Received: 31 January 2022 | Revised: 25 April 2022 | Accepted: 11 May 2022

DOI: 10.1002/mp.15718

RESEARCH ARTICLE

MEDICAL PHYSICS

Ultralow-parameter denoising: Trainable bilateral filter layers in computed tomography

Fabian Wagner[1] | Mareike Thies[1] | Mingxuan Gu[1] | Yixing Huang[1] | Sabrina Pechmann[2] | Mayank Patwari[1] | Stefan Ploner[1] | Oliver Aust[3,4] | Stefan Uderhardt[3,4] | Georg Schett[3,4] | Silke Christiansen[2,5] | Andreas Maier[1]

[1]Pattern Recognition Lab, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen 91058, Germany

[2]Fraunhofer Institute for Ceramic Technologies and Systems IKTS, Forchheim 91301, Germany

[3]Department of Internal Medicine 3 - Rheumatology and Immunology, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen 91054, Germany

[4]University Hospital Erlangen, Erlangen 91054, Germany

[5]Institute for Nanotechnology and Correlative Microscopy e.V. INAM, Forchheim 91301, Germany

Correspondence
Fabian Wagner, Pattern Recognition Lab, Friedrich-Alexander Universität Erlangen-Nürnberg, Erlangen 91058, Germany.
Email: fabian.wagner@fau.de

Funding information
European Research Council, Grant/Award Number: 810316

Abstract

Background: Computed tomography (CT) is widely used as an imaging tool to visualize three-dimensional structures with expressive bone-soft tissue contrast. However, CT resolution can be severely degraded through low-dose acquisitions, highlighting the importance of effective denoising algorithms.
Purpose: Most data-driven denoising techniques are based on deep neural networks, and therefore, contain hundreds of thousands of trainable parameters, making them incomprehensible and prone to prediction failures. Developing understandable and robust denoising algorithms achieving state-of-the-art performance helps to minimize radiation dose while maintaining data integrity.
Methods: This work presents an open-source CT denoising framework based on the idea of bilateral filtering. We propose a bilateral filter that can be incorporated into any deep learning pipeline and optimized in a purely data-driven way by calculating the gradient flow toward its hyperparameters and its input. Denoising in pure image-to-image pipelines and across different domains such as raw detector data and reconstructed volume, using a differentiable backprojection layer, is demonstrated. In contrast to other models, our bilateral filter layer consists of only four trainable parameters and constrains the applied operation to follow the traditional bilateral filter algorithm by design.
Results: Although only using three spatial parameters and one intensity range parameter per filter layer, the proposed denoising pipelines can compete with deep state-of-the-art denoising architectures with several hundred thousand parameters. Competitive denoising performance is achieved on x-ray microscope bone data and the 2016 Low Dose CT Grand Challenge data set. We report structural similarity index measures of 0.7094 and 0.9674 and peak signal-to-noise ratio values of 33.17 and 43.07 on the respective data sets.
Conclusions: Due to the extremely low number of trainable parameters with well-defined effect, prediction reliance and data integrity is guaranteed at any time in the proposed pipelines, in contrast to most other deep learning-based denoising architectures.

KEYWORDS
bilateral filter, denoising, known operator learning, low-dose CT

1 INTRODUCTION

Ionizing radiation used in computed tomography (CT) can cause stochastic effects in living tissue. Therefore, the amount of deposited energy in each investigated sample, the so-called dose, is to be minimized following the *As Low As Reasonably Achievable* (ALARA) principle.[1] However, noise in CT acquisitions is determined by the number of x-rays penetrating the scanned tissue. Further decreasing patient dose by reducing the radiation exposure results in degraded image quality due to increased Poisson noise in CT projections.[2]

As the emergence of the first CT scanners, denoising algorithms were developed and applied to restore image quality while keeping the radiation dose moderate.[3] Nonlinear filters[4–7] and iterative reconstruction techniques[8–10] have been successfully applied. Although such approaches usually require hand-tuned hyperparameters, cannot abstract complex features, or are known to be computationally expensive, purely data-driven, end-to-end trainable, approaches have been proposed in recent years[11–17] fueled by the emergence of deep learning and, in particular, convolutional neural networks. Most of these models achieve competitive denoising performance but are built on deep neural networks with multiple layers and contain hundreds of thousands of trainable parameters. Although deep architectures help networks extract complex features from data, such models are often regarded as black boxes as it is impossible to fully comprehend their data processing and control failing network predictions. Besides, adversarial examples in the form of small perturbations of the network input can lead to undesired drastic changes in the prediction due to the high dimension of extracted features.[18–20] Such uncertainties often prohibit deep learning applications in the medical imaging field where the reliability of the data must be maintained.[21,22] Additionally, training large numbers of parameters in neural networks usually requires broad medical data sets with paired ground truth data, which can be hard to obtain.

The bilateral filter has been successfully applied on CT data[6] as it performs combined filtering in spatial and intensity domain with Gaussian kernels smoothing the noise fluctuations while preserving edges.[23] However, hyperparameters—fundamentally determining the filter performance—usually have to be hand-picked. Multiple works have been proposed aiming for automatically finding optimal filter parameters[24–28] using risk estimators or grasshopper optimization. All these techniques are not suitable for integration into an end-to-end, data-driven, optimization pipeline as they are based on different statistical assumptions on the noise and do not support gradient-based optimization that has been demonstrated particularly effective through deep learning applications. A different work presents a bilateral filter based on convolutional filtering the permutohedral lattice, a higher dimensional representation of the image data, which can be optimized.[29] However, their algorithm requires so-called *splatting* and *slicing* operations, which can only approximate the data, and therefore, introduce uncertainties. Additionally, the requirement of a higher dimensional grid increases computational demands. Patwari et al.[12] proposed the JBFnet, a deep neural network architecture inspired by joint bilateral filtering that can be trained in a purely data-driven way. They showed competitive performance to much deeper networks, although reducing the number of trainable parameters by choosing a shallow convolutional architecture. Besides, denoising approaches indirectly optimizing hyperparameters of (joint) bilateral filters using reinforcement learning have been proposed.[30,31] However, their training is more sophisticated as it is again based on deep architectures for choosing the correct parameter updates. Additionally, Patwari et al.[31] make use of a residual network to generate an image prior as well as a reward network.

We aim to extend the aforementioned approaches by proposing a trainable bilateral filter layer using only the inherent four filter parameters (including three spatial filter dimensions) during training and inference. By analytically deriving the gradient flow toward the parameters as well as to the layer input, we can directly optimize all filter parameters via backpropagation and incorporate the C++/CUDA-accelerated layer into any trainable pipeline using the *PyTorch* framework.[32] Additionally, we show simultaneous optimization of filter parameters in the projection and image domain, using a differentiable backprojection layer.[33] Due to the very low number of trainable parameters, we can optimize our pipeline with only very little training data and in a self-supervised manner using Noise2Void training,[34] while still achieving competitive performance compared to state-of-the-art deep neural networks. We explain the competitive denoising performance with the theoretical findings of Maier et al.,[35,36] who proved that incorporating prior knowledge into artificial neural networks lowers the upper error bound of the model prediction. The experiments are performed on the 2016 Low Dose CT Grand Challenge data set[37] (25% dose) for benchmark purposes as well as on a low-dose x-ray microscope (XRM) bone data set (10% dose).

2 MATERIALS AND METHODS

2.1 Trainable bilateral filter

The bilateral filter had great success in CT denoising due to its ability to smooth image content while preserving edges. This is achieved by a composed filter kernel with spatial and intensity range contributions. With the noisy input reconstruction **X** and the denoised prediction

$\mathbf{Y}$, the discrete filter operation can be written as[23]

$$\hat{Y}_k = \frac{1}{w_k} \underbrace{\sum_{n \in \mathcal{N}} G_{\sigma_s}(\mathbf{p_k} - \mathbf{p_n}) G_{\sigma_r}(X_k - X_n) X_n}_{=:\alpha_k} \tag{1}$$

with the definition of the normalization factor w_k

$$w_k := \sum_{n \in \mathcal{N}} G_{\sigma_s}(\mathbf{p_k} - \mathbf{p_n}) G_{\sigma_r}(X_k - X_n), \tag{2}$$

voxel index $k \in \mathbb{N}$, and the Gaussian function

$$G_{\sigma_r}(c) := \exp\left(-\frac{c^2}{2\sigma_r^2}\right). \tag{3}$$

Each predicted output voxel $\hat{Y}_k$ is calculated from the neighborhood $\mathcal{N}$ of voxel X_k, indexed by $n \in \mathcal{N}$. The spatial kernel G_{σ_s} performs image smoothing and is dependent on the distance between the positions $\mathbf{p_k} \in \mathbb{N}^d$ and $\mathbf{p_n} \in \mathbb{N}^d$. In the d-dimensional case, G_{σ_s} is composed of multiple Gaussian kernels, for example,

$$G_{\sigma_s}(\mathbf{c}) = \prod_{s \in \{x,y,z\}} \exp\left(-\frac{c_s^2}{2\sigma_s^2}\right) \tag{4}$$

assuming $d = 3$ spatial filter dimensions and the spatial distance vector $\mathbf{c} = (c_x, c_y, c_z)$. The additional intensity range kernel G_{σ_r} is responsible for preserving edges during filtering as similar voxel values are weighted more heavily by incorporating the intensity distance $(X_k - X_n)$.

Kernel widths σ_s and σ_r are hyperparameters of the bilateral filter and are usually hand-picked by the user dependent on image content and voxel intensity range. However, tuning the filter layers by hand is cumbersome, and finding an optimal parameter set cannot be guaranteed. To optimize the parameters in a data-driven fashion and incorporate them in a fully differentiable pipeline, the gradient, given by the derivative of the loss function L with respect to each parameter σ_γ, must be calculated

$$\frac{\partial L}{\partial \sigma_\gamma} = \frac{\partial L}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \sigma_\gamma} = \sum_k \frac{\partial L}{\partial \hat{Y}_k} \frac{\partial \hat{Y}_k}{\partial \sigma_\gamma}. \tag{5}$$

Automatically deriving the gradient with a framework like *PyTorch* is in principle possible, but comes with a huge computational and memory overhead. It would require calculating gradients for each computation in the filter forward pass, which is computationally infeasible with conventional hardware in a reasonable run time. Instead, we directly calculate the analytical solution of $\frac{\partial \hat{Y}_k}{\partial \sigma_\gamma}$, which is described in detail in the Appendix. Additionally, the gradient with respect to every voxel of the noisy input reconstruction X_i needs to be derived to propagate a loss into previous filter layers during backpropagation to allow stacking multiple bilateral filters or incorporating the layer into a deep architecture

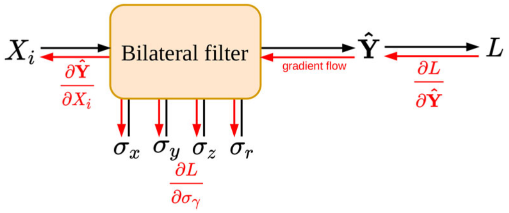
X_i
Bilateral filter
$\hat{\mathbf{Y}}$
L
$\frac{\partial \hat{\mathbf{Y}}}{\partial X_i}$
gradient flow
$\frac{\partial L}{\partial \hat{\mathbf{Y}}}$
σ_x σ_y σ_z σ_r
$\frac{\partial L}{\partial \sigma_\gamma}$

FIGURE 1 Working scheme of the proposed trainable bilateral filter layer. Black arrows mark the forward pass, whereas red arrows illustrate the gradient flow toward the input X_i and the filter parameters σ_γ during backpropagation

$$\frac{\partial L}{\partial X_i} = \frac{\partial L}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial X_i} = \sum_k \frac{\partial L}{\partial \hat{Y}_k} \frac{\partial \hat{Y}_k}{\partial X_i}. \tag{6}$$

With each predicted $\hat{Y}_k$ being dependent on intensity differences of two input voxels X_k and X_n, the analytical calculation of the gradient flow toward the filter input

$$\frac{\partial \hat{Y}_k}{\partial X_i} = -w_k^{-2} \alpha_k \frac{\partial w_k}{\partial X_i} + w_k^{-1} \frac{\partial \alpha_k}{\partial X_i} \tag{7}$$

is more complex than the derivative with respect to the kernel widths σ_γ. Eventually, it requires distinguishing the two cases $k = i$ and $k \neq i$ for finding an applicable solution. They refer to the off-center and center elements of the bilateral filter kernel, respectively. The detailed calculation is again conducted in the Appendix, accompanied by implementation notes. Figure 1 illustrates the data flow in a bilateral filter layer. The forward and backward pass of the layer is implemented in C++ and CUDA to leverage performance and integrated into the *PyTorch* framework using the *pybind11* module.[38] Our open-source filter layer is publicly available under https://github.com/faebstn96/trainable-bilateral-filter-source.

2.2 | Denoising pipelines

Multiple denoising pipelines containing trainable bilateral filter layers, as illustrated in Figure 2, are employed to investigate the performance and limitations of the layers. First, a simple postprocessing approach is studied, directly applying three subsequent bilateral filter layers on the reconstructed volume. The filters are optimized for the mean squared error (MSE) loss calculated between prediction $\mathbf{Y}$ and high-dose volume $\mathbf{Y}$.

Additionally, self-supervised training is performed using the Noise2Void (N2V) training scheme. The noisy volume is perturbed by randomly replacing a defined

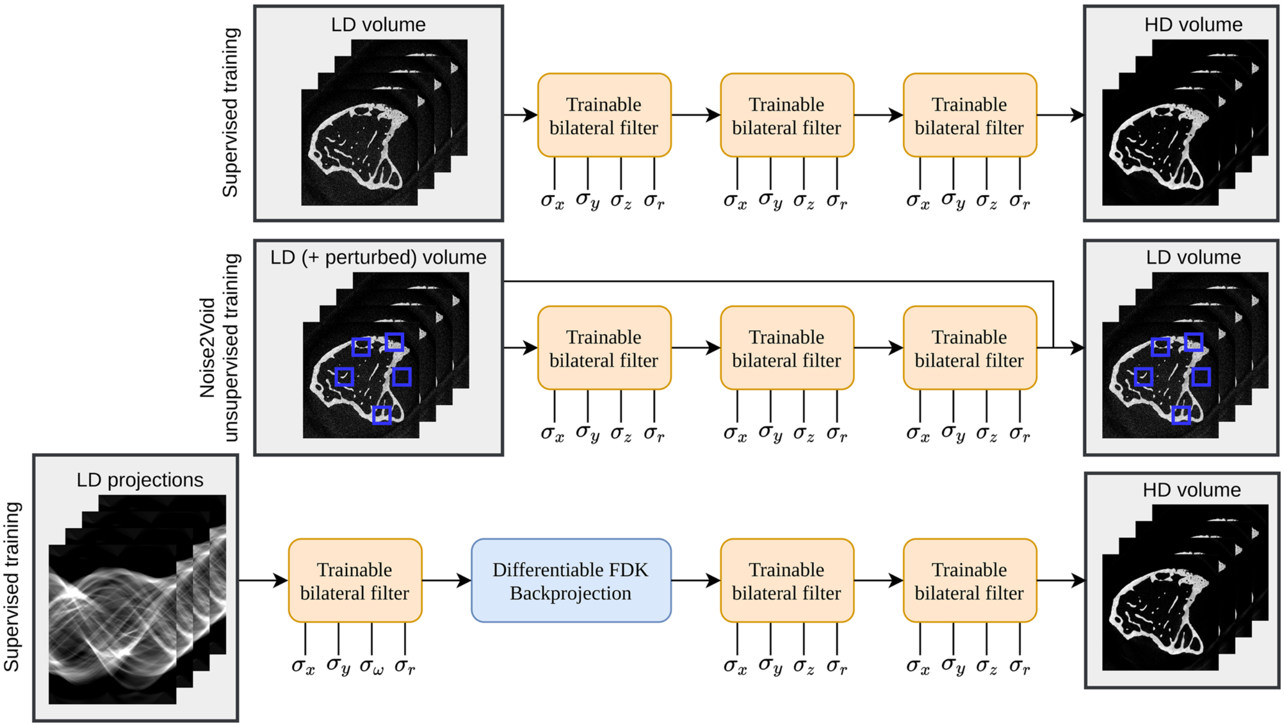
Supervised training
LD volume
Trainable bilateral filter
Trainable bilateral filter
Trainable bilateral filter
HD volume
σ_x σ_y σ_z σ_r
Noise2Void unsupervised training
LD (+ perturbed) volume
LD volume
Supervised training
LD projections
Differentiable FDK Backprojection
σ_x σ_y σ_ω σ_r
HD volume

FIGURE 2 Illustration of the employed denoising pipelines containing the proposed bilateral filter layers. Multiple filters are trained in a supervised (first row) and self-supervised (Noise2Void,[34] second row) fashion. Additionally, bilateral filter layers are optimized in projection and image domain simultaneously using a differentiable backprojection operator (sin & reco BFs, third row). Note that in the projection domain, filtering is also performed in angular direction via σ_ω. The σ_γ parameters represent the only trainable parameters of the networks and are all optimized independently. HD refers to the high-dose reconstructions, whereas LD denotes the low-dose data incorporating noise

ratio of voxels (1 %) with voxel intensities from their respective 5^3 voxel neighborhoods and fed through the denoising pipeline. Subsequently, an MSE loss is calculated between prediction and noisy nonperturbed volume at the voxel positions of the perturbations. If the noise contribution of neighboring voxels is approximately uncorrelated, the updated model will converge to predict the denoised version of the volume, as proved by Krull et al.[34] In reality, noise between reconstructed voxels cannot be regarded as uncorrelated, as information from CT projections is spread over the entire reconstruction. However, noise in low-dose CT reconstructions is still a local phenomenon that motivates experiments using N2V training. The advantage of the N2V training scheme is that no paired high-dose data are required. Accordingly, no additional ground truth knowledge can be employed during training.

Finally, a Feldkamp, Davis, and Kress (FDK) algorithm-based reconstruction pipeline with bilateral filter layers in the projection domain and filters in the image domain is trained end-to-end on XRM data. The employed pipeline is illustrated in the last row of Figure 2 and makes use of the fully differentiable reconstruction pipeline of Thies et al.,[39] which is described in the following section. All filter parameters are optimized for the MSE loss between **Y** and the high-dose reconstruction **Y** in the image domain.

2.3 | XRM reconstruction pipeline

The FDK-based[40] reconstruction pipeline by Thies et al.[39] connects acquired CT projection data with its 3D interpretable reconstruction. The backprojection is incorporated as a known operator,[35] using the differentiable layer provided by the *Pyro-NN* framework by Syben et al.[33] The reconstruction pipeline is adapted to XRM projection data from a Zeiss Xradia 620 Versa microscope, which is a high-resolution cone-beam CT for small samples, and allows to propagate a loss calculated in the image domain to any location within the reconstruction pipeline enabled by its differentiable implementation. Accordingly, bilateral filter layers can be trained at multiple locations in the pipeline in a purely data-driven manner using the *PyTorch* framework. Particularly, we perform experiments demonstrating the superior performance of bilateral filter layers applied in both projection and image domain simultaneously over denoising in image domain only.

2.4 | Experimental setup

We compared the performance of the trainable bilateral filter layer-based pipelines to four deep state-of-the-art denoising architectures on the 10 abdomen scans from the 2016 Low Dose CT Grand Challenge data set[37] with 1 mm slice thickness reconstructed in slices of size 512 × 512. Here, patient *L291* was selected for validation, patients *L310, L333*, and *L506* for testing and all 3411 slices from the remaining seven scans for training.

In a second experiment, we investigated the denoising performance on a data set of high-resolution ex-vivo XRM scans of mouse tibia bones. In the context of preclinical osteoporosis research, lacunae, bone structures with a diameter of 3–20 μm were investigated.[41] This is instructive for understanding the bone metabolism[42,43] as they contain osteocytes, cells being heavily involved in the bone remodeling process.[44,45] However, the huge amount of radiation dose required for resolving micrometer-sized lacunae in a mouse so far prohibits in-vivo imaging,[46–48] and therefore, requires effective dose reduction techniques to push the XRM in-vivo limit to the micrometer scale. We created a low-dose XRM data set from ex-vivo mouse tibia bone samples to investigate in-vivo XRM acquisitions.[49] The sample preparation was carried out as approved by the local animal ethics committee in compliance with all ethical regulations (license TS-10/2017). A total of 1400 high-resolution projections were acquired for every sample in a short-scan setting with 28 s exposure time per projection image. Low-dose and high-dose projections were reconstructed in a 2048^3 grid of 1.4 μm isotropic voxel size using the pipeline of Thies et al.[39] Data acquired with 10 % dose were simulated by incorporating noise following Yu et al.[50] Two separate stacks of 30 slices each were reconstructed from every bone scan as the total amount of reconstructed slices would by far exceed the capacity of the denoising models and the variety of learnable local features. The final XRM bone data set contains five scans for training, one for validation, and four for testing, which is in total equivalent to 9600 512 × 512 image patches, exceeding the size of the 2016 Low Dose CT Grand Challenge data set.

The quadratic autoencoder architecture (QAE) proposed by Fan et al.,[13] as well as the JBFnet by Patwari et al.,[12] both achieved remarkable results on the CT denoising task, outperforming many architectures proposed during the 2016 Low Dose CT Grand Challenge. The RED-CNN network, published by Chen et al.,[11] is a deep, convolutional architecture with more than 1.8 Mio parameters achieving state-of-the-art results likewise. Another denoising approach based on a generative adversarial network is WGAN,[14] using a combination of Wasserstein and perceptual loss. Encouraged by their performance, these four methods were selected for comparison with multiple pipelines containing the proposed trainable bilateral filter layers on the aforementioned data sets.

In order to allow a fair comparison between the different denoising approaches, the networks were implemented and trained based on officially published code repositories and the description from their publications. The Adam optimizer with learning rates 4×10^{-4} (decaying), 1×10^{-4} (constant), 1×10^{-4} (decaying), and 1×10^{-5} (constant) was used for QAE, JBFnet, RED-CNN, and WGAN, respectively. Multiple training runs with the reported learning rates from the respective reference publication varied by factors of $0.1, 0.5, 1, 2, 10$ were performed and the best-performing method on the validation data was selected. Both QAE and RED-CNN were trained using the MSE loss, whereas JBFnet makes use of a custom loss described in their publication[12] as well as pretraining of its prior module. Generator and discriminator of WGAN were trained alternately using Wasserstein loss for stable convergence and a perceptual loss as presented in their work.[14] For the WGAN, different training strategies were explored, varying the relative number of optimization iterations between generator and discriminator. We found that conducting four discriminator optimization steps followed by one generator step performed best, which represents the configuration also chosen by Yang et al.[14] All models were trained until the validation loss did not improve for five consecutive epochs.

In all bilateral filter-based models, $\sigma_{x,y,z}$ and σ_r were initialized with 0.5 and 0.01 and optimized with two separate Adam optimizers with learning rates 0.01 and 0.005, respectively, as the spatial and intensity range parameters operate on two independent scales. We tried different parameter initializations, but found that after convergence of all parameters, the networks' performances turned out to be very similar. The tiny amount of required training data was demonstrated by training the bilateral filter pipelines only on a single stack of 21 neighboring slices with a size of 512 × 512 voxels from one training scan of the Grand Challenge data set. For the XRM bone data set, a stack of 15 neighboring patches of size 512 × 512 voxels was used. Note that the bilateral filter layers cannot overfit the data assuming a comparable amount of noise within all scans due to the low number of trainable parameters with well-defined influence. Optimization was performed until convergence of the training loss for up to 5000 iterations that took up to 20 min (on an NVIDIA Quadro RTX 4000).

Methods trained using the N2V technique were optimized by learning to predict the noisy low-dose reconstruction from a perturbed version of the same low-dose data. Here, 1 % of the voxels are replaced randomly by voxel intensities from its 5 × 5 × 5 voxel neighborhood and the modified data is fed through the model. Eventually, the MSE loss is calculated only at positions of replaced voxels between the prediction and the noisy low-dose reconstruction.

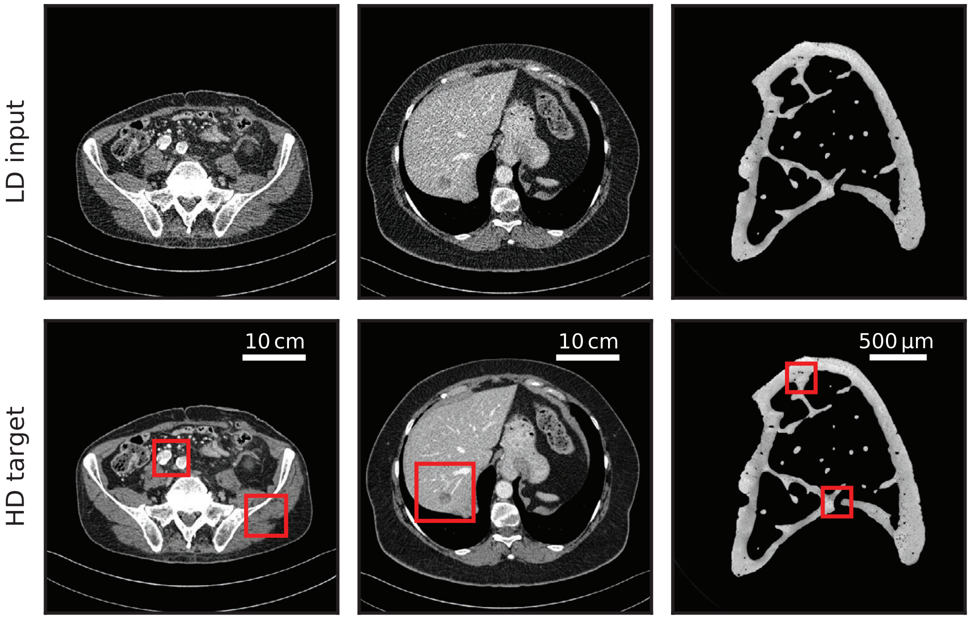

FIGURE 3 Example input and target slices from the investigated Grand Challenge data set (25 % dose, first and second columns) and the XRM bone data set (10 % dose, third column). Red squares highlight the ROIs for the visual comparison. The display windows are [−150, 250] HU and [0.25, 0.7] arb. unit for the respective data sets

3 | RESULTS

3.1 | Denoising results

We present qualitative denoising results on selected slices from both test data sets, visualized in Figure 3. Multiple magnified regions of interest (ROIs) allow comparing the denoising performance on small image features of the Grand Challenge data set for better visualization. Figure 4 displays ROIs of high-contrast anatomies in the abdomen area as well as a low-contrast liver lesion. The additionally provided difference images (prediction—target) particularly highlight artifacts in the model predictions. Closer studying local features reveals oversmoothed results in low-contrast regions for the QAE and the RED-CNN compared to the high-dose target, highlighted by orange circles and arrows. However, high-frequency details like edges are well preserved as there are few structures visible in the difference images. We find that predictions of bilateral filter-based pipelines, JBFnet, and WGAN visually appear closer to the target images through achieving a reasonable trade-off between noise removal and preserving high frequencies, compared to the other approaches. However, blurred edges are visible in the difference images for JBFnet, WGAN, and the self-supervised bilateral filter pipeline (3BFs N2V). The predictions of the supervised-trained bilateral filter pipeline (3BFs) and the JBFnet visually contain a noise pattern close to the target ROI, while stronger removing edges around high-contrast features compared to QAE and RED-CNN. Quantitatively, RED-CNN, 3BFs, and QAE outperform the other models on the Grand Challenge data set in terms of structural similarity index measure (SSIM) and peak signal-to-noise ratio (PSNR), as presented in Table 1. In addition, Table 2 presents the investigated similarity metrics for each model on the individual test patients of the Grand Challenge data set to provide insight on how different models perform on single patients.

The model predictions on the XRM bone data set are depicted in Figure 5, including the entire reconstruction pipeline, simultaneously denoising in sinogram and reconstruction domain (sin & reco BFs). Visually, sin & reco BFs, RED-CNN, and QAE outperform the other models in terms of noise removal and edge sensitivity. Predictions of the self-supervised bilateral filter pipeline preserve edges but still contain a substantial amount of noise. In contrast, the difference images of WGAN and JBFnet reveal more removed high-frequency details at edges, compared to all other methods. Quantitatively, denoising with three trained bilateral filters in the reconstruction domain performs slightly worse than QAE and RED-CNN in terms of SSIM and PSNR, as listed in Table 3. Different configurations of the end-to-end trainable reconstruction pipeline were investigated, varying the number of filter layers from one to three in the sinogram domain and from two to zero in the reconstruction domain. Moving two of the bilateral filter layers into the sinogram domain turns out to improve the denoising performance compared to purely image-based bilateral filtering pipelines to match the SSIM

TABLE 1 Quantitative denoising results of the compared pipelines on the test patients of the 2016 Low Dose CT Grand Challenge data set. For each model, the number of trainable parameters and the average inference time per 512 × 512 image slice (Quadro RTX 4000) are presented. The names of our proposed pipelines and the best-performing models are highlighted

	Grand challenge data set			**[s]**
	SSIM (mean ± std)	**PSNR (mean ± std)**	**# params**	**runtime**
Low-dose CT	0.8991 ± 0.033	38.64 ± 1.62	–	–
WGAN[14]	0.9341 ± 0.085	40.47 ± 0.86	7,800,000	**0.1**
RED-CNN[11]	**0.9680 ± 0.010**	**43.63 ± 1.19**	1,800,000	4.2
JBFnet[12]	**0.9666 ± 0.012**	42.77 ± 1.13	118,000	5.6
QAE[13]	0.9650 ± 0.010	**43.35 ± 1.18**	137,000	2.0
1 BF layer	0.9656 ± 0.013	42.73 ± 1.38	**4**	**0.2**
3 BF layers	**0.9674 ± 0.012**	**43.07 ± 1.42**	**12**	**0.5**
3 BFs (N2V[34])	0.9577 ± 0.015	41.15 ± 1.17	**12**	**0.5**

TABLE 2 Quantitative denoising results of each test patient from the 2016 Low Dose CT Grand Challenge data set. The names of our proposed pipelines are highlighted

Patient ID		**LD**	**WGAN**	**RED-CNN**	**JBFnet**
L310	SSIM	0.9209 ± 0.03	0.9329 ± 0.01	0.9718 ± 0.01	0.9728 ± 0.01
	PSNR	39.83 ± 1.8	40.59 ± 1.0	44.13 ± 1.4	43.29 ± 1.4
L333	SSIM	0.8841 ± 0.03	0.9340 ± 0.01	0.9647 ± 0.01	0.9604 ± 0.01
	PSNR	38.01 ± 1.4	40.38 ± 0.9	43.32 ± 1.1	42.37 ± 1.0
L506	SSIM	0.8945 ± 0.02	0.9354 ± 0.01	0.9680 ± 0.01	0.9674 ± 0.01
	PSNR	38.16 ± 0.9	40.45 ± 0.7	43.50 ± 0.8	42.71 ± 0.7
Patient ID		**QAE**	**1 BF**	**3 BFs**	**3 BFs (N2V)**
L310	SSIM	0.9679 ± 0.01	0.9722 ± 0.01	0.9732 ± 0.01	0.9658 ± 0.02
	PSNR	43.79 ± 1.4	43.56 ± 1.6	43.87 ± 1.7	41.53 ± 1.5
L333	SSIM	0.9624 ± 0.01	0.9600 ± 0.01	0.9625 ± 0.01	0.9507 ± 0.01
	PSNR	43.07 ± 1.1	42.26 ± 1.2	42.64 ± 1.3	40.90 ± 1.1
L506	SSIM	0.9652 ± 0.01	0.9654 ± 0.01	0.9672 ± 0.01	0.9575 ± 0.01
	PSNR	43.22 ± 0.8	42.42 ± 0.8	42.75 ± 0.9	41.03 ± 0.6

TABLE 3 Quantitative denoising results of the compared pipelines on the test patients of the XRM bone data set. For each model, the number of trainable parameters is shown in the right column

	XRM Bone data set		
	SSIM (mean ± std)	**PSNR (mean ± std)**	**# parameters**
Low-dose CT	0.1652 ± 0.0084	19.95 ± 0.27	–
WGAN[14]	0.5170 ± 0.0135	29.45 ± 0.23	7,800,000
RED-CNN[11]	**0.7122 ± 0.0087**	**33.09 ± 0.20**	1,800,000
JBFnet[12]	0.6564 ± 0.0139	30.98 ± 0.31	118,000
QAE[13]	**0.7126 ± 0.0088**	**33.09 ± 0.20**	137,000
1 BF layer	0.6599 ± 0.0167	31.40 ± 0.30	**4**
3 BF layers	0.6698 ± 0.0158	32.19 ± 0.26	**12**
1 sin & 2 reco BFs	0.7053 ± 0.0120	33.10 ± 0.22	**12**
2 sin & 1 reco BFs	**0.7094 ± 0.0118**	**33.17 ± 0.22**	**12**
3 sin & 0 reco BFs	0.7019 ± 0.0123	32.82 ± 0.23	**12**
3 BF layers (N2V[34])	0.4590 ± 0.0280	27.61 ± 0.57	**12**

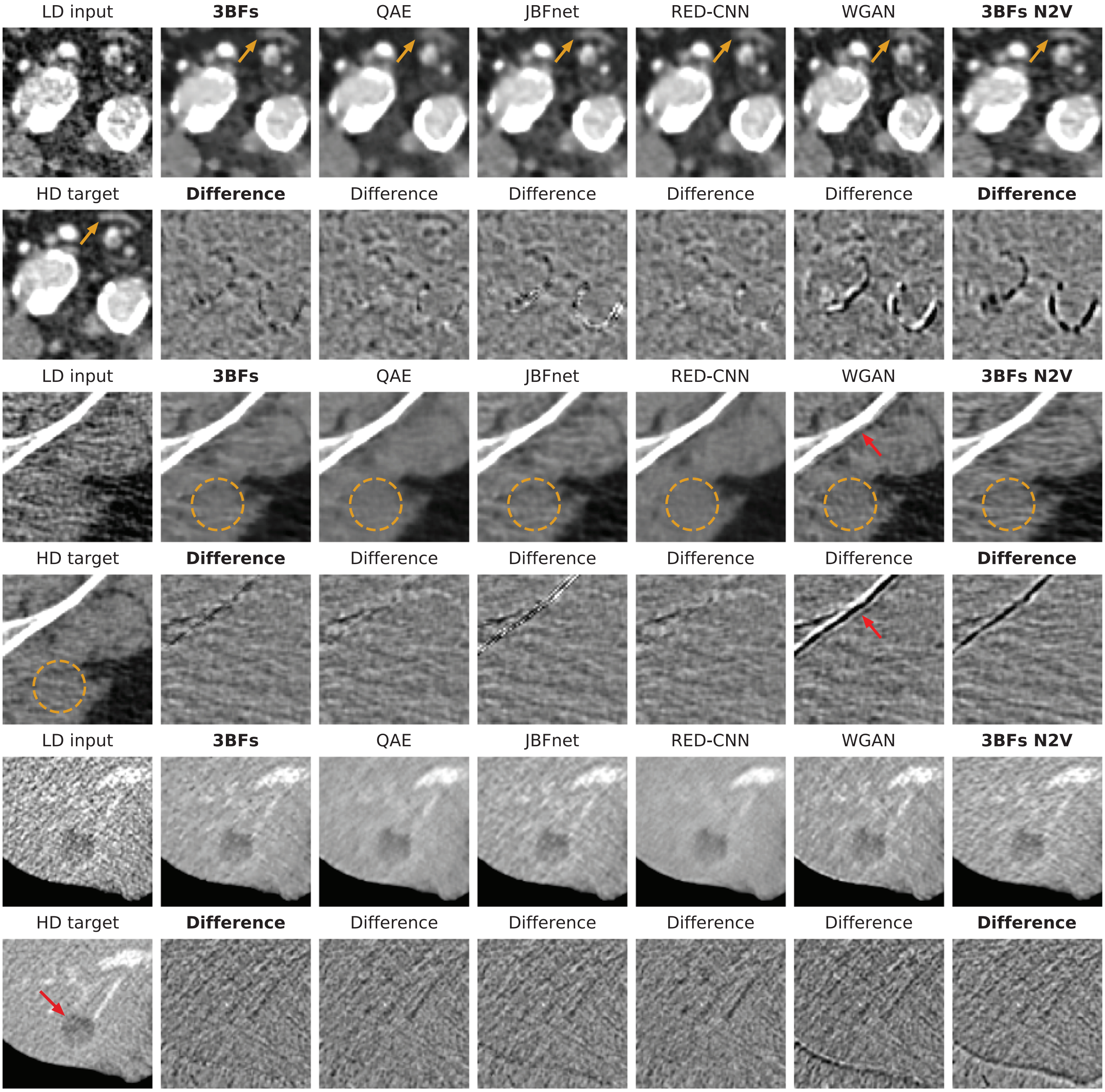

FIGURE 4 The first and third rows show denoising predictions from the 2016 Low Dose CT Grand Challenge data set in the ROIs illustrated in Figure 3 for all employed methods. Our proposed pipelines are highlighted in bold letters. The display window is [−150, 250] HU. The red arrow in the sixth row highlights a pathologically relevant liver lesion. The second, fourth, and sixth rows present difference images between prediction and target where prediction artifacts like blurred edges become particularly visible. Difference images are plotted in the window [−150, 150] HU

and PSNR of the best-performing deep models on the test data. All trainable bilateral filter-based pipelines use several orders of magnitude fewer parameters compared to the deep reference architectures. Simultaneously, competitive denoising performance is achieved, outperforming multiple deep reference methods quantitatively in terms of the investigated metrics and qualitatively.

Closer studying the quantitative results reveals that the best-performing methods achieve similar performances. Therefore, we conducted Wilcoxon signed-rank tests between our best methods, namely, 3 BFs and 2 sin & 1 reco BFs, and all deep reference models to investigate the significance of performance differences. Hence, we found all differences on both data sets to be significant on a *p*-value of 0.01. Note that the shown standard deviation of the different methods is therefore rather meaningful in terms of the content variation within the testing data instead of the uncertainty of individual methods.

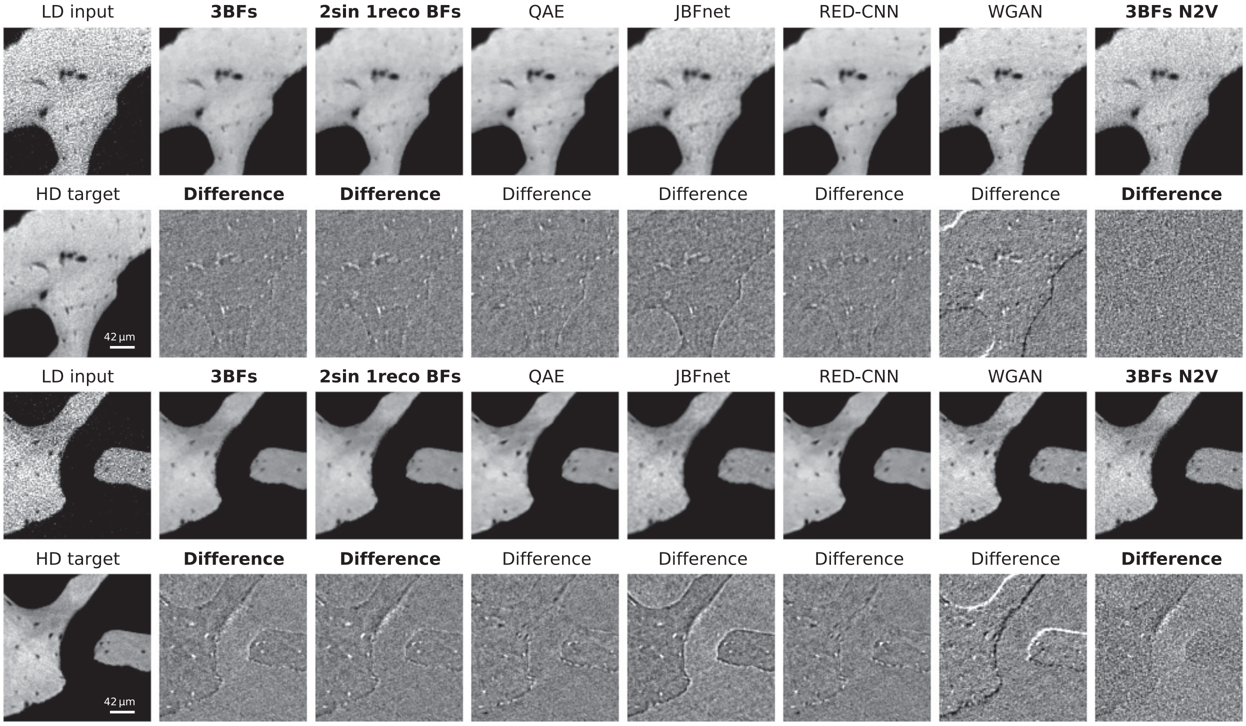

FIGURE 5 The first and third rows show denoising predictions from the XRM bone data set in the ROIs illustrated in Figure 3 for all employed methods. Our proposed pipelines are highlighted in bold letters. The display window is [0.25, 0.7] arb. unit. The second and fourth rows present difference images between prediction and target where prediction artifacts like blurred edges become particularly visible. Difference images are plotted in the window [−0.1, 0.1] arb. unit

3.2 | Depth of the bilateral filter pipeline

The benefit of iteratively removing noise with subsequent bilateral filters is investigated by training bilateral filter pipelines of different depths. Pipelines containing up to six filter layers are trained until convergence and tested on the Grand Challenge data set. Mean SSIM and PSNR suggest that the denoising performance indeed benefits from stacking three filter layers, as shown in Figure 6. Adding more filters slightly reduces the performance as more parameters have to properly converge to relatively small values to preserve edges. The relatively simple shape of the bilateral filter kernel is not able to extract complex features from images, but rather iteratively removes noise from the data in subsequent layers. Therefore, later denoising layers might remove more image content compared to noise, which can result in slightly degraded image quality for more than four stacked filters.

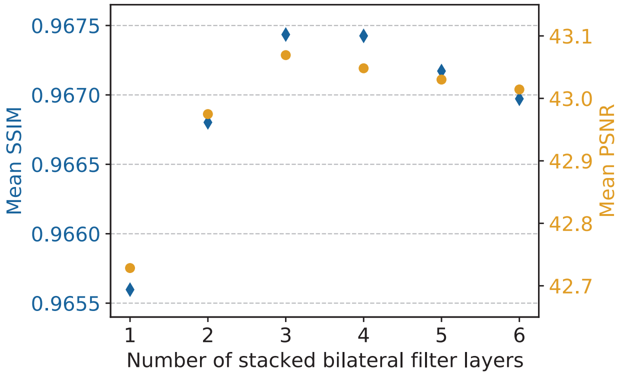

FIGURE 6 The denoising performance of multiple stacked bilateral filter layers is compared based on the average SSIM and PSNR on the test patients of the 2016 Low Dose CT Grand Challenge data set. Denoising with two or more trained layers is advantageous compared to only using a single layer

4 | Discussion

The proposed pipelines quantitatively and qualitatively show comparable denoising to state-of-the-art deep architectures. Although information from a single CT projection is spread through the whole volume, the appearance of noise in the reconstructed volume is a local phenomenon. Smart filtering within a finite neighborhood can therefore perform surprisingly well, as is the case for the bilateral filter. However, especially the choice of optimal intensity range parameters σ_r is very crucial for the denoising performance. If σ_r is chosen too large, regions of constant attenuation are well restored, but edges are blurred concurrently, leading to degraded predictions. Optimizing the hyperparameters in a purely data-driven way can therefore leverage the applicability of the bilateral filter, as well as guarantee a near-optimal choice of parameters, as we demonstrated by achieving quantitative scores comparable to state-of-the-art methods.

We empirically found that the denoising performances of optimized pipelines are independent of the parameter initialization. For pipelines containing more than a single filter layer, the filter parameters first converge to very similar values between the layers until the loss is almost minimized. During further training, individual filter parameters start to converge to distinct values, slowly reducing the training loss further. During this fine-tuning phase, stacked bilateral filters learn to focus on different features, for example, preserving edges or smoothing planes.

Compared to the ground truth targets, the predictions of RED-CNN and QAE visually appear oversmoothed in low-contrast regions, which can remove features being beneficial for the physician, as visible around the liver lesion in the fifth row of Figure 4. Simultaneously, the deep models can remove artifacts from CT reconstructions, like streaks created through beam hardening. Such an example is highlighted with orange circles in Figure 4 and demonstrates the great flexibility of deep convolutional models. Their ability to extract complex image features, however, comes at the cost of interpretability and reliability, which is usually desired in medical applications.

Although supervised training of the bilateral filter-based pipelines performs comparable to deep reference models on both investigated data sets, the self-supervised approach using the Noise2Void technique does not fully converge to the optimal parameter set. In general, improved image quality can be recognized; however, the presented difference images identify blurred edges due to insufficiently learned filter parameters. If noise is not fully distinguishable from content, the intensity range parameters σ_r do not properly converge to the optimum values. In their work, Krull et al.[34] already identified this limitation of their Noise2Void algorithm for images with inherently high irregularities or noise correlation between voxels, as it is the case for CT reconstructions.

Approximation and estimation error are commonly combined to estimate the risk bound of neural networks. The two contributions refer first to the distance between target function and closest neural network function and second to the distance between the estimated and the ideal network function.[51] Our quantitative and visual results show that deep neural networks and trainable bilateral filter pipelines can both well predict denoised reconstructions after extensive training, and therefore, show comparable approximation risk. However, we think that trainable bilateral filter layers have a considerably smaller estimation error due to the restricted influence of the few trained parameters. The decreased error bound can be explained by treating the filter algorithm similar to a known operator.[35,36] In contrast, deep neural network-based approaches can learn network functions that only work well within the training data distribution but fail to produce reliable results for other samples from the CT data distribution, which indicates a larger estimation error. Mathematically proving a lower estimation error of bilateral filter-based pipelines is, however, difficult as to the best of our knowledge, the absolute error bound of a general convolutional neural network has not yet been derived explicitly.

In a clinical environment, data integrity at any point in the medical data processing pipeline is crucial for a reliable diagnosis from the physician. During training, deep models are optimized on a finite number of training samples to extract features generalizing over the entire distribution of data. However, in a clinical environment, data can be acquired, which are not properly represented by the finite training data distribution. Image processing algorithms must be able to handle such samples to avoid failing downstream tasks, including the diagnosis by the physician. Figure 7 exhibits failing network predictions of one deep reference method on out-of-distribution CT slices around medical implants in abdomen scans of the *Low Dose CT Image and Projection Data*.[52] The reference method was before trained on the 2016 Low Dose CT Grand Challenge training data set until convergence and achieves state-of-the-art performance on the test data, as presented in Table 1. A different predicted artifact of a deep reference method is highlighted in Figure 4 with red arrows. Here, a shadow-like structure right next to the bone region is introduced, which is also visible in the difference image. By algorithmic design, trainable bilateral filter layers only filter in a finite neighborhood with well-defined Gaussian kernels and cannot produce such artifacts in their predictions. The inherent lack of complex feature abstraction in the proposed filter layer enforces proximity between prediction and measured ground truth and is advantageous for maintaining reliable data compared to the nontransparent data processing performed in deep state-of-the-art denoising models. Additionally, the low amount of required training data makes trainable bilateral filter layers convenient to use and easy to integrate into complex data processing pipelines.

Future work should aim to extend the end-to-end trainable reconstruction pipeline to clinical CT data. This, however, requires developing a differentiable reconstruction operator for helical CT trajectories to backpropagate a loss into sinogram domain. Experiments on such a pipeline would be particularly interesting aiming for reducing photon starvation artifacts that are visible in the abdomen scans around high-absorbing bones. Learning a bilateral denoising filter prior to log transformation of the projections could therefore help leveraging the image quality of low-dose acquisitions. Further, clinical studies are required comparing the denoising performance of trainable bilateral filter-based pipelines

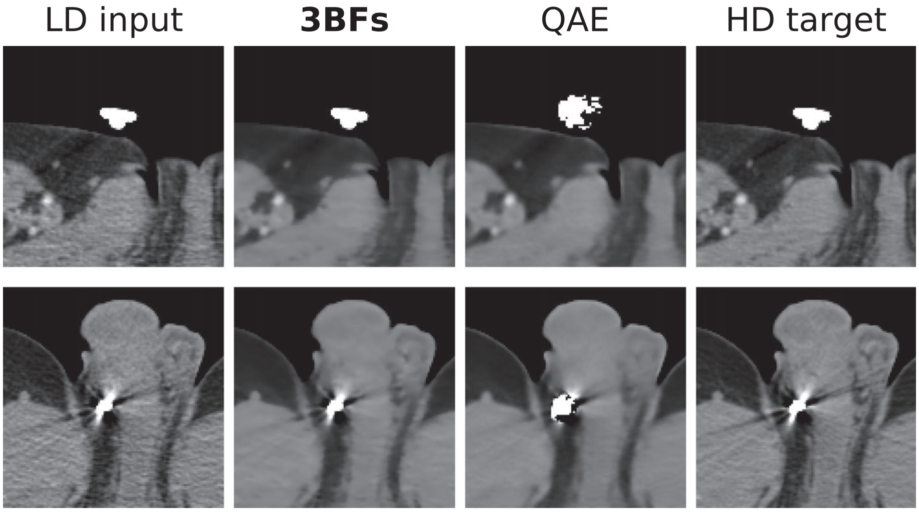

FIGURE 7 Example of failing network predictions of the trained deep QAE model on strongly absorbing objects in the scan of patient *L006* and *L193* from the *Low Dose CT Image and Projection Data*.[52] The display window is [−150, 250] HU

with other methods by radiologists and to investigate the true impact of the layers in a clinical CT workflow.

5 | CONCLUSIONS

We present a trainable, fully differentiable bilateral filter layer that can be directly incorporated in any deep architecture using the *PyTorch* framework with GPU acceleration. We show that three trainable bilateral filter layers with only four parameters each can achieve state-of-the-art performance of deep neural networks with hundred of thousands of trainable parameters on the low-dose CT denoising task, while performing comprehensible data processing and producing physically reliable predictions due to the few, simple employed filter kernels. The low number of parameters allows training with little data, self-supervised training using the Noise2Void scheme, and incorporating the filter layer at multiple locations in a CT reconstruction pipeline to further leverage denoising performance. In summary, trainable bilateral filters allow extensive dose reduction while maintaining high image quality and data integrity.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (ERC Grant No. 810316).

Open access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST

The authors have no relevant conflicts of interest to disclose.

REFERENCES

1. Boone JM, Hendee WR, McNitt-Gray MF, Seltzer SE. Radiation exposure from CT scans: how to close our knowledge gaps, monitor and safeguard exposure-proceedings and recommendations of the Radiation Dose Summit, sponsored by NIBIB, February 24–25, 2011. *Radiology*. 2012;265:544-554.
2. Barrett HH, Gordon S, Hershel R. Statistical limitations in transaxial tomography. *Comput Biol Med*. 1976;6:307-323.
3. Maier A, Fahrig R. GPU denoising for computed tomography. In: *Graphics Processing Unit-Based High Performance Computing in Radiation Therapy*. CRC Press; 2015;113.
4. Kelm ZS, Blezek D, Bartholmai B, Erickson BJ. Optimizing non-local means for denoising low dose CT. In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE; 2009:662-665.
5. Maier A, Wigström L, Hofmann HG, et al. Three-dimensional anisotropic adaptive filtering of projection data for noise reduction in cone beam CT. *Med Phys*. 2011;38:5896-5909.
6. Manduca A, Yu L, Trzasko JD, et al. Projection space denoising with bilateral filtering and CT noise modeling for dose reduction in CT. *Med Phys*. 2009;36:4911-4919.
7. Manhart M, Fahrig R, Hornegger J, Doerfler A, Maier A. Guided noise reduction for spectral CT with energy-selective photon counting detectors. In: *Proceedings of the Third CT Meeting*. 2014:91-94.
8. Han X, Bian J, Eaker DR, et al. Algorithm-enabled low-dose micro-CT imaging. *IEEE Trans Med Imaging*. 2010;30:606-620.
9. Beister M, Kolditz D, Kalender WA. Iterative reconstruction methods in X-ray CT. *Physica Med*. 2012;28:94-108.
10. Gilbert P. Iterative methods for the three-dimensional reconstruction of an object from projections. *J Theor Biol*. 1972;36:105-117.
11. Chen H, Zhang Y, Kalra MK, et al. Low-dose CT with a residual encoder-decoder convolutional neural network. *IEEE Trans Med Imaging*. 2017;36:2524-2535.
12. Patwari M, Gutjahr R, Raupach R, Maier A. JBFnet - low dose CT denoising by trainable joint bilateral filtering. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention—MICCAI 2020*. Springer; 2020:506-515.
13. Fan F, Shan H, Kalra MK, et al. Quadratic autoencoder (Q-AE) for low-dose CT denoising. *IEEE Trans Med Imaging*. 2019;39:2035-2050.
14. Yang Q, Yan P, Zhang Y, et al. Low-dose CT image denoising using a generative adversarial network with Wasserstein distance and perceptual loss. *IEEE Trans Med Imaging*. 2018;37:1348-1357.

15. Kang E, Chang W, Yoo J, Ye JC. Deep convolutional framelet denoising for low-dose CT via wavelet residual network. *IEEE Trans Med Imaging*. 2018;37:1358-1369.
16. Kang E, Min J, Ye JC. A deep convolutional neural network using directional wavelets for low-dose X-ray CT reconstruction. *Med Phys*. 2017;44:e360-e375.
17. Ketcha MD, Marrama M, Souza A, et al. Sinogram + image domain neural network approach for metal artifact reduction in low-dose cone-beam computed tomography. *J Med Imaging*. 2021;8:1-16.
18. Yuan X, He P, Zhu Q, Li X. Adversarial examples: attacks and defenses for deep learning. *IEEE Trans Neural Netw Learn Syst*. 2019;30:2805-2824.
19. Wu P, Sisniega A, Uneri A, et al. Using uncertainty in deep learning reconstruction for cone-beam CT of the brain. arXiv preprint arXiv:2108.09229. 2021.
20. Zhang C, Li Y, Chen G-H. Deep learning in image reconstruction: vulnerability under adversarial attacks and potential defense strategies. In: *Medical Imaging 2021: Physics of Medical Imaging*, Vol. 11595, International Society for Optics and Photonics; 2021:115951U.
21. Antun V, Renna F, Poon C, Adcock B, Hansen AC. On instabilities of deep learning in image reconstruction and the potential costs of AI. *Proc Natl Acad Sci USA*. 2020;117:30088-30095.
22. Huang Y, Würfl T, Breininger K, Liu L, Lauritsch G, Maier A. Some investigations on robustness of deep learning in limited angle tomography. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention—MICCAI 2018*. Springer; 2018:145-153.
23. Tomasi C, Manduchi R. Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision*. IEEE; 1998:839-846.
24. Chen Y, Shu Y. Optimization of bilateral filter parameters via chi-square unbiased risk estimate. *IEEE Signal Process Lett*. 2013;21:97-100.
25. Anoop V, Bipin PR. Medical image enhancement by a bilateral filter using optimization technique. *J Med Syst*. 2019;43:1-12.
26. Kishan H, Seelamantula CS. Optimal parameter selection for bilateral filters using Poisson Unbiased Risk Estimate. In: *2012 19th IEEE International Conference on Image Processing*. IEEE; 2012:121-124.
27. Dai T, Zhang Y, Dong L, Li L, Liu X, Xia S. Content-aware bilateral filtering. In: *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*. IEEE; 2018:1-6.
28. Peng H, Rao R. Bilateral kernel parameter optimization by risk minimization. In: *2010 IEEE International Conference on Image Processing*. IEEE; 2010:3293-3296.
29. Jampani V, Kiefel M, Gehler PV. Learning sparse high dimensional filters: image filtering, dense crfs and bilateral neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016:4452-4461.
30. Kang W, Patwari M. Low Dose Helical CBCT denoising by using domain filtering with deep reinforcement learning. arXiv preprint arXiv:2104.00889. 2021.
31. Patwari M, Gutjahr R, Raupach R, Maier A. Limited parameter denoising for low-dose X-ray computed tomography using deep reinforcement learning. *Med Phys*. 2022.
32. Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: *Proceedings of NeurIPS*. Curran Associates, Inc.; 2019:8024-8035.
33. Syben C, Michen M, Stimpel B, Seitz S, Ploner S, Maier AK. PYRO-NN: python reconstruction operators in neural networks. *Med Phys*. 2019;46:5110-5115.
34. Krull A, Buchholz T-O, Jug F. Noise2void-learning denoising from single noisy images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019:2129-2137.
35. Maier A, Syben C, Stimpel B, et al. Learning with known operators reduces maximum error bounds. *Nat Machine Intell*. 2019;1:373-380.
36. Maier A, Schebesch F, Syben C, et al. Precision learning: towards use of known operators in neural networks. In: *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE; 2018:183-188.
37. McCollough CH, Bartley AC, Carter RE, et al. Low-dose CT for the detection and classification of metastatic liver lesions: results of the 2016 low dose CT grand challenge. *Med Phys*. 2017;44:e339-e352.
38. Jakob W. pybind11 – Seamless operability between C++11 and Python. 2021.
39. Thies M. Differentiable reconstruction for x-ray microscopy data. 2021. https://doi.org/10.24433/CO.2740182.v2
40. Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *Josa a*. 1984;1:612-619.
41. Hannah KM, Thomas CD, Clement JG, De Carlo F, Peele AG. Bimodal distribution of osteocyte lacunar size in the human femoral cortex as revealed by micro-CT. *Bone*. 2010;47:866-871.
42. Grüneboom A, Hawwari I, Weidner D, et al. A network of transcortical capillaries as mainstay for blood circulation in long bones. *Nat Metabol*. 2019;1:236-250.
43. Grüneboom A, Kling L, Christiansen S, et al. Next-generation imaging of the skeletal system and its blood supply. *Nat Rev Rheumatol*. 2019;15:533-549.
44. Gruber R, Pietschmann P, Peterlik M. Introduction to bone development, remodelling and repair. In: Grampp S, ed. *Radiology of Osteoporosis*. Springer; 2008:1-23.
45. Buenzli PR, Sims NA. Quantifying the osteocyte network in the human skeleton. *Bone*. 2015;75:144-150.
46. Wagner F, Thies M, Karolczak M, et al. Monte Carlo dose simulation for in-vivo X-ray nanoscopy. In: *Bildverarbeitung für die Medizin 2022*. Springer; 2022:107-112.
47. Mill L, Kling L, Grüneboom A, Schett G, Christiansen S, Maier A. Towards in-vivo X-ray nanoscopy. In: *Bildverarbeitung für die Medizin 2019*. Springer; 2019:251-256.
48. Huang Y, Mill L, Stoll R, et al. Semi-permeable filters for interior region of interest dose reduction in X-ray microscopy. In: *Bildverarbeitung für die Medizin*. Springer; 2021:61-66.
49. Aust O, Thies M, Weidner D, et al. Tibia cortical bone segmentation in micro-CT and X-ray microscopy data using a single neural network. In: *Bildverarbeitung für die Medizin 2022*. Springer; 2022:333-338.
50. Yu L, Shiung M, Jondal D, McCollough CH. Development and validation of a practical lower-dose-simulation tool for optimizing computed tomography scan protocols. *J Comput Assist Tomogr*. 2012;36:477-487.
51. Barron AR. Approximation and estimation bounds for artificial neural networks. *Mach Learn*. 1994;14:115-133.
52. Moen TR, Chen B, Holmes III DR, et al. Low-dose CT image and projection dataset. *Med Phys*. 2021;48:902-911.

How to cite this article: Wagner F, Thies M, Gu M, et al. Ultralow-parameter denoising: Trainable bilateral filter layers in computed tomography. *Med Phys*. 2022;49:5107–5120. https://doi.org/10.1002/mp.15718

APPENDIX I: ANALYTICAL DERIVATIVE OF THE BILATERAL FILTER

We aim to make the spatial and intensity range hyperparameters σ_s and σ_r of the bilateral filter trainable in a gradient descent algorithm. In order to update the parameters during optimization, the derivative of the loss function L with respect to each parameter σ_γ is required

$$\frac{\partial L}{\partial \sigma_\gamma} = \frac{\partial L}{\partial \mathbf{Y}}\frac{\partial \mathbf{Y}}{\partial \sigma_\gamma} = \sum_k \frac{\partial L}{\partial \hat{Y}_k}\frac{\partial \hat{Y}_k}{\partial \sigma_\gamma}. \tag{A.1}$$

Following the definition of the bilateral filter in (1), the derivative $\frac{\partial \hat{Y}_k}{\partial \sigma_\gamma}$ yields

$$\frac{\partial \hat{Y}_k}{\partial \sigma_\gamma} = -w_k^{-2}\alpha_k \frac{\partial w_k}{\partial \sigma_\gamma} + w_k^{-1}\frac{\partial \alpha_k}{\partial \sigma_\gamma} \tag{A.2}$$

with

$$\frac{\partial w_k}{\partial \sigma_\gamma} = \sum_{n\in\mathcal{N}} \frac{\partial}{\partial \sigma_\gamma} G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_n})G_{\sigma_r}(X_k - X_n), \tag{A.3}$$

$$\frac{\partial \alpha_k}{\partial \sigma_\gamma} = \sum_{n\in\mathcal{N}} X_n \frac{\partial}{\partial \sigma_\gamma} G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_n})G_{\sigma_r}(X_k - X_n), \tag{A.4}$$

and the derivative of the Gaussian kernel

$$\frac{\partial}{\partial \sigma_\gamma} G_\sigma(c) = G_\sigma(c)\frac{c^2}{\sigma_\gamma^3}. \tag{A.5}$$

Additionally, the gradient with respect to every voxel of the noisy input volume X_i needs to be derived, as it should be possible to propagate a loss into previous filter layers during backpropagation

$$\frac{\partial L}{\partial X_i} = \frac{\partial L}{\partial \mathbf{Y}}\frac{\partial \mathbf{Y}}{\partial X_i} = \sum_k \frac{\partial L}{\partial \hat{Y}_k}\frac{\partial \hat{Y}_k}{\partial X_i}. \tag{A.6}$$

The derivative of the output voxel $\hat{Y}_k$ with respect to the input voxel X_i yields

$$\frac{\partial \hat{Y}_k}{\partial X_i} = -w_k^{-2}\alpha_k \frac{\partial w_k}{\partial X_i} + w_k^{-1}\frac{\partial \alpha_k}{\partial X_i} \tag{A.7}$$

by again using the definition of the bilateral filter from (1) and applying the product rule analog to (A.2).

The filtered output voxel $\hat{Y}_k$ is dependent on both input voxels X_k and X_n. To carry out the derivative with respect to X_i the two cases $k \neq i$ and $k = i$ are distinguished.

Case 1: ($k \neq i$)

$$\left.\frac{\partial w_k}{\partial X_i}\right|_{k\neq i} = \sum_{n\in\mathcal{N}} G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_n})\frac{\partial}{\partial X_i}G_{\sigma_r}(X_k - X_n)$$
$$= G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_i})G_{\sigma_r}(X_k - X_i)\frac{X_k - X_i}{\sigma_r^2} \tag{A.8}$$

$$\left.\frac{\partial \alpha_k}{\partial X_i}\right|_{k\neq i} = \sum_{n\in\mathcal{N}} G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_n})\frac{\partial}{\partial X_i}G_{\sigma_r}(X_k - X_n)X_n$$
$$= G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_i})\left[\left(\frac{\partial}{\partial X_i}G_{\sigma_r}(X_k - X_i)\right)X_i + G_{\sigma_r}(X_k - X_i)\left(\frac{\partial}{\partial X_i}X_i\right)\right]$$
$$= G_{\sigma_s}(\mathbf{p_k}-\mathbf{p_i})\cdot G_{\sigma_r}(X_k - X_i)\left[\frac{X_k - X_i}{\sigma_r^2}X_i + 1\right]. \tag{A.9}$$

In both expressions, only the ith term of the sum ($n = i$) contributes.

Case 2: ($k = i$)

$$\left.\frac{\partial w_k}{\partial X_i}\right|_{k=i} = \frac{\partial}{\partial X_i}\sum_{n\in\mathcal{N}} G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_n})G_{\sigma_r}(X_i - X_n)$$
$$= \frac{\partial}{\partial X_i}\left[\underbrace{G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_i})G_{\sigma_r}(X_i - X_i)}_{=1} + \sum_{n\in\mathcal{N},n\neq i} G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_n})G_{\sigma_r}(X_i - X_n)\right]$$
$$= \sum_{n\in\mathcal{N},n\neq i} G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_n})\frac{\partial}{\partial X_i}G_{\sigma_r}(X_i - X_n)$$
$$= \sum_{n\in\mathcal{N}} G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_n})G_{\sigma_r}(X_i - X_n)\frac{X_n - X_i}{\sigma_r^2}, \tag{A.10}$$

$$\left.\frac{\partial \alpha_k}{\partial X_i}\right|_{k=i} = \frac{\partial}{\partial X_i}\sum_{n\in\mathcal{N}} G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_n})G_{\sigma_r}(X_i - X_n)X_n$$
$$= \frac{\partial}{\partial X_i}\left[\underbrace{G_{\sigma_s}(\mathbf{p_i}-\mathbf{p_i})G_{\sigma_r}(X_i - X_i)}_{=1}X_i\right.$$

$$
\left. + \sum_{n\in\mathcal{N}, n\neq i} G_{\sigma_s}(\mathbf{p_i} - \mathbf{p_n}) G_{\sigma_r}(X_i - X_n) X_n \right]
$$

$$
= 1 + \sum_{n\in\mathcal{N}, n\neq i} \left[G_{\sigma_s}(\mathbf{p_i} - \mathbf{p_n}) \cdot G_{\sigma_r}(X_i - X_n) X_n \frac{X_n - X_i}{\sigma_r^2} \right]
$$

$$
= 1 + \sum_{n\in\mathcal{N}} \left[G_{\sigma_s}(\mathbf{p_i} - \mathbf{p_n}) \cdot G_{\sigma_r}(X_i - X_n) X_n \frac{X_n - X_i}{\sigma_r^2} \right]. \tag{A.11}
$$

Note that in the last steps, the sum can be carried out over the entire neighborhood N as the contribution of the term $n = i$ is zero, respectively.

APPENDIX II: Implementation

In practice, (A.6) describes a convolution of the kernel $\frac{\partial \hat{Y}_k}{\partial X_i}$ with the backpropagated loss $\frac{\partial L}{\partial \hat{Y}_k}$ in a neighborhood denoted by k that is given by the finite kernel size—analog to a conventional convolutional layer. However, due to the dependency of the kernel on two voxels of the input volume X_k and X_i, the analytical derivative of the convolutional kernel is more elaborate. When calculating the sum in (A.6), both (A.8) and (A.9) are used for the contributions $k \neq i$ and only the term $k = i$ is derived in (A.10) and (A.11). Note that for faster computation, the sums in (A.10) and (A.11) can be precalculated in the forward pass of the filtering. The finite spatial kernel is calculated by taking into account voxels covering five times the spatial standard deviation in each spatial dimension, but at least $5 \times 5 \times 5$ voxels. This could affect the performance for large spatial kernels with $\sigma_S \gg 1$, but was never a relevant factor in any of our experiments.

To validate the correct implementation of the filter derivative, we compared the analytical gradient, provided by the backward pass of the filter layer, with a numerical approximation of the gradient, computed via small finite differences. Our public repository contains a gradient check script using the *torch.autograd.gradcheck* function from the *PyTorch* framework.