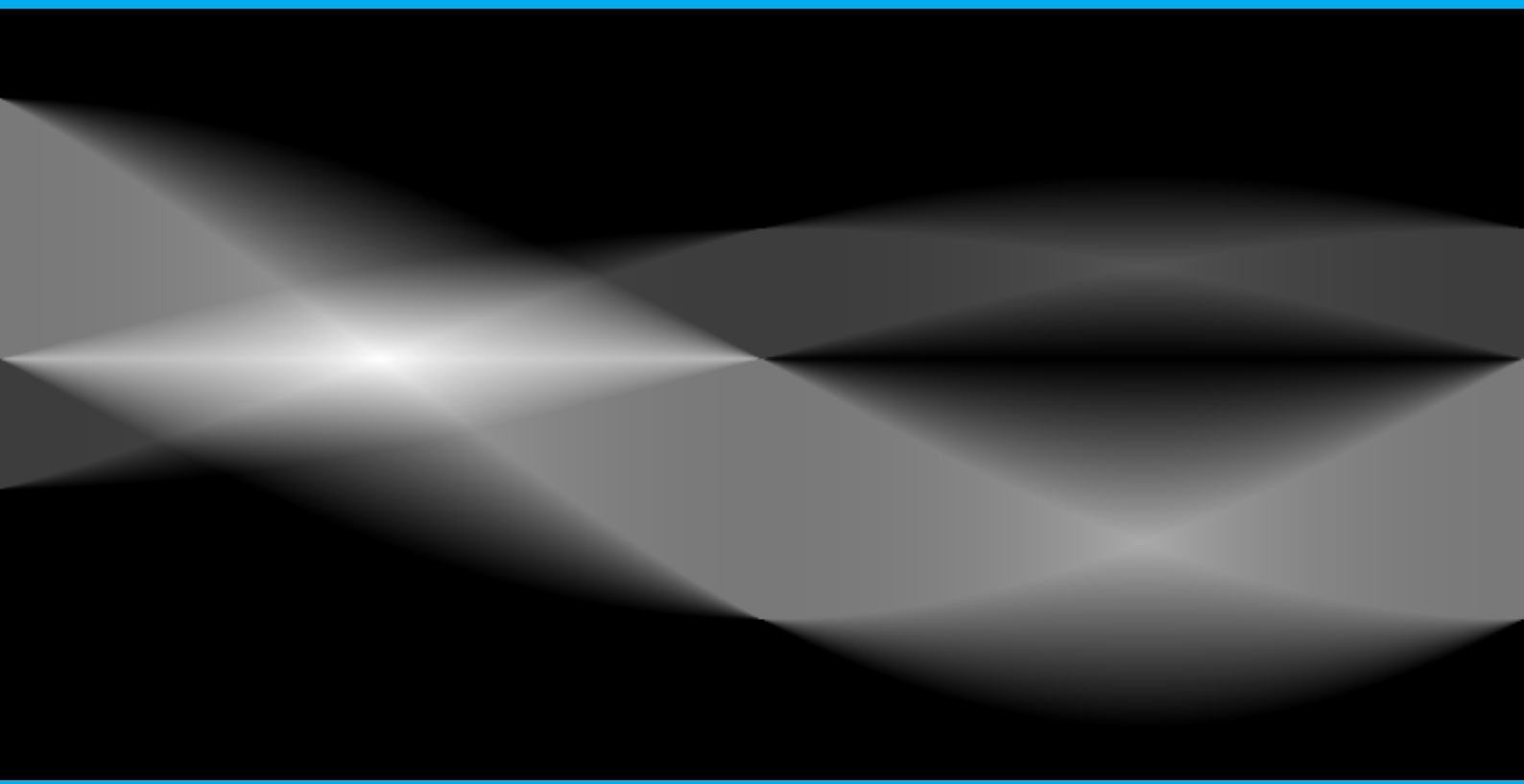
Modelling X-Ray Photon Transport Through a Transformer-Based Neural Network in Computed Tomography Forward Projection

D.T. Hoendermis



TU Delft

Modelling X-Ray Photon Transport Through a Transformer-Based Neural Network in Computed Tomography Forward Projection

Master Thesis Report

by

D.T. Hoendermis

to obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on May 22, 2023 at 15:00

Thesis committee:

Chair:	Dr. Z. Perkó
Supervisors:	Dr. Ir. M.C. Goorden
Member:	Dr. F.M. Vos
Student number:	4398327

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Faculty of Mechanical, Maritime and Materials Engineering · Delft University of Technology

Acknowledgements

I would like to thank my supervisors, Marlies Goorden en Zoltán Perkó, for their guidance and insights during the course of this project, as well as Frans Vos for taking part in my thesis comittee. I am also grateful to PhD candidate David Leibold for helping me navigate through the implementation of the data generation part of my project. I would like to thank my friends and family for their encouragement, especially Robbert, who has been an invaluable support during the past year. Finally I would like to thank Yuki for her emotional support and daily enthusiasm.

D.T. Hoendermis
Delft, May 2023

Abstract

Computed Tomography (CT) has made an undeniable contribution to global healthcare by aiding in the detection, diagnosis and monitoring of various diseases. This is especially the case for cancer, which remains one of the largest health concerns worldwide [69]. Radiotherapy is one of the main treatments for cancer and relies heavily on CT images to calculate radiation dose. With research on radiotherapy moving to adaptive treatments aiming to calculate these doses at real-time speeds while maintaining high precision, a need for accurate CT imaging at comparable real-time speeds has emerged.
Currently, the best performing CT image reconstruction methods are iterative reconstruction (IR) methods. While these methods have the ability to produce high-quality results, they suffer from slow reconstruction speed especially when incorporating computationally expensive physics-based models. Existing methods that provide low computation times are accompanied by artifacts due to the implementation of simplified approximations of physics.

Reconstruction algorithms are needed that reduce these artifacts and simultaneously improve overall computation time. Methods proposed with these goals focus mostly on separately improving the raw projection data and the reconstructed image data. By not relating the two types of data to each other, important physical processes that influence the data acquisition process are not incorporated. Methods that do focus on this transformation often rely on forward projection steps using simplified models or have very high computational loads.

Recently, the Dose Transformer Algorithm (DoTA) [47], [48] and improved DoTA (iDoTA) [49] have shown to successfully calculate radiation therapy dose by modelling particle transport in 3D with the use of a neural network. By implementing a Transformer architecture [62], DoTA is able to capture the relationship between elements in a 3D CT volume while processing it as an input sequence. This results in an accurate prediction of particle transport, while significantly reducing computation times compared to other methods.

This thesis aims to explore the possibilities of DoTA with respect to CT x-ray photon transport. A neural network that is based on the DoTA-architecture is presented. The neural network predicts projection data from a CT image, having learned to model the photon transport between the X-ray source and detector without using conventional projection operators.
The network contains a Transformer-based architecture, using causal multi-headed self-attention to process 2D CT images as a sequence of 1D lines in the direction of the X-ray beam. The ground truth data contains Monte Carlo projections of cylindrical water phantoms with cylindrical and box-shaped inserts composed of one of five materials. Input data corresponding to the phantoms is in the shape of an array containing the corresponding normalized Hounsfield values. A grid search was carried out to determine the optimal model architecture, tuning the number of Transformer heads, Transformer blocks and filters in the last convolution layer.

The model predictions are compared to the Monte Carlo projections and raytracing projections generated with Astra Toolbox [45], as well as a Two-Angle Convolution (TAC) network [11]. The average NRMSE between the projections and ground truth of the Transformer was 0.725% compared to 2.20% and 1.09% repectively for the raytracer and TAC projections.

The performance of the Transformer was tested on unseen geometries compared to the raytracer and showed the ability to predict from unseen types of geometries and intensity values, although it does not generalize well to input phantoms with a different outer shape than the cylindrical phantom used in the input. This is due to the bias in the input data and likely resolved with a more robust data set.

An IR algorithm was implemented for the reconstruction of two different phantoms. For the Transformer and raytracer, the highest achieved CNR values are similar for low-contrast regions (6.88 and

8.28 for the raytracer compared to 7.10 and 7.35 for the Transformer) as well as high-contrast regions (37.40 and 41.94 for the raytracer compared to 39.01 and 39.80 for the Transformer). Convergence rates based on low-contrast CNR are higher for the raytracer (39 and 34 iterations compared to 41 and 41 iterations for the Transformer, respectively).
Regarding beam-hardening effects, the model predictions perform significantly better than the raytracing projections on individual projections as well as image reconstructions.
Even though the Transformer algorithm does not clearly outperform the raytracer based on quantitative measures, the results of the image reconstructions are promising considering that the IR algorithm has not been tuned for use with the Transformer, suggesting that a higher performance is obtainable with adjustments such as the implementation of a different backprojector or a different value for correction factors used in the algorithm.

Limitations in prediction quality are likely related to factors outside of the model predictions, such as biases in the input data and resolution loss due to interpolation of the input data. When its prediction speed is optimised, the CT Transformer model has potential to replace conventional forward projections in IR methods, achieving Monte Carlo-level accuracy with a fraction of the computation time. Increasing the speed of the Transformer network to a higher level could assist in the development of adaptive treatment techniques and potentially improve radiotherapy treatments with higher accuracy and lower health risks.

Contents

1 Introduction

Computed Tomography (CT) has had an enormous impact on global health. It especially plays a large role in the treatment of cancer, which is among the lading causes of death worldwide [69]. Not only does it enable the early detection, diagnosis and monitoring of various types of cancer, it is also a crucial factor in radiotherapy which is one of the main types of cancer treatment. In radiotherapy, cancer cells are targeted with photon or proton beams delivering a high radiation dose to the malignant cells wile minimizing damage to the surrounding healthy tissue. Treatment often consists of multiple dose deliveries over a period of time. Due to changes in patient anatomy or the physical setup, the treatment plan for these dose deliveries need to be constantly adjusted and evaluated to ensure a correct dose delivery.
Current research is geared towards adaptive treatments, aiming to calculate dose just before or during treatment with real-time dose calculations that could even adjust for anatomy changes due to motion or intestinal movements. These highly accurate treatment plans would allow a significant reduction in radiation dose by targeting cancer cells more accurately and reducing the error margins that impact healthy tissue. To realize these types of treatments, high-speed dose calculations are needed.

With this aim, the Dose Transformer Algorithm (DoTA) was proposed in 2021 [47], which used a deep learning model to accurately predict proton dose calculations at a fraction of the speed compared to other methods. The network uses the beam energy and 3D CT volume as input, from which it predicts a proton dose distribution. The later introduced improved Dose Transformer Algorithm (iDoTA) is capable of predicting doses from much broader photon beams as well [49] in a time span that is an order of magnitude smaller than conventional approaches. The success of the DoTA alogorithm in high-speed dose calculation brings adaptive radiotherapy a step closer to reality. However, dose calculation speed is not the only bottleneck in real-time adaptive treatments. To perform dose calculations, CT images are needed to provide delineation of the patient geometry. To achieve high-speed performance without compromising on accuracy, high-quality CT image reconstruction methods are needed as well.

Currently, the best performing CT image reconstruction methods are iterative reconstruction (IR) methods, which suffer from a rapid decline in image quality with higher speed methods. IR methods implement models that take into account the geometry of the CT system and the physical processes in the image acquisition process, allowing a more accurate reconstruction of the image. These most accurate methods use Monte-Carlo based calculations which require extensive computation times, while faster methods use simplistic physics modelling which comes at the cost of a reduced accuracy. Improving the physics models in IR methods without the extra computation time could reduce the reconstruction time as well as the amount of extra processing needed.

1.1. Related Work

The research field of Deep Learning Reconstruction (DLR) has gained an increasing amount of interest in the past few years. Several DLR studies have been proposed, showing promising results of DLR algorithms in CT image reconstruction.

Most research on DLR focuses on either *indirect* or *direct* reconstruction. Indirect DLR focuses on noise- and artifact reduction by processing the raw sinogram data [60],[33], [63] or the reconstructed image separately [26], while the transformation between the sinogram and image domain is performed with conventional FBP or IR algorithms. Applying the neural network outside this transformation step simplifies the learning process as the network does not need to learn the relation between the polar coordinates of the sinogram domain and the Cartesian coordinates of the image domain. However, by not adjusting the FP and BP operations, these methods remain based on either limited physics modelling resulting in lower image quality, or intricate physics models that perform well but need a substantial amount of computation time.

Direct DLR replaces the FBP and IR steps with a neural network and reconstructs an image from sinogram data in one pass, creating the opportunity to improve the physics models compared to FBP and IR with learned weights. However, obtaining a training data set that contain images without FBP and IR artifacts is challenging, but required to prevent the model from reproducing these artifacts. Additionally, these networks often require large computational resources [74], [23]. Other networks use an *unrolled iterative algorithm*, in which an IR algorithm is transformed into a neural network, with each separate iteration expressed as a layer in the network [39], [20]. These algorithms use deep learning to mimic an IR algorithm, representing each iteration as a separate layer in the neural network. This way, the parameters used to update the image estimate are learned from training data contrary to the predefined parameters in conventional IR methods. Several of these unrolled IR algorithms have shown to improve the speed and accuracy of the image reconstruction. However, these models rely on the same approximations of the particle physics as their traditional IR counterparts, and therefore do not resolve the limitations that arise from these approximations.

Research on techniques related to the forward projection step with a neural network is sparse. One paper implemented a partially learned forward projector, using a neural network as a correction to the FP operation in an IR method [35]. Besides this, they introduced a forward-adjoint correction, in which both the FP and BP operations are corrected for by the network. This was expanded on by [57], in which a learned FP corrector was implemented with several methods including a method that used forward-adjoint correction as well. Contrary to [35], the paper showed that the learned FP corrector without adjoint correction outperformed the approach with an adjoint correction in certain cases. The learned operator in [57] also showed capable of generalising outside of the training data.

No research has been found that implemented a fully learned forward projection operator, instead of a correction factor to an existing forward projection. Such a model could predict a more accurate FP directly, removing the need for additional correction and potentially reduce the overall computation time. To achieve real-time CT image reconstruction, there is a need for reconstruction methods with a reduced reconstruction time without a reduction of quality. The key parts that influence the performance of current IR algorithms are the forward- and backprojection steps, but the majority of deep learning research focuses on noise- and artifact reducing steps.

The recently proposed iDoTA [49] has shown the capabilities of Transformer-based neural networks, successfully modelling photon transport in dose calculations at an increased speed. This research project explores the potential of the Transformer architecture applied to photon transport in CT imaging. By adapting the DoTA algorithm, the aim is to create a neural network that predicts forward projections at a Monte Carlo-level accuracy with a substantially lower speed.

1.2. Research Goals

The aim of this research is to explore the potential of the DoTA-architecture applied to photon particle transport in CT imaging. Representing the particle transport in CT image reconstruction as a sequence and modelling the trajectory of photons through material could open up new possibilities to improve CT image reconstruction, aiming for higher speeds with increased accuracy.

IR methods used for CT image reconstruction model the x-ray trajectories from the x-ray source to a detector, which are acquired at several angles and combined into a projection data set called a *sinogram*. Transforming this sinogram into an image representation is called the *forward projection* (FP). The transformation from the image back to the sinogram is called a *backprojection* (BP). Both FP and BP steps are implemented in IR algorithms. Forward projections produced with Monte Carlo (MC)-based methods are highly accurate but require large computation times. Most conventional IR methods implement raytracing techniques which are based on a simplified model of the x-ray physics and are much faster than MC methods. These methods approximate the x-ray beam to include a single energy, despite the beam containing a distribution of multiple energies. This results in artifacts related to beam-hardening or scattering which require removal with additional processing. Due to this, as well as the number of iterations needed to converge to a satisfactory image reconstruction, the reconstruction speed of these methods remains unsatisfactory.

A neural network modelling the x-ray photon particle transport could be beneficial in the prediction of forward projections with the accuracy of Monte Carlo methods, needing a fraction of the computation time. In recent years, the Transformer architecture has revolutionized the deep learning field, enabling fast networks with long-range dependencies through the use of self-attention and parallelization. Representing the particle transport in CT imaging as a sequence, modelling the trajectory of photons through material could open up new possibilities to improve CT image reconstruction.
This thesis project aims to find out whether the successful results of DoTA and iDoTA in dose calculations translate to the modelling of forward projections in CT as well. A model is created based on the DoTA-architecture, which is trained on MC data to predict a forward projection from 2D CT image input.

The research goal of this thesis project can be summarised into the following research question:

> *How suitable is a Transformer-based neural network for modelling photon transport in CT imaging?*

This research question can be divided into several sub-questions:

1. What should the data set that is needed to train the network look like?
2. What kind of network architecture and which hyperparameters will result in optimal predictions?
3. How can this model be used to improve the image reconstruction process?

This thesis report will start with a general background theory on CT and Deep Learning in chapters 2 and 3. Next, the methods used to answer the research questions are discussed in chapter 4. The results are presented in chapter 5 and discussed in chapter 6. Finally, conclusions are drawn from the presented results and recommendations for future work are given in chapter 7.

2

Basics of Computed Tomography

This chapter provides a background on the principles of computed tomography, focusing on the concepts related to this research project. A general overview of CT image reconstruction is given, before focusing on concepts related to the propagation of x-ray particles including physical processes related to attenuation and the appearance of artefacts. This is followed by a description of iterative reconstruction (IR) methods used in practice and the models that are used to represent the propagation of the x-ray beam.

2.1. X-ray physics

Computed tomography uses the transmission of penetrating electromagnetic waves known as *x-rays* to re- construct detailed internal images of the human body. While travelling through a material, an x-ray beam loses intensity depending on the specific materials it encounters. Because of this, the measured intensity of the beam exiting the body contains information about the internal structure of the patient. Collecting x-ray projection measurements from multiple angles enables the reconstruction of slices (*tomographs*) of the human body.

Undoubtedly, x-ray physics is a fundamental concept that lays the basis of CT image reconstruction. Knowledge of the way in which particle physics influences the transport of the x-ray beam is crucial in the understanding of other principles behind CT image reconstruction. X-rays are penetrating electromagnetic waves containing much higher frequencies than visible light. They are a form of *ionizing radiation*, which is a form of radiation that contains enough energy to eject an electron from an atom, resulting in a *free electron* and an *ion*. These particles can induce damage by disrupting molecular bonds in molecules, which is the reason that ionizing radiation is harmful to biological tissue.

This section describes the basic concepts of x-ray physics and the way in which they influence the CT image reconstruction process. Paragraph 2.1.1 describes the generation of x-rays and their distinguishing features, followed by a description of x-ray interactions within materials in 2.1.2 and x-ray beam attenuation, which is the underlying concept of CT reconstruction, in paragraph 2.1.3.

2.1.1. Generation of x-rays

X-rays are distinguished from other ionizing radiation such as gamma rays by their source. While gamma rays are created by the nucleus of radioactive particles, x-rays are created by two different processes that occur when high-velocity electrons hit a metal target:

- **Characteristic x-ray radiation**: the energy transfer from the high velocity electrons eject electrons in the target atoms. The resulting vacancies are filled by electrons from higher energy levels, creating a net energy loss resulting in the emission of x-ray photons with an energy equal to this energy difference. The term *characteristic* originates from this energy difference, as the value of the energy difference is characteristic for different atoms.

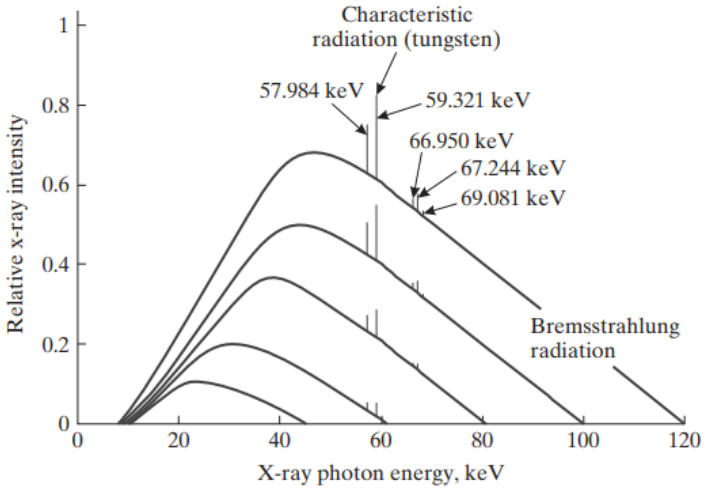


Figure 2.1: Spectrum of an x-ray beam for different energies using a tungsten anode material. The shape of the spectrum results from the continious spectrum of the bremsstrahlung and the sharp peaks of the characteristic x-rays. Figure from [52].

- **Bremsstrahlung**: this radiation comes from the interaction of a high-energy electron with the nucleus of an atom. The positive charge of the nucleus attracts the electron, bending its path. The electron loses kinetic energy which is converted into a photon which is what makes up the bremsstrahlung radiation, named after the German word for braking.

The source that is used to generate x-ray photons in CT consists of a vacuum tube with a glass cathode and solid metal anode. The cathode is heated, producing electrons which are accelerated by the tube voltage difference between the negative cathode and the positive anode. After hitting the anode, the x-ray photons are produced with characteristic and bremsstrahlung radiation, the latter of which makes up the main part of the x-ray beam. The energy spectrum is different for both types. The bremsstrahlung specturm is a continuous spectrum with its highest energy equal to the source potential, although the produced x-rays are much more likely to have lower energies. The spectrum of the characteristic x-rays consists of peaks corresponding to the specific target material. The shape of the combined spectrum produced by the x-ray source is shown in Figure 2.1, which displays the x-ray spectra for different values of the source potential. X-rays with lower energies are absorbed by the source material, resulting in the spectrum being reduced to zero.

2.1.2. X-ray interactions with materials

The photons that have been produced by the x-ray source penetrate the target material where they undergo different types of physical interactions which influence their trajectory and as a result the acquired projection data. The two main types of interactions that are important for x-ray imaging are as follows:

- **Photoelectric effect**: an x-ray photon interacts with an electron in the material, ejecting it from its orbit. The energy of the x-ray photon is absorbed fully by this electron, giving it a kinetic energy of

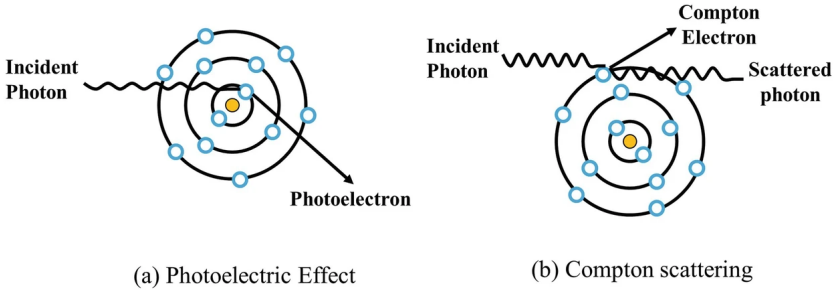
$$E_{e^-} = h\nu - E_B, \tag{2.1}$$

  where E_B is the binding energy of the ejected electron. Characteristic radiation is produced due to the filling of the vacancy left by the electron. These free electrons can have detrimental effects on biological tissue due to its ionizing nature. The photoelectric effect is illustrated in Figure 2.2a.

- **Compton scattering**: an x-ray photon ejects an electron in the outer shell of an atom which creates a new electron called a *Compton electron*. The photon loses energy to this electron and changes direction. The energy of the Compton electron is given by

$$h\nu' = \frac{h\nu}{1 + (1 - cos\theta)h\nu/(m_0c^2)}, \tag{2.2}$$

where $m_0c^2 = 511keV$ is the energy corresponding to the rest mass m_0 of an electron and θ is the angle by which the photon is scattered. The concept of Compton scattering is illustrated in Figure 2.2b.



(a) Photoelectric Effect (b) Compton scattering

Figure 2.2: In the photoelectric effect (a), an electron is ejected by an x-ray photon which converts the photon's energy into kinetic energy. In Compton scattering (b), an x-ray photon hits an atom, creating a new electron (Compton electron). Due to the loss of energy to this new electron, the photon experiences a change in direction. Figure from [59].

2.1.3. X-ray beam attenuation

While propagating through the human body the x-ray beam , the x-ray beam experiences a decrease in intensity due to the different types of particle interactions, which is called *attenuation*. When the x-ray beam exits the body, its intensity distribution will have changed due to these attenuation differences. Since the attenuation correlates with types of materials the detected projections can be used to reconstruct an internal image of the body, which requires an understanding of the attenuation mechanisms of the x-ray beam.

The intensity of an x-ray beam is determined by the amount of energy of the photons that is produced each second per unit area. For a monochromatic x-ray beam containing photons with only one energy, the intensity is given by

$$I = E\phi, \tag{2.3}$$

where $E = hv$, which is the energy of each photon in a monoenergetic beam,and ϕ is the energy fluence rate given by

$$\phi = \frac{N}{A\Delta t}, \tag{2.4}$$

where N is the number of photons, A the unit area and Δt the time interval of the measurement.
As the x-ray beam propagates through tissue, the intensity decreases. The amount of attenuation depends on the *linear attenuation coefficient* μ of the material, which determines the amount of photons that are absorbed or deflected per unit length due to the interactions described in paragraph 2.1.2. The value of μ is determined by the energy of the x-ray beam and the material through which it propagates. For a monochromatic x-ray beam that starts with intensity I_0, the decreased intensity after travelling a distance x through a material with a linear attenuation coefficient μ can be described with Equation 2.5:

$$I_x = I_0 \cdot e^{-\mu x dx} \tag{2.5}$$

The x-ray beams used in medical imaging are *polychromatic*, which means that they contain x-rays of multiple different energies. On top of this, biological tissue is heterogeneous in nature which means that it contains several different materials which cannot be described with the same value for μ. Because of this, equation 2.5 cannot be used to describe the intensity in a realistic setting. Equation 2.6 describes the intensity I_x for a polyenergetic beam, which is essentially the sum of the integrals as described in Equation 2.5 for each energy E in the x-ray beam and each μ in the material along with the corresponding path length x.

$$I_x = I_0(E) \cdot \int_E e^{-\int_x \mu(x,E)dx} dE, \tag{2.6}$$

To solve this equation, the polychromatic x-ray beam is approximated as a monochromatic beam with *effective energy* E_{eff}, which is the energy at which the monochromatic beam produces similar measurements as the polychromatic beam. This way, the intensity measurement can be written as:

$$I = I_0 \cdot e^{-\int_x \mu(x;E_{eff})dx}, \tag{2.7}$$

in which I_x is the measured intensity and I_0 is the reference intensity, which is the measured intensity without an object between the source and scanner. This equation expresses the measured intensity as a line integral of the linear attenuation coefficient for the effective energy of the x-ray beam. This is one of the main concepts used in the formation of a CT image, which is explained in paragraph 2.1.4.

2.1.4. 2D image reconstruction

The line integral from Equation 2.7 can be rearranged into Equation 2.8, which gives the projection measurement g_d as an integral over the attenuation coefficients in the volume at the effective energy E_{eff}.

$$g_d = -ln(\frac{I}{I_0}) = \int_0^x \mu(x;E_{eff})dx \tag{2.8}$$

To reconstruct an image using Equation 2.8, the reference intensity I_0 must be known. It can be determined with a calibration measurement of the detector, in which no object is placed between the source and detector.

Differences in hardware between CT systems such as the type of x-ray source or detector often results in different reconstructed values for μ. To accurately compare reconstructed images from different CT scanners, the reconstructed attenuation values are transformed into CT numbers h, expressed in *Hounsfield Units (HU)*, using Equation 2.9:

$$h = 1000 \times \frac{\mu - \mu_{water}}{\mu_{water}} \tag{2.9}$$

For water, $h = 0\ HU$ and for air, which has a value of $\mu = 0$, the value of the CT number is $h = -1000\ HU$. For the human body, the largest CT numbers are found in bones, which have a CT number that is usually around $h \approx 1000\ HU$.

Figure 2.3a illustrates this concept for a fan-beam CT (FBCT) system which uses a linear detector array. The source emits a fan-shaped x-ray beam and is placed opposite of a curved detector. During image acquisition, the system is rotated which enables it to take projection measurements of the object over the entire $360°$ range. The CT system displayed in Figure 2.3b is a cone-beam CT (CBCT) system which uses a flat 2D detector array to obtain projection images for each angle, requiring only one rotation around the patient to obtain projection data for 3D image reconstruction. This paragraph will explain the concept of CT image formation in 2D, which serves as a simplified representation of the more complex 3D reconstruction process.

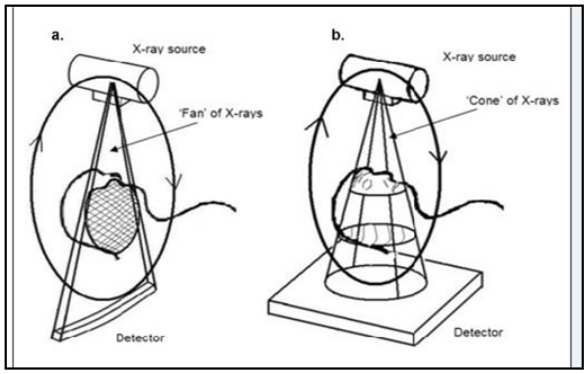
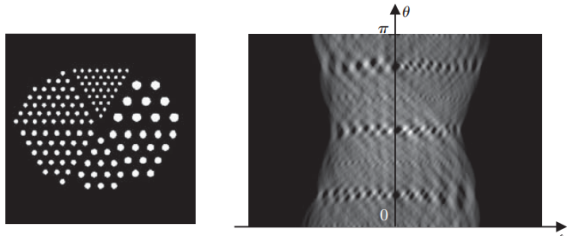


Figure 2.3: An illustration of two types of CT system. (a) A fan-beam CT (FBCT) system which uses a linear detector array. The source emits a fan-shaped beam and is placed opposite to the detector. During data acquisition, the system rotates to obtain projection data over the entire $360°$ range and reconstruct a slice of the volume. (b) A cone-beam CT (CBCT) system (b) which uses a flat 2D detector panel, requiring only one rotation to reconstruct a 3D volume. Figure from [42].

After acquiring the projection measurements for all angles, they are arranged in a 2D array called a *sinogram*, illustrated in Figure 2.4. The acquired sinogram data does not yet look like an image. To obtain a representative image from this sinogram, it has to be reconstructed into an image. An intuitive way to do this is with backprojection: each projection is 'smeared back' across the image plane with the same angle it was acquired. Doing this for all angles results in a blurry 2D image representation of the imaged object, as shown in figure 2.5.



(a) Object **(b)** Sinogram

Figure 2.4: An object (a) and the corresponding sinogram (b), which is a collection of projection measurements from several projection angles.The horizontal axis represents the length l of the detector and the vertical axis represents the angle θ from which the projections were acquired. Figure from [52].

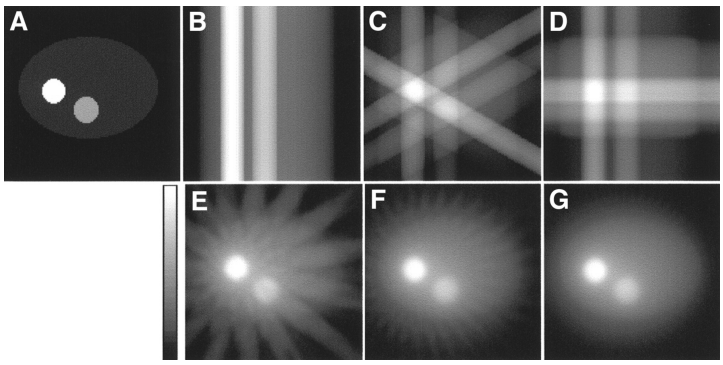


Figure 2.5: A depiction of the concept of backprojection (BP). The acquired projection data is 'smeared' back along its corresponding angle. (A) shows the original object. (B) shows a BP for one angle and (C) to (G) show the BP for an increasing amount of angles. With enough repetitions, a blurred depiction of the imaged object arises as displayed in (G). Figure from [50].

The quality of the image reconstructed with a backprojection is insufficient for practical use, as shown in Figure 2.5. To improve the image quality, the first CT scans in clinical practice used filtered backprojection (FBP) to construct the images. It applies a high-pass filter before performing the backprojection, which significantly reduces the amount of blurring in the reconstructed image. FBP has a short reconstruction time and creates high quality images, which is why it has been the industry standard for decades. However, FBP does not allow any modelling of the physical emission processes, making it hard to eliminate noise and artifacts. The image quality of FBP also significantly reduces with lower dose, which is undesirable as the dose should be kept as low as possible to minimize the impact on the patient's health. Several types of iterative reconstructions (IR) methods have been developed and have shown to significantly improve upon the appearance of noise and artifacts compared to conventional FBP [65].

2.2. Imaging artifacts

Various factors have an impact on the projection data obtained from the detector. Effects that are not accounted for in the reconstruction process give rise to artifacts in the image, degrading the image quality. Their effect on the reconstructed image varies from mild degradation of the image to rendering it completely useless. They can be divided into physics-based artifacts, patient-based artifacts and hardware-based artifacts [8]. This paragraph primarily focuses on the physics-based artifacts, which have the greatest potential for improvement by the forward projection network proposed in this research project, as the aim is to incorporate these processes in the forward projection model.

Frequently occurring artifacts in CT are *beam hardening artifacts*, which arise due to the approximation of the polychromatic X-ray beam as a monochromatic beam with effective energy E_{eff}. These artifacts arise because the mean energy of the polychromatic X-ray beam increases as it propagates through the volume due to the lower-energy X-ray photons being absorbed more rapidly than high-energy photons. This is called the 'hardening' of the X-ray beam. This can produce *cupping* and *streaking* artifacts. Cupping artifacts are best seen with a uniform cylindrical phantom. The middle of the phantom will cause more beam-hardening compared to the sides, as the beam passes through more material. This changes the rate of attenuation, resulting in a relatively higher intensity being detected. Without beam-hardening correction, the reconstructed image will show a 'cupped' shape, illustrated in Figures 2.6a and 2.6b.

Other beam-hardening artifacts include streaks and dark bands, which typically occur between two dense objects. If the beam passes through only one of the objects at a certain angle, it is hardened differently than when it passes through both objects at other angles. These effects most often occur in bony regions or when using a contrast medium. An example of streaking artifacts due to beam hardening is shown in Figure 2.6c.

Artifacts caused by certain aspects of the patient or system geometry may occur as well. Figure 2.7a displays two types of aliasing artifacts that arise when an image is reconstructed with undersampled data. When an insufficient amount of projection angles is used, *View aliasing* occurs which presents as fine stripes at the edges of dense objects. *Ray aliasing* is caused by undersampling of the projection itself, and is expressed as stripes that appear close to the edges of dense objects. Motion artifacts, displayed in Figure 2.7b are caused by patient motion result in misregistration in the reconstruction process, resulting in shading and streaking artifacts. An example due the scanner geometry is the ring artifact, shown in Figure 2.7c. A misaligned or failed detector used in a rotating system creates a circular artifact, as the center of the detected projections are not aligned.

2.3. Iterative Reconstruction

This section will give a brief overview of iterative reconstruction methods used in CT imaging. First, the general concept of iterative reconstruction (IR) methods is explained, describing their improvements regarding image quality and dose reduction compared to FBP. Model-based IR (MBIR) is introduced along with the improvements of performance and the challenges they come with. This is followed by a more in-depth explanation of raytracing methods, which are commonly used within MBIR algorihtms

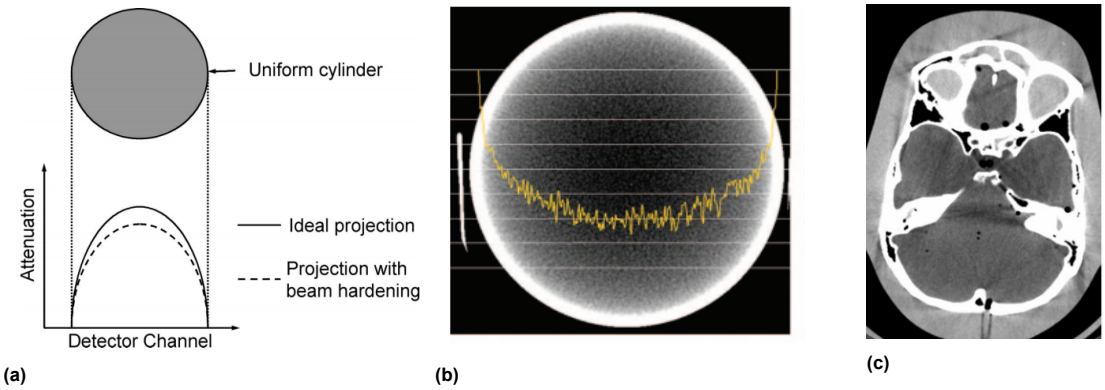


Figure 2.6: Examples of beam hardening artifacts. (a) A projection for a situation with and without beam hardening. The x-ray beam experiences more beam hardening in the center due to the larger amount of material in its path, which changes the detected profile. (b) Cupping artifact resulting from the distorted projections due to beam hardening. The line signal represents the detector response.The object gets assigned lower HU values closer to the center of the object, resulting in a 'cupped' intensity profile. (c) Streaking between two dense objects due to beam hardening. When the beam propagates through only one of the objects at certain angles, it experiences a different amount of beam hardening than if it passes through both objects at other angles, resulting in streaks or dark bands in the image. Figures from [8].

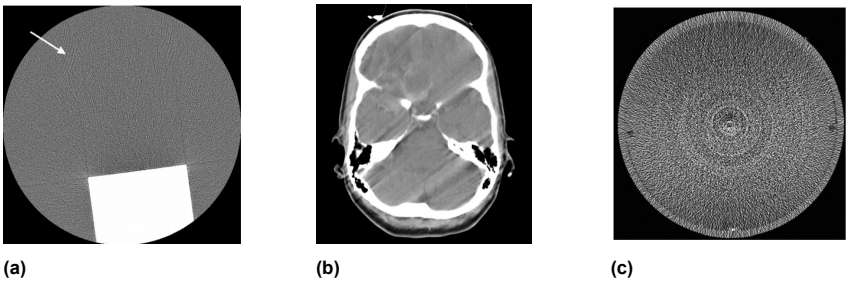


Figure 2.7: (a) An image containing two types of aliasing artifacts caused by undersampling. *View aliasing* due to an insufficient number of angles, visible as fine stripes at a distance of the edges of dense objects and *Ray aliasing* due to undersampling of the projection, appearing as stripes close to the edges of dense objects. (b) Depicts artifacts due to patient motion interfering with the reconstruction process This causes shading and streaking the image due to the misregistration of the projected values. (c) A ring artifact, caused by a misaligned or defect detector resulting in rings in the reconstructed image. Figures from [8].

to perform the forward projection step. The next paragraph presents the Simultaneous Iterative Reconstruction Technique (SIRT), which is a model-based iterative reconstruction (MBIR) method that is used in paragraph 4.4.2 to evaluate the results of this project.

2.3.1. Iterative Reconstruction

In 2009, the first IR method was introduced in clinical practice and several followed in the years after that [68]. Figure 2.8 compares the concept of FBP to Hybrid IR and Model-Based IR (MBIR), which are two categories of IR methods. Hybrid-IR, also called statistical IR, uses statistical iterative methods to reduce noise in the sinogram and image data separately, before and after the backprojection step [65]. They have shown great improvements in image quality compared to FBP, allowing significant dose reductions. A main challenge in these methods is the contrast-dependent spatial resolution, which degrades much more rapidly with lower dose compared to FBP, in which the spatial resolution is similar in all regions of the image [38]. Especially with large dose reductions, the detection of low-contrast tissue degrades ([29],[7]), sometimes rendering them invisible.
The performance of IR on low-noise regions can be improved with the use of more accurate modelling of the x-ray physics and CT system geometry. This is done by MBIR methods which perform BP as well as FP operations. A forward projection reconstructs a sinogram from CT image data by modelling the detector response resulting from the x-ray beam propagating through the medium. To calculate these forward projections, MBIR algorithms use varying methods to model the physical processes and CT system geometry. An MBIR reconstruction starts with an initial guess of the image, after which a forward projection is carried out to calculate what the raw data would look like if the initial image were correct. Next, the artificial raw data and true raw data are compared and the image is updated. These steps are repeated until the true and artificial raw data are similar enough [65].

IR algorithms have been reported to reduce radiation dose for a CT image with 23 to 76% [67] while retaining image quality, with MBIR enabling the largest decrease in radiation dose [66]. However, the implementation of these nonlinear models results in an increased computation time which makes them time consuming and difficult to use in clinical practice.

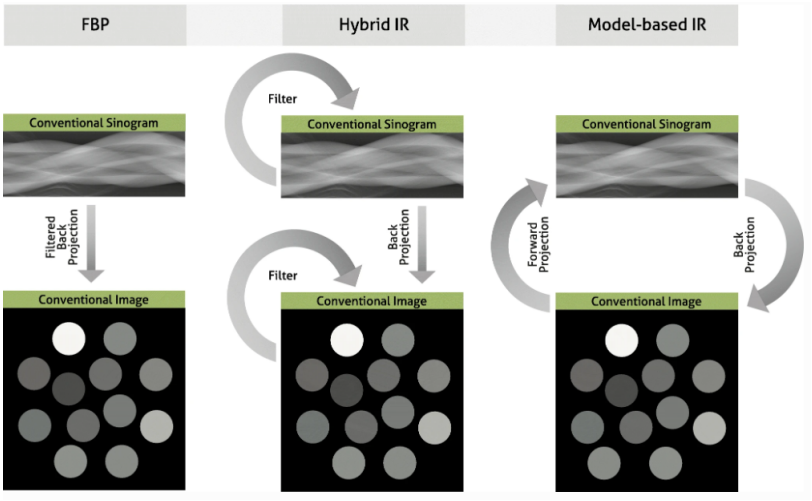
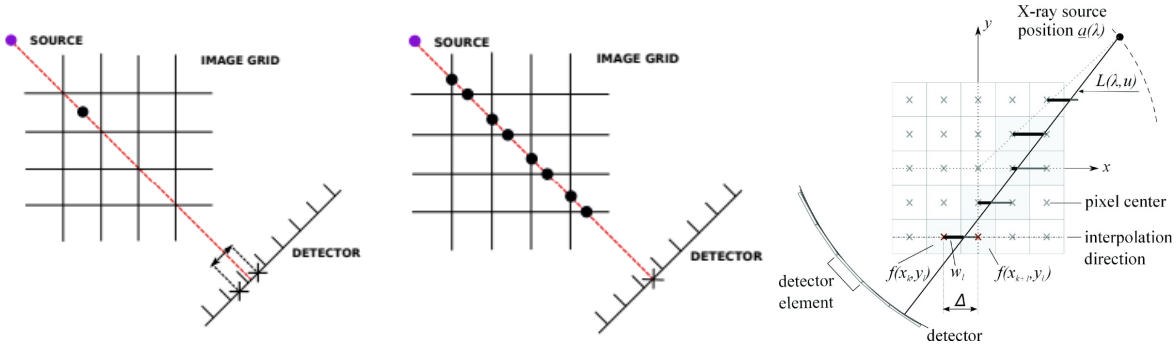


Figure 2.8: An illustration of the concepts of filtered backprojection (FBP), hybrid iterative reconstruction (IR) and model-based IR. In FBP, an image is reconstructed by applying a high-pass filter to the projection data, followed by a backprojection. Hybrid IR methods apply iterative methods to reduce noise and artifacts in the projection or image data separately. Model-based IR methods iteratively perform forward- and backprojections in which they implement physics-based models. Figure from [65].

2.3.2. Projection operators

The key parts of IR methods are the algorithms that perform the *forward-* and *backprojections.* An FP reconstructs a sinogram from CT image data by modelling the detector response resulting from the x-ray beam propagating through the medium. This section will focus on methods to implement the FP operation, although most techniques that are used for the FP operation are applicable to BP operations

as well. There are various methods that can be used to perform forward projections. Monte Carlo (MC) methods use physics-based models to simulate each individual particle trajectory, creating a projection with high accuracy. Unfortunately, these methods are very time- and computationally intensive and are not a viable option for use in practice. Analytical forward projectors based on the Fourier transform are much faster as they use single mathematical operations, but they do not incorporate physical effects very well, producing a lower quality reconstruction.
Most forward projection methods used in practice are raytracing methods which rely on numerical integration. Commonly used types are *ray-driven* and *pixel-driven* methods [3], illustrated in Figures 2.9a and 2.9b. Pixel-driven methods trace a line starting from the X-ray source through each pixel center, ending in the detector. The two detector cells that surround the endpoint of the ray are assigned the pixel value with a weighting determined by a linear interpolation. The ray-driven methods trace a ray from the source to the detector as well but this time the ray ends in the center of each detector pixel. Intersection points in the pixels are computed using the Siddon algorithm [55] and the value of the detector cell is the sum of the encountered pixels, weighted according to their ray path length.
Another raytracing method, called *Joseph's method* [27], is illustrated in Figure 2.9c. This methods measures straight line integrals, approximating the integral as a summation in x or y, depending on the projection angle. The linear interpolation is performed along the other axis. The type of forward projector influences the final result and which method to use depends on the type of image reconstruction, computational resources and other requirements. The raytracing algorithm that is used in this project [45] and described in section 4.4 implements Joseph's method, which provides a trade-off between image quality and computational expense [22].



(a) Pixel-driven approach. Figure from [3].

(b) Ray-driven approach. Figure. from [3].

(c) Joseph's method. Figure from [22].

Figure 2.9: Illustration of different forward projection methods. (a) Pixel-driven methods trace a line from the X-ray source through each pixel center, ending in the detector. The two detector cells that surround the endpoint of the ray are assigned the pixel value with a weighting determined by a linear interpolation. (b) Ray-driven methods trace a ray from the source to the detector as well but this time the ray ends in the center of each detector pixel. Intersection points in the pixels are computed using the Siddon algorithm [55] and the value of the detector cell is the sum of the encountered pixels, weighted according to their ray path length. (c) Joseph's method measures straight line integrals, approximating the integral as a summation in x or y, depending on the projection angle. The linear interpolation is performed along the other axis.

2.3.3. Simultaneous Iterative Reconstruction Technique

This Simultaneous Iterative Reconstruction Technique (SIRT) algorihtm is a MBIR method which reconstructs an image using algebraic methods, performing operations simultaneously on all pixels instead of one pixel at a time, which is used by other methods [34]. In SIRT, the image reconstruction process is represented as a system of linear equations as shown in Equation 2.10:

$$Ax = b \tag{2.10}$$

where A is the system matrix, which represents the forward projection process, x is the image and b the sinogram. Each row in the system matrix A contains weights that describe the sum of the encountered pixels for single ray. The transposed system matrix A_T describes which pixels are hit by a ray and can be used to perform a backprojection. SIRT iterates between forward and backprojections, updating the image using Equation 2.11:

$$X^{t+1} = x^t + CA^T R(b - Ax^{(t)}) \tag{2.11}$$

where C and R are diagonal matrices containing the inverse of the sum of the columns and rows of the system matrix, which compensate for the number of rays that hit each pixel and the number of pixels

that are hit by each ray and t is the number of the iteration.

The iterative process starts with an initial estimate of the reconstructed image X_0=0. The steps in each iteration are as follows:

- The current image reconstruction $x^{(t)}$ is forward projected with $Ax^{(t)}$
- This is subtracted from the original projections: $b - Ax^{(t)}$
- The difference is backprojected, multiplying it with A^T and the correction factors C and R, resulting in $CA^TR(b - Ax^{(t)})$
- This term is added to the current reconstruction which results in Equation 2.11: $X^{t+1} = x^t + CA^TR(b - Ax^{(t)})$

This process is repeated until a sufficient image quality is reached and the model has converged to an optimal reconstruction.

Both the FP and BP operators used in the SIRT algorithm are often raytracing operators as described in paragraph 2.3.2. While FP operations compute a projection from an image representation, BP operations transform the projections into images. The operators can be either *matched* or *unmatched*. Projection operators are considered matched when they perform the same operations, but in reverse order. Matched projection pairs are not required to make the SIRT algorithm converge. Their is no consensus on whether to use either matched or unmatched projection pairs in IR algorithms [73]. Unmatched projection pairs have shown to result in both better [21] and worse [3] performance compared to matched projection pairs.

3

Deep Learning

In the last decade, deep learning (DL) has created a world of possibilities in many fields. While it is mostly known for its applications in language processing or image and speech recognition, its possibilities reach much further. Some examples of real-life applications include document generation [44], fraud detection [43] and driverless cars [5] . In the medical field deep learning has also been applied in several ways including cancer cell classification [72], organ segmentation [54], disease diagnosis [1] and drug discovery [41].
DL applications in CT image reconstruction focus mostly on image pre- and postprocessing, removing noise and artifacts from raw data or reconstructed images, allowing dose reductions between 30% and 71% compared to hybrid IR methods [31]. This chapter will explain the deep learning concepts related to this research project. First, a basic overview is given of the fundamental concepts behind deep learning. This is followed by a section on sequence processing networks, including convolutional neural networks and the Transformer architecture that forms the basis of the deep learning model proposed in this thesis.

3.1. Basic DL concepts

Deep learning is a specialized subset of *machine learning*, a type of artificial intelligence focused on creating methods that allow machines to 'learn' from data by identifying patterns and making predictions for several types of tasks. They can be categorized into three main types: *supervised, unsupervised* and *reinforcement* learning. In supervised learning, a model is trained on labeled data with a known output. On the other hand, unsupervised learning does not use labelled data and aims to make the network learn patterns and structure in the data. In reinforcement learning, the model learns by receiving feedback on its output and adjusting its weights accordingly.

Compared to other machine learning algorithms, deep learning networks are more complex and do not need structured data. They are inspired by biological neural networks, containing multiple nodes called neurons which can transmit signals to each other like synapses between neurons in human brains. Using a sufficient amount of data, neural networks 'learn' using the concept of *backpropagation*, in which the network output is compared to the desired output. The difference between the actual output and the desired output is called the *loss* and is backpropagated through the network to adjust the weights and biases of the neurons. With enough repetitions, the accuracy of the predictions improves and the network has learned to produce the desired output. Training a neural network requires a significant amount of computational power, especially for complex networks and large data sets. When these resources are available, deep learning networks have shown to achieve incredible performances in many different areas.

3.1.1. Deep Feedforward Networks

Deep feedforward networks, also called multilayer perceptrons (MLPs), are the first and simplest type of neural network on which many other networks are based. The network contains neurons that are organized into several layers. The first layer receives the external data and is called the input layer. The last layer is the output layer and produces the network predictions. The layers in between are called hidden layers, of which the number can vary from zero to multiple depending on the complexity of the network. While the training data specifies the desired output from the output layer, it does not specify the outputs of the in-between layers. The algorithm learns the optimal parameters of these layers during the training process, using them to lead to the desired output from the output layer. An example of an MLP is shown in figure 3.1b.

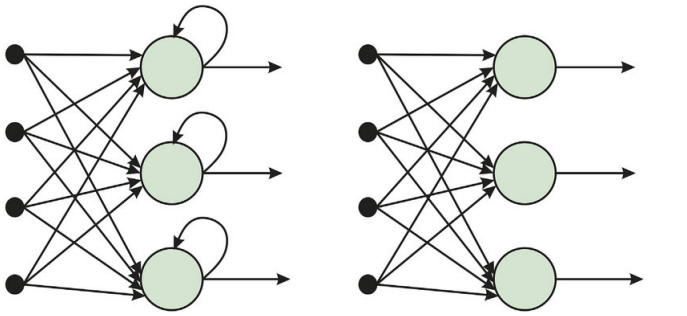


Figure 3.1: a) An example of a recurrent neural network, in which there are connections between neurons in the same or previous layers. b) An example of a feed-forward network, the simplest type of neural network. It consists of an input layer, one or more 'hidden' layers and an output layer. Figure from [19].

3.1.2. Convolutional Neural Networks

Convolutional neural networks (CNNs) have been extremely succesful in a large variety of applications. CNNs are networks that use convolution in one or more layers. Using convolution in neural networks has three main advantages. First, they have *sparse connectivity*, which reduces the number of parameters that need to be stored. Secondly, parameter sharing allows the model to use parameters for more than one function, which reduces the memory requirements of the model. Finally, CNNs have translational equivariance, which means that a translational change in the input causes an equal translation in the output.

3.1.3. Sequence Modelling

Certain deep learning networks are especially useful for processing sequential data. The most basic type of sequence transduction is the recurrent neural network (RNN). In RNNs, there are connections between neurons in the same or in previous layers. This makes it possible for the network to take historical information into account, allowing it to process input of any length. An example of an RNN is shown in figure 3.1a.
RNNs can also be trained to map inputs to outputs with different lengths. This is particularly useful in applications such as speech recognition or translation. In 2014, Cho et al. [13] and in the same year Sutskever et al. [58] proposed an encoder-decoder architecture that can be used for this. In this network, the encoder processes the input sequence and relates it to the context C. The decoder generates an output sequence from this context. One shortcoming of the encoder-decoder architecture is that the output context C from the encoder can be so small that it cannot accurately represent a long sequence. Bahdanau et al. proposed to make C a variable length instead of a fixed-length vector. They also proposed an *attention mechanism*, which makes the network learn to correlate elements of C to certain elements in the output sequence [6].

A problem with traditional RNNs is the vanishing gradient problem. This is when the gradient of the loss function decreases exponentially with time, making the network less able to learn long-term dependencies. Models that have been proven to be successful in mitigating the vanishing gradient problem are

gated RNNs. The most prominent examples of this are Long Short-Term Memory (LSTM) networks or networks that are based on the gated recurrent unit (GRU).
An LSTM network is a type of RNN which introduces input gates and forget gates, allowing the network to choose whether to memorize or forget certain data. Because of this it is able to hold on to information for a long time, making the network suitable for learning long-term dependencies. LSTM networks are especially useful for sequence transduction. They have shown to be successful in several applications, such as speech- and handwriting recognition or time series prediction. RNNs based on GRUs differ from LSTM networks in that they have one unit controlling both the forgetting factor and the decision to update.

3.1.4. Attention

In a recurrent network, an input sequence is processed by an encoder element-by-element after which the decoder also produces the output sequence element-by-element. An example to illustrate the problem that may arise is in natural language processing: when a RNN translates a sentence from one language to another, it will start to generate the first word of the output sentence after its last input was the last word of the input sentence. The larger the input, the more difficult it becomes for recurrent networks to 'remember' enough information to make accurate predictions. With *attention*, the vanishing gradient problem can be improved upon. Similar to attention in humans, attention in a neural network means that not all information is given equal weight. It allows the neural network to selectively focus on certain elements while ignoring elements that are deemed irrelevant. With sufficient training, the model learns to assign certain weights to the input, focusing on the elements that are important to the task at hand. This results in an improved computation time, by avoiding the processing of irrelevant information.

In the attention mechanism, the model can access all elements of an input sequence at once and assign weights to each element. This is done by mapping a query q and a set of key-value pairs k_i and v_{k_i} to each output. The queries represent the current input to the attention mechanism, while the keys represent the input against which the query is compared. The values are the inputs selected by the attention mechanism which are used to produce the output. The process is as follows:

- For each query q a score e_{q,k_i} is computed against a set of keys using the dot product of the query with each key k_i:
$$e_{q,k_i} = q \cdot k_i \tag{3.1}$$
This score value is a measure of the resemblance of the key to the query.
- A softmax operation is applied which generates the weights:
$$\alpha_{q,k_i} = softmax(e_{q,k_i}) \tag{3.2}$$
- The total attention is computed with a weighted sum of the value vectors v_{k_i}, pairing each value vector with the corresponding key:
$$attention(q, K, V) = \sum_i \alpha_{q,k_i} v_{k_i} \tag{3.3}$$

This network will learn to assign higher weights to values that are most relevant, enabling it to focus on specific parts of the input.

Because the sequence elements are not processed in order, the network needs another way to register the position of each element. This is implemented with positional encoding, in which a *positional embedding* is added to the input. The attention mechanism can be used to relate input and output sequences to each other and is often applied between the encoder and the decoder of a model. On the other hand, a similar concept called *self-attention* relates elements of the same sequence to each other. This enables the model to capture for example the relationship between words in the same sentence.

3.2. Transformer

In 2017, the Transformer algorithm was proposed [62]. This network does not use recurrence or convolutions but relies entirely on this self-attention mechanism. It contains a *multi-headed self-attention (MSA)* block in both the encoder and decoder. This block consists of multiple attention layers (*heads*) that are performed in parallel, performing the computations in Equations 3.1 to 3.3 simultaneously for all queries. The resulting weight matrices are concatenated to produce the final result. This allows the heads to capture different types of information in the input sequence.
With its introduction, the Transformer changed the way that attention is used and can be regarded as one of the main advances in deep learning in recent years. Many different networks based on the Transformer have emerged with widespread applications including natural language processing [15], computer vision [18] and predicting protein structures [28]. Transformer-based models have recently even brought deep learning to the general public with the introduction of widely accessible text- and image generation models [44], [12].

The Transformer's ability to access each element in a sequence at once, combined with the use of self-attention to relate elements within the same sequence to each other give it the ability to reconstruct x-ray particle trajectories by relating each element row of the input image to previous elements. By gathering information about the materials through which the beam has propagated, it is able to determine an integral representation of the material attenuation coefficients, resulting in an accurate prediction of the detector response.

4

Methods

This chapter describes the methodology of this thesis project. A general overview of the research setup is given after which the three main parts of the research project are presented. Section 4.2.1 outlines the generation and processing of the data used for training the neural network. This is followed by section 4.3, which describes the neural network architecture and optimization. Finally, section 4.4 gives an overview of evaluation process, including the types of input data used to test the neural network's performance, other methods which the network is evaluated against as well as qualitative and quantitative evaluation metrics.

4.1. Research setup

The aim of this project is to create a neural network that can model the particle transport of x-ray photons in CT imaging by predicting a forward projection from a CT input image. Monte Carlo ground truth data is used to train the network to learn the physical interactions that influence x-ray particle trajectories. This ground truth data is created using a simulated CT system, which generates projections of several imaging phantoms that represent simplified human tissue in the form of 3D water cylinders with homogeneous box- and cylindrical-shaped inserts with varying shapes and materials. An x-ray source projects a fan-shaped beam onto these phantoms which is detected by a flat detector strip. To simulate a clockwise-rotating detector that acquires projections from different angles, the phantom is rotated counterclockwise while the CT system remains in place. When combined, these projections correspond to a 2D cross-section of the phantom, resulting in circular and rectangular shapes in the reconstructed image. Figure 4.1 illustrates an isometric view and a top view of the simulated setup, indicating the phantom, the detector, the source S, the distance from the source to the detector (SDD) and the distance from the source to the center of the phantom (SOD).

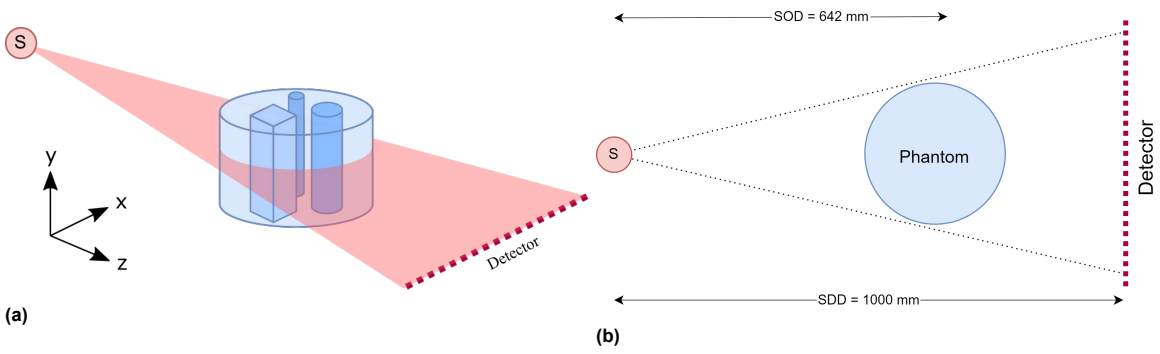
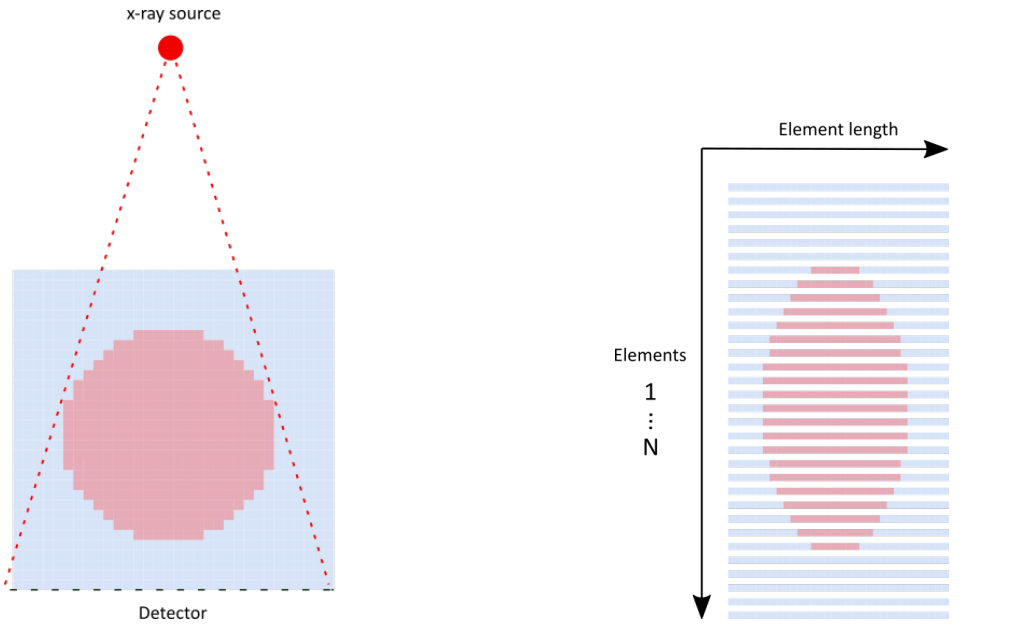


Figure 4.1: (a) an isometric view and (b) a top view of the simulated CT system. S indicates the source, SDD the distance between the source and detector and SOD the distance between the source and the center of the phantom.

The input to the neural network is a 2D image array that represents the phantom corresponding to the ground truth projection. The input image is defined as a sequence with each row of the image treated as a single element. The sequence runs from the x-ray source to the detector array with the top row of the image as the first element. The sequence then 'moves' through the phantom, ending at the last element located at the detector array. This concept is illustrated in figure 4.2. The neural network implements a Transformer architecture to process this input sequence. It computes the relationship of each element to all preceding elements, which enables it to learn the physical interactions that influence x-ray particle trajectories in CT imaging. This approach allows the network to predict forward projections with Monte Carlo-level accuracy at a fraction of the computational cost.



(a) The 2D input array to the network, in perspective to the x-ray source and detector array.

(b) The same input array depicted as a sequence of rows

Figure 4.2: An illustration of the process in which the neural network processes CT image data. The input array is depicted in (a) along with the x-ray source and detector array as a reference of the perspective. This array is processed by the neural network as a sequence of rows running from the x-ray source to the transformer, illustrated in (b). Using a Transformer architecture, the neural network computes the relationship of each element to all preceding elements. This, together with Monte Carlo-based forward projections acting as the ground truth, enables it to learn the particle transport in CT imaging while accounting for physical interactions associated with the imaging process.

This research project consists of three main parts. The first part involves generating a robust data set to train the neural network parameters, which includes input CT images with varying geometries and HU values, as well as ground truth forward projections. The ground truth projections are generated with Monte Carlo simulations based on the same phantoms used in the input images. The second part is the development and training of the neural network, which includes optimizing the network through a hyperparameter search. In the third and final part, the neural network performance is evaluated by comparing its predictions to the ground truth as well as a conventional raytracing method. The neural network's performance within an IR algorithm is assessed to demonstrate its performance regarding physics-based artifacts compared to the raytracing method. The following paragraphs describe the methodology of each part in more detail.

4.2. Training data

4.2.1. GATE simulations

The Monte Carlo projection data is generated with the GEANT4 Application for Tomographic Emission toolkit (GATE) [25], which is a Monte Carlo simulations toolkit for medical physics applications. Using GATE, an x-ray source that irradiates a detector is simulated and in each detector pixel the deposited

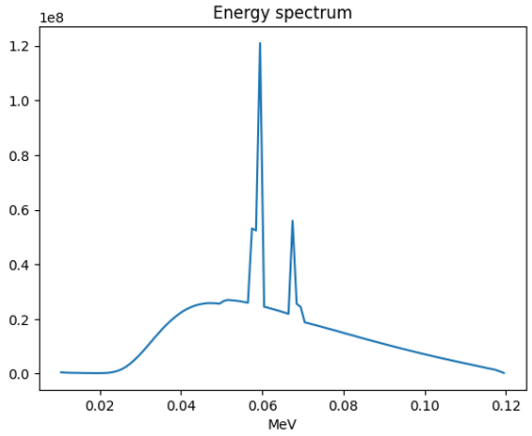


Figure 4.3: Spectrum of the simulated x-ray beam with a 120 kilovoltage peak (kVp), created from a measurement without a phantom. The x-axis indicates the energy bins which range from 0.01 to 0.12 MeV with steps of 0.01 MeV. The y-axis indicates the number of particles detected for each energy bin.

Table 4.1: The specifications of the source and detector geometry of the simulated CT system, as well as the spectrum and filtration of the x-ray beam. The system is based on [30].

CT simulation configuration			
Source		Detector	
Spectrum	120 kVp	Height	7 pixels
Diameter	0.5 mm	Width	439 pixels
Filtration	10 mm Al	Pixel size	1 mm
SDD	1000 mm	Pitch	1 mm
SOD	642 mm	Pixel thickness	1.4 mm

energy of each individual photon hitting the detector pixel is registered. The simulations are adapted from [32], in which simulations are based on a fan-beam CT system by Siemens that includes a photon-counting detector and an energy-integrating detector [30]. For this project, only the energy-integrating detector is utilized. The CT system geometry, x-ray physics and phantoms are specified in the simulation and used to generate projection data of each phantom. The physics model in the simulations is based on the `emstandard_opt4` list in GATE. The x-ray source has a diameter of $0.5mm$ and generates a fan-shaped polychromatic x-ray beam. The spectrum of the beam is obtained from [51] and displayed in Figure 4.3. It is based on a tungsten source with with 10 mm Al filtration removing low energy x-rays from the spectrum and has a 120 kilovoltage peak (kVp). Each simulated projection contains 10^9 primary particles and a large range cut of $1.0mm$ is used in the simulations, which means that secondary particles are only produced when their range is expected to exceed this distance. To reduce computation times, a beam shaper is used which is placed at a distance of $40mm$ from the source, reducing the beam into a rectangular shape and preventing the simulation of particles outside of the detector range.

The detector is made of GOS scintillation material (Gd_2O_2S) and has a width and height of $439\ mm$ and $7\ mm$ respectively. The size of the pixels is 1 mm, the thickness is 1.4 mm and the size between the centers of two adjacent pixels (*pitch*) is 1 mm. Contrary to the curved detectors in typical fan-beam setups, the detector is straight, similar to the flat 2D detectors typically used in CBCT but with a reduced height. In a real CT system the detector signal is obtained from a photodiode detecting the scintillation light from the material which lowers the detection efficiency while in the simulation, all particles that hit the detector are registered. For each pixel the detected energies are summed, resulting in an energy-integrated detector response. The specifications of the CT system are summarized in table 4.1.

4.2.2. Phantom geometry

Each phantom consists of a water cylinder with varying box- or cylindrical shaped inserts, surrounded by air. The phantom inserts are made of one of five synthetic materials that have HU values of -800, -300, 200, 700 and 1200 which all lie within the range of human tissue [9]. Table 4.2 displays the HU value, density ρ and linear attenuation coefficient μ of each material, as well as the elemental composition of the synthetic materials. In the Monte Carlo simulations, the materials are defined with their elemental composition while in the raytracing methods (described in paragraph 4.4.2) the materials are defined based on their μ value. For air and water, the densities are obtained from the standard material definitions in GATE and μ is obtained from [71] using $E_{eff} = 80\ keV$. The value of μ for the synthetic materials is calculated with Equation 2.9. The densities and elemental compositions of

the synthetic materials have been computed with the Schneider stoichometric calibration method [53]. Table 4.2 shows similar elemental compositions for material 1 and 2. This is due to the HU values being divided into bins that get assigned the same elemental compositions. According to [53], dividing the values within these bins into different compositions does not significantly enhance the accuracy of the elemental weights.
Due to an error in the generation of these compositions, the mass fraction of Sodium is assigned to Potassium, adding to the original mass fraction of Potassium. This results in minor errors in the material compositions, which leads to less realistic materials [53]. This is a minor concern, as the aim of this research is to evaluate the performance of the network on a variety of materials rather than specific or realistic materials.

Table 4.2: Density (ρ), HU value and linear attenuation coefficient (μ) for the materials that appear in the training data set which are air, water and the synthetic materials 1 to 5 of which the elemental composition is given as well. The densities and elemental compositions of the synthetic materials are determined based on their HU values using the Schneider stoichometric calibration method [53]. For air and water, ρ is obtained from the material definitions in GATE [25] and μ is obtained from [71]. For the synthetic materials, μ is calculated according to equation 2.9

				Elemental mass fraction									
	ρ $(\frac{mg}{cm^3})$	HU	μ (cm^{-1})	H	C	N	O	Mg	P	S	Cl	K	Ca
Air	1.2900	-1000	2.144e-4										
Water	1000.0	0	1.837e-1										
Synthetic materials													
1	207.15	-800	3.691e-2	0.103	0.105	0.031	0.749	-	0.002	0.003	0.003	0.004	-
2	722.00	-300	1.287e-1	0.103	0.105	0.031	0.749	-	0.002	0.003	0.003	0.004	-
3	1135.5	200	2.204e-1	0.095	0.455	0.025	0.355	0.000	0.021	0.001	0.001	0.002	0.045
4	1431.4	700	3.121e-1	0.066	0.310	0.033	0.394	0.001	0.061	0.002	-	0.001	0.132
5	1727.4	1200	4.039e-1	0.045	0.210	0.039	0.420	0.002	0.088	0.003	-	0.001	0.192

For each phantom, projections are generated for twelve angles ranging from 0 to 330 degrees with steps of 30 degrees. Instead of rotating the simulated CT system, a separate phantom is generated for each rotation angle with a rotation in the opposite direction of the supposed CT system rotation. Due to computational limitations, less than twelve projections were generated for several phantoms. These projections were included in the data set as the projections are processed separately without interaction with projection data from other angles.

4.2.3. Data processing

The input samples to the neural network must be 2D arrays representing the imaged phantom at a certain projection angle. To generate this input, the phantoms described in paragraph 4.2.2 are generated in Python to obtain a discrete 2D array containing the HU values of the materials. Each pixel in the image corresponds to a 1 mm distance in the GATE simulation. The number of columns in the array is equal to the number of detector pixels (439), and the number of rows is 500. The number of rows between the center of the phantom and the detector is 358, corresponding to the 358 mm distance between the two in the simulations. The number of pixels between the center of the phantom and the source is 142, which is a smaller distance than the SOD of 642 mm in the simulation. The removed area relates to part of the distance that the x-ray beam travels through the air between the source and the phantom. As the beam in this area has not yet propagated through tissue, the simulated transport will be nearly identical for all phantoms. Removing the rows corresponding with this distance significantly reduces the computation time of the neural network. The distance through the air between the phantom and the detector is left unchanged, as the particle transport in this area will be different for each phantom. An example of an input image array is shown in figure 4.4.

The Monte Carlo projection data described in paragraph 4.2.1 provides the energy-integrated response for every pixel in the detector. As this project focuses on 2D reconstruction, the detector response is summed along the height containing 7 rows, resulting in a 1D response with the same length as the detector width (439). To create a projection measurement from this data, a measurement of the reference

intensity I_0 is simulated using the simulation setup without a phantom. To reduce noise, the measurement is carried out using $5 \cdot 10^{11}$ particles after which I_0 is obtained using equation 4.1, in which I_0 is the scaled reference measurement and $I_{0_{5e11}}$ the reference measurement for $5 \cdot 10^{11}$ particles. The data is transformed into a projection measurement g_d with equation 4.2.

$$I_0 = I_{0_{5e11}} \cdot \frac{10^9}{5 \cdot 10^{11}} \tag{4.1}$$

$$g_d = -ln(\frac{I_d}{I_0}) \tag{4.2}$$

An example of a resulting projection measurement is displayed in figure 4.4.
Finally, for both the input and ground truth data the minimum and maximum values over the entire data set are determined and used to normalize both data sets before being processed by the neural network.

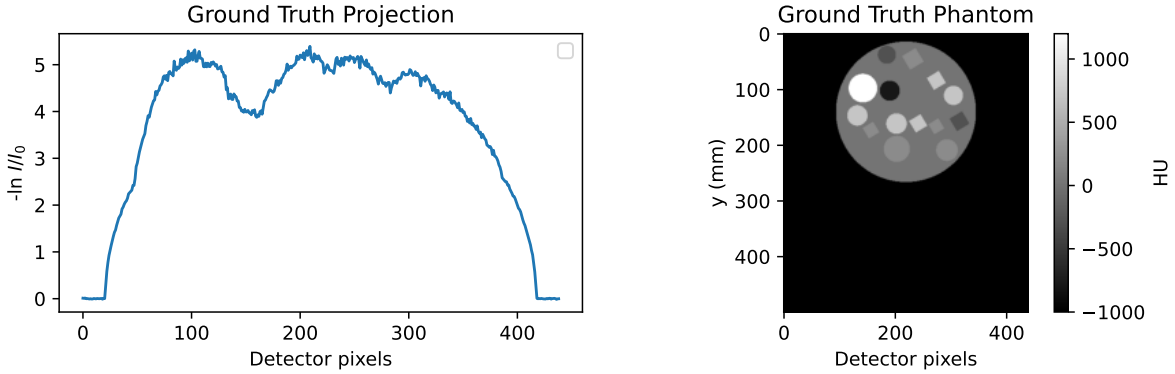


Figure 4.4: Examples of the training data of the neural network. A Monte Carlo-based projection used as the ground truth (left) and the corresponding phantom image used as the input (right).

4.2.4. Hardware and software configuration

The simulations were generated on the TU Delft HPC11 (High-Performance Computing) cluster. Each submitted job utilised one node and projections were generated consecutively. Each projection simulation was split over ten simultaneously running CPU cores each simulating a tenth of the particle trajectories, after which the detector responses were summed.

4.3. Neural network

The input image depicting the phantom is processed by the neural network, starting with the convolutional encoder which extracts the most important geometrical features from the input and translates it into a compressed representation. This is fed into a Transformer encoder that applies a causal self-attention architecture to associate each element of the sequence to all preceding elements. The last element, which relates to all other elements, is isolated and processed by a convolutional decoder which transforms it into the correct output shape. A simplified depiction of the neural network and the data samples is shown in figure 4.5.
The next paragraphs elaborate on the methods used for the generation of the training data and the details of the neural network architecture. This is followed by a description of the hyperparameter optimization and training of the neural network. Finally, the evaluation process is described including an explanation of the evaluation metrics and the use of a SIRT reconstruction method using the trained model.

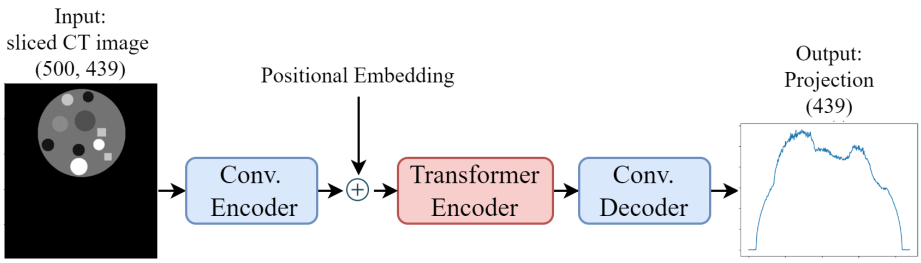


Figure 4.5: Simplified overview of the architecture of the neural network. The input image is processed by a convolutional encoder that extracts geometrical features from the input. The positional data is preserved through embedding after which the sequence is fed into the transformer encoder, which computes dependencies between the sequence elements. From this output, the last element is transformed by the convolutional decoder into the correct prediction shape.

4.3.1. Convolutional encoder

The convolutional encoder extracts geometrical features from the input sequence and transforms it into a sequence of compressed elements called *tokens*, which are used as input to the Transformer encoder. The structure of the convolutional encoder is shown in figure 4.6. First, the input sequence is zero-padded to obtain a sequence of size $(500, 440)$, which is easier to process. Next, the sequence is processed element-wise by three identical convolutional blocks. These consist of a convolution layer with 64 channels and a kernel size of 3, a max pooling layer with a stride of 2, a group normalization layer with 16 groups [70] and a rectified linear activation unit (ReLU). The resulting sequence with shape $(500, 55, 64)$ is processed by a final convolution layer with 12 channels, a group normalization layer with 4 groups and a ReLU layer. Next the data is flattened resulting in a sequence of shape $(500, 660)$. This sequence of tokens is given a learnable positional embedding before being fed into the Transformer encoder.

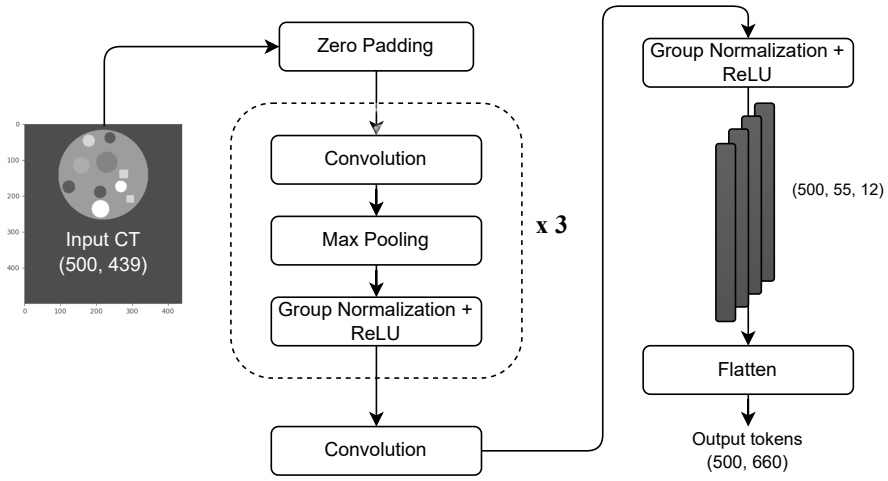


Figure 4.6: Architecture of the convolutional encoder. The zero-padded input is processed by multiple convolution blocks containing convolutional, maxpooling, group normalization and ReLU layers. The encoder extracts geometrical features and transforms the input into a compressed sequence which is flattened to form the sequence of tokens that, after positional embedding, are used as input to the Transformer encoder.

4.3.2. Transformer encoder

The tokens resulting from the convolutional encoder are processed by the Transformer encoder, which captures the causal relationship between the input tokens. The Transformer encoder is illustrated in figure 4.7. First, the input tokens pass through a layer normalization layer [4] before being processed by a MSA block containing 32 heads. By applying a causal mask to the MSA layer, each token will only be influenced by its preceding tokens. The MSA layer has a dropout rate of 0.2 and a *Gaussian Error Linear Unit* (GELU) activation [24]. After this the input passes through a multilayer perceptron (MLP) block consisting of two instances of a dense layer followed by a dropout layer. A residual connection is applied after each block, connecting the output from one block to the input from the next block without

being processed. The output of the Transformer encoder has the same shape as the input $(500, 660)$ and is fed into the convolutional decoder to be reconstructed into the final projection.

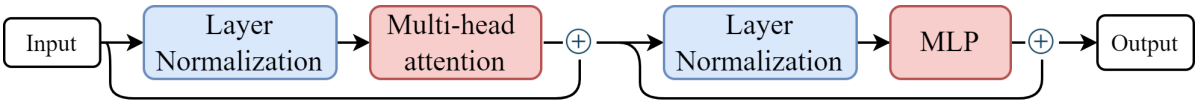


Figure 4.7: Architecture of the Transformer encoder. The input tokens go through a layer normalization layer and a multi-head self-attention (MSA) block which computes the causal relationship between the tokens. This is followed by another layer normalization layer and a multilayer perceptron (MLP) block. Residual connections are applied after each block.

4.3.3. Convolutional decoder

The tokens resulting from the Transformer encoder are processed by a convolutional decoder which transforms the compressed tokens into the final output shape. Because the Transformer uses a causal MSA block, the last element contains information related to all other elements. Because of this, only the last element is extracted from the sequence and processed by the convolutional decoder, which has an architecture that resembles the reverse of the convolutional encoder architecture, only performing operations on the last element. The architecture is illustrated in figure 4.8.
First, the flattened input is reshaped to 2D, after which the last element is extracted. Next are three convolution blocks that are identical to the convolution blocks in the convolutional encoder, except with upsampling layers instead of maxpooling layers. This is followed by a final convolution layer and a layer which removes the last element of the sequence to obtain the original input length, resulting in the final prediction.

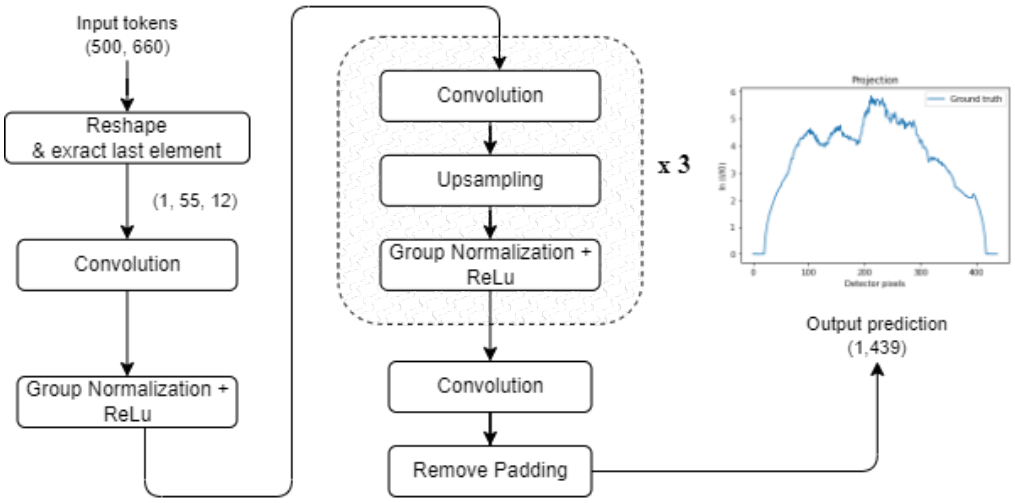


Figure 4.8: Architecture of the convolutional decoder. The flattened input is reshaped to 2D after which the last element is extracted. The element is processed by multiple convolution blocks containing convolutional, upsampling, group normalization and ReLU layers, resulting in the final output prediction.

4.3.4. Hyperparameter optimization

To optimize the performance of the neural network, various network architectures are tested to obtain the best performing model. While it would be ideal to explore all possible configurations, the computational power required to accomplish this is too extensive. Instead, a hyperparameter search in which all possible combinations of a small number of parameters are tested. The hyperparameters that were tuned are displayed in Table 4.3 along with the values that were used in the grid search.

Table 4.3: Hyperparameters that were optimized in a grid search. K is the number of filters in the last convolution layer of the convolutional encoder, N_H is the number of heads in the Transformer blocks and N_T is the number of Transformer blocks. The grid search is carried out using every possible combination of the displayed parameter values.

Hyperparameter	Value (#)	
K	8	12
N_H	16	32
N_T	1	2

The model configurations in the grid search are evaluated on a test data set containing 1320 samples. The best performing model is trained with a data set containing $38,381$ samples and a learning rate that decays when the training reaches a plateau, combined with restarting the learning rate to a higher value when the training shows insufficient improvement [17].

4.3.5. Hardware and software configuration

The neural network was was built in Python using Tensorflow 2.6 [61]. Training was performed using GPU resources of DelftBlue Supercomputer, provided by the Delft High Performance Computing Centre (DHPC) [14].

4.4. Evaluation

The performance of the Transformer model is evaluated by comparing its predictions to conventional raytracing projections performed on GPU hardware (Nvidia GeForce GTX 1070Ti). The individual projections of both networks are compared using the MC ground truth data as a baseline. Additionally, the Transformer model is implemented within a SIRT algorithm to assess its performance within an IR method. A SIRT algorithm that uses a conventional raytracing FP operator is used for comparison, as well as a Two-Angle Convolution (TAC) network. This network produces forward projections, using raytracing projections at right angles as input rather than phantom images. Both the individual projections and the image reconstructions are evaluated based on their average error, prediction speed and presence of physics-based artifacts.

Besides this, projections of the Transformer model and raytracer are compared to low-noise ground truth data as well. The aim of this comparison is to display the MC projection with a higher resolution which better visualizes the underlying structures of the input phantom. This way, small details and errors in the Transformer and raytracer projections can be distinguished from each other. These low-noise MC projections are not used to compare the Transformer and the raytracer to each other, as the Transformer has not been trained with the same level of noise as in the low-noise MC projections which makes it unable to learn from smaller details.

To evaluate how well the Transformer model is able to generalize, it is tested on a data set that contains several types of input data which are not included in the training data set. Besides giving an indication of the generalizability of the model, it may also reveal potential biases in the network due to insufficiently robust training data.

The individual data sets are described in more detail in paragraph 4.4.1 and the algorithms used in the SIRT comparison are described in paragraph 4.4.2. Finally, paragraph 4.4.3 describes which metrics are used in the evaluation of the various results.

4.4.1. Evaluation data sets

This paragraph describes the different types of data sets used for evaluation. Each of these data sets contain only unseen samples, which means that they are not included in the training data of the model.

- **Conventional phantoms** This data set contains 4359 samples, generated in the same way as the training data. These results are compared to raytracing and MC projections on the same data set using the measures described in paragraph 4.4.3.

- **Low-noise MC ground truth** This data set contains twelve unseen input samples generated in the same way as the training data, compared to raytracing projections as well as MC projections that are generated with ten times as much primary particles as the MC training data, reducing the amount of noise in the MC data. This allows for a better visualization of the underlying structures of the phantom, as high noise levels in the MC projections could make it difficult to distinguish the actual signal from noise. However, it should be kept in mind that the model should not be directly compared to the raytracing or MC data using this ground truth data, as it has not been trained on data with the same level of accuracy. Because of this, the Transformer network has not had the oppurtunity to learn smaller details in the data.
- **Atypical phantoms** Four different phantoms with unseen types of geometries and intensity distributions are used to assess the robustness of the model. This includes a phantom with a hexagonal and ellipsoidal insert, a phantom with a box-shape water surrounding instead of the usual cylinder shape, a phantom without a surrounding water phantom and a phantom that is created from an actual chest CT image, processed to have the same HU value range and cylindrical shape as the training phantoms. The HU value distribution is different than that of the training phantoms, as it contains considerably more unique values than the training data. The atypical phantoms used for testing are displayed in figure 4.9. In these comparisons, no MC ground truth projections are used due to computational limitations. Instead, the raytracing projections give a general indication of the supposed shape of the projections. Despite the fact that these raytracing projections are less accurate than MC projections, using them to compare the Transformer projections regarding these atypical phantoms may provide a general indication on the robustness of the model.

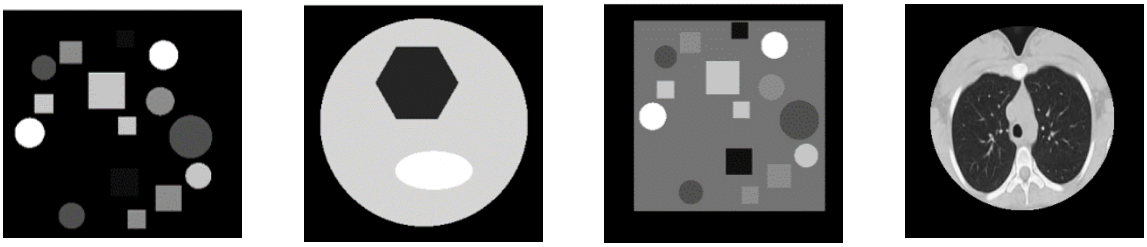


Figure 4.9: Phantoms used to evaluate the robustness of the neural network. The phantoms each contain an aspect that does not appear in the training data. One phantom is included that does not possess a water cylinder surrounding the phantom inserts (a), one phantom contains a hexagonal and ellipsoidal insert, which are unseen types of geometries (b), one phantom consists of a box-shaped water phantom surrounding the phantom inserts as opposed to the cylindrical shape in the training data (c) and one phantom is produced using an actual chest CT image. For this image, the value range is matched to the training data and a mask is applied in the shape of the water cylinder that is present in the training data. This phantom includes unseen geometries as well as an unseen HU value distribution (d).

4.4.2. Comparison with other methods

Raytracing and SIRT algorithm

The raytracing forward projector and the SIRT algorithm used in the evaluation are supplied by the Astra Toolbox [45], which supports GPU-accelerated forward- and backward projection operations as well as a number of reconstruction algorithms for several types of user-defined 2D and 3D geometries. The forward projection operator is configured based on the CT system geometry of the GATE simulations. The input arrays contain the linear attenuation coefficients of the materials which are given in Table 4.2. The Astra forward projection operator used in the SIRT algorithm is replaced with the neural network. Projections for all angles are generated in the forward projection step. The neural network is only able to predict projections for a fixed angle due to the image array being processed as a sequence of rows. To generate projections with different angles, the input array must be rotated. These rotations are performed with the use of nearest-neighbor interpolation, which is the same method used in generating the input images.

The purpose of this evaluation is to determine the performance of the network within an IR algorithm irrespective of the robustness of the training data. Because of this, the prediction of nonsensical results due to insufficiently robust training data must be avoided by ensuring that the input to the neural

network in between iterations resembles the training data.

To achieve this, the following adaptations to the algorithm are made:

- The initial estimate, which is normally an empty image array, is substituted with an image array containing an empty water phantom similar to the phantoms used in the training set
- A mask is applied between iterations, before the forward projection operation. This changes the values outside of the cylindrical phantom to the background value of air.

The CNR for both a low and high contrast region of interest (ROI) is measured every iteration. Using these measurements the peak CNR and the convergence rate, which is the number of iterations needed to reach the peak CNR, is determined for both algorithms and compared. The images are compared qualitatively as well, based on similarity to the ground truth image and presence of noise and artifacts.

Comparison with Two-Angle Convolution (TAC) network
A Two-Angle Convolution (TAC) was recently developed [11] with the aim to improve raytracing projections for use in CT image reconstruction. This method does not generate forward projections from CT image data but its input consists of two raytracing projections of the same phantom with a projection angle difference of 90 degrees. The network then predicts an improved projection for both angles. The average NRMSE (described in paragraph 4.4.3) of the Transformer network and TAC network are compared as well as the NRMSE for the Astra raytracer. The input used to measure the NRMSE belong to the same data set as those used for the TAC network, but they are not the exact same samples.

4.4.3. Evaluation metrics

- *Average Normalized Root Mean Square Error*
  The normalized root-mean square error ($NRMSE$) is used as a measure of the error between the projections and the MC ground truth. It is comparable to the MSE , which is used as an evaluation measure during the training of the model. However, the $NRMSE$ provides a more intuitive measure as it contains the same unit of measurement as the projection data. By normalizing, the measure can be expressed as a percentage. The $NRMSE$ is calculated with equation 4.3. To compare the networks, the average $NRMSE$ over the entire data set is calculated.

$$NRMSE = \frac{\Sigma(S_i - O_i)^2}{\Sigma O_i^2} \tag{4.3}$$

  where S_I and O_i are the values of element i for the prediction and ground truth respectively.

- *Contrast-to-Noise Ratio*
  The contrast-to-noise ratio (CNR) is a measure of the noise level in a certain region of interest (ROI). The CNR is calculated for ROIs with high and low contrast. High contrast regions are typically regions with materials that have very high or low density, resulting in a large intensity difference between the object and the background. Low contrast regions contain densities similar to the background material, making them harder to distinguish. It is calculated as follows:

$$CNR = \frac{S_{ROI} - S_{Background}}{\sigma_{background}}, \tag{4.4}$$

  where S_{ROI} and $S_{Background}$ are the average pixel value in the ROI and background and $\sigma_{background}$ is the standard deviation of the background values, given by:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n-1}} \tag{4.5}$$

  For each image reconstruction, the CNR is calculated for both a high- and low-contrast ROI. The high-contrast region contains a material with high-density and the low-contrast ROI contains a material with a density close to that of the background. The background region used in this measurement is a region in the phantom without a material insert, thus containing only water.

- *Structural Similarity Index Measure*
  The Structural Simularity Index Measure (SSIM) [40] is a measure of the similarity between two images using the luminance, contrast and structural components. It divides the images into regions with a certain size. The calculated SSIM for all regions are then combined, producing the final SSIM value for the image. The mean, variance and covariance of pixels within windows with a size of 11x11 pixels in the predicted and ground truth image are computed using equation 4.6, after which the mean SSIM is used to evaluate the images.

$$SSIM(x,y) = L(x,y) \cdot C(x,y) \cdot S(x,y) \tag{4.6}$$

  with

$$L(x,y) = \frac{2\mu_x\mu_y + C_1}{2\mu_x^2 + \mu_y^2 + C_1}$$

$$C(x,y) = \frac{2\sigma_x\sigma_y + C_2}{2\sigma_x^2 + \sigma_y^2 + C_2}$$

$$S(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

  where μ_x, μ_y and σ_x, σ_y are the mean and standard deviation of the values in the reconstructed image x and the ground truth image y and σ_{xy} the co-variance over a window in these images and $C1, C2, C3$ are constants.

5

Results

In this chapter, the Transformer network is evaluated based on its performance on unseen data compared with a conventional raytracing method [45] and the MC ground truth. The first section describes the results of the hyperparameter grid search. The effects of adjusting the various parameters are shown and the best performing neural network configuration is presented. This final model is trained on a larger data set and evaluated in various ways, which is described in the second section. First, the separate predictions of the network are compared to the MC ground truth, as well as a conventional raytracing method. Input phantoms that are similar to the training data are tested as well as atypical phantoms, which are used to indicate the robustness of the Transformer network. The final section displays the reconstructed images for two phantoms, generated with a SIRT algorithm using both Transformer and raytracing forward projections. They are evaluated with qualitative and quantitative metrics, comparing them based on appearance of artifacts and similarity to the ground truth images.

5.1. Hyperparameter Optimization

5.1.1. Grid search

The results of the hyperparameter search are displayed in Table 5.1. The best performing architecture resulting from this search produced an $MSE = 7.09 \cdot 10^{-5}$, with an average computation time of $23\ ms$ per prediction. The architecture contains $K = 8$ filters in the final convolution layer of the convolution encoder, $N_H = 32$ number of Transformer heads and $N_T = 1$ number of Transformer blocks in the Transformer encoder. The average MSE is in the same order of magnitude for all hyperparameter configurations. For all three parameters, a higher value produces a lower MSE in three out of four cases. The results show that the best and worst performing configuration both have $N_T = 1$ which suggests that the number of Transformer blocks has little impact on the performance. Additionally, increasing the value for any of the tuned parameters significantly increases the number of weights, while the corresponding improvement in the MSE is not as significant and may even be counterproductive.

Table 5.1: Results of the hyperparameter grid search with the best configuration highlighted in gray. The parameters that were tuned are the number of filters in the final convolution layer of the convolutional encoder K, the number of Transformer heads N_H and the number of Transformer blocks N_T. The table shows the resulting MSE and computational time for each configuration, as well as the number of weights corresponding to each combination, indicating the computational expense of the configuration. Increasing the value of any parameter leads to a significant increase in the number of weights and computation time while the corresponding change in MSE is not proportional.

K	N_H	N_T	Time (ms)	Weights	MSE
8	16	1	16 ms	13.1×10^{6}	9.79×10^{-5}
8	16	2	33 ms	25.8×10^{6}	7.59×10^{-5}
8	32	1	23 ms	25.5×10^{6}	7.09×10^{-5}
8	32	2	37 ms	50.6×10^{6}	7.78×10^{-5}
12	16	1	22 ms	29.2×10^{6}	8.94×10^{-5}
12	16	2	36 ms	57.9×10^{6}	7.56×10^{-5}
12	32	1	36 ms	57.0×10^{6}	7.62×10^{-5}
12	32	2	65 ms	113.7×10^{6}	7.28×10^{-5}

5.2. Neural network performance

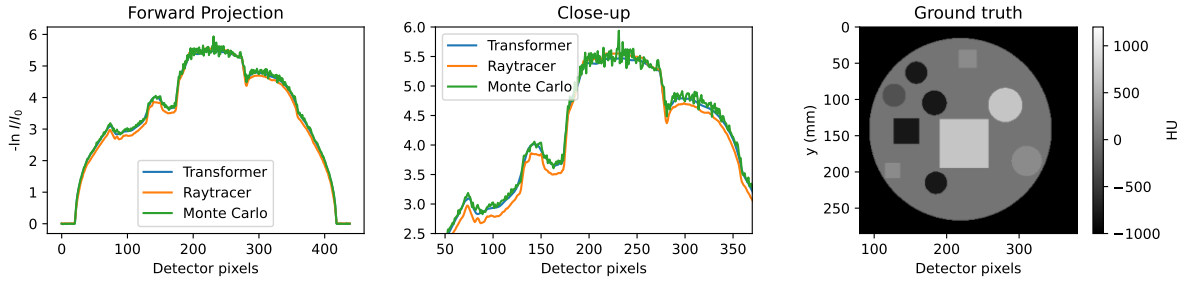
In this section, the predictions of the optimized and trained neural network are evaluated with the methods described in section 4.4. The first paragraph presents the results of the model tested on several unseen phantoms with similar features as the training phantoms. These are compared with raytracer projections, MC projections that are similar to the MC projections from the training data set and low-noise MC projections generated with a higher particle count. Next, the predictions obtained from four atypical phantoms with unseen geometries and value distributions are presented along with the corresponding raytracer projections. This is followed by the CT image reconstructions generated by the SIRT algorithms using a raytracer and a Transformer forward projection step, which are evaluated with the metrics outlined in paragraph 4.4.3.

5.2.1. Forward projections

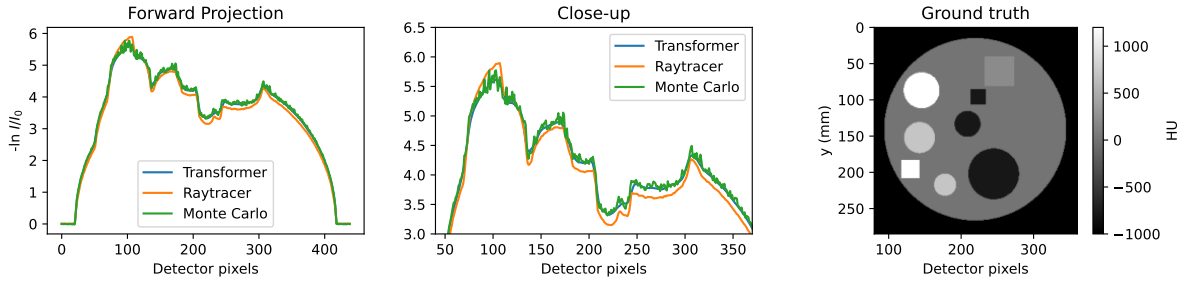
The NRMSE and prediction time of the neural network based on an unseen test data set of 1320 samples is displayed in Table 5.2 along with the values resulting from the raytracer forward projections on the same data set. The input samples for TAC network predictions are extracted from the same data set, but not the exact same samples were used. The raytracing algorithm predicts all angles at once. The displayed value for the raytracer prediction time of 0.033 ms is the total computation time divided by the number of angles (180). The results show that the raytracer is almost 1000 times faster than the Transformer while the average NRMSE is 1.475 percentage points higher. Compared to the TAC network, the Transformer has an average NRMSE that is 0.365 percentage points lower.

Table 5.2: The average MSE and NRMSE resulting from the forward projections of the raytracing and Transformer network over an unseen data set containing 4359 samples, calculated with Monte Carlo ground truth data. The raytracer predicts all angles at once. The displayed value for the prediction time is the computation time divided by the number of angles (180). The results show that the raytracer prediction time is a factor of almost 1000 faster with an NRMSE that is 1.475 percentage points higher. The TAC network produces a NRMSE that is 0.365 percentage points higher than that of the Transformer.

	Raytracer	Transformer	Two-Angle Convolution
Avg. NRMSE (%)	2.20	0.725	1.09
Prediction time (ms)	0.033	31	-



(a)



(b)

Figure 5.1: Transformer predictions on unseen phantoms compared to Monte Carlo (MC) and raytracer projections. The Transformer projection is very similar to the MC ground truth. The raytracer predicts an accurate shape, although the values are too low for most parts of the projections. In the closeup of the projection displayed in (b), the raytracer projection shows signs of beam hardening, as it predicts relatively high values on the left side (between pixels 50 and 125) which correspond to the white high density regions shown in the ground truth phantom at the right side of the image.

Figure 5.1 displays two examples of neural network predictions on unseen input data, as well as the corresponding MC ground truth projection, raytracer projection and phantom input image. A complete projection is depicted along with an enlarged area which shows more detail. The X-ray source and detector are located respectively above and below the phantoms. In each example displayed in Figure 5.1, the projection of the neural network shows a large resemblance to the MC projection, while the raytracing projection displays mostly too low, and in some cases too high values. This is clearly visible in the closeup of Figure 5.1b, where the raytracer projection computes a too high value on the left side of the projection between detector pixel 75 and 125, which can likely be attributed to the high-density region of the left side of the phantom between pixel 100 and 125.
Figure 5.2 displays the results of the Transformer predictions on unseen phantoms compared to low-noise MC and raytracer projections. The MC projections have a higher resolution, depicting even very small peaks as shown in the closeup of Figure 5.2b. The Transformer predictions show less details than the MC projections, which is to be expected as the Transformer model is trained on ground truth data containing more noise. The raytracer predicts small details well, although the values are too low for most parts of the projections.

5.3. Atypical phantoms

The robustness of the model is tested on several input arrays containing phantoms with varying unseen geometries and material densities. The predictions are compared only to raytracing projections of the same phantoms, which are not as accurate as MC projections that are used as ground truth for the results in previous paragraph. Despite this, they give a general idea of the supposed shape of the projections and using them in the comparison can give an approximation of the performance of the neural network.
Figure 5.3 displays the projections for a phantom without a water cylinder surrounding the other materials, which is a consistent factor in the training data. The network prediction is consistently higher than the raytracing projection and the edges are shaped similarly to projections that include the surrounding water cylinder.
Figure 5.4 show the projections for a cylinder containing a hexagonal and ellipsoidal insert. The network

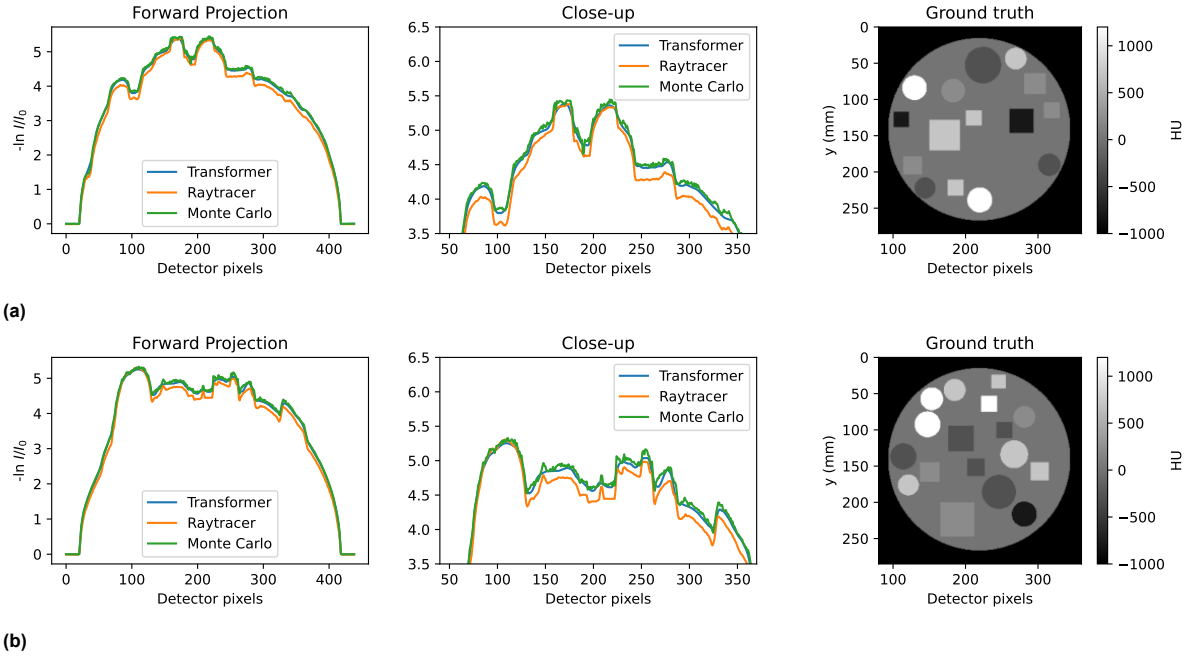


Figure 5.2: Transformer predictions on unseen phantoms compared to low-noise MC and raytracer projections. The MC projections have a higher resolution, depicting even very small peaks as shown in the closeup of (b). The Transformer predictions show less details than the MC projections, which is to be expected as the Transformer model is trained on ground truth data containing more noise. The raytracer predicts small details well, although the values are too low for most parts of the projections.

prediction shows a large resemblance to the raytracing projection, with the slightly higher intensities being the largest difference.
Figure 5.5 displays the projections for a phantom with a box-shaped water phantom instead of the cylindrical shaped phantoms used during training. Despite the box-shaped phantom, neural network prediction shows the same shape around the edges as projections corresponding to cylindrical phantoms. The difference is less visible in the center, where the network prediction is more similar to the raytracing projection.
In Figure 5.6, projections are shown for an actual chest CT image that is processed to have the same value range and outer shape as the training data. The neural network projections have roughly the same shape as the raytracing projections, with lower values for areas with relatively high density. This is best visible in the center and at the edges.

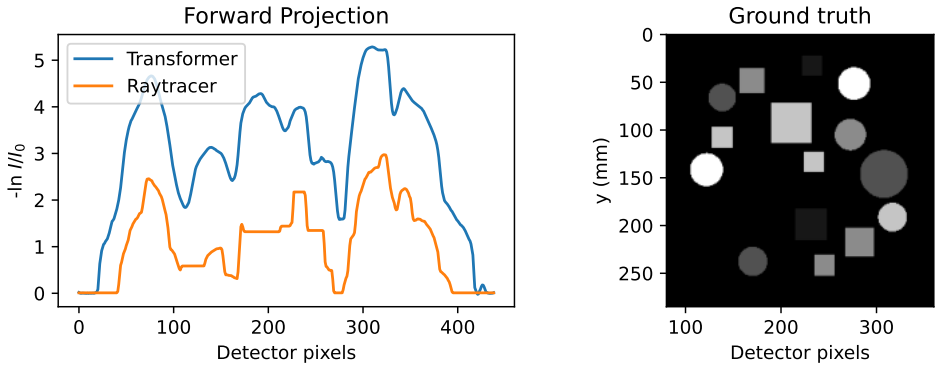


Figure 5.3: Projections (a) of the neural network and raytracer of a phantom (b) that is missing the usual water cylinder surrounding the phantom inserts. The network projection displays significantly higher values than the raytracing projection and is shaped similar to projections of phantoms that do contain the water cylinder.

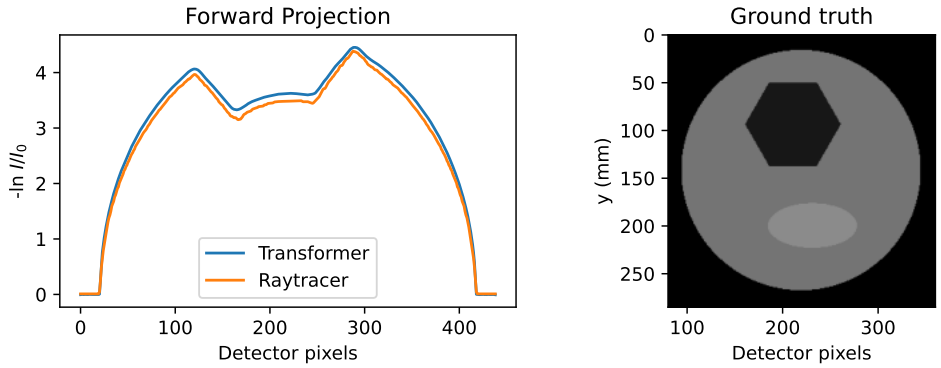


Figure 5.4: Projections (a) of the neural network and raytracer of a phantom (b) that includes inserts with a hexagonal and a ellipsoidal geometry, which are not present in the training data. The projections of the neural network and the raytracer are similar, with the raytracing projection displaying slightly lower values.

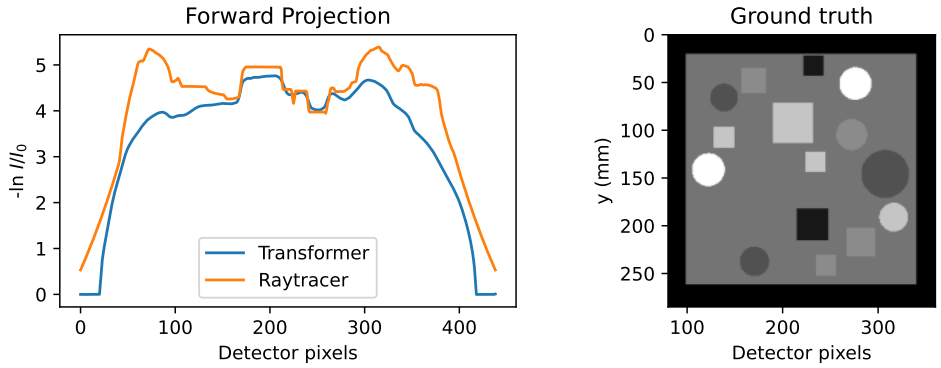
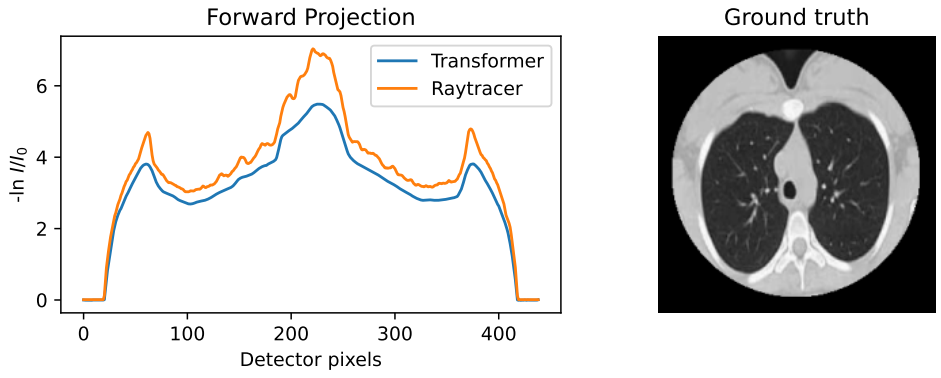


Figure 5.5: Projections of the neural network and raytracer of a box-shaped water phantom surrounding the inserts instead of the usual cylindrical shape. The projections of the neural network and raytracer are shaped differently. This is especially noticeable at the edges, where the neural network projection is shaped similarly as projections of a cylindrical shaped phantom.



(a)

Figure 5.6: Projections (a) of the neural network and raytracer of an input array (b) created from an actual chest scan obtained from [2]. The image is processed to lie in the same range as the input training data and a circular mask is applied to recreate the shape of the water phantom present in all training data. The projections of the neural network and the raytracer have a similar shape, with the raytracer displaying higher values especially in the center, which includes a relatively large amount of dense material.

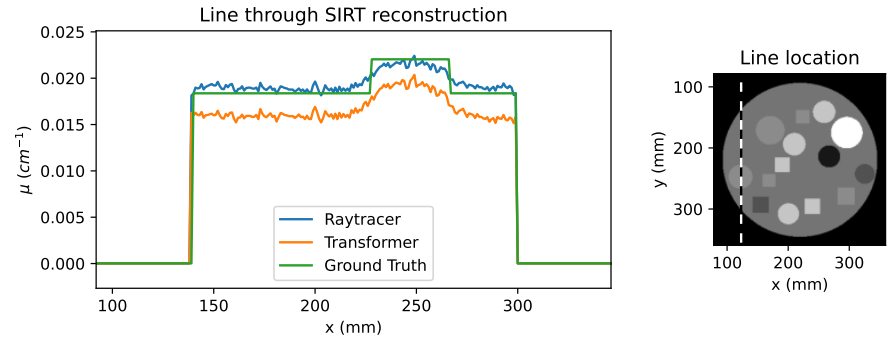
5.4. Simultaneous Iterative Reconstruction Technique

Two phantoms have been reconstructed using both a Transformer and a Raytracer SIRT reconstruction with projection data for 180 angles. For both phantoms, the convergence of the algorithms based on the SSIM and CNR in the low- and high-contrast region is displayed in Figure 5.7. The highest values measured for the CNR and SSIM are displayed in Table 5.3 along with the corresponding iteration numbers.
Based on the SSIM, the TF reconstruction converges in 23 and 2 iterations for phantom 1 and 2 respectively, which is faster than the RT reconstruction which converges after 27 and 27 iterations for phantom 1 and 2. The difference in SSIM values is negligible. The values of the SSIM for the raytracer and Transformer are the same for phantom 1 (94.30 %). For phantom 2, the raytracer produces a value that is 0.22 % higher compared to the Transformer SSIM. Based on the CNR measurements for both phantoms, the raytracer reconstruction converges faster in both the low- and high-contrast ROI, with the biggest difference for the high-contrast ROI of phantom 2 in which the raytracer converges 7 iterations earlier than the Transformer. The SIRT reconstructions of both methods for phantom 1 and 2 are shown in Figures 5.8 and 5.9, along with the ground truth images. While the reconstructions of the raytracer and Transformer show a large resemblance, some differences are noticeable. One example is the intensity difference in the two images, as the Transformer reconstruction shows a slightly darker background compared to the raytracing image. This is in line with the prediction results shown in Figure 5.1, where the Transformer projection is consistently higher than the raytracer projection. This corresponds to lower attenuation values which translates to darker values in the image.

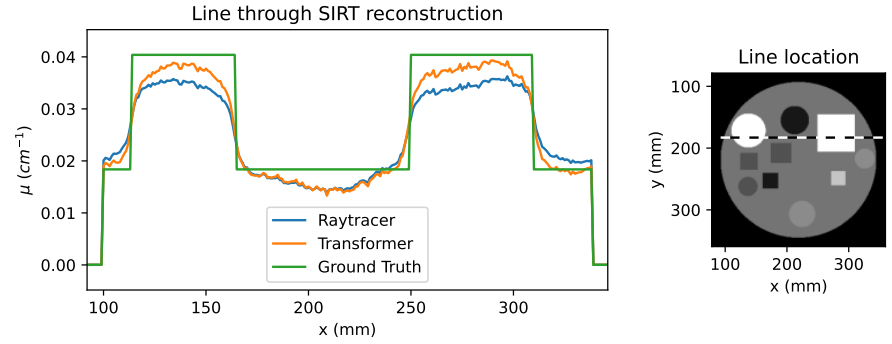
Another difference between the reconstructions is that the Transformer reconstruction seems to distinguish materials in low-intensity differences better. An example is shown in 5.8a in which a low-contrast region is enlarged, showing clearer edges and a higher contrast in the Transformer reconstruction.
Figure 5.10 shows a line profile on the same location through the reconstructed images for both methods as well as the ground truth phantom at the iteration of the highest CNR in the low-contrast ROI. Both Figures 5.10a and 5.10b show that the Transformer prediction is lower than the ground truth in almost every region, although its differences in intensity is very similar to those in the ground truth image. In Figure 5.10a, the raytracer projection is more similar to the ground truth with regard to intensity, but it shows a smaller increase in value than both the ground truth and Transformer projections between pixel 225 and 275, where the line goes through a different material. This difference is more noticeable in Figure 5.10b, where it goes through a high-contrast region. Here, the intensities of the raytracer and neural network reconstructions are similar around the edges but diverge with higher densities. The RT reconstruction displays beam hardening, as the relative intensity for higher densities is much lower than for the ground truth, while the Transformer reconstruction displays the same value distribution as the ground truth image.

Table 5.3: Evaluation metrics of the reconstructed images using the SIRT algorithm with a raytracing and Transformer forward projection model. Values are displayed for the highest CNR in both high- and low-contrast ROIs (displayed in Figure 5.7) and the corresponding iteration number, as well as the SSIM values calculated for each image corresponding to this iteration number. For both phantoms reconstructed with the Transformer algorithm, the reconstruction with the highest CNR has the same iteration number for both ROIs. This is the same for the Raytracer reconstructions of phantom 2, while the iteration number only differs by one for the reconstructions of phantom 2 with the highest CNR.

	Raytracer		**Transformer**	
	ROI 1 high-contrast	ROI 2 low-contrast	ROI 1 high-contrast	ROI 2 low-contrast
Phantom 1				
Highest CNR	37.40	6.88	39.01	7.10
Iteration	39	38	41	41
SSIM (%)	94.30		94.30	
Iteration	27	-	23	-
Phantom 2				
Highest CNR	41.94	8.28	39.80	7.35
Iteration	34	34	41	41
SSIM (%)	94.03	-	93.81	-
Iteration	27	-	2	-



(a)



(b)

Figure 5.10: Line through the location shown in the phantom next to the line profile for phantom 1 (a) and 2 (b) for both SIRT reconstructions, at the iteration of the highest CNR in the low-contrast ROI, using the neural network and a raytracer. While the Transformer reconstruction displays values that are lower than the ground truth, it does not display signs of beam hardening. The raytracer projection shows beam hardening whhich is especialy noticeable in (b), where there is a relative lower increase in intensity on the sides of the line profile, corresponding to the white high density regions shown in the ground truth phantom.

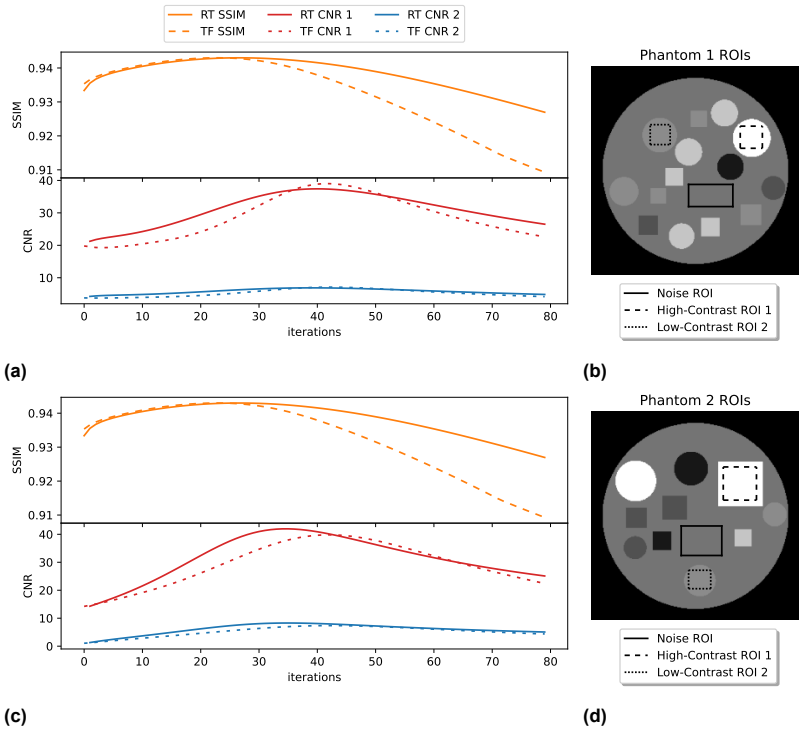
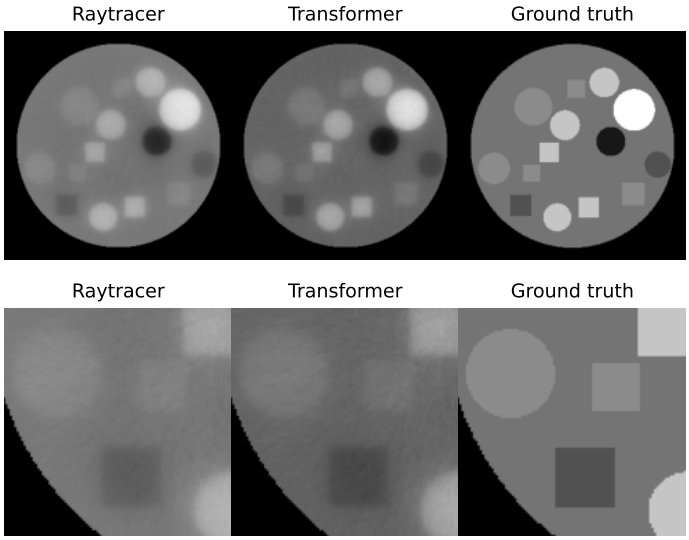
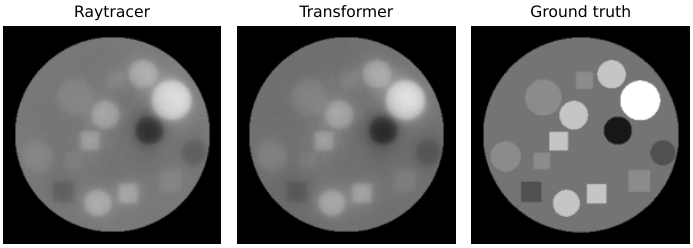


Figure 5.7: Convergence of the raytracing (RT) and Transformer (TF) algorithm measured with CNR (a),(c) for the reconstruction of two phantoms in ROIs illustrated with the ground truth phantom in (b) and (d). For both ROIs in phantom 1, the Transformer achieves a higher CNR while the raytracer shows a faster convergence.In phantom 2, the raytracer achieves both a higher CNR as a faster convergence. The SSIM values for both algorithms decline rapidly. The highest SSIM for phantom 1 is the same for both algorithms while for phantom 2, the raytracer value slightly higher. The values and convergence rates are displayed in Table 5.3

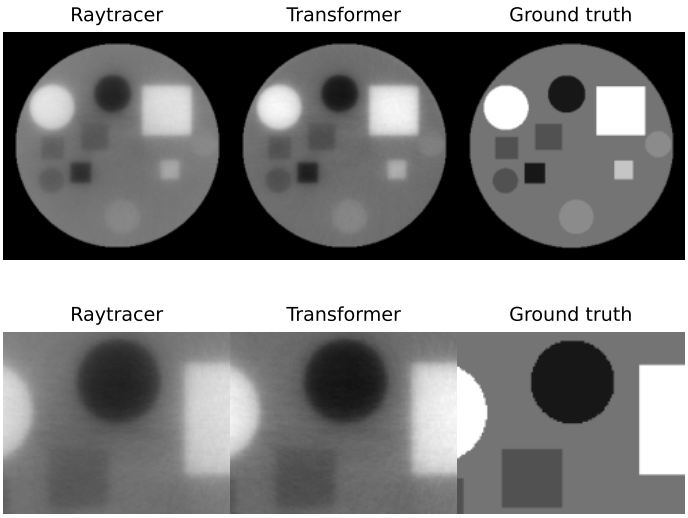


(a) Reconstructions with best CNR for low-contrast ROI 2

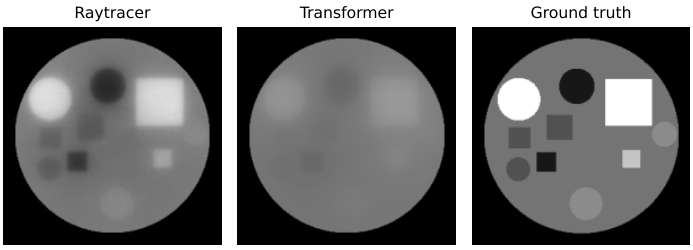


(b) Reconstructions with best SSIM

Figure 5.8: SIRT reconstructions of phantom 1, generated with a SIRT algorithm implementing a Transformer and raytracing forward projection model. The reconstructions in 5.8a correspond to the iteration number with the highest CNR in the low-contrast ROI displayed in Figure 5.7b. The reconstructions in 5.8b display the reconstructions corresponding to the highest SSIM. The CNR and SSIM values are presented in Table 5.3.



(a) Reconstructions with best CNR for low-contrast ROI 2



(b) Reconstructions with best SSIM

Figure 5.9: SIRT reconstructions of phantom 2, generated with a SIRT algorithm implementing a Transformer and raytracing forward projection model. The reconstructions in 5.9a correspond to the iteration number with the highest CNR in the low-contrast ROI displayed in Figure 5.7d. The reconstructions in 5.9b display the reconstructions corresponding to the highest SSIM. The CNR and SSIM values are presented in Table 5.3. The Transformer reaches the highest SSIM value in the second iteration which results in an image of poor quality.

6

Discussion

The Transformer model has shown promising results. For the type of data that is has been trained on, it is able to reliably model the particle transport of photons through a phantom and predict a detector response that is very similar to the MC ground truth. While smaller details and edges are lost, the predictions match the MC intensity very well. This is even the case for high-density materials, where the Transformer predictions do not display the beam-hardening artifacts that are visible on the raytracing projections. This indicates that the model is not simply learning a raytracing-like approach, but also incorporating nonlinear physical effects.

6.1. Prediction quality

The Transformer model performs well compared to the raytracer based on the average NRMSE, which is with a value of 0.725% more than three times lower than that of the raytracer, which has an average NRMSE of 2.20%. The TAC network produces an average NRMSE of 1.09% which is lower than the raytracing method but not as good as the Transformer prediction. The better performance of the Transformer is clearly visible in the projections, in which the Transformer projections has an intensity distribution that matches the ground truth much better than the raytracer projections. For the Transformer predictions, the intensity values are very similar to the MC values for all material densities, contrary to the raytracer projections which display signs of beam-hardening in almost all projections. Significant differences are visible, for example in the closeup of Figure 5.2b in which the raytracer projection shows a beam hardening effect by producing high values in high-density regions on the left side of the image, while the values are too low in all other regions. For the Transformer, the predictions are similar to the ground truth in all regions.

With regard to resolution, the raytracer outperforms the Transformer model. Small dips and peaks are absent in the Transformer projections and sharp edges are smoothed, as shown in figure 6.2, which displays a close-up of a Transformer prediction compared to the raytracer and ground truth.
There are several possible solutions to increase the performance in this area. For one, using a training data set that contains MC ground truth with a lower noise level could increase the resolution of the Transformer predictions. Currently, the noise in the MC projections is still prominent, especially for high-density materials for which less particles are detected. This effect is displayed in figure 6.1. When smaller details are not consistently showing up in the ground truth data, it is difficult for the Transformer network to learn to predict them.

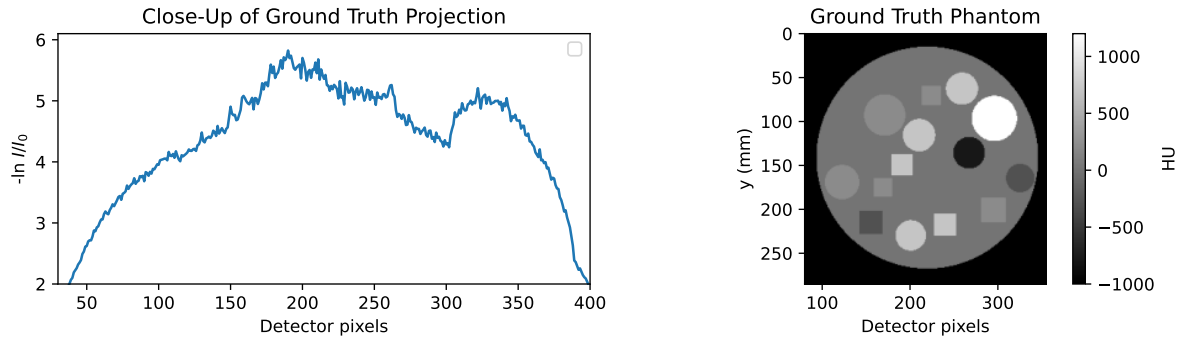


Figure 6.1: Close-up of a Monte Carlo projection and corresponding phantom illustrating the relative noise difference for different attenuation levels. The noise is much more noticeable in the center of the projection, when there is more attenuation due to the cylindrical geometry of the phantom.

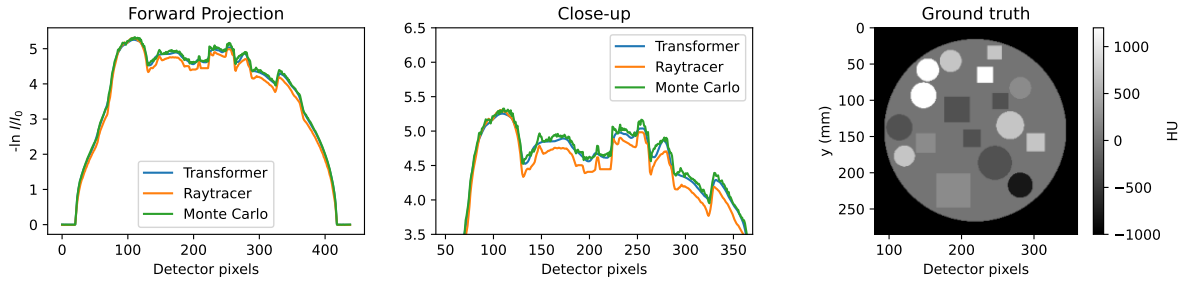


Figure 6.2: Transformer prediction, low-noise MC projection and corresponding phantom illustrating the prediction quality for small dips and peaks in the MC data.

The lower resolution of the Transformer predictions could be caused by the discretization of the continuous phantom as well. Image rotation for angles $\theta \neq n \cdot 90°$ results in a lower resolution, as well as position and intensity variations depending on the interpolation method. The images in the input data are interpolated using nearest-neighbor interpolation, while the Astra toolbox raytracer uses linear interpolation in the FP operation, which generally produces better results [37]. Using a different interpolation method when creating the input data could improve the resolution of the predictions [16], [37] , as well as higher resolution input data. However, increasing the resolution should be done with caution as it significantly increases the computation time. To counter this, extra downsampling operations could be implemented in the network architecture which reduce the size of the network.

6.2. Model Robustness

Presented with unseen data containing noise, smoothing or new data values, the model performs remarkably well. The model is able to predict geometric shapes different from the training data, providing that they are contained within a specific cylindrical phantom. When the surrounding water phantom is shaped differently or missing, predictions are less accurate as the model is still predicting the cylindrical shape. This is due to the biased training data, in which the cylindrical shape is a consistent factor for each sample. Judging by the model's performance on new geometries within these cylinders, this bias can be resolved with more diverse training data that contains various geometries for the surrounding water phantom. Another addition could be the use of heterogeneous material that is more similar to biological tissue, as well as the use of a larger amount of HU values. Additionally, a larger range of HU values could be used. Currently, the largest value is $1200\ HU$ for material 5, which is still well within the range of human tissue, as the values for bone might even reach up to $3000\ HU$. Besides better generalizability, expanding the training data could allow more features of the model as well. Training data containing different energy spectra, CT system geometries, particles or physical processes could create a model that is applicable in a huge variety of situations.

6.3. IR implementation

Image quality

While the different metrics used to assess the images produce a different outcome, the values of the Transformer and raytracer reconstructions are similar together. For phantom 1, the SSIM values are the same for both methods while for phantom 2, the SSIM is 0.22% higher for the raytracer compared to the Transformer, which is only a slight difference. The SSIM value for the Transformer decreases rapidly after the second iteration while the image quality seems to still improve. This suggests that the SSIM value should be calculated differently. The ROIs that are used to calculate the SSIM have a large impact on the result and should be chosen with consideration [36]. The SSIM in this comparison was calculated over the entire image, including the masked parts around the phantom which is the same for both the reconstructions and the ground truth. A better measure might have been to calculate the SSIM based on only the values inside the phantom. Another adjustment could be to normalize the images before computing the SSIM, which brings the images to the same range but keeps the distributions of the image the same.

When using the CNR as a measure of image quality, the results are different for the two reconstructed phantoms. For phantom 1, Transformer achieves a higher value for both the low-contrast and the high-contrast ROI, while for phantom 2 the CNR of the Transformer is worse for both ROIs compared to the raytracer. Based on the CNR measures, it is not clear which algorithm performs better. Additional evaluation measures should be calculated to compare the Transformer and raytracing algorithm quantitatively. The focus should be on calculating a correct SSIM value. While CNR has been a widely used metric to assess CT image quality, it has been suggested that the SSIM is a more suitable metric for IR-based image reconstructions due to their different noise reduction characteristics [40].

Convergence rate

Using CNR as a measurement, the Transformer SIRT algorithm does not converge as fast as the raytracer algorithm, with the Transformer algorithm needing an additional 2 and 7 iterations on top of the 39 and 34 iterations of the raytracing algorithm.

When the SSIM is used to measure performance, the Transformer converges faster with 23 and 2 iterations compared to the 27 and 27 iterations of the raytracer for phantom 1 and 2 respectively. Given that the SIRT algorithm has been optimized for use with a raytracing FP operator, it would be expected that the raytracer performs better regarding the convergence rate.

The raytracing BP operator and correction factors C and R (shown in Equation 2.11) used in the algorithm are computed with the system matrix corresponding to the raytracer FP operation. The type of backprojector that is used in the algorithm can have a large impact on the convergence and image quality [73]. When the algorithm is adapted to the Transformer predictions, an even faster convergence rate might be reached. One example to adjust the algorithm is to implement a *relaxation factor* which assigns a weight to the correction factors in the algorithm and has shown to have a significant impact on convergence rate [64].

The model has shown potential to be implemented as a forward projector in IR methods. The images that are reconstructed by the Transformer model do not display the beam-hardening effects that are associated with the raytracer, although the values of the Transformer images are consistently too low. This could be related to the type of BP operator and correction factors that are used in the algorithm as well. The TAC model has already shown that its improvements on raytracing projections can increase the convergence rate of the network [11]. This suggests that using a differently tuned reconstruction algorithm might improve convergence for the Transformer as well.

6.4. Prediction speed

The speed of the model predictions is another important aspect to consider, as it is one of the main drawbacks of current iterative methods. At the moment, the Transformer model is orders of magnitude slower than the raytracing algorithm. On average, the raytracing algorithm uses $0.033\ ms$ to compute a projection, while the Transformer model needs $31\ ms$, which is a factor of almost 1000 higher.

However, the model has not yet been optimized for speed. There are several adjustments that can be made to increase the speed. The focus should be on decreasing the amount of learnable parameters while still obtaining satisfactory results, for example by changing the amount and type of layers, serializing the model [46] or removing connections from the model after training (pruning) [10].
The results from the parameter search show only slight differences between the different configurations, while the differences in prediction time were significant. This suggests that even more reduction in prediction time might be attainable, combined with only a slight decrease in accuracy. With this in mind, a more extensive parameter optimization should be carried out to determine a more efficient architecture, focusing on a combination of speed and performance.
To improve the speed of the IR algorithm with the Transformer implementation, rotation of the intermediate images could be performed in part simultaneously with model predictions to minimize GPU down-time.

7

Conclusion

The Transformer architecture has revolutionized deep learning, proving itself useful in a wide range of fields. This research shows its versatility once again, demonstrating the ability to model photon particle transport in CT imaging from source to detector. With a CT image as a line-by-line input sequence in the direction of the X-ray beam, the CT Transformer successfully predicts the resulting detector response using causal self-attention. The model predictions do not display beam-hardening artifacts, contrary to conventional raytracing algorithms. The model's robustness is mainly limited by its training characteristics. It is able to predict unseen geometries but is restricted by the shape of the surrounding phantom present in each sample image. This is very likely solved by removing biases from the training data. Given that the model is trained on only a few different Hounsfield units and with distinct boundaries between shapes, it performs remarkably well on data that is smoothed or contains noise. The performance of the model is lacking with regard to resolution. A potential factor in this is the interpolation method used for generating the CT input data which results in resolution loss. This can be resolved with other interpolation methods and higher resolution input, though these adjustments increase the computational expense. Another solution would be the use of better quality ground truth data, which could improve the prediction of small details without increasing the computational cost of the model.
The model has been implemented in an iterative reconstruction algorithm, producing better results with regard to beam hardening than conventional raytracing methods but lacking in reconstruction speed. This can be improved with adjustments to the backprojection operator and correction factors. The largest improvements can be achieved with minimizing the prediction speed of the model, which is almost a factor of 1000 higher than the raytracing algorithm. This can be achieved by reducing the size of the model, for which another parameter optimization needs to be performed. With this, the optimal trade-off between reconstruction speed and image quality can be determined.

7.1. Future work

Since iDoTa has performed photon transport modelling in 3D using the same type of architecture, the CT Transformer is likely capable of this as well. The development of fast and accurate 3D forward projections is an important challenge in current research, as this could further the realization of real-time imaging methods. In radiotherapy, real-time corrections during dose delivery could increase precision and in turn reduce overall radiation dose. This type of adaptive radiotherapy would require CT input images to be obtained at real-time speeds as well. If the reconstruction time can be reduced with the suggested improvements, the Transformer model might be able to facilitate these reconstructions.
To implement this model in adaptive radiotherapy, it would need to be for 3D CBCT input to predict a projection image. CBCT has the ability to acquire projection data during treatment because the detector and source only need to perform one rotation without moving along the length of the patient, which would be the case for 3D (helical) CT. This way the CT system can be integrated with the radiotherapy system [56]. A potential challenge of this is that the CT transformer model needs to receive the entire CT image as input whereas DoTA and iDoTA only use cropped parts of the CT image. On top of that, an extra dimension has large implications for the computational resources required to run the model. At this point, the 2D input to the CT transformer is already larger than the 3D input to the original DoTA.

To process a CT volume large enough to be clinically relevant, the size of the model should be reduced significantly. This means a trade-off must be made regarding the prediction quality and speed of the model.

Other than real-time imaging, the CT Transformer could improve reconstruction quality in other imaging methods that include forward projections, for example in spectral CT methods like photon-counting CT (PCCT) or Dual-Energy CT. PCCT detects individual particle energies, obtaining a separate energy spectrum for each detector pixel compared to conventional energy-integrating detectors that register only the sum of the detected particle energies. Dual-energy CT (DECT) uses two different energy spectra to distinguish materials based on their attenuation property at different energies. Adjustments to the model architecture would need to be made to accommodate these different input and output types.

Particle transport is present in a wide range of fields which could benefit from a deep learning particle transport model. Besides applications in medical physics, such a model might be helpful in fluid and thermal dynamics, significantly improving computation times compared to regularly used MC methods. Other possibilities include neutron modelling used for development of nuclear facilities, transport of aerosol particles to determine air quality, space radiation modelling to aid in spacecraft design or particle transport in materials science. Each of these applications would require several adjustments and an appropriate training data set, but the success and versatility demonstrated by the Transformer in recent years encourages to find out its full potential.

Bibliography

[1] Md Manjurul Ahsan, Shahana Akter Luna, and Zahed Siddique. “Machine-Learning-Based Disease Diagnosis: A Comprehensive Review”. In: *Healthcare* 10.3 (Mar. 2022), p. 541. ISSN: 2227-9032. DOI: 10.3390/healthcare10030541. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/ (visited on 05/08/2023).

[2] Brad Albertina et al. *The Cancer Genome Atlas Lung Adenocarcinoma Collection (TCGA-LUAD)*. Type: dataset. 2016. DOI: 10.7937/K9/TCIA.2016.JGNIHEP5. URL: https://wiki.cancerimagingarchive.net/x/wgBp (visited on 05/08/2023).

[3] Filippo Arcadu, Marco Stampanoni, and Federica Marone. *On the crucial impact of the coupling projector-backprojector in iterative tomographic reconstruction*. arXiv:1612.05515 [cs]. Dec. 2016. URL: http://arxiv.org/abs/1612.05515 (visited on 04/19/2023).

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. arXiv:1607.06450 [cs, stat] version: 1. July 2016. DOI: 10.48550/arXiv.1607.06450. URL: http://arxiv.org/abs/1607.06450 (visited on 05/08/2023).

[5] Mrinal R. Bachute and Javed M. Subhedar. “Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms”. en. In: *Machine Learning with Applications* 6 (Dec. 2021), p. 100164. ISSN: 26668270. DOI: 10.1016/j.mlwa.2021.100164. URL: https://linkinghub.elsevier.com/retrieve/pii/S2666827021000827 (visited on 05/08/2023).

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: (2014). Publisher: arXiv Version Number: 7. DOI: 10.48550/ARXIV.1409.0473. URL: https://arxiv.org/abs/1409.0473 (visited on 06/17/2022).

[7] Mark E. Baker et al. “Contrast-to-Noise Ratio and Low-Contrast Object Resolution on Full- and Low-Dose MDCT: SAFIRE Versus Filtered Back Projection in a Low-Contrast Object Phantom and in the Liver”. en. In: *American Journal of Roentgenology* 199.1 (July 2012), pp. 8–18. ISSN: 0361-803X, 1546-3141. DOI: 10.2214/AJR.11.7421. URL: https://www.ajronline.org/doi/10.2214/AJR.11.7421 (visited on 05/06/2023).

[8] Julia F. Barrett and Nicholas Keat. “Artifacts in CT: Recognition and Avoidance”. en. In: *RadioGraphics* 24.6 (Nov. 2004), pp. 1679–1691. ISSN: 0271-5333, 1527-1323. DOI: 10.1148/rg.246045065. URL: http://pubs.rsna.org/doi/10.1148/rg.246045065 (visited on 04/18/2023).

[9] Richard Bibb, Dominic Eggbeer, and Abby Paterson. “2 - Medical imaging”. en. In: *Medical Modelling (Second Edition)*. Ed. by Richard Bibb, Dominic Eggbeer, and Abby Paterson. Oxford: Woodhead Publishing, Jan. 2015, pp. 7–34. ISBN: 9781782423003. DOI: 10.1016/B978-1-78242-300-3.00002-0. URL: https://www.sciencedirect.com/science/article/pii/B9781782423003000020 (visited on 04/30/2023).

[10] Davis Blalock et al. *What is the State of Neural Network Pruning?* arXiv:2003.03033 [cs, stat]. Mar. 2020. URL: http://arxiv.org/abs/2003.03033 (visited on 05/08/2023).

[11] Thijs Brehm. “Improving ray tracing based iterative image reconstruction in computed tomography by using an artificial neural network in the forward projection”. Bachelor Thesis. Delft University of Technology, Mar. 2023.

[12] Tom B. Brown et al. “Language Models are Few-Shot Learners”. In: (2020). DOI: 10.48550/ARXIV.2005.14165. URL: https://arxiv.org/abs/2005.14165 (visited on 05/08/2023).

[13] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”. In: (2014). DOI: 10.48550/ARXIV.1406.1078. URL: https://arxiv.org/abs/1406.1078 (visited on 05/08/2023).

[14] *DelftBluePhase1*. nl-NL. URL: https://www.tudelft.nl/dhpc/ark/delftbluephase1 (visited on 04/30/2023).

[15] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: (2018). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.1810.04805. URL: https://arxiv.org/abs/1810.04805 (visited on 06/15/2022).

[16] E.V.R. Di Bella et al. “A comparison of rotation-based methods for iterative reconstruction algorithms”. In: *IEEE Transactions on Nuclear Science* 43.6 (Dec. 1996), pp. 3370–3376. ISSN: 1558-1578. DOI: 10.1109/23.552756.

[17] Yimin Ding. “The Impact of Learning Rate Decay and Periodical Learning Rate Restart on Artificial Neural Network”. In: *2021 2nd International Conference on Artificial Intelligence in Electronics Engineering*. AIEE 2021. New York, NY, USA: Association for Computing Machinery, July 2021, pp. 6–14. ISBN: 9781450389273. DOI: 10.1145/3460268.3460270. URL: https://doi.org/10.1145/3460268.3460270 (visited on 05/08/2023).

[18] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: (2020). Publisher: arXiv Version Number: 2. DOI: 10.48550/ARXIV.2010.11929. URL: https://arxiv.org/abs/2010.11929 (visited on 06/15/2022).

[19] Ashkan Eliasy and Justyna Przychodzen. “The role of AI in capital structure to enhance corporate funding strategies”. en. In: *Array* 6 (July 2020), p. 100017. ISSN: 25900056. DOI: 10.1016/j.array.2020.100017. URL: https://linkinghub.elsevier.com/retrieve/pii/S2590005620300023 (visited on 08/29/2022).

[20] Kuang Gong et al. “EMnet: an unrolled deep neural network for PET image reconstruction”. In: *Medical Imaging 2019: Physics of Medical Imaging*. Ed. by Hilde Bosmans, Guang-Hong Chen, and Taly Gilat Schmidt. San Diego, United States: SPIE, Mar. 2019, p. 185. ISBN: 978-1-5106-2543-3 978-1-5106-2544-0. DOI: 10.1117/12.2513096. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10948/2513096/EMnet--an-unrolled-deep-neural-network-for-PET-image/10.1117/12.2513096.full (visited on 08/29/2022).

[21] R. Guedouar and B. Zarrad. “A comparative study between matched and mis-matched projection/back projection pairs used with ASIRT reconstruction method”. en. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 619.1-3 (July 2010), pp. 225–229. ISSN: 01689002. DOI: 10.1016/j.nima.2010.02.077. URL: https://linkinghub.elsevier.com/retrieve/pii/S0168900210002755 (visited on 05/08/2023).

[22] Katharina Hahn et al. “A comparison of linear interpolation models for iterative CT reconstruction: Linear interpolation models in iterative CT reconstruction”. en. In: *Medical Physics* 43.12 (Nov. 2016), pp. 6455–6473. ISSN: 00942405. DOI: 10.1118/1.4966134. URL: http://doi.wiley.com/10.1118/1.4966134 (visited on 05/08/2023).

[23] J. He, Y. Wang, and J. Ma. “Radon Inversion via Deep Learning”. In: *IEEE Transactions on Medical Imaging* 39.6 (2020), pp. 2076–2087. DOI: 10.1109/TMI.2020.2964266. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85085904930&doi=10.1109%2fTMI.2020.2964266&partnerID=40&md5=54d3026d496aa4b3f1a6a351991809b5.

[24] Dan Hendrycks and Kevin Gimpel. “Gaussian Error Linear Units (GELUs)”. In: (2016). DOI: 10.48550/ARXIV.1606.08415. URL: https://arxiv.org/abs/1606.08415 (visited on 05/08/2023).

[25] S Jan et al. “GATE V6: a major enhancement of the GATE simulation platform enabling modelling of CT and radiotherapy”. In: *Physics in Medicine and Biology* 56.4 (Feb. 2011), pp. 881–901. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/56/4/001. URL: https://iopscience.iop.org/article/10.1088/0031-9155/56/4/001 (visited on 05/02/2022).

[26] Kyong Hwan Jin et al. “Deep Convolutional Neural Network for Inverse Problems in Imaging”. In: *IEEE Transactions on Image Processing* 26.9 (Sept. 2017), pp. 4509–4522. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2017.2713099. URL: http://ieeexplore.ieee.org/document/7949028/ (visited on 05/05/2022).

[27] P. M. Joseph. “An Improved Algorithm for Reprojecting Rays through Pixel Images”. eng. In: *IEEE transactions on medical imaging* 1.3 (1982), pp. 192–196. ISSN: 0278-0062. DOI: 10.1109/TMI.1982.4307572.

[28] John Jumper et al. “Highly accurate protein structure prediction with AlphaFold”. en. In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-021-03819-2. URL: https://www.nature.com/articles/s41586-021-03819-2 (visited on 06/15/2022).

[29] Kalpana M. Kanal et al. “Image Noise and Liver Lesion Detection With MDCT: A Phantom Study”. en. In: *American Journal of Roentgenology* 197.2 (Aug. 2011), pp. 437–441. ISSN: 0361-803X, 1546-3141. DOI: 10.2214/AJR.10.5726. URL: https://www.ajronline.org/doi/10.2214/AJR.10.5726 (visited on 05/06/2023).

[30] S. Kappler et al. “First results from a hybrid prototype CT scanner for exploring benefits of quantum-counting in clinical CT”. In: ed. by Norbert J. Pelc, Robert M. Nishikawa, and Bruce R. Whiting. San Diego, California, USA, Feb. 2012, p. 83130X. DOI: 10.1117/12.911295. URL: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.911295 (visited on 04/30/2023).

[31] Lennart R. Koetzier et al. “Deep Learning Image Reconstruction for CT: Technical Principles and Clinical Prospects”. en. In: *Radiology* 306.3 (Mar. 2023), e221257. ISSN: 0033-8419, 1527-1315. DOI: 10.1148/radiol.221257. URL: http://pubs.rsna.org/doi/10.1148/radiol.221257 (visited on 04/23/2023).

[32] David Leibold et al. “Point spread function of photon-counting detectors under pile-up conditions: a proposed framework”. In: *Medical Imaging 2022: Physics of Medical Imaging*. Ed. by Wei Zhao and Lifeng Yu. San Diego, United States: SPIE, Apr. 2022, p. 159. ISBN: 9781510649378 9781510649385. DOI: 10.1117/12.2612861. URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12031/2612861/Point-spread-function-of-photon-counting-detectors-under-pile-up/10.1117/12.2612861.full (visited on 04/30/2023).

[33] Z. Li et al. “A sinogram inpainting method based on generative adversarial network for limited-angle computed tomography”. In: vol. 11072. 2019. DOI: 10.1117/12.2533757. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074250637&doi=10.1117%2f12.2533757&partnerID=40&md5=205c0811c1d9987f5f6c36909b46c947.

[34] Georgios Liaptsis, Alan Clarke, and P. Nithiarasu. “Comparison of different iterative reconstruction algorithms for X-ray volumetric inspection”. en-US. In: *e-Journal of Nondestructive Testing* 23.08 (Aug. 2018). URL: https://www.ndt.net/search/docs.php3?id=22973&msgID=0&rootID=0 (visited on 05/08/2023).

[35] Sebastian Lunz et al. “On Learned Operator Correction in Inverse Problems”. en. In: *SIAM Journal on Imaging Sciences* 14.1 (Jan. 2021), pp. 92–127. ISSN: 1936-4954. DOI: 10.1137/20M1338460. URL: https://epubs.siam.org/doi/10.1137/20M1338460 (visited on 05/08/2023).

[36] Sho Maruyama. *Properties of the SSIM metric in medical image assessment: Correspondence between measurements and the spatial frequency spectrum*. preprint. In Review, Aug. 2022. DOI: 10.21203/rs.3.rs-1872101/v1. URL: https://www.researchsquare.com/article/rs-1872101/v1 (visited on 05/08/2023).

[37] Erik H. W. Meijering. “Spline interpolation in medical imaging: Comparison with other convolution-based approaches”. In: *2000 10th European Signal Processing Conference*. Sept. 2000, pp. 1–8.

[38] Achille Mileto et al. “State of the Art in Abdominal CT: The Limits of Iterative Reconstruction Algorithms”. en. In: *Radiology* 293.3 (Dec. 2019), pp. 491–503. ISSN: 0033-8419, 1527-1315. DOI: 10.1148/radiol.2019191422. URL: http://pubs.rsna.org/doi/10.1148/radiol.2019191422 (visited on 05/06/2023).

[39] Vishal Monga, Yuelong Li, and Yonina C. Eldar. *Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing*. arXiv:1912.10557 [cs, eess]. Aug. 2020. URL: http://arxiv.org/abs/1912.10557 (visited on 08/28/2022).

[40] Vicky Mudeng, Minseok Kim, and Se-woon Choe. “Prospects of Structural Similarity Index for Medical Image Analysis”. en. In: *Applied Sciences* 12.8 (Apr. 2022), p. 3754. ISSN: 2076-3417. DOI: 10.3390/app12083754. URL: https://www.mdpi.com/2076-3417/12/8/3754 (visited on 05/08/2023).

[41] Sagorika Nag et al. “Deep learning tools for advancing drug discovery and development”. en. In: *3 Biotech* 12.5 (May 2022), p. 110. ISSN: 2190-572X, 2190-5738. DOI: 10.1007/s13205-022-03165-8. URL: https://link.springer.com/10.1007/s13205-022-03165-8 (visited on 05/08/2023).

[42] Lalita G Nanjannawar et al. “CBCT in Orthodontics: The Wave of Future”. en. In: *The Journal of Contemporary Dental Practice* 14.1 (Feb. 2013), pp. 153–157. ISSN: 1526-3711. DOI: 10.5005/jp-journals-10024-1291. URL: https://www.thejcdp.com/doi/10.5005/jp-journals-10024-1291 (visited on 05/04/2023).

[43] Thanh Thi Nguyen et al. “Deep Learning Methods for Credit Card Fraud Detection”. In: (2020). DOI: 10.48550/ARXIV.2012.03754. URL: https://arxiv.org/abs/2012.03754 (visited on 05/08/2023).

[44] OpenAI. *GPT-4 Technical Report*. arXiv:2303.08774 [cs]. Mar. 2023. DOI: 10.48550/arXiv.2303.08774. URL: http://arxiv.org/abs/2303.08774 (visited on 05/08/2023).

[45] Willem Jan Palenstijn et al. “A distributed ASTRA toolbox”. eng. In: *Advanced Structural and Chemical Imaging* 2.1 (2017), p. 19. ISSN: 2198-0926. DOI: 10.1186/s40679-016-0032-z.

[46] Akshay Parashar, Payal Anand, and Arun Abraham. “Performance Analysis and Optimization of Serialization Techniques for Deep Neural Networks”. en. In: *Computer Vision, Pattern Recognition, Image Processing, and Graphics*. Ed. by R. Venkatesh Babu, Mahadeva Prasanna, and Vinay P. Namboodiri. Communications in Computer and Information Science. Singapore: Springer, 2020, pp. 250–260. ISBN: 9789811586972. DOI: 10.1007/978-981-15-8697-2_23.

[47] Oscar Pastor-Serrano and Zoltán Perkó. “Learning the Physics of Particle Transport via Transformers”. In: *arXiv:2109.03951 [cs]* (Sept. 2021). arXiv: 2109.03951. URL: http://arxiv.org/abs/2109.03951 (visited on 04/11/2022).

[48] Oscar Pastor-Serrano and Zoltán Perkó. “Millisecond speed deep learning based proton dose calculation with Monte Carlo accuracy”. In: *arXiv:2202.02653 [physics]* (Feb. 2022). arXiv: 2202.02653. URL: http://arxiv.org/abs/2202.02653 (visited on 04/11/2022).

[49] Oscar Pastor□Serrano et al. “Sub□second photon dose prediction via transformer neural networks”. en. In: *Medical Physics* (Feb. 2023), mp.16231. ISSN: 0094-2405, 2473-4209. DOI: 10.1002/mp.16231. URL: https://onlinelibrary.wiley.com/doi/10.1002/mp.16231 (visited on 05/08/2023).

[50] Philippe P. Bruyant. “Analytic and Iterative Reconstruction Algorithms in SPECT”. In: *Journal of Nuclear Medicine* 43.10 (Oct. 2002), p. 1343. URL: http://jnm.snmjournals.org/content/43/10/1343.abstract.

[51] G Poludniowski et al. “*SpekCalc* : a program to calculate photon spectra from tungsten anode x-ray tubes”. In: *Physics in Medicine and Biology* 54.19 (Oct. 2009), N433–N438. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/54/19/N01. URL: https://iopscience.iop.org/article/10.1088/0031-9155/54/19/N01 (visited on 04/30/2023).

[52] Jerry L. Prince and Jonathan M. Links. *Medical imaging signals and systems*. 2nd ed. Boston: Pearson, 2015. ISBN: 978-0-13-214518-3.

[53] Uwe Schneider, Eros Pedroni, and Antony Lomax. “The calibration of CT Hounsfield units for radiotherapy treatment planning”. In: *Physics in Medicine and Biology* 41.1 (Jan. 1996), pp. 111–124. ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/41/1/009. URL: https://iopscience.iop.org/article/10.1088/0031-9155/41/1/009 (visited on 04/30/2023).

[54] Oliver Schoppe et al. “Deep learning-enabled multi-organ segmentation in whole-body mouse scans”. en. In: *Nature Communications* 11.1 (Nov. 2020), p. 5626. ISSN: 2041-1723. DOI: 10.1038/s41467-020-19449-7. URL: https://www.nature.com/articles/s41467-020-19449-7 (visited on 05/08/2023).

[55] Robert L. Siddon. “Fast calculation of the exact radiological path for a three-dimensional CT array: Technical Reports: 3D CT array path calculation”. en. In: *Medical Physics* 12.2 (Mar. 1985), pp. 252–255. ISSN: 00942405. DOI: 10.1118/1.595715. URL: http://doi.wiley.com/10.1118/1.595715 (visited on 04/19/2023).

[56] Kavitha Srinivasan, Mohammad Mohammadi, and Justin Shepherd. “Applications of linac-mounted kilovoltage Cone-beam Computed Tomography in modern radiation therapy: A review”. In: *Polish Journal of Radiology* 79 (July 2014), pp. 181–193. ISSN: 1733-134X. DOI: 10.12659/PJR.890745. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4085117/ (visited on 05/09/2023).

[57] Emanuel Strom et al. “Photon-Counting CT Reconstruction With a Learned Forward Operator”. In: *IEEE Transactions on Computational Imaging* 8 (2022), pp. 536–550. ISSN: 2333-9403, 2334-0118, 2573-0436. DOI: 10.1109/TCI.2022.3183405. URL: https://ieeexplore.ieee.org/document/9797844/ (visited on 05/08/2023).

[58] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: (2014). DOI: 10.48550/ARXIV.1409.3215. URL: https://arxiv.org/abs/1409.3215 (visited on 05/08/2023).

[59] Pankaj Tandon et al. “Interaction of Ionizing Radiation with Matter”. en. In: *Radiation Safety Guide for Nuclear Medicine Professionals*. Singapore: Springer Nature Singapore, 2022, pp. 21–35. ISBN: 978-981-19451-7-5 978-981-19451-8-2. DOI: 10.1007/978-981-19-4518-2_3. URL: https://link.springer.com/10.1007/978-981-19-4518-2_3 (visited on 05/06/2023).

[60] Chao Tang et al. *Generative Adversarial Network-Based Sinogram Super-Resolution for Computed Tomography Imaging*. Number: arXiv:2008.03142 arXiv:2008.03142 [physics]. Aug. 2020. URL: http://arxiv.org/abs/2008.03142 (visited on 05/31/2022).

[61] TensorFlow Developers. *TensorFlow*. Mar. 2023. DOI: 10.5281/ZENODO.4724125. URL: https://zenodo.org/record/4724125 (visited on 04/30/2023).

[62] Ashish Vaswani et al. “Attention Is All You Need”. In: *arXiv:1706.03762 [cs]* (Dec. 2017). arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762 (visited on 04/11/2022).

[63] Yizhong Wang et al. “An effective sinogram inpainting for complementary limited-angle dual-energy computed tomography imaging using generative adversarial networks”. In: *Journal of X-Ray Science and Technology* 29.1 (Feb. 2021), pp. 37–61. ISSN: 08953996, 10959114. DOI: 10.3233/XST-200736. URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/XST-200736 (visited on 06/16/2022).

[64] Wang Wei, Ye Biwen, and Wang Jiexian. “Application of a simultaneous iterations reconstruction technique for a 3-D water vapor tomography system”. en. In: *Geodesy and Geodynamics* 4.1 (Feb. 2013), pp. 41–45. ISSN: 16749847. DOI: 10.3724/SP.J.1246.2013.01041. URL: https://linkinghub.elsevier.com/retrieve/pii/S1674984715300616 (visited on 05/06/2023).

[65] Martin J. Willemink and Peter B. Noël. “The evolution of image reconstruction for CT—from filtered back projection to artificial intelligence”. en. In: *European Radiology* 29.5 (May 2019), pp. 2185–2195. ISSN: 0938-7994, 1432-1084. DOI: 10.1007/s00330-018-5810-7. URL: http://link.springer.com/10.1007/s00330-018-5810-7 (visited on 05/02/2022).

[66] Martin J. Willemink et al. “Computed Tomography Radiation Dose Reduction: Effect of Different Iterative Reconstruction Algorithms on Image Quality”. en. In: *Journal of Computer Assisted Tomography* 38.6 (2014), pp. 815–823. ISSN: 0363-8715. DOI: 10.1097/RCT.0000000000000128. URL: http://journals.lww.com/00004728-201411000-00002 (visited on 05/01/2022).

[67] Martin J. Willemink et al. “Iterative reconstruction techniques for computed tomography Part 1: Technical principles”. en. In: *European Radiology* 23.6 (June 2013), pp. 1623–1631. ISSN: 0938-7994, 1432-1084. DOI: 10.1007/s00330-012-2765-y. URL: http://link.springer.com/10.1007/s00330-012-2765-y (visited on 05/01/2022).

[68] Martin J. Willemink et al. “Iterative reconstruction techniques for computed tomography part 2: initial results in dose reduction and image quality”. en. In: *European Radiology* 23.6 (June 2013), pp. 1632–1642. ISSN: 0938-7994, 1432-1084. DOI: 10.1007/s00330-012-2764-z. URL: http://link.springer.com/10.1007/s00330-012-2764-z (visited on 05/01/2022).

[69] *World health statistics 2020: monitoring health for the SDGs : sustainable development goals*. eng. OCLC: 1269508557. Geneva, Switzerland: World Health Organization, 2020. ISBN: 978-92-4-000510-5.

[70] Yuxin Wu and Kaiming He. *Group Normalization*. en. Mar. 2018. URL: https://arxiv.org/abs/1803.08494v3 (visited on 05/01/2023).

[71] “X-Ray Mass Attenuation Coefficients”. en. In: *NIST* (Sept. 2009). URL: https://www.nist.gov/pml/x-ray-mass-attenuation-coefficients (visited on 04/30/2023).

[72] Wenjin Yu et al. “Deep Learning-Based Classification of Cancer Cell in Leptomeningeal Metastasis on Cytomorphologic Features of Cerebrospinal Fluid”. In: *Frontiers in Oncology* 12 (Feb. 2022), p. 821594. ISSN: 2234-943X. DOI: 10.3389/fonc.2022.821594. URL: https://www.frontiersin.org/articles/10.3389/fonc.2022.821594/full (visited on 05/08/2023).

[73] Gengsheng L. Zeng. “Counter examples for unmatched projector/backprojector in an iterative algorithm”. en. In: *Chinese Journal of Academic Radiology* 1.1 (July 2019), pp. 13–24. ISSN: 2520-8985, 2520-8993. DOI: 10.1007/s42058-019-00006-1. URL: http://link.springer.com/10.1007/s42058-019-00006-1 (visited on 05/06/2023).

[74] Bo Zhu et al. “Image reconstruction by domain-transform manifold learning”. eng. In: *Nature* 555.7697 (Mar. 2018), pp. 487–492. ISSN: 1476-4687. DOI: 10.1038/nature25988.