Multi-Scale Learned Iterative Reconstruction

Andreas Hauptmann, Jonas Adler, Simon Arridge, and Ozan Öktem

***Abstract*—Model-based learned iterative reconstruction methods have recently been shown to outperform classical reconstruction algorithms. Applicability of these methods to large scale inverse problems is however limited by the available memory for training and extensive training times, the latter due to computationally expensive forward models. As a possible solution to these restrictions we propose a multi-scale learned iterative reconstruction scheme that computes iterates on discretisations of increasing resolution. This procedure does not only reduce memory requirements, it also considerably speeds up reconstruction and training times, but most importantly is scalable to large scale inverse problems with non-trivial forward operators, such as those that arise in many 3D tomographic applications. In particular, we propose a hybrid network that combines the multi-scale iterative approach with a particularly expressive network architecture which in combination exhibits excellent scalability in 3D.**

Applicability of the algorithm is demonstrated for 3D cone beam computed tomography from real measurement data of an organic phantom. Additionally, we examine scalability and reconstruction quality in comparison to established learned reconstruction methods in two dimensions for low dose computed tomography on human phantoms.

***Index Terms*—Model-based learning, iterative reconstruction, cone beam computed tomography, deep learning, inverse problems**

I. Introduction

Computed tomography (CT) is an imaging technology where the interior anatomy of a subject is computed from a series of X-ray radiographs acquired by radiating the subject from different directions. CT has had a profound impact on medical practice and it is now an indispensable technology in a wide spectrum of clinical and industrial applications. It has also been essential for advancing our understanding of disease in medical research.

CT imaging is however associated with risks, especially when it is used for screening. CT relies on repeatedly exposing a patient to ionising radiation of X-rays and hence there is an ongoing effort to minimise the total dose delivered to a patient during a CT scan. For that purpose, low dose CT protocols can be employed where fewer X-ray photons are measured, which consequently reduces the signal-to-noise ratio in acquired data. Widely used reconstruction techniques in clinical practice, such as filtered backprojection, are based on sampling theory and as such are not properly adapted to account for the statistical characteristics of measured data with high noise level. Hence, applying these schemes on low-dose CT data will produce sub-optimal images which consequently prevents low dose protocols from being widely adapted. Furthermore, in industrial and scientific applications which often utilise μCT systems, reconstructions are typically computed by the Feldkamp-Davis-Kress (FDK) algorithm [1] used for cone beam cone beam CT (CBCT) measurements. Here the same requirement of many angles applies, but additionally reconstructions often exhibit cone beam artefacts due to the measurement geometry. Accurate measurement procedures to overcome these issues can be highly time consuming and effectively limit experimental capacity, hence there is a need for advanced and computationally efficient reconstructions algorithms from few angle measurements.

Over the years, a wide range of reconstruction methods have been developed that better account for the aforementioned statistical properties in few angle and low-dose CT scans. Among these, the most powerful and flexible have been variational model-based methods [2], [3], [4]. These offer a plug-and-play architecture for reconstruction where a user provides a model for how data is generated in absence of noise (forward operator), a statistical model for noise in data, and a prior model for desired reconstructions. The forward operator together with the statistical model for data ensures consistency against measured data, whereas the prior mainly prevents overfitting by penalising images that have ‘irregular’ behaviour. These variational methods can sometimes be interpreted as computing the most likely solution (maximum a posteriori estimate) [5, sec. 3.3.2], see also [6], [7], [8]. Variational model-based reconstruction is, however, computationally demanding since it involves solving a large-scale optimisation problem. This becomes prohibitive in time-critical applications, like clinical CT imaging, and especially so when the prior model is non-smooth, as in sparsity promoting priors. Another challenge lies in choosing an appropriate prior [9], [10], [11], [12], [13] and [5, sec. 3.4].

Motivated by these shortcomings, recently there have been several efforts in using methods from deep learning for reconstruction [5]. One particular approach trains a deep neural network with a suitable architecture against supervised data using a squared ℓ^2-loss [5, sec. 5.1.2]. This can be seen as an approximate way to compute the average solution (conditional mean estimate). When properly adapted, such

This work was partially supported by the Academy of Finland Project 312123 (Finnish Centre of Excellence in Inverse Modelling and Imaging, 2018–2025) and Project 334817, as well as British Heart Foundation grant NH/18/1/33511, EPSRC grant EP/M020533/1, and EPSRC-Wellcome grant WT101957. The authors would also like to acknowledge the Swedish Foundation for Strategic Research grants *Low complexity reconstruction for medicine* (AM13-0049) and *3D reconstruction with simulated image formation models* (ID14-0055). Funding has also been provided by Elekta (Stockholm, Sweden).

AH and JA contributed equally

A. Hauptmann is with the Research Unit of Mathematical Sciences; University of Oulu, Oulu, Finland and with the Department of Computer Science; University College London, London, United Kingdom.

J. Adler did this work at Elekta, Stockholm, Sweden and KTH – Royal Institute of Technology, Stockolm, Sweden. He is currently with DeepMind, London, UK.

S. Arridge is with the Department of Computer Science; University College London, London, United Kingdom.

O. Öktem is with is with the Department of Mathematics, KTH – Royal Institute of Technology, Stockholm, Sweden.

data driven approaches considerably outperform purely model based reconstruction techniques regarding *both* reconstruction quality and reconstruction speed [14].

One natural approach is to use deep learning in the above context to directly learn the mapping from data to image [15]. Such an approach scales poorly, it requires re-training when data acquisition changes, and it relies on access to huge amounts of training data. Hence, this is not a feasible approach for clinical CT where high quality training data is scarce, as access to projection data is limited. Another approach is to use deep learning as a post-processing tool to improve upon an initial reconstruction. This is computationally feasible as shown in [16], [17], [18], but such an approach is essentially limited by the information content of the initial reconstruction and the richness of a-priori information learned from training data, which potentially increases bias in the reconstruction.

Learned iterative reconstruction methods seek to overcome these drawbacks by combining deep learning with a model-based approach. More precisely, the idea is to use a deep neural network architecture for reconstruction that incorporates an explicit handcrafted forward operator and the adjoint of its derivative [14], [19], [20], [21], [22]. The idea is to unroll a suitable iterative scheme (usually taken from a model-based approach) that in the limit defines a reconstruction operator [5, sec. 4.1.9]. This yields further improvements to reconstruction quality as compared to the direct learning or post-processing approaches mentioned before. Furthermore, including an explicit forward operator improves robustness and generalisability [23], [24], see also [25]. Additionally, it also reduces the amount of training data, since networks tend to have fewer parameters and the forward operator encodes a major portion of the relations in data that come from the acquisition geometry.

As already indicated, learned iterative reconstruction methods are typically trained in an end-to-end manner. Hence, the entire unrolled fixed-point scheme is treated as a single network and all its parameters are trained jointly. This provides an optimal set of network parameters under suitable optimisation procedures, but it also comes with two challenges. First, the memory footprint of storing and manipulating the network is too large for most single GPU configurations. Furthermore, during training the loss function is evaluated several times. Each of these involves evaluating the forward operator and its adjoint, or the adjoint of its derivative, which quickly leads to unreasonable training times. Hence, current learned iterative reconstruction algorithms do not scale well to large-scale and higher dimensions, such as fully 3D CT.

One possible solution to address these computational challenges is to adopt a greedy approach for training. Here each unrolled iteration in the network is trained separately [20]. In this way, training of each unrolled iterate and evaluation of the forward operator can be separated, thus rendering a training procedure feasible. On the other hand, such an approach clearly does not represent an optimal selection of network parameters as compared to jointly optimising over all network parameters for all unrolled iterates, as discussed in [20, sec. III-A]. Therefore, such a greedy approach renders a trained network for reconstruction that may fall short in reconstruction quality compared to end-to-end schemes, we refer to [26] for further discussion on greedy schemes. Additionally, reconstruction times are still comparably slow due to multiple applications of the forward operator. In some cases however, the issue of computation times can be tackled by using faster approximate models [27], if available, but memory footprint remains an issue.

In summary, the computational challenges of utilising learned iterative reconstructions are twofold: (i) Managing memory footprint; (ii) Feasible computation and training times. As some of these issues could be simply solved with enough computing power, we deliberately consider the case of limited computational resources in this study, instead of utilising large computing facilities, which may not be accessible to a wide range of researchers. Thus, we will limit ourselves here to a single GPU configuration, that necessitates the development of more memory efficient algorithms. Additionally, to address the second issue we aim to improve reconstruction speed without compromising reconstruction quality.

To achieve this we propose a new approach for training learned iterative reconstruction methods that scales to demanding large-scale tomographic imaging problems. It is a multi-scale scheme that is motivated by the fact that the continuum forward operator can be discretised on various scales. In fact, the ray transform is known to be scale invariant [28], which defines the forward operator in CT, and this consistency across scales can be utilised for reconstruction [29], [30]. In particular, in our case each unrolled iterate in the network involves discretising the ray transform on a voxalised grid and the discretisation becomes increasingly fine as the unrolled iterates progress until the final resolution is achieved. Hence, the full high-resolution forward operator is only needed for the final unrolled iterate. Clearly, the approach is not limited to CT and readily applies to other tomographic modalities that involve the ray transform. Furthermore, it can be extended to any modality that arises as discretisation from a continuum model, such as MRI or even seismic imaging, in contrast to purely discrete problems.

This paper is structured as follows. In Section II we review common approaches for learned reconstructions and discuss possible limitations for large-scale applications. In Section III we introduce the notion of multi-scale schemes. In Section IV we extend the multi-scale scheme to a hybrid network and apply the proposed network to reconstruct from real CBCT measurements of an organic phantom in 3D. In the following Section V we discuss scalability and evaluate performance in comparison to other learned reconstruction methods in 2D for phantoms from human abdominal CT scans. In Section VI we discuss extensions and limitations of the proposed multi-scale approaches. Some final conclusions are presented in Section VII.

II. Learned reconstructions for tomographic imaging

In computed tomography we aim to reconstruct an image of the inside of a patient or object of interest from X-ray measurements. Mathematically, this reconstruction task is an inverse problem where we seek to recover the unknown absorption coefficient $f^* \in X$ (image) from measured photons $g \in Y$ at the sensor (projection data or sinogram) where

$$g = \mathcal{A}(f^*) + \delta g. \tag{1}$$

Here, $\mathcal{A}\colon X \to Y$ is the forward operator, that is assumed to be known, and models how data is generated in absence of noise; $\delta g \in Y$ denotes noise in the observation.

In the following we will assume that $\mathcal{A}$ is a linear operator whose sampling is given by the data acquisition geometry, such as the fan beam transform in 2D and cone beam in 3D.

Reconstruction is typically an ill-posed task, so one needs to use noise-robust inversion procedures. Either by direct reconstruction algorithms, such as filtered backprojection (FBP), or by iterative algorithms that solve a variational problem

$$\hat{f} := \operatorname*{arg\,min}_{f \geq 0} \bigl\{ \mathcal{D}(f; g) + \alpha \mathcal{R}(f) \bigr\}. \tag{2}$$

Here, $f \mapsto \mathcal{D}(f; g)$ measures the goodness of fit against data g, $f \mapsto \mathcal{R}(f)$ is a regularisation term that ensures stability, and $\alpha > 0$ is a weighting parameter that regulates the need for stability against the need to fit data. These methods tend to perform well, but are ultimately limited by the expressiveness of the hand-crafted regularisation term $\mathcal{R}\colon X \to \mathbb{R}$. Recently, several researchers have proposed to either combine direct reconstructions with a learning based post-processing or to learn an iterative algorithm. In the following we give a short overview of possible approaches that involve the model in the reconstruction process. Either once in Section II-A and hence rely more on the expressiveness of the learned network, or multiple times in Section II-B, which consequently increases the influence of the model in the reconstruction task.

A. Reconstruction and post-processing

A straightforward approach to use data driven methods in reconstruction is by post-processing an initial reconstruction. More precisely, let $\mathcal{A}^\dagger\colon Y \to X$ be an analytically known reconstruction operator that is proven to be robust. One can then train a convolutional neural network to remove reconstruction artefacts that arise from using $\mathcal{A}^\dagger$ [16], [17], [31]. These artefacts can be quite notable when data is highly noisy or under-sampled. The learned *inverse mapping* is then given as

$$\mathcal{A}^\dagger_\theta := \Lambda_\theta \circ \mathcal{A}^\dagger \,.$$

The advantage in this approach lies in the analytical knowledge of the reconstruction operator, and hence networks can be designed to exploit structure in reconstruction artefacts. For instance in spatio-temporal problems, if under-sampling artefacts are known to be incoherent in time, the network only needs to learn to combine the spatial information by a temporal interpolation [32], [33]. On the other hand, for lower dimensional problems, the capacity of the network is essentially limited by the richness of the training data [34], [35]. Clearly such an approach is computationally fast since it only requires a single operator evaluation. On the downside, large capacity networks tend to over-fit to the training data and especially so when the training data is scarce. Furthermore, as shown in [14], [19], [20], [36] the results are clearly outperformed by learned iterative reconstruction algorithms that we next describe.

B. Learned iterative reconstructions

In learned iterative reconstruction schemes, neural networks are interlaced with evaluations of the forward operator $\mathcal{A}$, its adjoint $\mathcal{A}^*$, and possibly other hand-crafted operators. For example, a simple learned gradient-like scheme [14], [37] would be given by

$$f_{i+1} = \Lambda_{\theta_i}\bigl(f_i, \mathcal{A}^*(\mathcal{A}(f_i) - g)\bigr), \; i = 0, \ldots, N-1. \tag{3}$$

This defines a reconstruction operator when stopped after N iterates:

$$\mathcal{A}^\dagger_\theta(g) := f_N \quad \text{where } \theta = (\theta_0, \ldots, \theta_{N-1})$$

and initialisation $f_0 = \mathcal{A}^\dagger(g)$. Note that Λ_{θ_i} is a *learned updating operator* for the i:th iterate. The terminology 'gradient-like' comes from the following observation: if we consider minimising $\mathcal{D}(f; g) = \frac{1}{2}\bigl\|\mathcal{A}(f) - g\bigr\|_2^2$, then $\Lambda_\theta(f, h) := f - \theta h$ corresponds to a learned update in a gradient descent scheme, where the step length θ is the only learned parameter.

The parameters θ in the reconstruction operator $\mathcal{A}^\dagger_\theta$ are learned by end-to-end supervised training. More precisely, assume one has access to supervised training data $(f^{(j)}, g^{(j)}) \in X \times Y$ where $g^{(j)} \approx \mathcal{A}(f^{(j)})$. Then an optimal parameter is found by

$$\min_\theta \frac{1}{m} \sum_{j=1}^{m} \mathrm{L}_\theta(f^{(j)}, g^{(j)})$$

where the loss function is given as

$$\mathrm{L}_\theta(f, g) := \bigl\|\mathcal{A}^\dagger_\theta(g) - f\bigr\|_X^2 \quad \text{for } (f, g) \in X \times Y.$$

Note here that computing the gradient of the loss function w.r.t. θ requires performing back-propagation through all of the unrolled iterates $i = 0, \ldots, N-1$.

In gradient boosting, that follow the greedy training [20], the loss function is changed. Instead of looking for a reconstruction operator that is optimal end-to-end, we only require iterate-wise optimality. For the learned gradient scheme above, this amounts to the following loss function for the i:th unrolled iterate:

$$\mathrm{L}_{\theta_i}(f_i, g) = \Bigl\|\Lambda_{\theta_i}\bigl(f_i, \mathcal{A}^*(\mathcal{A}(f_i) - g)\bigr) - f\Bigr\|_X^2$$

where $f_i := \Lambda_{\theta_{i-1}}\bigl(f_{i-1}, \mathcal{A}^*(\mathcal{A}(f_{i-1}) - g)\bigr)$ and initialisation $f_0 = \mathcal{A}^\dagger(g)$. These schemes can be viewed as a greedy approach and consequently constitute an upper bound to end-to-end networks. Thus, in the following we seek for a possibility to utilise end-to-end networks for large-scale problems.

III. Multi-scale learned iterative reconstructions

The major limitations when employing learned iterative reconstruction methods for large problems are their prohibitive training times and memory requirements. This is mainly due to the fact that all iterations are performed at full resolution and hence require to evaluate the full scale forward operator for each iterate. To overcome this limitation we propose a multi-scale scheme.

A. Discretisation sequence

In the inverse problem in eq. (1), both the unknown image f^* and data g are considered as continuum objects, which in imaging are typically represented by real-valued functions defined on some domains. In reality discrete data is recorded through a measurement device and we can only compute a digitised version of the unknown f^*. By discretisation we refer loosely to the procedure for defining a finite dimensional version of eq. (1) that is given by the finite sampling of the data and the digitisation of f^*. Likewise, a *discretisation sequence* is a finite sequence of discretisations that start from a coarse discretisation and is successively refined towards the desired finest resolution. The refinement and coarsening of the discretisation is through specific up- and down-sampling schemes that will be defined later. Consequently, motivated by the discretisation invariance of the ray transform, we aim to iteratively increase the resolution of our reconstructions. For that purpose, let $S_0, \dots, S_N$ denote a fixed sequence of discretisations of X and Y that increase in resolution through subsequent up-sampling. In the following we will associate each iterate f_i with such a discretisation space S_i. Stated more formally, a discretisation sequence is given by

$$S_i := X_i \times Y_i \quad \text{for } i = 0, \dots, N.$$

Here, $X_i \subset X$ is a finite dimensional subspace with dimension $\dim(X_i) \leq \dim(X_{i+1})$. Likewise, $Y_i \subset Y$ with $\dim(Y_i) \leq \dim(Y_{i+1})$. Furthermore, let $\{f_i, g_i\} \in S_i$ denote the reconstructed image and data in each discretisation space. In the following we will need a projection operator in the data space $\pi_i \colon Y \to Y_i$, for $i = 0, \dots, N$, and an up-sampling operator in the image space $\tau_i \colon X_{i-1} \to X_i$, for $i = 1, \dots, N$. Whereas the projection operator maps the data into the respective discretisation space, the up-sampling operator maps the reconstruction in the i:th discretisation space to the subsequent one in the discretisation sequence. Note that if $\dim(X_{i-1}) = \dim(X_i)$, then the up-sampling reduces to the identity $\tau_i = \mathbf{id}$.

The discretisation sequence $S_0, \dots, S_N$ defines as well a sequence of discretised versions of the inverse problem in eq. (1). More precisely, for each discretisation S_i we obtain the corresponding inverse problem of recovering $f_i^* \in X_i$ from finitely sampled data $g_i \in Y_i$ where

$$g_i = \mathcal{A}_i(f_i^*) + \delta g_i$$

with δg_i denoting the noise in data and $\mathcal{A}_i \colon X_i \to Y_i$ denoting the corresponding forward operator. Similarly, we have $\mathcal{A}_i^* \colon Y_i \to X_i$ for the adjoint and $\mathcal{A}_i^\dagger \colon Y_i \to X_i$ for the pseudo-inverse on the discretisation space S_i, e.g. the filtered backprojection in 2D or FDK in 3D. With these concepts we can now formulate the multi-scale iterative reconstructions schemes.

B. A multi-scale learned gradient scheme

The underlying principle of the proposed multi-scale scheme is to start at the coarsest discretisation space S_0 and after each iterate we up-sample until we obtain the reconstruction in the final discretisation space in the desired full-resolution. This way each iterate has its own discretisation space and hence the number of iterations we perform is $N + 1$, equal to the number of discretisation spaces. Since we aim to train the algorithm end-to-end, this maximum number of iterations has to be fixed. For each iterate we then compute the gradient in the corresponding discretisation space $\nabla \mathcal{D}_i(f_i; g) \in S_i$ given by

$$\nabla \mathcal{D}_i(f_i; g) := \mathcal{A}_i^*\big(\mathcal{A}_i(f_i) - \pi_i(g)\big). \tag{4}$$

Following the structure of learned gradient schemes eq. (3), we perform a learned update with the current reconstruction f_i and the corresponding gradient $\mathcal{D}_i(f_i; g)$, followed by an up-sampling to the next finer resolution,

$$\begin{cases} f_i = \Lambda_{\theta_i}\big(\widetilde{f}_i, \nabla \mathcal{D}_i(\widetilde{f}_i; g)\big) \\ \widetilde{f}_{i+1} = \tau_{i+1}(f_i). \end{cases}$$

The full MS-LGS is summarised in algorithm 1 and a schematic is illustrated in fig. 1.

Algorithm 1 Multi-scale learned gradient schemes (MS-LGS)

```
for $i = 0, \dots, N$ do
    if $i = 0$ then
        $\widetilde{f}_0 \leftarrow \mathcal{A}_0^\dagger \pi_0(g)$
    else
        $\widetilde{f}_i \leftarrow \tau_i(f_{i-1})$
    end if
    $f_i \leftarrow \Lambda_{\theta_i}\left(\widetilde{f}_i, \nabla \mathcal{D}_i(\widetilde{f}_i; g)\right)$
end for
$f^* \leftarrow f_N$
```

1) Including a filtered gradient: Let us first note, that the up-sampling operator in each iteration restricts the high frequency components that can be present after up-sampling. Additionally, the normal operator $\mathcal{A}^* \mathcal{A}$ is known to be smoothing of order 1 [28], which means, effectively, that any high frequency components in the final reconstruction can only be introduced by the network, similarly to the role of the regulariser in classical variational techniques. Thus, to complement the information for the network, we consider a version of MS-LGS with an additional filtered gradient that retains higher frequencies. That means we do not only compute the classic gradient $\nabla \mathcal{D}_i(f_i; g)$ in each iteration, but additionally a filtered version by substituting the adjoint with the filtered backprojection, or FDK in 3D,

$$\nabla^\dagger \mathcal{D}_i(f_i; g) := \mathcal{A}_i^\dagger\big(\mathcal{A}_i(f_i) - \pi_i(g)\big). \tag{5}$$

A similar approach has been studied earlier for classic iterative methods in [38]. In our case the filtered gradient will be computed additionally to the classic gradient eq. (4) and hence this will increase the computational cost by the application of one filtered backprojection in each step, but, as can be

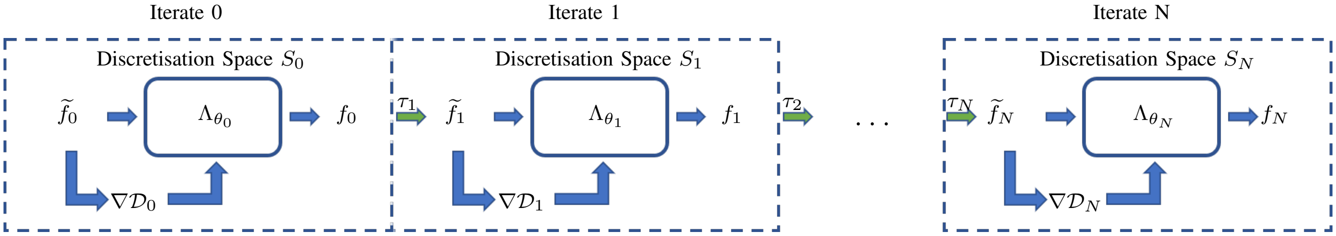

Fig. 1: Visualisation of the MS-LGS as outlined in algorithm 1. Each iteration is performed on their respective discretisation space, where the gradient $\nabla\mathcal{D}_i := \nabla\mathcal{D}_i(f_i; g)$ is computed and the update is performed by the network Λ_{θ_i}. After each update an up-sampling by τ_i to the next finer space is performed until the final resolution S_N is achieved.

seen later, improves reconstruction quality. For notational convenience, we will denote the set of inputs to the network in each scale by

$$[\widetilde{f}_i] := \left\{\widetilde{f}_i, \nabla\mathcal{D}_i(\widetilde{f}_i; g), \nabla^\dagger\mathcal{D}_i(\widetilde{f}_i; g)\right\}. \tag{6}$$

In the resulting scheme, multi-scale learned filtered gradient schemes (MS-LFGS), with the additional computation of the filtered gradient we then have the update equations

$$f_i \leftarrow \Lambda_{\theta_i}([\widetilde{f}_i])$$

instead of line 7 in algorithm 1.

2) Computational cost: Concerning the total computational cost: Due to sub-sampling on the coarser discretisation spaces the computation of projections is essentially governed by the computations on the final resolution. If we assume that the computational cost of evaluating the network Λ_{θ_i} is negligible in comparison to the forward and adjoint operator (or pseudo-inverse), then the total computational complexity is governed by the cost of the operator at the finest scale.

Formally, the total computational cost can be roughly estimated as follows. Let us assume that at each scale we double each dimension, then the number of voxels scale by 2^d. Thus, the computational cost on each scale increases in the same manner and the estimated total computational cost on all scales can be bounded by a geometric series

$$C_d := \sum_{k=0}^{\infty}\left(\frac{1}{2^d}\right)^k = \frac{1}{1-1/2^d}. \tag{7}$$

For $d = 2$ we have $C_2 = 4/3$ and $C_3 = 8/7$ for $d = 3$. We note, that the same estimate applies to memory requirements of the multi-scale scheme. This emphasises that the proposed approach is especially suitable for higher dimensional applications, since the computational cost on the course discretisation spaces becomes neglectable, as we will see in the next section for an application to 3D cone beam CT.

IV. Reconstruction of 3D cone beam measurements

Let us now discuss the reconstruction task from three dimensional cone beam measurements. We note that due to the structure of the multi-scale approaches, the reconstruction quality will essentially depend on the expressibility of the last layer and hence it is only reasonable to make the last iterate as informative as possible. To achieve scalability with an expressive network at the last iterate, we propose to combine MS-LFGS, as described in Section III-B1, with the established U-Net architecture [39] with the addition that the gradient information is reused in each scale of U-Net. This network is specifically designed to utilise the previously computed information across all-scales.

A. Cone beam measurement data

We evaluate the applicability of the proposed networks to reconstructions in 3D with an application to CBCT. For this purpose we utilise a database provided by the FleX-ray lab at Centrum Wiskunde & Informatica [40], consisting of 42 walnuts scanned in a custom made μCT. For each target there are 3 separate scans consisting of 1201 angles with uniform increment of 0.3° and varying source locations at the top, middle, and bottom of the target. That is for a mean target size of 30mm the scanning positions are at -15mm, 0, 15mm with respect to the central slice. As these three scans result in different cone beam artefacts, they are combined to create a reference ground-truth reconstruction of size 501^3 negating the cone beam artefacts. We refer to [40] for further details on the scanning setup and geometry.

For our experiments we utilise the central scanning position at the centre of the target, with a source-to-target distance of 66mm and source-to-detector distance 199mm. We select 60 uniformly spaced angles, resulting in an angular increment of 6°. Additionally we down-sample both, measurements and ground-truth reconstruction, by a factor of 3. This results in a reconstruction size of 168^3, where each of the 60 projections is of size 256×324. We note that we chose the maximum reconstruction size possible under the memory constraints of this study.

The supplied data is given as linearised measurements, thus we will use the linear projection model as our forward operator

$$\mathcal{A}(f)(\ell) = \int_\ell f(x)\,\mathrm{d}x, \quad \text{for } \ell \in \mathcal{M}, \tag{8}$$

where $\mathcal{M}$ is the three dimensional manifold of lines in $\mathbb{R}^3$ defined by the cone beam measurement geometry described above.

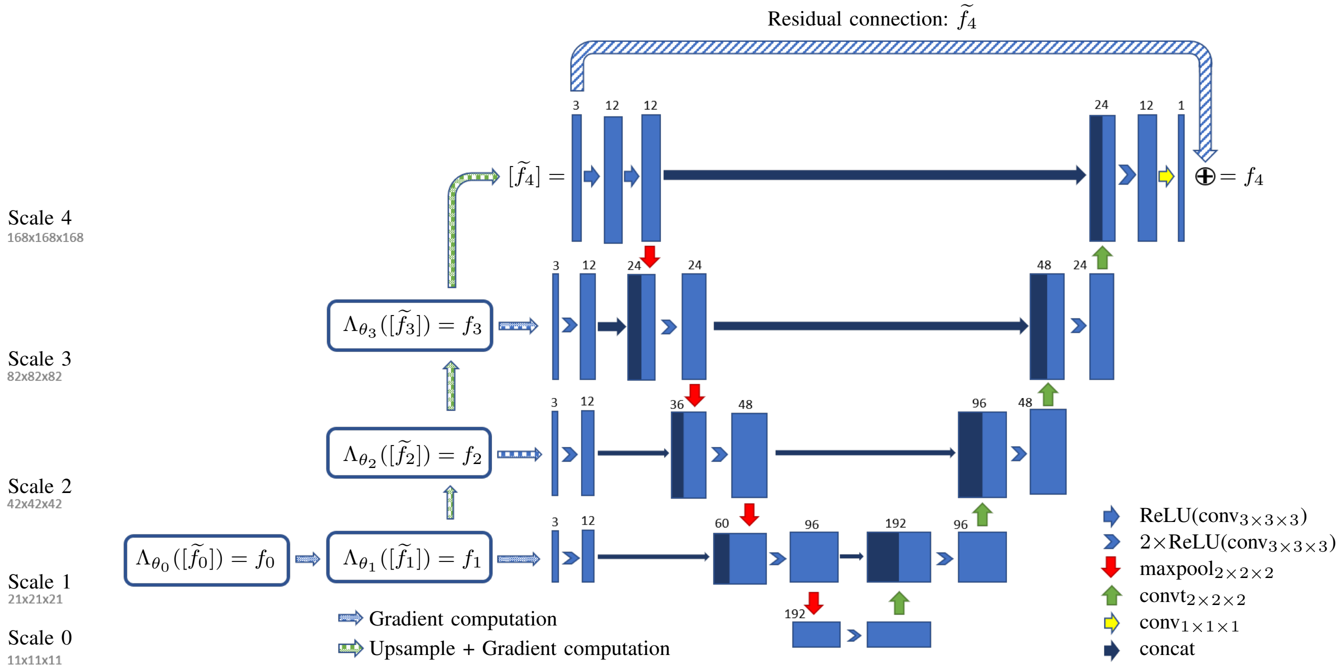
Residual connection: $\widetilde{f}_4$
$[\widetilde{f}_4] =$
$\oplus = f_4$
Scale 4
168x168x168
Scale 3
82x82x82
Scale 2
42x42x42
Scale 1
21x21x21
Scale 0
11x11x11
$\Lambda_{\theta_3}([\widetilde{f}_3]) = f_3$
$\Lambda_{\theta_2}([\widetilde{f}_2]) = f_2$
$\Lambda_{\theta_1}([\widetilde{f}_1]) = f_1$
$\Lambda_{\theta_0}([\widetilde{f}_0]) = f_0$
3
12
12
24
12
1
3
12
24
24
48
24
3
12
36
48
96
48
3
12
60
96
192
96
192
Gradient computation
Upsample + Gradient computation
ReLU(conv$_{3\times3\times3}$)
2×ReLU(conv$_{3\times3\times3}$)
maxpool$_{2\times2\times2}$
convt$_{2\times2\times2}$
conv$_{1\times1\times1}$
concat

Fig. 2: The proposed ∂U-Net architecture for multi-scale learned iterative reconstructions of CBCT reconstructions in 3D. The left part of the network consists of a MS-LFGS, which uses a U-Net on the right in the final iterate. Additionally, the output and corresponding gradient information of each iterate is re-used in the respective scale of the U-Net.

B. A hybrid multi-scale network: ∂U-Net

As the final reconstruction quality in the multi-scale scheme is primarily dependent on the last iterate operating on the final resolution, it is advisable to make this last iterate as expressive as possible without significantly increasing bias in the reconstructions. For this purpose we propose an across-scales network, that is essentially a combination of MS-LFGS and U-net that utilises the computed gradient information across all scales; in the following we will call this architecture ∂U-Net. Details of the network design are discussed next.

1) Implementation details: The resulting ∂U-Net architecture chosen for the application to CBCT is illustrated in Figure 2. We have chosen the number of iterates as $N+1 = 5$; for the corresponding discretisation spaces, we fix the resolution of the finest desired reconstruction space as $X_N = \mathbb{R}^{n\times n\times n}$, with $n = 168$. The coarser resolutions are then obtained by reducing the resolution for each downsampling by a factor of 2 in each dimension until scale 1, here scale 0 has the same resolution to avoid overfitting due to very small image sizes in the first iterate. Thus, the coarsest scale is obtained by 3 times downsampling, that is a factor of 8 per dimension and hence the total image size is reduced by a factor of 512. In the projection space, we keep the number of angles at 60 for each scale, but downsample the detector size by the same factor as the image size, *i.e.* reducing each dimension by factor 2 until scale 1.

The mapping π_i to the coarser scale is implemented by an area mean, the up-sampling with τ_i is performed by trilinear interpolation. After each network update, we compute the set of filtered and classical gradient as in eq. (6) for the current scale, that is

$$[f_i] = \left[\Lambda_{\theta_i}([\widetilde{f}_i])\right], \tag{9}$$

as well as the gradient set of the up-sampled output $\widetilde{f}_{i+1} = \tau_{i+1}(f_i)$. Where the former gradient set in eq. (9) is passed to U-Net in the respective scale, subsequently expanded by a double convolutional layer and then concatenated with the result of the max-pooling in U-Net, and the latter gradient set of the up-sampled output, i.e. $\widetilde{f}_{i+1} = \tau_{i+1}(f_i)$, is used for the next iterate in the gradient scheme. Here the sub-networks are given in a ResNet style following [14], [36]. Specifically, we chose a double convolutional layer with 12 channels and a final layer with 1 output channel. The output is then given by a residual update

$$\Lambda_{\theta_i}([\widetilde{f}_i]) = f_i + s_i \mathcal{G}_{\widetilde{\theta}_i}([\widetilde{f}_i]), \tag{10}$$

where $\mathcal{G}_{\widetilde{\theta}_i}$ denotes the chosen architecture for the updates, i.e. the the three convolutions, and s_i is a learnable step size initialised by 0 following [41]. The learnable parameters in each iterate are then given by $\theta_i = \{s_i, \widetilde{\theta}_i\}$.

All algorithms, including reference methods, are implemented in Python using PyTorch [42] for the networks. The image and projection spaces are implemented with ODL (Operator Discretization Library) [43] using ASTRA [44] as back-end for evaluating the ray transform and its adjoint. Training details and parameter choices will be stated in the following sections.

C. Reconstructions

Additionally to reconstructions with the proposed ∂U-net, we will compare the quality to reconstruction with FDK

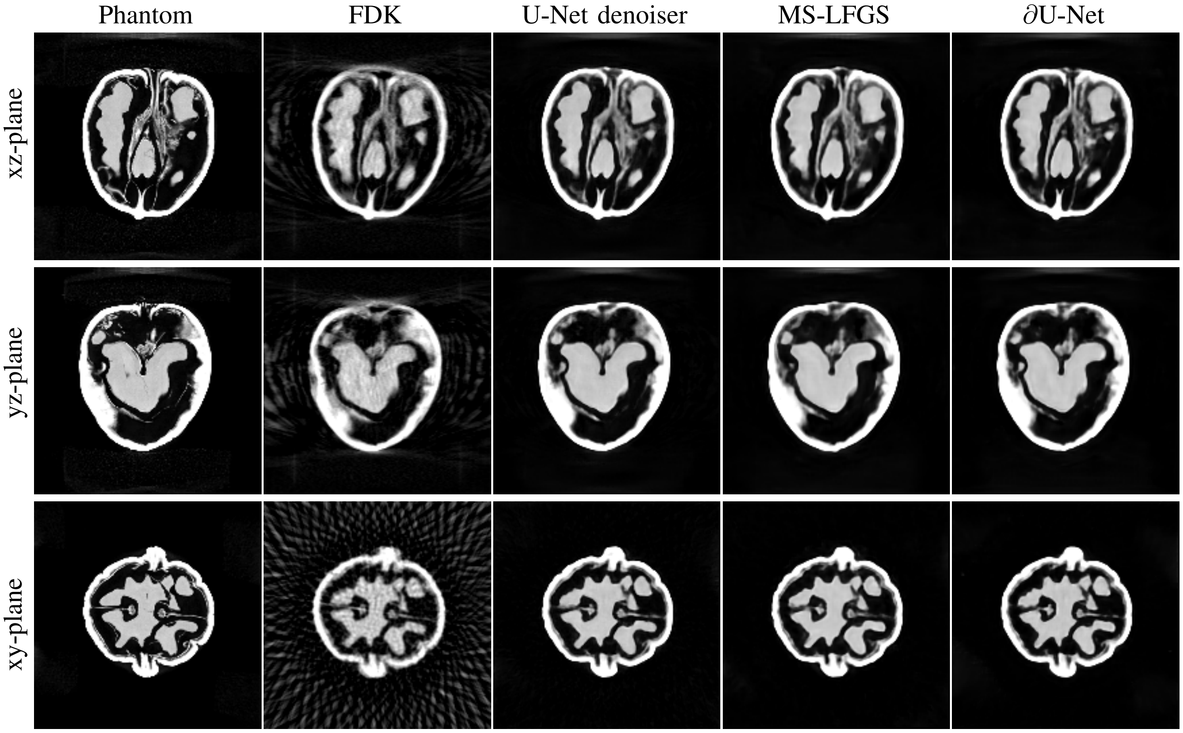

Fig. 3: Reconstructions of the Walnut used for testing from 60 angles and resolution 168^3. Reconstructions are compared to the phantom computed from a total of 1200 angles and three scanning positions to negate cone beam artefacts. The reconstruction by FDK is computed with Hann filter and frequency scaling $h = 0.6$ (PSNR=26.95). The proposed algorithm ∂U-Net (PSNR=34.69) is compared to post-processing by U-Net (PSNR=34.62) and the multi-scale approach MS-LFGS (PSNR=34.13) using a mini U-Net in each scale.

TABLE I: Quantitative measures and computational resources for CBCT reconstructions of the walnut data from 60 angles. Computed values are given for the test data in comparison to the ground-truth from full measurements. Additionally, we present estimated benchmark values for LGS.

	PSNR	SSIM	TRAIN	EXEC.	PARAMETER	MEMORY
FDK	26.95	0.424	∼1m	260ms	1	723MB
U-NET DENOISER	34.62	0.910	4h29m	528ms	$6.3{\cdot}10^6$	6097MB
MS-LFGS (RESNET)	32.98	0.878	5h04m	526ms	$2.4{\cdot}10^4$	2547MB
MS-LFGS (MINI U-NET)	34.13	0.903	7h02m	645ms	$2.1{\cdot}10^5$	4853MB
∂U-NET	**34.69**	**0.914**	7h28m	795ms	$4.1{\cdot}10^6$	5313MB
LGS (5 ITER., ESTIMATED)	–	–	∼25h	2.5s	$\sim 2 \cdot 10^5$	∼ 12500MB

followed by post-processing with U-Net, following [17], as well as a reconstruction with the basic MS-LFGS as described in Section III. We note that this is essentially an ablation study on how each part performs separately. The U-Net architecture follows the same scheme as outlined in Figure 2, with the difference that the initial channel width is 16 and doubled in each scale, leading to slightly more parameters. For MS-LFGS we chose two variants here, one that is based as well on a ResNet architecture as used in the ∂U-Net and a second variant, where all sub-networks $\mathcal{G}_{\widetilde{\theta}_i}$ in (10) are given by a down-scaled version of U-Net, which we call mini U-Net, similarly to what has been used in [27]. This mini U-Net consists of only 2 scales (one max-pool layer), instead of the classic 4, and an initial channel depth of 12 on the first scale to be conforming with the ∂U-Net. All updates in the iterate schemes are performed following the residual updates in eq. (10).

To make the comparison uniform for all test cases we performed training for all algorithms in the same manner. In particular we chose Adam as the optimiser with an ℓ^2-loss to the ground-truth; each network is trained for 10,000 iterations with one training sample per minimisation step. The initial learning rate is set to 10^{-3} with a cosine decay. These choices have shown to perform well for all presented algorithms.

For training we have chosen 40 out of the 42 walnuts, which leaves 2 for validation and testing. The obtained reconstructions for the test walnut (number 41) are shown in Figure 3. It can be seen, that all learned methods are capable of successfully suppressing the cone beam artefacts in comparison to the FDK reconstruction.

D. Quantitative results

Visually all three learned reconstructions perform well and produce an informative reconstruction from just 60 projection angles. To compare the reconstructions in more detail, we have computed quantitative measures shown in Table I, specifically PSNR and SSIM with respect to the provided ground-truth image. Additionally, we provide training and execution times for all algorithms, number of parameters and needed memory for evaluation of the trained network.

The results suggest that the basic multi-scale approach is not competitive in terms of PSNR and SSIM. As we have indicated earlier, this is most likely due to the limited expressiveness of the final network. This can be clearly seen by the comparison of MS-LFGS based on ResNet and the mini U-Net for each iterate, as increasing the depth of the networks improves reconstruction quality clearly. In particular, the proposed ∂U-Net, that combines the MS-LFGS architecture with a U-Net in the final iterate, improves reconstruction quality further and slightly outperforms the established post-processing and denoising by U-Net approach.

Concerning training and execution times, clearly U-net is fastest to train and execute, roughly taking double the time of FDK. It is noteworthy that the iterative approaches only add a slight overhead in execution time, where MS-LFGS using a ResNet structure is even faster. The most computationally expensive algorithm is ∂U-Net, but has only an overhead of 50% to the basic U-Net. This emphasises the excellent scalability of the multi-scale approaches in 3D.

In comparison, the basic LGS as described in Section II-B with 5 iterates and a ResNet structure would require roughly 5 times the resources, in terms of memory and computation times, see Table I for the estimated values. Clearly, one would not only need more computing power to train the algorithm, but also reconstruction times fall short to the multi-scale approaches

E. Robustness

Even though the reconstruction quality of ∂U-Net might only slightly outperform the denoising with U-Net, it provides a scalable model-based iterative reconstruction technique. This is of particular importance for applications where training data is scarce and objects might vary, as model-based iterative reconstructions have been shown to be more robust with respect to perturbations in the data and geometry, as demonstrated in several studies [20], [23], [24], see also [25] for a theoretical discussion. To emphasise this point, we have performed a robustness study with respect to noise. As the training data was given by real projection data it contained a natural noise component. For the robustness study, we have added additional normally distributed noise to the projection data for the test set and recorded the PSNR values of the reconstructions. The results of this experiment are illustrated in Figure 4. It can be seen that all model-based iterative approaches are more robust with respect to additional noise, whereas post-processing with U-Net does deteriorate much quicker. It is also interesting to note, that ∂U-net does show similar robustness as the MS-LFGS approaches, but under higher noise starts to deteriorate also a bit faster, which can be expected as it is a hybrid network combining both approaches.

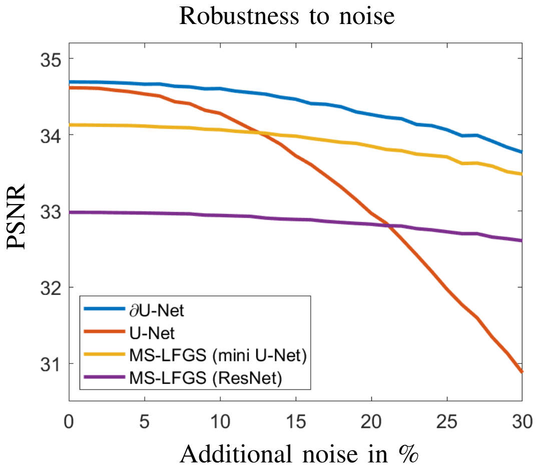

Fig. 4: Robustness study with respect to additional noise in the test data. Specifically, normally distributed random noise is added to the projection data and reconstruction quality is evaluated for all algorithms under consideration.

V. Comparative study in 2D

In this section we aim to evaluate the performance of the proposed ∂U-Net and multi-scale schemes in comparison to learned gradient schemes as in [14] that operate in each iterate on the full resolution. As these approaches do not scale well to 3D we restrict ourselves here to two dimensions. We will first examine scalability on simulated data and then evaluate reconstruction performance with realistically generated data from human phantoms supplied for the 2016 AAPM Low Dose CT Grand Challenge.

A. Implementation

Let us first discuss the implementation choices for the multi-scale schemes. As in the previous section, we fix the number of iterations to $N+1=5$. To create the discretisation spaces, we fix the resolution of the finest desired reconstruction space as $X_N = \mathbb{R}^{n\times n}$. The coarser resolutions are then obtained by reducing the resolution for each downsampling by a factor of 2 in each dimension. That means, the coarsest scale is obtained by 4 times downsampling which reduces the data size in 2D by a factor of 256. In this part, we reduce the amount of angles by a factor of 2 as well, the projection resolution is determined for each scale separately to fully cover the domain. Following the study in 3D, the mapping π_i to the coarser scale is implemented by an area mean, whereas the up-sampling with τ_i is performed here by bilinear interpolation.

We will restrict the network architectures in this section to learned gradient schemes with a mini U-net as the sub-network. As this choice has shown to be more competitive for the reconstruction of the walnut data in 3D. For the ∂U-net, we follow the architecture outlined in Figure 2, where

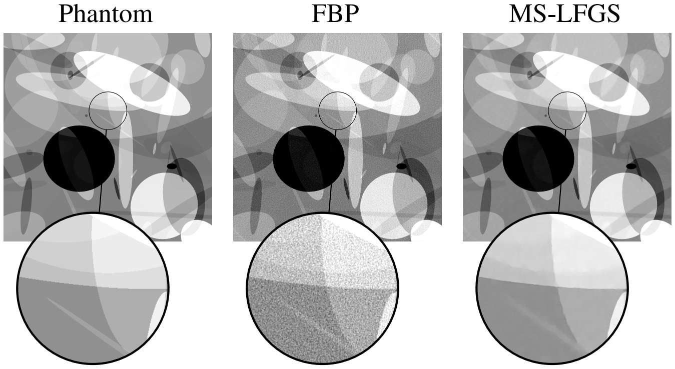

Fig. 5: Reconstruction of an ellipse phantom of size 1536^2 from 512 angles with 5% normally distributed random noise. (Left) Phantom used to create the data, (Middle) reconstruction by filtered backprojection, (Right) obtained reconstruction with MS-LFGS.

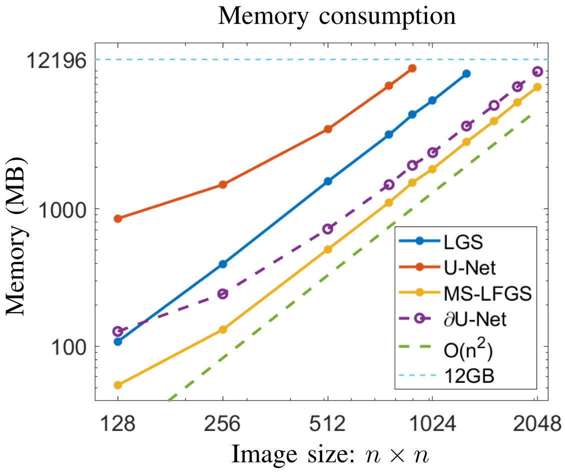

Fig. 6: Memory consumption in the training phase of proposed algorithms and reference learned methods for simulated data in 2D of increasing size. Maximal available memory on the GPU was 12196MB.

we adjust the channel width to 16 in the first scale of the U-Net, this also applies to the sub-networks used in the iterative multi-scale part.

B. Memory scaling of reconstruction algorithms

Let us first examine the scalability in terms of memory footprint of the proposed multi-scale algorithms in comparison to reference learned reconstruction methods. For comparison we choose post-processing with U-Net, following [17] with initial channel width of 64, and LGS [14]. Here LGS is implemented consistent with the proposed MS-LFGS algorithm, that means we use 5 iterations and a mini U-Net for the sub-networks. In fact, we note that this can be seen as a subclass of MS-LGS, where all discretisation spaces are of the same resolution and the scaling operators are given by the identity.

For the training procedure we created phantoms by randomly generated ellipses, see Figure 5. The measurement data is then produced by the ray transform eq. (8), with a fan beam geometry and 512 angles. The simulated measurement is then corrupted by additional 5% of normally distributed random noise.

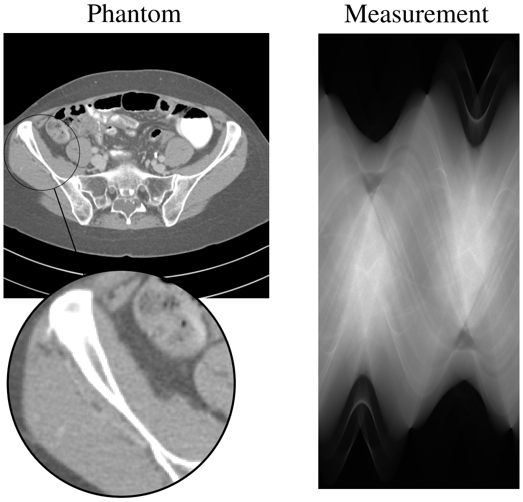

Fig. 7: Sample slice from the test patient windowed to $[-300, 300]$HU and the corresponding measurement data from 600 angles with a mean of 8000 photon counts.

Since the aim of this experiment is to examine memory consumption only, we have trained each network for 1000 iterations with one sample in each iteration and recorded the maximum memory consumption. The smallest phantom size was chosen as 128^2 and was increased until memory consumption exceeded the available memory on a single GPU with 12GB memory, or more specifically 12196 MB. The resulting plot is shown in Figure 6.

We note that memory consumption of all networks scales with $O(n^d)$, where d is the dimension. A reduction in memory consumption can be mainly achieved by usage of smaller networks and as such reduction by a constant. Nevertheless, memory consumption of LGS depends on the number of iterations also, i.e. we have $O(Nn^d)$. Following Section III-B2, for multi-scale approaches this iteration dependence can be bounded as well by the factor C_d in (7) and thus we obtain the basic memory dependence of $O(n^d)$.

A reconstruction obtained with MS-LFGS for a resolution of 1536^2 is shown in Figure 5 in comparison to a reconstruction by filtered backprojection.

C. Application to human CT scans

In order to evaluate the reconstruction quality on a clinically relevant case, we simulate realistic measurement data from human abdomen CT scans provided by the Mayo Clinic for the 2016 AAPM Low Dose CT Grand Challenge [45]. The data set consists of high-dose scans from 10 patients. We used the provided reconstructions with 3 mm slice thickness and image size 512×512. We divided the data into 9 patients for training, resulting in 2168 slices, and 1 patient for testing purposes with 210 slices.

For the data simulation, we used a fan beam geometry with source to axis distance 500 mm and axis to detector distance

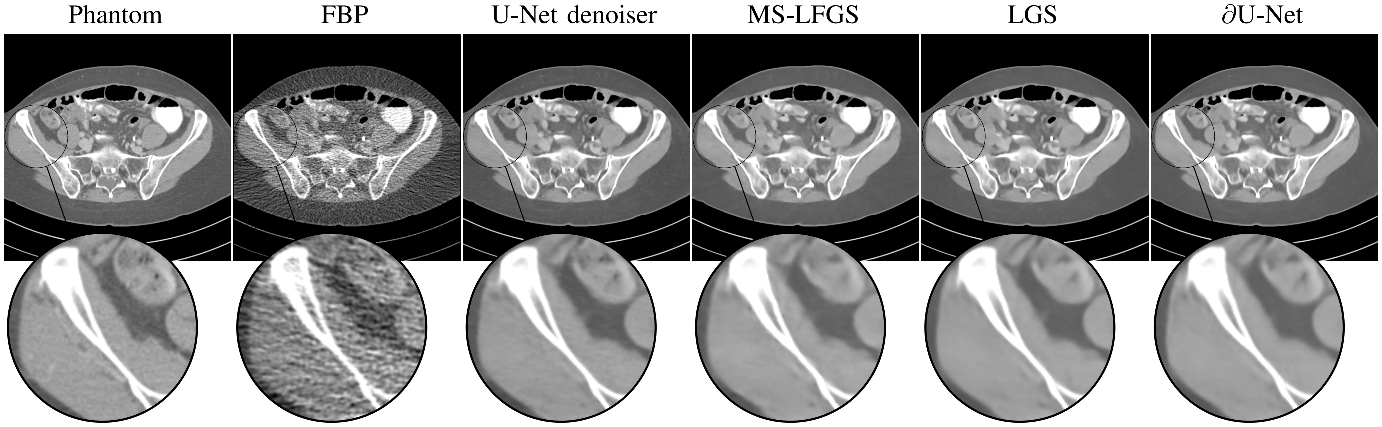

Fig. 8: Reconstructions of the test patient for measurement case 1 with 600 angles. All images are windowed and displayed on $[-300, 300]$HU. The filtered backprojection here is computed with $h = 0.4$.

500 mm. In order to create realistic measurement data, we use the non-linear forward model given by the Beer-Lamberts law:

$$\mathcal{A}(f)(\ell) = e^{-\mu \int_\ell f(x) \, \mathrm{d}x}.$$

Here, ℓ denotes the line along which the x-ray photons travel and we select the mass attenuation coefficient $\mu = 0.2$ cm^2/g, which corresponds approximately to the value of water. We simulate low dose scans with Poisson noise in the measurement data. For the computations we linearise the obtained data by applying $-\log(\cdot)/\mu$ to the measurements, by which the forward model simplifies to the ray-transform as in eq. (8). A slice from the test patient with the corresponding measurement data is shown in Figure 7.

We remind that we chose the number of iterations as $N + 1 = 5$ and hence the image resolution in the coarsest discretisation space S_0 is just 32×32. For the experiments we consider a scenario that roughly represents a clinical low-dose CT scan with 600 angles and a photon count of 8000.

1) Training procedure for low dose scans: We train both multi-scale schemes as outlined in Section III, the proposed hybrid ∂U-Net, as well as a full-scale learned gradient scheme (LGS) and post-processing with U-Net. In each case, we compute an initial reconstruction by filtered backprojection with the Hann filter and frequency scaling of $h = 0.6$, this reconstruction is also chosen as the input to the post-processing with U-Net. The same parameters are selected to compute the filtered gradient eq. (5) for the MS-LFGS.

To make the comparison uniform for all test cases we trained all algorithms in the same manner. In particular we chose Adam as the optimiser with an ℓ^2-loss; each network is trained for 20,000 iterations with one training sample per minimisation step. The initial learning rate is set to 10^{-3} with a cosine decay These choices have shown to perform well for all presented algorithms. In the following we will discuss the reconstruction results along with a quantitative evaluation.

D. Evaluation of reconstruction quality in 2D

The resulting reconstructions from 600 angles are shown in Figure 8. Let us first note that U-Net does generally produce sharper images than the learned approaches, but can tend to reconstruct artificial realistic looking features. All iterative approaches tend to produce smoother reconstructions, in particular we observe that in areas of uncertainty the learned approaches are more conservative in recreating features and rather tend to reconstruct uniform areas instead of reproducing features from the training data.

We have computed quantitative measures for all cases as shown in Table II. Comparing the multi-scale schemes, it is apparent here as well that the filtered gradient is necessary for competitive reconstruction quality. Overall, the proposed ∂U-Net does perform best of all algorithms, followed by LGS and then MS-LFGS. We note here, that it is expected that LGS performs better than both multi-scale schemes, as it operates on the full resolution in each iteration, but consequently does not scale very well. Nevertheless, the hybrid network ∂U-Net is capable of producing competetive results, while being scalable.

Regarding memory consumption, the multi-scale approaches are expected to be cheapest in terms of memory and training times. Whereas ∂U-Net clearly reduces memory consumption in comparison to LGS, we can see that here in 2D the training times are slightly longer, due to multiple filtered backprojections in the lower scales. We note that this effect is negated in 3D as seen in Table I, since computational complexity reduces by a factor of 8 on each scale in 3D instead of just 4 in 2D. It is also interesting to point out that MS-LGS is faster in execution times than filtered backprojection followed by U-Net, even though reconstruction quality might not be competitive this can be of use in highly time critical applications.

VI. Discussion

The presented framework for multi-scale learned iterative reconstructions in Section III provides a general framework for a scalable iterative learned image reconstruction. Combining these multi-scale schemes with a U-Net in the last iterate provides a hybrid network capable of outperforming the previously proposed LGS approaches. Nevertheless, as this study is primarily of methodological nature, we would like to

TABLE II: Quantitative measures for low dose scans along with benchmark results for each algorithm. Averaged over 210 slices of test patient. Mean values are shown with their standard deviation.

	PSNR	SSIM	TRAIN	EXEC.	PARAMETER	MEMORY
FBP	32.48 ±1.55	0.73 ±0.0612	∼10s	33ms	1	1477MB
LGS	**43.25 ±1.24**	**0.963 ±0.0032**	2h31m	149ms	128970	2229MB
U-NET DENOISER	42.76 ±1.52	0.960 ±0.0026	1h38m	67ms	$3.1{\cdot}10^7$	2733MB
MS-LGS	41.42 ±1.33	0.948±0.0041	1h42m	53ms	128970	1143MB
MS-LFGS	42.85 ±1.25	0.960 ±0.0034	2h07m	154ms	129690	1143MB
∂U-NET	**43.51 ±1.23**	**0.965 ±0.0032**	3h25m	224ms	$2.3{\cdot}10^6$	1351MB

discuss in the following a few aspects on how the presented framework can be extended.

A. The scalability issue

Recently, some efforts have been made to extend learned iterative reconstruction algorithms to 3D applications. These approaches mainly tackle the memory aspect of the scalability issue, which prevents scalability to higher dimensions by hardware restrictions. For instance, by using invertable networks [46] one does not need to store the whole network for computation of the gradient in the training. Whereas this solves the important issue of memory footprint, it does not address computational complexity of the forward operator and as such is primarily applicable to forward operators of low complexity, such as the Fast Fourier Transform used in magnetic resonance imaging. For computationally more expensive forward operators, scalability is essentially limited by extensive training times due to the evaluation of the model. The proposed multi-scale schemes provide a possible solution to this dilemma, as the model is only once evaluated on the full resolution. This is illustrated in Table III, where we present the order of computational resources needed for the discussed algorithms in this study. The multi-scale approach addresses both points of the scalability issue, memory footprint and computation times. In comparison to LGS, which additionally scales with number of iterations, the multi-scale approach reduces this to the order of a single iteration.

TABLE III: Scaling properties of discussed algorithms in terms of memory footprint and operator evaluations. Here, n is the image size, d is the image dimension (usually $d = 2, 3$), and N refers to number of unrolled iterations in learned iterative schemes.

	MEMORY	OPERATOR EVAL.
FBP	$O(n^d)$	1
U-NET DENOISER	$O(n^d)$	1
LGS	$O(Nn^d)$	$O(N)$
MULTI-SCALE	$O(n^d)$	$O(1)$

In fact, the multi-scale schemes showcase their strength especially in higher dimensions as the reduced evaluation cost scales with the dimension. This can be clearly seen when comparing the study in 2D and 3D as presented here. For instance, the hybrid ∂U-Net compared to the basic U-Net has an overhead in evaluation time of roughly 300% in 2D, this reduces to only 50% in 3D. Which underlines the suitability of the proposed ∂U-Net for higher dimensional applications.

B. Influence of scales

As discussed above, the computational advantage of the multiscale approach is primarily due to the low-cost computations on the coarse resolution, but these come with some subtleties. We want to note that the early iterates on low resolutions are prone to overfitting and can negatively influence the reconstruction quality on the following iterates. Thus, one has to be careful to appropriately deal with the low resolution iterates For instance, the ResNet structure chosen in ∂U-Net for the iterative part is more resilient than the mini U-Net. The computed updates in each iterate in the ∂U-Net for the walnut reconstructions are shown in Figure 9 and as it can be seen the reconstructions nicely gain sharpness in each iterate until the final reconstruction is achieved.

For the pure multi-scale schemes, as used in MS-LFGS, instead of using only the mini U-Net, one could also consider mixing architectures, especially in the low resolution iterates changing to the ResNet structure for more stability. This has been omitted from this study for the sake of brevity.

C. Extensions of the multi-scale approach

In this study we have chosen the structure of the multi-scale algorithms as simplistic as possible. Nevertheless, the proposed framework does offer larger flexibility in choices that might be more suitable for other applications. In particular with respect to network design and choice of discretisation spaces. In the following we would like to mention some possibilities how the multi-scale schemes can be extended:

- In our study the mini U-Net has shown to be effective to restore high-frequency components more effectively than a basic ResNet style CNN as utilised in [14]. We note that also more memory efficient networks might be used, such as the MS-D Net [47] or, as mentioned above, invertable architectures. Possible extensions of the ∂U-Net to other architectures based on dilated convolutions instead of pooling layers can be investigated as well.
- In the multi-scale schemes we have chosen to identify each discretisation space with one iteration. This limitation can be easily relaxed, for instance by computing two iterations in the same discretisation space, as done

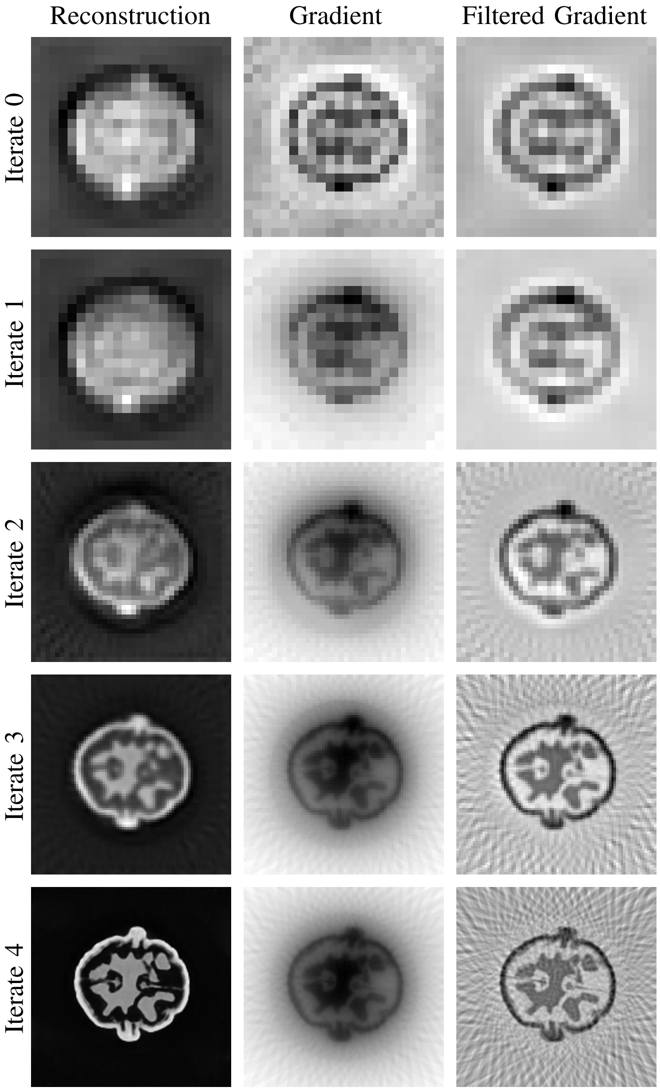

Fig. 9: Representation of the multi-scale scheme used in the ∂U-Net with obtained reconstruction, gradient, and filtered gradient on each discretisation space. Reconstructions obtained for the test walnut from 60 angles, here we show the middle slice in the xy-plane.

for the ∂U-Net. In case all iterates are computed on the same space, this simplifies to the basic LGS.

- We have chosen to reduce the resolution in all dimensions equally. It would be also possible to only reduce the resolution along one dimension in each step and alternate in dimensions. Along the same lines, the up-sampling operator can be chosen differently, including the possibility of a learned up-sampling.
- Lastly, the multi-scale framework is not limited to learned gradient schemes and can be extended to other learned approaches such as variational networks [19] and learned primal-dual [36].

VII. Conclusions

We have presented a general framework for scalable learned iterative reconstruction algorithms for large-scale problems and higher dimensions, by restricting the expensive computation of the full resolution forward operator to only one application in the final reconstruction space and as such reduces computation times as well as memory footprint of the learned iterative scheme. This multi-scale approach is especially powerful in higher dimensions, such as 3D, where the computational cost of the early iterates is negligible. We have presented two methods to obtain such a scalable learned iterative reconstruction, a basic multi-scale learned (filtered) gradient scheme based on the previous work [14] as well as hybrid model-based iterative network combined with U-Net, that reuses previously computed gradients on each scale in the respective U-Net scales.

The presented algorithms are evaluated by reconstructing 3D volumes of walnuts from real measurements, successfully demonstrating scalability of model-based iterative reconstructions to higher dimensions for non-trivial forward operators. The proposed architectures produce competitive results compared to post-processing with U-Net with an increasing robustness due to the iterative model-based nature of the methods. Additionally, we have evaluated the proposed algorithms in 2D in comparison to an established learned gradient scheme, that does not provide easy scalability.

Whereas this work is primarily a methodological study, we believe that it will be of high relevance to applications where high dimensionality of the imaging problem is inherent with computationally demanding forward operators, such as it is the case in cone beam CT.

Acknowledgment

We acknowledge the support of NVIDIA Corporation with one Titan Xp GPU. 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 in 2D using a human phantom.

Codes are available at: https://github.com/asHauptmann/multiscale

References

[1] L. A. Feldkamp, L. C. Davis, and J. W. Kress. Practical cone-beam algorithm. *Journal of the Optical Society of America A*, 1(6):612–619, 1984.

[2] W. Stiller. Basics of iterative reconstruction methods in computed tomography: A vendor-independent overview. *European Journal of Radiology*, 109:147–154, 2018.

[3] L. L. Geyer, J. U. Schoepf, F. G. Meinel, J. W. Nance, G. Bastarrika, J. A. Leipsic, N. S. Paul, M. Rengo, A. Laghi, and C. N. De Cecco. State of the art: Iterative CT reconstruction techniques. *Radiology*, 276(2):339–357, 2015.

[4] E. Y Sidky, J. H. Jørgensen, and X. Pan. Convex optimization problem prototyping for image reconstruction in computed tomography with the Chambolle–Pock algorithm. *Physics in Medicine & Biology*, 57(10):3065–3091, 2012.

[5] S. Arridge, P. Maass, O. Öktem, and C.-B. Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019.

[6] J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, volume 160 of *Applied Mathematical Sciences*. Springer Verlag, 2004.

[7] S. Siltanen, V. Kolehmainen, S. Järvenpää, J. P. Kaipio, P. Koistinen, M. Lassas, J. Pirttilä, and E. Somersalo. Statistical inversion for medical X-ray tomography with few radiographs: I. General theory. *Physics in Medicine & Biology*, 48(10):1437–1463, 2003.

[8] V. Kolehmainen, S. Siltanen, S. Järvenpää, J. P. Kaipio, P. Koistinen, M. Lassas, J. Pirttilä, and E. Somersalo. Statistical inversion for medical X-ray tomography with few radiographs: II. Application to dental radiology. *Physics in Medicine & Biology*, 48(10):1465–1490, 2003.

[9] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[10] K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.

[11] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*, 3(3):253–276, 2010.

[12] M. Rantala, S. Vanska, S. Jarvenpaa, M. Kalke, M. Lassas, J. Moberg, and S. Siltanen. Wavelet-based reconstruction for limited-angle X-ray tomography. *IEEE transactions on medical imaging*, 25(2):210–217, 2006.

[13] T. A. Bubba, F. Porta, G. Zanghirati, and S. Bonettini. A nonsmooth regularization approach based on shearlets for Poisson noise removal in ROI tomography. *Applied Mathematics and Computation*, 318:131–152, 2018.

[14] J. Adler and O. Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, 2017.

[15] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen. Image reconstruction by domain-transform manifold learning. *Nature*, 555(7697):487, 2018.

[16] E. Kang, J. Min, and J. C. Ye. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Medical Physics*, 44(10):360–375, 2017.

[17] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser. Deep convolutional neural network for inverse problems in imaging. *IEEE Transactions on Image Processing*, 26(9):4509–4522, 2017.

[18] J. C. Ye, Y. Han, and E. Cha. Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048, 2018.

[19] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic resonance in medicine*, 79(6):3055–3071, 2018.

[20] A. Hauptmann, F. Lucka, M. Betcke, N. Huynh, J. Adler, B. Cox, P. Beard, S. Ourselin, and S. Arridge. Model based learning for accelerated, limited-view 3D photoacoustic tomography. *IEEE Transactions on Medical Imaging*, 37(6):1382–1393, 2018.

[21] J. Schlemper, J. Caballero, J. V. Hajnal, A. Price, and D. Rueckert. A deep cascade of convolutional neural networks for MR image reconstruction. In *International Conference on Information Processing in Medical Imaging*, pages 647–658. Springer, 2017.

[22] H. Li, J. Schwab, S. Antholzer, and M. Haltmeier. NETT: Solving inverse problems with deep neural networks. *ArXiv preprint*, 1803.00092, 2018.

[23] Y. E. Boink, C. Brune, and S. Manohar. Robustness of a partially learned photoacoustic reconstruction algorithm. In *Photons Plus Ultrasound: Imaging and Sensing 2019 (SPIE BIOS 2019)*, volume 10878D. International Society for Optics and Photonics, 2019.

[24] Y. E. Boink, S. Manohar, and C. Brune. A partially learned algorithm for joint photoacoustic reconstruction and segmentation. *IEEE transactions on medical imaging*, 39(1):129–139, 2019.

[25] A. K. Maier, C. Syben, B. Stimpel, T. Würfl, M. Hoffmann, F. Schebesch, W. Fu, L. Mill, L. Kling, and S. Christiansen. Learning with known operators reduces maximum error bounds. *Nature machine intelligence*, 1(8):373–380, 2019.

[26] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *19th Conference on Neural Information Processing Systems (NIPS 2007), Vancouver, British Columbia, Canada*, pages 153–160, 2007.

[27] A. Hauptmann, B. Cox, F. Lucka, N. Huynh, M. Betcke, P. Beard, and S. Arridge. Approximate k-space models and deep learning for fast photoacoustic reconstruction. In *International Workshop on Machine Learning for Medical Image Reconstruction*, pages 103–111. Springer, 2018.

[28] F. Natterer. *The mathematics of computerized tomography*. SIAM, 2001.

[29] M. Lassas, E. Saksman, and S. Siltanen. Discretization-invariant Bayesian inversion and Besov space priors. *Inverse Problems and Imaging*, 3:87–122, 2009.

[30] Z. Purisha, J. Rimpeläinen, T. Bubba, and S. Siltanen. Controlled wavelet domain sparsity for x-ray tomography. *Measurement Science and Technology*, 29(1):014002, 2017.

[31] J. Schwab, S. Antholzer, and M. Haltmeier. Deep null space learning for inverse problems: convergence analysis and rates. *Inverse Problems*, 2018.

[32] A. Hauptmann, S. Arridge, F. Lucka, V. Muthurangu, and J. A. Steeden. Real-time cardiovascular MR with spatio-temporal artifact suppression using deep learning–proof of concept in congenital heart disease. *Magnetic resonance in medicine*, 81(2):1143–1156, 2019.

[33] A. Kofler, M. Dewey, T. Schaeffter, C. Wald, and C. Kolbitsch. Spatio-temporal deep learning-based undersampling artefact reduction for 2D Radial Cine MRI with limited training data. *IEEE transactions on medical imaging*, 39(3):703–717, 2019.

[34] S. J. Hamilton and A. Hauptmann. Deep D-bar: Real-time electrical impedance tomography imaging with deep neural networks. *IEEE transactions on medical imaging*, 37(10):2367–2377, 2018.

[35] S. J. Hamilton, A. Hänninen, A. Hauptmann, and V. Kolehmainen. Beltrami-net: domain independent deep D-bar learning for absolute imaging with electrical impedance tomography (a-EIT). *Physiological measurement*, 40(7):074002, 2019.

[36] J: Adler and O. Öktem. Learned primal-dual reconstruction. *IEEE transactions on medical imaging*, 37(6):1322–1332, 2018.

[37] P. Putzky and M. Welling. Recurrent inference machines for solving inverse problems. *ArXiv preprint*, 1706.04008, 2017.

[38] H. Gao. Fused analytical and iterative reconstruction (AIR) via modified proximal forward–backward splitting: a FDK-based iterative image reconstruction example for CBCT. *Physics in Medicine & Biology*, 61(19):7187–7204, 2016.

[39] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[40] H. Der Sarkissian, F. Lucka, M. van Eijnatten, G. Colacicco, S. B. Coban, and K. J. Batenburg. A cone-beam X-ray computed tomography data collection designed for machine learning. *Scientific data*, 6(1):1–8, 2019.

[41] Z. Hongyi, N. D. Yann, and M. Tengyu. Fixup Initialization: residual learning without normalization. In *International Conference on Learning Representations*, 2019.

[42] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.*, 2017.

[43] J. Adler, H. Kohr, and O. Öktem. Operator discretization library (ODL). *Software available from https://github. com/odlgroup/odl*, 2017.

[44] W. van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabravolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers. Fast and flexible x-ray tomography using the ASTRA toolbox. *Optics Express*, 24(22):25129–25147, 2016.

[45] C. McCollough. Tu-fg-207a-04: Overview of the low dose CT grand challenge. *Medical physics*, 43(6):3759–3760, 2016.

[46] P. Putzky and M. Welling. Invert to learn to invert. In *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada*, pages 444–454, 2019.

[47] D. M. Pelt and J. A. Sethian. A mixed-scale dense convolutional neural network for image analysis. *Proceedings of the National Academy of Sciences*, 115(2):254–259, 2018.