Correcting Metal Artifacts in CT Sinograms Using U-Net

by

Nicholas Dal Farra

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Waterloo, Ontario, Canada, 2021

Examining Committee Membership

The following served on the Examining Committee for this research paper.

External Examiner: Yaoliang Yu
Assistant Professor, Cheriton School of C.S., University of Waterloo

Supervisor(s): Justin W.L. Wan
Professor, Cheriton School of C.S., University of Waterloo

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The presence of metal implants in patients can lead to streak artifacts in computed tomography (CT) scans. We train Convolutional Neural Networks (CNNs) to correct these errors. We modify the U-Net architecture to reduce overfitting and adopt its output for metal artifact reduction (MAR) tasks. We train two versions of this network for artifact correction; one is trained to output residual error in projection values, while the other outputs corrected projection data directly. Streak artifacts are simulated in clinical CT images by altering the attenuation of x-rays passing through simulated dual hip prostheses. Both networks are capable of reducing numerical error more effectively than competing projection-completion methods (linear interpolation and a CNN inpainting method) in a majority of test cases.

Acknowledgements

Thanks goes to Justin Wan, whose wisdom contributed greatly to this work. I give thanks to my parents, for their support during the pandemic. Finally, I am grateful for the CM department's curation of our little community.

Table of Contents

List of Figures

Chapter 1

Introduction

The Computed Tomography (CT) scan is one of the most commonly used imaging modalities in modern medicine. In 2016, the number of CT scans performed on older adults in Ontario was over four times greater than the number of MRI scans and only 30% less than the number of ultrasounds [5]. One factor explaining the prevalence of CT is that it permits scans of patients with metal inside their bodies. Considering that hip replacements and other prostheses reside overwhelmingly in older patients [6] and that the total proportion of older Canadians has been increasing 2% per annum since 2001 [7], CT imaging may increase significantly in upcoming years.

CT scanners operate by shooting x-rays through a patient at many angles and measuring the x-rays which pass through, which in turn determines the how many x-rays *did not* pass through. Attenuation along different x-ray paths is called the *projection data* and it is used to rebuild a CT image. The resulting scan is a window into the body for medical practitioners who use its contents to inform diagnoses and design treatments.

While the CT modality assumes an idealized passage of x-rays through the patient, metals interfere with their transmission and create errors in the projection data. These errors produce metal artifacts (Figure 1.1) which can occlude diagnostically relevant features. Metal artifacts are a particular hindrance in planning cancer treatments, where imprecise tumor localization leads to improper radiation doses and reduced effectiveness during treatment [8].

Researchers and medical practitioners have attempted to remedy these artifacts through the use of *metal artifact reduction* (MAR) techniques. As an example, by increasing electric potential across the x-ray emitter it is possible to shift the emission spectrum towards energies which are less affected by metal. This approach is representative of the *acquisition*

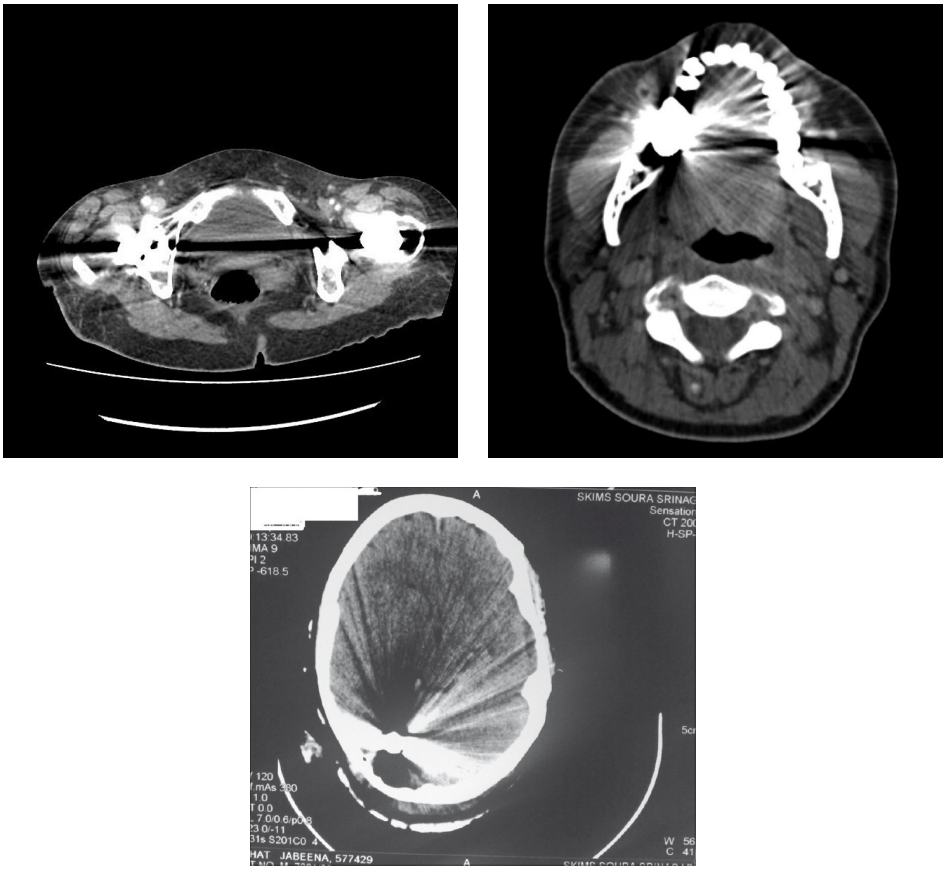

Figure 1.1: Metal artifacts from hip replacements, dental caps, and bullet shrapnel.

improvement family of techniques. There are an additional three relevant MAR categories: *physics-based pre-processing*, *projection-completion*, and *iterative reconstruction*.

We will briefly describe some of the approaches which have been developed in the literature [9]. Under acquisition improvement, Dual-Energy CT (scans using two distinct x-ray spectra) [10] is used to build an absorption profile for each pixel in the image. This attenuation profile can then be used to produce an enhanced CT image and significantly reduces streaking artifacts. Physics-based pre-processing is performed by [11] using non-linear multi-dimensional filtering on pixels to correct for photonic effects. One of the earliest-developed MAR techniques, linear interpolation [12] is a projection-completion technique that replaces the projection values of x-rays that pass through metal objects. Finally, [13] demonstrates that the iterative expectation-maximization (EM) and algebraic

reconstruction technique (ART) algorithms produce fewer metal artifacts than the popular Filtered Back-Projection (FBP) algorithm when reconstructing images.

However, in recent years a new class of MAR algorithm has emerged. The explosion of research on neural networks has lead to the development of machine learning (ML) MAR techniques [14, 15, 16, 3]. Most of these techniques use a variant of the Convolution Neural Network (CNN) architecture which specializes in isolating features in images [17]. The method of [16] is to use of CNNs is to blend computationally cheap MAR techniques into a single more accurate output. Sinogram inpainting with CNNs is performed by [15] to reduce metal artifacts in simulated suitcase objects, and [3] uses a similar method to interpolate sinogram values for patients with dual hip replacements.

A neural network's ability to generalize is closely related to its depth and architecture [18]. U-Net [19] is a state-of-the-art network architecture designed for segmentation tasks in biomedical imaging. It incorporates over two dozen hidden layers and millions of parameters; repurposing this architecture for MAR has the potential to improve the performance of network-based MAR approaches. Additionally, many sinogram-inpainting CNN techniques [16, 3] require knowledge of the position of the metal. This requires an additional backward and forward projection of the image data, as well as the application of a metal-identification procedure. We propose avoiding this step entirely by allowing the neural network to modify all pixels in the projection data. We explore two approaches: one approach involves training a U-Net to output residual errors in the sinogram, while the other trains a network to output corrected sinograms directly.

In Chapter 2 we discuss background material such as the CT modality, causes of metal artifacts, and Convolutional Neural Networks. Chapter 3 outlines our proposed method including our network design, machine learning configuration, and the generation of training data. In Chapter 4 we explore the utility of our proposed model in three test cases and compare performance with linear interpolation [12] and CNN inpainting [3] baselines. Finally in Chapter 5 we summarize our approach and potential directions for future research.

Chapter 2

Background

In this chapter we describe the physical and theoretical principles underpinning Computed Tomography and highlight the role of CT scans as 2D function approximators. Next we explore the properties of metals and x-rays which lead to metal artifacts. Finally we introduce the training and deployment of neural networks for image processing problems.

2.1 Computed Tomography

2.1.1 CT: What is it?

One of the most commonly used imaging modalities in medicine [5], the CT or Computed Axial Tomography (CAT) scan gives a picture of the different materials inside patients' bodies. It does this by shooting x-rays through the patient over a 180° arc and then measuring the amount of radiation that manages to pass through the patient. This means that every CT machine needs two essential components; an *emitter* for producing x-rays, and a *detector array* for measuring them.

The emitter and detector array rotate in synchronization along a gantry, calculating the proportion of blocked x-rays at each angle. This attenuation data (also called projection data) is stored in a 2D image image called a *sinogram*. Sinograms are then fed into a back-projection algorithm to create the final CT image.

Materials in CT scans are identified based on how readily they attenuate photons. Attenuation is measured in *Hounsfield Units* (HU), named after the inventor of the first commercial CT scanner. The Hounsfield scale is a dimensionless linear transformation

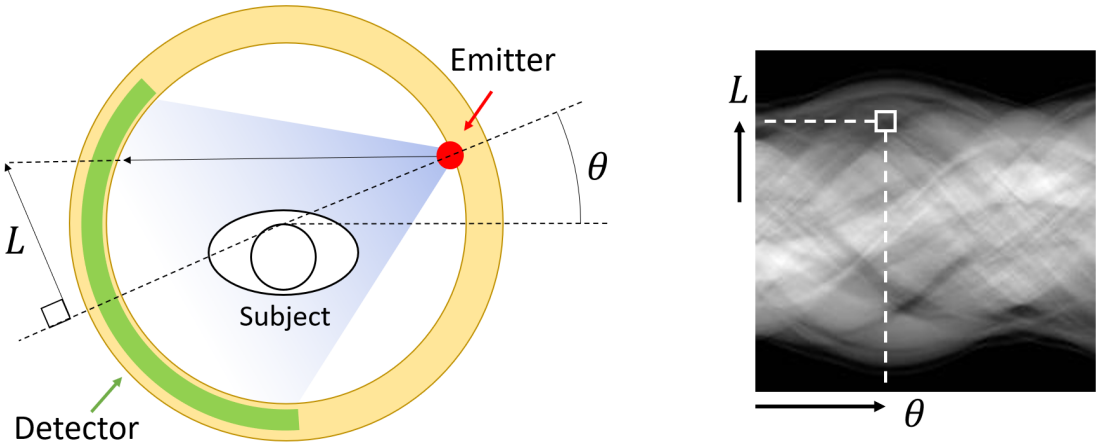

Figure 2.1: The generation and interpretation of a sinogram. White pixels in the sinogram represent high attenuation while black pixels represent negligible attenuation.

of physical SI units and is used to map common body materials to known values in the reconstructed image. Specifically, the scale is defined so that air measures as -1000 HU and water measures as 0 HU. Fat, bone, and other bodily materials can range anywhere from -120 to 1900 HU.

Displaying CT scans on monitors poses its own challenges. While CT values are stored in a 16-bit format and can thus be 2^{16} possible values, most digital displays are only capable of representing integer grayscale values in the range $[0, 255]$ (2^8 distinct values). This means pixel in the CT image must be quantized to fit into a digital display range.

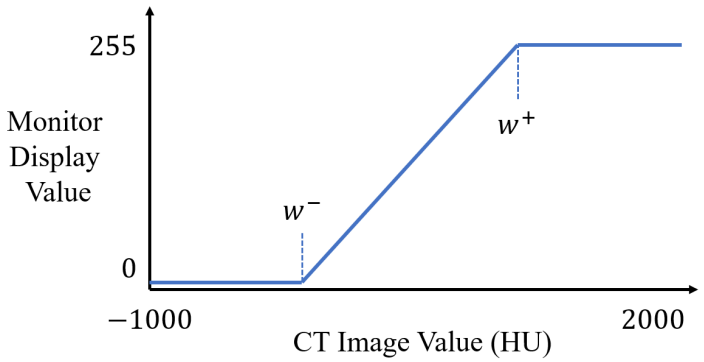

Figure 2.2: A windowing function maps CT image values to displayed intensity.

A solution to this problem is *windowing*. To window a CT image, we choose a minimum window value w^- in HU below which all CT image values are mapped to 0 in the displayed image. Similarly we choose a w^+ value such that all CT values above w^+ are mapped to 255. Finally, CT values in the range $[w^-, w^+]$ are mapped to their displayed value by linearly interpolating between $(w^-, 0)$ and $(w^+, 255)$ then rounding the output to the nearest integer. Windowing is almost always used when CT images are viewed in a digital format, and the choice of window depends on the diagnostic purpose of the scan.

The SI unit for attenuation is cm^{-1} and it describes the *linear attenuation coefficient* of a region of space. Linear attenuation coefficients are denoted μ and they are connected to the probability of attenuation by Equation (2.1) (the Beer-Lambert Law).

$$P(\text{permeation}) = \exp(-\mu x) = 1 - P(\text{attenuation}) \tag{2.1}$$

where x is the depth of the medium in cm. Another important relationship is the connection between μ and m, the *mass-attenuation coefficient*. While μ closely depends on material density ρ (in g/cm^3), m is an intrinsic and density-independent property of a material. The relationship between these values is expressed in Equation (2.2).

$$\mu = m\rho \tag{2.2}$$

Finally, measurements in CT scans are highly dependent on x-ray energy. In general, x-rays of higher energies are more likely to pass through a given medium. The energy dependence of m can vary quite differently between materials, often depending on the nuclear properties of a material's composing atoms. Further, CT emitters produce x-rays over a broad spectrum of energies instead of one singular energy (Figure 2.3). Together these properties can cause *beam-hardening*, which we explore further in Section 2.2.1.

The fact that μ is energy dependent gives rise to an additional complication. Conversion between HU and cm^{-1} requires linearly mapping the value of μ_{air} to -1000 and μ_{water} to 0. However, this requires that we choose an energy at which to specify μ_{air} and μ_{water}. This chosen energy is referred to as the *equivalent monochromatic energy* of the scan. Making this choice lets us convert between HU and cm^{-1} using Equation (2.3).

$$\text{HU} = 1000\,\frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}} - \mu_{\text{air}}} \tag{2.3}$$

2.1.2 The Radon Transform and Radial Projection

In 1917 Johann Radon proved that a function satisfying certain regularity conditions can be exactly reconstructed from an infinite collection of radial projections [20]. Let $f(x, y)$ be

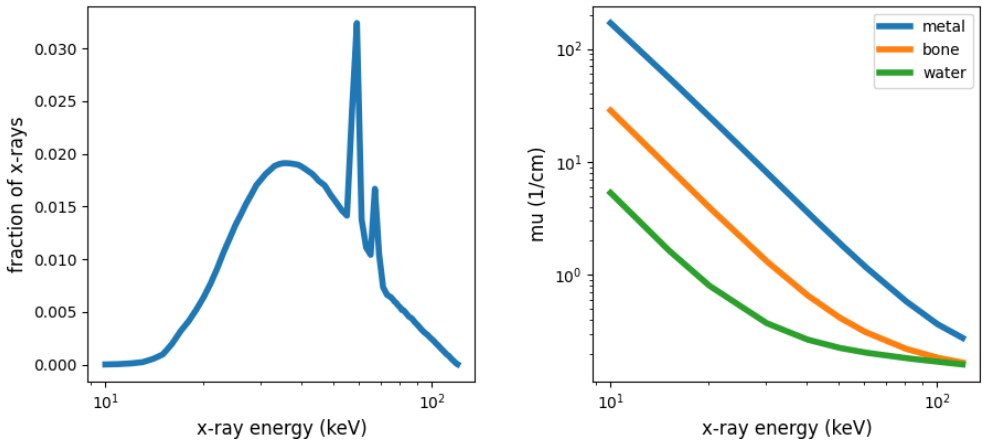

Figure 2.3: The emission spectrum of a CT x-ray emitter [1], as well as μ for ASTM-F75 alloy, bone, and water at different energies [2].

the function which maps 2D points in space to μ at that point, where (x, y) is a coordinate generally defined in reference to the CT scanner. Generally f represents the patient being scanned. Assume $f(x, y) = 0$ for all (x, y) outside of the scanner. The objective of CT is to accurately reconstruct f using a finite set of the function's radial projections.

The theoretical functional which maps f to its radial projections is known as the *Radon Transform* (denoted R) and it is given by Equation (2.4).

$$R(f)(L) = \int_{\ell \in L} f(\ell)\, |d\ell| \tag{2.4}$$

where L is a line in the $\mathbb{R}^2$ plane. During a CT scan, x-rays are attenuated according to the attenuation coefficients inside the patient (Figure 2.4). By measuring the number of x-rays which permeate the patient, it is possible to calculate the attenuation along x-ray paths through the patient. The CT scanner estimates $R(f)(L)$ along a large number of x-ray paths so that the attenuation function f can be later reconstructed.

However, there are many methods to reconstruct f from the set of projections. There exist two major categories of algorithms: analytic methods and iterative methods. While analytic methods like *Filtered Back-Projection* tend to be computationally efficient, the development of more powerful and cost-effective computing resources has enabled a shift towards iterative methods. Iterative methods often require solving systems of hundreds of thousands of variables but tend to produce results with lower error than analytic approaches. One such method is the *Simultaneous Algebraic Reconstruction Technique*

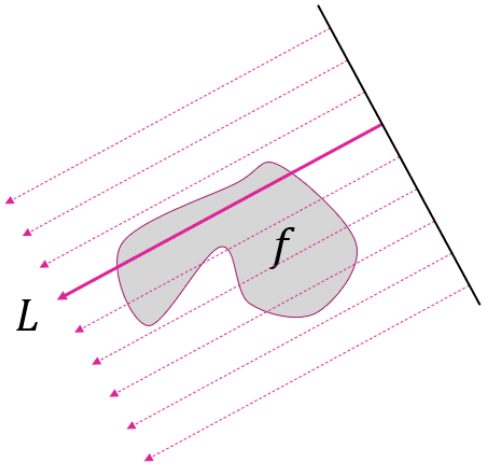

Figure 2.4: The Radon transform of f computes integrals of f along an input line L.

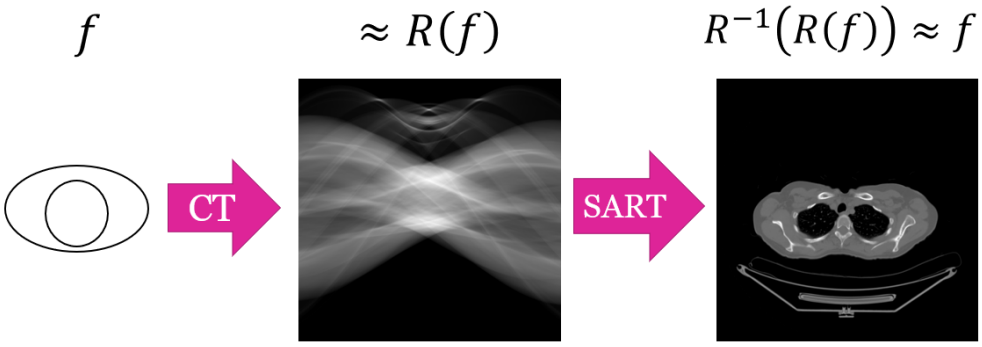

Figure 2.5: While the CT scanner uses x-rays to approximate $R(f)$, a reconstruction algorithm is required to isolate f. We make use of SART.

(SART) [21] which improves the earlier ART algorithm. For the remainder of this essay we use R^{-1} to denote the functional which reconstructs f from evaluations of $R(f)$ along multiple paths. In our implementation we use the SART algorithm.

2.2 Metal Artifacts

2.2.1 Causes

Metals leads to several artifact-causing phenomena in CT imaging. These effects include beam-hardening, scatter, noise, and Non-Linear Partial Volume effect (NLPV).

Beam-hardening results from the interaction of polychromatic x-ray spectra with highly-attenuating materials. An x-ray spectrum is considered "hard" when energies are disproportionately skewed towards higher values. Low-energy x-rays are attenuated at a greater rate than those of higher energy, therefore the distribution of x-rays after passing through metal is "hardened" and more resistant to attenuation.

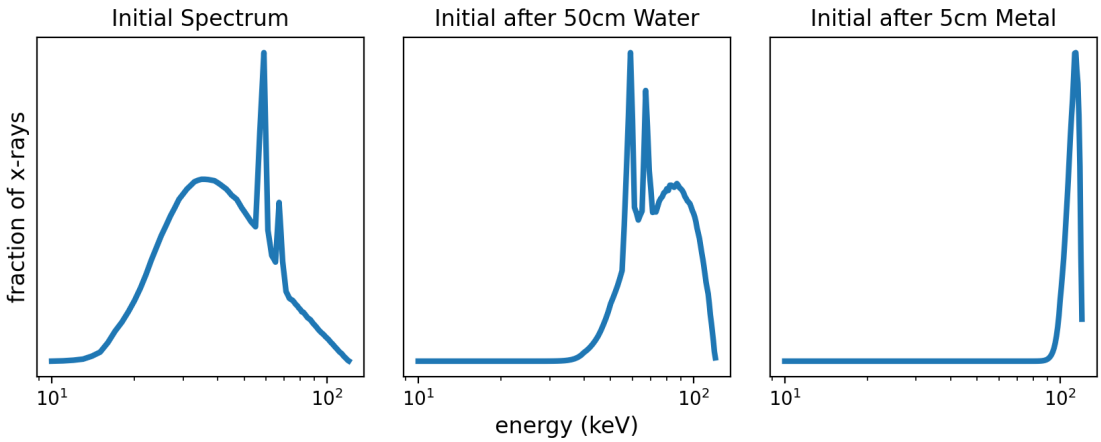

Figure 2.6: Expected CT spectra before and after passing through various media. Calculated using Beer's Law (Equation (2.1)). Metal is ASTM-F75 alloy (Section 3.4.2).

As can be seen in Figure 2.6, just 5 cm of metal has a stronger skewing effect on the x-ray distribution than 50 cm of water (comparable to passing through a human body). This leads to under-estimation of attenuation along high-attenuation x-ray paths, resulting in dark regions or "deletion streaks". These deletion streaks are particularly noticeable along paths intersecting multiple metal objects (see the hip scan in Figure 1.1).

Scatter, more specifically *Compton scatter*, is one of the two principal mechanisms of attenuation (the other being absorption). Compton scatter occurs when photons interact with a charged particle (usually an electron) causing the photon to be deflected from its original path of travel. One drawback of higher energy x-rays in CT is that they have an increased tendency to scatter instead of being absorbed. Scatter deflects x-rays into

inappropriate detector cells and can cause bright lines outside of the deletion streak. Metals are sites of high attenuation and thus are common sources of scattering artifacts.

Background noise also plays an important role in CT imaging. Sources of radiation such as soil, the sun, and other medical equipment can be observed by CT detector arrays. This does not normally pose an issue as the number of background events are negligible relative to the radiation produced by the emitter. However for highly attenuating x-ray paths, background events make up a more significant fraction of detector readings. This leads to small dark and bright streaks which appear to change with each performance of a scan, even when all other CT parameters are held constant.

Lastly we discuss the NLPV effect. Although they are presented as an infinitesimal 2D slice, CT scans actually measure the average of μ in thin 3D voxels. Excessive voxel depth can cause the attenuation characteristics of voxels to depart non-linearly from the average μ of their contents [22]. This produces streaking artifacts at the edges of spatial regions of high contrast such as the boundaries of metal objects.

2.2.2 A Simple Correction Technique: Linear Interpolation

One of the earliest MAR techniques, Linear Interpolation (LI) is a projection completion algorithm which replaces metal-affected sinogram values with linearly interpolated values from their nearest unaffected neighbours. LI is a simple and efficient technique which has been shown to significantly reduce metal streaking artifacts. However this technique may also introduce blur, and its performance depends significantly on the accuracy of an operator-performed segmentation of metal objects [12].

The LI algorithm consists of four main steps and one optional step. We assume that both the CT projection data and metal-affected reconstructed image are available as inputs.

1. Beginning with the metal-affected CT image, an operator highlights all pixels containing metal. The highlighted area is referred to as the "operator mask".

2. The operator mask is forward-projected into the sinogram domain using the Radon transform. The resulting image is called the "mask sinogram".

3. Non-zero pixels in the mask sinogram indicate metal-affected pixels in the CT sinogram. Therefore, we linearly interpolate metal-affected pixels in the projection data (indicated by values > 0 in the mask sinogram) using data from the two nearest non-metal attenuation values. This produces the "*Corrected*" data in Figure 2.7.

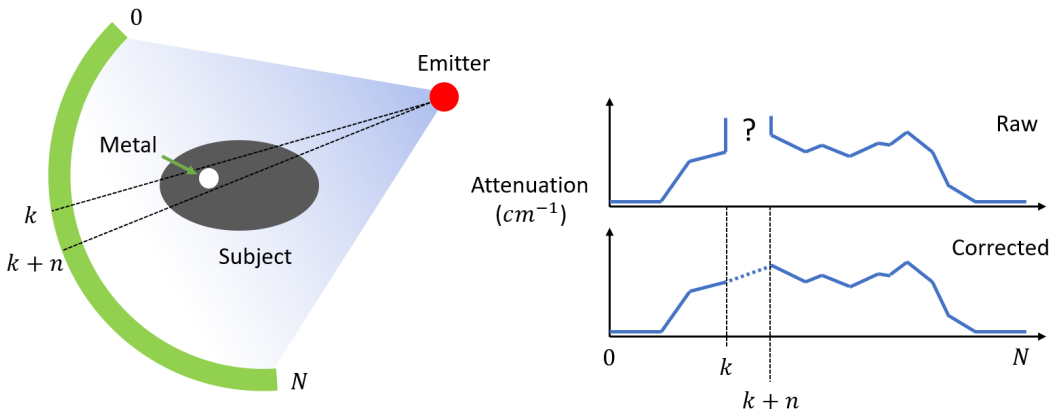

Figure 2.7: Linear interpolation of metal-affected projection values in the sinogram domain.

4. Corrected projection values are transformed to the spatial domain using an inverse Radon transform.

5. (Optional) Insert attenuation values for metal into the spatial region highlighted by the operator. This re-introduces the metal whose projection values have been removed and makes the output image a truer representation of the subject's interior.

2.3 Deep Learning and Artificial Neural Networks

2.3.1 What are Neural Networks?

Neural networks are machine learning tools that can be used to perform many complex tasks. Their structure is inspired by that of a biological brain; they are composed of artificial "neurons" whose activations covary to produce a specific signal or output. Similarly to humans, networks require trial and error to successfully learn new tasks. Their inputs and outputs can be images, category labels, text, or even sound.

Training a neural network can be understood as learning to approximate a function f, where f is the "true" function mapping inputs to answers for the given task. "True" is mentioned in quotations because the existence of such an f is assumed but is not proven for most real-world tasks. Consider the task of classifying animal pictures by species. While f may be extremely difficult to describe a priori, networks are often able to learn

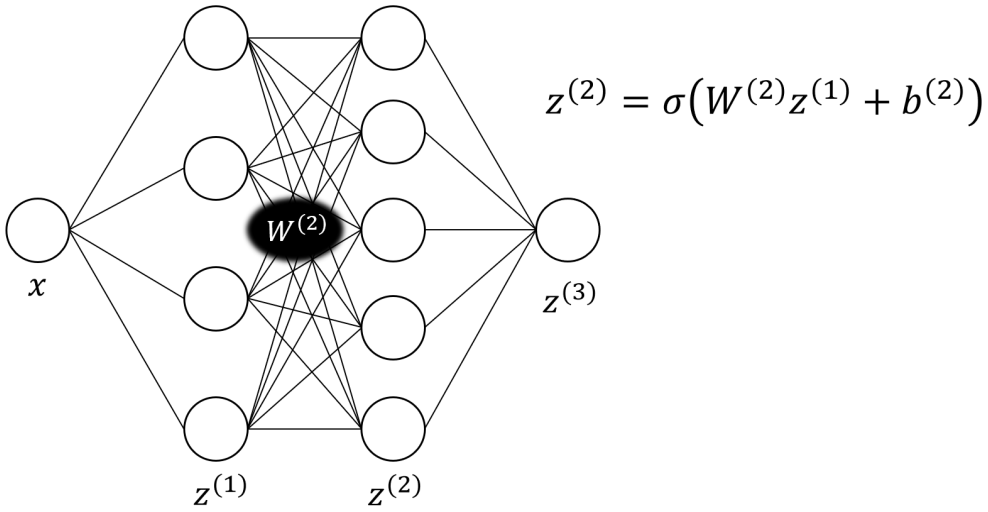

Figure 2.8: A small neural network and the activation equation of the third layer. The activation of the first layer is simply the input data.

accurate representations of f using only labeled pairs of inputs and outputs. Neural nets are so successful as function approximators that they have been used to outperform human experts on complex tasks such as predicting breast cancer [23] and playing Go [24].

Neural nets often contain several layers of neurons. The number of layers in a network describes its *depth*, and the number of neurons in a layer is that layer's *width*. Networks with a depth beyond 2 or 3 layers are called *deep neural networks*. Layers are composed of artificial neurons whose outputs are called *activations*. The vector containing all activations of layer i is denoted $z^{(i)}$ in Figure 2.8. The activation of neuron j at layer i is dictated by four key elements: the activations of the previous layer ($z^{(i-1)}$), the input *weights* ($W^{(i)}$) and biases ($b^{(i)}$), and the activation function (σ). Weights and biases change during training while activation generally remain fixed.

Consider the activation of layer 2. $W^{(2)} \in \mathbb{R}^{4\times 5}$ is a matrix whose entries define the influence of $z^{(1)} \in \mathbb{R}^4$ on $z^{(2)} \in \mathbb{R}^5$. This influence is realized by a matrix-vector multiplication $W^{(2)}z^{(1)}$. Associated with neuron j is a bias $b_j^{(2)}$ which predisposes $z_j^{(2)}$ towards or against activation. Biases are arranged into a vector $b^{(i)} \in \mathbb{R}^5$ and contribute additively towards neuron j's pre-activation value. Finally, σ is a non-linear function (monotonically non-decreasing in general) which computes the final output of neuron j.

2.3.2 Supervised Learning

Supervised learning describes the context where a network is trained using labeled input-output pairs. Consider the problem of recognizing hand-written digits. Training a network to perform this task may require thousands of image-label pairs called *training samples*. During training the network gets better at mapping the input image to its correct label. The network is learning to approximate f, the function mapping images of digits to the number they express.

Suppose our input data is x and we are training the network to output $y = f(x)$. The objective of training is to emulate $f(x)$ in real-world implementations, i.e. predict $f(x)$ on x which may not be in the training set. A network's error on unseen samples is called its *generalization error*. Restated, the objective when training a neural network is to minimize generalization error. Networks used in real-world implementations will not be able to train on new data they see because the label $y = f(x)$, also called the *training target*, is unknown. Therefore low generalization error is essential to the utility of neural networks in real-world implementations.

2.3.3 Training a Neural Network

Define $\theta \in \mathbb{R}^N$ to be the vector containing all learnable parameters (weights and biases) in the network. Suppose $D = \{(x_i, f(x_i))\}$ is the set of training samples and their labels. Assume x_i's are sampled from a distribution P_X. The objective of training is to find parameters that minimize the expected loss of the network over the input distribution; that is, solve for some

$$\hat{\theta} \in \arg\min_{\theta} E_X \left[\, l(g(\theta; X), f(X)) \, \right] \tag{2.5}$$

where $l(\,\cdot\,,\,\cdot\,)$ is the loss function, f is the function approximated by the network, and $g(\theta;\,\cdot\,)$ is the evaluation of the network with parameter set θ.

Neural networks may contain millions or billions of learnable parameters. This precludes a brute-force search over the parameter space (the set of possible θ's) in most cases. Therefore, some form of local search must be performed over the parameter space.

Loss Functions

An important matter is the quantification of error in a network's output. The formula used to calculate error during training is the *loss function*. Two common loss functions for

image processing are *Root Mean Square Error* (RMSE) and *Mean Average Error* (MAE). Let $\boldsymbol{x} \in \mathbb{R}^{M \times N}$ be the result of some image processing method and $\boldsymbol{y} \in \mathbb{R}^{M \times N}$ be the ground truth. Then the RMSE and MAE are given by Equations (2.6) and (2.7) respectively.

$$\text{RMSE}(\boldsymbol{x}, \boldsymbol{y}) = (MN)^{-1/2} \left(\sum_{i \leq M} \sum_{j \leq N} |x_{i,j} - y_{i,j}|^2 \right)^{1/2} \tag{2.6}$$

$$\text{MAE}(\boldsymbol{x}, \boldsymbol{y}) = (MN)^{-1} \left(\sum_{i \leq M} \sum_{j \leq N} |x_{i,j} - y_{i,j}| \right) \tag{2.7}$$

Stochastic Gradient Descent (SGD)

Local search algorithms perform incremental adjustments of θ to solve Equation (2.5). The procedure used to update parameters on each iteration is referred to as the *optimizer*. Such an optimizer is Stochastic Gradient Descent (SGD).

The core idea of SGD is to evaluate a network's error on training samples and then adjust parameters in such a way that would have reduced the network's error. Implicitly, the second step requires calculating each parameter's contribution to the error in the output. The evaluation of network error is called the *forward pass* while the calculation of each parameter's contribution to the error is called the *backwards pass*. While a forward pass is achieved by simply evaluating the network on an input and calculating the error, the backward pass is only made possible by the backpropagation algorithm [25] which makes use of the chain rule from calculus.

At this point we introduce the concept of a *loss landscape* or loss surface. Informally, the loss landscape is a function $L : \mathbb{R}^N \to \mathbb{R}$ which maps choices of network parameters to the expected loss (w.r.t P_X) of a network implementing those parameters. The astute reader may recognize L as the function being minimized in Equation (2.5), i.e. $L(\theta) := E_X [\, l(g(\theta; X), f(X)) \,]$. The minimization of $L(\theta)$ is achieved by adjusting θ in the direction of steepest decrease in $L(\theta)$. Since $\nabla L(\theta)$ always inhabits the direction of steepest increase in L at input θ, the direction of steepest decrease is given by $-\nabla L(\theta)$. Let α be the size of the step in direction $-\nabla L(\theta)$, also called the *step size*.

The expected loss function L is not known exactly because P_X and f are not known. Therefore we approximate L using the sample mean of error on the dataset D. The insight of SGD is to only train on a subset of the data on each iteration; this helps to escape local minima during training and decreases the computational demand of each training

iteration. Using a random sampling of the dataset is the origin of the word "stochastic" in the optimizer's name. The SGD update formula is given by Equation (2.8).

$$\theta_{i+1} = \theta_i - \alpha \nabla_{\theta_i} \left(\frac{1}{|S_i|} \sum_{(x_i, f(x_i)) \in S_i} l(g(\theta_i; x_i), f(x_i)) \right) \tag{2.8}$$

where S_i is a random sampling of chosen size from the dataset. S_i is known as the *mini-batch* and $|S_i|$ as the *mini-batch size*, although it is often referred to simply as batch size. We call one *epoch* the number of training iterations after which the network has trained on $|D|$ samples.

Recall that the overall objective of training is minimizing expected loss over P_X, not just D. It is the case that when networks are trained extensively, their expected loss on samples from D continues to decrease while performance over P_X may actually worsen. This problem is known as *overfitting* and it occurs when the network memorizes the correct outputs instead of generalizing into an approximation of f.

One solution to overfitting is to randomly partition the dataset into a training set and a validation set. Samples from the validation set are withheld from the network during training, and they are representative of unseen samples from the true distribution P_X. Performance on the validation set is monitored during training alongside the network's performance on the training set. If performance on the two sets begin to diverge later into training, this is an indication that training should be terminated as past this point the network will begin to overfit. A common choice of size for the validation set is 20% of the size of the original dataset.

Adaptive Moment Estimation (ADAM)

Significant research has been performed on improving the basic program set forth by SGD. An optimizer that has gained popularity in recent years is Adaptive Moment Estimation (ADAM) which makes several improvements upon SGD [26]. ADAM is compatible with mini-batch training, so that updates are performed using a strict subset of the available data on each iteration. Here we describe some of the relevant features of this optimizer.

Firstly, ADAM uses a separate learning rate for each parameter in the network instead of a fixed learning rate. Inspired by the RMSProp optimizer [27], this has been shown to expedite convergence in networks during training. A second important feature of ADAM is that it implements momentum. Momentum in network training is akin to momentum in physics whereby θ_i can be seen as a rolling ball in $|\theta|$-dimensional space, and momentum

describes the predisposition of θ_i to continue along its current trajectory in opposition to the influence of external forces. This can be advantageous when in helping to escape local minima in L. A third relevant contribution of ADAM is that parameter-specific learning rates are adaptive. The optimizer makes use of second-moment data to slow training when the network parameters are nearing convergence. Finally, ADAM fully automates parameter selection so that no choice of α or any other training parameters are necessary (although initial learning rates and momentum decay may be specified). For these reasons we use ADAM as the primary optimizer of our networks.

Stochastic Weight Averaging (SWA)

Stochastic Weight Averaging (SWA) is a modification applied to a base optimizer that modifies the final iterations of training. The authors of [28] show that many optimizers will orbit about a region of low loss near the end of training. They show that averaging network parameters during training successfully produces parameters nearer to the center of the low loss region which lowers validation error in the network.

The SWA optimizer requires two hyperparameters to be specified: SWA iterations and sampling period. Suppose the network is to be trained for N total iterations. Then at iteration $N -$ SWA iterations, SWA takes over the role of the base optimizer and begins performing SGD at the base optimizer's learning rate (SWA is compatible with parameter-specific learning rates such as those used in ADAM). Additionally, a running average is initialized with the current value of the network's parameters. SGD is performed for $n =$ "SWA iterations" iterations and the running average is updated every $m =$ "sampling period" iterations. Upon finishing m iterations of SGD, the network's parameters are replaced with the running average and training is terminated.

2.3.4 Convolutional Neural Networks

A key development in research on neural networks was the introduction of so-called *Convolutional Neural Networks* (CNNs). Designed specifically for tasks related to imaging, CNNs use the convolution operator to isolate features in 2D inputs. CNNs have been successfully used to win several image processing competitions due to their efficacy [17].

In a CNN, conventional neurons are replaced with learnable units called *convolution kernels*. Convolutional kernels are matrices which are "rotated" 180° and then slid along an input image to isolate features and extract relevant information. Convolution is a very general tool that is used to measure gradients, create blur, or sharpen images. Thus,

assimilating convolution into neural networks gives them an ability to learn operations which are specifically suited to manipulating images [29, Ch. 5.4, p. 289].

In a CNN, layers are made up of one or more convolution kernels. While a standard neural net learns weights between neurons, CNNs learn kernel elements $k_{i,j}$. Additionally, instead of having intermediate activations represented as vectors, intermediate activations in CNNs are 2D images. This is due to the fact that the output of a convolution operation is also an image. CNNs make use of activation functions by applying them element-wise to the intermediate outputs of convolutions. Additionally, each convolution kernel has an associated bias term which acts like the bias term in a regular network.

One issue with convolution is the treatment of convolution near the image border. Many conventions exist to address this. A convention in deep learning is to only convolve at pixels where all kernel elements overlap with elements in the input image [30]. While this avoids convolution with pixels beyond the input, it results in an output image with reduced dimensions. If maintaining constant input/output dimensions is desired, one solution is to pad the input with zeros around its perimeter. We make use of this technique in our own CNN MAR solution.

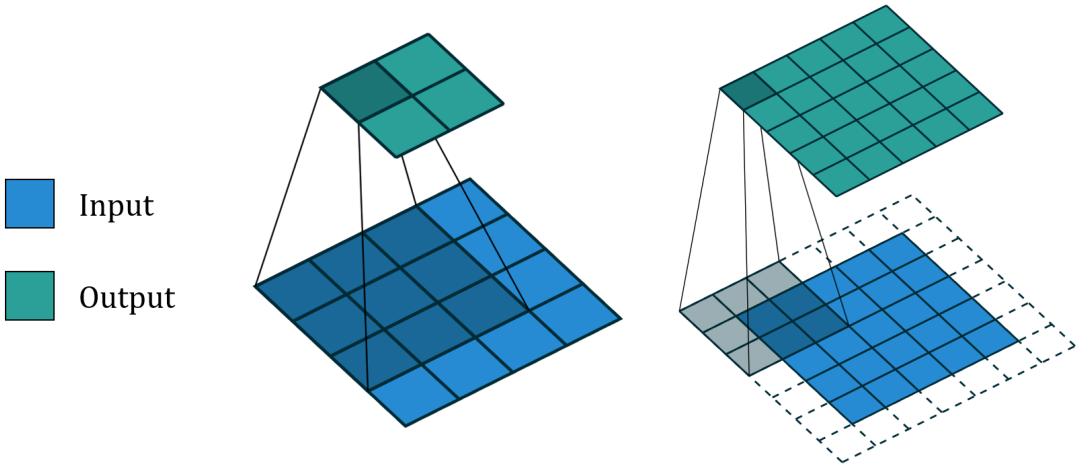

Figure 2.9: Some conventions result in decreased output dimensions after convolution. One solution is to pad the input with zeros.

One MAR technique that uses CNNs is to inpaint using output from a neural network [3]. The CNN predicts corrected projection values by training on a dataset of simulated metal artifacts. Simultaneously, corrupted projection data is back-projected and used to isolate pixels containing metal. Then the Radon transform is applied to metal-containing

pixels and the resulting non-zero sinogram pixels represent locations to be replaced by CNN output. Although requiring computationally expensive forwards and backwards projections, corrections are limited to the trace of the metal in the sinogram. This helps to reduce error in the final reconstructed CT image.

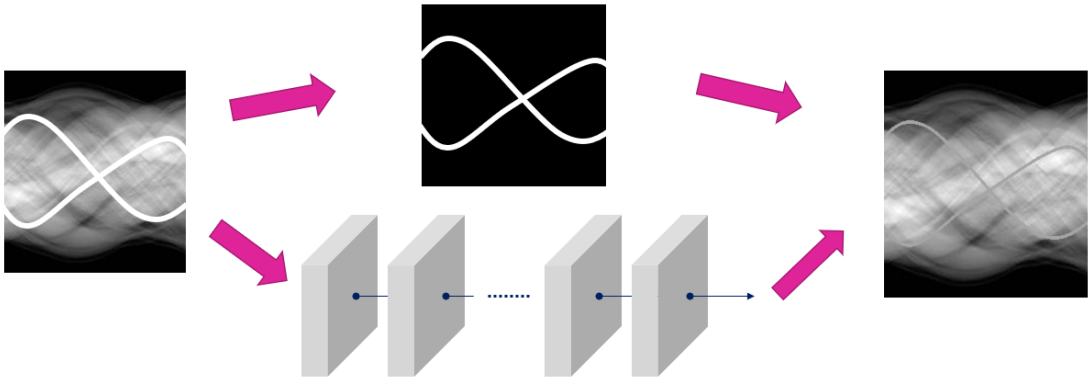

Figure 2.10: The method of [3] is to use a CNN to predict the correct projection values then replace metal-affected projection data with these values.

Chapter 3

Methodology

We propose a projection-completion method using a deep CNN to correct values in the sinogram. While CNN methods have already been used effectively in MAR [3, 16, 14], we explore the utility of two new techniques:

1. We implement a modified version of U-Net [19], a state-of-the-art architecture for segmentation tasks in medical imaging.
2. We eliminate the need for information about metal position by adopting all corrections made by the network.

We train networks to reduce artifacts using two different approaches: the first approach targets residual errors in the sinogram and subtracts them, while the second approach is to return corrected sinograms directly. To access ground truth during training we simulate dual hip replacements made of a Cobalt-Chromium alloy commonly used in bone replacements.

3.1 Overview

Our proposed method belongs within the class of projection-completion MAR algorithms, which includes techniques such as Linear Interpolation (Section 2.2.2) and [3]. However unlike most projection-completion methods, we allow our technique to modify all projection values within the sinogram. Most projection-completion techniques require information

about the location of metal to guide the replacement of corrupted values in the sinogram. This may require performing computationally expensive forward and inverse Radon transforms. By ignoring metal location altogether we can avoid the need for these extra processing steps.

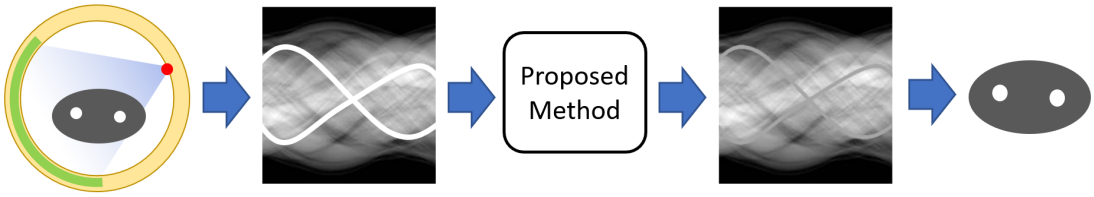

Figure 3.1: The proposed method is applied to projection data rather than images.

The generalizability (Section 2.3.2) of a MAR technique is the measure of its accuracy on samples which are not like the ones it was trained on. Ability to generalize is an important feature for CNN-MAR techniques. While achieving good performance would be assisted by training on a representative dataset of all possible metal artifacts, acquiring such a dataset would be an expensive and time-consuming task, and using all of the samples during training would demand great amounts of computational resources. Good generalization allows a CNN-MAR method to perform well on new implants and patients without needing such a dataset.

The alternative to good generalization is to train separate CNNs for different MAR tasks. While this is certainly possible, it introduces numerous complications such as choosing how networks should specialize, increased memory demands, and requiring operator knowledge to select the right MAR network. Such complications do not arise when using alternative methods like Linear Interpolation.

The success of deep networks is founded in their superior capability to generalize [18]. Deeper MAR networks are capable of better generalization across patients, implants, and metal types. However, it was found that adding more layers to the Fully Convolutional Network (FCN) architecture commonly used MAR networks [3, 15] did not improve performance in most cases. As such, we implement a version of U-Net modified for MAR tasks. U-Net is a deep network architecture that won several 2015 challenges in image segmentation due to its low generalization error. Our objective is to leverage the network's success in a MAR setting.

3.2 Training Target

For reasons discussed in Section 2.2.1, metal causes errors in the projection data. We will construct a mathematical model to express this problem. Let us represent metal artifacts as additive error $\boldsymbol{\epsilon}$ in the projection data. Suppose that $R(f)$ is the Radon transform of f. Applying the inverse radon transform to the true projection data plus $\boldsymbol{\epsilon}$, we get an image containing metal artifacts which we denote f'. Algebraically we have

$$f' = R^{-1}\left(R(f) + \boldsymbol{\epsilon}\right) \tag{3.1}$$

Equivalently, we can write Equation (3.1) as

$$R(f') = R(f) + \boldsymbol{\epsilon} \tag{3.2}$$

Our proposed method is to correct data at every pixel the sinogram. There are many ways in which a network can be used to do this. Here we propose two techniques:

1. Train the network to output $\boldsymbol{\epsilon}$ given $R(f')$, then *subtract* the network's output from $R(f')$ before applying R^{-1}.
2. Train the network to output $R(f)$ given $R(f')$, then apply R^{-1} directly on the output.

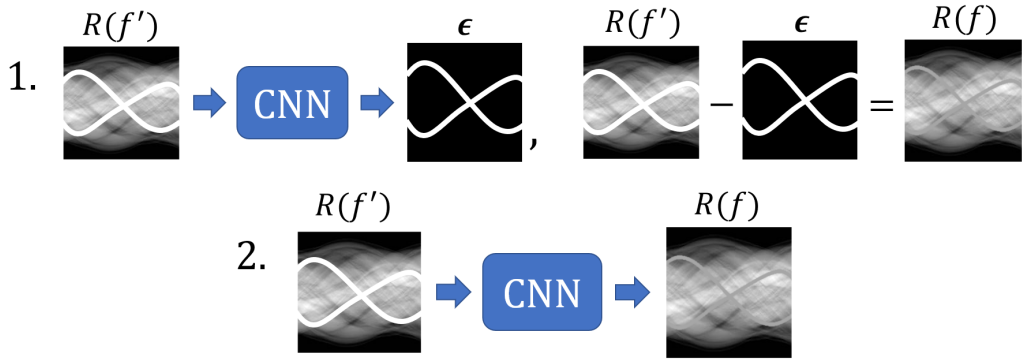

Figure 3.2: We explore two ways in which networks can be used to reduce metal artifacts.

In this essay we refer to technique 1 as the "residual" model (as it targets residual errors in the sinogram) and technique 2 as the "direct" model. While the direct model is used frequently in the literature [3, 15, 16], targeting residuals in the spatial image with CNNs has been used to successfully perform MAR [14]. However, targeting residuals in the sinogram is a novel approach which we will explore during our experiments.

3.3 Network Architecture: U-Net

There are several challenges associated with choosing a deep network architecture for MAR. Of primary interest is achieving low generalization error. Networks like GPT-3 [31] which are highly capable of generalization make use of dozens of layers and billions of learnable parameters. During development we found that simply adding more layers the popular Fully Convolutional Network (FCN) architecture did not always reduce generalization error, and sometimes even increased it. Thus there is motivation in exploring deep network models with more parameters.

This leads us to consider the U-Net architecture [19, 1]. The inaugural U-Net contained over 31 million learnable parameters. This, combined with its 25 hidden layers, helped it to achieve the least generalization error in multiple 2015 cell-tracking competitions [19]. We will make use of this network's strengths by modifying it to suit our MAR application. Before discussing our modifications, we describe existing features of the architecture.

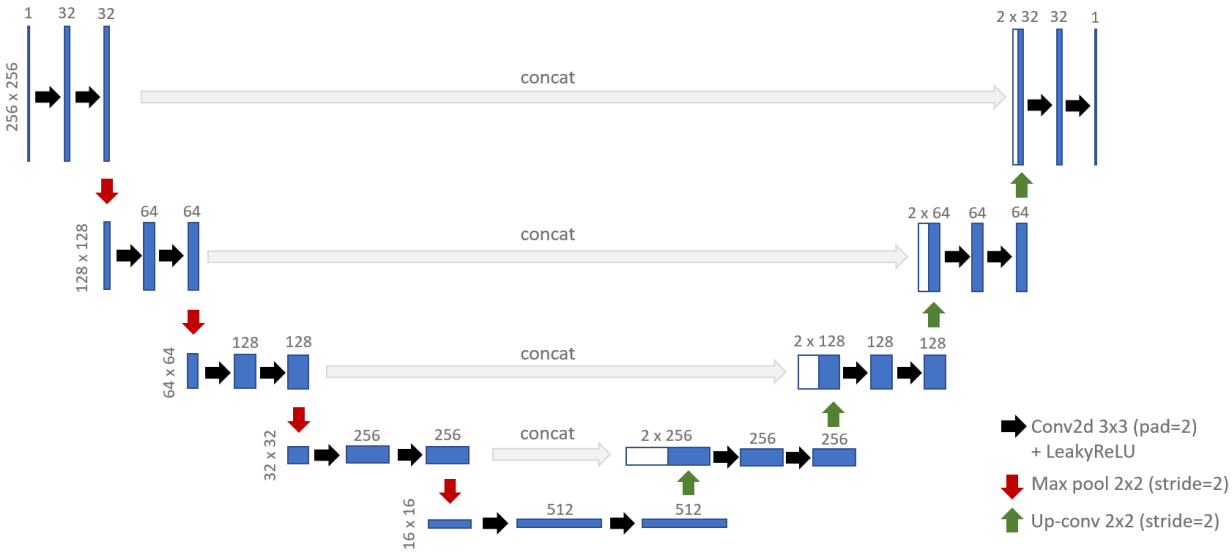

Figure 3.3: A sample U-Net architecture which we use to perform MAR.

U-Net gets its name from its shape in network diagrams (Figure 3.3). The network consists of five "levels" which an input descends and then ascends before yielding an output. In our diagram, blue rectangles represent intermediate activations between network layers. Vertical text on the far-left side of the diagram denote the shape of intermediate activations, while horizontal text on top of activations denotes the number of channels. For hidden

layers (any non-input/output layers to the network) the number of channels reflects the number of convolution kernels in the preceding layer. For the input and output layers, having one channel reflects that the images are in grayscale.

Upon being passed as an input, images pass through two convolutional layers (black rightward arrows) consisting of many 3×3 kernels and an activation function. To descend a level in the "U", activations are subjected to 2D Max Pooling (red downward arrows). This down-samples the intermediate activations while preserving the most intense signals. After down-sampling, images are passed through two new convolutional layers having twice the kernels as the previous level.

Ascension of the "U" takes three steps per level. First, an up-convolution halves the number of kernels while doubling the activation's height and width (green upward arrows). Up-convolution is an up-sampling technique whose parameters are learned during training. Next, activations from earlier in the network are concatenated to the up-convolved outputs. Earlier activations are represented as white squares in Figure 3.3. Such concatenations been shown to regularize the loss landscape[1] of neural networks, expediting training and reducing generalization error [32]. Lastly we apply two more convolutional layers and conclude our calculations for the level.

3.3.1 Modifications to U-Net

A direct implementation of U-Net [19] does not work well for our problem; we will discuss each of our changes and their significance. Firstly, U-Net was originally designed for segmentation problems. This required a terminal 1x1 convolution layer and a normalized softmax [33] activation function to map pixels to their probability of being in segmentation classes. Because our task is MAR as opposed to segmentation, we remove both the 1x1 convolution layer and output normalization from the end of U-Net.

Another change we make is the substitution of ReLU activations for LeakyReLU activations. While ReLU activations map negative inputs to zero, LeakyReLU instead multiplies them by a small factor α (Equation (3.3)).

$$\text{ReLU}(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad \text{LeakyReLU}_\alpha(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \tag{3.3}$$

LeakyReLU is often used to solve the *dead neuron problem* whereby neurons with consistently negative pre-activation values cease to contribute features in learning. LeakyReLU

[1] Loss landscapes specify expected error for a particular choice of θ. See Section 2.3.3.

activations are commonly used in MAR networks [3, 15] and we found that their use reduced overfitting in our network (Figure 3.4).

However, sometimes we observed a phenomenon whereby massive explosions in training loss caused the network to exhibit worse performance than when it was randomly initialized (Figure 3.5). We describe this phenomenon as a failed training attempt. Failure was closely related to using values of $\alpha > 0.05$. In this sense, increasing α can be seen as a tradeoff between reducing overfitting while risking catastrophic explosions in loss during training.

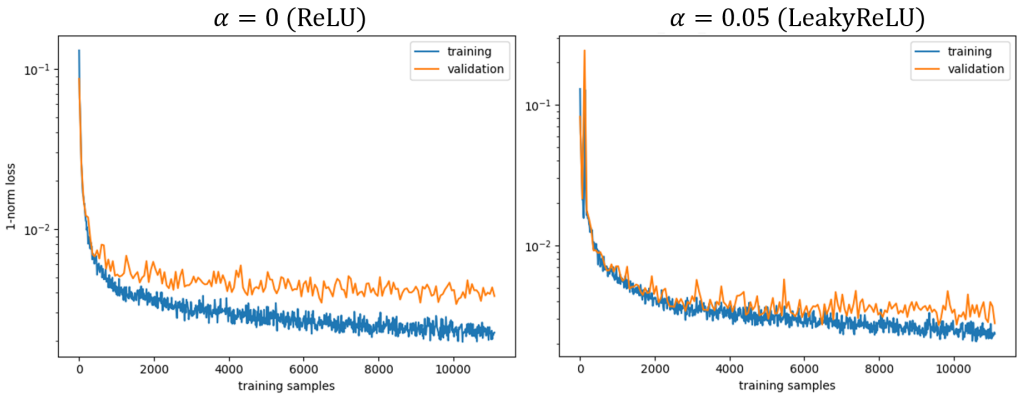

Figure 3.4: LeakyReLU reduces overfitting in our architecture. Overfitting is characterized by diverging training and validation error during learning (Section 2.3.3).

When training with $\alpha = 0.05$ we found that the residual network failed during 25% of training attempts while the direct network failed during 50% of attempts. To account for this we reverted to using ReLU activations for the direct model. We also use larger mini-batch sizes for the direct model during experiments. The combination of these adjustments made the probability of failure around 25% for both networks.

An additional modification we make is the introduction of padding (Section 2.3.4) in convolutional layers. In the absence of padding, U-Net [19] requires a strict input shape and the use of cropping pre-concatenation so that the shapes of intermediate activations remained integral and concatenated activations have like shapes. By using padding we maintain a constant activation shape along each level of the "U". This allows the network to be used on images with side lengths of 2^M for any M greater than the number of U-Net levels. We make use of this feature in our experiments when we train the network using down-scaled images.

Lastly, we reduce the number of kernels in each convolutional layer by a factor of 2 from [19]. This means in the first level we use 32 instead of 64, 64 instead of 128 at the second

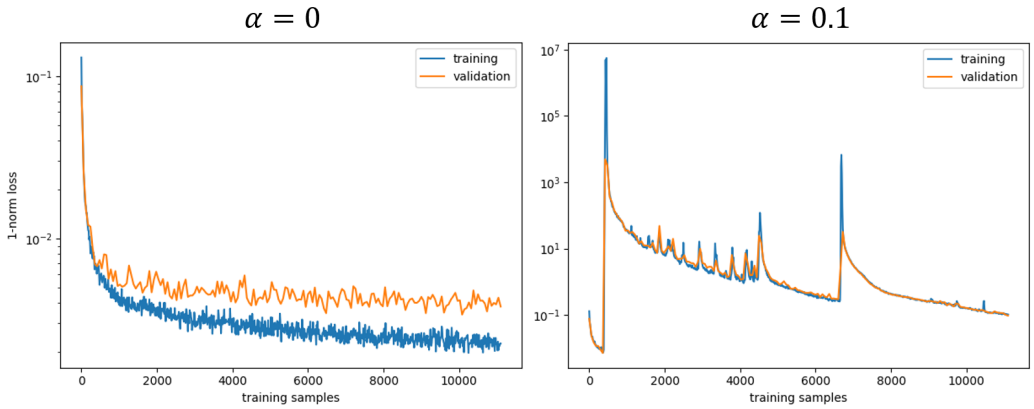

Figure 3.5: Using LeakyReLU parameters beyond 0.05 resulted in worse performance than the random initial network.

level, etc. This choice was made to expedite training while still preserving a relatively high number of learnable parameters; even with this reduction, our modified network contains 7.8 million parameters as opposed to the 180 thousand of [3, 15].

3.4 Training Data

A MAR researcher's ideal dataset contains artifact-affected and artifact-free scans of patients in identical positions. This can be partially achieved by performing two CT scans in the same position with different x-ray emission spectra (Figure 3.6). However multiple scans and higher scan energies increase the effective radiation dose to the patient, which is a small but statistically significant risk factor in carcinogenesis [34]. As such extensive MAR datasets do not exist and metal artifacts must be simulated.

Simulating metal artifacts involves recreating the insertion of prostheses into metal-free scans and then generating artifacts. This technique allows a precise knowledge of the position and composition of metal objects. Additionally, artifact-free CT data is readily accessible via online anonymized datasets [35], does not require CT hardware or operation, and does not expose patients to excess radiation. One downside of artifact simulation is that generated artifacts may not accurately reflect physical artifacts, leading to poor generalization on real-world tasks.

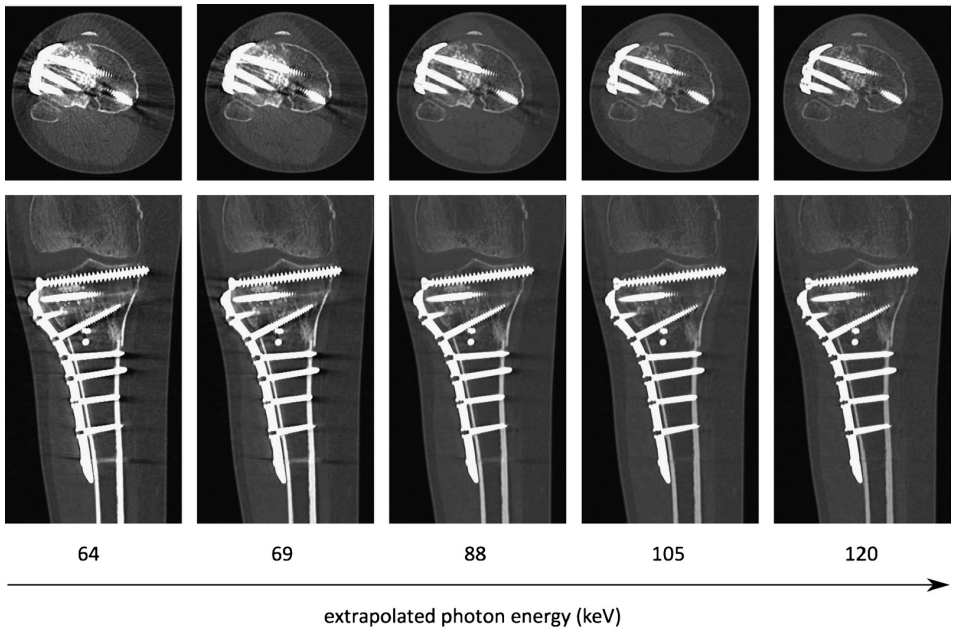

Figure 3.6: Metal artifacts are reduced as the potential across the x-ray tube increases. Reprinted from [4] with permission from Wolters Kluwer.

3.4.1 Prostheses

We generate datasets of patients with dual hip prostheses. This represents a challenging use case for MAR as large amounts of metal produce more severe artifacts [9]. Axial sections of hip prostheses are modeled as circles of slightly smaller diameter than patients' femurs. Their material is assumed to be ASTM-F75 standard Cobalt-Chromium alloy, a common material used in orthopedic implants. The alloy's composition is approximately 70% cobalt and 30% chromium by mass, and it has a density of $\rho = 8.4$ g/cm^3. The mass-attenuation coefficient is approximated as the weighted sum of cobalt and chromium's respective mass attenuation coefficients, weighted by their atomic ratios in the alloy. Therefore the linear attenuation coefficient of the alloy is

$$\mu = \rho \times (m_{\mathrm{Co}}\omega_{\mathrm{Co}} + m_{\mathrm{Cr}}\omega_{\mathrm{Cr}}) \tag{3.4}$$

by Equation (2.2), where ω_k is the fraction of atoms which are of element k. Mass attenuation data is accessed through [2]. At 100 keV this results in $\mu_{\mathrm{metal}} = 3.102$. As opposed to commonly used Titanium implants which has a μ value of 1.632, ASTM-F75 has a relatively high attenuation value. This increases the overall severity of artifacts.

3.4.2 Generation

We generate realistic-looking streaking and deletion artifacts by manipulating projection values in the sinogram (Figure 3.8). We begin with a CT image in cm^{-1}, i.e. an image converted from HU to linear attenuation units at an equivalent monochromatic energy of 100 keV (Equation (2.3)).

First, a mask image of two circles is generated. These circles are uniformly selected from a circle around the femur and they represent the location of implants in the patient. Note that this can result in metal overlapping with or totally outside of the femur. We consider this to be a form of augmentation, although during testing we position metal strictly within the femur to more faithfully simulate real implants.

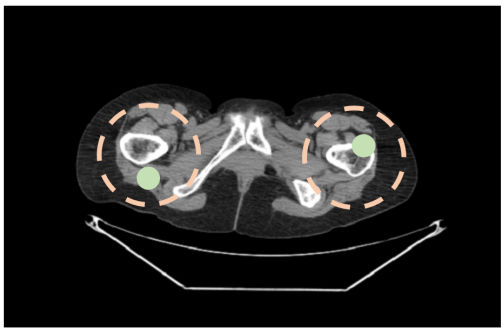

Figure 3.7: Training images have implants which are distributed uniformly around the femur. This results in some implants exterior to or overlapping with bones.

Next, the hip replacements are "inserted" into the patient by replacing all pixels indexed by the metal mask with the value of μ_{metal} (Section 3.4.1). The resulting image is the "clean" image and represents the ground truth during training. Additionally we multiply the metal mask by μ_{metal} to get an isolated image of the metal. The clean and metal images are then converted into projection data via the Radon transform.

The final step (denoted $\circledast$ in Figure 3.8) selects all pixels in the metal sinogram above a threshold to create a sinogram mask. This threshold controls the number of sinogram pixels affected in artifact generation. The sinogram mask is then used to select pixels in the clean sinogram, which are set to the maximum pixel value in the clean sinogram. This effectively simulates an increased attenuation through metal caused by the large attenuation gaps between materials at low x-ray energies (Figure 2.3). After applying the inverse Radon transform, the result is realistic-looking metal artifacts from dual hip prostheses (Figure 3.9).

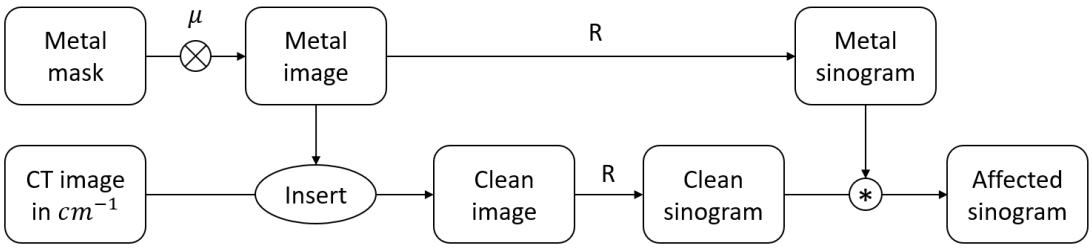

Figure 3.8: Artifact generation process. R denotes the discrete Radon transform.

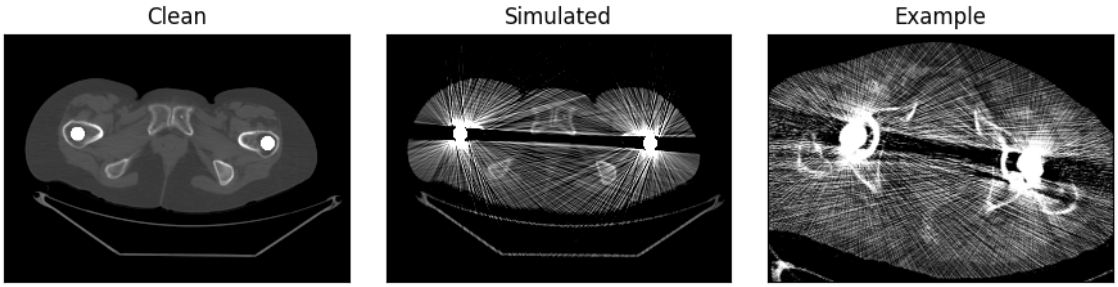

Figure 3.9: Result of the artifact generation process, with real metal artifacts for comparison. In the clean and simulated image the viewing window is set to [-500, 1200] HU.

As opposed to [16, 1], this technique does not require computationally intensive polychromatic CT simulations and manages to produce realistic-looking artifacts (Figure 3.9). However our technique does not simulate scatter, and so the manipulated projection values are localized entirely in narrow bands (Figure 3.10).

3.4.3 Augmentation and Standardization

Data augmentation is the procedure of manipulating data to artificially increase the size of the dataset. It is important to augment data in such a way that all inputs to the network could feasibly be observed in test data. Good augmentation strategies can produce significant performance gains and have the potential to greatly increase the size of a dataset [36]. Common procedures include vertically or horizontally flipping images, warping, and combining training samples.

We apply data augmentation at two main phases. The first phase is during data gen-

Figure 3.10: Simulated image with artifacts, as well as $\boldsymbol{\epsilon}$, and $R^{-1}(\boldsymbol{\epsilon})$.

eration. We apply a mild zoom-out transformation to both the metal image and clean image as well as randomly position the metal implants (Figure 3.7). The zoom-out transformation simulates a resizing and stretching of patients in the dataset. We achieve this by cropping a region around the image that is 0% to 10% larger than the original and that has an aspect ratio of 90% to 110% width/height (Figure 3.11). This region is then resized to a 256px square image, resulting in a random stretching and shrinking of patient scans.

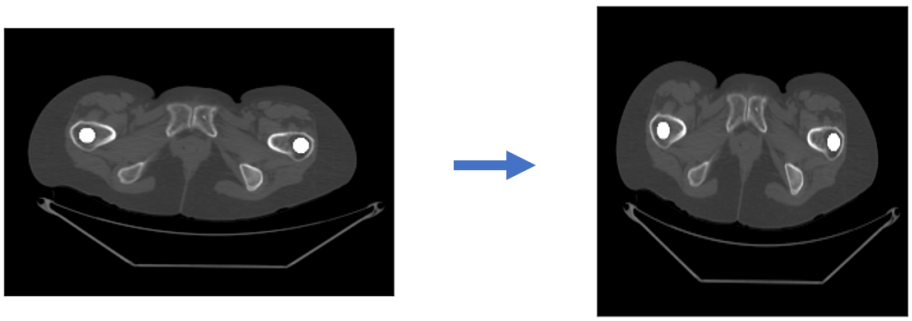

Figure 3.11: Stretching and cropping is performed to simulate patients with different body shapes. This image exaggerates the technique for demonstration purposes.

The second data augmentation phase is performed at training time. Before sinograms are passed to the network as input, they are flipped horizontally with a probability of 50%. This is equivalent to a horizontal flip of the patient and effectively doubles the size of the dataset. While horizontal flips may yield unrealistic data due to asymmetry in the human body, we desire a MAR network which is agnostic towards the anatomical realism of its

inputs. A MAR algorithm should correct the physical phenomena causing artifacts; simply memorizing human anatomy could lead to the inpainting of false data and may obscure diagnostic features like tumors, inflammation, or fractured implants.

Standardization is the process of transforming input data to have mean zero and standard deviation one. It is commonly used by AI practitioners to improve neural networks performance [37]. Standardizing images is performed by subtracting the mean pixel value then dividing by the standard deviation (with mean and standard deviation referring to the distribution of pixels in the dataset). We apply standardization to both the inputs and training targets, then multiply the network's output by the standard deviation of pixels and add the mean pixel value to produce a final image.

Chapter 4

Experimental Results

In this chapter we perform three experiments to explore the utility of our proposed techniques. We compare the results of our proposed methods against the Linear Interpolation method [12] (Section 2.2.2) and a CNN-based inpainting technique [3]. In each experiment we use clinical CT data obtained from [38] and simulate dual hip prostheses and metal artifacts (Section 3.4.2). CT images are downsized from 512 by 512 to 256 by 256 to expedite computation.

In Experiment 1 we train and validate using generated images from the same patient. The objective is to observe if our proposed methods can learn to correct a narrow distribution of metal artifacts. In Experiment 2 we use the same data as Experiment 1, but isolate the upper femoral region for validation testing. This simulates a use case where our network is trained to remove metal artifacts in a patient using the metal-free regions of the patient's body. In a final experiment we test the ability of our proposed methods to generalize across patients, a requirement for our MAR technique to be considered practical in real-world use.

4.1 Experimental Setup

We train our networks to correct metal artifacts from dual hip prostheses. These tests do not explore the most general use case of correcting artifacts from arbitrary metal objects. As none of the scans in our original dataset were of patients with hip replacements, we simulate metal artifacts using the techniques outlined in Section 3.4.2.

De-identified scans without metal are accessed via the Cancer Imaging Archive (TCIA) [35], specifically the Head and Neck Cancer CT Atlas [38]. The 98 GB dataset contains 885 CT scans from 215 patients for a total of 159,776 CT images. While all scans contain the head and neck of cancer patients, many scans contain femoral images which we use for generating data.

Networks for our proposed methods are each trained for one epoch over their respective training sets which are described before each experiment. We employ the ADAM optimizer for all but the last 30 iterations during which SWA is used with sampling period of 6. The "residual" and "direct" networks are trained to output residuals and corrected sinograms, respectively. Mini-batch sizes of 12 and 16 are used for the training of "residual" and "direct" networks.

4.1.1 Performance Metrics

We employ multiple metrics to measure numerical and perceived error in corrected CT images. To reiterate, performance is measured on the final CT image and not in the sinograms, even though loss during training is measured on sinograms. To quantify numerical error we use the mean average error (MAE) and root mean square error (RMSE) (Equations (2.7, 2.6)). To approximate perceived error we use the *Structural Similarity Index Measure* (SSIM) [39]. The SSIM compares luminance, contrast, and structure in multiple image patches to quantify perceived error. The SSIM takes values between 0 and 1, where SSIM $= 1$ iff the input images are equal. SSIM has been shown to more accurately measure perceived error than MSE [40].

We also present visual depictions of each method in each experiment. When a CT scan is shown, the CT viewing window is set to $[-700, 1500]$ HU (see Section 2).

Even using identical metrics, published errors in MAR research may not be immediately comparable because of choices in units (HU or cm^{-1}) or equivalent monochromatic energy when converting HU to cm^{-1} (Equation (2.3)). To further expound performance, we also provide errors relative to the error of the image with artifacts. If the artifact-free and metal-affected images are f and f' respectively, l is a loss function like RMSE or MAE, and the MAR technique output is $g(f)$, then relative error is given by

$$\frac{l(g(f'), f)}{l(f', f)} \times 100\%$$

This unitless metric allows for a more meaningful comparison of results between different published MAR methods.

4.1.2 Performance Comparisons

We compare our proposed methods against two other MAR techniques, the first being the LI algorithm (Section 2.2.2) and the second being a CNN-based inpainting method.

In implementing the LI algorithm we must consider the important detail of highlighting metal pixels in the metal-affected image, which is a necessary step in the algorithm. It is impractical to manually highlight pixels in all of our test images. Additionally, human variance in the highlighting procedure would introduce variance in performance which is impossible to reproduce. However, simply using the known metal mask to highlight metal pixels is an unrealistically accurate application of the LI algorithm.

Recall that the "operator mask" is a Boolean mask of the human-highlighted pixels. Let "mask" be a Boolean image indicating which pixels contain metal in the ground truth image. To simulate operator highlighting we define the operator mask pixel-wise as follows:

$$\text{operator mask}(x, y) = \begin{cases} 1, & \exists \Delta x, \Delta y \in \{-1, 0, 1\} : \text{mask}(x + \Delta x, y + \Delta y) = 1 \\ 0, & \text{otherwise} \end{cases}$$

The resulting operator mask is simply the ground-truth mask after being grown by 1 pixel in all directions. This simulates highlighting being performed by a skilled but still imperfect CT operator, which is the intended use case for the algorithm.

The second technique we compare against is the CNN-based method of [3], which we refer to at the QiNN method after its author. For a description of the algorithm see Section 2.3.4. The QiNN method uses a CNN to predict corrected sinogram values for x-rays which pass through metal. Information about metal location is obtained by thresholding; we designate any pixel with $x_{i,j} \geq \mu_{\text{metal}}$ as being made of metal.

We train the QiNN method using the same datasets and number of epochs as the proposed techniques, and use a mini-batch size of 12. The QiNN network is optimized using ADAM with MSE loss as per the original implementation. While the training time for U-Net is longer on average, the application of the proposed methods is faster than the QiNN method because it requires Radon and inverse Radon transforms required to isolate the sinogram trace of metal.

4.2 Experiment 1: Single Patient, Random Validation Samples

We explore if our proposed methods can correct metal artifacts at different positions in the same patient. This will validate if the proposed methods can correct artifacts within a low-variance distribution of scans. Variance of inputs is low because we are training using only a single patient's CT images with only one type of metal implants. Figure 4.1 shows a typical sample from the training set after applying SART (Section 2.1.2).

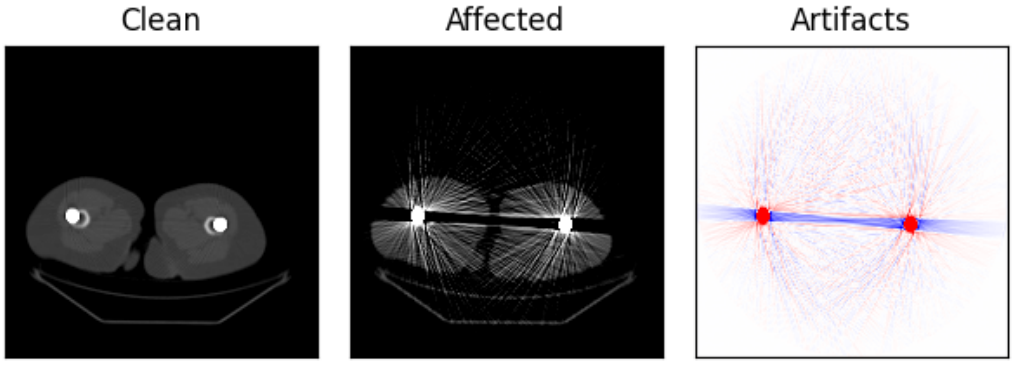

Figure 4.1: Reconstructed metal-free and metal-affected scans from the single-patient dataset, and a spatial depiction of metal artifacts. Training occurs on the Radon transform of these images.

The "clean" image's projection data is ground truth during training while the "affected" image's projection data forms the input to the network. Training targets are either the residual projection errors or the clean projection data as specified in Section 3.2.

The patient scan used for data generation contains 49 CT images, which are each subjected to augmentation to produce 204 unique training samples. The training set consists of 8000 samples with ASTM-F75 hip prostheses placed in or near to the femur. The validation set of 2000 images is used to monitor overfitting during training. Finally, a test set of 20 new samples from different slices of the patient is generated with implants strictly overlapping the femur. Each method is tested and average results are reported.

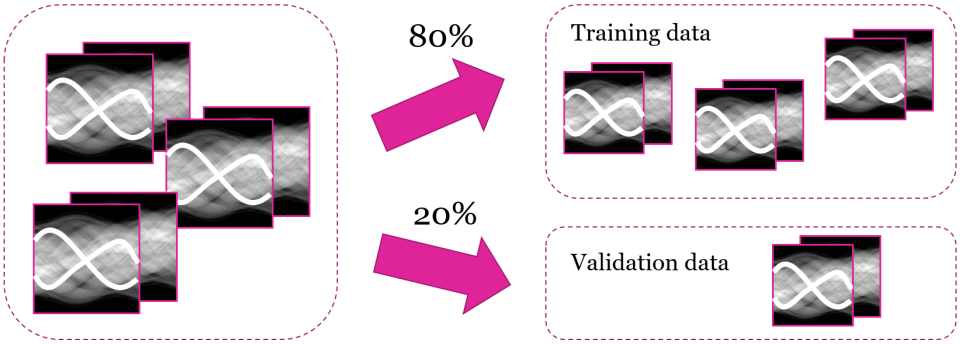

Figure 4.2: Training and validation samples form a random partition of the dataset.

4.2.1 Results

Figure 4.3 demonstrates method performances, with one row for each MAR technique and three columns for random samples from the dataset. Table 4.1 and Table 4.2 quantify the performance of each method on the test set of 20 images.

Table 4.1: Average performance of each method on the Experiment 1 test set. Error is measured in cm^{-1}.

Metric	No MAR	LI	QiNN	Residual	Direct
MAE	1.08e-03	3.03e-04	**7.32e-05**	8.48e-05	8.82e-05
RMSE	7.78e-04	2.51e-04	**2.47e-05**	3.43e-05	3.49e-05
SSIM	0.590	0.947	**0.960**	0.944	0.944

Table 4.2: Average relative error of each method on the Experiment 1 test set. Relative errors are unitless.

Metric	No MAR	LI	QiNN	Residual	Direct
Relative MAE	100%	28.1%	**6.79%**	7.86%	7.86%
Relative RMSE	100%	32.33%	**3.18%**	4.41%	4.41%

Looking at the qualitative images in Figure 4.3, both the proposed residual and direct models have residual dark streaks in their outputs. The proposed techniques significantly reduce the bright lines emanating from the implants. For the QiNN method the most noticeable remaining artifacts are short bright and dark lines emanating from the implants

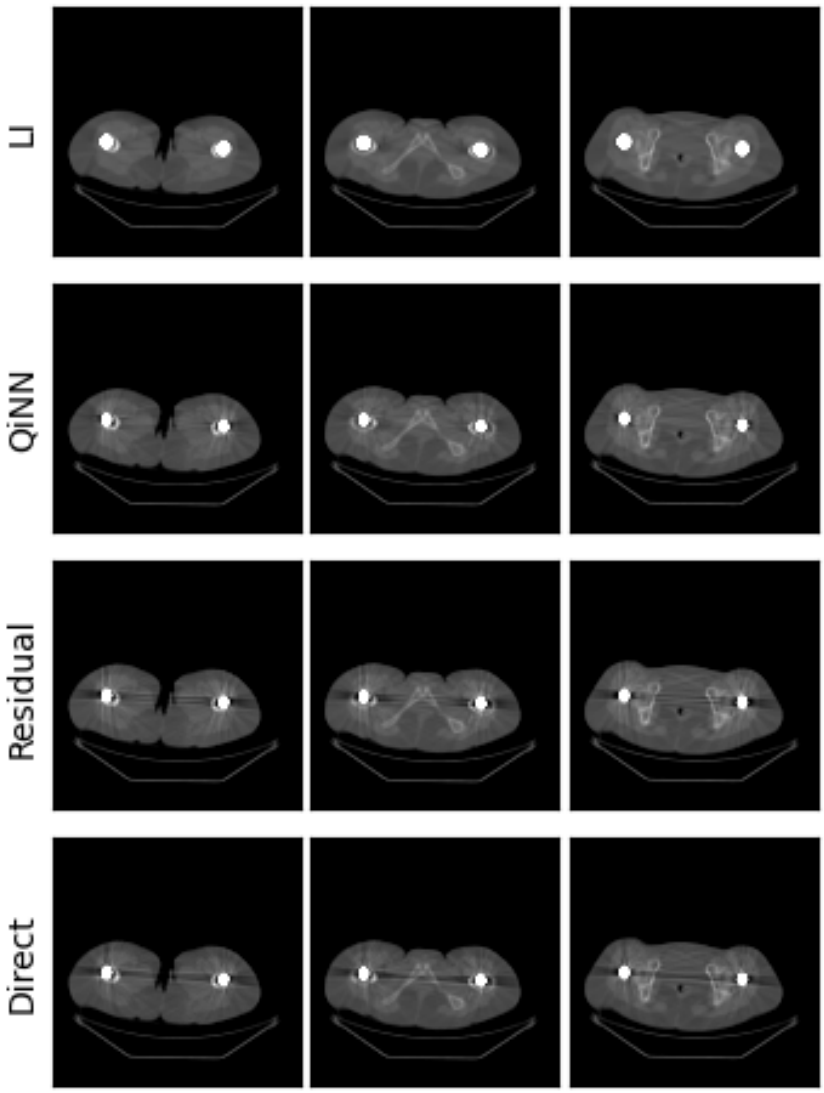

Figure 4.3: Each column contains a random test sample from Experiment 1 and each row displaying a different method's output on those samples.

in all directions, while the dark deletion streak between the prostheses is no longer noticeable. The LI method exhibits blurring, and inaccuracy in the replaced metal region creates

noticeable inaccuracy compared to the other methods. However the dark deletion streak is almost entirely removed.

We find that the proposed methods are able to significantly reduce metal artifacts in a single patient, although they are certainly not eliminated. Quantitative results confirm that the qualitative observation that proposed techniques are outperformed by the baseline CNN method in this experiment. Additionally, the LI method is superior at reducing metal artifacts although the blur and mislabeling of metal pixels leads it to have the highest residual error of the techniques. The residual and direct methods perform nearly identically in this experiment.

This simple experiment demonstrates that our methods can reduce a narrow distribution of metal artifacts, although they are both outperformed visually by both baseline techniques. Over the upcoming experiments we will explore if our models can generalize to increasingly dissimilar test distributions.

4.3 Experiment 2: Single Patient, Withheld Femoral Data

Experiment 1 demonstrates that our methods are capable of somewhat reducing artifacts within a narrow distribution of similar training and testing data. Experiment 2 will determine if our methods can extrapolate their learning from patient images to correct artifacts in unseen regions in the same patient. This makes the upcoming experiment a better test of generalizability than Experiment 1.

Consider how one might implement our proposed methods using data from only a single patient. To perform supervised learning we require access to ground truth images, precluding the use of metal-affected regions during training. Therefore we must train our networks using artifacts that we simulate *outside* of the metal-containing region of the patient. Using the single-patient dataset from Experiment 1, we reserve all images from the upper-femur as validation samples and simulate artifacts near the hip region for training. Figure 4.4 displays our data model in this experiment.

Testing takes place on the 16 CT images from the upper-femoral region, where real-world hip replacements reside. Test images are not subject to any augmentation and metal implants are placed directly within or over top of the femur. Average performance over the test set is reported in tables.

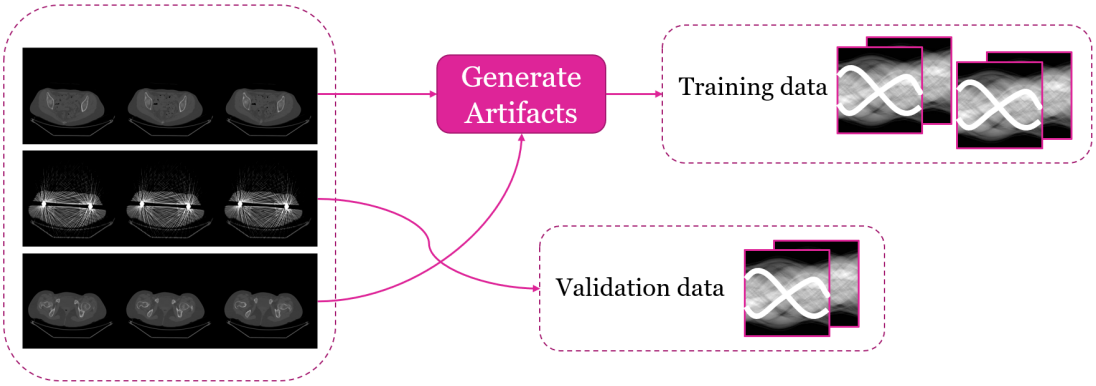

Figure 4.4: Artifacts will be simulated in metal-free body regions to access ground truth during training. Testing will be performed on previously unseen CT slices.

4.3.1 Results

Figure 4.5 demonstrates each method's performance on three random samples, with each column for each sample and one row for each method. Tables 4.3 and 4.4 display the average performance of each method on the test set.

Table 4.3: Average performance of each method on the Experiment 2 test set. Error is measured in cm^{-1}.

Metric	No MAR	LI	QiNN	Residual	Direct
MAE	1.07e-03	3.03e-04	1.08e-04	**8.00e-05**	8.65e-05
RMSE	7.72e-04	2.53e-04	4.12e-05	**3.03e-05**	4.25e-05
SSIM	0.593	0.945	**0.956**	0.954	0.929

Table 4.4: Average relative error of each method on the Experiment 2 test set. Relative errors are unitless.

Metric	No MAR	LI	QiNN	Residual	Direct
Relative MAE	100%	28.29%	10.05%	**7.46%**	8.07%
Relative RMSE	100%	32.77%	5.34%	**3.93%**	5.50%

Looking at Figure 4.5, the performance of the LI method appears consistent with Experiment 1. This makes sense because LI is a deterministic algorithm whose performance

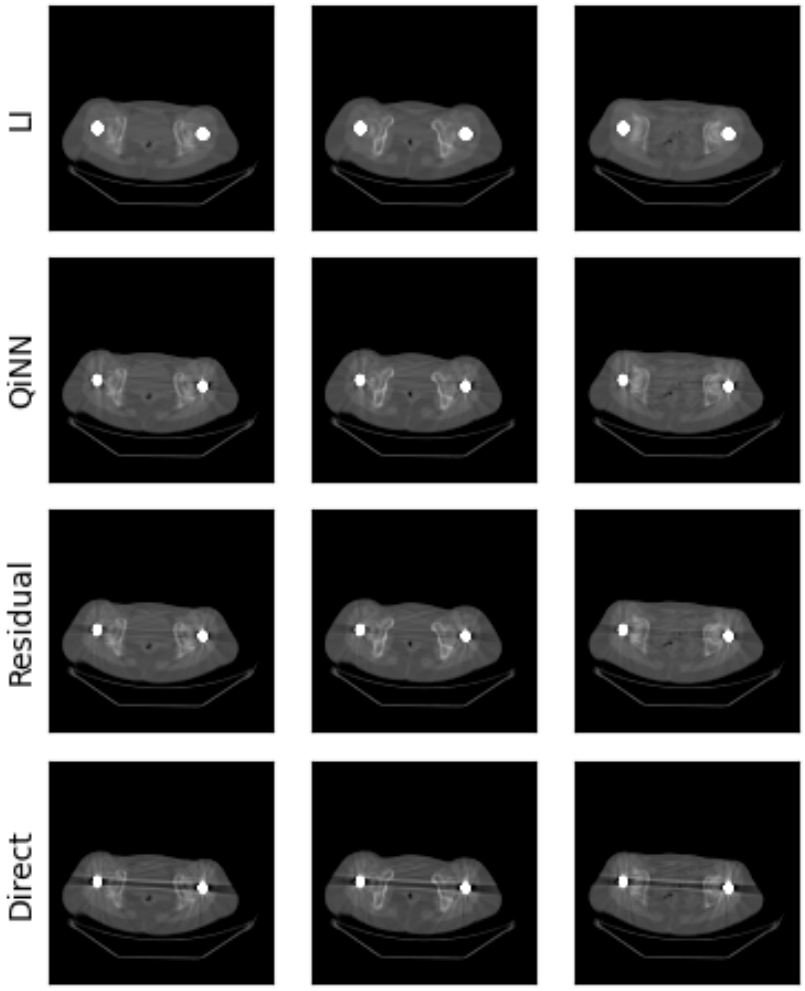

Figure 4.5: Each column contains a random test sample from Experiment 2 and each row displaying a different method's output on those samples.

is not influenced by any training program. This is confirmed by the quantitative results in Table 4.3, where the MAE, RMSE, and SSIM all lie within ±10% of their values from Experiment 1. The QiNN outputs look visually similar in quality to Experiment 1 although there is some degradation in quantitative performance as the MAE and RMSE both increase by about 50% from Experiment 1 (Table 4.3). The QiNN SSIM scores are within

1% of one another between experiments.

Looking at Figure 4.5 in isolation may make the numerical results in Tables 4.3 and 4.4 confusing; although it appears that LI and QiNN clearly reduce artifacts more effectively, they exhibit greater MAE and RMSE than the residual network. We offer an explanation for this phenomenon: windowing (see Figure 2.2). Figure 4.5 displays CT images that have been windowed between $[-700, 1500]$ HU while the MAE and RMSE are calculated on the exact outputs of the methods, without any windowing. This means that some of the errors created by the LI and QiNN method may not be visible inside the displayed ranges of images in our report. Additionally, blur effects such as those introduced by LI are difficult to observe in down-scaled report images.

The performance of the proposed methods are notably different in this experiment. For the residual model, visual artifacts appear more reduced than in the first experiment, while the opposite is true for the direct model. This is reflected quantitatively by the fact that the SSIM score of the residual method increases from 0.944 to 0.954 while the direct method decreases from 0.944 to 0.929. The improvement of the residual method from Experiment 1 was not expected, and it demonstrates that targeting sinogram residuals is a feasible representation for learning in MAR.

4.4 Experiment 3: Generalization Across Patients

We train all models using data from four patients and validate their performance on a fifth. Metal artifacts are simulated in and around the femur during training, while the test set restricts implants to be strictly within the femur. Figure 4.6 shows our data model for this experiment.

The four scans used for training contain 193 near-femur images, each of which are used to generate 72 samples. The training set consists of 14 thousand samples. Prostheses are modeled as ASTM-F75 circles in the axial plane. The test set consists of 38 CT images which were not used in training.

4.4.1 Results

Figure 4.7 demonstrates the output of each method on random samples from the dataset. Tables 4.5 and 4.6 display the average performance of each method over the test set.

Looking at Figure 4.7, we again note that the performance of the LI method has not changed. However, we note that the QiNN method leaves some more residual artifacts than

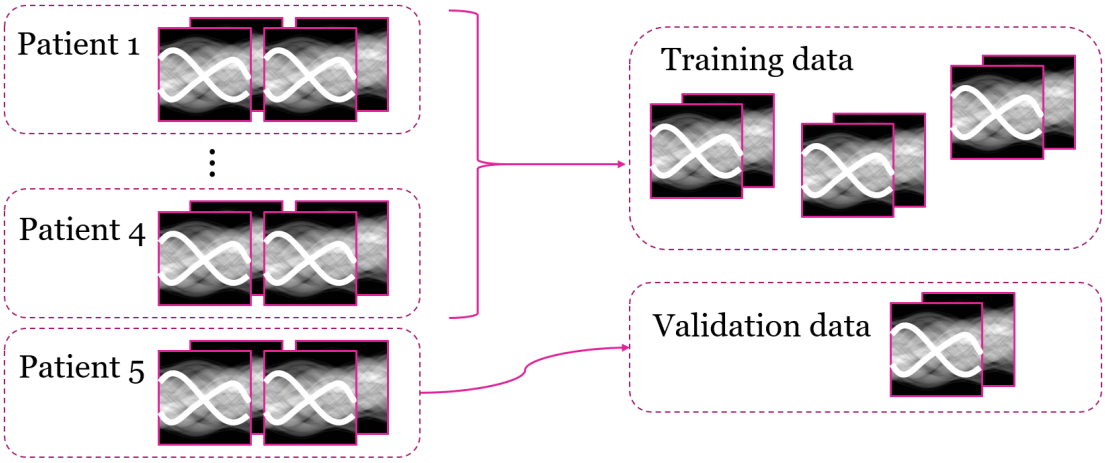

Figure 4.6: Multiple patients are used to generate a training set, while the validation and test set both draw from a patient not used in training.

Table 4.5: Performance of each method in Experiment 3. Error is measured in cm^{-1}.

Metric	No MAR	LI	QiNN	Residual	Direct
MAE	9.89e-04	2.98e-04	1.04e-04	1.47e-04	**9.65e-05**
RMSE	7.13e-04	2.65e-04	5.43e-05	7.42e-05	**4.97e-05**
SSIM	0.597	0.944	**0.947**	0.914	0.897

Table 4.6: Relative errors in Experiment 3. Relative errors are unitless.

Metric	No MAR	LI	QiNN	Residual	Direct
Relative MAE	100.00%	30.13%	10.55%	14.87%	**9.76%**
Relative RMSE	100.00%	37.19%	7.62%	10.40%	**6.96%**

in previous experiments. The residual model's visual performance decreases noticeably from the previous experiment, although interestingly the direct model performs better (comparing with Figure 4.5). However, the quantitative data suggests that the visual improvement of the direct model in Figure 4.7 is mostly due to the particular samples in the figure, as quantitatively the direct model achieves the lowest SSIM score of all the proposed methods (Table 4.5).

Again, in this experiment the MAE and RMSE values in Table 4.5 appear to be inconsistent with the SSIM scores. While the direct model leaves the most remaining artifacts,

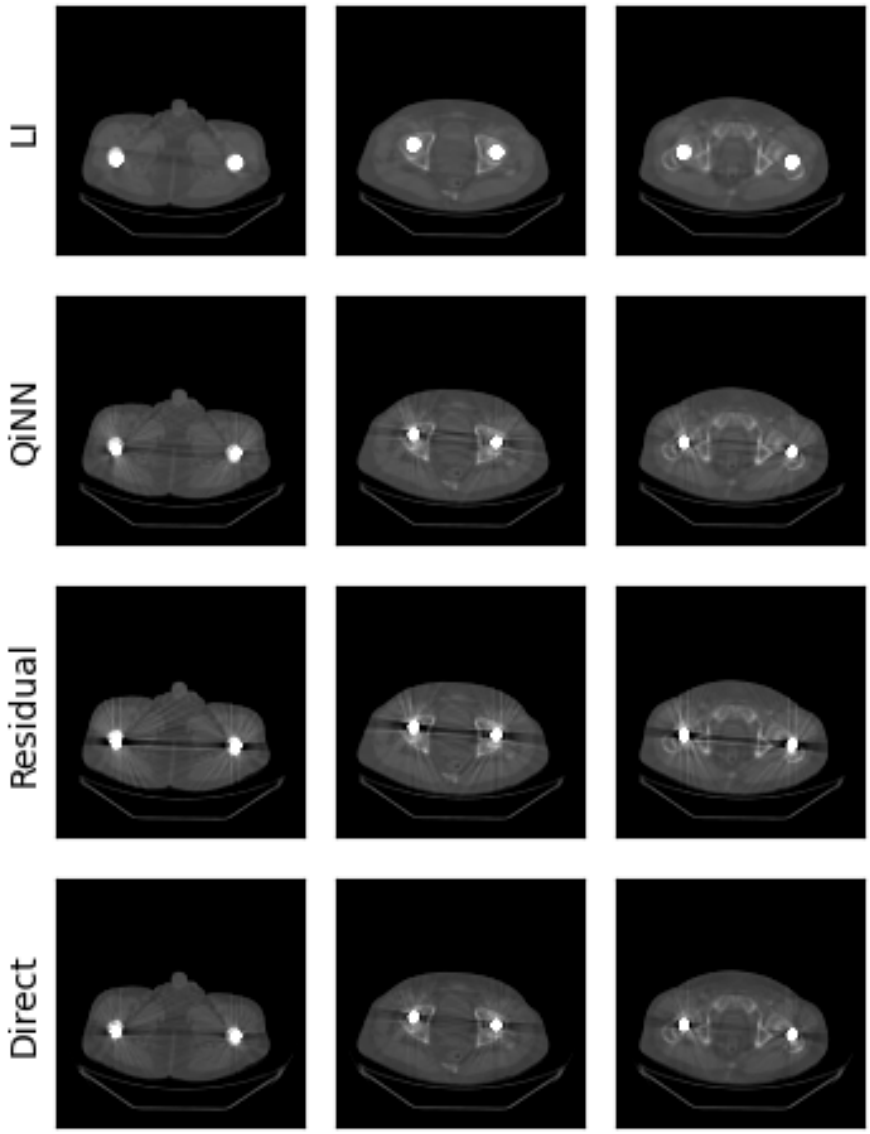

Figure 4.7: Each column contains a random test sample from Experiment 3 and each row displaying a different method's output on those samples.

it achieves some the lowest MAE and RMSE scores. We believe that, as in Experiment 2, the windowing function used to display images may occlude residual errors which are significant in the calculation of MAE and RMSE.

Both of the proposed methods exhibit decreased performance from Experiment 2. Most notably, the SSIM scores decrease from 0.954 to 0.914 for the residual method and 0.929 to 0.897 for the direct method. A decreased ability to perform artifact reduction suggests that our methods are not as capable of generalizing to new patients as the baseline methods. The fact that the QiNN method continues to outperform LI in all categories (Table 4.5) suggests that good generalization across patients is possible using CNN MAR, but that the proposed model fails to achieve it in its current state.

Chapter 5

Conclusion

We have developed U-Net architectures which reduce MAE and RMSE more effectively than linear and CNN-based projection-completion techniques when generalizing to unseen CT images. However, the proposed methods fail to reduce artifacts as effectively as the baseline methods. Our approach involves modifying U-Net for MAR which was accomplished by removing output normalization, adding padding to convolution, and integrating LeakyReLU activations into the residual network. Of the proposed methods, the best generalization across patients is achieved when the objective of the network is to output corrected sinograms directly. However, eliminating the need for knowledge of metal location increased the error of our technique. As such it is recommended to limit the correction of projection data to the trace of metal in the sinogram.

There are multiple directions in which this research can be expanded. Firstly, a network trained using a physically realistic artifact model would be more readily transferable to clinical use. There is no guarantee that our artificial artifact model accurately depicts real metal artifacts; modifying projection data using physical principles would potentially result in a network that can generalize to clinical CT images. Additionally, it would be of great utility to train a network to correct arbitrary metal artifacts. Such a network would need too be trained on a broad dataset of metal objects, materials, and patient images to learn MAR in a more general context. Finally, further exploration of architecture design choices could help to improve performance. Potentially useful modifications include the incorporation of larger convolution kernels and adjusting the number of U-Net levels or number of convolutional layers per level.

References

[1] Mitsuki Sakamoto, Yuta Hiasa, Yoshito Otake, Masaki Takao, Yuki Suzuki, Nobuhiko Sugano, and Yoshinobu Sato. Automated segmentation of hip and thigh muscles in metal artifact contaminated CT using CNN. In Hiroshi Fujita, Feng Lin, and Jong Hyo Kim, editors, *International Forum on Medical Imaging in Asia 2019*. SPIE, March 2019.

[2] Stephen Seltzer. XCOM-Photon Cross Sections Database, NIST Standard Reference Database 8, 1987.

[3] Qi Mai and Justin W.L. Wan. Metal artifacts reduction in CT scans using convolutional neural network with ground truth elimination. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, July 2020.

[4] Felix G. Meinel, Bernhard Bischoff, Qiaowei Zhang, Fabian Bamberg, Maximilian F. Reiser, and Thorsten R.C. Johnson. Metal artifact reduction by dual-energy computed tomography using energetic extrapolation. *Investigative Radiology*, 47(7):406–414, July 2012.

[5] Rebecca Smith-Bindman, Marilyn L. Kwan, Emily C. Marlow, Mary Kay Theis, Wesley Bolch, Stephanie Y. Cheng, Erin J. A. Bowles, James R. Duncan, Robert T. Greenlee, Lawrence H. Kushi, Jason D. Pole, Alanna K. Rahm, Natasha K. Stout, Sheila Weinmann, and Diana L. Miglioretti. Trends in use of medical imaging in US health care systems and in Ontario, Canada, 2000-2016. *JAMA*, 322(9):843, September 2019.

[6] Hilal Maradit Kremers, Dirk R Larson, Cynthia S Crowson, Walter K Kremers, Raynard E Washington, Claudia A Steiner, William A Jiranek, and Daniel J Berry. Prevalence of total hip and knee replacement in the united states. *The Journal of Bone and Joint Surgery-American Volume*, 97(17):1386–1397, September 2015.

[7] Statistics Canada. Population estimates on July 1st, by age and sex, 2017.

[8] Warren Kilby, John Sage, and Vicki Rabett. Tolerance levels for quality assurance of electron density values generated from CT in radiotherapy treatment planning. *Physics in Medicine and Biology*, 47(9):1485–1492, April 2002.

[9] Lars Gjesteby, Bruno De Man, Yannan Jin, Harald Paganetti, Joost Verburg, Drosoula Giantsoudi, and Ge Wang. Metal artifact reduction in CT: Where are we after four decades? *IEEE Access*, 4:5826–5849, 2016.

[10] Fabian Bamberg, Alexander Dierks, Konstantin Nikolaou, Maximilian F. Reiser, Christoph R. Becker, and Thorsten R. C. Johnson. Metal artifact reduction by dual energy computed tomography using monoenergetic extrapolation. *European Radiology*, 21(7):1424–1429, January 2011.

[11] Marc Kachelrieß, Oliver Watzke, and Willi A. Kalender. Generalized multi-dimensional adaptive filtering for conventional and spiral single-slice, multi-slice, and cone-beam CT. *Medical Physics*, 28(4):475–490, April 2001.

[12] Robert Hebel Willi A. Kalender and Johannes Ebersberger. Reduction of CT artifacts caused by metallic implants. *Radiology*, 164(2), 1987.

[13] Ge Wang, D.L. Snyder, J.A. O'Sullivan, and M.W. Vannier. Iterative deblurring for CT metal artifact reduction. *IEEE Transactions on Medical Imaging*, 15(5):657–664, 1996.

[14] Shiyu Xu and Hao Dang. Deep residual learning enabled metal artifact reduction in CT. In *Medical Imaging 2018: Physics of Medical Imaging*, volume 10573 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 105733O, March 2018.

[15] Muhammad Usman Ghani and W. Karl. Deep learning based sinogram correction for metal artifact reduction. *Electronic Imaging*, 2018:4721–4728, 01 2018.

[16] Yanbo Zhang and Hengyong Yu. Convolutional neural network based metal artifact reduction in x-ray computed tomography. *IEEE Transactions on Medical Imaging*, 37(6):1370–1381, June 2018.

[17] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.

[18] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5949–5958, Red Hook, NY, USA, 2017. Curran Associates Inc.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science*, pages 234–241. Springer International Publishing, 2015.

[20] Johann Radon. On the determination of functions from their integral values along certain manifolds (1917). *IEEE Transactions on Medical Imaging*, 5(4), 1986.

[21] A. H. Andersen and A. C. Kak. Simultaneous algebraic reconstruction technique (SART): A superior implementation of the art algorithm. *Ultrasonic Imaging*, 6(1):81–94, January 1984.

[22] G. H. Glover and N. J. Pelc. Nonlinear partial volume artifacts in x-ray computed tomography. *Medical Physics*, 7(3):238–248, May 1980.

[23] Scott Mayer McKinney, Marcin Sieniek, Varun Godbole, Jonathan Godwin, Natasha Antropova, Hutan Ashrafian, Trevor Back, Mary Chesus, Greg S. Corrado, Ara Darzi, Mozziyar Etemadi, Florencia Garcia-Vicente, Fiona J. Gilbert, Mark Halling-Brown, Demis Hassabis, Sunny Jansen, Alan Karthikesalingam, Christopher J. Kelly, Dominic King, Joseph R. Ledsam, David Melnick, Hormuz Mostofi, Lily Peng, Joshua Jay Reicher, Bernardino Romera-Paredes, Richard Sidebottom, Mustafa Suleyman, Daniel Tse, Kenneth C. Young, Jeffrey De Fauw, and Shravya Shetty. International evaluation of an AI system for breast cancer screening. *Nature*, 577(7788):89–94, January 2020.

[24] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016.

[25] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986.

[26] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning*

Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[27] Yann Dauphin, Harm de Vries, and Yoshua Bengio. Equilibrated adaptive learning rates for non-convex optimization. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[28] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Ricardo Silva, Amir Globerson, and Amir Globerson, editors, *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, 34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

[29] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer, 2nd edition, 2020.

[30] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, June 2018.

[31] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.

[32] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[33] Muhammad Khalid, Junaid Baber, Mumraiz Khan Kasi, Maheen Bakhtyar, Varsha Devi, and Naveed Sheikh. Empirical evaluation of activation functions in deep convolution neural network for facial expression recognition. In *2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, pages 204–207, 2020.

[34] David J. Brenner and Eric J. Hall. Computed tomography — an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, November 2007.

[35] Kenneth Clark, Bruce Vendt, Kirk Smith, John Freymann, Justin Kirby, Paul Koppel, Stephen Moore, Stanley Phillips, David Maffitt, Michael Pringle, Lawrence Tarbox, and Fred Prior. The cancer imaging archive (TCIA): Maintaining and operating a public information repository. *Journal of Digital Imaging*, 26(6):1045–1057, July 2013.

[36] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1), July 2019.

[37] Yann A. LeCun, Léon Bottou, Genevieve B. Orr, and Klaus-Robert Müller. Efficient BackProp. In *Lecture Notes in Computer Science*, pages 9–48. Springer Berlin Heidelberg, 2012.

[38] Aaron Grossberg, Abdallah Mohamed, Hesham El Halawani, William Bennett, Kirk Smith, Tracy Nolan, Sasikarn Chamchod, Michael Kantor, Theodora Browne, Katherine Hutcheson, Gary Gunn, Adam Garden, Steven Frank, David Rosenthal, John Freymann, and Clifton Fuller. Data from Head and Neck Cancer CT Atlas, 2017.

[39] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[40] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. A comprehensive evaluation of full reference image quality assessment algorithms. In *2012 19th IEEE International Conference on Image Processing*. IEEE, September 2012.