Delft University of Technology

Master Thesis

MSc Applied Mathematics: Discrete Mathematics and Optimization

EEMCS

Modeling ray angles in deep learning based dose calculation algorithms

Jim van der Valk Bouman (ID: 4379926)

Supervised by:
Dr. Ir. Sebastiaan Breedveld
Dr. Krzysztof Postek
Dr. Zoltán Perkó
Oscar Pastor-Serrano

December 20, 2022

Abstract

A fundamental tool in radiotherapy treatment planning is the dose calculation algorithm, which models the dose that will be distributed for given beam parameters and patient geometry. Various available algorithms include Monte Carlo simulations (MC) and pencil beam algorithms (PBA), with the former being computationally expensive but offering high precision and the latter sacrificing precision for speed. A recent study presents the deep-learning based Dose Transformer Algorithm (DoTA) which provides MC accuracy at speeds 33 times faster than PBA. However, as currently implemented, DoTA dose computations assume that each ray enters the patient geometry perpendicularly, while clinical treatment plans consist of many diverging rays with angles of entry up to 5°.

In this project, we extend the current model to include angular dependency. The resulting models DoTA-A and DoTA-S improve on DoTA by including angle of entry as an additional input on top of the beam energy and patient geometry. DoTA-A includes the actual angle values as input, while for DoTA-S an expected beam shape is precalculated with a trajectory based on the angle of entry. A training dataset of more than 30.000 samples with MC baseline dose is generated from a public patient dataset, using a 2 mm resolution. The architecture of the models is similar to that of DoTA, with convolutional layers extracting important spatial features from the input geometry and a transformer layer using a self-attention mechanism to weigh token inter-dependence.

The models DoTA-A and DoTA-S are evaluated and compared on different test sets with MC baseline doses. Both models are shown to be more accurate than PBA, with DoTA-S having the best performance by most metrics. We demonstrate the relevance of ray angles in dose calculations by comparing DoTA-A and DoTA-S to perpendicular MC predictions, which were considered ground-truth for DoTA. The models DoTA-A and DoTA-S compute dose distributions at an average speed of 10 ms to 15 ms per dose, with the predictions achieving an average relative error of 1% across various test sets. The average relative error of the perpendicular MC predictions lies around 3%, demonstrating the importance of angle of entry as an input variable in dose calculation algorithms. The gamma pass rates (for $\delta = 1\%$,$\Delta = 3$mm) of a full treatment plan with dose distributions predicted by our models are 97.60% for DoTA-A and 95.74% for DoTA-S, indicating that there is no strictly better model between the two.

Preface

This master thesis project investigates the application of deep learning in dose calculation algorithms for proton radiotherapy. This topic was proposed in an open vacancy by assistant professor Krzysztof Postek at TU Delft and assistant professor Sebastiaan Breedveld at the unit Medical Physics, department of Radiation Oncology Erasmus MC Cancer Institute. They have supervised me since I started this project, initially as an internship position at the Erasmus MC.

We soon discovered that a promising deep learning based dose calculation algorithm called DoTA had just been developed by PhD candidate Oscar Pastor-Serrano and dr. Zoltán Perkó at the faculty of Applied Sciences TU Delft, as part of an ongoing PhD project. We discussed how I could best contribute to the research with my thesis project and asked if they could supervise my research as well. They agreed, and we decided that I would try to improve on DoTA by including the angle of entry of the individual rays in its input.

I started the project by familiarizing myself with the theoretical background of radiotherapy and deep learning, which took some time since I had minimal knowledge of either beforehand. With help from my supervisors, I eventually understood the inner workings of DoTA well enough to design, train and evaluate my own models, DoTA-A and DoTA-S. Judging by different evaluation metrics used in this project, these models provide more accurate dose predictions than DoTA in a clinical setting.

My thesis committee consist of:

- Jos Weber, Associate Professor TU Delft (Discrete Mathematics and Optimization department)
- Krzysztof Postek, supervisor and Assistant Professor TU Delft (Discrete Mathematics and Optimization department)
- Alexander Heinlein, Assistant Professor TU Delft (Numerical Analysis department)
- Sebastiaan Breedveld, supervisor and Assistant Professor Erasmus MC (Department of Radiotherapy, Medical Physics unit)

I would like to thank my supervisors for their suggestions and guidance, as well as PhD candidates Michelle Oud and Jesus Rojo Santiago, and Medical Physicist Steven Habraken for their contributions. I would also like to thank my friends and family for their invaluable love and support over the past year.

- Jim van der Valk Bouman (December 6, 2022)

Contents

1 Introduction

Cancer is a leading cause of death worldwide, accounting for nearly one in every six deaths in 2020. The survival rate of cancer patients has been steadily increasing over the last decades, due to earlier detection and improvement of the treatment techniques. One of the most prominent treatment methods is radiotherapy. It is used in around 50% of cases, sometimes in combination with chemotherapy or surgery. In radiotherapy, ionizing radiation is used to damage the malignant tumor cells in the patient; however, this unavoidably damages the healthy tissue surrounding the tumor as well. Too much damage to healthy tissue can compromise the patients quality of life in different ways, even after successful treatment. Constructing a treatment that eradicates the tumor while sparing healthy tissue is a persistent challenge in radiotherapy. It involves many trade-offs which are different from patient to patient, and slight inaccuracies can have lasting consequences on the patients health (Breedveld et al. 2019).

Dose calculation algorithms play an important role in radiotherapy and are a fundamental tool for optimizing treatment plans. Dose calculation algorithms compute the dose delivered under specified conditions, i.e. treatment machine parameters and patient geometry. The dose needs to be computed accurately and fast to facilitate precise treatment planning where the treatment can start as soon as possible. Dose calculation algorithms are necessary for other steps of the clinical workflow as well. Current clinical dose computation algorithms are typically either fast but inaccurate in complex geometries, or highly accurate but computationally expensive. The need for accurate computations often forces clinics to use the latter class of algorithms, but their slow speed can be problematic. For instance, they are too slow for real-time adaptive treatments. This is a technique where small changes to the patient geometry over the course of a treatment, such as internal motion, are taken into account to update the treatment plan in real time (Jagt et al. 2018).

1.1 Goal of this Research Project

The goal of this thesis is to explore the possibilities of using deep learning to accurately predict the output of state-of-the-art dose calculation algorithms. We use the successful deep learning-based dose calculation model DoTA (Pastor-Serrano and Perkó 2022) as a starting point and attempt to improve on it. In particular, we focus on the following difference between the proton beams that are modeled by DoTA and the actual beams used in a clinical setting.

DoTA predicts the dose distributed under specific machine settings and for a given patient geometry. The model takes the beam energy and patient geometry at the beams point of entry as input, and outputs the predicted dose. When predicting full dose distributions for hundreds of proton rays, each ray is considered separately, adding the predicted doses together to produce the full dose distribution. This does not consider the relative position of the rays and the prediction operates under the assumption that they enter the patient geometry perpendicularly.

In reality, the proton rays originate in a point source and pass through a magnetic scanner where the beamlets diverge towards their respective point of entry. Each ray therefore enters the patient volume at a slight angle. The diagram in Figure 1 shows a simplified view of the angles of entry that DoTA does not account for.

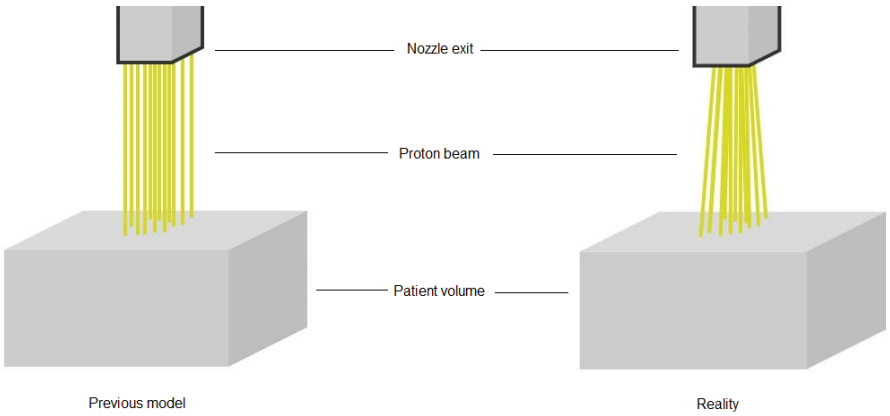

Figure 1: Left: The original DoTA predicts the dose distributed by rays entering the patient geometry perpendicularly. Right: When using these predictions for a full proton beam this does not account for slight angles of entry of the rays, which come from a point source.

These angles of entry lie in the interval $[-5.16°, 5.16°]$, assuming typical conditions of a nozzle around two meters from the patient and diverging beamlets whose points of entry on the top layer of the patient geometry lie in a region of at most 40 cm by 40 cm. Without considering these angles, the distributed dose is slightly but noticeably different. Therefore, including the angle of entry into its computation of doses will bring the algorithm in line with the clinical context. This addition is the main goal of the research project, as well as gaining a better understanding of the role dose calculation algorithms play in radiotherapy and the inner workings of the deep learning architectures that are used. To summarize, the main research question of this project is:

Can we accurately include the entry angle in a deep learning based dose calculation algorithm?

1.2 Thesis outline

In Section 2, the radiotherapy workflow is described in more detail with a focus on dose calculation algorithms, giving examples of currently used and researched models. In Section 3, we give theoretical background information on deep learning and the specific architectures used in this project. In Section 4 we describe the architecture of our model, discuss the different choices that were made in designing them and the training and evaluation procedures. In Section 5 the results of this project are presented, including a description of the generated data, performance of the models under different metrics and comparisons to other models. In Section 6 we discuss the results and their implications.

2 Radiotherapy background

In this section, we introduce the clinical context in which the research project takes place. We first describe the general radiotherapy workflow before going into more detail on the most relevant steps. We describe why high-quality dose calculation algorithms are crucial in this context, and we conclude the section by discussing some of the more important dose calculation algorithms that are used in clinical practice, along with their advantages and disadvantages.

2.1 Radiotherapy workflow

Radiotherapy treatment starts at the first consultation between the patient and the radiation oncologist, where the details of the clinical situation are discussed along with the risks and benefits of treatment. The physician considers information such as the location and stage of the tumor, its mutational status, general state of the patient and so on to decide on a treatment strategy. This could include surgery, chemotherapy, radiotherapy and a number of other modalities, and even multimodal therapy in which multiple treatment strategies can synergize for better clinical outcomes. If the physician and patient have decided to proceed with radiation therapy after patient assessment, the following steps are taken to construct a treatment plan (Feng et al. 2018).

Figure 2: Radiotherapy workflow.

- First, the physician will schedule a simulation with specific instructions. In most cases, this means acquiring high quality CT images of the patient. The physicians instructions include details about scan range, treatment site and patient preparation. The scan is exported to a planning system where the physician can continue with the treatment planning process. The physician then outlines the target volume (i.e. the tumor) and organs at risk (OARs) on the CT images for future reference.
- The treatment planning stage begins with the setting of dosimetric goals for the target and other tissues. The dosimetric goals include a minimum radiation dose to the target volume, and maximum tolerance levels for the different OARs. Then, an appropriate treatment technique is chosen. There are different methods for irradiating the patient and for the most common, external beam radiotherapy (EBRT), there are different particles that can be used with their own advantages and disadvantages. We discuss the different techniques and trade-offs in Section 2.2 below.

  Assuming EBRT was chosen, the beam angles and beamlet intensities are opti-

mized to best accommodate the planning goals. This is the most complex and computationally expensive step, mainly due to the large search space consisting of all possible beam angles and machine settings. This is also a step where dose calculation algorithms play a large role, and we will discuss it in more detail below. Finally, the dose distribution resulting from the selected beam settings is reviewed manually, and the plan may be updated by tightening or relaxing the dosimetric goals in an interactive loop before the final treatment plan is presented to the physician.

- If the treatment plan is accepted by the physician, the plan is subjected to quality assurance protocols to prevent errors and to give high confidence that the patient will receive the prescribed treatment correctly. These protocols can include additional dose calculation to ensure the correctness of the dose computed by the clinical treatment planning system.

Finally, the patient is treated according to the treatment plan. The patient is monitored closely during and after treatment, with follow-up appointments lasting around 3 to 6 weeks after treatment has finished.

2.2 Radiotherapy types

Radiotherapy can be delivered in different ways, commonly divided in three classes based on the position of the radiation source. The most common is external beam radiotherapy (EBRT). Alternatively, a sealed radiation source can be placed in or next to the volume requiring treatment (brachytherapy) or radioactive substances can be introduced into the body via injection or ingestion (radionuclide therapy). A physician can decide on a combination of multiple methods as well (multimodal therapy).

EBRT uses an external source of radiation pointed at the desired part of the patient's body. The most widely used sources are X-rays and electron beams, but heavier particles such as protons can also be used. The different properties of these particles and their dose distributions give each choice advantages and disadvantages in radiotherapy. Figure 3 compares the dose deposition behavior for photon beams (X-ray), electron beams and proton beams.

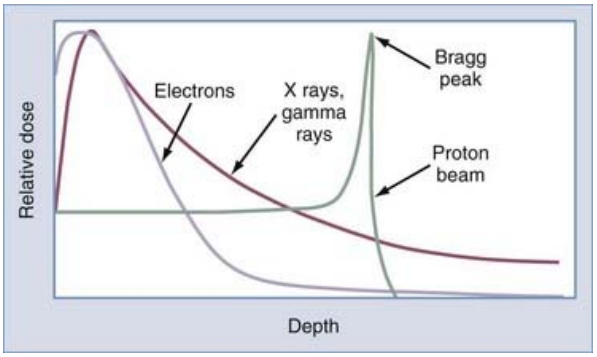

Figure 3: Comparisons of dose deposition for photon beams. Proton beams can deliver well-localized high radiation with essentially no exit dose, by taking advantage of the Bragg peak effect and cross-firing multiple beams. (Sheehan 2015)

We can see that the dose depositions of X-rays and electron beams are comparable, but that electron beams exhibit rapid dose falloff and deposit more dose near the surface

of the tissue. This makes electron therapy well suited for target volumes extending to or near the patient's skin. In some cases, electron beams are combined with surgery to apply the radiation directly to the tumor, which is called intraoperative electron radiotherapy.

X-rays exhibit a slight build-up of dose upon entering the tissue, which has the advantage of sparing the patient's skin from the highest radiation. They deposit a significant amount of energy at depth and can therefore be used to treat tumors deep within the body. They are used very commonly in radiotherapy for a wide range of cancers, with different energies used depending on the desired outcome.

Proton beams display the Bragg peak effect seen in Figure 3, where the vast majority of radiation is delivered in a narrow range of depth. This has the prime advantage in radiotherapy of minimizing damage to healthy tissue, with low entry and exit dose (dose in front of and behind the target volume). The slow falloff of X-ray dose deposition in particular causes relatively high exit doses, unavoidably damaging healthy tissue and potentially causing secondary cancers. Proton radiotherapy may be selected in cases where it is important to minimize radiation to OARs while keeping dose to the target volume high: compared to the most advanced photon treatment techniques, proton therapy can deliver similar or higher radiation doses to target volumes with a 50%–60% reduction of total body radiation dose (Kandula et al. 2013). Figure 4 shows the dose distributions of two treatment plans for the same patient, with one plan being optimized for photon treatment and the other for proton treatment.

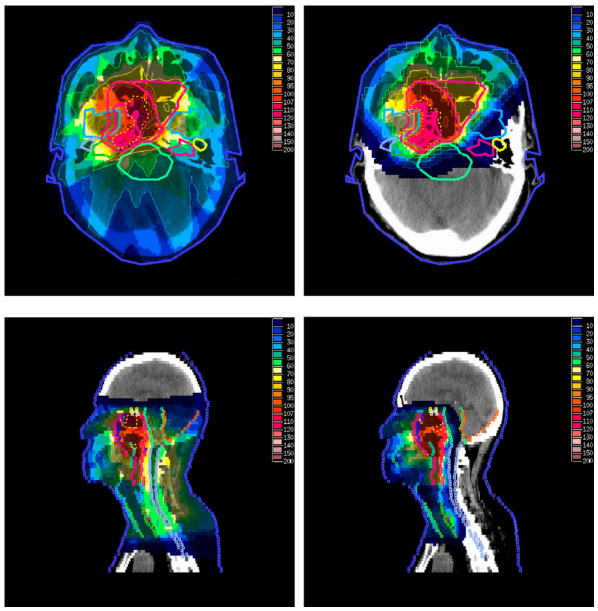

Figure 4: Total predicted dose distributions of plans optimized for photon (left) and proton (right) radiotherapy, with OARs and tumor delineated. With similar radiation to the target volume, low damage to healthy tissue is the main advantage of proton over photon treatments (Taheri-Kadkhoda et al. 2008).

2.3 Treatment planning

Treatment planning refers to the stage in the radiotherapy workflow where the beam specifications for a treatment are determined. A treatment plan describes the machine parameters that are to be used, including the number of beams, their intensities and the spatial configuration of the patient and machine. The treatment plan is the output of an optimization program which optimizes with the given dosimetric goals. The problem is divided into two parts: the selection of the *beam angles* and the *beam intensity profiles*. Beam angle selection refers to choosing the amount of beams that are used and their spatial configuration relative to the patient. A common procedure is to discretize the continuous space of possible beam angles to reduce the search space. This leaves 3-9 possible beam angles in typical settings where only the gantry arm changes position. For each angle, the beam intensity profile is optimized, and the beam resulting in the best treatment plan is fixed. Now additional beams are added to the treatment plan until a maximum number of beams is reached or no substantial improvement to the treatment plan is made.

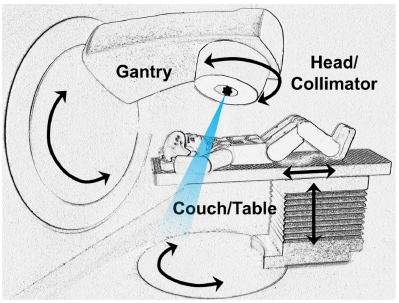

Figure 5: Illustration of a common radiotherapy machine. In this project we only consider beam angles imposed by rotation of the gantry. The ionizing beam exits the head through the collimator, which is able to shape and modulate the beam (Breedveld et al. 2019)

The beam intensity profile optimization is more numerically challenging, and can be described as follows. The *beam* is discretized into n small *rays* (or *beamlets*) based on direction and energy level. These rays form the set of decision variables $x \in \mathbb{R}^n$ for the numerical optimization problem. The value of each element x_j represents how long the ray is "on" for $i = j, \ldots, n$. The patient is also discretized into N voxels, often using a lower resolution than the original scan to keep the problem numerically manageable. The dose across all voxels is measured through the voxel dose vector $d \in \mathbb{R}^N$. The relation between the beamlets x and the distributed dose d is linear, meaning we have:

$$d = d(x) = Ax \tag{1}$$

where A is called the *dose influence matrix*. The established dosimetric goals are now described in terms of d. The structures of interest (i.e. target volume and OARs) are listed, and we define d_i as the voxel dose vectors for the voxels that structure i delineates on the CT. A typical mathematical formulation of the problem then looks as follows:

$$\begin{aligned} \min_{x} \quad & f(d_1) \\ \text{s.t.} \quad & g_j(d_i) \leq b_j \\ & g'_j(d_i) \geq b'_j, \\ & h(x) \leq c \\ & x \geq 0 \\ & d_i = A_i x \end{aligned} \tag{2}$$

In this formulation, i always runs over the listed structures and j runs over some index set. An example of the cost-function f is $f(d_1) := \sum d_1$ where the total dose to structure 1 is minimized as far as possible while respecting the constraints. The dosimetric constraints are described by cost-functions g_j and g'_j, where the limits b_j and b'_j describe maximum and minimum dose to the respective structures. The target volume often has both a minimum and a maximum dose constraint on it, while OARs usually have a maximum dose constraint to prevent complications. The requirement $h(x) \leq c$ models hardware limitations, for example putting an upper bound on the treatment delivery time.

Solving this multi-criteria optimization problem is computationally expensive. The cost functions can be non-convex and the optimization problem is commonly solved multiple times for each patient before an optimal treatment plan is selected. An efficient method of solving these problems is therefore very important, and a lot of research is being done to improve the methods currently in use (Kim et al. 2020). In this project, however, we focus on the computation of the dose influence matrix A, which is done using dose calculation algorithms.

2.4 Dose calculation algorithms

A *dose calculation algorithm* is an algorithm that computes the expected dose distribution for given beam settings and patient geometry. In the above formulation, the desired output of the dose calculation algorithm is the dose influence matrix A with the patient CT and beam settings as input. The distributed dose must be computed for each of the thousands available rays, so computation speed is an important factor for the entire treatment planning process. Dose calculation algorithms are important for other steps of the radiotherapy workflow as well: for example, for quality assurance of the final treatment plan, a high quality dose prediction of the full plan is necessary to visually inspect and verify plan robustness.

When comparing the dose distribution provided by a dose calculation algorithm to the actual dose that will be delivered under the given machine settings, high accuracy of the algorithm is crucial in radiation therapy. The International Commission on Radiation Units and Measurements (ICRU) has recommended an overall relative dose accuracy within 5%; considering the uncertainties resulting from the patient setup, machine calibration and treatment planning system, it is necessary to have a dose calculation algorithm that can predict dose distribution with a 3% relative error margin (Shalek 1977). Accurate computation of dose distribution requires the accurate modeling of particle transport, which is a complicated task in an inhomogeneous medium such as the human body, especially for tumors located in the lung. We describe two of the most commonly used dose calculation methods: Monte Carlo simulation and Pencil Beam methods.

Monte Carlo (MC) methods are a general class of computational algorithms that rely on random sampling to solve problems which are deterministic in principle. In the context of dose calculation, MC-based algorithms sample a large number of particles (photon, proton, ...) and simulate the transport of the individual particles. This requires knowledge of interactions of the particle (energy transfer, production of secondary particles, ...) and the probability of each interaction, based on the composition of the tissue as described by the patient CT. An example individual particle transport prediction could be based on sampling the distance before the first particle interaction from a known probability distribution, sampling a type of interaction from the probabilities of each interaction taking place, then repeating the procedure until a user defined energy cut-off has been reached. Through the large number of simulations ($n > 10^7$), the deposited energy in each voxel can be calculated.

MC-based algorithms have been proven to yield the best dose accuracy compared to other algorithms and are the current clinical standard. The main drawback of MC implementations has historically been the long computation times. The algorithms have been sped up over the years, largely due to newer hardware and the parallel computing power provided by GPUs (Wan Chan Tseung, J. Ma, and Beltran 2015). However, developments in treatment planning techniques such as real-time adaptive treatments, where new plans must be generated in real time to adapt the treatment to anatomical changes, require high-speed and accurate dose calculations during optimization (Kontaxis et al. 2017).

Pencil Beam Algorithms (PBA) are analytical in nature and are based on the assumption that particle rays behave approximately like many, infinitely narrow pencil beams. Each of these pencil beams has a central axis ray along which it deposits some dose. The deposited dose around this axis is derived from the basic scattering and absorption processes that the particles undergo, and is sometimes calculated using MC simulations (Carolan 2010). The deposited dose by a single pencil beam also depends on the beam's intensity. Patient inhomogeneities are accounted for by modifying the shape of the pencil beam dose distribution based on the density of the tissue that the pencil beam travels through. To calculate the dose deposited by the entire particle beamlet, the dose distributions of all pencil beams are summed up. PBA methods are significantly faster than MC-based methods, but suffer from relatively high inaccuracy (Teoh et al. 2019). This is especially the case around inhomogeneous geometries, such as the lungs (Taylor, Kry, and Followill 2017). Sorriaux et al. (2017) found gamma pass rates for lung patient full plan PBA predictions as low as 44.7% (see Section 4.4).

2.5 Related research

Most current research on the reduction of dose calculation times focuses on either improving current dose calculation algorithms or the implementation of deep learning (Pastor-Serrano and Perkó 2022). Some deep learning based algorithms approximate full dose distributions for treatment plans based on historical data of other optimal plans (Ronneberger, Fischer, and Brox 2015). Convolutional neural networks trained to approximate full treatment plan dose distributions often use additional information such as organ and tumor delineations as input (Chen et al. 2019) (Nguyen et al. 2019). Other studies focus on specific steps of the radiotherapy workflow, improving dose calculation times despite not providing generally applicable dose calculation algorithms (Meyer et al. 2018).

Besides DoTA, we are aware of only three papers where deep learning was imple-

mented in proton dose calculation algorithms. However, although these models offer significantly faster dose predictions than current clinical algorithms, they are not independent or generally applicable. Two of the papers use low-accuracy dose predictions of other models as their input and convert them to achieve higher accuracy, with the initial predictions coming from either PBA (Wu, Nguyen, et al. 2021) or relatively fast, low-accuracy MC predictions (Javaid et al. 2021). The third uses treatment plan and site information to pre-calculate the necessary input variables, and can therefore only be used as a dose calculation algorithm for a specific treatment site (Nomura et al. 2020).

To summarize, radiotherapy and especially the treatment planning stage benefit greatly from fast and accurate dose calculation algorithms. Current models offer a tradeoff between slow but accurate MC-based algorithms and the fast but inaccurate PBA. In this project, we aim to offer high-speed MC-precision dose calculations through a deep learning model which leverages specific architectures well-suited for this task. In Section 3, we give the relevant theoretical context from the field of deep learning and introduce the most important components of our dose calculation algorithm.

3 Deep Learning Background

This section provides theoretical background information for the models constructed in this project. We start by introducing the fields of machine learning and deep learning, as well as the most simple and often-used model architectures. We then go into more detail on two architectures, convolutional neural networks and transformers, which form the building blocks of our models. More details on the topics discussed in this chapter can be found in the extensive Deep Learning textbook by Goodfellow, Bengio and Courville (2016).

3.1 Deep Learning

The field of machine learning has made substantial progress over the last decades, made possible in large part by the continuous increase in computing power available. *Machine learning (ML)* refers to techniques used in artificial intelligence where algorithms use data to infer conclusions or predictions related to that data, without explicit instructions coming from the programming. In other words, ML models are built to extract patterns from the data fed to them, without human instructions on how to interpret the data. The automated nature of ML has proven extremely effective in a variety of applications, such as computer vision, speech recognition and many medical settings, as well as any field in which pattern recognition plays a large role. ML methods are often split into two categories. In *unsupervised learning*, the model tries to extract patterns from the input data, often trying to cluster data points based on similarity or finding probability distributions that describe that data as well as possible. In *supervised learning*, the input data is labeled, and the model tries to find the relation between the presented data and the corresponding labels, often as to accurately predict the label of new unlabeled data.

Simple ML algorithms are able to extract useful information from well-structured datasets, but when the features that must be extracted are more complex (e.g. object recognition from a large grid of colored pixels) and require sophisticated understanding of the data structure, the algorithms used are often based on deep learning. Deep learning refers to ML model architectures with multiple layers, where instead of considering raw input data, information in each layer is extracted from the output of the previous layer. In the example of object recognition, the first layer of a deep learning model might extract edges from the input pixels, the second layer could extract corners and contours from those edges and so on. The processed data can then more easily be interpreted by the final layer. The layers in between the input and output layers are called *hidden layers*.

3.2 Neural Networks

The quintessential deep learning models are *(artificial) neural networks*. Their name is derived from biological neural networks found in animal brains, on which the model architectures are loosely based. Typically, a neural network tries to approximate some function $f^*(x)$ which assigns a label y to each input x (supervised learning). The network defines a mapping $y = f_\theta(x)$ and learns the value of the parameters θ that result in the best function approximation. The word network here refers to the layer structure described above: for example, for a neural network with three layers we have $f_\theta(x) = f_\theta^{(3)}(f_\theta^{(2)}(f_\theta^{(1)}(x)))$ where the functions $f_\theta^{(i)}$ describe each layer of the network. These layers are typically vector valued, which is what the word neural refers to. We can think of a layer as many vector-to-scalar functions called artificial neurons or

nodes, which process information transmitted from nodes in the previous layer. The simplest type of neural network architectures are *feed-forward* neural networks, which simply means that information only moves through the network layers in the forward direction, without any loops or cycles. A schematic overview of a feed-forward neural network architecture can be seen in Figure 6.

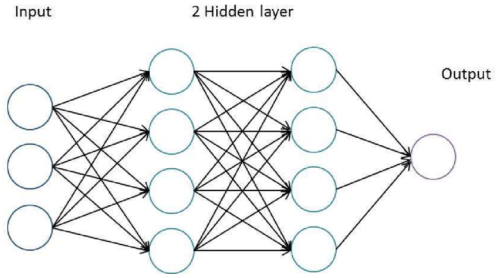

Figure 6: An example a simple feed forward neural network. The arrows represent the node-to-node functions: these depend on the weights θ that are learned by the model during training (X. Ma et al. 2019).

Training neural networks means finding the weights θ that give the best approximation f_θ of f^*. Performance is measured through a *loss function* that measures the difference between the predicted values $f_\theta(x)$ and the true values y for a given, labeled dataset. A common choice is the mean square error (MSE) given by $L_{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2$, where i iterates over all data points (x_i, y_i). The weights θ are then iteratively optimized through an *optimization algorithm*. Here, most methods make use of the gradient of the loss function, updating the weights in the opposite direction (steepest descent) with some predetermined step size. The weights are updated iteratively over all data points, usually considering a small *batch* of data points before updating the weights. The number of complete passes through the training data before training terminates is called the number of *epochs*.

3.3 Residual connections

Residual connections or skip-connections are a simple addition to the architecture of neural networks, designed to speed up the convergence of training. In traditional feed-forward neural networks, data flows from one layer to the next sequentially. Residual connections provide a path for data to deeper layers of a neural network, skipping the layers in between. The core idea is that layers that usually cause slow training convergence, for example due to vanishing gradients, can be skipped in these more shallow "sub-networks", which will converge faster without losing overall model accuracy. Although not extensively studied, neural networks with residual connections showed their potential when one such model won the ImageNet 2015 competition and became the most cited neural network of the 21st century (He et al. 2016).

3.4 Convolutional neural network

Convolutional neural networks are a type of artificial network designed to process grid-like data. They improve traditional neural networks by leveraging a few important ideas. Consider a hidden layer in a neural network, with m inputs coming from the nodes in the previous layer and n output nodes. In a traditional neural network, the interaction between the input and output nodes is described by a matrix multiplication by a parameter-valued matrix of size $m \times n$.

In a convolutional neural network, the amount of connections from the input nodes to each output is instead limited to a fixed number k. The matrix with the corresponding k parameters is called the *kernel*. The convolution operation which convolutional layers apply to their input is visualized in Figure 7. A similar operation that is sometimes used is the *transposed convolution* operation which increases the size of the input data instead of decreasing it. Instead of multiplying the kernel with k input elements to generate one output element, for transposed convolutions the kernel is multiplied element-wise with one input element to generate k output elements.

Another difference between convolutional and traditional neural networks is that in the convolutional case, the same kernel is used for each of the n output nodes, meaning that the parameters of each interaction are shared. A significant advantage of convolutional neural networks is the reduction in runtime and memory requirements: the amount of parameters that need to be stored are now k instead of $m \cdot n$, where k is usually several orders of magnitude smaller than m. Another advantage comes from the parameter sharing: instead of trying to learn each relation between the input and output nodes, the layer detects one relation across all the inputs. One common example is edge detection in images, which convolutional neural networks can do with very small kernels and many times more efficiently than traditional models.

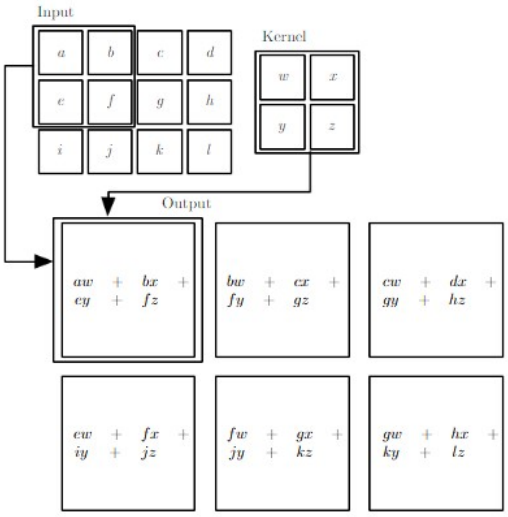

Figure 7: An example of 2-D convolution. The kernel uses the same weights for each output node, greatly reducing the memory requirements (Goodfellow, Bengio, and Courville 2016).

A convolutional network typically includes multiple blocks consisting the same sequential layers. We describe a common layer structure here. In the first layer, the convolution operation described above is performed on the input data in parallel to produce the processed data. A normalization layer which normalizes the processed data by the mean and variance may be included and has been shown to greatly improve model performance (Ioffe and Szegedy 2015) (Wu and He 2018). The data is then passed through a pooling layer. Pooling layers reduce the size of the data by combining the outputs of small groups of nodes into a single node. This makes the representation of the data less invariant to small translations of the input: two data points that are only marginally different will give the same output after pooling and can therefore be treated similarly. The most common types of pooling are max pooling and average pooling, where the value of the single node is determined by the

maximum and average value of the group of nodes that is considered, respectively. Finally, an activation layer is often included, which applies a non-linear function to the layer output before passing it to the next layer. The reason is that the (convolutional) layer has so far only applied linear functions to the input data, and we want the model to be able to approximate non-linear functions as well. The default recommendation in modern neural networks is the rectified linear unit (ReLU) activation function (Jarrett et al. 2009) (Glorot, Bordes, and Bengio 2011), defined by the element-wise operation $g(z) := \max\{0, z\}$.

3.5 Transformer

A *Transformer* is a deep learning model introduced in recent years, with great success in natural language processing. Transformer models leverage the self-attention mechanism (Vaswani et al. 2017). The idea is that, similar to cognitive attention, some parts of the data are given extra focus when they appear to be more important. This is achieved through the inclusion of attention units, with dedicated parameters called attention weights which keep track of which parts of the data seem most important.

We describe a common construction of a self-attention (SA) layer. The L input elements get embedded into an input matrix $X \in \mathbb{R}^{L\times d}$ for some dimension d. This embedding operation has learned weights, which the model will optimize during training. X is then projected into three matrices of the same shape (say $d \times d_h$): the query matrix $X^Q := W^Q X$, the key matrix $X^K := W^K X$ and the value matrix $X^V := W^V X$, again using learned weights for the projections. The output of the attention layer is then given by:

$$SA(X) = \text{softmax}\left(\frac{X^Q (X^K)^T}{\sqrt{d_h}}\right) \cdot X^V \in \mathbb{R}^{L\times d} \tag{3}$$

The softmax function used here is a common function which normalizes its input into a probability distribution:

$$\begin{aligned} &\text{softmax} : \mathbb{R}^K \to (0,1)^K \\ &\text{softmax}(z)_i := \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \end{aligned}$$

The idea of this formula is that values in X^V are amplified when the corresponding element of the key X^K are more similar to elements in X^Q. In this way, elements of the output $SA(X)$ are weighted based on how similar they are to other elements. Conceptually, the attention layer is able to detect patterns in which input tokens influence each other strongly.

In *Multi-head Self Attention* (MSA) multiple sets of attention weights are used, which are called *attention heads*. N_h SA operations are computed in parallel, and each of the N_h attention heads can detect different self-attention patterns. The outputs of the different operations are concatenated and linearly projected with learned weights like for the single-head case. A transformer model typically combines these self-attention mechanisms with feed-forward neural networks, in an encoder-decoder structure. Here the data is first processed through the encoder layers, with the attention units capturing information on which parts of the data are relevant to each other. The decoder layers then take the encoded output and contextual information and generate an output sequence.

MSA is invariant to the order in which the input elements are presented, but by applying a fixed positional embedding before the MSA operations it is possible to encode positional information into the self-attention mechanism. One particular case is *causal self-attention*, where a masking function removes all connections from a token to tokens that come after it in the input sequence. This is often used in temporal settings, where information from the past and present is considered for MSA but not information from the future. We will discuss how this makes causal self-attention particularly suited to our setting in Section 4.

The model architectures desribed above were recently used in a dose-calculation algorithm by Oscar Pastor-Serrano and Zoltán Perkó, referred to as the Dose Transformer Algorithm (DoTA) (Pastor-Serrano and Perkó 2022). The model was trained using MC generated dose distributions and outperforms both PBA, being 33 times faster and more accurate in inhomogeneous geometries, and MC itself by offering the same accuracy at a speed 4000 times faster. DoTA serves as a starting point for the models in this project and Section 4 describes our methods, model architectures and how they expand on the original DoTA. Our models aim to improve DoTAs clinical capabilities by including additional information for each beam, bringing the predicted dose distributions closer to clinical reality.

4 Methodology

In this section, we describe the methods used for the construction and evaluation of the models DoTA-A and DoTA-S. We first describe how these models are constructed from the original DoTA, and the different ways in which they interpret the angle of entry as an additional input. We then give an overview of the data generation procedure and the choices we made with regards to the training data. The models architectures are also described, going over the building blocks that were used and the training procedure. Finally we go into detail on the different test sets and evaluation methods we used to test the performance of our models.

4.1 Variables

The dose calculation algorithms that we consider in this project compute the output dose distribution corresponding to given machine settings and patient geometry. Mathematically, we can consider the following variables:

- Input geometry $x \in \mathbb{R}^{L \times H \times W}$. This is a subset of the full patient CT scan, taking values on the Hounsfield scale. The Hounsfield scale describes radiodensity, which is the ability of certain kinds of radiation to pass through a particular material. The radiodensity of air is defined as -1000 Hounsfield Units (HU) and water is defined at 0 HU, with different body tissues such as fat and bone typically taking HU values in the range $[-1000, 1000]$. The volume x with HU values attached to each voxel is taken as the geometric input for our dose calculation algorithms. Other algorithms sometimes require translating the HU values to other descriptors such as material density or stopping power. Since the exact HU values can differ between CT acquisitions based on parameters used by the CT scanner, these conversions are also machine-specific.
- Input particle energy $\epsilon \in E \subset \mathbb{R}^+$. This measures the initial positive charge of the protons in a proton ray in megaelectrovolts (MeV). The protons gradually transfer energy to material that they traverse through, which is what causes the radiation damage to the patient tissue. Therefore, proton rays with higher MeV values typically cause radiation damage deeper inside the patient geometry than rays with low MeV values. For our models, we take the energy range $E = [70, 140]$. This range of energy values is typical for treatments in inhomogeneous geometries such as the lungs, or for relatively shallow tumors in the prostate and head and neck areas.
- Output dose distribution $y \in \mathbb{R}^{L \times H \times W}$. This measures the dose distributed by a proton ray, with each voxel in y measuring the dose absorbed in the corresponding voxel of the geometry x. The dose is typically measured in the SI unit Gray (Gy), although equivalent units are also sometimes used. In radiotherapy, the dose delivered to a patient is usually between 20 and 100 Gray, depending on the site and stage of the cancer being treated.

The depth L, height H and width W of the geometry x and dose volume y are fixed for consistency. In this project, when referencing the dimensions of a volume we refer to the depth-axis as Y (the direction the patient is facing), the height-axis as Z (the axis on which the patients body is positioned) and the width-axis as X. For our models, we let $L = 150$ and $H = W = 25$. From the full patient CT and corresponding dose distribution, we extract the cropped volumes x and y in such a way that the target point of the proton ray lies exactly at the center voxel in the first

layer of x. This usually means that highest dose value of the first layer lies in the center, with dose values decreasing towards the edges in a way that resembles a 2D Gaussian distribution.

The models presented in this project are built on top of the DoTA model. This model captures the relation between input and output through a nonlinear mapping $f_\theta(x, \epsilon) : \mathbb{R}^{L\times H\times W} \times E \to \mathbb{R}^{L\times H\times W}$, performed by a series of artificial neural networks. Our models, however, include additional information in the input data to describe the direction of the ray we want to compute the dose distribution of. We consider two descriptions of this information:

- The **angles** $\phi = (\phi_X, \phi_Z) \in A \subset \mathbb{R}^2$, where $\phi = (0, 0)$ corresponds to a perpendicular ray. Here we limit ourselves to clinically achievable angles, which under typical conditions fall in the range $[-5.16°, 5.16°]$.
- The **shape volume** $s \in \mathbb{R}^{L\times H\times W}$, which contains the shape the dose distribution is roughly expected to take based on the angles and machine settings.

We refer to the corresponding models as *DoTA-A* and *DoTA-S* respectively. The following sections explain the two different descriptions and how they are constructed.

Angle description The DoTA-A model defines a mapping

$$f_\theta^A(x, \phi, \epsilon) : \mathbb{R}^{L\times H\times W} \times A \times E \to \mathbb{R}^{L\times H\times W} \tag{4}$$

In this description we include the angle ϕ under which a ray travels from the nozzle to the patient. When a ray exits the gantry head non-perpendicularly in radiotherapy, the machine reads a specified *target point* on the top layer of the patient geometry to determine the direction. The diverging beamlets share a focal point at a fixed distance above the isocenter, which is machine specific but can be read from the Beam Data Library (BDL) text file (Souris, Lee, and Sterpin 2016). Therefore, using the coordinates of the center-most voxel (isocenter) on the top layer of the patient geometry, the distance from the focal point to this isocenter and the coordinates of the target point, we can calculate the angle under which the ray travels.

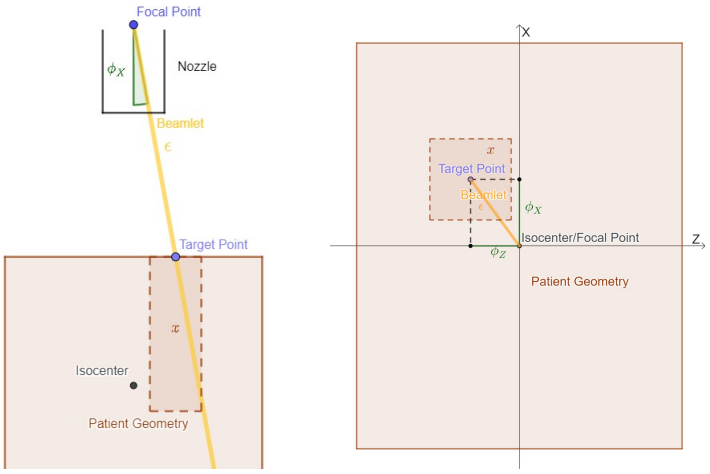

Figure 8: Diagram illustrating the construction of recorded angles ϕ_X and ϕ_Z. Left: X/Y view (side). Right: X/Z view (top-down). Note: in reality the focal point lies two meters (!) above the patient geometry, and the recorded angles are often invisibly slight.

Since the angles in the X and Z direction are imposed by two distinct magnetic fields, the isocenter-to-focal-point distance is different for both angles as well. We treat each case separately, considering the X/Y plane to determine ϕ_X and onto the Z/Y plane to determine ϕ_Z. Note that the actual three-dimensional angle of entry is *not* recorded; the angles ϕ_X and ϕ_Z only describe the angle of entry in their corresponding two-dimensional plane. These descriptions are equivalent however, as can be seen in Figure 8. In the X/Z plane, the angles ϕ_X and ϕ_Z imply the coordinates of the target point, and a conversion from Cartesian to polar coordinates would give a description of the three-dimensional angle of entry plus the beamlet direction in the X/Z plane. The description $\phi = (\phi_X, \phi_Z)$ seems less geometrically intuitive, but is easier to work with mathematically since we frequently compute the angles from the target points coordinates and vice versa.

Shape description The DoTA-S model defines a mapping

$$f_\theta^S(x, s, \epsilon) : \left(\mathbb{R}^{L\times H\times W}\right)^2 \times E \to \mathbb{R}^{L\times H\times W}$$

In this description we compute the shape $s \in \mathbb{R}^{L\times H\times W}$ that the dose distribution of a proton beamlet is expected to take, based on the used machine settings, energy and angle of entry. Figure 9 illustrates the construction of s.

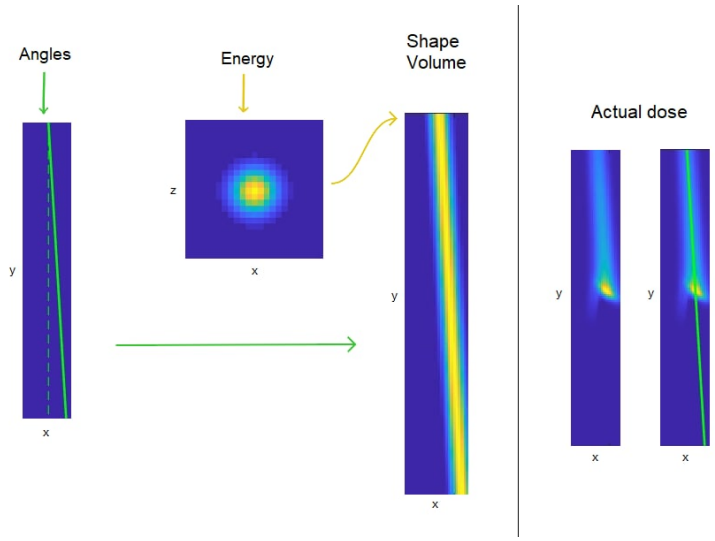

Figure 9: Left: The angle and energy values allow us to predict the ray direction and dose distribution on the first layer, respectively. For each subsequent layer we shift this 2D distribution along the expected direction, which gives the shape volume s. Right: The expected direction aligns with the actual dose distribution.

The dose distribution of such a beamlet upon first entering the patient geometry resembles a 2-dimensional normal distribution, with variance (in X and Z directions) depending on the energy of the beam. The variances per energy can be read from the Beam Data Library (BDL) file of the machine used.

Given the angles of entry, we can also predict the direction of the ray in the patient geometry. On the first layer, the center of the dose is exactly the entry point of the beamlet. For subsequent layers, we move along the line from the virtual source point through the target point to predict the center of the dose distribution at this depth. We use the same variances for the normal distribution in each layer, shifting

the mean along this line. This results in a volume s predicting the shape of the dose distribution.

Note that the shape volumes for fixed machine settings only depend on the energy and angles of a ray. The patient geometry is not considered and the actual dose distribution in complex geometries is very different from our predicted shape. We also need more memory to store the input data in this way, since instead of $\phi \in \mathbb{R}^2$ we now store $s \in \mathbb{R}^{L \times H \times W}$.

The advantage of DoTA-S is that it simplifies the relation between input and output. In the mapping (4) defined by DoTA-A, the relation between ϕ and the output dose distribution y is highly non-trivial. A small change in one of the angles ϕ_X and ϕ_Z can cause a large change in the shape of the dose distribution which can hurt the models accuracy. In DoTA-S we circumvent this issue by constructing the input s to be similar to the desired output y.

4.2 Data generation

The first step in the construction of our models was the generation of a training dataset. A training dataset consists of a large amount of training samples, which in our case are sets $\{x, \phi, \epsilon, y\}$ for DoTA-A and $\{x, s, \epsilon, y\}$ for DoTA-S. We generated a large amount of input variables x, ϕ and ϵ in a randomized way, which are described below. After a collection of input values was established, we calculated the corresponding baseline dose distribution y using the open source Monte Carlo particle simulation code MCsquare (Souris, Lee, and Sterpin 2016), which is optimized for modern multi-core CPUs. MCsquare calculations are done in the open source treatment planning software matRad (Cisternas et al. 2015), utilizing 8 Intel Xeon E5-26990 CPUs in parallel.

The angle of entry is modeled in MCsquare through the specification of a target point for each ray. The software assumes that the gantry nozzle lies straight above the isocenter of the CT, and by default it will simulate particle trajectories where the ray travels straight down. To impose the angle of entry for our training data we instead specify a target point for each ray.

The data was generated from a CT scan dataset of 53 patients, of which 21 lung, 20 head and neck, and 12 prostate scans. For each patient, we first generated 21 different patient geometries by rotating the scan around the Z axis. These rotations simulate the physical rotation of the gantry arm, which is able to rotate around the Z axis of the patient to irradiate under different beam angles. For each of these 21 geometries, we then sampled multiple target points and attached a random energy in the interval $E = [70, 140]$ to each of them, giving rise to multiple sets $\{x, \phi, \epsilon\}$. The geometry, angle and energy values were then given as input to MCsquare to determine the output dose volume y. Finally, each set $\{x, \phi, \epsilon, y\}$ was stored as a single training sample.

Target point selection The method of target point sampling went through two major iterations. While any sample that models a clinically achievable ray could be used for our training data, there are a few considerations to make. For the models to perform well on unseen test samples, the training data needs to include enough variety between the different geometries, angles and energy values. We do not want to select a target point that causes the modeled ray to miss the patient partly or entirely, and therefore we want to select target points that are not too close to the

limits of the patient geometry. Additionally, the total amount of data points needs to be kept reasonable with regards to the speed of the model training and the data generation itself.

- **Cycle 1:** For the first generation cycle, target points were taken from a regular grid across each patient geometry. The grid was constructed automatically for each geometry such that all corresponding rays fell well within the boundaries of the patient geometry. Figure 10 gives an example of the construction of such a target point grid. A rectangular range was first constructed such that all points within it are expected to lie within patient geometry limits. Then, the regular grid of target points was generated within this range with no point lying too close to the boundaries. The predetermined step size of the target point grid was 30 mm for both axes.

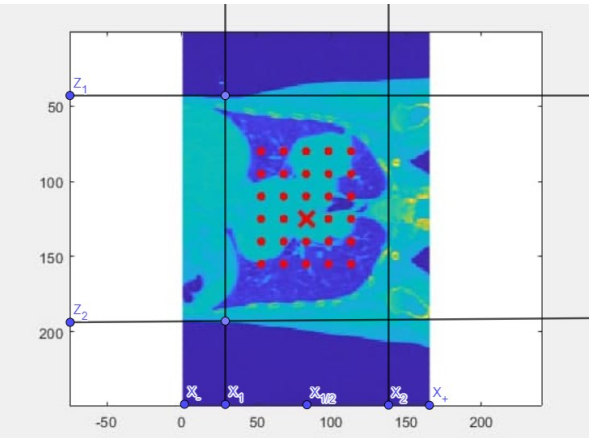

Figure 10: X/Z slice of a lung patient CT scan illustrating the first generation method with target point grids. The red cross is the isocenter of the CT, with the nozzle exit located exactly above it. Red dots are target points that were sampled. The values X_1 and X_2 are fixed at two-thirds from the isocenter to either patient geometry limit on the X-axis. Then, Z_1 and Z_2 are fixed such that all points in the rectangle with vertices (X_i, Z_i) at isocenter depth lie within the patient limits.

  After analyzing the first generated dataset, it became clear that too few different angles were considered. The first training dataset had around 100 different angles in total, while the amount of angles that can be selected in clinical practice is more than 10.000. We initially hoped that the model would be able to interpolate the smaller angles from the training data, but after training on the first dataset and evaluating the results we saw that the models were unable to predict these smaller angles accurately. This led us to include many more different angles in the data generation cycles going forward.

- **Cycle 2 & 3:** From the second generation cycle onward, we introduced randomness to our target point selection method. We could not use every target point in the available search space, as this would result in more than 8000 different data points for each patient geometry. Since we wanted to use multiple patients and multiple rotated geometries from each patient, we limited the amount of angles chosen to keep the total amount of samples manageable. We chose to randomly sample 50 points within a fixed distance of the patient geometry limits.

Dataset clean-up Upon inspecting the generated datasets, we found that many of the samples contained largely empty geometry volumes, with dose volumes that were

accurately simulated but mostly consisted of low amounts dose deposited in air outside of patient limits. These samples were the consequence of the automated target point selection procedure, which selects as many diverse points as possible and therefore selects many edge cases. The randomness in the selection procedure for the second dataset also generated many samples that did not contain relevant information.

To exclude these trivial samples from the training data, we computed the percentage of air voxels in the geometry volume of each sample. Samples where this percentage is higher than 80% are sure to contain no relevant information for the model, while samples with a percentage between 40% and 70% usually contain tissue on one side of the geometry volume and air on the other. These last samples correspond to rays that only graze the patient, and therefore deposit some radiation in tissue but mostly go through air. Rays like these are unlikely to be applied in clinical practice, so we decided to select the subset of samples with less than 40% air voxels in the geometry as an improved dataset.

Multiple samples per geometry One idea that the original DoTA model made use of, was the inclusion of samples with the same geometry x but different energy values ϵ. Intuitively, presenting the model with dose distributions in the same geometry but with two different energy values taught the model the impact of the energy value on the output dose, and therefore made the model more robust to changes in ϵ. This idea was adopted to make our model more robust to changes in angle of entry as well.

After sampling two random energies and computing the corresponding two dose volumes, the CT and target point were shifted 7 mm in a random cardinal direction. This simulates the change in dose distribution that is obtained when moving the nozzle while retaining the same target point, resulting in a slightly different angle of entry for the same target geometry. For this new angle, two dose volumes were then computed with the same two energy levels. The volumes were then cropped around the point of entry and the four resulting samples consisting of one geometry volume, four dose volumes, two energy levels and two pairs of angles were stored together to reduce the amount of storage space needed. The difference between these dose volumes (and those used to train DoTA, in which the rays traveled straight down) is illustrated in Figure 11.

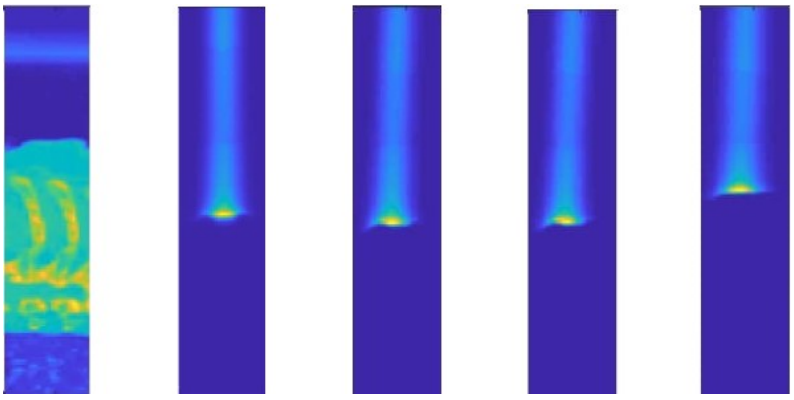

Figure 11: Examples of generated dose volumes illustrating the effect of changing the parameters. (1) Sliced patient geometry (2) Dose volume with no imposed angle (as used for original DoTA, not used here) (3) Dose for first energy and pair of angles (4) Dose with the same energy but slightly different angles (5) Dose with same angle as (3) but lower particle energy.

This method was taken a step further from the second generation cycle onward. Trained on the first dataset, we saw the model struggle to accurately capture the relation between the angle values and output dose. We therefore included more samples with the same geometry and energy but different angles, in an attempt to further increase the models sensitivity to changes in angle of entry. In particular, for 10 target points out of every 50, we took the corresponding angles ϕ and generated 10 more samples, corresponding to the same geometry and energy but with the opposite angles $-\phi$. This simulates a shift of the nozzle across the target point in the X/Z plane, to the exact opposite relative position.

Combining the samples from all cycles of data generation, the final training dataset consisted of 30765 samples, with diverse patient geometries and angle values. Details on the full training dataset can be seen in Section 5.1, including the total amount of samples, the distribution of all angles in the training dataset and the impact that removing the trivial samples had on the angle variety. The shape volumes s were constructed for each sample in the dataset, so that it could be used to train both models DoTA-A and DoTA-S. The next section describes the architectures of both models, as well as hyperparameter choices made through model validation.

4.3 Model architecture and training

Model architecture The models DoTA-A and DoTA-S have similar architectures, both consisting of a convolutional encoder, a transformer encoder and a convolutional decoder. A schematic description of the model architectures is given in Figures 13 and 14.

The first layer of both models is the **convolutional encoder** which takes the geometry volume $x \in \mathbb{R}^{L\times H\times W}$ as input, considering the horizontal slices $\{x_i \in \mathbb{R}^{H\times W} | i = 1, \dots, L\}$. In the case of DoTA-S, the volume s is taken as input as well and is treated the same way. The encoder consists of two convolutional blocks and one final convolution. Each convolutional block consist of the following sequential layers (see also Figure 12):

1. A **convolutional layer**, which performs a convolution operation on each x_i separately. Here, we use kernels of size 5×5 in 64 channels, with each channel having a different set of weights for its kernel and producing a different output. The kernel weights are regularized when updated to avoid problems with vanishing gradients. To ensure the output of each channel has the same size as the input, we apply padding before the convolution operation. For the first convolutional block in the encoder, this means we add rows and columns of zeros at the edges of each x_i so that there are $H \times W$ possible kernel positions. The output of this layer is an abstract feature map $\{z_i \in \mathbb{R}^{H\times W\times 64} | i = 1, \dots, L\}$.

   Note that for the second convolutional block in the encoder, the input is of the form $\{z'_i \in \mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 64} | i = 1, \dots, L\}$ as a result of the max pooling layer described below. Therefore, we now use kernels of size $5\times 5\times 64$ in 64 channels, and we proceed in the same way as before to produce an output of the same size as the input. For the convolutional decoder, the kernel sizes are adjusted in a similar fashion.

2. A **group normalization layer**. Normalizing feature maps is known to enhance model training, and for our models we used group normalization with $G = 16$ groups (Wu and He 2018), although many different options for normalization are available. Group normalization splits the feature map sequentially along

the channel axis into G groups, which in our case results in groups with shape $L \times H \times W \times 4$. For each of these groups the mean and standard deviation is computed, and the groups are normalized accordingly.

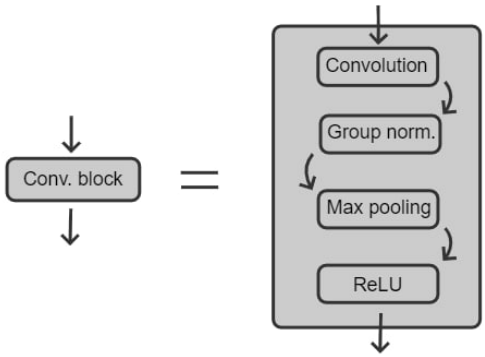

Figure 12: The structure of each convolutional block in DoTA-A and DoTA-S.

3. A **max pooling layer**. We use a 2×2 filter for the pooling operation which reduces the size of the feature map to $L \times \frac{H}{2} \times \frac{W}{2} \times 64$. Each $H \times W$-sized grid is split into 2×2-sized patches, and the value of an element in the $\frac{H}{2} \times \frac{W}{2}$-sized output equals the maximum of the 4 values in the corresponding patch.
4. A **Rectified Linear Unit (ReLU) activation layer**. Activation layers allow convolutional neural network to compute non-trivial problems by applying a non-linear function (note: all functions that were applied to the input data so far were linear). The most popular non-linear activation function for deep neural networks is ReLU, which is defined as $f(x) := \max(0, x)$, applied element-wise to the feature map. ReLU has a number of advantages compared to other activation functions like the logistic sigmoid and the hyperbolic tangent, such as scale-invariance, efficient computation and often producing sparse outputs.

The final convolution uses a predetermined number of K channels (the K corresponding kernels have size $\frac{H}{4} \times \frac{W}{4} \times 64$) and embeds the resulting sequence of elements in tokens $\{z_i | z_i \in \mathbb{R}^D, i = 1, \ldots, L\}$ for $D = \frac{H}{4} \cdot \frac{W}{4} \cdot K$.

After the convolutional encoder, we add tokens for the energy and the angles as $z_\epsilon = W_\epsilon \epsilon \in \mathbb{R}^D$ and $z_\phi = W_\phi(\phi_X, \phi_Z) \in \mathbb{R}^D$ respectively to the encoder output, where the linear projections $W_\epsilon \in \mathbb{R}^{D \times 1}$ and $W_\phi \in \mathbb{R}^{D \times 2}$ have learned weights. The next layer is a **transformer** encoder, which takes the $L + 2$ tokens as input to extract information about the inter-dependence of the tokens through its self-attention mechanism. We refer to Section 3.5 for an explanation of the inner workings of the transformer layer.

We use a single transformer layer with a predetermined number of N_h attention heads. Transformer blocks use a large amount of learned weights to capture the inter-dependence of the different tokens, making optimizing models that use them especially data-hungry. In our case, depending on the hyperparameters, the transformer layer makes up for $90 - 99\%$ of the total learned weights of the models. For the original DoTA model, increasing the amount of transformer blocks in this layer did not improve

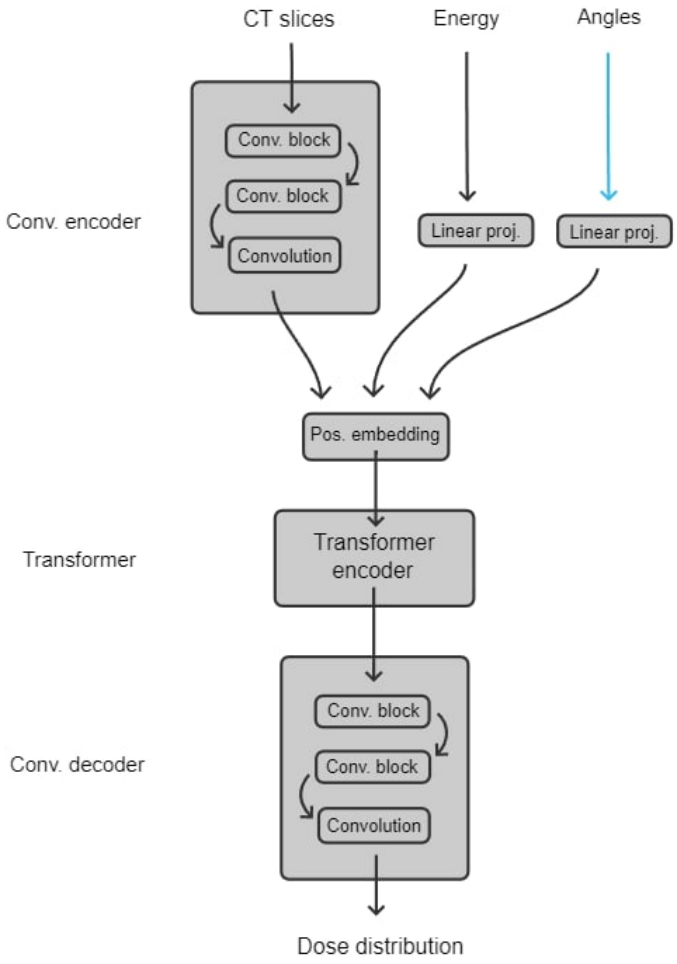

Figure 13: The architecture of DoTA-A.

model performance while greatly increasing the model size, probably due to the large amount of data needed to optimize. DoTA-A and DoTA-S are expected a priori to need more data to achieve similar results as DoTA, since the relation that needs to be captured is more complex. Based on this observation, DoTA-A and DoTA-S use only one transformer block as well. A similar reasoning holds for the amount of convolutional blocks in the convolutional encoder and decoder, which we kept at two.

The transformer is particularly suited to our setting when using causal self-attention, which does not check later elements in the sequence for relevance to earlier tokens. In our case, energy deposition happens mostly sequentially in the forward beam direction, meaning that to accurately predict dose values at any depth, deeper geometric information is not expected to be relevant. Causal self-attention only considers the relevance of earlier tokens, so for each of the L geometric tokens, only geometric information above the corresponding layer is considered.

The output of the transformer has the same size $(L + 2) \times D$ as its input. The **convolutional decoder** transforms these tokens into the desired size $L \times H \times W$

with two convolutional blocks and one final convolution, similarly to the convolutional encoder. The main difference between the convolutional encoder and decoder is that for the decoder all convolutions are transposed, which increases the dimension of their input. The final convolution transforms the output of the second block into 2D dose slices $\{y_i \in \mathbb{R}^{H\times W} | i = 1, \ldots, L\}$ which are combined to produce the output dose volume y.

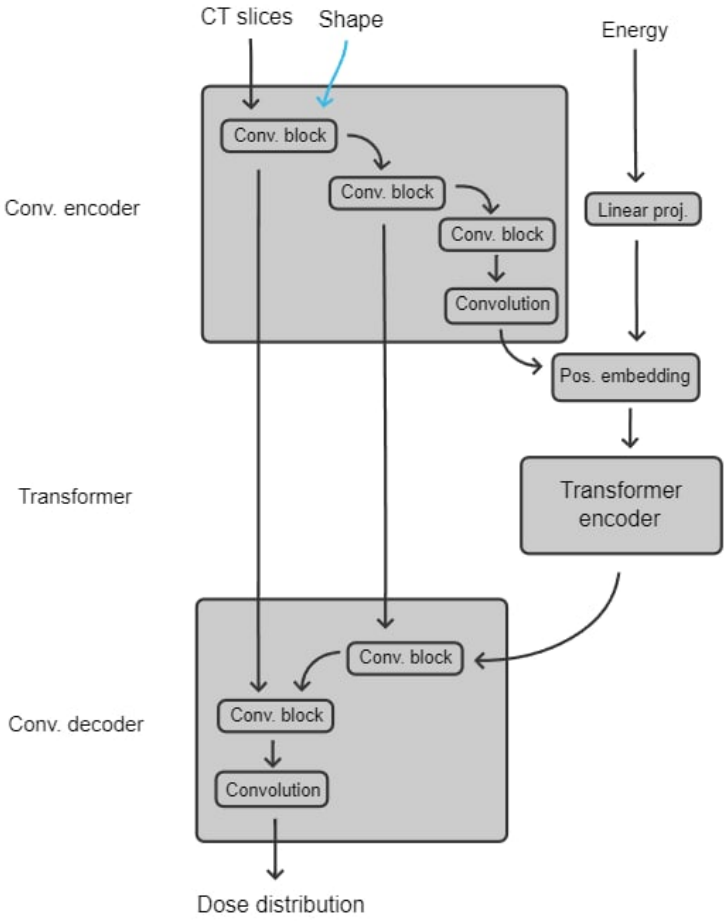

Figure 14: The architecture of DoTA-S. The residual connections provide a path to deeper layers that bypasses the convolutional layers in between.

In DoTA-S, the geometric input is presented together with the shape volume as $(x, s) \in \mathbb{R}^{L\times H\times W\times 2}$ (the kernels in the first convolutional layer thus have size $5\times5\times2$). To exploit the similarity between the input and output data, we include residual connections in its architecture. The reasoning here is that on top of the general training speed increase that residual connections bring, data will be able to flow to deeper layers in the model without passing through every convolutional layer. Since the input shape volumes s and the output dose volumes are similar by construction of s, the shorter path from input to output could help the model recognize this similarity and use it in its predictions.

Training Using these architectures, DoTA-A and DoTA-S were trained with the generated training data. Using a voxel resolution of 2 mm × 2 mm × 2 mm, each cropped geometry, dose and shape volume in our training set has a size of $150 \times 25 \times 25$ voxels.

The training data is loaded in mini-batches of 8 samples. Upon loading each sample, we randomly pick a rotation angle $\alpha \in \{0°, 90°, 180°, 270°\}$ and use this angle for a rotation of the geometry, dose and shape volumes around the vertical axis (i.e. the depth axis Y). By adjusting the angle values $\phi = (\phi_X, \phi_Z)$ accordingly to one of the four options $(\pm\phi_X, \pm\phi_Z)$ and $(\mp\phi_Z, \pm\phi_X)$, the augmented samples are then still valid. This augmentation step increases the effective amount of samples available to the models. Since the convolutional encoder embeds the input data into tokens of length $D = \frac{H}{4} \cdot \frac{W}{4} \cdot K$, we then crop the geometry, dose and shape volumes such that $H = W = 24$.

For the training procedure, we use the end-to-end machine learning platform Tensorflow (Abadi et al. 2016) in Python 3.8. To optimize the model weights, we use the LAMB (Layer-wise Adaptive Moments optimizer for Batch training) optimizer. The LAMB optimizer uses the popular ADAM (Adaptive Moment Estimation) algorithm as its basis, but adds normalization steps to the update rules which make training more stable for large batch sizes. ADAM itself is based on standard stochastic gradient descent methods, but instead of updating the weights by just the gradients of the loss function, the moving averages of the gradients are taken into account. For a detailed description of the ADAM and LAMB optimizers, we refer the reader to (You et al. 2019).

We use MSE as a loss function, which measures the error between the model output and given baseline dose. We train the model for 90 epochs (i.e. iterations over all training data). The learning rate starts at 10^{-3}, is halved every 4 epochs, and reset to 10^{-3} after epoch 30 and 60. Of the full training dataset, 90% was used for training and a fixed 10% of samples was used as a validation dataset. After each epoch, the model performance on the validation dataset was measured, and the validation loss after all 90 epochs was used as the main validation metric. The weights resulting in the lowest validation MSE was saved for each model.

Validation The above model architecture contains two main hyperparameters: the amount of channels K in the convolutional encoders final convolution, and the amount of attention heads N_h in the transformer block. Both of these hyperparameters are strongly connected to the models size. Recall that over 90% of the learned weights of the model are used to model the inter-dependencies between token elements in the transformer block. K determines the token dimension $D := 36K$ of the convolutional encoder output, so increasing K also increases the amount of weights in the transformer. Increasing the amount of attention heads N_h adds more sets of weights to the transformer and therefore has the same effect.

To choose the values of these two hyperparameters, we performed validation on the validation set consisting of a fixed 10% of our dataset. We performed a grid search across all combinations of $K \in \{12, 16, 20, 24\}$ and $N_h \in \{8, 16, 24, 32\}$. After training both models with each of these combinations, the lowest validation loss was achieved for $K = 16$ and $N_h = 32$ for DoTA-A, and $K = 24$ and $N_h = 32$ for DoTA-S. These are the hyperparameter values that were used to evaluate our models, which we describe in Section 4.4.

The best performing hyperparameters including a high amount of attention heads for

both models can be interpreted as a sign that many interdependencies exist between the tokens. With highly inhomogeneous geometries from a variety of treatment sites, attention heads can be devoted to the interdependence of specific geometrical patterns and the dose around or below them. For example, an attention head could capture information about the behavior of dose deposition around different tissues. Attention heads are also used to store relevance of the energy and angle tokens.

DoTA-S gives the best validation results when using 24 channels in the final encoder convolution whereas DoTA-A performs the best when using 16 channels. Increasing the amount of channels can increase the amount of features that are detected in the convolutional encoder, and since DoTA-S gives the shape volume s as input for the convolutional encoder as well as the patient geometry volume x, the extra channels could help the model detect features from the shape volume.

4.4 Evaluation

To evaluate the models presented in this project we compared their predictions on a test set to the baseline dose distributions coming from Monte Carlo simulations. We compare the DoTA-A and DoTA-S models to the MC baseline to see how well they approximate the ground-truth, and compare the results to those of PBA generated dose distributions, calculated in matRad as well (Cisternas et al. 2015).

Additionally, for each test sample we want to evaluate the inclusion of the angle information as a whole, relative to models that do not include any angle information such as the original DoTA which is trained on perpendicular rays only. To measure this difference, we calculate another MC dose distribution for each sample, this time with each ray entering the geometry perpendicularly at the target point. Any model that does not consider angle of entry is at most as accurate to clinical reality as this method, which we call *perpendicular MC*.

Test sets The test sets were generated separately from the training sets, so the evaluations test the models performance on samples that they have not seen before. A few different test sets were used: water volumes, single rays and full plan.

- For the water volume test set, the dose distributions for proton beams through water were generated using MC square. Since the geometry is completely homogeneous, this can be considered a benchmark evaluation method for any dose calculation algorithm. The models were only trained on samples with realistic patient geometries, so the question is if the models are able to extrapolate the particle physics in a simple setting that is unlike anything it has seen so far. The test samples all contain an empty water phantom geometry x, and the target points were chosen in a fixed 7×7 grid to generate 49 different angles of entry. For each of these angles, two samples were generated using energy values of 90 and 120 MeV, for a total of 98 samples in this test set.
- The single ray test set consists of samples generated in an identical way to the samples in the training set. A patient CT was used for these samples that was not used for the training set so that the inputs are new to the models. However, since the models are trained on samples constructed in the exact same way, we expect the performance on these samples to be good. We use a target point selection as in the second and third generation cycle, meaning this test set includes a variety of smaller and larger angle values. After removing samples

with less than 60% non-empty geometry (as for the training set), this test set consisted of 228 test samples.

- For the full plan test set, a treatment plan for a lung patient (which was not used in the training set) was fixed, consisting of 2 beams of ±1000 rays each. Each ray has its corresponding target point, energy value and relative weight listed in the treatment plan. For each of these rays the baseline dose distribution was calculated using MCsquare, and the individual ray dose distributions were added together to create the full plan dose distribution. The models then predicted the dose distribution for each of the rays as well, and the full plan distributions were compared. This is the context in which dose calculation algorithms are used clinically, and the results are not necessarily the same as those for the single rays. For example, positive and negative inaccuracies could cancel out when added together in the full dose distribution, and the error for lower energy rays which account for less dose in the full plan have less impact than higher energy rays.

Evaluation metrics To compare our models on the test sets described above, a few different metrics were used. First, we measure the average speed it takes for the models to produce the dose distributions. To evaluate prediction accuracy, the most straightforward metric is MSE, which was also used as loss function for the training of the models. Additionally, for a given prediction and baseline dose distribution, we calculate the absolute dose difference between them for each voxel. We calculate the maximum, mean and standard deviation of the error across all voxels in a single sample, and average the resulting values over all samples in the test set. For the mean voxel dose difference, we also include the variance and maximum across the test samples.

The unit used for these calculations is percentage of maximum baseline dose, meaning that for each sample we divide both prediction and baseline dose values by the maximum dose value in the baseline dose distribution. This means that the same error percentage for different test samples might not correspond to the same actual dose difference, but instead refers to the relative accuracy of the prediction for each sample.

We also exclude all voxels with a dose value of zero in both the prediction and the baseline from our calculations. Since a ray only deposits dose along its path and our cropped dose volumes have fixed size, most voxels in the dose volumes generally have a dose value of zero. This causes the mean of the error values to be very close to zero and makes them harder to interpret. This is especially apparent for the full plan test set, where in a dose volume with more than 10^7 voxels only 3% to 4% of them typically have non-zero dose values.

One caveat to the exclusion of zero-dose voxels is that a prediction could be very inaccurate but have many inaccurate low dose-value voxels included in the calculations. Even if the corresponding baseline dose values are zero, the low dose-value of these voxels causes the error to be close to zero as well, and the average error across all voxels could give the impression that the prediction is relatively good. The inaccuracy for such samples should however be reflected by a high deviation and maximum error across all voxels. MSE evaluations, although harder to interpret in a dosimetric sense, also avoid this problem and offer a more objective method for comparing relative model performance.

Gamma evaluation Another comparison tool for the predicted and ground-truth dose distributions is gamma analysis (Low et al. 1998). Gamma analysis is based on the observation that two dose distributions can be functionally identical even when their voxel-by-voxel difference (such as measured by MSE) is large. A prediction with a high dose value in a certain voxel could have a baseline with the same high dose value in an adjacent voxel. Even though the exact voxel dose difference could then be high, the prediction can still be considered accurate for practical purposes and this is reflected in the gamma value.

For each voxel in the predicted dose distribution, gamma analysis searches a voxel in the ground-truth dose distribution with a dose value that is similar enough and which lies within a certain range. The thresholds are expressed through the maximum dose difference parameter δ and the distance-to-agreement parameter Δ. Common values for clinical gamma evaluations are $\delta = 1\%$ and $\Delta = 1$mm, which means that a dose value within 1% of the baseline dose is searched in a sphere around the target voxel of radius 1mm. Mathematically, the gamma value is calculated as the minimum distance in dose-distance space between the target point in the prediction and any point in the baseline:

$$\gamma(p) := \min_{\hat{p}} \{\Gamma(p, \hat{p})\}$$

$$\Gamma_{\delta,\Delta}(p, \hat{p}) := \sqrt{\frac{|p - \hat{p}|^2}{\delta^2} + \frac{|D(p) - D(\hat{p})|^2}{\Delta^2}}$$

where p are the coordinates of a point in the predicted dose volume, $\hat{p}$ are the coordinates a point in the ground-truth dose volume, $D(p)$ is the dose value at any point and δ and Δ are the threshold parameters described above. In this description, a voxel in the prediction is considered to pass the gamma evaluation when $\gamma(p) < 1$. For an entire dose prediction, the gamma pass rate is then defined by the percentage of all voxels which pass the gamma evaluation.

Since our models were trained and evaluated on a 2 mm resolution, we used gamma evaluation parameters of $\delta = 1\%$ and $\Delta = 3$mm for our single ray test sets so that the range includes at least the surrounding voxels. For our full plan evaluations, we used the four parameter combinations $(\delta, \Delta) \in \{(1, 3), (1, 1), (2, 2), (3, 3)\}$. When performing gamma evaluations for $\Delta = 1$ mm on 2 mm resolution predictions, the necessary dose values are interpolated from the surrounding voxels. These evaluations are therefore not expected to produce a high pass rate but are still included for completeness.

5 Results

In this section, we present the results of this research project. We first show statistics on the training dataset that was generated, with particular attention to the angle values which are the focus of this project. We then give and discuss the computation speeds of the different dose calculation algorithms which we compare. Finally, we present the evaluation results across different evaluation metrics and test sets, including single ray test samples and a full treatment plan simulation.

5.1 Training data

With the generation methods described in Section 4.2, the training dataset for our models was generated over three generation cycles. Table 1 describes the results for each cycle. Each sample consists of one cropped geometry volume $x \in \mathbb{R}^{150\times25\times25}$, two energy values ϵ, two pairs of angles $\phi = (\phi_X, \phi_Z)$ and four dose volumes $y \in \mathbb{R}^{150\times25\times25}$ corresponding to each energy-angle combination. Here, the geometry volume is cropped such that the ray's point of entry lies exactly at the coordinate $(1, 13, 13)$, which is the center voxel on the first layer.

The data was generated from a dataset of 53 patient scans, with the scans taken at different regions of the body depending on the patients tumor location. The patients can thus be categorized as being either Head and Neck (H&N), Lung (L) or Pelvic (P) patients. Some scans include multiple regions as well. For each of the second and third cycle, we selected only two patients for each region but generated a much larger amount of different angles for each patient, resulting in a similar amount of samples as the first cycle with more angle variety.

Cycle	# of samples (with $< 40\%$ air)	# of patients (H&N/L/P)	# of angles
1	17469 (11499)	53 (21/22/14)	411
2	13449 (9748)	6 (2/2/2)	11469
3	13500 (9518)	6 (2/2/2)	12684
Total	44418 (30765)	-	15880

Table 1: Construction of the training dataset over three cycles.

Angles Figure 15 below describes the angles $\phi = (\phi_X, \phi_Z)$ that appeared in the samples generated over our three generation cycles. We see that the first cycle only produced few fixed angles in a grid across the available range of $[-5.16°, 5.16°]$ in either direction. The second and third cycle use a stochastic approach, and generate a more diverse set of angles.

Figures 15 and 16 show that the training data consists of a large variety of angles of entry, with the most commonly used angles in the range $[-2°, 2°]$ appearing more frequently. Values of ϕ_X and ϕ_Z that are close to the machines physical limitations are uncommon in clinical practice. Since the patient and gantry arm can also be rotated for a treatment plan, these rotations are usually chosen such that the target points lie relatively straight under the nozzle exit. This does not take away from the relevance of the angle of entry, since each beam consists of multiple rays diverging to a collection of target points. Note that by convention, patients are oriented along the Z-axis in the CT scans. Thus, although rays with a high value of ϕ_Z might still deposit dose in patient tissue, high values of ϕ_X typically correspond to the ray traveling through mostly air and therefore being deleted in our dataset clean-up.

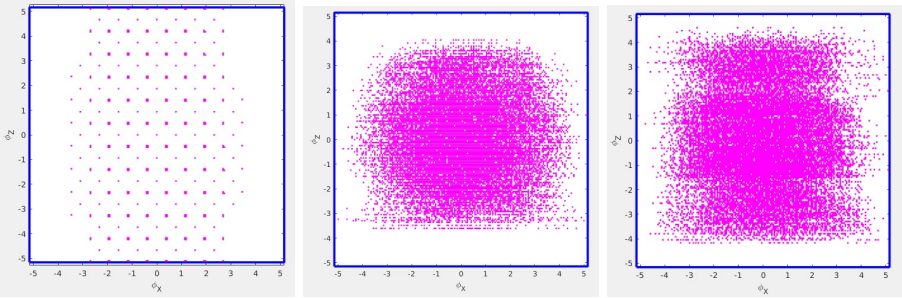

Figure 15: The angles of all samples generated per generation cycle, in degrees (from left to right: cycle 1, 2 and 3). The blue outline represents the maximum clinically achievable angles.

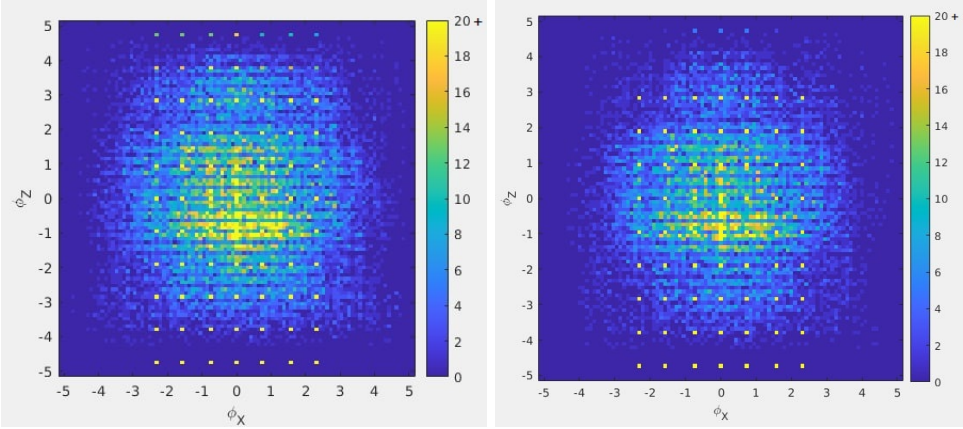

Figure 16: Heat-map of all angles in the final training set, colored by number of appearances. Left: original training set after 3 cycles. Right: after removing samples with over 40% air in the geometry. For visibility, angles appearing between 20 and 30 times are grouped together.

In Figure 16, the angles that appear in our training set (combining data from all three generation cycles) are shown in a heatmap. The full range of angles is discretized into a 100×100 grid and each voxel is colored by the amount of angles in our training set that lie inside of it.

In the right figure we see the angles that remain after excluding those samples whose geometry volume consists of more than 40% air voxels (as per Section 4.2). This causes 13 653 samples to be excluded from the training set, which was around one third of the original amount. Removing these samples does not significantly reduce the amount of different angles in the training set: the reduced training set still has 13 219 different angles, meaning 2 661 were discarded. We see that the distribution of angles is also not distorted by the removal of samples with more than 40% air in the geometry.

For each sample in the final training set, we constructed shape volumes s for all four energy-angle combination, using the methods described in Section 4.1. This training set was then used for the training of DoTA-A and DoTA-S, which we will describe in the next section.

5.2 Prediction speed

In this section we discuss the prediction speed of the dose calculation algorithms used for our comparisons. Table 2 lists the average time per dose it took for each algorithm to calculate the dose distributions of samples in our test sets. The predictions for DoTA, DoTA-A and DoTA-S were performed using GPU hardware (NVidia Quadro RTX 6000) and for PBA and MC we used CPUs for the calculations (Intel Xeon CPU E5-2690) utilizing 8 CPUs in parallel.

Model	Average runtime per ray (s)
DoTA	0.005
DoTA-A	0.013
DoTA-S	0.018
PBA	0.83
MC	38.40

Table 2: Average runtime per test set ray of various dose calculation algorithms.

We see that the prediction speed of DoTA-A offers prediction speed of a factor 60 faster than PBA, with DoTA-S being slightly slower and offering a factor 40 reduction. Both models DoTA-A and DoTA-S are slightly slower than the original DoTA. Compared to MC simulations, DoTA-A and DoTA-S offer a 3000 and 2000 times reduction in calculation time respectively.

The difference between the average runtimes of DoTA, DoTA-A and DoTA-S is likely due to the different model sizes. The original DoTA used around 10^7 learned weights, while DoTA-A and DoTA-S use around $4.3 \cdot 10^7$ and $9.7 \cdot 10^7$ learned weights respectively. The amount of learned weights is roughly proportional to the total amount of operations that the models perform on its input to compute the corresponding output. As we see, the proportions of the three model sizes roughly correspond to the proportions of the model prediction speeds. We might expect DoTA-S to be more than twice as slow as DoTA-A because it has more than twice as many learned weights; the roughly 40% increase in runtime might be an indication that many of the weights in DoTA-S are set to zero, simplifying the operations that the model performs.

5.3 Accuracy

This section lists the results of our evaluations on the different test sets. As described in Section 4.4, we first measure the absolute difference across the voxels within a sample, only considering non-zero dose value voxels. The mean, standard deviation and maximum for the relative error across all voxels is then calculated, and the average of these metrics across all test samples is measured. We also compute the deviation and maximum across the test set over each samples mean relative error.

We additionally computed the gamma pass rates of voxels with non-zero dose value, using a dose difference threshold of $\Delta = 1\%$ and a distance threshold of 3mm. This gave a percentage of passed voxels for each sample. The average gamma pass rate of all samples in the test set is provided, along with the standard deviation across all samples and the lowest and highest pass rates in the test set.

Water In tables 3 and 4 we see the results for the water phantom test set. We see that DoTA-S is the best performing model for this test set, scoring a gamma pass rate close to 100% for all samples in the test set. The models have not encountered

Water test set	Error in % of max. baseline dose				
(over voxels in a sample)	Mean absolute error			σ	Max.
(over all samples)	μ	σ	Max.	μ	μ
DoTA-A	0.91%	0.24 %	1.57%	1.43%	19.98 %
DoTA-S	**0.60%**	0.12%	**0.84%**	**0.96%**	**13.59%**
PBA	1.22%	**0.06%**	1.40%	2.93%	46.73%
Perp. MC	2.77%	1.21%	5.33%	3.98%	35.15%

Table 3: Error between model predictions and MC baseline for the water test set.

Water test set	γ pass rates				MSE
(over all samples)	μ	σ	Min.	Max.	μ
DoTA-A	99.18%	0.85%	97.11%	100%	3.13
DoTA-S	**99.88%**	**0.14 %**	**99.15%**	**100%**	**1.32**
PBA	92.78%	4.11%	81.32%	99.41%	10.26
Perp. MC	87.14 %	12.44 %	60.16 %	100 %	28.01

Table 4: Gamma pass rates and MSE of all models for the water test set. The gamma pass rate is the percentage of non-zero dose value voxels that pass gamma evaluation with $\delta = 1\%$ and $\Delta = 3$mm.

empty geometries such as the ones used in this test set during training, so the high performance demonstrates the models' capacity to extrapolate accurately from the training data. Figure 17 shows the different model predictions for the test sample where DoTA-S had the worst performance.

The high performance of DoTA-S compared to DoTA-A can be explained by the increased similarity of the shape volume s to the target dose in this test set. For inhomogeneous geometries, the shape volume will only be a rough outline of the shape the dose is expected to take, but it does not consider geometry. For the water test set, the shape of the actual dose is continuous in a similar way to the shape volumes that DoTA-S takes as input.

The performance of PBA is poor compared to the other models, but gives a higher accuracy for the water test set than for the other test sets. We expected to see this behavior since the way PBA calculates dose by drawing a central pencil beam and expanding the dose around it is very similar to the dose deposition behavior of proton rays in water volumes. PBA is known to provide less accurate predictions in inhomogeneous geometries, as we will see for the next test sets. While the perpendicular MC predictions are fairly accurate for samples with smaller angle values, it gets increasingly less accurate for larger angles.

To visualize which type of sample the models have the worst performance on, we plot the SMSE of our model predictions for all samples in Figure 18, sorting the samples by either energy or angle values. The SMSE is defined as $\sqrt{\text{MSE}}$ where MSE is the standard mean square error between the prediction and baseline dose; we take the square root to simplify the visualization, since we are interested in relative error between samples. For the water test set, sorting by energy splits the test set in half since the 98 samples in the test set consist of 49 angles and a dose corresponding to the energy values 90 and 120 MeV for each angle. When sorting by angle values, we calculate $\phi_X^2 + \phi_Z^2$ for each sample as a metric for relative combined angle size, and sort the samples by these values. The perpendicular MC calculated doses are not shown in these plots, since we know that the inaccuracy of these predictions only depends on the angle values.

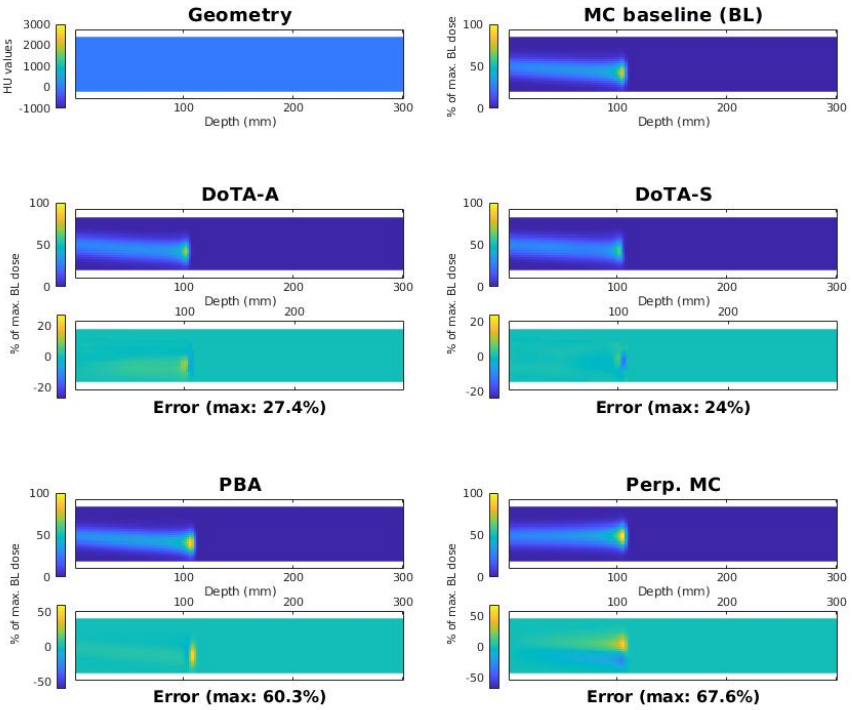

Figure 17: Predictions and relative error of the different models, showing the same central X/Y slice for each volume. This is the sample in the water test set with the highest MSE between DoTA-S predicted dose and the MC baseline ($\epsilon = 120$ MeV, $(\phi_X, \phi_Z) = (3.1°, -3.7°)$).

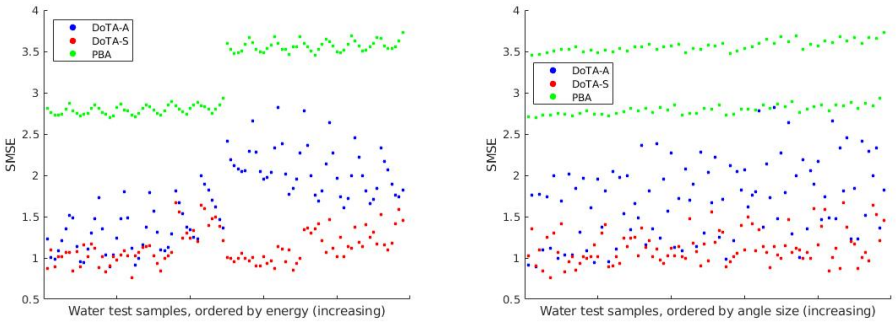

Figure 18: Left: SMSE of water test set samples for different models, sorted by energy. Right: Sorted by relative combined angle values instead.

As we see in Figure 18, out of the three models, DoTA-S is the most stable with regards to increases in both energy and angle values and outperforms the other models for most samples. DoTA-A seems to perform slightly worse on average on samples with large angle values, and the samples where DoTA-A has the worst performance have both high energy and angle size. We can also see that PBA is significantly worse when predicting doses with a higher energy value in this test set.

Test patient	Error in % of max. baseline dose				
(over voxels in a sample)	Mean absolute error			σ	Max.
(over all samples)	μ	σ	Max.	μ	μ
DoTA-A	1.45%	0.52%	3.89%	2.20 %	**35.42** %
DoTA-S	**1.09%**	**0.34 %**	**2.10%**	**2.10%**	36.13%
PBA	1.88%	0.85%	4.47%	4.71%	90.86%
Perp. MC	3.04 %	1.86%	11.25%	5.44%	57.95%

Table 5: Error between model predictions and MC baseline for the test patient dataset.

Test patient	γ pass rates				MSE
(over all samples)	μ	σ	Min.	Max.	μ
DoTA-A	97.07%	**2.25%**	**89.15%**	99.95 %	7.69
DoTA-S	**97.22%**	2.37%	86.95%	**99.99%**	**6.25**
PBA	84.74%	7.32%	60.99%	99.02%	30.03
Perp. MC	73.30 %	18.60 %	32.93 %	99.97 %	50.37

Table 6: Gamma pass rates and MSE of all models for the test patient dataset. The gamma pass rate is the percentage of non-zero dose value voxels that pass gamma evaluation with $\delta = 1\%$ and $\Delta = 3$mm.

Test patient In tables 5 and 6 we see the test results for the test set consisting of samples similar to the training data, using a patient geometry that the models have not yet encountered. For this test set, DoTA-S is again the best performing model for most metrics, although DoTA-A has similar results and performs slightly better by some metrics. The patient geometry is largely inhomogeneous, with the CT including the patients lungs as well as the head and neck area. This makes achieving a similar accuracy as for the water test set unlikely, but we see that DoTA-A and DoTA-S approximate MC baseline dose distributions better than PBA and significantly better than perpendicular MC simulations, which demonstrates the importance of the angle of entry in dose calculation algorithms.

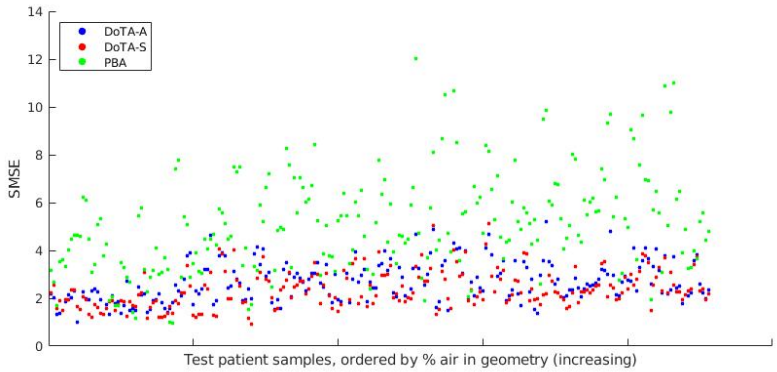

Figure 19: SMSE of test patient samples for different models, sorted by percentage of air in the sample geometry.

Like for the water test set, we plot the SMSE of all samples in Figure 20, sorting the samples by energy and angle values. The error values show that PBA struggles to achieve the same high accuracy as DoTA-A and DoTA-S in this test set, most likely due to the inhomogeneous patient geometry. To test this hypothesis, we additionally

sort the samples in this test set by amount of air in the geometry in Figure 19. Recall that we removed all samples with more than 40% air in the geometry from both the training and the test datasets.

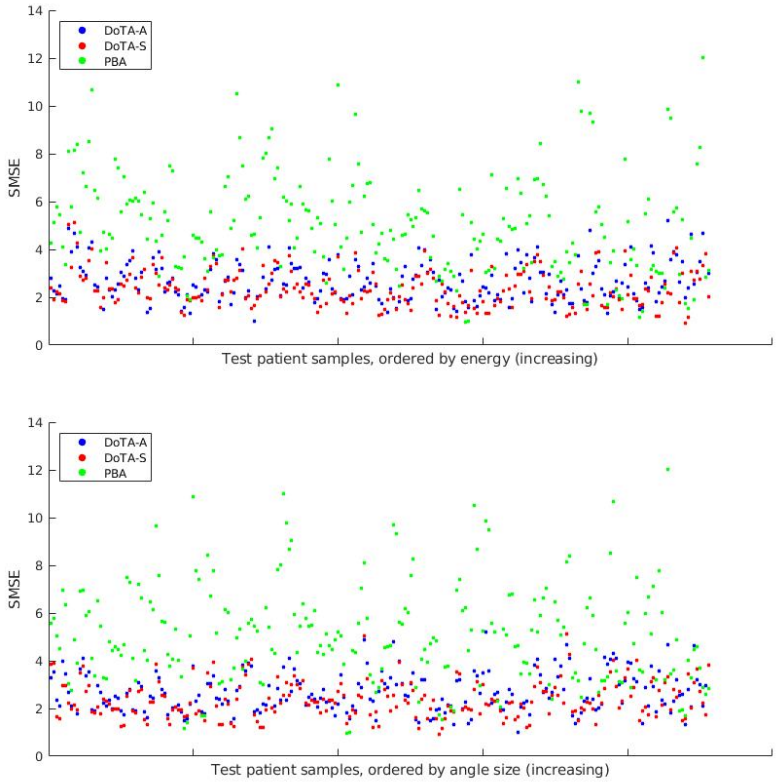

Figure 20: Above: SMSE of test patient samples for different models, sorted by energy. Below: Samples are now sorted by relative combined angle values.

Figure 19 shows that all models perform worse for samples with high percentages of air in the geometry. This is the most apparent for PBA, while the samples that give the worst results for DoTA-A and DoTA-S usually have a combination of high energy, angle values and air percentage. In Figure 20, we see that DoTA-A and DoTA-S display high robustness with regards to changes in energy and angle, and this observation supported by their low variance in error and gamma pass rates across the test set.

In Figure 22 we see the predictions of the different models for one sample of the patient test set. As we see, this dose passes through an inhomogeneous part of lung tissue, and the geometry volume consists of 35.1% air (not including the air in the lungs, which has a different radiodensity from air outside of the patient).

We see that PBA struggles to accurately predict the dose in this inhomogeneous setting and greatly overestimates the intensity and location of the Bragg peak. This confirms the cited problems with PBA predictions at lung treatment sites (Taylor, Kry, and Followill 2017), and we will see the same issue arise for the treatment plan evaluation. DoTA-S is the most successful for this test set, with DoTA-A performing only slightly worse.

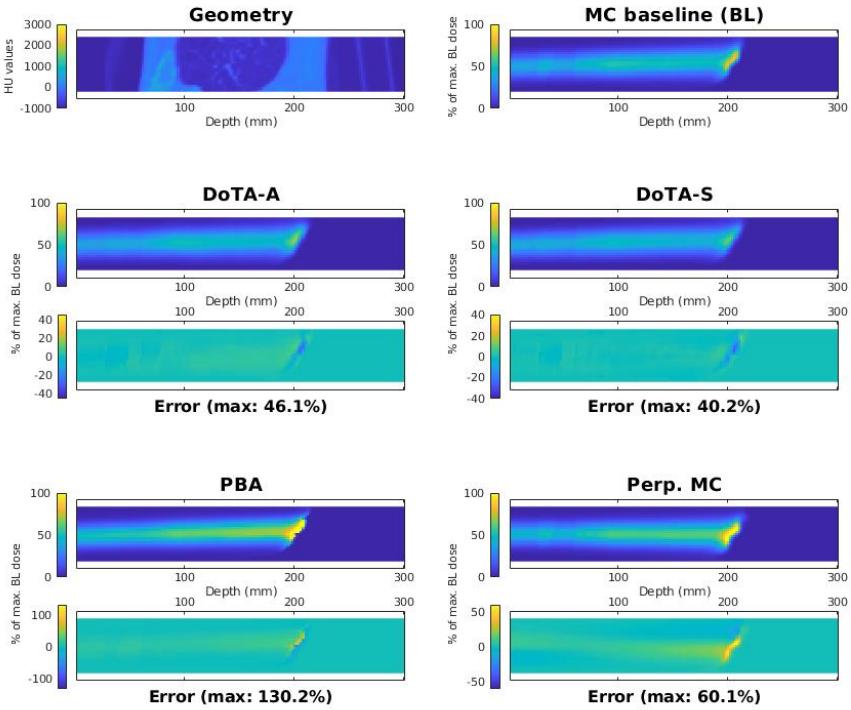

Figure 21: Predictions and relative error of the different models, showing the same central X/Y slice for each volume. This sample from the patient test set has 35.1% air in the geometry ($\epsilon = 98$ MeV, $(\phi_X, \phi_Z) = (-1.4°, -2.1°)$).

Treatment plan For the full treatment plan, we compare the predicted and MC baseline dose distributions resulting from a full plan of around 2000 rays. The plan consists of two beams that correspond to different gantry angles, and each beam consists of around 1000 rays using a variety of energy values and angles of entry. For each model, the dose distributions for each ray was inferred separately, and tables 7 and 8 show the accuracy of the full plan dose distributions obtained by summing the dose distributions for all rays together with their assigned relative weights.

Treatment plan	Mean absolute error	σ	Max.	MSE
DoTA-A	0.89%	**1.20%**	**18.20%**	**2.23**
DoTA-S	**0.88%**	1.39 %	22,88%	2.71
PBA	8.42 %	14.38%	156.64%	277.67
Perp. MC	3.27 %	4.99%	39.16%	35.60

Table 7: Test results for the full treatment plan, combining predictions for around 2000 rays. Absolute voxel difference is measured in % of maximum baseline dose value and considers only non-zero dose voxels.

Surprisingly, DoTA-A provides a more accurate full plan dose prediction than DoTA-S for this treatment plan. Across the 2245 individual rays that make up the treatment plan, DoTA-S dose predictions are more accurate than those of DoTA-A for more than half the rays. However, due to the rays having different weights and energy

Treatment plan	γ pass rates			
(δ, Δ)	$(1\%, 3\text{mm})$	$(1\%, 1\text{mm})$	$(2\%, 2\text{mm})$	$(3\%, 3\text{mm})$
DoTA-A	**97.60%**	**88.32%**	**97.68%**	**99.43%**
DoTA-S	95.74%	86.76%	95.71%	98.21%
PBA	66.34 %	44.92%	61.47%	71.83%
Perp. MC	78.33 %	57.11%	75.09%	84.84%

Table 8: Gamma pass rates for the full treatment plan. Measured over all non-zero dose value voxels in the treatment plan, combining the predictions of around 2000 rays.

values, the contribution of individual rays to the total dose distribution varies. In other words, a likely reason for the high performance of DoTA-A is the relatively high weights in the total plan of those rays where DoTA-S performed worse than DoTA-A.

Additionally, note that the voxel error for dose predictions of individual rays can be either negative or positive if the model respectively over- or underestimates the target dose values. When adding the dose distributions of individual rays together, combining the error values on one voxel can either cause the error to be canceled out or amplified. One other explanation for the higher performance of DoTA-A for this particular plan could therefore be that the individual ray errors of DoTA-A cancel each other out in the full dose distribution more frequently than for DoTA-S.

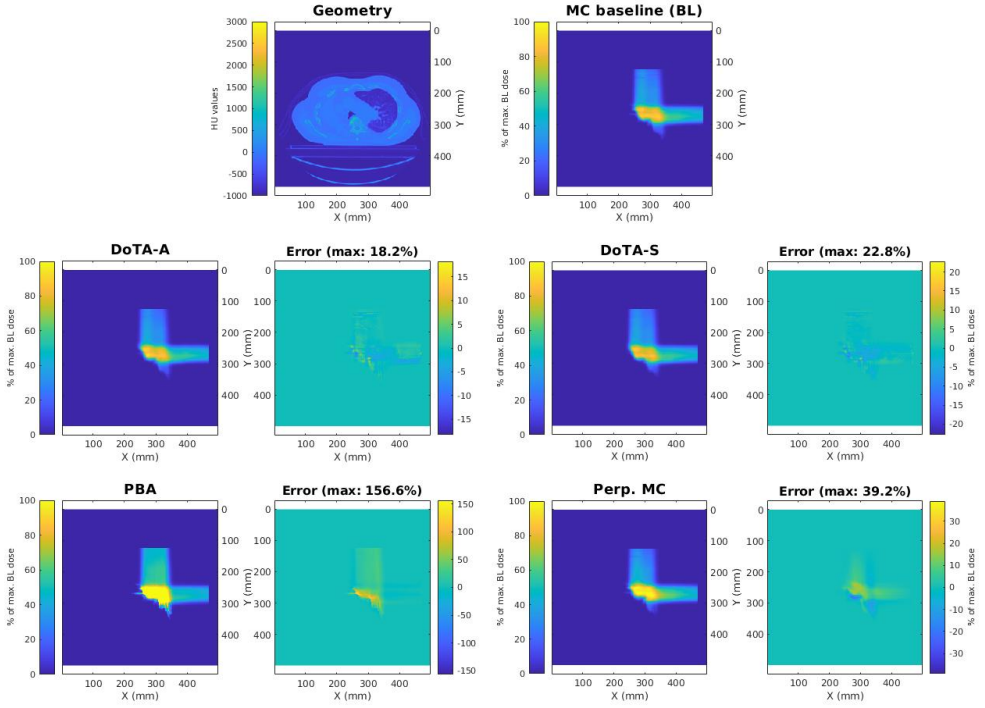

Figure 22: Predictions and relative error of the different models, showing the same X/Y slice at isocenter depth for each volume.

Likely due to the treatment plan being constructed for a lung patient, we see that the PBA predictions are very inaccurate for the full dose distribution. The PBA ray predictions overestimate and misplace the dose deposition, especially in the lungs where the baseline dose has the highest intensity. The positive errors in dose value

get amplified when the individual rays get added together to create a dose error of up to 150% of the maximum baseline dose. The gamma pass rate of our PBA prediction is still higher than found by Sorriaux et al. (2017), where for $\delta = 2\%$ and $\Delta = 2$ mm gamma pass rates of 44.7% was observed for a full lung patient plan dose distribution. For these parameters and our treatment plan, we find a PBA prediction gamma pass rate of 61.47%.

The perpendicular MC prediction suffers less from the inhomogeneous patient geometry, and the angles used for this treatment plan were relatively slight compared to those used in our training and test datasets (all angles used in this plan satisfied $\phi_X \in [-0.3°, 2.1°]$ and $\phi_Z \in [0.4°, 2.8°]$). Still, neglecting the ray angle of entry causes the full plan dose distribution to be at least three times more inaccurate than DoTA-A and DoTA-S predictions. While the maximum error value is only twice as high for the perpendicular MC prediction as for DoTA-A and DoTA-S, the area with high dose deposition shows a consistent error. For DoTA-A and DoTA-S, the incorrect voxels seem to be more evenly distributed with dose values both higher and lower than desired. This difference is also illustrated by the deviation in error and the gamma pass rates.

6 Discussion

Here, we interpret the results from the previous section. We examine the performance of the models constructed in this project, considering both prediction speed and accuracy, and if these or similar models have potential in clinical practice. We also take a critical look of the methods used in this project, suggesting approaches that could lead to better results and ways to bypass the shortcomings of DoTA-A and DoTA-S.

6.1 Prediction speed

The average runtimes of DoTA-A and DoTA-S when predicting a single dose are 13ms and 18ms respectively. This is around three times slower than the original DoTA, but much faster than the PBA and MC dose calculation algorithms which are currently used in clinical practice. Our models are able to predict the dose distributed by 56 and 77 rays per second. This gives them potential in clinical use, drastically reducing the time required in general treatment planning and possibly allowing for the implementation of techniques such as real-time adaptive treatment planning.

As discussed in 5.2, the computation speed of our models seems to be mostly proportional to the model sizes, which in turn are determined by the choice of hyperparameters K and N_h. While our used values of $N_h = 32$, $K = 16$ and $K = 24$ for DoTA-A and DoTA-S respectively gave the highest performance on our validation set, the difference between validation losses for different combinations of hyperparameters was often marginal. Therefore, it is likely that there exists a trade-off in the choice of hyperparameters, where smaller models have higher prediction speed but sacrifice accuracy in the process. Finding the optimal choice of model architecture with both factors in mind could be explored further, perhaps increasing the amount of training data to compensate for the decrease in model size. In this way, DoTA-A and DoTA-S could possibly achieve the same millisecond prediction speed as DoTA without sacrificing prediction accuracy.

In practical applications, DoTA-S requires additional computation time since for each ray, the shape volumes need to be constructed from the angle and energy information as a prerequisite input variable. This roughly doubles the computation speed of the model, which would outweigh its marginal and disputable increase in accuracy over DoTA-A. When implementing such a model in practice, one solution could be to construct a lookup table of shape volumes for all angle and energy values in the available ranges. Since the shape volumes do not depend on patient geometry, they could be generated a priori and stored for later use, which would virtually remove this part of the DoTA-S computation speed limitations.

6.2 Accuracy

From the evaluations across three different test sets, we see that the models DoTA-A and DoTA-S approximate Monte Carlo ground-truth dose distributions with an average relative error of around 1% and average gamma pass rates of 97%. This is a significant improvement over the original DoTA model; recall that DoTA approximates a perpendicular MC baseline, which for the same test sets achieved an average relative error of around 3% and gamma pass rates around 78%. These errors indicate that dose predictions based on perpendicular rays are unsuited for clinical use, and that the inclusion of ray angle as an input for dose calculation algorithms is an important factor in achieving high accuracy.

Our comparisons show that the PBA model we used performs a lot poorer than the other models. However, this PBA model is not the same as those that are used in clinical treatment planning systems. Clinical PBA models typically achieve higher accuracy by modifying the basic PBA algorithm, for example to account for particle scattering behavior. The PBA model we used is part of an open-source toolbox (Cisternas et al. 2015). However, maintaining high accuracy in inhomogeneous geometries such as the lungs remains a challenge even for clinical PBA models. The performance of these clinical PBA models compared to deep learning based models such as DoTA-A and DoTA-S could be investigated in further research.

Upon reviewing the evaluation results for the full treatment plan, a definitive answer to the question which of DoTA-A and DoTA-S provides a higher accuracy is hard to give. DoTA-S performed better on the single ray test sets, but the full treatment plan results illustrate that prediction accuracy is not consistent across different settings, and show why manually reviewing the final dose distribution is important in practical applications. Since DoTA-S has more than twice as many learned weights as DoTA-A, it is possible that the former would benefit more from an increase in training data. Transformer based models are known to be extremely data hungry, and it is likely that both models could achieve much higher accuracy when exposed to more training data.

The accuracy and speed of both models gives them potential in clinical practice, but they are unlikely to be implemented in their current state since a maximum dose error of 3% is a clinical requirement which our models do not fulfill in some settings. The performance of the models is also worse than that of DoTA relative to its own baseline. This is not completely unexpected however, since the input-output relation is more complex with the inclusion of angle-dependency compared to the original DoTA.

6.3 Future research

There are multiple ways in which future research could improve on or implement the models DoTA-A and DoTA-S. Experimenting with the training data, model architectures and training procedure could provide better performing models. In general, experimenting with different training dataset showed a correlation between larger training datasets and better performance. The shape volume construction is an idea that could be explored further, perhaps with applications in different settings or stages of the radiotherapy workflow. Model architecture could also be designed around the shape volume in a similar fashion to the residual connections included in DoTA-S.

Different ways of implementing angle dependency into dose calculation algorithms could also be a further area of research. An example is taking predictions which don't take the ray angle into account at all, such as perpendicular MC, as an initial input similar to (Javaid et al. 2021). Rotating these initial doses or other distortions based on the angle of entry could also approximate realistic dose distributions, although the patient geometry in the new trajectory would have to be taken into account somehow. Modeling the angle of entry by rotating the patient CT and computing a perpendicular MC dose prediction could be investigated, although the dose deposition around the first CT layer would have to be corrected.

Another interesting continuation could be to investigate the performance of DoTA-A and DoTA-S when trained exclusively using samples from a single treatment site. The original DoTA aimed to capture general particle transport, which is a challenging task,

and the addition of angle dependency made the relation our models try to capture even more complex. Limiting the kind of geometries for which the models should be able to predict dose distributions could improve the accuracy of the models, and although the general application of modeling particle transport would be lost, clinical applications for treatment planning on specific treatment sites could become possible.

References

Abadi, Martın et al. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467*.

Breedveld, Sebastiaan et al. (2019). "Multi-criteria optimization and decision-making in radiotherapy". In: *European Journal of Operational Research* 277.1, pp. 1–19.

Carolan, GM (2010). "Pencil beam dose calculation algorithm". In: *Wollongong, NSW: Illawarra Cancer Care Centre*.

Chen et al. (2019). "A feasibility study on an automated method to generate patient-specific dose distributions for radiotherapy using deep learning". In: *Medical physics* 46.1, pp. 56–64.

Cisternas, Eduardo et al. (2015). "matRad-a multi-modality open source 3D treatment planning toolkit". In: *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada*. Springer, pp. 1608–1611.

Feng, Mary et al. (Apr. 2018). "Machine Learning in Radiation Oncology: Opportunities, Requirements, and Needs". In: *Frontiers in Oncology* 8. DOI: 10.3389/fonc.2018.00110.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, pp. 315–323.

Goodfellow, Ian J., Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International conference on machine learning*. PMLR, pp. 448–456.

Jagt, Thyrza et al. (2018). "An automated planning strategy for near real-time adaptive proton therapy in prostate cancer". In: *Physics in Medicine & Biology* 63.13, p. 135017.

Jarrett, Kevin et al. (2009). "What is the best multi-stage architecture for object recognition?" In: *2009 IEEE 12th international conference on computer vision*. IEEE, pp. 2146–2153.

Javaid, Umair et al. (2021). "Denoising proton therapy Monte Carlo dose distributions in multiple tumor sites: A comparative neural networks architecture study". In: *Physica Medica* 89, pp. 93–103.

Kandula, Shravan et al. (2013). "Spot-scanning beam proton therapy vs intensity-modulated radiation therapy for ipsilateral head and neck malignancies: a treatment planning comparison". In: *Medical Dosimetry* 38.4, pp. 390–394.

Kim, Dong Wook et al. (Sept. 2020). "History of the Photon Beam Dose Calculation Algorithm in Radiation Treatment Planning System". In: *Progress in Medical Physics* 31, pp. 54–62. DOI: 10.14316/pmp.2020.31.3.54.

Kontaxis, C et al. (2017). "Towards fast online intrafraction replanning for free-breathing stereotactic body radiation therapy with the MR-linac". In: *Physics in Medicine & Biology* 62.18, p. 7233.

Low, Daniel A et al. (1998). "A technique for the quantitative evaluation of dose distributions". In: *Medical physics* 25.5, pp. 656–661.

Ma, Xun et al. (2019). "Initial Margin Simulation with Deep Learning". In: *Available at SSRN 3357626*.

Meyer, Philippe et al. (2018). "Survey on deep learning for radiotherapy". In: *Computers in biology and medicine* 98, pp. 126–146.

Nguyen, Dan et al. (2019). "A feasibility study for predicting optimal radiation therapy dose distributions of prostate cancer patients from patient anatomy using deep learning". In: *Scientific reports* 9.1, pp. 1–10.

Nomura, Yusuke et al. (2020). "Fast spot-scanning proton dose calculation method with uncertainty quantification using a three-dimensional convolutional neural network". In: *Physics in Medicine & Biology* 65.21, p. 215007.

Pastor-Serrano, Oscar and Zoltán Perkó (Apr. 2022). "Millisecond speed deep learning based proton dose calculation with Monte Carlo accuracy". In: *Physics in Medicine & Biology* 67. DOI: 10.1088/1361-6560/ac692e.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer, pp. 234–241.

Shalek, Robert J (1977). "Determination of absorbed dose in a patient irradiated by beams of X or gamma rays in radiotherapy procedures". In: *Medical Physics* 4.5, p. 461.

Sheehan Schlesinger, Yen (Mar. 2015). "The Radiobiology and Physics of Radiosugery". In.

Souris, Kevin, John Lee, and Edmond Sterpin (Apr. 2016). "Fast multipurpose Monte Carlo simulation for proton therapy using multi- and many-core CPU architectures". In: *Medical Physics* 43, pp. 1700–1712. DOI: 10.1118/1.4943377.

Taheri-Kadkhoda, Zahra et al. (2008). "Intensity-modulated radiotherapy of nasopharyngeal carcinoma: a comparative treatment planning study of photons and protons". In: *Radiation Oncology* 3.1, pp. 1–15.

Taylor, Paige A, Stephen F Kry, and David S Followill (2017). "Pencil beam algorithms are unsuitable for proton dose calculations in lung". In: *International Journal of Radiation Oncology* Biology* Physics* 99.3, pp. 750–756.

Teoh, Suliana et al. (Nov. 2019). "Is an analytical dose engine sufficient for intensity modulate proton therapy in lung cancer?" In: *British Journal of Radiology*. DOI: 10.1259/bjr.20190583.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Wan Chan Tseung, H, Jiasen Ma, and Chris Beltran (2015). "A fast GPU-based Monte Carlo simulation of proton transport with detailed modeling of nonelastic interactions". In: *Medical physics* 42.6Part1, pp. 2967–2978.

Wu and Kaiming He (2018). "Group normalization". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19.

Wu, Dan Nguyen, et al. (2021). "Improving proton dose calculation accuracy by using deep learning". In: *Machine Learning: Science and Technology* 2.1, p. 015017.

You, Yang et al. (2019). "Large batch optimization for deep learning: Training bert in 76 minutes". In: *arXiv preprint arXiv:1904.00962*.