SDF: A Framework for Self-supervision

Emilien Valat Andreas Hauptmann Ozan Öktem December 2, 2024

Abstract

Reconstructing images using Computed Tomography (CT) in an industrial context leads to specific challenges that differ from those encountered in other areas, such as clinical CT. Indeed, non-destructive testing with industrial CT will often involve scanning multiple similar objects while maintaining high throughput, requiring short scanning times, which is not a relevant concern in clinical CT. Under-sampling the tomographic data (sinograms) is a natural way to reduce the scanning time at the cost of image quality since the latter depends on the number of measurements. In such a scenario, post-processing techniques are required to compensate for the image artifacts induced by the sinogram sparsity.

We introduce the Self-supervised Denoiser Framework (SDF), a self-supervised training method that leverages pre-training on highly sampled sinogram data to enhance the quality of images reconstructed from undersampled sinogram data. The main contribution of SDF is that it proposes to train an image denoiser in the sinogram space by setting the learning task as the prediction of one sinogram subset from another. As such, it does not require ground-truth image data, leverages the abundant data modality in CT, the sinogram, and can drastically enhance the quality of images reconstructed from a fraction of the measurements.

We demonstrate that SDF produces better image quality, in terms of peak signal-to-noise ratio (PSNR), than other analytical and self-supervised frameworks in both 2D fan-beam or 3D cone-beam CT (CBCT) settings. Moreover, we show that the enhancement provided by SDF carries over when fine-tuning the image denoiser on a few examples, making it a suitable pre-training technique in a context where there is little high-quality image data. Our results are established on experimental datasets, making SDF a strong candidate for being the building block of foundational image-enhancement models in CT.

1 Introduction

X-ray Computed Tomography (CT) is an imaging technique that computes volumetric images from a set of indirect observations, the sinogram, obtained by sampling an object with an X-ray beam under different viewing angles. CT is used routinely in various situations, such as medical imaging and non-destructive

testing for industrial applications. The latter setting involves scanning objects while maintaining high throughput, highlighting the need for fast sampling and reconstruction routines. A natural way to limit sampling time is to reduce the number of measurements of each object at the expense of the reconstructed image's quality. Advanced computational techniques are required to accommodate sparse measurements during the reconstruction process or enhance the images in a post-processing step.

Tomographic image reconstruction is an inverse problem. That is, given observations $y \in Y$, recovering the causal factor $x \in X$ that produced them. For X-ray CT, the set of observations is called the sinogram and is obtained by sampling an object from different viewing angles. The $data\ model$ is a formalised computational model for generating the sinogram from an image. It can be defined as:

$$y = Ax + \epsilon \tag{1}$$

where $A: X \to Y$ is the forward operator that models the interaction between the X-rays and the object without observation noise. The term ϵ represents the latter and is inherent to the sampling process. Although the radiative transport equation accurately models the interaction, it is common in CT to consider a simplified model that ignores beam hardening (monochromatic X-rays) and scattering. Under these simplifications and assuming data has undergone suitable pre-processing, one can set the forward operator $\mathcal A$ to the ray transform.

In tomography, reconstructing an image is equivalent to solving (1). As such, when A is the ray transform, this amounts to its inversion, which is an ill-posed process due to the noise introduced during sampling. Regularisation is then needed to mitigate the solution's intrinsic instability and the noise amplification that comes from the ill-posedness of the inversion. These regularisation methods can be analytical, iterative, variational, or data-driven [1]. The data-driven approach methods outperform others when sinogram data is missing due to undersampling, is highly noisy, or both, as shown experimentally in [2, 3]. Among these, the best-performing methods are often based on domain-adapted neural networks, meaning that they rely on reconstruction operators $\mathcal{R}_{\theta}: Y \to X$ given by deep neural networks (NNs) with an architecture that combines a handcrafted physics-based model about the forward operator and a learnable block. One other advantage of the data-driven reconstruction networks is that they can be interfaced with preprocessing and post-processing operators for tasks such as image classification [4], segmentation [5] and registration [6]. However, this integration with other networks can be problematic when the reconstruction operator is trained in a supervised manner. In the most classical case, where a neural network is trained to map a noisy or undersampled sinogram to a high-quality image, the training requires abundant target data, leading to the train networks tending to enforce features in the inferred image like the ones present in the target images they learned from. This, in turn, hinders the interfacing with post-processing operators, for these features might not be adapted inputs to the post-processing operators. As such, there is an interest in considering other learning protocols that do not involve groundtruth images during training. Although such approaches exist [7, 8], they are not designed for reconstructing images for under-sampled sinogram data. This paper contributes to this active field of research by providing a self-supervised framework that only relies on sinogram data for training denoisers to enhance images reconstructed from undersampled data.

2 Related Work

Self-supervised Denoiser Framework (SDF) is a learning technique that leverages the abundant data modality in X-ray CT, the sinogram, for pretraining an image denoiser. As such, it relates closely to the self-supervised framework Noise2Inverse [7].

The underlying idea is to create input-target image pairs by reconstructing them from different sinogram subsets and train a denoiser NN in a self-supervised way. NN's training goal is to infer an image reconstructed from one subset from an image reconstructed from a different one. Despite its good performance in removing measurement noise, the Noise2Inverse framework "does not remove artifacts resulting from under-sampling" according to its authors. We surmise that the non-local nature of CT images prevents an image denoiser trained on images exhibiting under-sampling artifacts from removing them. This supposition is illustrated by the difference in performance between training strategy 1: X and X: 1, highlighting the importance of artifact-free input images. Building on the Noise2Inverse framework, the Proj2Proj framework [9] investigates the use of projection data corruption and masking to train an image denoiser in a self-supervised fashion in the low-dose CT case, but is not designed to enhance images reconstructed from sparse measurements.

Self-supervision is also used successfully by [8]. They use a loss in the sinogram space to train an image denoiser, producing high-fidelity images for CBCT and helical trajectory. Their approach relies on reconstructing a high-quality image from all but 12 projections and resampling the image at the locations of the 12 ones left aside to minimise a loss in the sinogram space. As such, they compare "simulated" projections to the ones sampled initially. Although this approach is sound for medical CT, as undersampling is not used in this field, they do not evaluate their method on undersampled data. Moreover, they use a substantial image denoiser: for 3D cone-beam CT (CBCT), their network totals 1.73M parameters (988k for the sinogram network and 742k for the volume network), while we successfully use a 10k parameters network, removing the uncertainty about the need for large neural networks in our approach. We have also found their results hard to implement, as their method involves custom CUDA kernels, while our method is designed for pure PyTorch [10] implementation. Loss in the sinogram space is also used in [11] to enforce so-called robust equivariance.

Self-supervision is increasingly popular in the CT community, as the emerging methods to train image reconstruction NNs without high-quality image targets show. Self-supervision is tightly connected to foundational models, which

are general-purpose models pre-trained using self-supervision on data at scale and fine-tuned using supervision on limited data. Such models, like the GPT series from OpenAI or BERT [12], appear in the field of analysis of CT images[13] but are yet to be developed for CT image reconstruction. As such, there is a strong interest in the mathematical formalisation of self-supervision in image reconstruction and the experimental validation of the procedures on real datasets.

Here, we propose a new self-supervised method to train image denoisers for linear inverse problems and develop its mathematical formalisation. We then investigate SDF's performance for two noise settings and compare it with other analytical, iterative and data-driven methods. We further show that SDF is a suitable pretraining strategy for few-shot supervised training of high-quality image denoisers for undersampled sinogram data. We also assert the method's robustness against angular sparsity of the sinogram data and its ability to scale to 3D CBCT.

3 Methods

We define the reconstruction operator $\mathcal{R}_{\theta} \colon Y_0 \to X$ for sinograms in some fixed sub-space $Y_0 \subset Y$ as

$$\mathcal{R}_{\theta} := \Lambda_{\theta} \circ \mathcal{A}_{0}^{\dagger} \,. \tag{2}$$

In the above, $\Lambda_{\theta} \colon X \to X$ is an image restoration/denoising network, $\mathcal{A}_{0}^{\dagger} \colon Y_{0} \to X$ is some handcrafted pseudo inverse of $\mathcal{A}_{0} := \mathcal{A} \circ \pi_{0}$ with $\pi_{0} \colon Y \to Y_{0}$ denoting an appropriate (preferably linear) restriction/re-binning operator that maps sinograms in Y to those in Y_{0} .

3.1 Learning protocol

Learning the reconstruction operator $\mathcal{R}_{\theta} \colon Y_0 \to X$ in (2) is a by-product of learning the image restoration/denoising operator $\Lambda_{\theta} \colon X \to X$, which in turn is typically learned by training against supervised data in X. The key idea in SDF is to learn the reconstruction operator from self-supervised data in Y. We first subdivide the sinogram space Y into disjoint sets, so $Y = Y_1 \cup \ldots \cup Y_M$. The restrictions of the forward operator and its pseudo-inverse to these sets are then given as

$$A_i: X \to Y_i$$
 and $A_i^{\dagger}: Y_i \to X$ for $i = 1, \dots, M$.

With the above, we can for any pair i, j = 1, ..., M define the following sinogram-to-sinogram mapping, as illustrated in Fig. 1:

$$\Gamma_{i,j}^{\theta} \colon Y_i \to Y_j \quad \text{where} \quad \Gamma_{i,j}^{\theta} := \mathcal{A}_j \circ \Lambda_{\theta} \circ \mathcal{A}_i^{\dagger}.$$

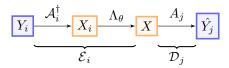


Figure 1: The blue nodes represent sinogram-space data, and the orange nodes represent image-space data. The composition of \mathcal{A}_i^{\dagger} with Λ_{θ} (denoted by \mathcal{E}_i) and A_j (denoted \mathcal{D}_j) can be seen as encoding and decoding parts of a sinogram autoencoder that has X (set of images) as its latent space.

Once the sinogram-to-sinogram mappings have been defined, one can use $\Gamma^{\theta}_{i,j}\colon Y_i\to Y_j$ to predict a sinogram $y_j\in Y_j$ from a sinogram $y_i\in Y_i$. To formalise this, we assume sinograms are generated by a Y-valued random variable $y\sim \mu$. Next, let $\zeta_i\colon Y\to Y_i$ be a re-binning operator that maps a sinogram in Y to one in Y_i . Then we can define the Y_i -valued random variable $y_i:=\zeta_i(y)$ and set-up the following learning problem:

$$\hat{\theta} \in \underset{\theta \in \Theta}{\operatorname{arg\,min}} \sum_{\substack{i,j=1\\i \neq j}}^{M} \mathbb{E}_{\mathsf{y}_{i},\mathsf{y}_{j}} \Big[\ell_{j} \big(\Gamma_{i,j}^{\theta}(\mathsf{y}_{i}),\mathsf{y}_{j} \big) \Big] \quad \text{for some loss } \ell_{j} \colon Y_{j} \times Y_{j} \to \mathbb{R}. \quad (3)$$

To formulate the empirical counterpart of (3), assume we have unsupervised training data in the form of sinograms $y_1, \ldots, y_m \in Y$ that are i.i.d. samples of y. We then use re-binning to generate sinograms in Y_1, \ldots, Y_M by

$$y_l^i := \zeta_i(y_l) \in Y_i$$
 for $i = 1, ..., M$ and $l = 1, ..., m$.

In particular, for each $i,j=1,\ldots,M$, we can generate pairs of sinograms $(y_l^i,y_l^j)\in Y_i\times Y_j$ for $l=1,\ldots,m$ that are random samples of $(Y_i\times Y_j)$ -valued random variable (y_i,y_j) . One may view these pairs as some form of "supervised" sinogram data that can be used for learning $\hat{\theta}\in\Theta$ by minimising

$$\theta \mapsto \sum_{\substack{i,j=1\\i\neq j}}^{M} \mathcal{L}_{i,j}(\theta)$$

where the objective $\mathcal{L}_{i,j} \colon \Theta \to \mathbb{R}$ is the empirical counterpart of the expectation in (3):

$$\mathcal{L}_{i,j}(\theta) := \frac{1}{m} \sum_{l=1}^{m} \ell_j \left(\Gamma_{i,j}^{\theta}(y_l^i), y_l^j \right) \quad \text{for } \theta \in \Theta \text{ and } i, j = 1, \dots, M$$

For i = 1, ..., M - 1 we can in particular learn $\hat{\theta} \in \Theta$ by minimising

$$\theta \mapsto \sum_{i=1}^{M} \mathcal{L}_{i,i+1}(\theta).$$
 (4)

One can now use a gradient descent (GD) scheme to minimise (4), which with step-size $\omega > 0$ reads as follows:

$$\theta_{k+1} = \theta_k - \omega \sum_{i=1}^{M} \nabla \mathcal{L}_{i,i+1}(\theta_k) \quad \text{for } k = 0, 1, \dots$$
 (5)

An alternate way that avoids the summation in (5) is the following:

$$\theta_{k+1} = \theta_k - \omega \nabla \mathcal{L}_{i_k, i_k+1}(\theta_k) \quad \text{where } i_k := k \pmod{M-1} + 1.$$
 (6)

The advantage of 6 is to reduce the memory footprint of the training procedure and will be used throughout the experiments.

3.2 Comments and remarks

One can think of SDF as training an auto-encoder of which latent space is the image. Autoencoders are neural networks that learn a latent representation of unlabeled data. They comprise an encoder $\mathcal E$ that maps the input to a low-dimensional, latent representation z and a decoder $\mathcal D$ that transforms the latter back to the input data. When trained to reconstruct a subset of their data given another, these autoencoders are said to be trained in a self-supervised fashion. The design of these networks, i.e. the properties of $\mathcal E$, $\mathcal D$ and z, is more a matter of hyper-parameter tuning than a proper model-informed procedure. As such, when building a sinogram autoencoder to recover image data, we propose that the latent space be an array with dimensions equal to the ones of the volume on which the reconstruction will be computed. Following that hypothesis, we suggest that $\mathcal E$ and $\mathcal D$ should entail the $\mathcal A^\dagger$ and $\mathcal A$ operators corresponding to the geometry at hand. We finally put forward the idea of training the neural network in a self-supervised fashion by dividing the sinogram into subsets, allowing the processing of sparse sinogram data.

As with the sinogram sub-sampling strategy, one could devise several strategies to iterate through the subsets. In this report, we train Λ_{θ} to infer the next closest subset to the one we provide, but inferring less correlated subsets might be relevant given the task at hand. We experiment with orbital subsets for CBCT and angular subsets for both CBCT and 2D CT.

4 Experiments

4.1 Datasets

We experiment only on datasets containing real measurements. For 2D CT, we use the 2DeteCT dataset [14]; for 3D CT, we use the Walnuts dataset [15]. The LION [16] toolbox provides pre-processing and data-loading functions for the two datasets. We refer the reader to the original papers for a description of the geometries used in these datasets, which we used as is.

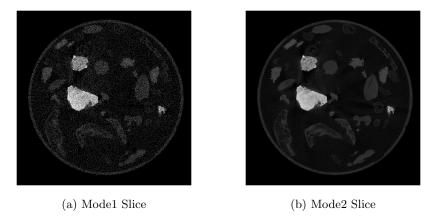


Figure 2: Slices of the same object reconstructed from mode1 (a) and mode2 (b) sinograms.

4.1.1 2DeteCT dataset

The 2DeteCT dataset contains 5000 2D slices reconstructed by Nesterov Accelerated Gradient Descent associated with sinograms acquired with different modes and fan-beam geometry. The "Mode1" corresponds to sinograms with low photon count, and the "Mode2" corresponds to sinograms with normal photon count. Mode1 sinograms can be considered as "noisy" and Mode2 as "clean", as per the wording of the original paper. Clean sinograms were acquired using a tube power of 90W, whereas their noisy counterparts were acquired using a tube power of 3W. This dataset allows us to experiment on sparse-view CT and low-photon count CT. Fig. 2 shows a slice of the 2DeteCT dataset acquired with Mode1 and Mode2.

4.1.2 Walnuts dataset

The Walnut dataset contains 42 tomographic sinograms acquired using a CBCT geometry comprising three orbits, noted 1, 2 and 3, with 1200 intensity measurements for each orbit. The corresponding 3D volumes are 501-voxel sized, and they are reconstructed from the over-sampled sinogram data by least-squares (LS) reconstruction¹ using accelerated gradient descent (AGD), this dataset allows us to experiment with angular and orbital sparsity. Fig. 3 shows a slice of a volume reconstructed with the Feldkamp-Davis-Kress (FDK) method from one orbit and least-squares reconstruction from three orbits.

4.2 Implementation Details

For a fair comparison between methods, we experiment with a simple Λ_{θ} network, defined as a vanilla-convolutional neural network (CNN), and illustrated

 $^{^{1}\}mathrm{This}$ is minimizing least-squares error in sinogram space without explicit regulariser.





(a) FDK + orbit 1

(b) LS + all orbits

Figure 3: Central slice of the same volume reconstructed from FDK ran on orbit 1 (a), and AG ran on all orbits (b).

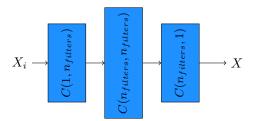


Figure 4: Architecture of the image denoiser Λ_{θ} used in all our experiments. C(n,m) denotes a convolutional layer with n input layers and m output layers, a kernel size of 5 and a padding of 2, with LeakyReLU activation. For 3D CT, $n_{filters} = 8$ and for 2D CT, $n_{filters} = 32$.

in Fig. 4. We use Normal weight initialisation, the Adam optimiser, with a learning rate of 10^{-5} and set the batch size to 8 for 2D and 4 for 3D. The training takes 40 epochs over the dataset, split unless mentioned otherwise, between 80% training, 10% validation and 10% testing. We use gradient clipping, as we found it to stabilise training. We use Tomosipo [17] and its ASTRA [18] backend to implement differentiable forward and backward operators.

4.3 Low Photon Count and Sparse-view CT

This experiment assesses how SDF performs at training a denoiser to improve sparse-view images reconstructed from Mode1 and Mode2 sinograms of the 2DeteCT dataset. To do so, we use an angular subsampling of factor 10, reducing the 3600 measurement sinograms to 360 uniformly distributed measurement sinograms. This yields ten angular subsets and $(\mathcal{A}, \mathcal{A}^{\dagger})$ pairs.

We compare SDF to analytic, iterative and self-supervised image reconstruc-

Mode	Method	Sparse-View PSNR	
1	FBP LS SDF Noise2Inverse	6.77 ± 2 26.64 ± 1.94 $\mathbf{26.96 \pm 1.94}$ 23.05 ± 1.96	
2	FBP LS SDF Noise2Inverse	21.99 ± 1.95 31.03 ± 1.88 30.59 ± 1.91 28.66 ± 1.93	

Table 1: Comparison of the peak signal-to-noise ratio (PSNR) reconstruction metric for FBP, LS, SDF and Noise2Inverse for two modes of the 2DeteCT dataset. Bold font is used to highlight the best metrics.

tion frameworks, respectively, the filtered backprojetion (FBP), the LS and the Noise2Inverse. Our use case is sparse-view CT, so we train the Noise2Inverse using the 1:N method, as described in [7]. Our golden standard for image quality is the Mode2, high-quality LS reconstruction computed using Nesterov-AGD, as described in [14]. Our results are presented in Table 1.

For Mode1 sinograms, SDF provides a 20dB and a 3.9dB improvement against FBP and Noise2Inverse, respectively. It is also better than the LS, which ran for 100 iterations but only by a short 0.3dB margin. For Mode2 sinograms, SDF provides an 8.6dB and a 1.9dB improvement against FBP and Noise2Inverse, respectively. LS remains the best method for high-dose sinograms, with a 0.4dB improvement compared to SDF. We show the same slice reconstructed using different methods in App. A.

4.4 SDF as a pre-training step for supervised methods

We want to measure how good of a pre-training method SDF is. To do so, we use the validation subset of the 2DeteCT dataset and fine-tune SDF and Noise2Inverse in a supervised way with the Mode2 golden standard reconstruction. We set the learning rate to $5 \cdot 10^{-6}$.

For further comparison, we train a neural network similar to the one described in Fig. 4 in a supervised fashion from scratch on images reconstructed with FBP and LS. All hyperparameters remain the same, except the learning rate we set to 10^{-4} . We present the results in Table 2.

The first observation is that the FBP benefits the most from supervision. On just 10% of the training dataset, this simple method yields a 20.8dB improvement for Mode1 and an 11.3dB improvement for Mode2, placing it second best for this mode. Then, the images produced by LS are the ones benefiting the less from CNN enhancement. Indeed, only 2.6dB is gained for Mode1, and, interestingly, training a CNN in a supervised manner on images reconstructed with LS diminishes the peak signal-to-noise ratio (PSNR) compared to

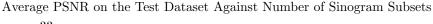
Mode	Method	Sparse-View PSNR	
1	FBP LS SDF Noise2Inverse	27.52 ± 1.95 $\mathbf{29.19 \pm 1.90}$ 28.34 ± 1.95 27.61 ± 1.94	
2	FBP LS SDF Noise2Inverse	33.32 ± 1.93 30.59 ± 1.89 $\mathbf{34.21 \pm 1.94}$ 32.59 ± 1.92	

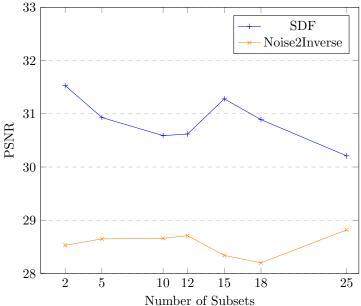
Table 2: Comparison of the peak signal-to-noise ratio (PSNR) reconstruction metric for FBP, LS, SDF and Noise2Inverse for two modes of the 2DeteCT dataset. Bold font is used to highlight the best metrics.

the original, machine-learning-free reconstruction. For the self-supervised pretraining followed by supervised fine-tuning, we report an improvement of SDF by 1.4dB and 3.6dB for Mode1 and Mode2, respectively. As for the Noise2Inverse method, supervised fine-tuning improves the Mode1 images by 4.6 dB and the Mode2 images by 3.9 dB, closing its gap with SDF. Although the comparative improvement yielded by SDF over Noise2Inverse is carried over by supervised fine-tuning, the delta between the two methods closes.

4.5 SDF applied to sparse sinograms

We control for the performance of unsupervised methods against angular sparsity, so we train SDF and Noise2Inverse on sinograms divided into 2, 5, 10, 12, 15, 18 and 25 subsets.





We observe a PSNR gap of about 3dB between SDF and Noise2Inverse. These results are surprising. Although we would expect a reduction of the average PSNR as the number of subsets increases, i.e. when there are fewer measurements per sinogram, we observe that for 15 subsets, the performance of SDF sharply increases by 0.7dB. This means that a sinogram can be sub-sampled from 3600 to 240 angular measurements and still maintain a similar image quality as 1800 measurement sinograms improved with SDF. This opens the door for further investigations on the optimal angular subsampling strategy. We show the difference between the same slice reconstructed with SDF and Noise2Inverse in Fig. 5 and a detailed comparison of the reconstruction's quality for each sparsity level in Fig. 3.

4.6 SDF for Orbital and Angular Subsets of CBCT Sinograms

We experiment on the Walnuts dataset to assess the scalability of our approach to 3D CBCT. This dataset comprises 42 CBCT scans of walnuts, which we split into 35 samples for training and 7 samples for testing. We have found sample 40 to be artifact-laden, as running an FDK on the data yielded a 5dB PSNR. We then exclude this specific sample from the test set, bringing it to 6 walnuts. We divide the sinogram into 36 subsets, 3 orbitals and 12 angular, and use a batch size of one. As such, the input of our network is the FDK of a sample reconstructed from 100 measurements evenly distributed across 2π radians. Each measurement is a 972×768 matrix. Our only comparison is the FDK of each sinogram subset. We evaluate the impact of each orbital subset

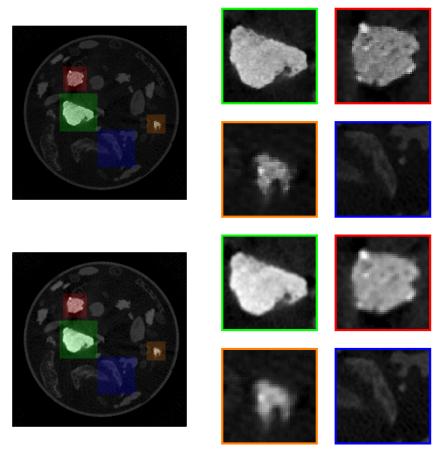


Figure 5: Close-up comparison of the same slice reconstructed from 240 measurements by SDF (top) and Noise2Inverse (bottom).

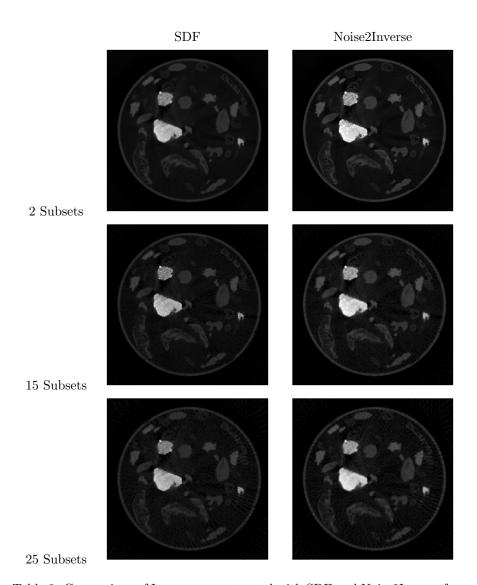


Table 3: Comparison of Images reconstructed with SDF and Noise2Inverse from 2, 15 and 25 subsets. One notable feature of the images reconstructed with SDF is that the high attenuation areas, the lava-stones, remain sharp even when few measurements are available, whereas Noise2Inverse seems to blur the same areas. That SDF feature might be desirable for segmentation purposes.

Method Name	Orbit 1 PSNR	Orbit 2 PSNR	Orbit 3 PSNR
FDK SDF	28.53 ± 2.31 36.21 ± 2.35	28.72 ± 2.34 37.05 ± 2.33	28.70 ± 2.34 37.03 ± 2.39
SDF	50.21 ± 2.50	57.00 ± 2.55	57.05 ± 2.59

Table 4: Results of experiments on angular and orbital subsets of the Walnuts dataset.

on the reconstruction's quality by presenting a per-orbit result in Table 4.

We report a 8.1dB PSNR improvement compared to the base, FDK. We also observe that images reconstructed from the first orbit using both methods are consistently worse than images reconstructed from orbits 2 and 3. Although these results are encouraging, we want to underline that SDF seems to smooth out fine details of the walnuts, as underlined in Fig. 6.

5 Conclusion

We present SDF, a novel framework for training image denoisers in a selfsupervised fashion. We establish its advantage over the other self-supervised reconstruction method, Noise2Inverse, for low and high photon counts. This might come from the fact that Noise2Inverse was not designed to address sparseview CT artifacts and that the 1: X training strategy they delineated is, according to them, inferior to its counterpart X:1. We showed that SDF was a suitable pre-training strategy, as supervised fine-tuning of the pre-trained network on a fraction of the dataset maintains the advantage on Noise2Inverse. However, the gap closes for the studied angular sparsity level. We then study how robust against angular sparsity SDF is compared to Noise2Inverse. We show that higher angular sparsity appears to be beneficial to SDF, a regime where Noise2Inverse struggles. Then, we demonstrate the potential of SDF on CBCT by combining orbital and angular sparsity by downsampling the input singgram by a factor of 36 whilst maintaining a 37dB PSNR similarity to the full-view image reconstructed with LS. This establishes solid ground for the adoption of SDF as a pre-training strategy, as the network can be trained on full-view data acquired on as little as 35 different samples and then used in settings with drastically increased angular sparsity, speeding up the acquisition process substantially.

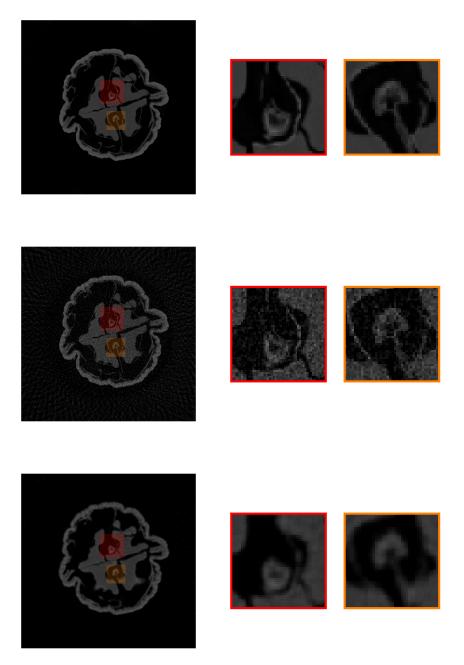


Figure 6: Central Slice of Walnut 39 reconstructed with LS on full sinogram, FDK and +SDF on subsampled sinogram.

Acknowledgements We thank Nathan Kutz for his talk at KTH focus period on SciML, which initiated this investigation. OÖ and EV acknowledge support from Swedish Energy Agency grant P2022-00286 and FORMAS grant 2022-00469. AH acknowledges support from DigitalFutures Scholar-in-Residence grant KTH-RPROJ-0146472 2.

References

- [1] Simon Arridge et al. "Solving inverse problems using data-driven models". In: *Acta Numerica* 28 (2019), pp. 1–174. DOI: 10.1017/S0962492919000059.
- [2] Martin Genzel et al. "Near-exact recovery for tomographic inverse problems via deep learning". In: *International Conference on Machine Learn*ing. PMLR, 2022, pp. 7368–7381.
- [3] Johannes Leuschner et al. "Quantitative Comparison of Deep Learning-Based Image Reconstruction Methods for Low-Dose and Sparse-Angle CT Applications". In: *Journal of Imaging* 7.3 (2021). ISSN: 2313-433X. DOI: 10.3390/jimaging7030044. URL: https://www.mdpi.com/2313-433X/7/3/44.
- [4] Jonas Adler et al. "Task adapted reconstruction for inverse problems". In: *Inverse Problems* 38.7 (2022). Publisher: IOP Publishing, p. 075006.
- [5] Emilien Valat et al. "Empirical evidence of the task-adapted reconstruction framework for joint CT reconstruction and segmentation". In: *Applied Mathematics for Modern Challenges* 2.3 (2024). Publisher: Applied Mathematics for Modern Challenges, pp. 287–300.
- [6] Maureen van Eijnatten et al. "3D deformable registration of longitudinal abdominopelvic CT images using unsupervised deep learning". In: Computer Methods and Programs in Biomedicine 208 (2021), p. 106261. ISSN: 0169-2607. DOI: https://doi.org/10.1016/j.cmpb.2021.106261. URL: https://www.sciencedirect.com/science/article/pii/ S0169260721003357.
- [7] Allard A. Hendriksen, Daniël Maria Pelt, and Kees Joost Batenburg. "Noise2Inverse: Self-Supervised Deep Convolutional Denoising for Tomography." In: *IEEE Trans. Computational Imaging* 6 (2020), pp. 1320–1335. URL: http://dblp.uni-trier.de/db/journals/tci/tci6.html#HendriksenPB20.
- [8] Onni Kosomaa et al. Simulator-Based Self-Supervision for Learned 3D Tomography Reconstruction. _eprint: 2212.07431. 2023. URL: https://arxiv.org/abs/2212.07431.
- [9] Mehmet Ozan Unal, Metin Ertas, and Isa Yildirim. "Proj2Proj: self-supervised low-dose CT reconstruction." eng. In: *PeerJ. Computer science* 10 (2024). Place: United States, e1849. ISSN: 2376-5992. DOI: 10.7717/peerj-cs.1849.

- [10] Adam Paszke et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. _eprint: 1912.01703. 2019. URL: https://arxiv.org/abs/1912.01703.
- [11] Dongdong Chen, Julián Tachella, and Mike E. Davies. Robust Equivariant Imaging: a fully unsupervised framework for learning to image from noisy and partial measurements. _eprint: 2111.12855. 2022. URL: https://arxiv.org/abs/2111.12855.
- [12] Jacob Devlin et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. _eprint: 1810.04805. 2019. URL: https://arxiv.org/abs/1810.04805.
- [13] Ibrahim Ethem Hamamci et al. Developing Generalist Foundation Models from a Multimodal Dataset for 3D Computed Tomography. Leprint: 2403.17834. 2024. URL: https://arxiv.org/abs/2403.17834.
- [14] Maximilian B. Kiss et al. "2DeteCT A large 2D expandable, trainable, experimental Computed Tomography dataset for machine learning". In: Scientific Data 10.1 (Sept. 2023). arXiv:2306.05907 [cs, eess], p. 576. ISSN: 2052-4463. DOI: 10.1038/s41597-023-02484-6. URL: http://arxiv.org/abs/2306.05907 (visited on 06/27/2024).
- [15] Henri Der Sarkissian et al. "A cone-beam X-ray computed tomography data collection designed for machine learning". en. In: Scientific Data 6.1 (Oct. 2019), p. 215. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0235-y. URL: https://www.nature.com/articles/s41597-019-0235-y (visited on 07/10/2024).
- [16] Ander Biguri. Learned Iterative Optimization Networks (LION). URL: https://github.com/CambridgeCIA/LION/tree/main.
- [17] Allard A. Hendriksen et al. "Tomosipo: fast, flexible, and convenient 3D tomography for complex scanning geometries in Python". In: Opt. Express 29.24 (Nov. 2021). Publisher: Optica Publishing Group, pp. 40494–40513. DOI: 10.1364/0E.439909. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-29-24-40494.
- [18] Wim van Aarle et al. "Fast and flexible X-ray tomography using the ASTRA toolbox". In: *Opt. Express* 24.22 (Oct. 2016). Publisher: Optica Publishing Group, pp. 25129–25147. DOI: 10.1364/0E.24.025129. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-24-22-25129.

Appendices

A 2DeteCT - Sparse-View Images

Throughout this appendix, we always refer to the same slice for Mode1 and Mode2, and always highlight the same areas for a comprehensive comparison.

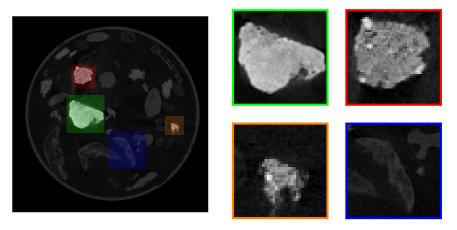


Figure 7: Slice of the 2DeteCT Mode2 Test dataset.

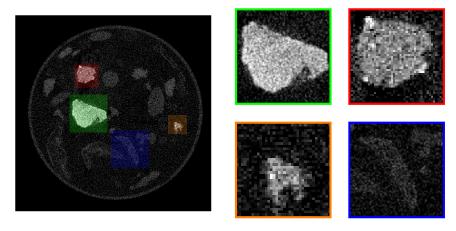


Figure 8: Slice of the 2DeteCT Mode1 Test dataset.

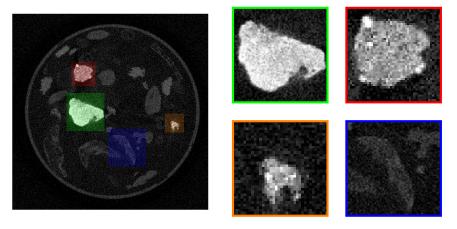


Figure 9: Slice of the 2DeteCT Mode2 Test dataset reconstructed with FBP.

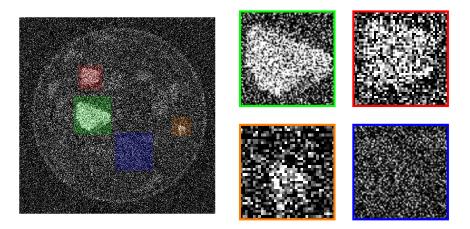


Figure 10: Slice of the 2DeteCT Mode1 Test dataset reconstructed with FBP.

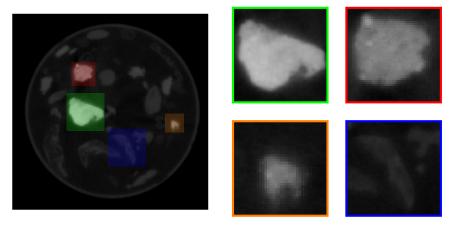


Figure 11: Slice of the 2DeteCT Mode2 Test dataset reconstructed with LS.

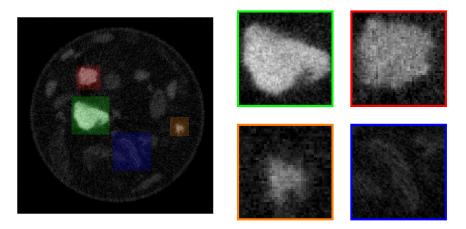


Figure 12: Slice of the 2DeteCT Mode1 Test dataset reconstructed with LS. $\,$

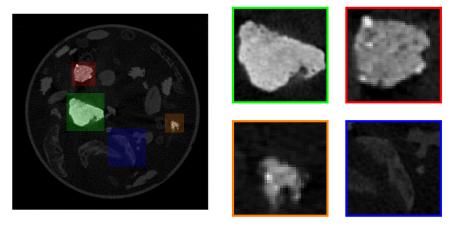


Figure 13: Slice of the 2DeteCT Mode2 Test dataset reconstructed with SDF.

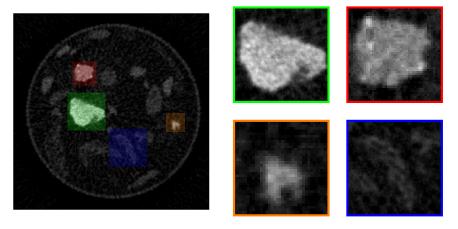


Figure 14: Slice of the 2DeteCT Mode1 Test dataset reconstructed with SDF.

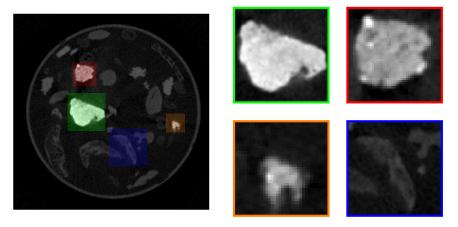


Figure 15: Slice of the 2DeteCT Mode2 Test dataset reconstructed with Noise2Inverse.

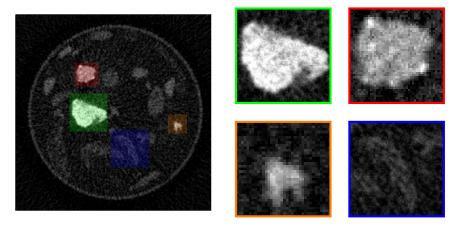


Figure 16: Slice of the 2DeteCT Mode1 Test dataset reconstructed with Noise2Inverse.