Enhancing Reconstruction of Time-of-Flight Neutron Computed Tomography using Artificial Intelligence

Shimin Tang[1[0000-0002-9220-3707]], Mohammad Samin Nur Chowdhury[2[0000-0002-9520-2427]], Diyu Yang[2[0000-0001-7079-4633]], Singanallur V. Venkatakrishnan[3[0000-0002-7637-4987]], Kyle D. Anderson[4[0000-0002-5312-7801]], Ryan Ross[4[0000-0003-3760-234X]], Gregery T. Buzzard[5[0000-0001-9777-3090]], Charles A. Bouman[2[0000-0001-8504-0383]], Yuxuan Zhang[1[0000-0002-0083-1408]], and Hassina Z. Bilheux[1[0000-0001-8574-2449]]

[1] Neutron Scattering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
[2] Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA
[3] Electrical and Engineering Infrastructure Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, USA
[4] Department of Anatomy and Cell Biology, Rush University Medical Center, Chicago, IL 60612, USA
[5] Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA
tangs@ornl.gov and bilheuxhn@ornl.gov

Abstract. Neutron time-of-flight imaging can provide a unique contrast mechanism of crystalline properties. Recently, computed tomography scans using time-of-flight instruments have been used to study structural and spectral characteristics of samples in 3D. To address the challenge of long measurement times associated with hyperspectral neutron computed tomography, the Oak Ridge National Laboratory neutron imaging team has recently demonstrated an autonomous system which can significantly reduce the measurement time by enabling high quality reconstructions from a sparse set of measurements. Some of the core components of such systems are the novel tomographic reconstruction algorithms including those based on artificial intelligence methods.

In this work, a new training method is proposed to improve the performance of the artificially intelligent CT reconstruction algorithms. This training method can improve the quality of the reconstruction from very sparse time-of-flight scans. Our method helps hyperspectral tomography systems to obtain high-quality reconstructions with sparse scanning, which can potentially enable hyperspectral neutron computed tomography with reasonable acquisition times and particularly impact research projects which need to scan multiple similar sample beamlines such as the newly constructed VENUS beamline at the Spallation Neutron Source.

Keywords: time-of-flight, neutron computing tomography, artificial intelligence.

1 Introduction

Neutron hyperspectral or time-of-flight (TOF) imaging provides a unique contrast mechanism in objects by measuring their crystalline properties using Bragg edge radiography, or by detecting their elemental or isotopic content using resonance tomography. The collection of these data sets is lengthy due to the hyperspectral nature of the experiment. Recently, demonstration measurements have been performed resulting in 4 dimensional (4D) hyperspectral reconstructions, i.e. 3D structural and 1D wavelength combined information. However, due to the inherently low signal of each wavelength-dependent projection radiograph, a hyperspectral CT experiment takes longer than traditional a white beam neutron computed tomography (nCT). Therefore, the Oak Ridge National Laboratory (ORNL) neutron imaging team developed an autonomous Hyperspectral Computed Tomography (HyperCT) system to drastically reduce the overall experimental time. HyperCT reduced acquisition time by approximately a factor 3 to 5 (depending on the sample geometry and contrast) using a sample adaptive scanning angle selection method [1] combined with a convolutional neural network (CNN) streaming reconstruction quality evaluation method [2]. Generally, HyperCT can achieve fast TOF nCTs using fewer projections than traditional nCTs that use equally spaced projections and filtered-back projection (FBP) as a reconstruction method. In special cases, such as when multiple similarly shaped objects need to be scanned within a limited neutron beam time, one sample can be scanned using a large portion of the allotted time (hence becoming the ground truth training data) and HyperCT can be used to measure other samples using sparser angular hyperspectral projections.

In some instances, the object structure is very complex, thus the quality of the reconstruction from sparser projections may not provide useful details. More advanced reconstruction algorithms, such as those using deep learning approaches, may be required for sparse angular projections. Micieli *et al.* proposed the Neural Network Filtered Back-Projection (NN-FBP) method to improve reconstruction quality from fast nCTs [3]. Han *et al.* deployed a residual learning method to remove streaking artifacts caused by sparse projection views in CT images [4]. The general adversarial network (GAN) can also be used for 3D tomographic denoising. TomoGAN is an architecture capable of obtaining a high-quality output image from a low-quality input image, and it was applied to synchrotron-based x-ray CTs [5]. These methods all assumed the object was fully scanned with a fixed number of projections. However, models trained using certain samples and a fixed number of projections do not generalize well if the number of projections varies when measuring other samples. Training multiple models using the reconstruction from different numbers of projections individually could be one solution. Venkatakrishnan *et al.* trained several models (Artificial Intelligence AI-CT [6]) using the reconstructions from different numbers of projection for fast FBP and discussed the generalization of a different model [7]. However, training multiple models is a computationally intensive approach. On the other hand, training one model using reconstructions from multiple numbers of projections is a more reasonable solution. The use of deep neural network (DNN) algorithms raises another problem: how to combine data from different reconstructions during training so as to have a

model that works well across different experimental scenarios? Gnanasambandam *et al.* discussed this problem for an image denoiser and used multiple noised level data to train the image denoiser for a better performance [8].In this paper, we proposed a training method to improve the performance of a unique AI model across a range of experimental setups and samples. We introduced the concept of a sparsity factor to represent how sparse the projections used for reconstruction are. Then we included multiple sparsity factors into the training dataset to develop a single training model. Furthermore, we discussed the influence of different sparsity factor data distributions for the performance of the AI model. In particular, we demonstrated that our proposed method outperforms the traditional FPB method when collecting sparse projections.

2 Methodology

In this work, we are trying to shorten the neutron beam time required to perform hyperspectral nCT of multiple similar samples at the Spallation Neutron Source (SNS). In practice and due to the limited beam time, only one or two samples can be scanned using a large number of projections (of the order of the Nyquist sampling rate [9]). Hence, to complete the hyperspectral nCT of multiple samples, other samples can only be acquired using sparse scanning methods with the risk of influencing negatively the quality of the final 4D reconstructed objects. Therefore, we demonstrated a machine learning (ML) model to enhance the quality of the reconstructions using sparse projections. This model is trained to generate a high-quality reconstruction from a low-quality reconstruction. Thus, with this model, we can significantly reduce the number of projections needed for other similar samples. In this section, we will introduce details of the ML model workflow, as illustrated in Fig. 1, and its training mechanism.

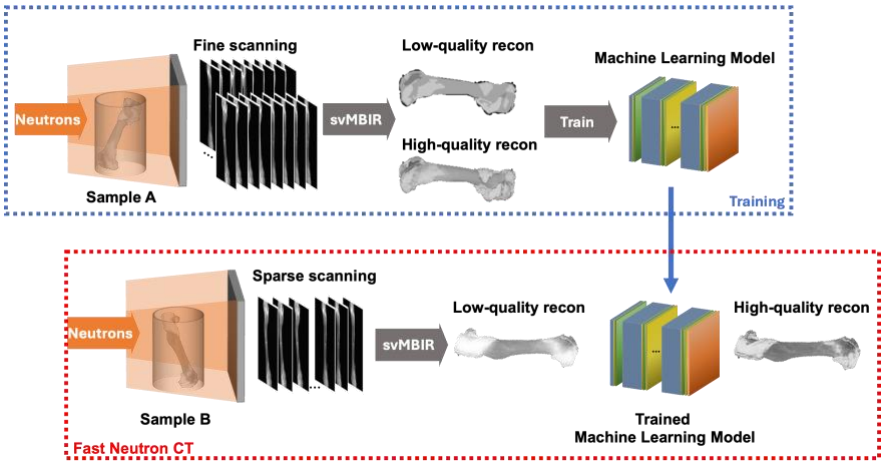

Fig. 1. The flowchart of hyperspectral nCT: the ML model is trained using the data reconstructed from a sufficiently large number of projections based on Nyquist's sampling theorem called "fine scanning" in this figure. Then the trained model is used to improve the quality of the reconstructed data from sparse projections of similar samples.

2.1 Enhancing the image quality of reconstructed data using the AI-CT model

We selected the 2.5D AI-CT [6] ML model to improve the quality of the reconstruction from sparse projection measurements. The structure is designed to learn the non-linear mapping between residual information and the low-quality reconstructed slices input data (from which the 4D volumes are built), where the residual information refers to the difference between high-quality and low-quality reconstructions. In this case, the low-quality reconstruction input data is generated from the sparse projections and has low SNR; we set it as $\boldsymbol{X}_{in}$. Moreover, there is a high-quality reconstruction (of the same object), $\boldsymbol{X}_{GT}$, generated from sufficient projections and has much a higher SNR, which is the ground truth (GT) of $\boldsymbol{X}_{in}$. The residual information is denoted as $\boldsymbol{v}$, where $\boldsymbol{v} = \boldsymbol{X}_{GT} - \boldsymbol{X}_{in}$. The non-linear mapping, $\mathcal{M}(\boldsymbol{X}_{in};\ \boldsymbol{\theta})$, is determined by minimizing the loss function, $\mathcal{L}(\boldsymbol{\theta})$, described as:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N_T}\sum_{i=1}^{N_T}\|\mathcal{M}(\boldsymbol{X}_{in};\ \boldsymbol{\theta}) - \boldsymbol{v}\|^2 \qquad (1)$$

where $\boldsymbol{\theta}$ refers to the parameters used in the model and N_T is the total number of training data at each training batch. $\|\cdot\|^2$ here denotes the l^2 norm.

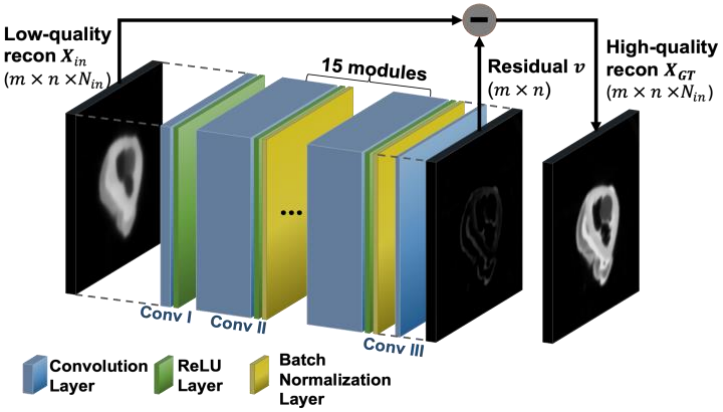

Fig. 2. The structure of the AI-CT model which consists of 17 convolution modules defined from 3 types of convolutions: Conv I is the combination of a convolution layer and a rectifier (ReLU); Conv II is the convolution layer followed by a batch normalization and ReLU; and Conv III refers to a single convolution layer. The low-quality reconstruction patch size is $m \times n \times N_{in}$, and finally, the output residual image size is $m \times n$. Subtracting residual image from input slices can eventually improves the quality of the reconstructed slices input data.

Fig. 2 shows a schematic of the 2.5D AI-CT model which consists of 17 convolution modules. There are 3 types of modules used in this model: The Conv I module refers to one convolution layer followed by a rectifier (ReLU) [10]. The Conv II module consists of one convolution layer, a batch normalization [11] and a ReLU. Finally, the Conv III module is a single convolution layer. The convolutional kernel in each convolution layer has the form $(3 \times 3) \times N_{in} \times N_{out}$, where (3×3) is the size of

convolution kernel, N_{in} and N_{out} are the input and output channel numbers, respectively. The model starts with one Conv I module, then 15 Conv II modules, and ends with one Conv III module. The input data is a batch of slices (3D), and the output data from the last module is the residual 2D image. This is the reason why this model is named the 2.5D AI-CT model. The training procedure aims to tune this residual image in order to minimize the difference between the slices input data and the residual image to approach the high-quality ground truth.

2.2 Training scheme

In order to train the AI-CT model, pairs of low-quality and high-quality reconstructions are entered into the model. In this work, we used the svMBIR algorithm [12] to generate all CT reconstructions. Our previous work [7], which also used AI-CT to improve the quality of the reconstruction, showed the model trained using the reconstructions from the same number of projections (K_1) did not significantly improve the quality of reconstructions generated from different number of projections (K_2). To solve this challenge, we adjusted the AI-CT training scheme by using reconstructions generated from different numbers of projections to train the model.

We define a sparsity factor, δ, to represent the sparsity level of projections used for each reconstruction. The number of projections used in sparse reconstruction is $\left\lfloor \frac{N_{sample}}{\delta} \right\rfloor$; where N_{sample} is fixed and is the total number of projections used for the high-quality reconstruction, and δ is an integer. When δ is large, the number of projections is sparse. When δ equals 1, all projections are used for the reconstruction. Additionally, we set the reconstruction generated using a number of projections with a sparsity factor δ_i as $\boldsymbol{X}^{\delta_i}$, where $\delta_i \in \Delta_{N_\Delta} = \{\delta_1, \dots, \delta_{N_\Delta}\}$ and $i \leq N_\Delta$; N_Δ refers to the number of the different sparsity factors used in this work. The AI-CT model that we use is a "2.5D" network, i.e. it uses a collection of neighboring slices in order to suppress artifacts from each input slice. In our work, we set the number of neighboring slices in AI-CT to 5. For one reconstruction $\boldsymbol{X}^{\delta_i}$ with a total of M cross-sections slices, we can have $N_{tot} = \frac{N_{sample}}{5}$ non-overlapping samples used to train the model. Each sample is noted as $\boldsymbol{x}_j^{\delta_i}$ where $j \leq \frac{N_{sample}}{5}$, then let $\Pi_{N_{\delta_i}}^{\delta_i} = \left\{\boldsymbol{x}_0^{\delta_i} \dots \boldsymbol{x}_j^{\delta_i} \dots \boldsymbol{x}_{N_{\delta_i}}^{\delta_i}\right\}$ be the set of samples randomly selected from $\boldsymbol{X}^{\delta_i}$ where N_{δ_i} is the number of samples. Therefore, the training dataset can consist of multiple sets from different $\boldsymbol{X}^{\delta_i}$ and can be noted as $T = \{\Pi_{N_{\delta_1}}^{\delta_1} \dots \Pi_{N_{\delta_{N_\Delta}}}^{\delta_{N_\Delta}}\}$. The total number of training samples, N_{tot}, is given by: $N_{tot} = \sum_{i=1}^{N_\Delta} N_{\delta_i} = \frac{M}{5}$.

2.3 Composition of the training dataset

Since the training dataset contains samples from reconstructions using different sparsity factors, the number of samples (N_{δ_i}) from each $\boldsymbol{X}^{\delta_i}$ is also a factor that can impact the performance of the trained model. To explore the influence of the composition of the

training data set, we designed 3 training data compositions for comparison. The first training dataset, T_{single}, consists of samples from the same reconstruction $X^{\delta_{i^*}}$ with a sparsity factor δ_{i^*}. Let us set $N_{\delta_{i^*}} = 0, i \neq i^*$, then $T_{single} = \left\{\Pi_{N_{\delta_{i^*}}}^{\delta_{i^*}}\right\}$, where $N_{\delta_{i^*}} = N_{tot}$ is the total number of training samples. The AI-CT model trained with T_{single} is called M_{single} and is considered the baseline in the performance evaluation. We composed the second training dataset, $T_{uniform}$, with the samples from reconstructions using multiple sparsity factors. In this dataset, an equal number of samples are randomly selected from each $\boldsymbol{X}^{\delta_i}$. Let $N_{\delta_i} = \frac{N_{tot}}{N_\Delta}$, and $T_{uniform} = \{\Pi_{N_{\delta_1}}^{\delta_1}, \dots, \Pi_{N_{\delta_{N_\Delta}}}^{\delta_{N_\Delta}}\}$. The model trained by $T_{uniform}$ is denoted as $M_{uniform}$. The last training dataset T_{RoT} uses a rule-of-thumb distribution [8] to compose the training dataset. With this rule, the majority of the samples are allocated to high-quality cases and a few to low-quality cases. The model trained by T_{RoT} is denoted as M_{RoT}.

2.4 Performance evaluation

Three models M_{single}, $M_{uniform}$, and M_{RoT} are trained using different compositions of training data, as described in section 2.3. To evaluate the performance of each model, the sparse-projection reconstructions are used as input to the trained neural network, and the improved reconstructions are generated as the output. The normalized mean squared error (NRMSE) between the improved reconstructions and corresponding ground truths is used to compare the performance.

3 Implementation

Neutron computed tomography was performed on two rat femurs called Sample A and B, respectively. A 1.5 mm diameter cylindrical titanium is implanted in each femur. The samples include rat bone tissue which is primarily comprised of an inorganic calcium phosphate mineral phase and an organic phase, primarily made of type I collagen. The embedded titanium implant is a cylindrical implant with a 10 mm^2 cross-section and 750 μm walk thickness. Sample A is used to train the three AI-CT models with a total of 1165 reconstructed slices. Sample B is the test sample which has a total of 1250 reconstructed slices. Each sample is scanned using 694 projection angles at the High Flux Isotope Reactor (HFIR) Multimodal Advanced Radiography Station (MARS) imaging beamline. Fig. 3 shows the 3D rendering of samples A and B, respectively. Sample A was used to generate 6 (N_Δ = 6) reconstructions with different sparsity factors, $\delta_i \in \Delta_6 = \{2, 4, 6, 8, 16, 32\}$. Reconstructed cross-sectional slices of Sample A generated by different numbers of projection (i.e., different sparsity factors) are also displayed. As the sparsity factor increases, the quality of the reconstructed slice significantly decreases showing blurred edges and limited sample details.

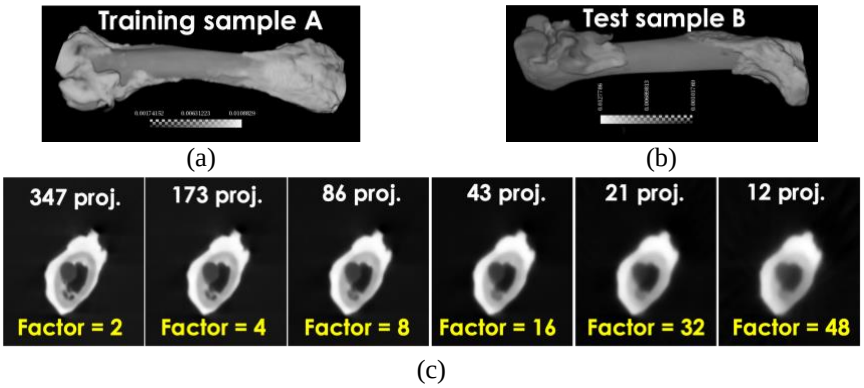

Fig. 3. 3D volume rendering of (a) Sample A used for model training; (b) Sample B used for testing models; and (c) reconstructed cross-sectional slices of Sample A using different sparsity factors (the corresponding number of projections are displayed on top of each slice). The rat femurs are approximately 40 mm long.

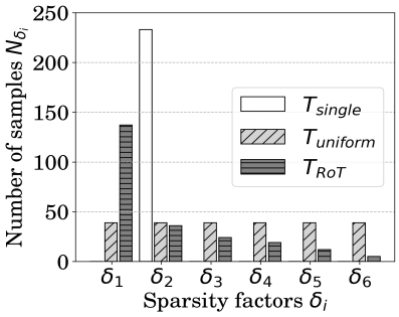

Fig. 4. The compositions of training datasets: the sample number selected from each reconstruction is plotted as a function of the sparsity factor used to generate the corresponding reconstructions.

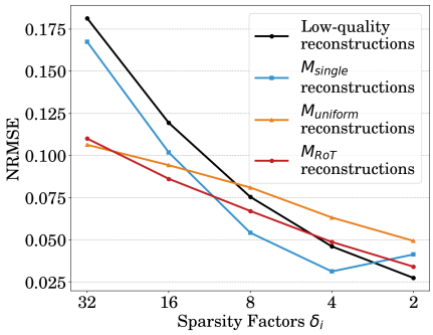

Fig. 5. NRMSE plot as a function of the sparsity factor for sample B

The total number of cross-sectional slices is $N_A = 1165$, thus the training dataset size is $N_{tot} = \frac{N_A}{5} = 233$. There are 3 training datasets that use the different compositions $\{T_{single}, T_{unifrom}, T_{RoT}\}$. Fig. 4 illustrates the dataset compositions as a function of the sparsity factors. We set $i^* = 2$, $\delta_{i^*} = \delta_2 = 4$ for T_{single}, then all training samples are from $\boldsymbol{X}^{\delta_2}$, which means $N_{total} = N_{\delta_2} = 233$ and $N_{\delta_i; i \neq 2} = 0$. For $T_{uniform}$, $N_{\delta_i} = \frac{N_{tot}}{N_\Delta} = 38$ which means 38 samples are randomly selected from each reconstruction $\boldsymbol{X}^{\delta_i} \in \{\boldsymbol{X}^{\delta_1}, \ldots, \boldsymbol{X}^{\delta_6}\}$. For T_{RoT}, we set $N_{\delta_1} = 0.6N_{tot}$, $N_{\delta_2} = 0.15N_{tot}$, $N_{\delta_3} = 0.1N_{tot}$, $N_{\delta_4} = 0.08N_{tot}$, $N_{\delta_5} = 0.05N_{tot}$, and $N_{\delta_6} = 0.02N_{tot}$; then the different number of samples are selected from $\boldsymbol{X}^{\delta_i} \in \{\boldsymbol{X}^{\delta_1}, \ldots, \boldsymbol{X}^{\delta_6}\}$ and $\boldsymbol{X}^{\delta_1}$ has the most samples. The number of samples decrease as the sparsity factor increases, as expected. After training, Sample B is used to test the performance of the ML model (in

this case, $N_B = 1250$). There are 5 sparsity factors (N_Δ=5) used for generating the test reconstructions $\{\boldsymbol{X}^{\delta_1}, \dots, \boldsymbol{X}^{\delta_5}\}$, where $\delta_i \in \Delta_5 = \{2, 4, 6, 8, 16\}$ and $\frac{N_B}{5} = 250$ samples in each reconstruction.

4 Results and Discussion

Fig. 5 illustrates the NRMSE of the reconstructed slices as a function of the sparsity factor for Sample B data using the different AI-CT models after they have been trained with Sample A. The results are compared to the original low-quality reconstructed slices (i.e. unmodified by the different ML AI-CT models) as a function of the sparsity factor. As expected, the NRMSE decreases with the decrease of the sparsity factor, i.e. as more reconstructed slices are used for both the original reconstructed slices and the ones improved with the 3 ML models. However, not all models improve the quality of the reconstructed slices. Except for $\delta = 2$, the M_{single} ML model produces an NRMSE lower than the one for the original reconstruction. However, the NRMSE is not reduced significantly when $\delta \geq 16$ indicating this training model is not appropriate for higher sparsity factors. The other two models, $M_{uniform}$ and M_{RoT}, have lower NRMSEs when $\delta > 8$. Since the information of the reconstruction with different sparsity factors is added to the training data (see Fig. 2), the NRMSE is significantly reduced for $M_{uniform}$ and M_{RoT} when $\delta > 8$. Moreover, M_{RoT} performs better across most δ values. However, when $\delta \leq 8$, $M_{uniform}$ becomes unstable and worsens the quality of the slices as compared to the original reconstruction. When the sparsity factor is not very high, for example $\delta = 2$, the reconstruction quality is sufficient and the NRMSEs for all 3 trained models are higher than the original reconstruction. M_{RoT} produces the lowest NRMSE among all three models which shows that M_{RoT} has the best generalization performance. We note that the training procedure in this work limited the size of the training dataset for fair comparison, but the training dataset size can be increased for practical cases. Thus, the performance of M_{RoT} for $\delta \leq 4$ can potentially improve further. A key insight from our work is that a model trained using the proposed "rule-of-thumb" distribution does not over-smooth the results for cases with a larger number of measurements – leading to a lower error in such cases compared to a model trained using the same number of training examples from each scenario.

Sparse projections yield low-quality reconstructed slices which may not provide detailed scientific information for researchers. Using the M_{RoT} model can boost the quality of sparse-view reconstructions. Fig. 6 shows the application of a K-Means clustering based segmentation on the reconstructed data. A cross-section of the rat bone slice (for δ=32) is shown on Fig.6(b) and the corresponding segmented result is shown on Fig.6(d). The slice reconstructed using the M_{RoT} model is shown on Fig.6(c) and its segmented result is showed on Fig.6(e). The titanium implant is barely seen in the low-quality slice of Fig. 6(b) and hence the segmentation cannot provide many details. When using M_{RoT}, the details of the slice become visible as seeing in Fig. 6(c) leading to a dramatic improvement in the segmentation of the slice as evidenced by the titanium implant shown in green.

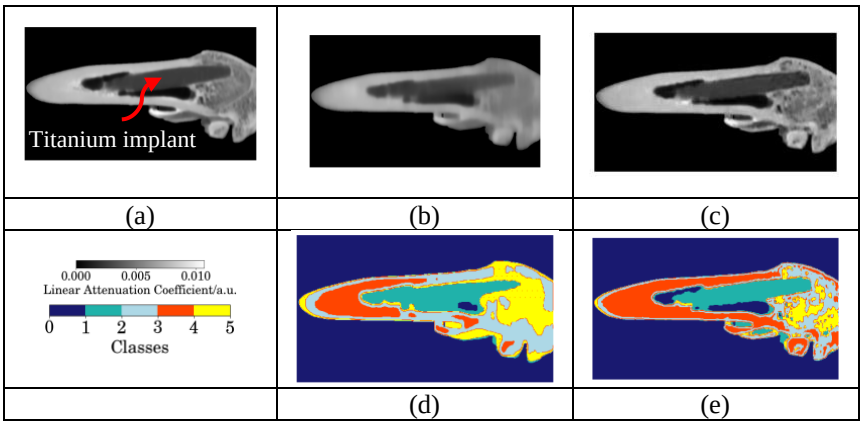

Fig. 6. An example of the application of AI-CT to the rat femurs: (a) ground truth cross-sectional slice from a high-quality scan showing the titanium implant located inside the rat femur; (b) the low-quality reconstruction using only 12 projections; (c) the reconstructed slice processed with M_{RoT}; (d) and (e) are K-Means cluster results of (b) and (c). The classes correspond to different regions in the sample and are used for visualization improvement.

5 Conclusions

In this work, we proposed a new scheme on how to train a ML model called 2.5D AI-CT to significantly improve its generalization performance across a range of experimental scenarios. With the help of the trained model, a high-quality reconstruction can be realized with sparse projections, which can potentially enable hyperspectral neutron computed tomography with reasonable acquisition times. Our research is particularly impactful when multiple similar samples need to be scanned, as required in the biomedical field for example, at beamlines such as the newly constructed VENUS instrument at the Spallation Neutron Source.

Acknowledgements

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). The rat femurs data was measured in MARS HFIR ORNL with IPTS-24959. This material is also based upon work that was supported in part by the Orthopedic Research and Education Foundation and the Rush University Scientific Leadership Council.

References

1. Yang, D., et al., *An Edge Alignment-based Orientation Selection Method for Neutron Tomography*. ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023: p. 1-5.
2. Tang, S., et al., *A machine learning decision criterion for reducing scan time for hyperspectral neutron computed tomography systems.* Scientific Reports, 2024. **14**(1): p. 15171.
3. Micieli, D., et al., *Accelerating Neutron Tomography experiments through Artificial Neural Network based reconstruction.* Scientific Reports, 2019. **9**(1): p. 2450.
4. Han, Y.S., J. Yoo, and J.C. Ye, *Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis.* arXiv preprint arXiv:1611.06391, 2016.
5. Liu, Z., et al., *TomoGAN: low-dose synchrotron x-ray tomography with generative adversarial networks: discussion.* JOSA A, 2020. **37**(3): p. 422-434.
6. Ziabari, A., et al. *2.5 D deep learning for CT image reconstruction using a multi-GPU implementation*. in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*. 2018. IEEE.
7. Venkatakrishnan, S., et al., *Convolutional neural network based non-iterative reconstruction for accelerating neutron tomography.* Machine Learning: Science and Technology, 2021. **2**(2): p. 025031.
8. Gnanasambandam, A. and S. Chan, *One Size Fits All: Can We Train One Denoiser for All Noise Levels?* Proceedings of the 37th International Conference on Machine Learning, 2020. **119**: p. 3576--3586.
9. Kak, A.C. and M. Slaney, *Principles of computerized tomographic imaging*. 2001: SIAM.
10. Xu, B., et al., *Empirical evaluation of rectified activations in convolutional network.* arXiv preprint arXiv:1505.00853, 2015.
11. Ioffe, S. and C. Szegedy. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. in *International conference on machine learning*. 2015. pmlr.
12. Team, S.D., *Super-Voxel Model Based Ierative Reconstruction (SVMBIR)*. 2020.