Projection-Domain Self-Supervision for Volumetric Helical CT Reconstruction

Onni Kosomaa, Samuli Laine, Tero Karras, Miika Aittala, and Jaakko Lehtinen

***Abstract*— We propose a deep learning method for three-dimensional reconstruction in low-dose helical cone-beam computed tomography. We reconstruct the volume directly, i.e., not from 2D slices, guaranteeing consistency along all axes. In a crucial step beyond prior work, we train our model in a self-supervised manner in the projection domain using noisy 2D projection data, without relying on 3D reference data or the output of a reference reconstruction method. This means the fidelity of our results is not limited by the quality and availability of such data. We evaluate our method on real helical cone-beam projections and simulated phantoms. Our reconstructions are sharper and less noisy than those of previous methods, and several decibels better in quantitative PSNR measurements. When applied to full-dose data, our method produces high-quality results orders of magnitude faster than iterative techniques.**

I. Introduction

COMPUTED tomography (CT) is a versatile medical imaging technique for producing tomographic images of body tissues from two-dimensional X-ray projections. In modern systems, the goal is to reconstruct a consistent three-dimensional volume instead of individual 2D slices, so that various cross-sections can be examined easily. For medical CT scans, the most popular mode of acquisition is moving a point-like radiation source and a 2D X-ray detector along a helical trajectory. Reconstructing tomographic images or 3D volumes from such helical cone-beam (CB) data is a difficult problem, and previous solutions typically resort to approximations such as re-binning the data (e.g., [1], [2]). As CT uses ionizing radiation, minimizing the dose is of paramount importance when operating with living subjects. Unfortunately, lowering the dose amplifies the noise in the X-ray images, which in turn makes reconstruction more difficult.

In this paper, we present a deep learning method for CBCT reconstruction. Our method closely resembles the weighted filtered backprojection (wFBP) algorithm [1], but with machine learning components introduced in crucial points to enable correction of errors introduced by noise and the finite number of input projections. Like wFBP, we directly produce a three-dimensional volume from a set of 2D X-ray projections, guaranteeing tomographic consistency along all axes. This improves the quality of coronal and sagittal cross-sections and facilitates automatic downstream tasks such as segmentation.

Machine learning models are built from parametric functions, such as neural networks, that are adjusted in an initial training stage to minimize the discrepancy between the model output and the desired result. In the CBCT context, this poses a dilemma: As we have no way to directly access the underlying ground truth, how can we measure the quality of our solution and train the model to an optimum? Existing solutions have significant shortcomings. First, training using synthetic data raises questions about validity due to the “domain gap” between training data and real test data. Second, using reference volumes reconstructed using another algorithm, perhaps from higher-quality inputs, limits the obtainable quality as the model learns to reproduce errors made by the reference method.

We address the supervision problem by training the model using data consistency in the projection domain as the loss function. This is enabled by using a differentiable CT simulator in the training loop: Intuitively, the output volume is good when real X-rays taken from the same volume — but not used as input to the model — look similar to simulated X-rays computed from the model output. This *self-supervised* training objective has the significant benefit that it requires no other reference data besides the noisy 2D projections. Consequently, the achievable output quality is not limited by the quality of, e.g., clean reference data or a reference reconstruction method, as is usually the case with previous machine learning-based reconstruction methods (e.g., [3], [4]). While projection consistency is often employed by iterative methods (e.g., [5], [6]), it has — to our knowledge — not been utilized as a training objective in learned end-to-end reconstruction methods.

We evaluate our method against several previous traditional and machine learning methods. We demonstrate sharper and less noisy results on real-world helical cone-beam data, and several decibels improvement over previous methods with synthetic phantoms where a ground truth volume is available. Although our main focus is on low-dose inputs, our method can be applied to full-dose data as well. In these cases, our method produce high-quality solutions orders of magnitude faster than iterative techniques.

Project page with supplemental results is available at `https://users.aalto.fi/~kosomao1/self-sup-ct/`

II. Previous work

Most of the widely used reconstruction methods are based on filtered backprojection (FBP) [7]. In FBP, the X-ray data is

Author preprint.

Onni Kosomaa is with Department of Computer Science, Aalto University, Espoo, Finland (e-mail: onni.kosomaa@aalto.fi). Part of the research was conducted during an internship with NVIDIA Corporation, Santa Clara, CA 95051. Jaakko Lehtinen is with Department of Computer Science, Aalto University, and NVIDIA Corporation. Samuli Laine, Tero Karras, and Miika Aittala are with NVIDIA Corporation.

filtered using a ramp filter, backprojected onto the reconstruction grid along straight lines, and averaged; the ramp filter is needed to counteract the overrepresentation of low frequencies that the average would otherwise exhibit. However, both the spiral acquisition trajectory and cone-beam geometry used in CBCT pose significant challenges. This is often solved by re-binning the projection data into other geometries, such as in weighted filtered backprojection (wFBP) [1]. Another large family of approaches iteratively calculates a least squares solution [5], [6] with a projection consistency loss. These often require hand-tuned regularizers or priors, such as total variation (TV), and are much slower than FBP-based methods.

Deep learning methods for CT can be roughly divided into post-processing, iterative, and end-to-end reconstruction methods. For a comprehensive survey, see Wang *et al.* [7].

1) Learned post-processing: Learned post-processing methods take an existing reconstruction as an input and seek to remove artifacts and noise from it. Most approaches involve supervised training of neural networks to minimize the mean squared error (MSE) between the network output and a full-dose FBP reference. Chen *et al.* [3] proposed a residual encoder-decoder convolutional neural network (RED-CNN) for improving FBP reconstructions. The CNN is executed slice-by-slice axially, without utilizing three-dimensional information. Zamyatin *et al.* [4] extended the residual network to 3D. Adversarial losses have been proposed to mitigate the blurring caused by minimizing MSE in the reconstruction domain. Yang *et al.* [8] utilized a two-dimensional generator and discriminator, and Wolterink *et al.* [9] extended the generator to 3D while keeping the discriminator two-dimensional. While adversarial methods produce sharper and more detailed results than supervised methods, there is a risk of introducing spurious features that are not actually present in the input data [10].

Overall, post-processing approaches share two key shortcomings. First, perfect reference volumes are rarely available, which limits the quality of the learned results. Second, these methods do not have full visibility to the information in the raw input projections; they only see the approximate 3D volume produced by the initial reconstruction method.

2) Learned iterative reconstruction: Iterative techniques have also been improved using deep learning via, e.g., learned priors [11] and unfolded iterative processes [12]. We focus on a non-iterative approach in search of higher runtime efficiency.

3) Learned end-to-end reconstruction: Deep learning methods that operate directly on the raw input projections have also been proposed. Most commonly, these methods train multiple neural networks in a supervised manner by minimizing per-voxel MSE compared to a FBP reference. Würfl *et al.* [13] proposed jointly learning the projection and volume domain weights for 3D reconstruction. By implementing the gradient of the backprojection operator as a projection, they train the weights end-to-end. However, they target only limited-angle circular cone-beam reconstruction, which is in many ways a simpler task than helical CBCT.

He *et al.* [14] proposed a two-dimensional end-to-end reconstruction pipeline using several neural networks. The method consists of learned projection filtering, backprojection, and image post-processing components. They evaluated the method using simulated 1D circular parallel-beam projections. The networks were trained in a supervised manner in the volume domain using a full-dose reference reconstruction. The method does not extend to the helical cone-beam setup.

Operating on the projections allows these methods to see everything captured by the scanner, thereby circumventing a significant limitation of post-processing methods. Still, current algorithms all require reference volumes computed using other means, and are thus bound to learn the errors in the reference.

4) Training without clean data: As shown by Lehtinen *et al.* [15], it is possible to train neural networks using only noisy training data, given that certain conditions are met. Our work builds on this so-called *Noise2Noise* principle, as detailed later. A few previous works have used the approach to circumvent the lack of ideal reference volumes [16], [17]. These methods construct an uncorrelated estimate of the 3D volume from a sparse set of input projections using a method such as wFBP, and use this as a training target for learned reconstruction. Still, the reference reconstruction technique limits the output quality. Our method does not have this issue, as we employ no reference reconstruction method.

Blind-spot approaches [18] seek to train models with no paired training data whatsoever. Jing *et al.* [19] used such a network to denoise 2D slices of a reconstruction. Their method suffers from the fact that the corruptions in the reconstructions are not independent between pixels, violating the assumptions behind blind-spot models. As such, the results are worse than those of supervised methods.

III. Our reconstruction pipeline

Our reconstruction pipeline, illustrated in Fig. 1, aims to leverage the known good properties of weighted filtered backprojection (wFBP) while providing sufficient flexibility in both 2D and 3D domains to automatically adapt to non-idealities resulting from, e.g., helical cone-beam geometry, noise, and finite number of input projections. The overall structure is shared with wFBP, with four major deviations:

1) The raw 2D X-ray images are preprocessed using a learned 2D neural network before ramp filtering. The 2D network is not supervised directly; its function is to do whatever it can to help the subsequent stages.
2) We replace the fixed ramp filter with a learned one, enabling it to adapt to the non-idealities caused by the finite number of projections and the noise they contain.
3) Instead of re-binning the cone-beam projections to parallel beams, we backproject directly from the cone-beam geometry, taking special care to avoid aliasing.
4) Finally, the 3D voxel grid that results from the backprojection is processed by a learned 3D network. As its receptive field is relatively large, it can correct for blur, spiral, and other artifacts caused by the earlier stages.

A key benefit of our design is that it can be easily adapted to any acquisition setup with variable number of projections, spacing of the helical trajectory, and radiation doses, with the learned components complementing the fixed backprojection in a data-driven manner. We will first walk through the pipeline in detail, and then describe our self-supervised training setup in Section IV.

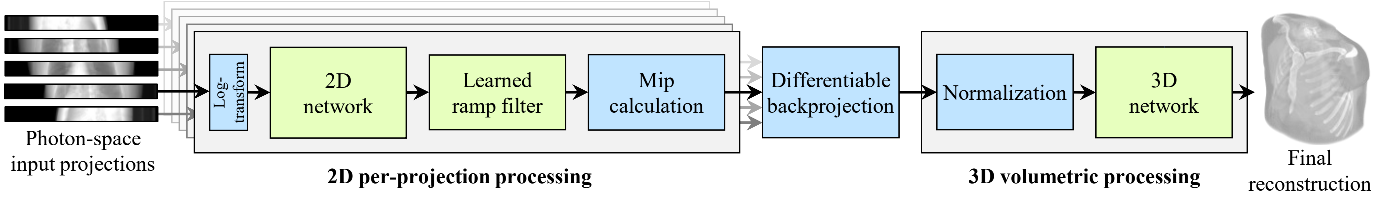


Fig. 1. Our reconstruction pipeline reconstructs a 3D voxel volume based on a set of raw cone-beam projections. First, we feed each of the input projections through a 2D neural network to produce denoised projections. Next, we filter the projections along the cone-beam rows using a ramp filter and backproject them using a differentiable 3D backprojection operator that uses mipmapping to prefilter the projections. We then normalize the reconstructed volume to obtain a per-voxel average and pass it through a 3D neural network to produce the final reconstructed volume.

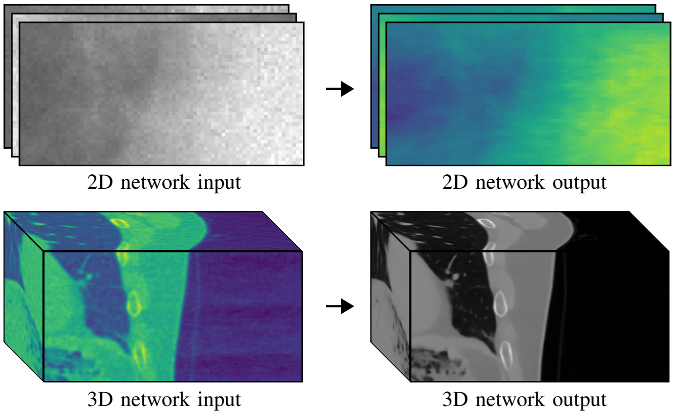


Fig. 2. Example crops of inputs and outputs of the learned neural networks (low-dose data from the LDCT [20] dataset). Top: The 2D network turns the log-space projections into feature maps for further processing. Bottom: The 3D network outputs the final reconstructed volume.

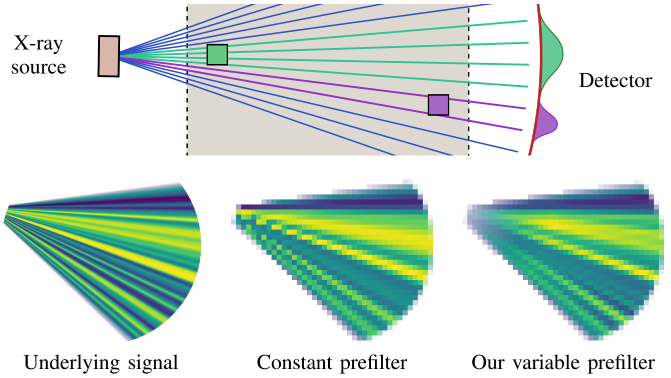


Fig. 3. A variable-size prefilter is necessary for high-quality cone-beam backprojection. Top: The green voxel near source intersects a thicker bundle of rays than the purple voxel near detector, and thus requires a wider prefilter in the projection domain to avoid aliasing. Bottom: Prefiltered sampling into the voxel grid. Sampling the underlying signal (left) with a constant-size prefilter in the projection domain (middle) yields aliasing near source, blurring near detector, or both as in this example. A variable-size prefilter (right) extracts all the frequencies that the sampling grid can represent. (2D illustration, not to scale)

A. Pipeline walkthrough

1) 2D network: Given a set of projections as an input, we first feed each of them through a learned 2D neural network that outputs a single-channel feature map in the same spatial resolution as the input. While the network has no other task than to prepare the projections for subsequent processing, we observe that it learns to perform 2D denoising to the inputs. This is illustrated in Fig. 2, top row. We visualize the resulting feature map in false color, as it is not guaranteed to be in interpretable units. While we could output a higher number of feature maps from the model, our experiments indicate no benefit from doing so.

2) Learned ramp filter: Next, to prepare for the backprojection operator that transfers the projection information from 2D to 3D, we filter the projections along the cone-beam rows using a learned ramp filter. This is implemented as a convolution with a one-dimensional kernel that is twice as wide as the input projections. The filter taps are initialized according to the inverse Fourier transform of the desired ramp frequency response, but they are treated as learnable parameters during training. This allows the pipeline to adjust the frequency spectrum of the projections, in case it is beneficial for the later operations — in practice, we have observed that the ramp filter changes very little during training.

3) Differentiable backprojection: To transfer the 2D feature maps into the 3D volume, we pass each of them to a fixed-function differentiable backprojection operator that accumulates log-space attenuation to all voxels intersected by the cone-beam in question. In contrast to wFBP, we perform backprojection directly using cone-beam geometry — i.e., along lines that connect sensor pixels with the radiation source — without re-binning to parallel beam projections first.

The backprojected value for a voxel is obtained by first projecting its center onto the 2D sensor using the radiation source as the center of projection, and then finding the value by interpolation on the sensor's pixel grid. The projection lines converge onto the radiation source, which causes the local frequency content of the backprojected signal to vary significantly: Close to the source, the projection lines are packed densely, while near the sensor their spacing is sparser (Fig. 3, top). As using a voxel grid fine enough to capture the densest beam bundles is impractical, we carefully anti-alias the result to ensure the backprojected signal can be represented by the voxel grid faithfully, i.e., without aliasing. As per standard Shannon–Nyquist sampling and reconstruction theory [21], this requires prefiltering (blurring) the 3D signal before sampling to remove spatial frequencies too high to be representable by the voxel grid.

In practice, prefiltering and sampling are combined into a single per-voxel operation. We first geometrically determine the ratio between the 3D voxel pitch and the 2D sensor pixel pitch by considering similar triangles. The ratio, which varies between voxels, determines the amount of bandlimiting that

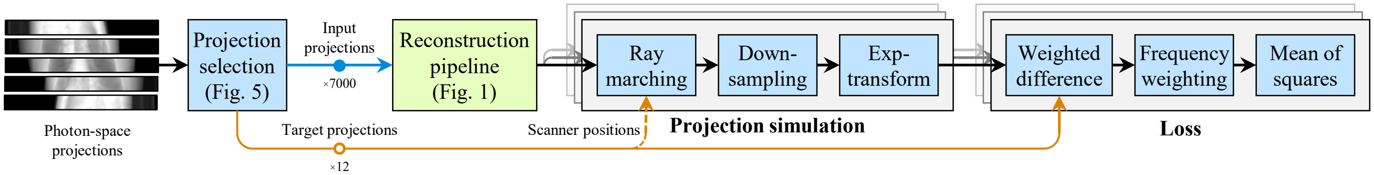


Fig. 4. To train the learned components of our reconstruction pipeline (Fig. 1), we use a self-supervised setup that does not require reference data. The training pipeline first selects 12 target projections from a randomly chosen rotation of the scanner. We feed the rest of the projections into our reconstruction pipeline to produce a reconstruction of the target region. From this reconstruction, we simulate projections from the same locations as the target projections, which are then downsampled and exp-transformed into photon intensity space. A weighted difference between the simulated and target projections is used for calculating an L_2 loss. To speed up convergence, we ramp filter the difference before the L_2 norm.

must be applied to the 2D feature map in order to avoid aliasing. As illustrated in Fig. 3 (bottom), more aggressive filtering is required for voxels closer to the radiation source.

While supporting a per-voxel arbitrarily-sized 2D prefilter is infeasible, an excellent approximation can be achieved through a variation of *mipmapping* [22], a technique common in computer graphics. In practice, when preparing to backproject a 2D feature map, we compute several progressively blurrier versions of it using Lanczos [23] prefilters of increasing support (decreasing bandwidth). After the desired prefilter bandwidth has been determined for a voxel, we reconstruct the backprojected sample value via trilinear interpolation from the two prefiltered projections whose filter sizes best match the desired bandwidth. This is a very close approximation to having a prefilter with arbitrary size and position on the denoised and ramp-filtered 2D projection. In our current configuration, we use five prefilter bandwidths chosen to cover the range of filter bandwidths required by the backprojection. In contrast to standard mipmapping, we keep the resolution of the prefiltered feature maps unchanged.

To normalize the result of accumulating the backprojections from the entire set of 2D projections into the volume, we track the number of backprojections contributing to each voxel and divide by this number to obtain a per-voxel average. While this normalization scheme is unable to account for the unevenness in the angular distributions of rays resulting from the cone-beam geometry, we have found that the resulting spiral artifacts are easily corrected by subsequent processing.

4) Learned 3D processing: As the final step of the reconstruction pipeline, we pass the volume through a learned 3D neural network. The purpose of this network is to perform final denoising and sharpening and to remove any remaining artifacts. An example is shown in Fig. 2, bottom row.

B. Implementation details

We use U-Nets [24], i.e., autoencoders with skip connections, for the 2D and 3D networks as they have been shown to perform well on a variety of tasks including denoising and removal of image artifacts [25]. The 2D network architecture follows Lehtinen *et al.* [15] exactly, whereas the 3D network is otherwise similar except that the intermediate channel counts have been halved to conserve memory, and 3×3 convolution kernels have been replaced with 3×3×3 kernels to enable volume processing. Network weights were initialized using He initialization [26]. Our differentiable backprojection operation was implemented as a custom PyTorch [27] operator using a combination of custom CUDA code and a modified version of the texture lookup function in Nvdiffrast [28].

IV. Training

We now turn to training the learned 2D neural network, ramp filter weights, and 3D neural network. We train the pipeline in an end-to-end fashion, meaning that only the fidelity of the final reconstruction provides the signal that guides the components to a joint optimum. We describe the overall architecture of the self-supervised loss function and training loop in Sec. IV-A and deal with photon noise in the training data in Sec. IV-B. The process is illustrated in Fig. 4.

A. Self-supervised training

To enable training without known reference 3D volumes, we combine a projection consistency loss, similar to many iterative reconstruction techniques, with a leave-out strategy that resembles cross validation: A volume reconstruction is considered faithful if left-out real X-rays look the same as simulated X-rays computed using the same scanner position. A key benefit of this approach is that it requires no reference data, either in 2D or 3D domain.

Each training iteration begins by selecting a random slab of the volume from a scan in the dataset, and identifying the set of X-rays whose backprojections overlap with the slab. The set is then randomly split into a large set of *input projections* and a small set of *target projections* (Fig. 5). The input projections are fed to our reconstruction pipeline, resulting in a 3D volume. We then compute, for each target projection, a virtual X-ray using the known positions of the radiation source and sensor using a differentiable X-ray simulator (Sec. IV-C). The final loss function is the mean squared error between the simulated projections and left-out target projections. As all components in the pipeline are differentiable, the gradient of the learnable parameters can be computed using standard backpropagation. We use Adam [29] as the optimization algorithm and run the training in parallel using 8 NVIDIA A100 GPUs. The networks were trained for 2.5 days (480 GPU hours) for the synthetic dataset, and 8 days (1536 GPU hours) for the real dataset to accommodate for the higher resolution.

B. Noisy target projections

A subtle point not addressed in the discussion above is that as we train with real X-ray data, we do not have noise-free projections at hand, i.e., the target projections contain

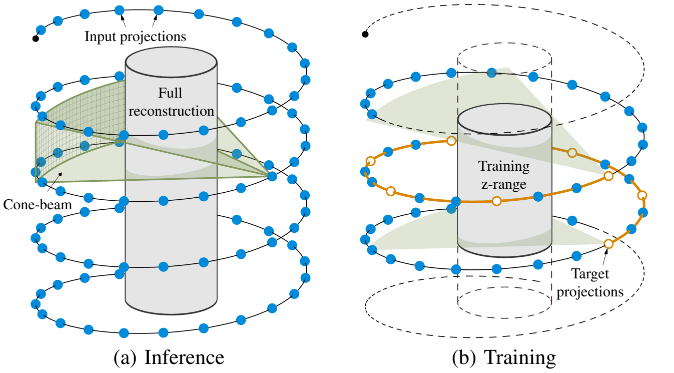


Fig. 5. Geometric setup of the reconstruction problem. (a) The scanner travels along a helical trajectory, capturing cone-beam projections at regular intervals with the positions of the X-ray source depicted by the blue dots. During inference, the full volume is reconstructed using all input projections. (b) In each training iteration, we randomly choose a small z range (gray) that corresponds to a single a rotation of the scanner, and use the projections that intersect it as either inputs (blue dots) or targets (orange dots). (Illustration not to scale)

all forms of noise inherent to X-ray imaging. Fortunately, it has been previously shown that under certain circumstances, it is possible to train neural networks using only noisy training targets. In particular, this *Noise2Noise* principle [15] states that if the corruptions in the training targets are zero-mean and uncorrelated with the corruptions in the inputs, an L_2 training loss will, given sufficient data, converge to the same optimum as a model trained with noise-free targets. This is exactly the situation we are faced with: The photon noise is zero-mean and uncorrelated between the input and target projections, and therefore, computing the loss between simulated projections and the noisy real targets is justified.

The requirement that the noise in the model inputs is uncorrelated with training targets is also the reason behind the leave-out strategy of not using target projections as model inputs. If this is not met, the quality of the results deteriorates dramatically, as demonstrated in Section V-E. While the leave-out strategy leaves gaps in the set of input projections seen by the reconstruction pipeline during training, we have found the impact of these gaps to be negligible as long as the number of target projections is kept small. For each training iteration, we use approx. 7000 input projections and 12 target projections.

A final detail to consider is that the noise in the acquired 2D projections is zero-mean in photon intensity, but not in log-attenuation because of the nonlinear transformation. As such, to use noisy training targets, we must compute the L_2 loss in photon intensity space. This, however, has the severe problem that pixels with high photon counts, i.e., low attenuation, have exponentially larger weight in the overall loss function than highly attenuated pixels, which is at odds with the practice of viewing the results in log-attenuation space. Therefore, we scale the photon-intensity L_2 loss so that each pixel's contribution to the overall loss is proportional to what it would have been if the loss were computed in log-attenuation space. The resulting loss function is

$$\mathcal{L}(\hat{X},\hat{Y}) = \left\| w(\hat{X}) \odot (\hat{X} - \hat{Y}) \right\|_2^2 , \;\; w(\hat{X}) = 1/\hat{X} \qquad (1)$$

where $\hat{X}$ and $\hat{Y}$ are the simulated projection and the training data projection in photon intensity space, respectively, and $\odot$ denotes element-wise multiplication. The weighting function $w(\hat{X})$ is proportional to the inverse derivative of the exp-transform. Importantly, we zero the gradients of $w(\hat{X})$ to prevent training from attempting to just minimize this weight [15].

We have so far assumed that the same projections are used as both inputs to the reconstruction pipeline as well as training targets. However, we may have paired projections with different dosages available at training time. Such paired data can be obtained by, e.g., acquiring higher-dose projections and adding noise to them that simulates a lower-dose scan. In this situation, we can use the higher-dose projections as $\hat{Y}$ in Eq. 1, while still using the lower-dose projections as input. The training pipeline (Fig. 4) remains almost unchanged: Each scanner position is now associated with two projections instead of one, and their use depends on whether the scanner position was designated as input (blue) or target (orange).

Both inputs and targets are free to exhibit noise that may be correlated between them, which is the case when lower-dose projections are simulated based on a higher-dose scan.

C. Projection simulation

The simulated projections are computed by ray marching through the reconstructed voxel grid using trilinear interpolation. To prevent aliasing, we cast 16 rays per output pixel arranged in a uniform grid pattern, and downsample the resulting image by 4×4 using a Lanczos filter. As is common in iterative reconstruction techniques [30], we low-pass filter the target projections to adjust the amount of ringing in the results. We used filter kernels $[0.2, 1.0, 0.2]$ and $[0.05, 1.0, 0.05]$ for synthetic and real-world data, respectively.

We have found experimentally that training convergence can be accelerated considerably by emphasizing high spatial frequencies in the pixelwise error images between simulated projections and target projections before computing the mean squared loss. In practice, we apply a classic (non-learned) ramp filter to the difference images, i.e., the weighted difference inside the norm in Eq. 1. Reminiscent of the derivation of the ramp filter in wFBP, this focuses the loss evenly on all frequencies in the volume, whereas low frequencies tend to dominate otherwise. We use this optimization in all of our training runs. Similar to many previous works (e.g., [31]), we also found it practically beneficial to use our custom mipmapping-based backprojection operator for computing the gradient of the ray-marching operation.

D. Augmentations

Three kinds of data augmentation are applied during training to increase the robustness of the neural networks. First, we choose a random rotation around the z axis for the voxel grid, ensuring that the networks learn no preferential orientation of features in the xy plane. Second, we add a random sub-voxel offset for the center of the reconstruction volume to break the alignment of the voxel grid in relation to the scanner geometry. These augmentations are implemented by perturbing the geometry information in training data, and do not involve, e.g., resampling of projection or volume data. Finally, we scale

TABLE I
DATASET SPECIFICATIONS

	Synthetic	Real
Source	XCAT [32]	LDCT [20]
Training scans	13	41
Evaluation scans	4	6
Projections per scan	9,000–12,000	11,000–15,000
Projection resolution	736×64	736×64
Spiral pitch	0.9	0.9
Scan range	210–300 mm	240–380 mm
xy voxel grid size	576×576	1024×1024
z voxel grid size (inference)	272–374	402–644
z voxel grid size (training)	128	160
Reconstruction voxel spacing	0.784 mm	0.586 mm
Reconstruction cylinder diameter	452 mm	600 mm

TABLE II
LOW-DOSE RECONSTRUCTION QUALITY USING SYNTHETIC DATA

Method	PSNR (dB)	RMSE	Runtime
Traditional methods			
wFBP [1]	23.23	0.1379	90s
IR-TV [5]	35.49	0.0336	∼1h
Supervised training			
wFBP + RED-CNN [3]	38.18	0.0247	106s
wFBP + 2D U-Net	39.17	0.0220	92s
wFBP + 3D U-Net	39.77	0.0206	93s
Self-supervised training (Our method)			
2D U-Net + Our BP + 3D U-Net	**41.11**	**0.0176**	27s

the log-attenuation values in all input and target projections by a random scalar in range $[0.75, 1.25]$ for each reconstruction during training. This is a valid transformation because the backprojection and projection operations are linear in log-domain, and it prevents the networks from learning typical attenuation coefficients and exploiting that information when reconstructing previously unseen data.

V. RESULTS

We evaluate our method on both synthetic and real-world data. Real-world data enables us to qualitatively confirm that our method scales up to the complexity of real subjects and scanner setups, while a synthetic dataset, where a noise-free ground truth is available, allows calculation of quantitative metrics. We train our reconstruction pipeline separately for each type of data. Dataset specifications are listed in Table I. Full versions of result images, including neighboring slices and an interactive viewer, are available as supplemental material.

A. Datasets and comparison methods

1) Synthetic data: We generate a synthetic helical cone-beam dataset using the XCAT CT projection simulator [32]. We simulate full-dose and low-dose (10% of full dose) scans of each phantom. We chose the scanner parameters to match those of the real-world dataset (described below) with the exception that we do not use a flying focal spot. Because the projections in the real-world dataset have beam hardening correction applied, we simulate monochromatic radiation with an energy level of 80 keV to avoid beam hardening effects in the synthetic data as well. Following the real-world setup further, we simulate tube current modulation that attempts to keep the photon Poisson noise roughly consistent throughout the scan. To produce high-quality ground truth, we export the voxel output from XCAT in 16× the target resolution and downscale it using a $16 \times 16 \times 16$-voxel box filter, yielding a total of 4096 samples per output voxel.

2) Real data: For the real data experiments we use the Low Dose CT Image and Projection Dataset (LDCT) [20], from which we use 47 chest scans captured on Siemens scanners. Each scan contains full-dose projections with various corrections (e.g., for beam hardening, scattering, nonuniformity) applied to them by the scanner manufacturer. In addition, each scan has a corresponding set of simulated low-dose (10% of full dose) projections created by adding noise on top of the full-dose projections. Unless otherwise noted, we use the simulated low-dose projections as inputs and full-dose projections as training targets in our method (Sec. IV-B). Finally, each scan has a reference 3D reconstruction computed by the scanner manufacturer. These references are noisy, so we cannot perform numerical comparisons against them.

3) Comparison methods: We compare our reconstructions with three previous methods: wFBP [1], total variation regularized iterative reconstruction (IR-TV) [5], and RED-CNN [3]. We use FreeCT [33] as the wFBP implementation. As further points of comparison, we evaluate the components of our reconstruction pipeline in isolation and in various combinations.

Our straightforward IR-TV implementation employs the same non-aliasing high-quality projection and backprojection operators as our reconstruction pipeline, and performs the optimization using the Adam optimizer [29]. In addition, we weight the projected rays according to approximate noise level using a Poisson noise model [34]. As is customary, we low-pass filter the input projections to minimize ringing using a $[c, 1.0, c]$ kernel where c is a free parameter. The parameter c and the strength of the TV regularizer where chosen to maximize PSNR on synthetic data. We re-implemented RED-CNN following the design of Chen *et al.* [3] exactly.

4) Metrics: For numerical evaluation, we compute peak signal-to-noise ratio (PSNR) in decibels between the reconstruction result and ground truth reference volume (only available in the XCAT dataset). The PSNR computation is done over the 3D volume instead of, e.g., averaging over individual 2D slices. We also compute the root-mean-square error (RMSE) of visual densities assuming a display window width of 2000 Hounsfield Units (HU) that covers the variation in XCAT data. The PSNR is computed from these scaled values as well, corresponding to an implicit choice of 2000 HU as the peak difference between raw densities. Both metrics are calculated without clipping or quantizing the data.

B. Quantitative results on low-dose synthetic inputs

Our main focus is on reconstruction from low-dose input projections, and we begin by comparing various methods in this regime using synthetic data. To enable fair comparison, we use full-dose data as the training target for all learning-based methods. Table II presents the PSNR and RMSE for each method, computed against noise-free XCAT ground truth

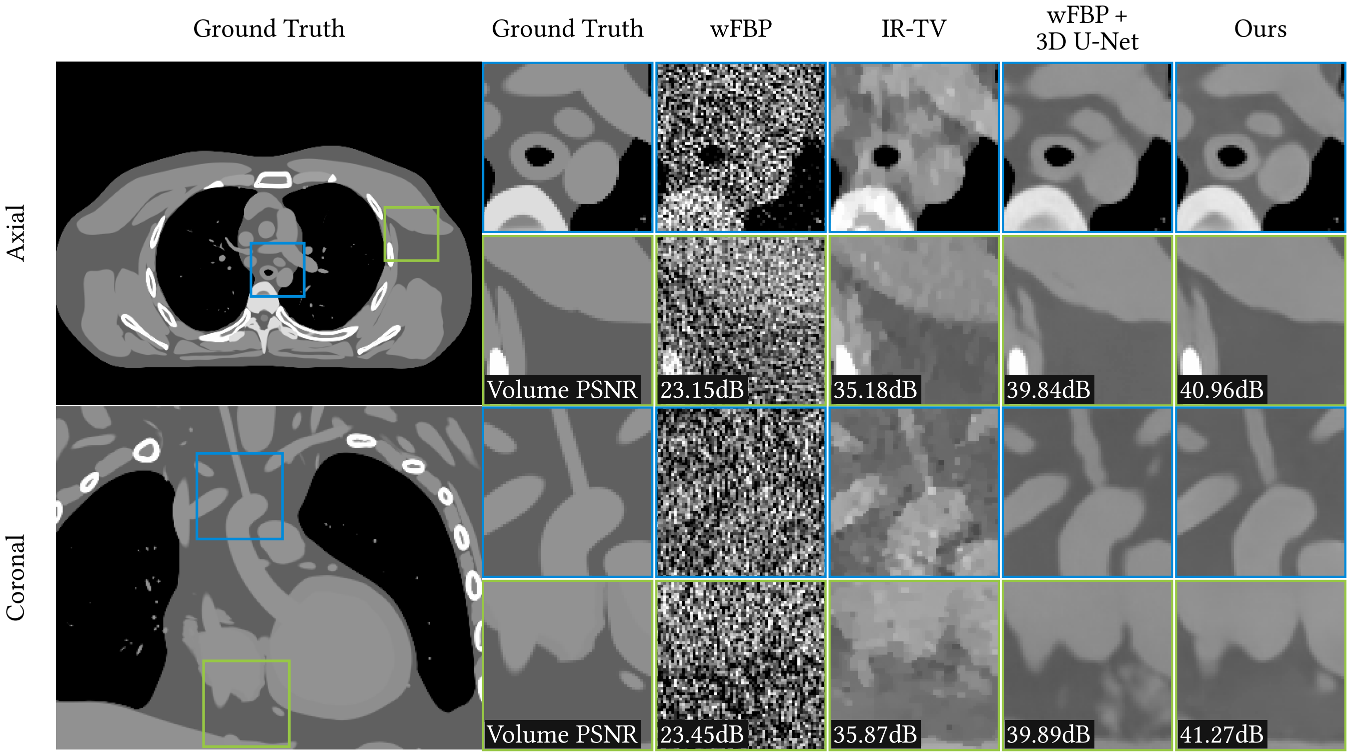


Fig. 6. Low-dose reconstructions of synthetic data with different methods. Display window is set to [−400, 400] HU. PSNR values refer to the individual volumes shown, not the entire dataset. Full images and neighboring slices are available in the supplemental material.

volumes. We start from traditional methods and build up towards our method by gradually adding learned components.

1) Traditional methods: The traditional baseline methods, wFBP and IR-TV, do not take advantage of any training data — with the exception that the parameters of IR-TV were tuned to maximize the output quality over the dataset. As such, their reconstruction quality is limited in the low-dose regime.

2) Supervised deep learning: We first apply RED-CNN to the output of wFBP, improving the result by several decibels. The RED-CNN denoiser was trained in a supervised fashion, with full-dose reconstructions made using wFBP as training targets, representing a realistic training scenario. As a control, we also similarly trained a denoiser network using the architecture of our 2D U-Net, achieving 0.99 dB better PSNR than RED-CNN. While the architecture change yields a significant improvement, the results are not as good as 3D models.

In the preceding denoising tests, processing is done one axial slice at a time, meaning that the denoiser cannot exploit 3D structure in the volume. To gauge the usefulness of volumetric denoising, we trained a standalone denoiser network based on our 3D U-Net architecture. Again using full-dose wFBP reconstructions as training targets and wFBP low-dose reconstructions as inputs, our 3D U-Net denoiser improves the output quality by further 0.60 dB, confirming that access to 3D structure improves denoising results numerically.

3) Self-supervised training (Our method): Up to this point, the reference training targets for all methods has been the full-dose reconstructions made using wFBP, which is a realistic scenario assuming that absolutely noise-free training targets are not available. However, the drawback is that the attainable output quality is limited by any reconstruction artifacts left in the training targets. Our full method that uses self-supervised training sidesteps this problem by computing the loss in projection domain. As seen on the last row of Table II, this improves the output quality significantly to 41.11 dB, surpassing the best comparison method by 1.34 dB.

C. *Qualitative results on low-dose inputs*

Fig. 6 shows a set of example low-dose reconstructions for the XCAT dataset, and Fig. 7 shows the corresponding results for the LDCT dataset. The figures confirm that the PSNR improvements seen in Table II correspond to visually better results for both synthetic and real data. As noise-free ground truth data is not available for LDCT, we show full-dose wFBP reconstructions in their place. To facilitate visual comparison, we slightly blur the 3D reconstructions produced by our method to better match the visual look of the ground-truth and wFBP results using a hand-tuned Lanczos filter.

Our main competitor, “wFBP + 3D U-Net” (2$^{\text{nd}}$ column from the right), suffers from the correlations between the low and full-dose projections in LDCT dataset: because the low-dose projections have been simulated by adding noise into full-dose projections, there is a common noise component between inputs and outputs that the 3D network learns to partially preserve. Our method removes the target projections from the set of input projections in each training step to avoid this issue.

Fig. 8 highlights the importance of performing denoising over the full 3D volume. 2D denoisers, such as RED-CNN, construct the 3D volume slice-by-slice, which results in strong artifacts along the non-axial cross-sections. Our 3D network, on the other hand, produces a consistent 3D volume.

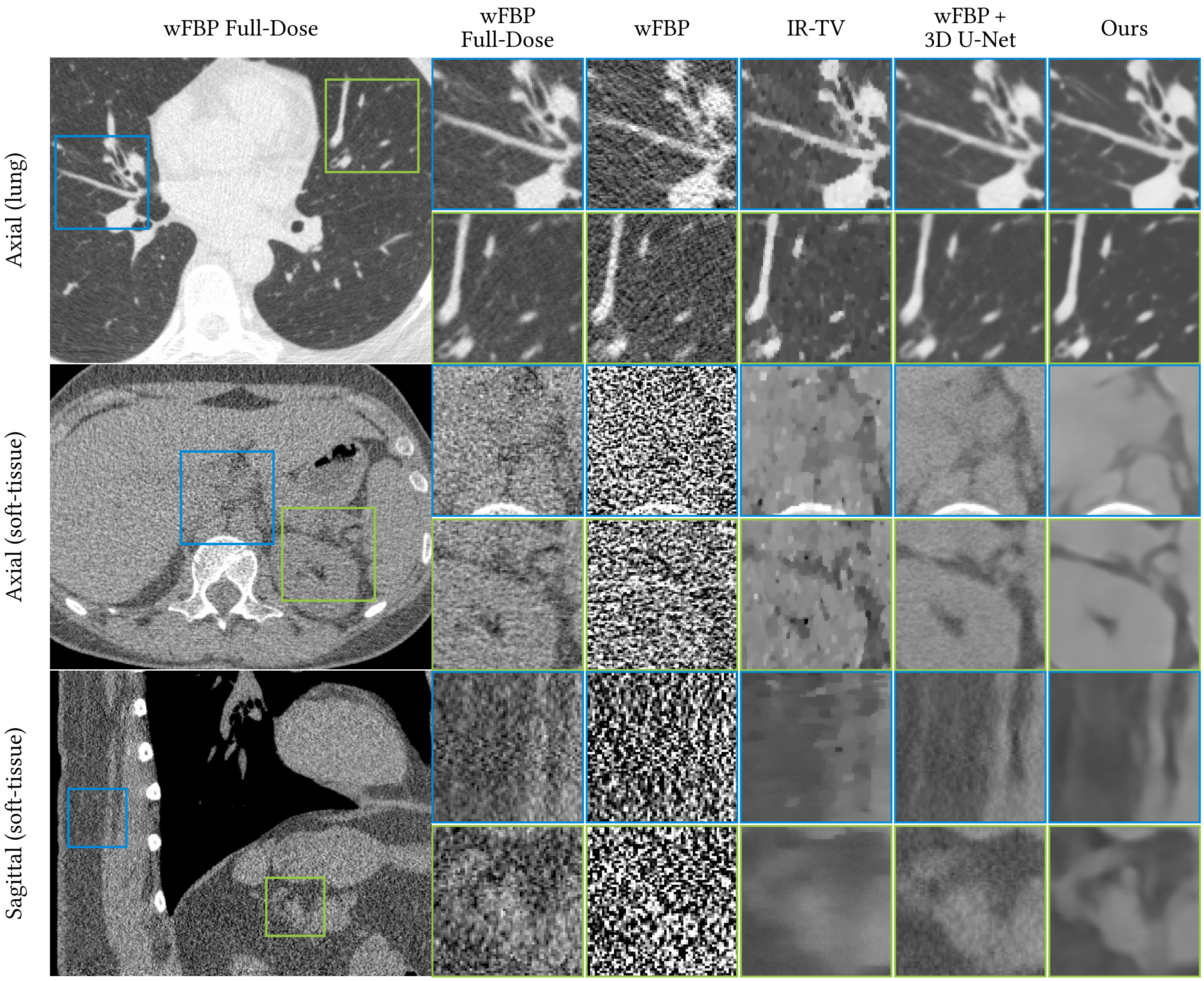


Fig. 7. Low-dose reconstructions of real-world data with different methods. For reference, full-dose wFBP (two leftmost columns) is shown in lieu of noise-free ground truth that is not available. Soft tissue window is set to [−300, 300] HU and lung window to [−1350, 150] HU. Full images and neighboring slices are available in the supplemental material.

D. Results using full-dose inputs

A unique property of our self-supervised loss is that our method can be trained to operate on full-dose scans as well. In contrast, supervised methods rely on the availability of a separate reference result; it is not meaningful to construct this reference from the same projections that are also used as input.

Table III shows that our method achieves the best numerical results compared to wFBP and iterative reconstruction; these traditional methods are the only ones applicable in this comparison. As illustrated in Fig. 9, our reconstructions have less noise than the comparison methods, and no detail is lost.

E. Additional experiments

1) Noise-free targets: To validate the correctness of our self-supervised training setup, we performed experiments with noise-free synthetic data. If both input and reference data are noise-free, our method converges to a virtually perfect result when using either supervised or self-supervised loss. This confirms that our networks are able to correct for any artifacts resulting from the volumetric backprojection in a data-driven way, and that the self-supervised loss achieves results on par with the supervised loss when having access

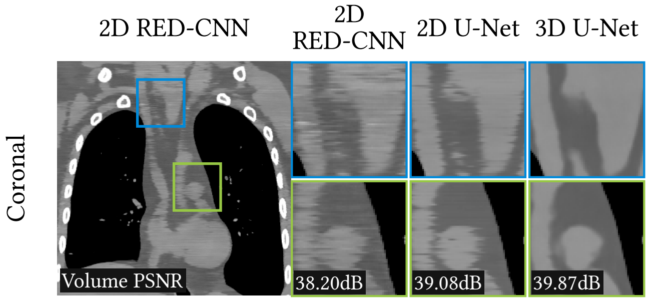


Fig. 8. Coronal slices of low-dose reconstructions of synthetic data using supervised 2D and 3D denoisers. As the 2D methods process each axial slice independently, they suffer from inconsistencies in other planes. A 3D denoiser is consistent along all axes.

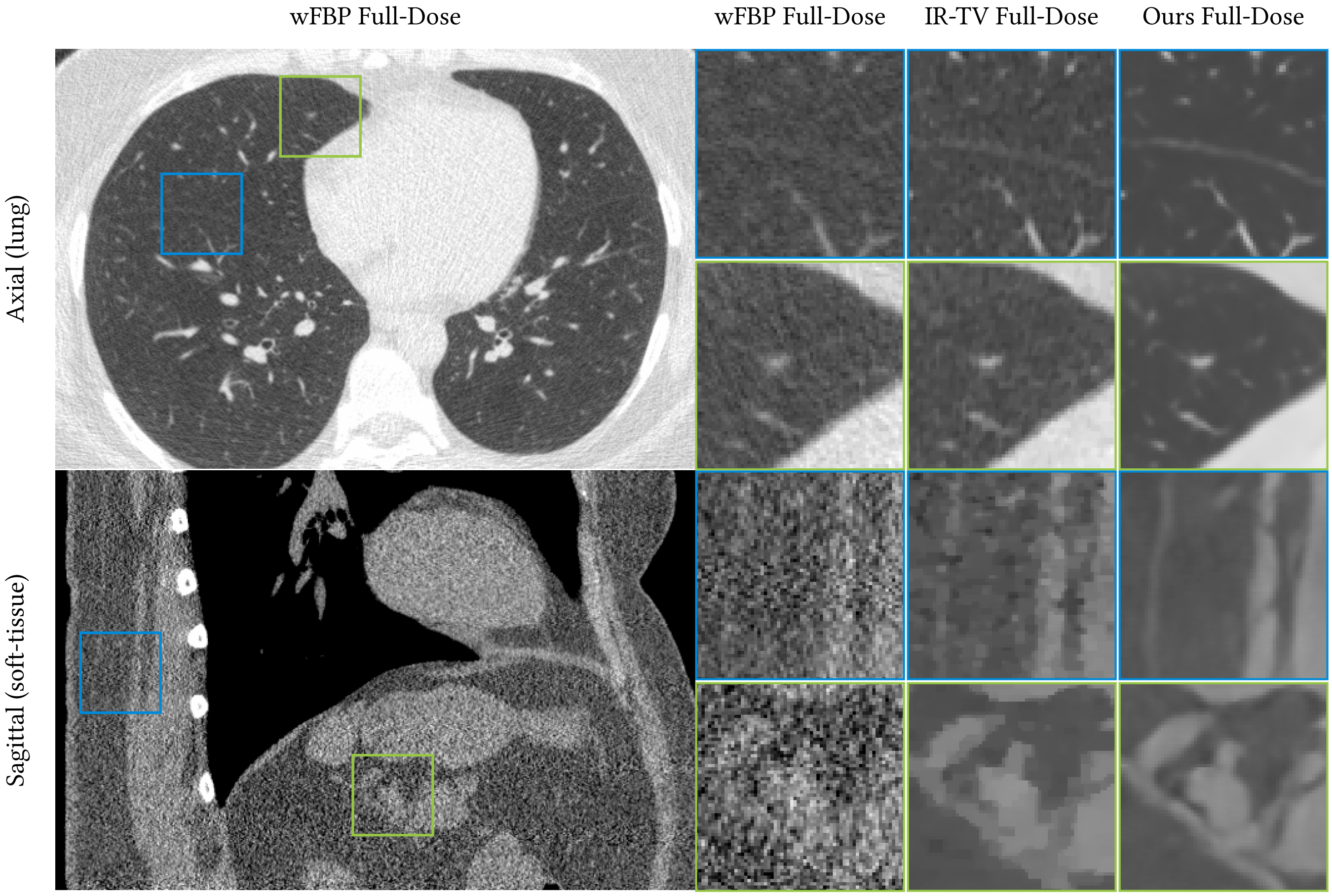


Fig. 9. Full-dose reconstructions of real-world data. Soft tissue window is set to [−300, 300] HU, and lung window to [−1350, 150] HU. Our self-supervised loss enables us to train our pipeline with full-dose input data, which is not possible with traditional supervised methods. Compared to FBP and total variation regularized iterative reconstruction, our results contain less noise and do not suffer from IR-TV's blockiness. Full images and neighboring slices are available in the supplemental material.

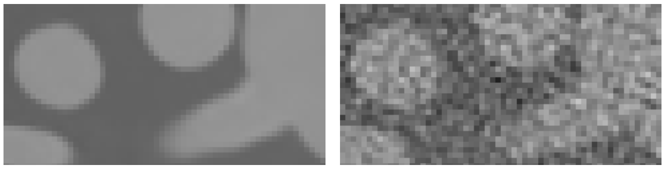
TABLE III
FULL-DOSE RECONSTRUCTION QUALITY USING SYNTHETIC DATA

Method	PSNR (dB)	RMSE	Runtime
Traditional methods			
wFBP [1]	33.41	0.0427	90s
IR-TV [5]	42.12	0.0157	∼1h
Self-supervised training (Our method)			
2D U-Net + Our BP + 3D U-Net	**44.21**	**0.0123**	27s

to synthetic ground truth reconstructions.

2) Correlated noise: To highlight the importance of not having correlated corruptions in the inputs and targets, we experimentally trained our method without removing the target projections from the input. In this case, the input noise is correlated with the target projections, and the networks learn to pass the noise on to the reconstructions (Fig. 10).

3) Photon-space loss: To gauge the effect of our photon-space loss (Eq. 1), we trained variants of our method using L_2 loss in log-space instead. The log-transformation is nonlinear and therefore skews the mean of the targets, violating the zero-mean noise requirement of Noise2Noise training [15]. When training with full-dose targets we did not observe significant differences in the numerical results, suggesting that the skew is small compared to photon counts. However, when training with low-dose targets, the photon counts are lower, and the log-transformation skew is relatively larger. With low-dose targets we observed a drop of 3.32 dB compared to our photon-space loss function. Visual inspection confirmed that the results overestimated the attenuation, as the loss function was skewed towards higher log-space attenuation values. This confirms the importance of calculating the loss in photon intensity space.



Target projections excluded from the input set

Target projections incorrectly included in the input set

Fig. 10. Example crops of our method trained with target projections excluded (left) and included (right). Both were trained with synthetic full-dose data as input and target. If the targets are not excluded, the network input becomes correlated with the targets and the networks learn to pass the noise through instead of removing it.

4) Network capacity: To analyze the effect of network capacity, we trained various networks on the synthetic dataset with varying channel counts. Halving the channel count of the 3D network hurt the numerical results significantly, but halving the channel count of the 2D network had a fairly small impact. Hence, it appears that increasing the channel count of the 3D network could improve the results further; at present this is prohibited by the memory space available in GPUs.

5) Custom backprojection: Given that we do not explicitly correct for potential artifacts arising from the cone-beam geometry in our differentiable backprojection operator, one may ask whether our results could be improved by replacing the beginning of our pipeline with FreeCT's wFBP implementation. In a separate experiment with low-dose inputs, we found that this is not the case: this combination reached an output quality of 39.59 dB, i.e., 1.52 dB lower than our proposed pipeline. We suspect that the cause of this failure is due to aliasing in wFBP outputs that do not cause issues in scenarios where the training targets are also computed using wFBP. However, in the self-supervised setting, this aliasing apparently makes it difficult for the 3D network to recover a high-quality volume that would faithfully match the projections.

To validate this hypothesis, we performed another experiment where we trained our pipeline in a supervised fashion, similar to the "wFBP + 3D U-Net" case in Table II. This combination reached an output quality of 39.38 dB, i.e., 0.39 dB lower than running our 3D U-Net directly on the output of wFBP. This confirms that wFBP is indeed beneficial in the supervised setting, but harmful in the self-supervised setting.

VI. Discussion and future work

We have shown that self-supervised training can be highly beneficial in helical CBCT reconstruction, and believe that the idea of combining projection simulation with end-to-end machine learning could be applied in a range of other tomography setups and other inverse imaging problems.

There are also several specific improvements that could be made in the CBCT case. Most importantly, our training-time model of the imaging setup, i.e., generation of simulated projections from reconstructed volume, is fairly simplistic. For example, we do not utilize tube current information. We also do not currently attempt to reproduce effects such as beam hardening, scattering, and metal artifacts. We believe that using uncorrected raw projection data and simulating these effects in the projection simulation step could lead to further significant improvements, as these artifact-inducing effects are presumably easier to simulate than to remove directly.

Acknowledgments

We thank Timo Aila for discussions and comments on the manuscript; Samuli Siltanen, Alexander Meaney, Antti Korvenoja, Mika Kortesniemi, Juha Järveläinen and Jussi Hirvonen for discussions; David Luebke and Kimmo Kaski for general support; and Tero Kuosmanen, Janne Hellsten and Samuel Klenberg for compute infrastructure support. The project made use of computational resources provided by the Aalto Science-IT project and the Finnish IT Center for Science (CSC). Onni Kosomaa's NVIDIA Research internship notwithstanding, he was funded by the Finnish Center for Artificial Intelligence (FCAI), a part of the Academy of Finland Flagship programme.

References

[1] K. Stierstorfer, A. Rauscher, J. Boese, H. Bruder, S. Schaller, and T. Flohr, "Weighted FBP—A simple approximate 3D FBP algorithm for multislice spiral CT with good dose usage for arbitrary pitch," *Phys. Med. Biol.*, vol. 49, no. 11, pp. 2209–2218, May 2004.

[2] F. Noo, M. Defrise, and R. Clackdoyle, "Single-slice rebinning method for helical cone-beam CT," *Phys. Med. Biol.*, vol. 44, no. 2, pp. 561–570, Feb 1999.

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

[4] A. A. Zamyatin, L. Yu, and D. Rozas, "3D residual convolutional neural network for low dose CT denoising," in *Proc. Med. Imag.*, vol. 12031. SPIE, 2022, pp. 634–645.

[5] E. Y. Sidky and X. Pan, "Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization," *Phys. Med. Biol.*, vol. 53, no. 17, pp. 4777–807, 2008.

[6] K. Kim, G. El Fakhri, and Q. Li, "Low-dose CT reconstruction using spatially encoded nonlocal penalty," *Med. Phys.*, vol. 44, no. 10, pp. e376–e390, 2017.

[7] G. Wang, J. C. Ye, K. Mueller, and J. A. Fessler, "Image reconstruction is a new frontier of machine learning," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1289–1296, 2018.

[8] Q. Yang *et al.*, "Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1348–1357, 2018.

[9] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, "Generative adversarial networks for noise reduction in low-dose CT," *IEEE Trans. Med. Imag.*, vol. 36, no. 12, pp. 2536–2545, 2017.

[10] J. P. Cohen, M. Luck, and S. Honari, "Distribution matching losses can hallucinate features in medical image translation," in *Proc. MICCAI*, 2018, pp. 529–536.

[11] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, "CNN-based projected gradient descent for consistent CT image reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1440–1453, 2018.

[12] J. Adler and O. Öktem, "Learned primal-dual reconstruction," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1322–1332, 2018.

[13] T. Würfl *et al.*, "Deep learning computed tomography: Learning projection-domain weights from image domain in limited angle problems," *IEEE Trans. Med. Imag.*, vol. 37, no. 6, pp. 1454–1463, 2018.

[14] J. He, Y. Wang, and J. Ma, "Radon inversion via deep learning," *IEEE Trans. Med. Imag.*, vol. 39, no. 6, pp. 2076–2087, 2020.

[15] J. Lehtinen *et al.*, "Noise2Noise: Learning image restoration without clean data," in *Proc. ICML*, 2018.

[16] A. A. Hendriksen, D. M. Pelt, and K. J. Batenburg, "Noise2Inverse: Self-supervised deep convolutional denoising for tomography," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1320–1335, 2020.

[17] D. Wu, K. Kim, and Q. Li, "Low-dose CT reconstruction with Noise2Noise network and testing-time fine-tuning," *Med. Phys.*, vol. 48, no. 12, pp. 7657–7672, 2021.

[18] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void – Learning denoising from single noisy images," in *Proc. CVPR*, 2019, pp. 2129–2137.

[19] J. Jing *et al.*, "Training low dose CT denoising network without high quality reference data," *Phys. Med. Biol.*, vol. 67, no. 8, p. 084002, Apr 2022.

[20] T. R. Moen *et al.*, "Low-dose CT image and projection dataset," *Med. Phys.*, vol. 48, no. 2, pp. 902–911, 2021.

[21] C. E. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, 1949.

[22] L. Williams, "Pyramidal parametrics," *SIGGRAPH Comput. Graph.*, vol. 17, no. 3, pp. 1–11, Jul 1983.

[23] C. Lanczos, *Applied Analysis*. Prentice Hall, 1956.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," *Proc. MICCAI*, vol. 9351, pp. 234–241, 2015.

[25] X. Mao, C. Shen, and Y. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. NIPS*, 2016, pp. 2802–2810.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. ICCV*, 2015, pp. 1026–1034.

[27] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NeurIPS*, 2019, pp. 8024–8035.

[28] S. Laine, J. Hellsten, T. Karras, Y. Seol, J. Lehtinen, and T. Aila, "Modular primitives for high-performance differentiable rendering," *ACM Trans. Graph.*, vol. 39, no. 6, 2020.

[29] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, 2015.

[30] H. Kunze, K. Stierstorfer, and W. Härer, “Pre-processing of projections for iterative reconstruction,” in *Proc. Fully3D*, 2005.

[31] G. L. Zeng and G. T. Gullberg, “Unmatched projector/backprojector pairs in an iterative reconstruction algorithm,” *IEEE Trans. Med. Imag.*, vol. 19, no. 5, pp. 548–555, 2000.

[32] W. P. Segars, M. Mahesh, T. J. Beck, E. C. Frey, and B. M. W. Tsui, “Realistic CT simulation using the 4D XCAT phantom,” *Med. Phys.*, vol. 35, no. 8, pp. 3800–3808, 2008.

[33] J. Hoffman, S. Young, F. Noo, and M. McNitt-Gray, “Technical note: FreeCT_wFBP: A robust, efficient, open-source implementation of weighted filtered backprojection for helical, fan-beam CT,” *Med. Phys.*, vol. 43, no. 3, pp. 1411–1420, 2016.

[34] J.-B. Thibault, K. D. Sauer, C. A. Bouman, and J. Hsieh, “A three-dimensional statistical approach to improved image quality for multislice helical CT,” *Med. Phys.*, vol. 34, no. 11, pp. 4526–4544, 2007.