WNet: A data-driven dual-domain denoising model for sparse-view computed tomography with a trainable reconstruction layer

Theodor Cheslerean-Boghiu, Felix C. Hofmann, Manuel Schultheiß, Franz Pfeiffer, Daniela Pfeiffer, and Tobias Lasser Member, IEEE.

Abstract—Deep learning based solutions are being successfully implemented for a wide variety of applications. Most notably, clinical use-cases have gained an increased interest and have been the main driver behind some of the cutting-edge data-driven algorithms proposed in the last years. For applications like sparse-view tomographic reconstructions, where the amount of measurement data is small in order to keep acquisition time short and radiation dose low, reduction of the streaking artifacts has prompted the development of data-driven denoising algorithms with the main goal of obtaining diagnostically viable images with only a subset of a full-scan data. We propose WNet, a data-driven dual-domain denoising model which contains a trainable reconstruction layer for sparse-view artifact denoising. Two encoder-decoder networks perform denoising in both sinogram- and reconstruction-domain simultaneously, while a third layer implementing the Filtered Backprojection algorithm is sandwiched between the first two and takes care of the reconstruction operation. We investigate the performance of the network on sparse-view chest CT scans, and we highlight the added benefit of having a trainable reconstruction layer over the more conventional fixed ones. We train and test our network on two clinically relevant datasets and we compare the obtained results with three different types of sparse-view CT denoising and reconstruction algorithms.

Index Terms—sparse-view computed tomography, deep learning, precision learning, dual-domain, trainable kernel

I. INTRODUCTION

X-Ray Computed Tomography (CT) is a type of non-destructive testing widely employed in medical imaging applications for diagnostic purposes. The use of X-radiation as the imaging medium, however, poses a challenging risk to patients exposing them to ionizing radiation. Thus, one of the main goals of today’s research for X-ray CT is to reduce radiation exposure to patients. One approach geared towards lowering the dose is sparse-view CT (SVCT), which aims to decrease the number of measured exposures. Unfortunately, while this approach lowers the total radiation dose a patient is exposed to, an insufficiently sized set of projections has been shown to generate streaking artifacts in images reconstructed with conventional analytical methods [1].

A. Classical Tomographic Reconstruction Techniques

Analytical methods like the Filtered Backprojection (FBP) [2] for parallel-beam tomographic reconstruction and the FDK algorithm [3] for the cone-beam tomographic reconstruction case have been shown to perform well when full-view measurement data is available. With a higher projection sparsity, such methods generate unwanted streaks in the final image diminishing their diagnostic value [4]. Current research has been done towards training the FDK filtering weights with some success in de-streaking limited angle CT reconstructions using a data-driven fine-tuned sinogram filtering step [3].

Iterative reconstruction techniques have been proposed to replace analytical ones for SVCT. Such methods allow for the incorporation of a-priori information about the underlying problem, in the form of penalization terms, with the development of various regularization techniques for compressed sensing like total variation (TV) [5, 6, 2] and its versions [7, 8, 9], or even algorithms which enforce sparsity in other domains like wavelets [10] or curvelets [11]. A major disadvantage of the iterative methods, however, is the high computational cost of the repeated forward and backward projection operations during the iterative update steps [12]. Additionally, empirically tuning several hyperparameters (e.g. regularization strength, loss balancing, and number of iterations until convergence is achieved) is required for the algorithm to perform well.

B. Deep Learning based Denoising of Sparse-view CT

Deep learning (DL) methods have shown excellent performance in a multitude of different areas like classification [13].
denoising \cite{14}, or superresolution \cite{15} for clinical applications. Studies have shown that DL methods naturally learn representative low, middle and high level features in the given training dataset \cite{16} to correctly perform denoising for new inputs. This makes them the perfect candidate to be used towards reducing or (ideally) eliminating regular streaking artifacts present in SVCT reconstructions.

A nice overview of state-of-the-art DL-based methods for medical image processing is provided by Maier et al. \cite{17}, where the authors mention several competing solutions for SVCT, many of whom are based on the UNet or GAN architectures. Some of the first instances of such de-streaking algorithms can be traced back to FBPConvNet \cite{18}, which, as a standard encoder-decoder model, takes a sparse-view FBP reconstruction and is trained towards producing a de-streaked version of it. Zhang et al. extended FBPConvNet with residual connections, DD-Net \cite{19}, which has shown greater capability of suppressing noise and artifacts while recovering contrast information. Han et al. \cite{12} argued that conventional CNNs assume local invariance in the input image but the global nature of the streaking artifacts requires CNN architectures with large receptive fields. They proposed an encoder-decoder framelet-domain denoising method which overcomes the limitation of the small receptive fields imposed by the convolutional operation. Lee et al. \cite{20} improved on the framelet-domain architecture with sparsity enforcing wavelet-based data-driven denoising to enlarge the receptive field of conventional convolutions. More recently, generator-discriminator architectures, like the Generative Adversarial Networks, have been shown to produce superior results to UNet-based model both for sparse-view and for low-dose CT, in part thanks to their usage of feature-extracting adversarial loss instead of conventional pixel-based loss \cite{21} \cite{22}.

The major limitation to these methods, however, remains the fact that artifact denoising is performed directly on the reconstructed image. This single-domain approach disregards any prior knowledge encoded in the original measurements. Dong et al. \cite{23} proposed to perform sinogram-domain instead of image-domain denoising and then reconstruct an artifact-free image. However, this approach is again limited to performing in a single domain. New studies have shown that cascaded CNNs performing denoising in both projection and reconstruction greatly improve the quality of the reconstruction over single-domain approaches \cite{24} \cite{25}.

C. Combined Denoising-Reconstruction Algorithms

Precision Learning is a concept where known tomographic operators are included into the training process, thus enabling both measurement-domain and reconstruction-domain feature learning. Embedding the operator in the network usually leads to a reduction in the total number of trainable parameters and to a lower maximal training error \cite{26}.

Dual-domain denoising architectures for SVCT that take advantage of this concept have been proposed and successfully employed towards reducing streaking artifacts \cite{1}. The insertion of a reconstruction module into the network enables the full use of information in both projection and image domain \cite{27}. The majority of the proposed networks with such a feature use a fixed FBP/FDK layer as the domain-transform operation inserted between two encoder-decoder networks used to extract features in the sinogram- and reconstruction-domain: \cite{1} for sparse-view CT, \cite{28} for helical CT, \cite{4} for limited angle CT, and \cite{27} for metal artifact reduction. More recently, transformer-based architectures have been shown to outperform CNNs in image classification and denoising applications, one of such architectures being proposed by Shi et al. \cite{29}, who use a transformer encoder to extract features in image domain and a transformer decoder to link those features to the ones extracted from the sinogram. Another transformer-based architecture proposed by Wang et al. \cite{30} uses a Swin Transformer to merge sinogram and image features.

The filters used by all the aforementioned networks in the FBP/FDK layers are fixed, discretized analytical filters originally developed for full-view CT. Such filters are, however, sensitive to noise and discretization errors \cite{31}. As none of these filters are optimal for SVCT, learning specialized filters for individual CT use-cases like SVCT is required. Initial development of custom FBP filters was proposed by Pelt et al. \cite{32} who proved the quality improvement of data-dependent FBP filters optimized for individual CT cases. The first steps towards enabling DL-based data-driven fine-tuning of the reconstruction step can be found in the works of Syben et al. \cite{31}, who proposed doing data-based optimization to lower the discretization artifacts for conventional CT, and Würfl et al. \cite{3}, who proposed learning the filter weights of the FDK operation for limited-angle CT. Jiao et al. \cite{33} used a CNN network to learn the FBP filtering operation for the SVCT scenario.

Going even further, recent developments have re-introduced the regularized iterative reconstruction techniques with data-driven generated priors like the DRONE model or the DL-PICCS algorithm \cite{25} \cite{34}. While such architectures show great performance for SVCT, their data-driven aspect is only one part of the whole pipeline, while the iterative reconstruction algorithm governs the total computational cost. This impacts its applicability to 3D CT where thousands of slices must be reconstructed in a short period of time. Additionally, it also requires optimization of several hyperparameters on a case by case basis to ensure convergence.

In this study we propose an improved dual-domain SVCT denoising hybrid network, in which the projection- and image-domain denoising operations are performed by two encoder-decoder networks while the reconstruction is performed in a FBP layer with a trainable filtering operation. We expand on the methods we previously proposed in Boghiu et al. \cite{35}. We show that allowing fine-tuning of the FBP filter minimizes discretization and interpolation errors in the sinogram.

Current methods of sinogram upsampling rely on conventional interpolation methods (nearest neighbor interpolation \cite{25}, custom bilinear interpolation \cite{1}) or on a single-step DL-based upsampling-denoising method like the one proposed by Chao et al. \cite{36}. In contrast, we employ a “geometry-aware” interpolation method to great effect in producing sinograms at 4x the angular resolution of a sparse sinogram improving on the 3x upsampling limitation of the sparse sinogram mentioned
Two encoder-decoder networks are used to successfully suppress streaking artifacts in both domains, generating competitive results to the state-of-the-art.

Compared to current data-driven algorithms for the same purpose, the use of the fine-tuned filter significantly reduces upsampling artifacts and allows for the optimal amount of information to be passed to the image-domain denoising module. Jiao et al. [23] showed that a CNN-network is capable of learning the FBP filtering operation of the sinogram. Their sinograms do not require, however, any denoising, as they do not perform any measurement upsampling.

Furthermore, we show that the our WNet architecture diminishes the need for complex training and inference operations used by the state-of-the-art. On one hand, we don’t need WGAN based denoising modules with discriminative loss which reduces the total number of trainable parameters. On the other hand, doing iterative reconstruction based on deep learning priors requires empirical hyperparameter optimization on a case-by-case basis [23], which can potentially limit the generalizability of the model. WNet does not rely on any iterative denoising after the inference step as the dependency between real measurements and the learned output is embedded inside the network.

This paper is structured as follows: Section II contains a brief description of the sparse-view tomographic problem and introduces the proposed dual-domain architecture. Then, in Section III we present the results obtained with the proposed method and we compare against several state-of-the-art methods in the literature. We discuss the performance of the proposed method and its competitors in Section IV-A while in Section IV-B we dive into an interpretability analysis of our network, where we take apart each of its modules and investigate their performance and role in the whole reconstruction pipeline. We also take a brief look into the generalization performance of our network in Section IV-C and then offer an outlook to future studies on this topic in Section IV-D. Finally, we sum up the findings of this paper in Section V.

II. METHODS

SVCT is a special case of regular X-ray CT where a significantly smaller set of measurements is used to reconstruct the imaged specimen. The low number of measurements increases the ill-posedness of the tomographic inverse problem, which leads to a decrease in quality in the reconstructed image. This is reflected in the presence of streaking and discretization artifacts in the result [23]. In this section we present the mathematical details of the tomographic inverse problem and we continue with a motivation for the dual-domain denoising approach.

A. Sparse-view Computed Tomography

Given a set of measurements \( y_k \in \mathbb{R}^{k \cdot M} \), where \( k \in \mathbb{N} \) denotes the number of angles employed to obtain the measurements with \( M \in \mathbb{N} \) number of detector elements, we define the following discrete forward model for a tomographic reconstruction setup:

\[
A_k \cdot x = y_k
\]  

where \( x \in \mathbb{R}^N \) is the measured volume (with unknown inner structure) and \( A \in \mathbb{R}^{N \times (k \cdot M)} \) an operator encoding information about the X-ray image formation process and the acquisition geometry.

There are various methods to solve this inverse problem, but we will focus on the Filtered Backprojection (FBP) reconstruction with the Ram-Lak filter. We solve eq. (1) for \( x \) using the FBP algorithm:

\[
\tilde{x}_k = A_k^T W y_k
\]

where \( A_k^T \) is the backprojection operation (defined as the transpose of the forward operator) and \( W \in \mathbb{R}^{(k \cdot M) \times k \cdot M} \) a convolution operator describing the filtering operation performed on the post-log sinogram values.

B. Artifact Suppression and Reconstruction

As eq. (1) is underdetermined for small \( k \) (e.g. \( 32 \leq k \leq 128 \)) the reconstruction \( \tilde{x}_k \) will contain unwanted artifacts (streaks) [39]. A denoising operation \( U : \mathbb{R}^N \rightarrow \mathbb{R}^N \) takes an input plagued by streaking artifacts (like \( \tilde{x}_k \)) and generates an output with less or (ideally) no artifacts, while recovering original information hidden behind the streaks.

A data-driven approach to finding an optimal \( U_i \) translates to minimizing the following user-defined loss function:

\[
\arg \min_{U_i} \mathcal{L}(U_i(\tilde{x}_k), x)
\]

where \( x \) is usually an approximation of the original solution.

All the loss functions \( \mathcal{L}_s \) mentioned throughout this manuscript are Huber losses given by the following formula:

\[
\mathcal{L}_{s_{/r,j}}(x,y) = \begin{cases} \frac{1}{2}(x-y)^2, & \text{if } |(x-y)| < 1 \\ |(x-y)| - \frac{1}{2}, & \text{otherwise} \end{cases}
\]

The Huber loss was chosen as it combines the fast initial convergence of the L1 norm in early epochs [40], while for later epochs, where the loss is in the vicinity of 0, the L2 norm ensures “nice” and continuous gradients [41].

As this approach performs the denoising process in the reconstruction domain, original information encoded in \( y_k \) is possibly filtered out through the FBP reconstruction step. Finding a way to restore image information in the measurement domain set will ensure that an operator like \( U_i \) has more information to work with in the image domain. This means that we ought to first find a mapping from \( y_k \) to \( y_K \), with \( K = C \cdot k \), \( C \in \mathbb{N} \), \( C > 1 \), and then apply the \( U_i \) denoising operation to the corresponding FBP reconstruction \( \tilde{x}_K = A_k^T W y_k \) such that we obtain a reconstruction with less artifacts thanks to the increased sampling rate of the sinogram [23].

Hereby, we introduce a “geometry-aware” interpolation operation implemented as an FBP operation on the original \( y_k \) followed by a forward projection to a K-view sinogram. The interpolated sinogram is subjected to a measurement consensus step \( C(\cdot) \) where the original measurements from \( y_k \) are inserted into the obtained K-view sinogram \( y_{K}^{\text{int}} \in \mathbb{R}^{K \cdot M} \) at the corresponding angular locations (See Fig. II-B) for a visualization of the consensus step:

\[
y_{K}^{\text{int}} = \mathcal{I}(y_k) = C(A_k A_k^T W y_k) = y_{K}^{\text{int}}
\]
Fig. 1. (a) WNet Architecture Architecture of the joint dual-domain denoising and reconstruction network WNet which is composed of three modules: Sinogram-denoising module (SdM), Reconstruction Module (ReM), and Image-denoising Module (IdM) (b) Reconstruction Pipeline The WNet model in the context of the whole reconstruction pipeline; the preprocessing stage of the initial sparse-view sinogram consists of a geometry-aware interpolation followed by a measurement consistency step (c) UNet Architecture UNet-based architecture of the SdM and IdM in the WNet model with individual building blocks highlighted as arrows of different colors.

The consensus step $\mathcal{C}(\cdot)$ only allows for the preservation of original information in the interpolated sinogram. A denoising operator $U_s : \mathbb{R}^{K \times M} \rightarrow \mathbb{R}^{K \times M}$ is then required to properly handle the interpolation and discretization artifacts in the generated $y_{K}^{s}$ sinogram ($y_{K}^{s} = U_s(I(y_{k}))$), which can be obtained by minimizing the following loss function:

$$\hat{U}_s = \arg\min_{U_s} \mathcal{L}_s(U_s(I(y_{k})), y_{K})$$

subject to $(y_{K}^{s})_{(i,k),*} = (y_{k})_{i,*}$ for every $i \in \mathbb{N}$ enforced by $\mathcal{C}$. $y_{K} \in \mathbb{R}^{K \times M}$ is a real measured $K$-view sinogram used as the ground truth to compute the loss function. In other words, we train a denoising operator $U_s$ towards removing the interpolation artifacts produced by $I$ while preserving the originally measured angles $(y_{k})$ in the denoised sinogram $(y_{K}^{s})$.

Given the discrete nature of the tomographic inverse problem in eq. (1), a reconstruction generated with the FBP method is prone to certain corruptions commonly called cupping and DC shift artifacts [31]. Since widely used filters like the Ram-Lak or Cosine filters have been proven to perform well when the set of measurements is sufficiently large [42], FBP for SVCT with these types of filters is not ideal. Shi et al. [43] showed that specific CT use-cases (e.g. SVCT) require optimized filters, while Syben et al. [31] proposed to learn such filters in a data-driven process. Since we use the FBP algorithm to compute a reconstruction from $y_{K}^{s}$ we can optimize the FBP filtering operator $W$ towards reducing the aforementioned artifacts by minimizing the following loss function:

$$\hat{W} = \arg\min_{W} \mathcal{L}_r(A_K^T \hat{U}_s(I(y_{k})), x)$$

Having an optimal sinogram denoising operator $\hat{U}_s$ and an optimal convolution operator $\hat{W}$ the last step is to apply the operator $U_s$, introduced earlier, on the improved reconstruction
\[ \hat{x}_K \]. An optimal \( U_i \) minimizes the following loss:

\[
\hat{U}_i = \arg \min_{U_i} \mathcal{L}_i(U_i(A_K^W \hat{W} U_s(I(y_k)), x)) \quad (8)
\]

Putting everything together, we finally obtain a dual-domain optimally denoised reconstruction:

\[
\hat{x} = \hat{U}_i(A_K^W \hat{W} U_s(I(y_k))) \approx x \quad (9)
\]

C. Model Architecture

Fig. [1] shows the proposed WNet model architecture consisting of its three main modules while Fig. [1b] highlights every step involved in the WNet reconstruction pipeline.

The first module of the WNet, the Sinogram-denoising Module (SdM), uses an encoder-decoder type of network \( U_s \) to perform sinogram-domain denoising. The goal is to filter out the sparse-view and upsampling artifacts present in the interpolated sinogram \( y_k^\tau \) which was obtained with the geometry-aware interpolation operation (Fig. [1b]) from eq. [3]. The output of a trained \( \hat{U}_s \) is the denoised sinogram \( y_k^{\hat{U}_s} \) (also subject to the consensus step).

The output of \( \hat{U}_s \) is then fed to the next module, the Reconstruction Module (ReM), which handles the FBP reconstruction of \( y_k^{\hat{U}_s} \). The ReM has two steps:

1) a trainable convolution operation which uses the trained kernel \( \hat{W} \) to filter the input sinogram \( y_k^{\hat{U}_s} \) and generate \( y_k^{\hat{W}} \).

2) succeeded by a fixed tomographic backprojection operation, modeled as the transpose of the forward operator introduced in eq. [1], \( A^T \), producing \( \hat{x}_K^{\hat{W}} \).

The third and final module, the Image-denoising Module (IdM), contains a second encoder-decoder network \( U_i \) which is employed towards suppressing the sparse artifacts in \( \hat{x}_K^{\hat{W}} \), generating a final denoised image \( \hat{x} \). The UNet-based architecture of both encoder-decoder networks \( U_s \) and \( U_i \) is visualized in Fig. [1].

D. Data-driven FBP kernel weights

An important part of the training process is the fine-tuning step performed on the ReM module filter \( \hat{W} \). At the start of the training procedure, this filter was initialized with the values of a Ram-Lak filter [44] and iteratively fine-tuned during training.

In an ideal continuous setting, inverting the tomographic reconstruction problem corresponds to finding the inverse of the Radon transform, which is given by the continuous FBP operation with a ramp filter [31]. However, in a real scenario, the discrete noisy measurements produce artifacts in the final reconstruction, for example cupping and DC shift artifacts [51], besides the sparse artifacts. Therefore, analytical solutions have been proposed, where the Ram-Lak filter has been replaced by custom made filters, like the Cosine or the Shepp-Logan filter, which penalize the frequency domain in different ways (the aforementioned three filters can be seen in Fig. [5]). We followed up to the study performed by Würlf et al. [3] and enabled the fine-tuning of the FBP kernel during the training process.
Fig. 3. Reconstruction results of a clinically relevant slice for determining COVID-19 pneumonia in a patient selected out of the test dataset. (top set) enlarged region-of-interest (ROI) for each individual reconstruction for a detailed view onto the local feature recovery performance of each algorithm investigated. Overall location of the ROI is shown as a red box in the full slice Ground Truth image which is the 2048-view FBP reconstruction \( 2048 \) from Fig 2. The display window for all reconstruction patches is \([-1024,150]\) HU. (bottom set) Difference images to the ground truth corresponding to the output of each method. The display window for all generated difference images is \([0,400]\) HU.

### E. Training Procedure

We train the three modules of WNet in a round-robin fashion as outlined in Alg. 1. We first train \( U_s \) in SdM for five epochs towards minimizing the loss function \( \mathcal{L}_s \). This ensures that ReM is trained from the beginning on a properly \( U_s \)-denoised \( K \)-view sinogram \( y^C_K \). The FBP filter is updated towards minimizing \( \mathcal{L}_r \). The third and last layer, IdM, takes a reconstruction obtained from an FBP operation with the fine-tuned filter and is trained using \( \mathcal{L}_i \). For both \( \mathcal{L}_r \) and \( \mathcal{L}_i \) we approximate \( x \) with a full-view FBP reconstruction \( \hat{x}_{2048} \), as we will explain in more details in the next section.

The “jump-start” of all WNet modules is succeeded by a fine-tuning phase, where during each iteration all three modules are concurrently updated by backpropagating the \( \mathcal{L}_i \) loss (see while-loop in Alg. 1):

\[
{\hat{U}_s, \hat{W}, \hat{U}_i} = \arg \min_{U_s, W, U_i} \mathcal{L}_i(U_i(A^T_K W U_s(y^C_K)), \hat{x}_{2048})
\]

### III. Experiments and Results

#### A. Dataset

A total of 30 CT chest scans of patients suffering from COVID-19 pneumonia were acquired at Klinikum Rechts der Isar. Each scan was stripped of all the slices above and below the chest area which did not contain any lung region. All 30 scans were obtained with a Philips iCT 256 based on a full-dose protocol in the lung window and reconstructed at a resolution of 512 by 512 pixels with a slice thickness of 0.9mm and a pixel size of \( \approx 0.72 \text{mm} \). Data access was approved by the institutional ethics committee at Klinikum Rechts der Isar (Ethikvotum 87/18 S) and the data was anonymized.

To generate the required sparse sinograms and reconstructions we simulated an acquisition protocol based on a parallel-beam configuration with a detector resolution of 720 pixels. In this study we chose \( k = 128 \) uniformly sampled over a full-arc (360°) for the initial sparse-view sinogram \( y_{128} \). This sparse-view protocol creates diagnostically challenging images for the radiologists. For the sinogram interpolation step, we set \( C = 4 \), leading to \( K = 512 \) angles in a quarter-view sinogram configuration. A quarter-view label sinogram \( y_{512/12} \) was generated using the same parameters, as was a full-view label sinogram \( y_{2048} \).

All slices were obtained in HU units and normalized to pixel values \([0, 1]\). All subsequent operations are performed in this pixel value range.

We generated a set of FBP reconstructions \( \hat{x}_{2048} \) from the simulated full-view sinograms \( y_{2048} \) as labels for WNet. We chose not to use the slice data from the original scans, since this data has been processed and reconstructed in a blackbox clinical environment, with possibly additional post-reconstruction processing steps applied to the result. Therefore, generating matched forward- and filtered backprojections with a known reconstruction pipeline ensures compatibility of the input and label set and enables WNet to learn features related to the streaky nature of the SVCT images while remaining relatively transparent to the employed reconstruction parameters.

20 out of 30 scans were selected for the training set, while the other 10 were split in half and assigned to the validation

---

**Algorithm 1 Training pipeline**

- Train \( U_s \) in SdM for 5 epochs with \( \mathcal{L}_s \).
- Train \( W \) in ReM for 5 epochs with \( \mathcal{L}_r \).
- Train \( U_i \) in IdM for 5 epochs with \( \mathcal{L}_i \).

while k in 0, 1, \cdots do

Update \( U_s, W, U_i \) with \( \mathcal{L}_i \).

end while
and test set. We extend the training and validation set by applying an augmentation step to the individual slices which consists of a clockwise rotation of the original slice by 90°, 180°, and 270°, thus expanding both sets by a factor of 4. In total we train WNet on 22800 slices and we validate it on 4832 slices. In the end, we test our trained network on the 5960 slices in the test set.

For visualization purposes a physician chose a range of diagnostically relevant slices that display typical features of a COVID-19 pneumonia, which we show throughout this manuscript in Figs. 2, 3, 4, 5, 6. The COVID-19 related findings in the chosen slice are based on the ground glass opacity of mixed consolidation in the dorsal region of the right lung, which is enlarged in the region-of-interest (ROI) in (red square in Fig. 3) “Ground Truth”.

B. Initialization and Training

WNet consists of around 17.3M trainable parameters split up into ≈ 8.5M for \( U_s \), ≈ 8.5M for \( U_i \) and 1024 for \( W \). To initialize the weights of \( U_s \) and \( U_i \) we sample from the Xavier normal distribution \([47]\), while the biases are generated with samples from the uniform distribution. The FBP operation in ReM is implemented with a Fourier-domain based filtering step. The filter weights are initialized with the Fourier-domain values of the Ram-lak filter, similar to the scikit implementation of FBP (called \texttt{iradon} in the scikit framework) \([48]\). The backprojection operation is implemented as a differentiable Pytorch layer, extending the \texttt{torch.autograd} functionality with a custom backward gradient propagation operation \([49]\). The backend used for the matched forward and backward projectors is the C++17 tomographic reconstruction framework \texttt{elsa} \([50]\). The projector employed throughout the experiments is using Siddon’s method in a parallel-beam configuration \([51]\).

The Python library \texttt{scikit-image} \([48]\) was used in the preprocessing stage for resizing and performing augmentation and in the training and inference stage to compute image comparison metrics.

To update the parameters of all three modules we use the Adam optimizer (\( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \)) with a learning rate of \( 10^{-4} \) for each individual module. The batch size was set during the whole training process to 4. The number of epochs was 5 for the “jump-start” and then the whole network was fine-tuned for another 20 to ensure convergence, which was observed to be reached after 15 epochs.

The choice of the Huber function for \( \mathcal{L}_s \), \( \mathcal{L}_i \), and \( \mathcal{L}_r \) is motivated empirically. We trained a stand-alone SdM and IdM module five times with three loss functions: L1 Norm, L2 Norm and Huber Norm and we concluded that the validation performance is on average approximately the same for all three types of loss function. As we trained our models against Ram-Lak filtered FBP reconstructions (Ground Truth), the models trained with Huber norm generated images with a good balance of smoothing and noise properties.

\(^2\)refers to an area of increased attenuation in the lung with preserved inner structures \([56]\).

We compare the impact of three different loss functions on the results of the corresponding trained network in the “Supplementary Materials” Section D and Fig. A4.

All networks and additional experiments were trained and run on a NVIDIA Quadro RTX 6000 GPU with 24GB of VRAM. The underlying system was powered by an AMD EPYC 7452 32-Core CPU with 512GB RAM and running Ubuntu 20.04. The WNet was implemented in Pytorch 1.9 \([49]\) with a CUDA 11.0 backend.
C. Comparison to state-of-the-art

We compare results obtained with methods from three different generations of reconstruction algorithms on our fixed test set. Fig. [2] contains the example slice reconstructions with each individual method while a corresponding zoomed-in region-of-interest view can be found in Fig. [3]. The region-of-interest view is employed to shed more light on the localized performance of each algorithm. We also added Section A and Fig A1 in the “Supplementary Materials” showing results against FBP ground truths obtained with two other filters: Cosine and Shepp-Logan.

The analytical reconstruction with Filtered Backprojection (FBP) was obtained from the simulated sparse-view sinogram \( y_{128} \). Similarly, we ran 250 iterations of Weighted Least Squares with Total Variation regularization (WLS+TV) to obtain an iterative reconstruction from the same sparse-view set \( y_{128} \). The regularization parameter was empirically set to \( 10^5 \).

Next, as FBPCovNet is one of the first networks which tackled the de-artifacting of SVCT, we reimplemented the network from [18] in our framework and trained it for 100 epochs on pairs of sparse-view FBP reconstructions (input) and full-view FBP reconstructions (label) following the procedures described in [18].

The DD-Net was another network from the first generation of encoder-decoder models for SVCT that we investigated. We trained DD-Net for 50 epochs on the same input-label pairs as FBPCovNet while employing the original architecture, code, and setup instructions from [19].

Finally, we compared our results to the results obtained by the most recent type of architecture, DRONE [25]. The architecture of DRONE consists of four networks which perform sequentially the tasks of projection-domain and image-domain denoising on both images and residual images (see Fig. 1 from [25] for more details about the architecture of DRONE). All four networks, similar to one another, implement the UNet architecture, each one of them containing approximately 10.9M parameters, for a gross total of 43.8M parameters. Every network is trained sequentially, its input representing the generated output of the previous one.

The DRONE model employs custom forward and backward tomographic operators from the ASTRA Toolbox [52] both inside the network as well as for the final iterative denoising step, based on a half-arc two-dimensional fan-beam geometry. We adapted the original implementation of DRONE [25] with the forward-backward operators pair of \texttt{elsa}-based operators and we changed the half-arc fan-beam geometry with the full-arc parallel-beam geometry employed by WNet.

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{Metrics/Methods} & \text{FBP} & \text{WLS+L1} & \text{FBPCovNet} & \text{DDNet} & \text{DRONE} & \text{WNet (ours)} \\
\hline
\text{PSNR} & 26.13 \pm 1.31 & 33.37 \pm 0.81 & 34.29 \pm 1.26 & 35.47 \pm 1.13 & 36.27 \pm 1.28 & 37.10 \pm 1.35 \\
\text{SSIM} & 0.659 \pm 0.018 & 0.919 \pm 0.011 & 0.919 \pm 0.011 & 0.926 \pm 0.014 & 0.955 \pm 0.010 & 0.9577 \pm 0.015 \\
\text{Training duration} & \text{--} & \text{--} & 4h & 31h & 55h & 52h \\
\text{Reconstruction duration} & \approx 0.01s & \approx 8.0s & \approx 0.5s & \approx 0.3s & \approx 7.3s & \approx 1.4s \\
\# trainable parameters & 0 & 0 & \approx 31M & \approx 5.5M & \approx 43M & \approx 17.5M \\
\hline
\end{array}
\]

\textbf{TABLE 1} (1ST, 2ND ROW) QUANTITATIVE METRICS FOR OUR EXPERIMENTS (AVERAGE AND STANDARD DEVIATION VALUES OVER THE TEST SET) (3RD, 4TH ROW) RECONSTRUCTION DURATION AND NUMBER OF TRAINABLE PARAMETERS

to ensure the compatibility of the results. We then followed the training procedures listed in [25] and performed in the end the required iterative TV denoising step. As for the denoising hyperparameters, we empirically set \( \lambda_1, \alpha_1, \alpha_2, \beta_1 \) and \( \beta_2 \) to 0.01, 0.6, 0.6, 0.001 and 0.001 as the original values (0.05, 0.14, 0.6, 0.003 and 0.0003) used by [25] did not fit our dataset.

To quantitatively evaluate the quality of the different results, we use the Peak Signal-to-Noise Ratio (PSNR) [53] and the Structural Similarity Index Measure (SSIM) [54] implementations from the \texttt{scikit} library. We listed the mean and standard deviation values for PNSR and SSIM over the test set in Table I where we highlighted in boldface font the best values. We also added rows for the mean training and reconstruction durations and the number of trainable parameters, to support the method comparison.

D. Module Analysis

During the training procedure we noticed several interesting behaviors of the different submodules of WNet. Therefore, we performed an investigation into each individual WNet module to highlight aspects of the trained operators \( \hat{U}_s, \hat{U}_i \) and \( \hat{W} \).

Fig. [4] contains the corresponding reconstructions (FBP with Ram-Lak filter) to each individual sinogram preprocessing steps in the WNet pipeline shown in Fig. [1]. To highlight the role of each step both in general and in local feature recovery we added a zoomed-in view into one of the regions of interest and the corresponding sinogram and sinogram difference images. The “Supplementary Materials” contain Section B and corresponding Fig. A2 discussing the difference between bilinear and geometry-aware interpolation for sinogram upsampling.

We illustrate the learned FBP filter from ReM and we plot it against three other widely used kernels in conventional CT applications. Reconstructions with matching colors to the original values \( \beta \) are shown in Fig. [5a]. We then followed the training procedures listed in [25] and performed in the end the required iterative TV denoising step. As for the denoising hyperparameters, we empirically set \( \lambda_1, \alpha_1, \alpha_2, \beta_1 \) and \( \beta_2 \) to 0.01, 0.6, 0.6, 0.001 and 0.001 as the original values (0.05, 0.14, 0.6, 0.003 and 0.0003) used by [25] did not fit our dataset.

To quantitatively evaluate the quality of the different results, we use the Peak Signal-to-Noise Ratio (PSNR) [53] and the Structural Similarity Index Measure (SSIM) [54] implementations from the \texttt{scikit} library. We listed the mean and standard deviation values for PNSR and SSIM over the test set in Table I where we highlighted in boldface font the best values. We also added rows for the mean training and reconstruction durations and the number of trainable parameters, to support the method comparison.

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{Metrics/Methods} & \text{FBP} & \text{WLS+L1} & \text{FBPCovNet} & \text{DDNet} & \text{DRONE} & \text{WNet (ours)} \\
\hline
\text{PSNR} & 26.13 \pm 1.31 & 33.37 \pm 0.81 & 34.29 \pm 1.26 & 35.47 \pm 1.13 & 36.27 \pm 1.28 & 37.10 \pm 1.35 \\
\text{SSIM} & 0.659 \pm 0.018 & 0.919 \pm 0.011 & 0.919 \pm 0.011 & 0.926 \pm 0.014 & 0.955 \pm 0.010 & 0.9577 \pm 0.015 \\
\text{Training duration} & \text{--} & \text{--} & 4h & 31h & 55h & 52h \\
\text{Reconstruction duration} & \approx 0.01s & \approx 8.0s & \approx 0.5s & \approx 0.3s & \approx 7.3s & \approx 1.4s \\
\# trainable parameters & 0 & 0 & \approx 31M & \approx 5.5M & \approx 43M & \approx 17.5M \\
\hline
\end{array}
\]

\textbf{TABLE 1} (1ST, 2ND ROW) QUANTITATIVE METRICS FOR OUR EXPERIMENTS (AVERAGE AND STANDARD DEVIATION VALUES OVER THE TEST SET) (3RD, 4TH ROW) RECONSTRUCTION DURATION AND NUMBER OF TRAINABLE PARAMETERS

E. Generalization

One key aspect of DL based methods is their data-driven nature and high dependency on a training set representative of the whole modeled distribution of inputs. Having an insufficiently large training set which does not contain all the representative information for the whole expected input domain, leads to poor performance of the neural network on samples with new features not represented in the training set. Therefore, understanding how a neural network “learns” to extract features becomes an important task in order to get insights into the training process and the generalizability performance of the trained model on new data.

We perform a short investigation into the performance of the trained WNet and DRONE on three out-of-distribution slices, which were part of neither the training nor the validation or the test set.

Fig. 7 contains slices extracted from a CT scan belonging to an ex-post clinical study of deceased patients at Klinikum Rechts der Isar. The acquisition was performed with a high dose CT Protocol with an exposure time of 497ms at a nominal tube current of 512 mA and 120 kV, and the images were reconstructed with a sharp reconstruction kernel in the lung window setting. A Philips iCT 256 scanner was used with a resolution of (768, 768) pixels at a voxel size of (0.4362, 0.4362, 0.9) mm. Images were downsampled to (512, 512) pixels.

We attached generalization comparisons of the same resulting images to Ram-Lak filtered ground truths in the “Supplementary Materials” Section C and corresponding Fig. A3.

IV. DISCUSSION

A. Comparison to state-of-the-art

The results in Fig. 2 and 3 show that outputs of convolutional-based networks tend to share similarities in feature reconstruction performance like the over-smoothing of edges and loss of small and very-small details. The contrast of the image produced by the DD-Net is higher than the FBPConvNet result, with smaller details visible, which is also reflected in the increase in PSNR by 1 dB and in SSIM by 1%, with the major difference being that DD-Net requires $\times 54$ less parameters than FBPConvNet and is $\times 10$ faster, proving the potential of residual connections in such scenarios. However, DD-Net images are still plagued by low sharpness and the presence of oversmooth regions.

The DRONE algorithm seems to have taken care of various drawbacks of previous convolutional-based methods, like over-smooth regions, blurry edges and loss of detail at the small and very-small scale, however, at the cost of network size and computational complexity. For the reconstruction of the slice in Fig. 2 the DRONE pipeline requires around 73s to complete one slice while WNet needs less than 2s to perform the corresponding inference step (on the same hardware in similar conditions). In a clinical setting, where time is of high importance, the DRONE network would require about 7h to reconstruct a 300 slice Chest CT volume, while our method would only need $\approx 6$ min at a comparable image quality. The difference in visual quality of the reconstructions is mostly influenced by the display window recommended by radiologists for chest CT. As both DRONE and WNet are data-driven algorithms, their weight initialization plays a role in

\[ \text{Supplementary Materials} \]
the final output of the network. As a consequence, some tiny
details in the denoised reconstructions having values at the
extremes of the display window spectrum, can appear to be
missing from the final result. The quantitative metrics shown
in Figs. 2, 4 and 6 however, are a better measure of image
quality and are not influenced by the windowing procedure.

While the DRONE and WNet results are comparable in
terms of contrast, resolution and overall image quality, WNet
generates the image with less than half the number of pa-
rameters in DRONE and most notably at a fraction of the
reconstruction time (see Table I), which lies in the same
order of magnitude with the inference times of other methods
like FBPConvNet or DD-Net. Most notably, WNet does not
require any hyperparameter tuning for the iterative denoising
or reconstruction process like WLS+L1 or DRONE do, thus
making it more efficient for similar use-cases where parameter
tweaking might be necessary for a DRONE-like pipeline.

Last but not least, we improved on the factor of 3 increase in
interpolated sinogram resolution mentioned in [25] (which is
limited by the nearest neighbor interpolation method employed
by the authors) by showing that sinogram upsampling by a
factor of 4 (from 128-view to 512-view sinograms) is indeed
feasible in the context of SVCT.

B. Network Interpretability

Progress in the development of new deep learning methods
for image processing is often a slow process based on small
incremental steps from previous model iterations. Therefore,
a key to understanding the way WNet "works" is to analyze
the outputs of its individual modules.
1) Sinogram-domain Denoising: In Fig. 4 it is noticeable that the geometry-aware interpolation \( \mathcal{T} \) coupled with the measurement consensus step \( C \), reduces some of the streaking artifacts seen in the original 128-view sparse reconstruction \( \hat{x}_{128} \). This quality improvement is also reflected in the quantitative measures of the two images: an increase of approx. 2dB in PSNR and 7% in SSIM for \( \hat{x}_{512} \) over \( \hat{x}_{128} \) (eq. 5). Only an optimal sinogram-denoising operator \( \hat{U}_s \) significantly reduces the amount of streaking and interpolation artifacts in the sinogram, such that in its corresponding reconstruction \( \hat{x}_{512}^U \), contrast and features are remarkably restored to some extent (see Sec. II-B for details on how \( \hat{U}_s \) is computed). The PSNR and SSIM metrics reflect the increase in information recovered in the output with an increase of another 7dB in PSNR and 20% in SSIM over \( \hat{x}_{512}^U \).

2) Reconstruction module fine-tuning: As shown in Fig. 5 the learned filter reveals similarities to other filters in the low-frequency domain. It starts to diverge from the Ram-Lak filter once it reaches the medium-frequency sector, its values lying close to the Cosine filter. In the high-frequency domain, coefficients diverge from the Cosine filter values towards the Shepp-Logan kernel. Note that, the learned filter is optimized for the specific sparse-view reconstruction problem (\( k = 128 \) and \( K = 512 \)) and might require re-training for a different sinogram sparsity factor.

3) Image-domain Denoising: The results in Fig. 6 show that \( U_i \) is a necessary step to handle the “left-over” sparse-artsifacts from the previous modules. The natural features present in the image, compared to a sinogram, enable this module to learn more meaningful representations and to recover small and very-small details in the image, thus creating a better result both qualitatively and quantitatively.

C. Generalization

In Fig. 7 we consider that the better generalizability performance of the DRONE model is a direct result of its iterative denoising step which can suppress overfitting artifacts generated by the trained network. As WNet lacks such an iterative step and as the training set does not cover head and abdomen data, its results won’t match the quality of those obtained with DRONE. However, the difference in computation time and network size is still the same, with DRONE taking over 70s to finish computation while WNet under 2s.

D. Future Work

Embedding the reconstruction step inside the deep learning architecture and enabling the training of reconstruction parameters is a valuable tool that opens new possibilities for more complex approaches like variational layers or unrolled regularized iterative algorithms. Furthermore, the two encoder-decoder modules in WNet, can be extended with additional residual connections, as it can be seen in DD-Net, or can be replaced by competitive WGAN implementations with feature losses as a future improvement.

Fig. 7. Generalization results of DRONE and WNet for scans that were acquired and reconstructed with different parameters compared to the ones in Fig. 2 (see Sec. II-B). The SSIM and PSNR values with respect to the 2048-view Cosine filtered FBP reconstruction “Ground Truth” were added in the top-left of each image (magenta). The display window for the Abdomen results is [-350,450] HU, for the Chest results is [-1024,150] HU, and for the Head results is [-40,120] HU. The display window for the difference images is [0,400] HU.

V. Conclusion

We propose a DL-based denoising method for Sparse-view Computed Tomography, enabling joint training of both the denoising operators and the projection filtering operation performed as part of the FBP step. In this study, we have focused on a particular SVCT setup with a relatively low number of measurements, and we have shown consistent good results compared to state-of-the-art methods. We have also shown that our method competes with iterative algorithms with a deep learning prior like the DRONE algorithm, while featuring a significantly reduced computational cost and time.

We also show that a trained data-driven precision learning approach still has the ability to generalize quite acceptably to new data obtained with a different acquisition protocol.
