}(\cdot, g)$, $\mathcal{K} := \mathcal{T}$ and $\mathcal{G} := \mathcal{S}$.

In algorithm 1, \mathcal{F}^* denotes the Fenchel conjugate of \mathcal{F} , $h \in U$ is the dual variable and $[\partial\mathcal{K}(f_i)]^* : U \rightarrow X$ is the adjoint of the (Fréchet) derivative of \mathcal{K} in point f_i .

Example (Total Variation (TV) Regularized CT): The PDHG method has been widely applied to CT [19]. In CT, the forward operator is given by the ray-transform $\mathcal{P} : X \rightarrow Y$, which integrates the signal over a set of lines \mathbb{M} given by the acquisition geometry. Hence, elements in Y are functions on lines

$$\mathcal{P}(f)(\ell) := \int_{\ell} f(x)dx \quad \text{for } \ell \in \mathbb{M}.$$

and the adjoint of the derivative is the back-projection [20].

A typical example of variational regularization in imaging is TV regularization, which applies to signals that are represented by scalar functions of bounded variation. The corresponding regularization functional is then given as the 1-norm of the gradient magnitude, i.e. $\mathcal{S}(f) := \|\nabla f\|_1$, $\nabla : X \rightarrow X^d$, d is the dimension of the space.

The PDHG method can be used to solve the TV regularized CT optimization problem

$$\min_{f \in X} \|\mathcal{P}(f) - g\|_2^2 + \lambda \|\nabla f\|_1.$$

Since the proximal of $f \mapsto \|\nabla f\|_1$ is hard to compute, the following identification is better suited for recasting the above into (8):

$$\begin{aligned} \mathcal{K} : X &\rightarrow Y \times X^d \text{ as } \mathcal{K}(f) := [\mathcal{P}(f), \nabla f], \\ \mathcal{F}([h^{(1)}, h^{(2)}]) &:= \|h^{(1)} - g\|_2^2 + \|h^{(2)}\|_1 \text{ and } \mathcal{G}(f) := 0. \end{aligned}$$

B. Learned PDHG

The aim is to derive a learned reconstruction scheme inspired by PDHG, algorithm 1. We follow the observation in [21] and [22], that proximal operators can be replaced by other operators that are not necessarily proximal operators. The aforementioned publications replace a proximal operator with a denoising operator such as Block Matching 3D (BM3D). Our idea is to replace the proximal operators by *parametrized* operators where the parameters are learned from training data, resulting in a learned reconstruction operator.

In order to make the learned reconstruction operator well defined and implementable on a computer we also need to select a stopping criterion. Choosing a proper stopping criterion is an active research area, but for simplicity and usability we use a fixed number of iterates. By selecting a fixed number of iterations, the computation budget is also fixed

prior to training, which is a highly desirable property in time critical applications.

Algorithm 2 below outlines the resulting variant of the PDHG algorithm with I iterations in which the primal proximal has been replaced by a learned proximal, Γ_{θ^d} and the dual proximal by a learned proximal Λ_{θ^p} . Note that in this article we consider only a single forward model and no regularizing operator, so we have $\mathcal{K} = \mathcal{T}$, $U = Y$, but we give the algorithm in full generality for completeness.

Algorithm 2 Learned PDHG

```

1: Initialize  $f_0 \in X, h_0 \in U$ 
2: for  $i = 1, \dots, I$  do
3:    $h_{i+1} \leftarrow \Gamma_{\theta^d}(h_i + \sigma \mathcal{K}(\bar{f}_i), g)$ 
4:    $f_{i+1} \leftarrow \Lambda_{\theta^p}(f_i - \tau [\partial \mathcal{K}(f_i)]^*(h_{i+1}))$ 
5:    $\bar{f}_{i+1} \leftarrow f_{i+1} + \theta(f_{i+1} - f_i)$ 
6: return  $f_I^{(1)}$ 

```

In algorithm 2, there are several parameters that need to be selected. These are the parameters of the dual proximal, θ^d , the primal proximal, θ^p , the step lengths, σ , τ and the overrelaxation parameter, θ . In a learned PDHG algorithm these would all be inferred, *learned*, from training data.

We implemented this algorithm and show its performance in the results section. While the performance was comparable to traditional methods, it did not improve upon the state of the art in deep learning based image reconstruction.

C. Learned Primal-Dual

To gain substantial improvements, guided by recent advances in machine learning, the following modifications to the learned PDHG algorithm were done.

- Following [5], [13], extend the primal space to allow the algorithm some “memory” between the iterations.

$$f = [f^{(1)}, f^{(2)}, \dots, f^{(N_{\text{primal}})}] \in X^{N_{\text{primal}}}$$

Similarly extend the dual space U to $U^{N_{\text{dual}}}$.

- Instead of explicitly enforcing updates of the form $h_i + \sigma \mathcal{K}(\bar{f}_i)$, allow the network to learn how to combine the previous update with the result of the operator evaluation.
- Instead of hard-coding the over-relaxation $\bar{f}_{i+1} \leftarrow f_{i+1} + \theta(f_{i+1} - f_i)$, let the network freely learn in what point the forward operator should be evaluated.
- Instead of using the same learned proximal operators in each iteration allow them to differ. This increases the size of the parameter space but it also notably improves reconstruction quality.

The above modifications result in a new algorithm, henceforth called the *Learned Primal-Dual* algorithm, that is outlined in algorithm 3.

1) *Choice of Starting Point*: In theory, the Learned Primal-Dual algorithm can be used with any choice of starting points f_0 and h_0 . The most simple starting point, both from a conceptual and computational perspective, is zero-initialization

$$\begin{aligned} f_0 &= [0, 0, \dots, 0] \\ h_0 &= [0, 0, \dots, 0] \end{aligned} \quad (9)$$

where 0 is the zero element in the primal or dual space.

Algorithm 3 Learned Primal-Dual

```

1: Initialize  $f_0 \in X^{N_{\text{primal}}}, h_0 \in U^{N_{\text{dual}}}$ 
2: for  $i = 1, \dots, I$  do
3:    $h_i \leftarrow \Gamma_{\theta_i^d}(h_{i-1}, \mathcal{K}(f_{i-1}^{(2)}), g)$ 
4:    $f_i \leftarrow \Lambda_{\theta_i^p}(f_{i-1}, [\partial \mathcal{K}(f_{i-1}^{(1)})]^*(h_i^{(1)}))$ 
5: return  $f_I^{(1)}$ 

```

In cases where a good starting guess is available, it would make sense to use it. One such option is to assume that there exists a pseudo-inverse \mathcal{T}^\dagger , e.g. FBP for CT. For the dual variable, the data g enters into each iterate so there is no need for a good initial guess. This gives the starting point

$$\begin{aligned} f_0 &= [\mathcal{T}^\dagger(g), \mathcal{T}^\dagger(g), \dots, \mathcal{T}^\dagger(g)] \\ h_0 &= [0, 0, \dots, 0] \end{aligned} \quad (10)$$

In our tests we found that providing the Learned Primal-Dual algorithm with such an initial guess marginally decreased training time, but did not give better final results. Given that using the pseudo-inverse \mathcal{T}^\dagger adds more complexity by making the learned reconstruction operator depend on an earlier reconstruction, we *report values only from zero-initialization*.

D. Connection to Variational Regularization

We note that by selecting $N_{\text{primal}} = 2$ and $N_{\text{dual}} = 1$ the Learned Primal-Dual algorithm naturally reduces to the classical PDHG algorithm by making the following choices:

$$\begin{aligned} \Gamma_{\theta_i^d}(h, \mathcal{K}(f^{(2)}), g) &= \text{prox}_{\sigma \mathcal{F}_g^*}(h + \sigma \mathcal{K}(f^{(2)})) \\ \Lambda_{\theta_i^p} \left(\begin{bmatrix} f^{(1)} \\ f^{(2)} \end{bmatrix}, [\partial \mathcal{K}(f^{(1)})]^*(h) \right) &= \left[\begin{array}{c} \text{prox}_{\tau \mathcal{G}}(f^{(1)} - \tau [\partial \mathcal{K}(f^{(1)})]^*(h)) \\ (1 + \theta) \text{prox}_{\tau \mathcal{G}}(f^{(1)} - \tau [\partial \mathcal{K}(f^{(1)})]^*(h)) - \theta f^{(1)} \end{array} \right]. \end{aligned}$$

Even if the learned proximal operators do not have explicit access to the proximals, the universal approximation property of neural networks [23] guarantees that given sufficient training data these equalities can be approximated arbitrarily well.

A wide range of other optimization schemes can also be seen as special cases of the Learned Primal-Dual algorithm. For example, the gradient descent algorithm with step-length α for solving (8) is given by

$$f_{i+1} = f_i - \alpha \left([\partial \mathcal{K}(f_i)]^*([\nabla \mathcal{F}](\mathcal{K}(f_i))) + [\nabla \mathcal{G}](f_i) \right)$$

and can be obtained by selecting

$$\begin{aligned} \Gamma_{\theta_i^d}(h, \mathcal{K}(f^{(2)}), g) &= [\nabla \mathcal{F}_g](\mathcal{K}(f^{(2)})) \\ \Lambda_{\theta_i^p} \left(\begin{bmatrix} f^{(1)} \\ f^{(2)} \end{bmatrix}, [\partial \mathcal{K}(f^{(1)})]^*(h) \right) &= \left[\begin{array}{c} f^{(1)} - \alpha([\nabla \mathcal{G}](f^{(1)}) + [\partial \mathcal{K}(f^{(1)})]^*(h)) \\ f^{(1)} - \alpha([\nabla \mathcal{G}](f^{(1)}) + [\partial \mathcal{K}(f^{(1)})]^*(h)) \end{array} \right]. \end{aligned}$$

More advanced gradient based methods such as Limited memory BFGS are likewise sub-cases obtained by appropriate choices of learned proximal operators.

In summary, the Learned Primal-Dual algorithm contains a wide range of optimization schemes as special cases. If the parameters are appropriately selected, then the proposed algorithm should always perform *at least as well* as current variational regularization schemes given the *same stopping criteria*, which here is a fixed number of iterates.

IV. IMPLEMENTATION AND EVALUATION

We evaluate the algorithm on two low dose CT problems. One simplified using analytical phantoms based on ellipses and one with a more realistic forward model and human phantoms. We briefly describe these test cases and how we implemented the Learned Primal-Dual algorithm. We also describe the methods we compare against.

A. Test Cases

1) *Ellipse Phantoms*: This problem is identical to [5] and we restate it briefly. Training data is randomly generated ellipses on a 128×128 pixel domain. The forward operator is the ray transform and hence $\mathcal{T} = \mathcal{P}$.

The projection geometry was a sparse 30 view parallel beam geometry with 182 detector pixels. 5% additive Gaussian noise was added to the projections. Since the forward operator is linear, the adjoint of the derivative is simply the adjoint, which for the ray transform is the back-projection

$$[\partial \mathcal{T}(f)]^* = \mathcal{P}^*$$

2) *Human Phantoms*: In order to evaluate the algorithm on a clinically realistic use-case we consider reconstruction of simulated data from human abdomen CT scans as provided by Mayo Clinic for the AAPM Low Dose CT Grand Challenge [24]. The data includes full dose CT scans from 10 patients, of which we used 9 for training and 1 for evaluation. We used the 3 mm slice thickness reconstructions, resulting in 2168 training images, each 512×512 pixel in size. Thus, given that we minimize the pointwise error, the total number of data-points is $512^2 \times 2168 \approx 5 \times 10^8$.

We used a two-dimensional fan-beam geometry with 1000 angles, 1000 pixels, source to axis distance 500 mm and axis to detector distance 500 mm. In this setting, we consider the more physically correct non-linear forward model given by Beer-Lamberts law

$$\mathcal{T}(f)(\ell) = e^{-\mu \mathcal{P}(f)(\ell)}$$

where the unit of f is g/cm^3 and μ is the mass attenuation coefficient, in this work selected to $0.2 \text{ cm}^2/\text{g}$ which is approximately the value in water at x-ray energies. We used Poisson noise corresponding to 10^4 incident photons per pixel before attenuation, which would correspond to a low dose CT scan. We find the action of the adjoint of the derivative by straightforward computation

$$[\partial \mathcal{T}(f)]^*(g) = -\mu \mathcal{P}^*(e^{-\mu \mathcal{P}(f)(\cdot)} g(\cdot)) \quad \text{for } g \in Y.$$

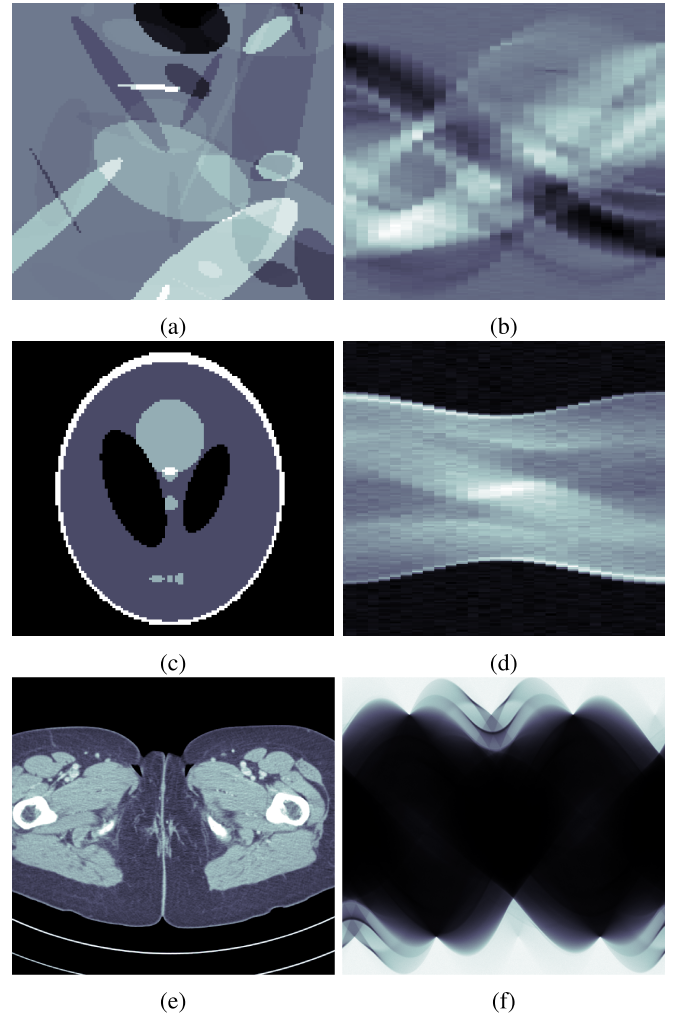


Fig. 1. Example of data which are used for training and validation. Top: Randomly generated ellipses. Middle: Validation data, here given by the modified Shepp-Logan phantom. Bottom: Validation data generated from the human phantoms. (a) Ellipse training phantom. (b) Ellipse training sinogram. (c) Shepp-Logan phantom. (d) Shepp-Logan sinogram. (e) Human phantom. (f) Human pre-log sinogram.

The forward model can also be linearised by applying $-\log(\cdot)/\mu$ to both the data and forward operator, which then simply becomes the ray-transform as for the ellipse data. We implemented both the pre-log (non-linear) and post-log (linear) forward models and compare their results.

For validation of the ellipse data case, we simply use the (modified) Shepp-Logan phantom and for the human phantom data we use one held out set of patient data. See fig. 1 for examples.

B. Implementation

The methods described above were implemented in Python using Operator Discretization Library (ODL) [25] and TensorFlow [26]. All operator-related components, such as the forward operator \mathcal{T} , were implemented in ODL, and these were then converted into TensorFlow layers using the `as_tensorflow_layer` functionality of ODL. The neural network layers and training were implemented using TensorFlow. The implementation utilizes abstract ODL structures

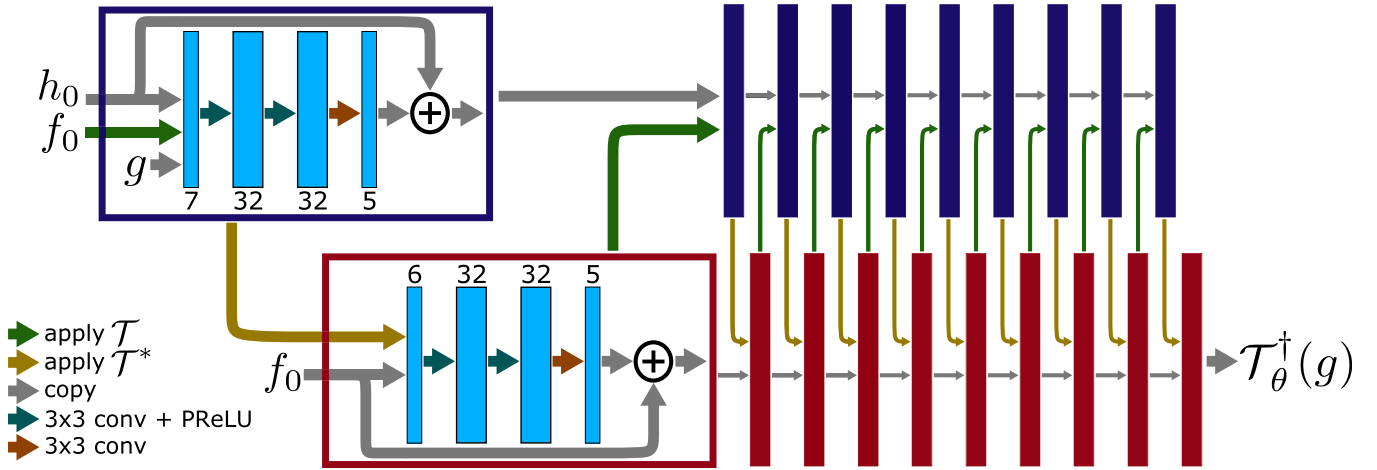


Fig. 2. Network architecture used to solve the tomography problem. The dual iterates are given in blue boxes, while the primal iterates are in the red boxes. The blue/red boxes all have the same architecture, which is illustrated in the corresponding large boxes. Several arrows pointing to one box indicates concatenation. The initial guesses enter from the left, while the data is supplied to the dual iterates. In the classical PDHG algorithm, the primal iterates would instead of a CNN be given by $\text{prox}_{\mathcal{T}^* \mathcal{G}}$ with over-relaxation, and the dual iterates would be given by $\text{prox}_{\sigma \mathcal{F}^*}$.

for representing functional analytic notions and is therefore generic and easily adaptable to other inverse problems. In particular, the code can be easily adapted to other imaging modalities.

We used the ODL operator `RayTransform` in order to evaluate the ray transform and its adjoint using the graphics processing unit (GPU) accelerated ‘`astra_gpu`’ backend [27].

1) *Deep Neural Network and Training Details:* Given the general Learned Primal-Dual scheme in algorithm 3, a parametrization of the learned proximal operators is needed in order to proceed. In many inverse problems, and particularly in CT and MRI reconstruction, most of the useful properties for both the forward operator and prior are approximately translation invariant. For this reason the resulting reconstruction operator should be approximately translation invariant, which indicates that CNNs are suitable for parametrizing the aforementioned reconstruction operator.

We used learned proximal operators of the form

$$\text{Id} + \mathcal{W}_{w_3, b_3} \circ \mathcal{A}_{c_2} \circ \mathcal{W}_{w_2, b_2} \circ \mathcal{A}_{c_1} \circ \mathcal{W}_{w_1, b_1}$$

where Id is the identity operator that makes the network a *residual* network. There are two main reasons for choosing such a structure. First, proximal operators (as the name implies) are typically close to the identity and second, there is rich evidence in the machine learning literature [28] that networks of this type are easier to train. Heuristically this is because each update does not need to learn the whole update, but only a small offset from the identity.

Additionally, we used affine operators \mathcal{W}_{w_j, b_j} parametrized by *weights* w_j and *biases* b_j . The affine operators are defined in terms of so called convolution operators (here given on the primal space, but equivalently on the dual space). These are given as affine combinations of regular convolutions, more specifically:

$$\mathcal{W}_{w_j, b_j} : X^n \rightarrow X^m$$

where the k :th component is given by

$$[\mathcal{W}_{w_j, b_j}([f^{(1)}, \dots, f^{(n)}])]^{(k)} = b_j^{(k)} + \sum_{l=1}^n w_j^{(l, k)} * f^{(l)}$$

where $b_j \in \mathbb{R}^m$ and $w_j \in X^{n \times m}$.

The non-linearities were chosen to be Parametric Rectified Linear Units (PReLU) functions

$$\mathcal{A}_{c_j}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -c_j x & \text{else.} \end{cases}$$

This type of non-linearity has proven successful in other applications such as classification [29].

We let the number of data that persists between the iterates be $N_{\text{primal}} = N_{\text{dual}} = 5$. The convolutions were all 3×3 pixel size, and the number of channels was, for each primal learned proximal, $6 \rightarrow 32 \rightarrow 32 \rightarrow 5$, and for the duals $7 \rightarrow 32 \rightarrow 32 \rightarrow 5$ where the higher number of inputs is due to the data g being supplied to the dual proximals.

We let the number of unrolled iterations be $I = 10$, that is the operator \mathcal{T} and the adjoint of its derivative $[\partial \mathcal{T}(f_i^{(1)})]^*$ are both evaluated 10 times by the network. Since each iterate involves two 3-layer networks, one for each proximal, the total depth of the network is 60 convolutional layers and the total number of parameters approximately $2.4 \cdot 10^5$. In the context of deep learning, this is a deep network but with a small number of parameters. The network is visualized in fig. 2.

We used the Xavier initialization scheme [30] for the convolution parameters, and initialized all biases to zero.

We trained the network by minimizing the empirical loss (5) using training data as explained above using the ADAM optimizer in TensorFlow [31]. We used 10^5 batches on each problem and used a learning rate schedule according to cosine annealing [32], i.e. the learning rate in step t was

$$\eta_t = \frac{\eta_0}{2} \left(1 + \cos\left(\pi \frac{t}{t_{\max}}\right) \right)$$

where the initial learning rate η_0 was set to 10^{-3} . We also let the parameter β_2 of the ADAM optimizer to 0.99 and let all other parameters use the default choices. We performed global gradient norm clipping [33], limiting the gradient norms to 1 in order to improve training stability and used a batch size of 5 for the ellipse data and 1 for the human phantoms.

We did not use any regularization of the learned parameters, nor did we utilize any tricks such as dropout or batch normalization. Neither did we perform any data augmentation.

The training was done using a single GTX 1080 TI GPU and took about 11 hours for the ellipse data and 40 hours for the human phantoms.

2) Incorporating the Forward Operator in Neural Networks:

In order to minimize the loss function (4), stochastic gradient descent (SGD) type methods are typically used and these require (an estimate of) the gradient of the loss function

$$[\nabla L](\theta) = \mathbb{E}_{(f,g) \sim \mu} \left[2[\partial_\theta \mathcal{T}_\theta^\dagger(\mathbf{g})]^* (\mathcal{T}_\theta^\dagger(\mathbf{g}) - \mathbf{f}) \right],$$

where $[\partial_\theta \mathcal{T}_\theta^\dagger(\mathbf{g})]^*$ is the adjoint of the derivative (with respect to θ) of the learned reconstruction operator applied in \mathbf{g} , often called gradient in the machine learning literature. This introduces a challenge since it will depend on each component of the neural network, including the learned proximal operators but also the forward operator \mathcal{T} and the backward operator $[\partial \mathcal{T}(f)]^*$, propagated through all I iterations.

To solve this, we used the built in automatic differentiation functionality of TensorFlow which uses the chain rule (back-propagation). This in turn requires the adjoints of the derivatives of each individual component which for the proximals were computed by TensorFlow and for the operators by ODL.

C. Comparison

We compare the algorithm to several widely used algorithms, including standard FBP and (isotropic) TV regularized reconstruction. We also compare against several learned schemes. These are briefly summarized here, see the references for full descriptions.

The FBP reconstructions were done with a Hann filter and used the method `fbp_op` in ODL. The TV reconstruction was performed using 1000 iterations of the classical PDHG algorithm, implemented in ODL as `pdhg`. The filter bandwidth in the FBP reconstruction and the regularization parameter in the TV reconstruction were selected in order to maximize the peak signal to noise ratio (PSNR).

The partially *Learned Gradient* method in [5] is similar to the algorithm proposed in this article, but differs in that instead of learning proximal operators it learns a gradient operator and the forward operator enters into the neural network through the gradient of the data likelihood. Publicly available code and parameters [34] were used.

The next comparison is against a deep learning based approach for post-processing based on a so called *U-Net* [35]. The U-Net was first proposed for image segmentation, but by changing the number of output channels to one, it can also be used for post-processing as was done in [7]. Here an initial reconstruction is first performed using FBP and a

neural network is trained on pairs of noisy FBP images and noiseless/low noise ground truth images, learning a mapping between them. We re-implemented the algorithm from [7] but found that using the training procedure as stated in the paper gave sub-optimal results. We hence report values from using the same training scheme as for our other algorithms in order to give a more fair comparison.

Additionally, our comparison includes learned PDHG, algorithm 2, as well as the following two simplified versions of the Learned Primal-Dual algorithm. The first is a *Learned Primal* algorithm, which does not learn any parameters for the dual proximal, instead it returns the residual

$$\Gamma_{\theta_i^d}(h_{i-1}, \mathcal{T}(f_{i-1}^{(2)}), g) = \mathcal{T}(f_{i-1}^{(2)}) - g$$

The second, *FBP + residual denoising* algorithm, further simplifies the problem by discarding the forward operator completely, and can be seen as selecting

$$\Lambda_{\theta_i^p}(f_{i-1}, [\partial \mathcal{T}(f_{i-1}^{(1)})]^*(h_i^{(1)})) = \Lambda_{\theta_i^p}(f_{i-1})$$

Since this method does not have access to the data g , we select the initial guess according to a FBP, see (10). This makes the algorithm a learned denoiser.

For the human phantoms we compare both non-linear and linearized versions of the forward operator, but given that training times are noticeably longer, we only compare to the previously established methods of FBP, TV and U-Net denoising.

All learned algorithms were trained using the same training scheme as outlined in section IV-B1, and measure the runtime, PSNR and the structural similarity index (SSIM) [36].

All methods that we compare against are available in the accompanying source code.

V. RESULTS

The quantitative results for the ellipse data is given in [table I](#), where we can see that the proposed Learned Primal-Dual algorithm out-performs the classical schemes (FBP and TV) significantly w.r.t. the reconstruction error as measured by both PSNR and SSIM. We also note that the Learned Primal-Dual algorithm gives a large improvement over the previous deep learning based methods such as the learned gradient scheme and U-Net based post-processing, giving an improvement exceeding 6 dB. The Learned Primal-Dual algorithm also outperforms the Learned PDHG and the FBP + residual denoising algorithms by wide margins.

The only method that achieves results close to the Learned Primal-Dual algorithm is the Learned Primal method, but the Learned Primal-Dual algorithm gives a noticeable improvement of 1.3 dB.

The results are visualized in [fig. 3](#). We note that small structures, such as the small inserts, are much more clearly visible in the Learned Primal and Learned Primal-Dual reconstructions than in the other reconstructions. We also note that both the Learned PDHG and Learned Primal reconstruction seem to have a halo artefact close to the outer bone which is absent in the Learned Primal-Dual reconstruction.

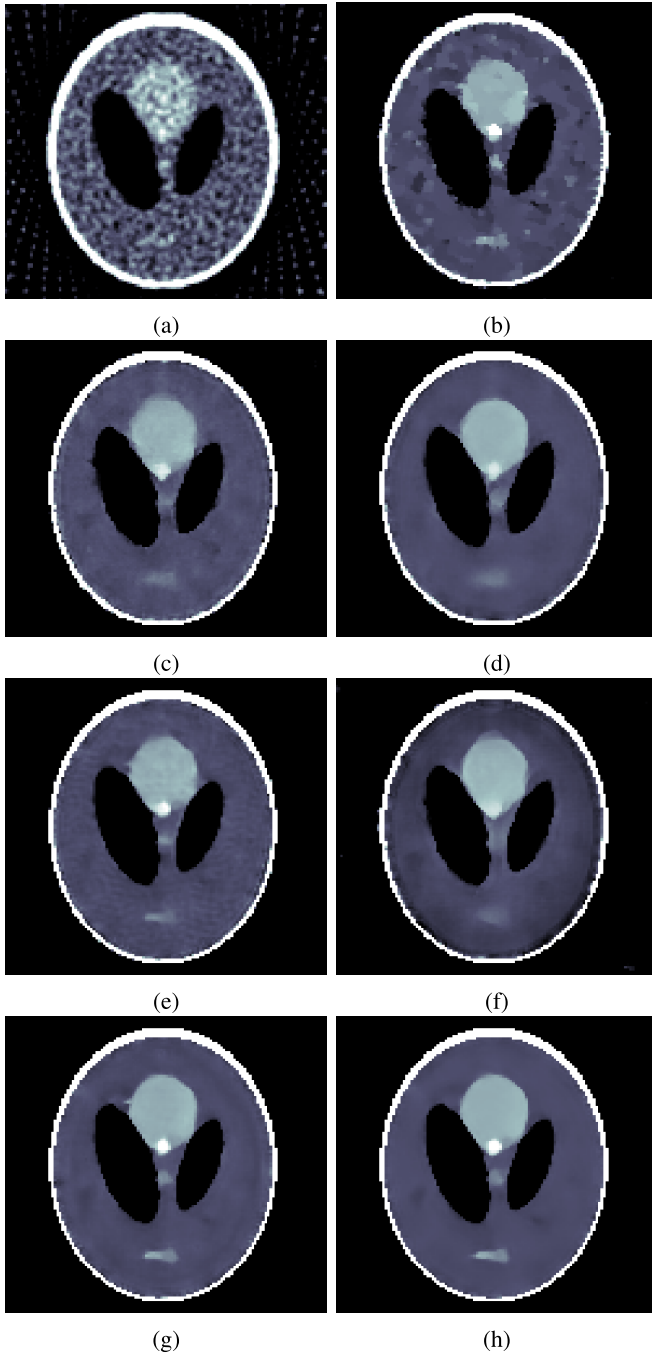


Fig. 3. Reconstructions for the ellipse data using the compared methods. The window is set to $[0.1, 0.4]$, corresponding to the soft tissue of the modified Shepp-Logan phantom. (a) FBP. (b) TV. (c) FBP + U-Net denoising. (d) FBP + residual denoising. (e) Learned gradient. (f) Learned PDHG. (g) Learned Primal. (h) Learned Primal-Dual.

With respect to run-time the learned methods that involve calls to the forward operator (Learned Gradient, PDHG, Primal, Primal-Dual) are slower than the methods that do not (FBP + U-Net denoising, Residual) by a factor ≈ 6 . When compared to TV regularized reconstruction all learned methods are at least 2 orders of magnitude faster.

Quantitative results for the human phantoms data are presented in table II. We note that the FBP reconstruction has a much more competitive image quality than it had for the

TABLE I
COMPARISON OF RECONSTRUCTION METHODS FOR THE ELLIPSES.
PSNR MEASURED IN dB AND RUNTIME IN ms

Method	PSNR	SSIM	Runtime	Parameters
FBP	19.75	0.597	4	1
TV	28.06	0.929	5 166	1
FBP + U-Net denoising	29.20	0.944	9	10^7
FBP + residual denoising	32.38	0.972	9	$1.2 \cdot 10^5$
Learned Gradient	32.29	0.981	56	$1.2 \cdot 10^4$
Learned PDHG	28.32	0.909	48	$2.4 \cdot 10^4$
Learned Primal	36.97	0.986	43	$1.2 \cdot 10^5$
Learned Primal-Dual	38.28	0.989	49	$2.4 \cdot 10^5$

ellipse data, both quantitatively and visually. It is likely for this reason that the FBP + U-Net denoising performs better than it did on the ellipses, outperforming TV by 4.4 dB. However, if we look at the SSIM we note that this improvement does not translate as well to the structural similarity, where the method is comparable to TV regularization.

Both quantitatively and visually, the linear and non-linear versions of the Learned Primal-Dual algorithm give very similar results. We will focus on the linear version which gave slightly better results.

The Learned Primal-Dual algorithm gives a 10.5 dB improvement over the FBP reconstruction, a 6.6 dB improvement over TV and 2.2 dB over the U-Net denoiser. This is less than for the ellipse data, but still represents a large improvement. On the other hand, while the U-Net denoiser did not improve the SSIM as compared to TV regularization, the Learned Primal-Dual algorithm gives a large improvement.

This improvement is also present in the images when inspected visually in fig. 4. In particular, we see that some artifacts visible in the FBP reconstruction are still discernible in the U-Net denoiser and TV reconstructions. Examples include streak artifacts, especially around the edges of the phantom and structures spuriously created from noise, such as a line in the muscle above the right bone. These are mostly absent in the Learned Primal-Dual reconstruction. However, we do note that the images do look slightly over-smoothed. Both of these observations become especially apparent if we look at the zoomed in regions, where we note that the Learned Primal-Dual algorithm is able to reconstruct finer detail than the other algorithms, but gives a very smooth texture.

With respect to the run time, the Learned Primal-Dual is more competitive with the FBP and U-Net denoiser algorithms for full size data than for the ellipse data. This is because the size of the data is much larger, which increases the runtime of the FBP reconstruction, which is also needed to compute the initial guess for the U-Net denoiser. As for the ellipse data, both learned methods outperform TV regularized reconstruction by two orders of magnitude with respect to runtime.

VI. DISCUSSION

The results show that the Learned Primal-Dual algorithm outperforms classical reconstruction algorithm by large margins as measured in both PSNR and SSIM and also improves upon learned post-processing methods for both simplified

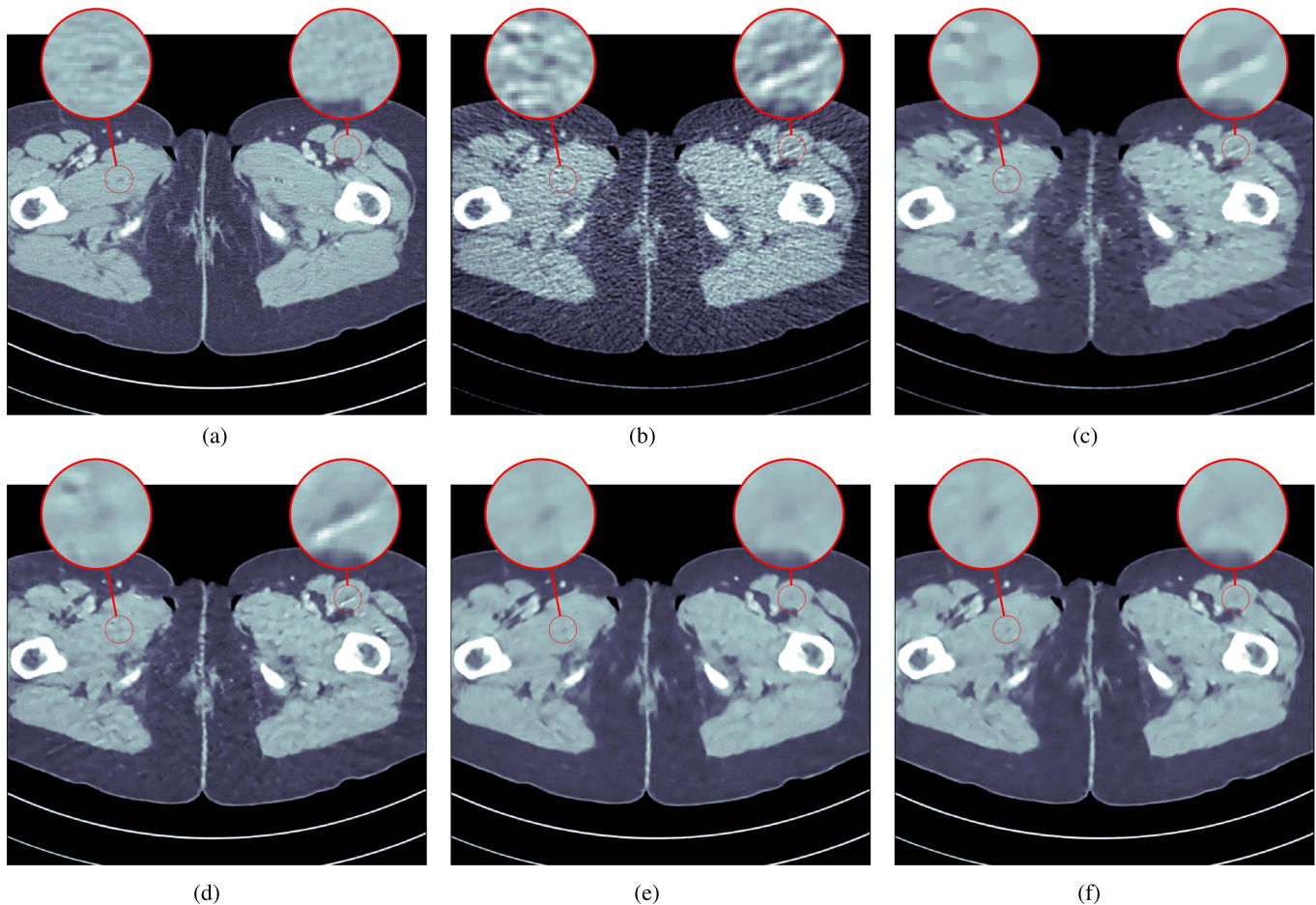


Fig. 4. Reconstructions of a human phantom along with two zoomed in regions indicated by small circles. The left zoom-in has a true feature whereas texture in right zoom-in is uniform. The window is set to $[-200, 200]$ HU. Among the methods tested, only the Learned Primal-Dual algorithm correctly recovers these regions. In the others, the true feature in the left zoom-in is indistinguishable from other false features of same size/contrast and right-zoom in has a streak artifact. Note also the clinically feasible runtime of the Learned Primal-Dual algorithm. To summarize, the Learned Primal-Dual algorithm offers performance advantages over other methods that translate into true clinical usefulness. (a) 512×512 pixel human phantom. (b) Filtered back-projection (FBP) PSNR 33.65 dB, SSIM 0.830, 423 ms. (c) Total variation (TV) PSNR 37.48 dB, SSIM 0.946, 64 371 ms. (d) FBP + U-Net denoising PSNR 41.92 dB, SSIM 0.941, 463 ms. (e) Primal-Dual, linear PSNR 44.10 dB, SSIM 0.969, 620 ms. (f) Primal-Dual, non-linear PSNR 43.91 dB, SSIM 0.969, 670 ms.

TABLE II

COMPARISON OF THE LEARNED PRIMAL-DUAL ALGORITHM WITH OTHER METHODS FOR THE HUMAN PHANTOM DATA. UNITS FOR ENTRIES ARE THE SAME AS IN TABLE I

Method	PSNR	SSIM	Runtime	Parameters
FBP	33.65	0.830	423	1
TV	37.48	0.946	64 371	1
FBP + U-Net denoising	41.92	0.941	463	10^7
Learned Primal-Dual, linear	44.11	0.969	620	$2.4 \cdot 10^5$
Learned Primal-Dual, non-linear	43.91	0.969	670	$2.4 \cdot 10^5$

ellipse data and for human phantoms. In addition, especially for the 512×512 human phantoms, the reconstruction time is comparable with even filtered back-projection and learned post-processing.

One interesting, and to the best of our knowledge, unique feature of the Learned Primal-Dual algorithm in the field of deep learning based CT reconstruction, is that it gives reconstructions working directly from data, without any initial reconstruction as input.

Since the algorithm is iterative, we can visualize the iterates to gain insight into how it works. In [fig. 5](#) we show some

iterates with the non-linear forward operator. We note that the reconstruction stays very bad until the 8:th iterate when most structures seem to come in place, but the image is still noisy. Between the 8:th and 10:th iterate, we see that the algorithm seems to perform an edge-enhancing step. It thus seems like the learned iterative scheme works in two steps, first finding the large scale structures and then fine-tuning the details.

Similarly to the edge-enhancement that seems to be performed in the primal space, we note that in the dual space the sinogram that is back-projected seems to be band-pass filtered to exclude both very low and very high frequencies.

We note that in the very noisy and under-sampled data used for the ellipse phantoms, the learned algorithms that make use of the forward operator, such as the Learned Gradient, Primal and Primal-Dual algorithms outperform even state of the art post-processing methods by large margins and that in this regimen, TV regularization performs relatively well when compared to post-processing methods. This improvement in reconstruction quality when incorporating the forward operator, while still substantial, is not as large for the human phantom in which the data was less noisy.

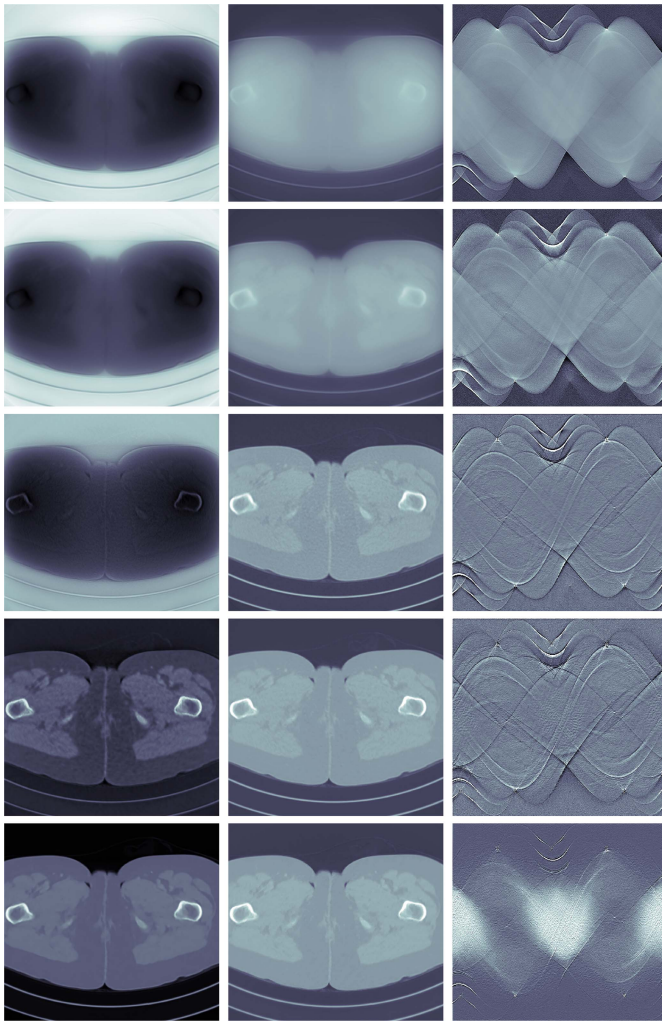


Fig. 5. Iterates 2, 4, 6, 8 and 10 in the Learned Primal-Dual algorithm when reconstructing the human phantoms using a non-linear forward model. Left: Reconstruction ($f_i^{(1)}$). Middle: Point of evaluation for the forward operator ($f_i^{(2)}$). Right: Point of evaluation for the adjoint of the derivative ($h_i^{(1)}$). Windows selected to cover most of the range of the values.

To explain this, we conjecture that in the limit of highly noisy data where the initial reconstruction as given by e.g. FBP becomes very bad, learned schemes that incorporate the forward model and work directly from data, such as the Learned Primal-Dual algorithm, has a considerable advantage over post-processing methods and that this advantage increases with decreasing data quality.

Further along these lines, note that for the human data the post-processing gives a large improvement in PSNR when compared to TV regularization, which is not necessarily reflected in the SSIM. On the other hand, the Learned Primal-Dual algorithm improves upon both PSNR and SSIM. This can be explained by the learned post-processing being limited by the information content of the FBP while the Learned Primal-Dual algorithm works directly with data and is thus limited by the information content of the data, which is greater or equal to that of the FBP. In theory, the Learned Primal-Dual algorithm can thus find structures that are not present in the FBP, something post-processing methods cannot.

In these experiments we found that while the algorithm seems to handle non-linear forward models well, we did not observe any notable performance improvement by doing so. This may indicate that performing reconstructions on post-log data is preferable.

The structure of the neural network was not extensively fine-tuned and we suspect that better results could be obtained by a better choice of network for the learned proximal operators. We also observed that the choice of optimizer and learning rate decay had a large impact on the results, and we suspect that further research into how to correctly train learned reconstruction operators will prove fruitful.

Finally, we observe that the reconstructions, while outperforming all of the compared methods with respect to PSNR and SSIM, suffers from a perceived over-smoothing when inspected visually. We suspect that the particular choice of objective function used in this article, the squared norm (4), is a main cause of this and invite future researchers to implement learned reconstruction operators that use more advanced loss functions such as perceptual losses [37].

VII. CONCLUSION

We have proposed a new algorithm in the family of deep learning based iterative reconstruction schemes. The algorithm is inspired by the PDHG algorithm, where we replace the proximal operators by learned operators. In contrast to several recently proposed algorithms, the new algorithm works directly from tomographic data and does not depend on any initial reconstruction.

We showed that the algorithm gives state of the art results on a computed tomography problem for both analytical and human phantoms. For analytical phantoms, it improves upon both classical algorithms such as FBP and TV, and post-processing based algorithms by at least 6 dB while also improving the SSIM. The improvements for the human phantom were more modest, but the algorithm still improves upon a TV regularized reconstruction by 6.6 dB and gives an improvement of 2.2 dB when compared to a learned post-processing.

We hope that this algorithm will inspire further research in Learned Primal-Dual schemes and that the method will be applied to other imaging modalities.

ACKNOWLEDGEMENTS

The authors also thank Dr. Cynthia McCollough, the Mayo Clinic, and the American Association of Physicists in Medicine for providing the data necessary for performing comparison using a human phantom.

REFERENCES

- [1] M. Bertero, H. Lantéri, and L. Zanni, "Iterative image reconstruction: A point of view," in *Proc. Interdiscipl. Workshop Math. Methods Biomed. Imag. Intensity-Modulated Radiation (IMRT)*, Pisa, Italy, 2008, pp. 37–63.
- [2] H. W. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems* (Mathematics and its Applications). Norwell, MA, USA: Kluwer, 2000.
- [3] O. Scherzer, M. Grasmair, H. Grossauer, M. Haltmeier, and F. Lenzen, *Variational Methods in Imaging* (Applied Mathematical Sciences), vol. 167. New York, NY, USA: Springer-Verlag, 2009.

- [4] G. Wang, "A perspective on deep imaging," *IEEE Access*, vol. 4, pp. 8914–8924, 2016.
- [5] J. Adler and O. Öktem, "Solving ill-posed inverse problems using iterative deep neural networks," *Inverse Problems*, vol. 33, no. 12, p. 124007, 2017.
- [6] 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.
- [7] 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. 2016.
- [8] 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, 2017.
- [9] 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. Imaging*, vol. 31, no. 9, pp. 1682–1697, Sep. 2012.
- [10] T. Meinhardt, M. Moller, C. Hazirbas, and D. Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *Proc. ICCV*, Oct. 2017, pp. 1781–1790.
- [11] I. Dokmanić, J. Bruna, S. Mallat, and M. de Hoop. (2016). "Inverse problems with invariant multiscale statistics." [Online]. Available: <https://arxiv.org/abs/1609.05502>
- [12] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 10–18.
- [13] P. Putzky and M. Welling. (2017). "Recurrent inference machines for solving inverse problems." [Online]. Available: <https://arxiv.org/abs/1706.04008>
- [14] K. Hammernik *et al.*, "Learning a variational network for reconstruction of accelerated MRI data," *Magn. Reson. Med.*, to be published. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/mrm.26977/abstract>, doi: 10.1002/mrm.26977.
- [15] M. Mardani *et al.* (May 2017). "Deep generative adversarial networks for compressed sensing automates MRI." [Online]. Available: <https://arxiv.org/abs/1706.00051>
- [16] J. Adler. (2017). *Learned Primal-Dual Reconstruction*. [Online]. Available: https://github.com/adler-j/learned_primal_dual
- [17] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2011.
- [18] T. Valkonen, "A primal–dual hybrid gradient method for nonlinear operators with applications to MRI," *Inverse Problems*, vol. 30, no. 5, p. 055012, 2014.
- [19] E. Y. Sidky, J. H. Jørgensen, and X. Pan, "Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm," *Phys. Med. Biol.*, vol. 57, no. 10, pp. 3065–3091, May 2012.
- [20] A. Markoe, *Analytic Tomography* (Encyclopedia of Mathematics and its Applications). Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [21] F. Heide *et al.*, "FlexISP: A flexible camera image processing framework," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 231:1–231:13, 2014.
- [22] Y. Romano, M. Elad, and P. Milanfar. (2016). "The little engine that could: Regularization by Denoising (RED)." [Online]. Available: <https://arxiv.org/abs/1611.02862>
- [23] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [24] C. McCollough, "TU-FG-207A-04: Overview of the low dose CT grand challenge," *Med. Phys.*, vol. 43, no. 6, pp. 3759–3760, 2016.
- [25] J. Adler, H. Kohr, and O. Öktem. (2017). *Operator Discretization Library (ODL)*. [Online]. Available: <https://github.com/odlgroup/odl>
- [26] M. Abadi *et al.* (2015). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [27] W. van Aarle *et al.*, "Fast and flexible X-ray tomography using the ASTRA toolbox," *Opt. Exp.*, vol. 24, no. 22, pp. 25129–25147, 2016.
- [28] 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.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.
- [30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.
- [31] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [32] I. Loshchilov and F. Hutter. (2016). "SGDR: Stochastic gradient descent with warm restarts." [Online]. Available: <https://arxiv.org/abs/1608.03983>
- [33] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR*, vol. abs/1211.5063, 2012. [Online]. Available: <http://arxiv.org/abs/1211.5063>
- [34] J. Adler. (2017). *Solving Ill-Posed Inverse Problems Using Iterative Deep Neural Networks*. [Online]. Available: https://github.com/adler-j/learned_gradient_tomography
- [35] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-24574-4_28
- [36] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [37] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 694–711.