UCLouvain

École polytechnique de Louvain

Deep learning to predict the optimal beam intensities for proton therapy treatments

Author: **Valentine Dormal**
Supervisors: **John Lee, Ana María Barragán Montero**
Readers: **Edmond Sterpin, Benjamin Roberfroid, Camille Draguet**
Academic year 2022–2023
Master [120] in Computer Science and Engineering

Abstract

Radiotherapy is one of the main treatments for cancer. When compared to conventional radiotherapy with photons, proton therapy holds the potential to spare more healthy tissues while ensuring tumor coverage.

In both modalities, a treatment plan needs to be generated. This task is time-consuming, and significant efforts have been invested in automating it. In particular, deep learning has already been employed to automate certain steps in this process. However, while deep neural networks can be used to predict the full dose distribution, their application has not yet been extended to the subsequent step in the planning process of proton therapy plans. This step involves determining the machine parameters, specifically the weights of the spots, that will most accurately achieve the predicted dose distribution.

This thesis aims to explore potential solutions to fill this gap. A fully automated pipeline containing two deep neural networks was implemented. A first network predicts the dose of each individual beam. The second network should take this information together with other inputs to provide a prediction of the spot weights. Both networks employed a U-Net architecture and were trained on a database containing plans with the same beam configuration and prescription for 39 patients with esophageal cancer.

If the obtained results were not satisfactory for direct use without further manual optimization, this work still represented the initial step toward achieving a fully automated workflow.

Acknowledgements

First of all, I would like to sincerely thank Ana and Camille for their time, support, and precious insights, as well as for answering all my questions. Their advice and guidance were very helpful throughout this thesis.

Then, I would like to thank my promoter, John Lee, for giving me the opportunity to work on this very interesting subject.

Finally, I want to thank my family for their support throughout this year.

Acronyms

AI Artificial Intelligence.

ANN Artificial Neural Network.

BEV Beam's Eye View.

CNN Convolutional Neural Network.

CT Computed Tomography.

CTV Clinical Target Volume.

DNN Deep Neural Network.

DVH Dose Volume Histogram.

GAN Generative Adversarial Network.

GPU Graphics Processing Unit.

GTV Gross Target Volume.

IMPT Intensity Modulated Proton Therapy.

IMRT Intensity Modulated Radiation Therapy.

ITV Internal Target Volume.

MAE Mean Absolute Error.

MRI Magnetic Resonance Image.

MSE Mean Square Error.

MU Monitor Unit.

OAR Organ(s) at Risk.

PBS Pencil Beam Scanning.

PTV Planning Target Volume.

ReLU Rectified Linear Unit.

RMSE Root Mean Square Error.

ROI Region of Interest.

SFUD Single-Field Uniform Dose.

SOBP Spread out Bragg peak.

TPS Treatment Planning System.

VMAT Volumetric modulated Arc Therapy.

Contents

Chapter 1

Introduction

In 2020, cancer was responsible for almost 10 million deaths worldwide, and by the year 2040, the number of cancer cases is projected to increase by 49.7% [1]. Among others, radiation therapy is one of the most widely used treatments for cancer [2]. Either alone or in combination with other treatment modalities, such as chemotherapy or surgery, it can be delivered internally or externally.
The goal is to stop the growth and division of cancer cells and, ultimately, destroy them. This is achieved through ionizing radiation, which is able to damage the DNA of the cells. Unfortunately, these damages affect not only malignant cells but also healthy surrounding cells. Hence, one of the key issues in radiotherapy is to spare healthy tissues as much as possible while ensuring that tumor cells still receive a sufficiently high radiation dose.

Several modalities are available for external beam radiation therapy. So far, high-energy X-rays delivering photons are the most common, and this modality is known as conventional radiotherapy. However, the use of charged particles such as protons has shown greater advantages [3][4][5][6][7]. In particular, the way protons deliver energy along their path allows for more targeted treatment and therefore less damage to healthy tissues. Indeed, protons deposit their maximal dose at the end of their trajectory, creating a peak of dose called the Bragg peak (Figure 1.1). In contrast, conventional therapy delivers the maximum dose at the entrance of the body, followed by a decreasing exponential curve along the trajectory. Compared to protons, it not only increases the dose received when it enters the body, but it also increases the dose delivered beyond the tumor. This is why proton therapy has raised great interest within the medical community over the past decade.

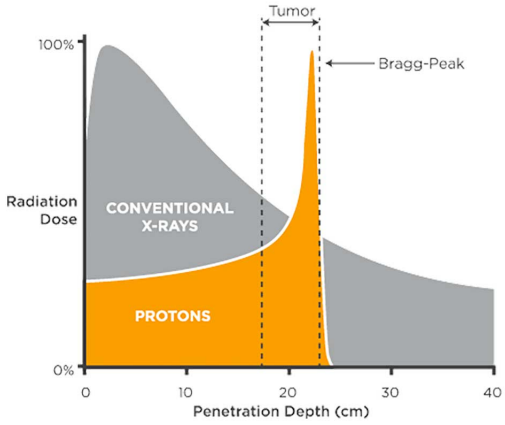


Figure 1.1: Difference in the dose-depth profile between conventional therapy and proton therapy. Credits: [8]

For any of these modalities, a treatment plan has to be established. First, images of the patient are acquired by making a Computed Tomography (CT) scan and/or a Magnetic Resonance Image (MRI). From these scans, the physician then draws the contours of the tumor and of the surrounding organs; this is the segmentation step. After that, the dosimetrist can generate the treatment plan by fine-tuning the beam parameters to obtain the most optimal dose distribution. A trade-off has to be reached between tumor coverage and healthy tissue sparing. This process can require several iterations and is thus time-consuming and complex. Therefore, the quality of the generated plan can be impacted by factors such as the time available or the clinical staff's experience [9][10][11].

It is only after all these steps that the patient can start the treatment. Usually, the patient must go through daily sessions spread over a few weeks. During this time period, the anatomy of the patient can change, making it suboptimal to use the exact same plan each time [12][13][14]. Speeding up the generation of treatment plans is thus much needed.
Thankfully, the development of several technologies has already taken a step forward towards a more automated treatment plan generation. For instance, numerous studies have focused on the use of Artificial Intelligence (AI) to automate the segmentation step [15][16][17][18][19].

This thesis will focus on the automation of another step: the tuning of the machine parameters. In a manual workflow, optimization software is used by the dosimetrist to fine-tune the device parameters. Based on the resulting dose distribution and some corresponding metrics, the plan can be validated by medical specialists. Otherwise, a new iteration with further tuning of the parameters is started.

In an automated workflow, a prediction of the dose distribution is first generated. The device parameters are then optimized to achieve that dose distribution as well as possible. Usually, the automated treatment pipelines proposed so far have only employed deep learning frameworks to predict the dose distribution. The final device parameters are then computed by what is called a dose-mimicking optimization algorithm (Figure 1.2.a). For proton therapy treatment, the integration of deep learning in this last step is yet to be explored, as only a limited amount of work has been conducted in that direction.

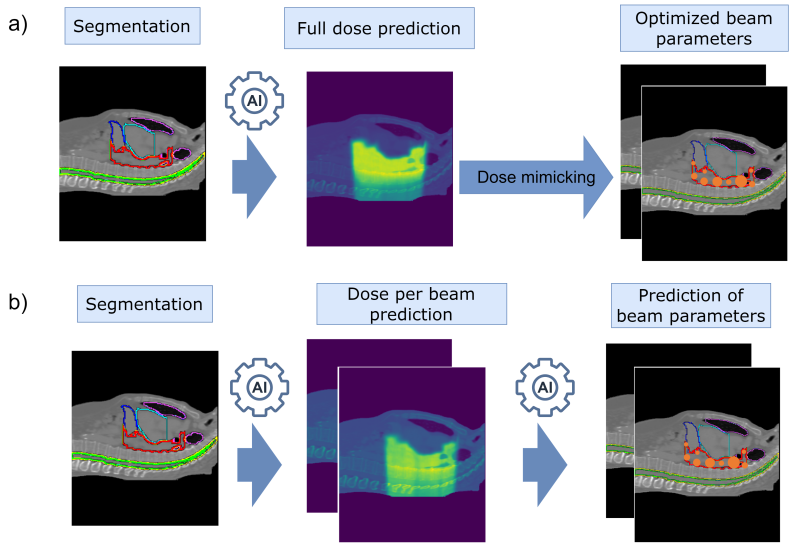


Figure 1.2: Two different workflows for the generation of a treatment plan. The images on the left show a CT scan with the organs and the tumor delineated. The central images represent the dose distributions. The orange dots of various sizes on the right images represent the machine parameters. **a)** Workflow with AI to predict the full dose and dose mimicking to find the beam parameters. **b)** The workflow developed in this thesis with AI to predict the beam dose distributions and the beam parameters.

The objective of this thesis is to develop and train a Deep Neural Network (DNN) capable of predicting the machine parameters based on the pre-generated dose distribution and some other patient-specific anatomical data (Figure 1.2.b). To develop a complete workflow, the network providing the pre-generated dose distributions should also be trained and tested as a secondary objective. With the aim of speeding up the process, the proposed workflow should be optimized as much as possible. Depending on the quality of the results, the parameters found could either be used directly or as a first guess to provide to the software currently employed for the iterative fine-tuning of the parameters. In this last case, it could still allow the overall plan generation to be faster by reducing the number of iterations needed.

This thesis is organized as follows: Chapter 2 provides context for the proton therapy treatment planning process. It also introduces the key concepts of deep learning relevant to understanding the networks used in radiotherapy applications. Chapter 3 includes a literature review, while the model's architecture and methods employed are described in Chapter 4. Chapter 5 presents the obtained results and is followed by a discussion of those results in Chapter 6. Finally, a conclusion in Chapter 7 closes this thesis.

Chapter 2

Context

2.1 Proton therapy

In external beam proton therapy treatments, a proton beam is used to ionize the diseased tissues. To reach the tumor site, protons must be accelerated to increase their energy level. When entering the body, this energy is transferred to the tissues. The resulting dose deposition is quantified in Gray [Gy] units, which correspond to the absorption of one joule of energy in one kilogram of matter.

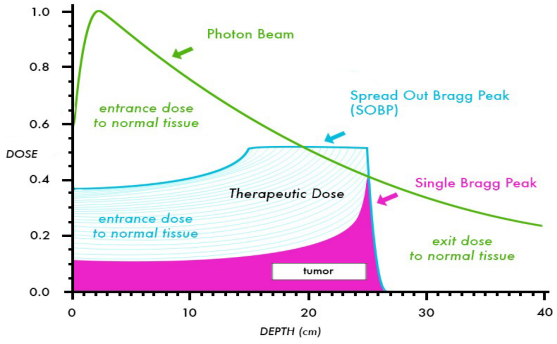


Figure 2.1: The SOBP is formed by several Bragg peaks of different energies to cover the tumor in depth. Credits: [20]

Protons have the advantage of delivering most of the dose at the end of their trajectory, forming the so-called Bragg peak. Compared to photons, only a small dose is deposited before that peak and almost none after it. The depth of the peak can be adjusted by changing the energy of the protons accordingly before they enter the body.

With a single-energy proton beam, the Bragg peak is too narrow to cover the tumor in depth. To achieve sufficient longitudinal coverage, multiple beams with varying energies need to be superimposed to form the Spread out Bragg peak (SOBP), as shown in Figure 2.1.

To reach the required energy levels, protons must first be accelerated into a cyclotron or synchrotron. With a synchrotron, the protons are accelerated to the targeted energy. On the other hand, protons in a cyclotron are accelerated to a fixed maximum energy. Energy degraders are then inserted along the path of the protons to lower their energy as needed.

To cover the three-dimensional tumor, the resulting narrow mono-energetic beam must be spread not only longitudinally by forming the SOBP but also laterally. To achieve lateral coverage, there are two main delivery techniques (Figure 2.2). The first one is the passive scattering technique, which involves the use of heavy materials to scatter the beam but also requires patient-specific compensators and collimators. The second technique is Pencil Beam Scanning (PBS), which is much more flexible and consequently represents the state-of-the-art delivery technique in proton therapy [12][21].

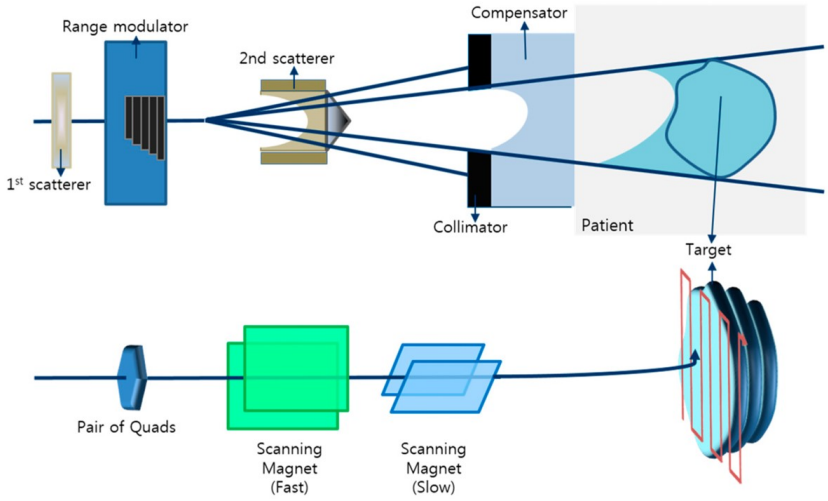


Figure 2.2: Two proton delivery techniques. **Upper image**: the passive scattering technique. **Lower image**: the Pencil Beam Scanning (PBS) technique. Credits: [22]

2.1.1 Pencil beam scanning

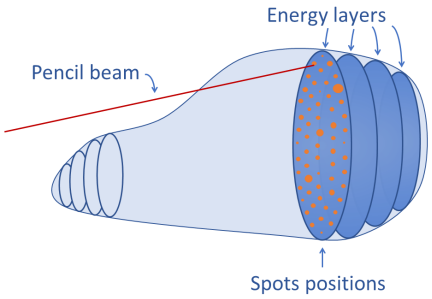


Figure 2.3: Illustration of PBS with the scanning of energy layers and spots with different weights shown as circles with various sizes

The main idea behind Pencil Beam Scanning is to employ scanning magnets to move the small pencil beam laterally. The tumor is first sliced into different energy layers that are perpendicular to the beam's direction. Those layers are scanned one at a time, from the deepest and most distant one (maximum energy) to the closest one (minimum energy). They thus form the SOBP and allow the tumor to be covered depth-wise. In each layer, dose delivery positions named spots are placed in order to cover the tumor in the lateral direction (Figure 2.3). One layer at a time, the beam will thus be positioned on each of these spots by adjusting the magnetic fields. To determine the dose delivered at each position, an intensity or weight is associated with each spot. This weight is proportional to the number of protons delivered at that specific location and is expressed in Monitor Unit (MU).

PBS can also be used with several beam angles in what is known as Intensity Modulated Proton Therapy (IMPT). For each beam angle, the same process is used as described earlier: the tumor is divided into energy layers perpendicular to the beam's direction, and each layer is rasterized as a discrete grid of spots. This way, the dose distribution is achieved by combining the Bragg peaks of protons coming from different angles. This has the advantage of further enhancing dose and shape conformity as well as healthy tissue sparing.

2.1.2 Treatment planning process

Before delivering the proton therapy treatment, a treatment plan specific to the patient must be generated by establishing several parameters. Those parameters include the number and configuration of beams, along with the position and weight of each spot, all of which determine the resulting dose distribution. The different steps to generate a conventional proton treatment plan are described in this section.

Computed tomography scan

First, medical images of the patient must be acquired in order to have precise information about the relative placement and shape of the different organs and the tumor. For this purpose, a Computed Tomography scan is made.

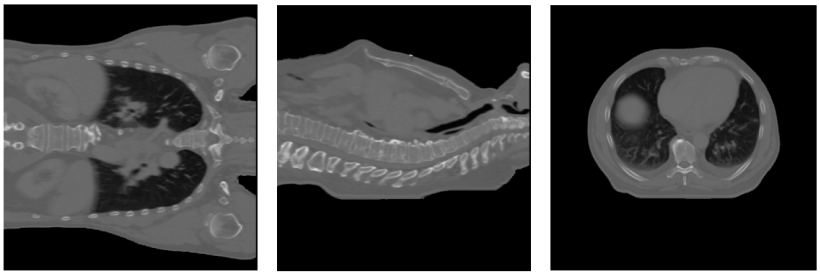


Figure 2.4: Different slices of a CT scan of a patient. **From left to right**: coronal, sagittal, and axial sections.

A CT scanner consists of an X-ray source that sends photons through the patient's body. On the other side of the source, detectors measure the attenuation of the X-ray. The X-ray source is able to rotate around the patient's body, and thus the 3D image is reconstructed from the measurements taken from many different angles (Figure 2.5).

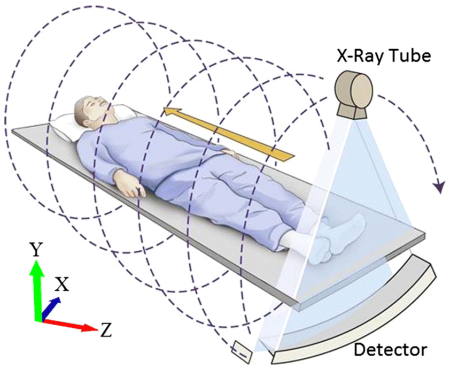


Figure 2.5: A CT scanner with its X-ray source and its detector on the opposite side. Credits: [23]

Depending on the type of tissue it passes through, the X-rays are more or less attenuated. The unit used to describe the radiation attenuation coefficient of each voxel of the CT scan is the Hounsfield unit [HU]. The attenuation coefficients are usually represented as gray values. For instance, dense tissues like bones lead to a higher attenuation coefficient, resulting in a brighter area in the CT scan (Figure 2.4) [24].

Tumor and organ contouring

With the CT scan acquired, the tumor and the Organ(s) at Risk (OAR) must be delineated. There are several contours that can be drawn around the tumor to define the target volume (Figure 2.6)[25]:

- **GTV:** The Gross Target Volume (GTV) is the tumor that is visible on the medical images.
- **CTV:** The Clinical Target Volume (CTV) encloses the GTV plus an additional margin to account for the spread of the disease, which leads to non-visible microscopic extensions of cancerous cells.
- **ITV:** The Internal Target Volume (ITV) includes the CTV and an additional internal margin that accounts for variations due to organ motions such as breathing.
- **PTV:** The Planning Target Volume (PTV) encompasses the ITV to which a margin is added to account for errors due to positioning, machine tolerances, and set-up uncertainties. The goal is to make sure that the CTV indeed receives the planned dose.

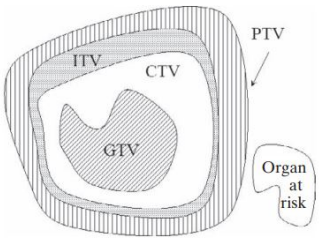


Figure 2.6: Different contours around the tumor. Credits: [25]

The different contours for the target volume and the OARs can be drawn either manually or with the help of various tools, including artificial intelligence technologies [15][16][17][18][19].

Dose prescription

A dose prescription must be set to the target volume. This value must take into consideration not only the tumor treated but also the surrounding organs, which may withstand different maximum levels of dose. Sometimes, higher boost doses can be prescribed for specific areas of the tumor. In the esophagus case studied in this thesis, the same dose prescription of 50.40 Gy is used across all patients.

Plan generation

With all this information, the generation of the plan can start. In what is known as inverse planning, the device parameters are determined by solving an optimization problem. To define this problem, the dosimetrist establishes a set of minimal and maximal dose constraints on the OARs and on the target volume. The objective function of the optimization problem is defined as a weighted sum of these different criteria.

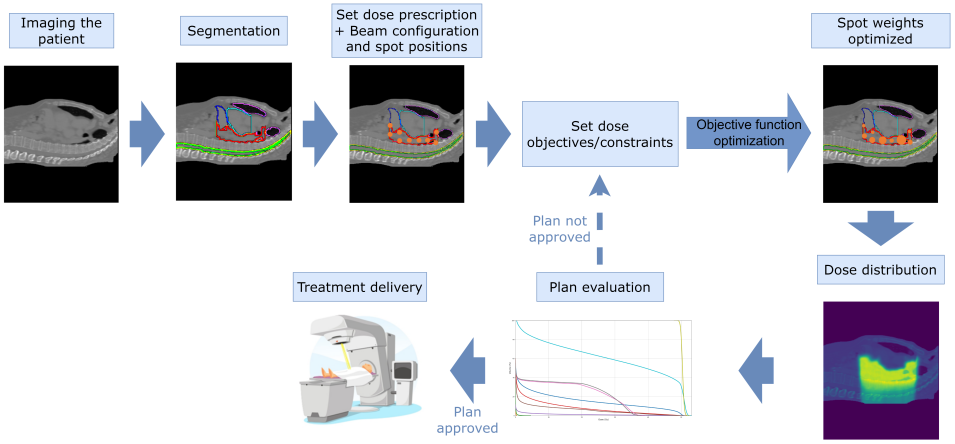


Figure 2.7: The workflow to generate a proton treatment plan with inverse planning.

A Treatment Planning System (TPS) such as RayStation (RaySearch Laboratories, Sweden) is used not only for the iterative optimization process but also for dose calculation and plan evaluation. Based on the patient's anatomy, the software fine-tunes the machine parameters to meet the constraints as well as possible. Once an optimum is reached, the dosimetrist can evaluate the obtained dose distribution. The goal is to achieve a dose distribution that provides sufficient dose coverage for the tumor while minimizing radiation exposure to the surrounding organs. If it is not satisfactory, the weights assigned to the criteria in the objective function must be adjusted in an additional iteration. This iterative process continues until an admissible treatment plan is obtained.

Inverse planning can be a time-consuming task since the TPS needs guidance before finding the optimal plan. This is why some new automated workflows using deep learning have been proposed to speed up the process (see Section 2.3).

Plan evaluation

To evaluate a plan, several metrics can be helpful. One tool often employed is a Dose Volume Histogram (DVH). This is a graphical representation of the

radiation dose delivered to the target volumes or organs at risk. To construct a cumulative DVH, a curve is associated with each Region of Interest (ROI). The X-axis represents the volume percentage, and the y-axis corresponds to the dose [Gy] (Figure 2.8).

In a good-quality plan, the DVH curve associated with the target volume should be constant at 100% of the volume before dropping steeply to 0% at the prescribed dose. Conversely, the curve associated with each OAR should always be as close to zero as possible.

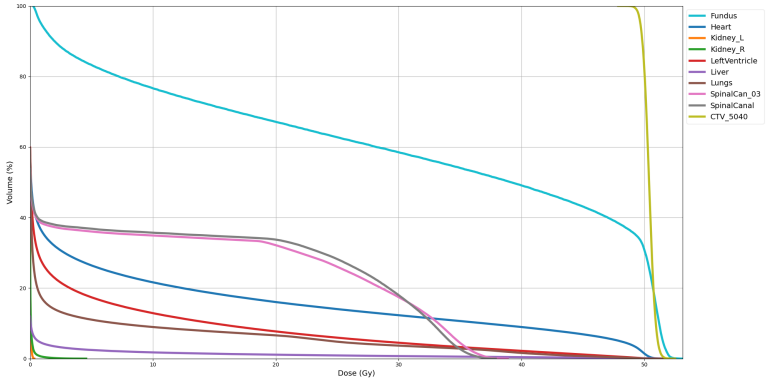


Figure 2.8: An example of a Dose Volume Histogram (DVH). The target curve is in yellow and aims for a 50.4 Gy dose. The other lines correspond to different organs.

From this graph, several metrics can be extracted. D_x is the maximal dose delivered to at least x percent of the volume of interest. Thus, D_5 is the maximum dose received by 5% of the volume. Depending on the ROI considered, commonly employed metrics are D_2, D_5, D_{95}, D_{98} and D_{mean} which is the average dose delivered to the volume of interest. V_x can also be extracted from the graph: this is the highest percentage of the volume of interest receiving a radiation dose of at least x Gy. All these metrics can be used to assess the quality of a plan and compare different plans.

Plan robustness

There are many sources of uncertainty in the treatment planning process. They can come from anatomical modifications over time, patient setup on the scan, tumor contouring, dose calculation approximations, organs, or tumor motion due to breathing for instance [26]. In conventional radiotherapy, some of these errors are accounted for by adding a margin to the CTV and thus performing a PTV-based optimization. But in proton therapy, the finite range of the protons is a double-edged sword. As it allows a more targeted treatment and therefore more healthy tissue sparing, we can no longer make the assumption that the

dose distribution is invariant under spatial shifts. Hence, for proton therapy, the CTV is used instead of the PTV, and uncertainties are addressed through robust optimization. To obtain a robust plan, various scenarios are considered, and the optimized plan should meet the predefined requirements even in the worst-case scenario. These scenarios include CT scans with a shift or a change of density for instance [27][28].

2.2 Deep learning

In the past few years, deep learning has experienced a surge of interest and a fast evolution. It was enabled by several factors, such as the availability of large datasets, the development of new deep learning techniques and architectures, the advancement in computing power, and especially in Graphics Processing Unit (GPU)s [29][30]. The range of applications in which it has already shown remarkable results is wide. From computer vision with face recognition, object detection, human pose estimation, and action recognition [31], to speech recognition [32], but also natural language processing with information retrieval and extraction, text classification and generation, and machine translation [33].

As the topic is broad, this section will only introduce some of the main concepts necessary to understand a deep learning model used in the context of radiotherapy.

2.2.1 Neural Networks

At the origin of deep learning is the Artificial Neural Network (ANN), which takes its inspiration from the structure of the human brain and the way it transmits information. It is composed of several interconnected units, which represent neurons. A neural network is composed of multiple layers of these units. The first layer is called the input layer and receives the data. The last layer is the output layer, and the layers in between are the hidden layers. A Deep Neural Network (DNN) is a network with several hidden layers.

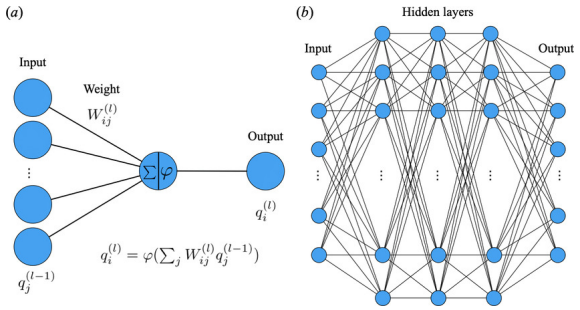


Figure 2.9: **a)** How the value associated with one unit of a deep neural network is computed. **b)** The different layers of a deep neural network. Credits: [34]

Like synapses in the brain, the units from one layer transmit information to the units of the next layer through links. A weight is associated with each link such that each unit computes the weighted sum of the input signals coming from the previous layer (Figure 2.9). A constant value called a bias can be added to that sum before applying an activation function. The output of that function is then transmitted to the units of the next layer. Based on the information processed in the hidden layers, the output layer is able to generate one or several final values, depending on the type of task. In classification tasks, the output is the predicted label, and in regression tasks, the output is the predicted quantity. Training a deep neural network consists in fine-tuning the weights and biases of the network in order to identify relevant features in the data, thereby minimizing the error in the provided output.

Back-propagation

In order for the network to learn, the loss function that needs to be minimized must be defined. This function measures the difference between the predicted output and the true goal output. Then, during the step known as back-propagation, the measured error is propagated back through the network to update the different weights and biases. This is done by computing the gradients of the loss function with respect to the weights and biases of each unit. These gradients are back-propagated to update the weights and biases of each layer in the network. Therefore, training a deep neural network is an iterative process of feeding the network with input data, comparing the obtained results with the expected ones, and then propagating the error backwards in the network to update its parameters [30][35].

Loss function

As mentioned earlier, a loss function, also called a cost function, must be defined according to the task at hand. For regression tasks, two loss functions are commonly employed [36].

The first one is the Mean Square Error (MSE). Since the difference between the true value y and the predicted value $\hat{y}$ is squared, this loss function is sensitive to outliers.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.1}$$

The second one is the Mean Absolute Error (MAE). This function is more robust to outliers than MSE, but its gradient is constant, which can impact convergence during the learning process.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{2.2}$$

Training, test, and validation set

To assess the performance of the trained model, a subset of the data called a test set must be held out separately from the data that is fed to the network during training. After the training is completed, the test set is used to evaluate the generalization error of the network.

Most models contain hyperparameters, which are settings that are not learned by the algorithm but that influence the model's behavior. When choosing values for the hyperparameters, the test set should not be used to make these decisions in order to have an unbiased evaluation of the final model's performance. This is why the remaining data is further split into two disjoint subsets: the training set and the validation set. The first larger subset is used to train the parameters. Then the validation set is used to assess the generalization error and update the hyperparameters accordingly. When all hyperparameters are set, the final model can be evaluated on the samples of the test set.

K-fold cross-validation

When working with a small dataset, the small size of the test set introduces more uncertainty in the estimated average test error. To remedy that, k-fold cross-validation is often performed. It consists in partitioning the dataset into k folds of similar sizes. The training process is repeated k times, each time taking different folds for testing and validating. The test error is estimated by averaging the k test errors.

Epochs and batches

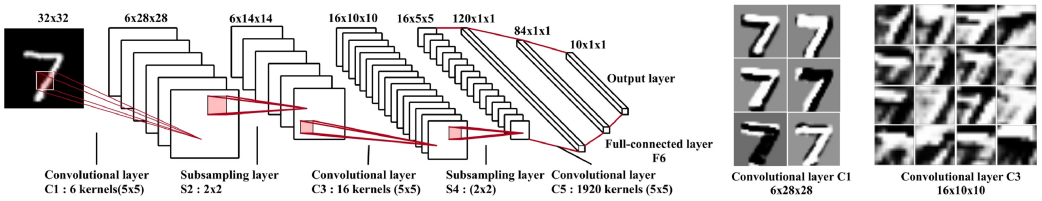
During training, the number of passes over the entire training set is known as the number of epochs. With too few epochs, the network is not able to learn, but too many epochs may lead to overfitting.
During each epoch, all the data from the training set is not usually processed all at once. It is rather divided into smaller subsets known as batches. The size of a batch is often limited by the amount of memory available. However, since the parameters are updated after processing all samples in a single batch, the batch size influences the number of parameter updates. A larger batch size can have benefits, such as decreasing the training time for equivalent accuracies [37].

2.2.2 Convolutional neural network

A Convolutional Neural Network (CNN) is a type of Deep Neural Network that is well-suited to tasks involving image processing. Unlike in conventional fully connected neural networks, the units of a CNN's layer are not connected to all the units of the next layer. Sparse connectivity reduces the number of weights to be trained, making it more efficient. Furthermore, since each unit is connected to only a local region of the input, the network is able to focus on helpful local

patterns in the data. These are some of the reasons why CNNs have shown great results for processing images as well as videos and speech [38][39][40].

A CNN is composed of several layers that perform various operations to extract relevant information from the input images. Some of the main types of operations are described hereafter. Different CNN architectures can be built by assembling all these layers of operations. An example of a classification task is shown in Figure 2.10.



(a) LeNet-5 network (b) Learned features

Figure 2.10: **a)** Architecture of a Convolutional Neural Network (CNN) network applied to a digit classification task. **b)** Some of the features learned by the network at different layers. Credits: [41]

- **Convolutional layer**

  The most important idea behind CNNs is to use the convolution operation to extract features from the input image in the form of feature maps. A convolution is the process of applying a filter called a kernel to the input image. The kernel is a matrix whose entries can be initially set at random and then fine-tuned to detect the relevant features in the data. A feature map is built by applying the filter to different parts of the input image. By moving the filter across the image, each value of the feature map is obtained. The shift applied to the kernel to compute each value is called the stride. Additionally, some padding can be applied to the input image in order to preserve its size. The size of the filter, the padding, and the value of the stride will thus determine the dimensions of the obtained feature map. For instance, Figure 2.11 shows the application of a 3x3 2D kernel with a padding of one and a stride of one. The convolution operation can be adapted to other dimensions by employing 3D kernels for 3D images, for instance.

  A convolutional layer applies a set of filters to its inputs to extract several helpful features. The first convolutional layers detect low-level features like edges, corners, or gradient orientations, while the deeper layers are able to extract high-level features that are more global and complex.

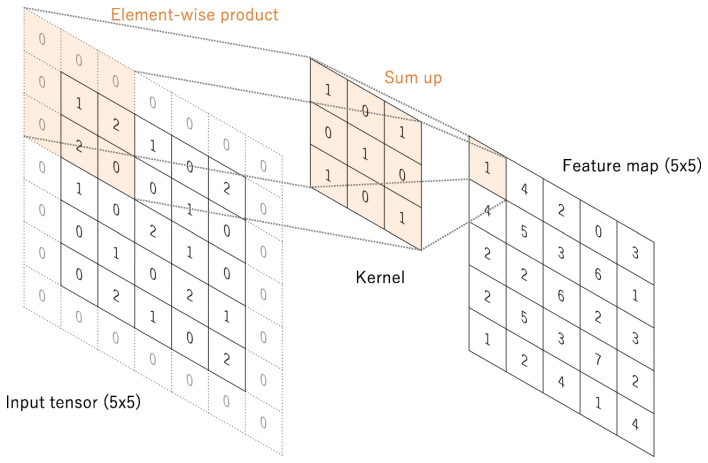


Figure 2.11: An example of a convolutional operation with a 3x3 filter, a padding of one, and a stride of one. Credits: [42]

- **Activation layer**

  An activation layer usually follows the convolutional layer. It applies a non-linear function to the output of the previous layer. By introducing non-linearity into the network, more complex structures in the data can be learned. The most commonly used activation functions in CNNs are the Rectified Linear Unit (ReLU), the sigmoid, and the tanh (Figure 2.12). However, the ReLU activation function has shown several advantages over the tanh and sigmoid functions. Among them, the most important ones are its computational efficiency and its ability to avoid the vanishing gradient problem. The vanishing gradient problem arises when the weights of the network are no longer updated due to vanishingly small values of the loss function's gradient. As activation functions with saturating values (such as tanh and sigmoid) are more prone to this problem, the ReLU function has shown faster training capabilities [43][30].

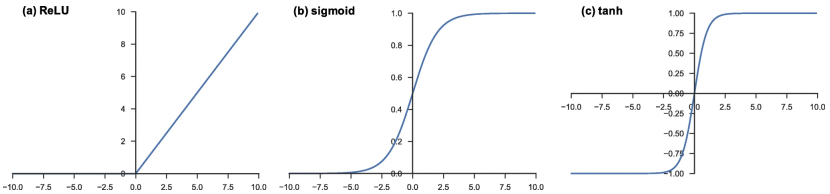


Figure 2.12: Three different activation functions. Credits: [42]

- **Pooling layer**

  A pooling or subsampling layer performs a downsampling operation on the inputs. The advantage of reducing the size of the data is that it decreases the number of parameters and consequently improves the efficiency of the network. The size of the pooling window and the stride used to shift that window across the input data determine the size of the reduction performed. Pooling also provides representations that are approximately invariant to small translations of inputs. The most common pooling operations are max pooling and average pooling. A max pooling operation will keep the highest value over a window of inputs. By only considering the high magnitudes, this pooling operation extracts the most salient feature, but the distinguishing feature may vanish when most values in the pooling region are high. For the average pooling operation, the average of all the values in the window is taken (Figure 2.13). By also taking into account the law magnitudes, the average pooling will have a smoothing effect, resulting in a feature map with decreasing contrast [44].

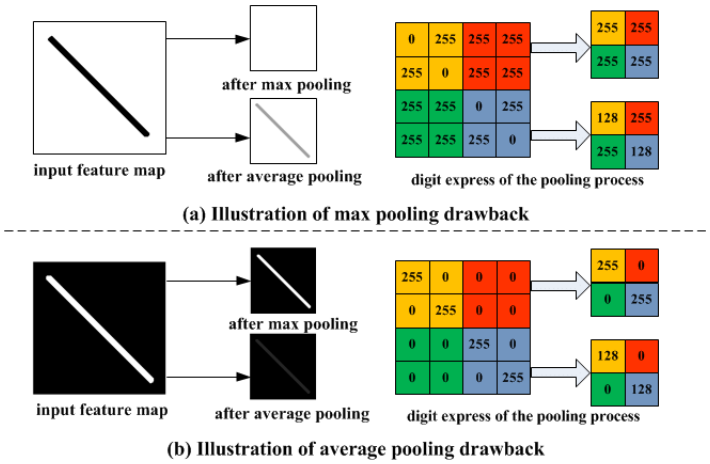


Figure 2.13: The two main pooling operations, with an illustration of their respective drawbacks. Credits: [44]

- **Normalization layer**

  Normalizing activations of a DNN can allow faster training of the network and better performance [45][46]. A normalization layer in a CNN can normalize the input data in different ways :

  – Batch normalization: Inputs of the layer are normalized by re-centering and re-scaling them. For that, the inputs of each channel are subtracted by their mean and then divided by their standard deviation. Means and variances are computed over the whole batch and across all dimensions for each feature map or channel [45][30].

– Instance normalization: It is similar to batch normalization except that instance normalization is performed separately for each sample of the batch instead of computing means and variances for the whole batch [46].

2.2.3 U-Net

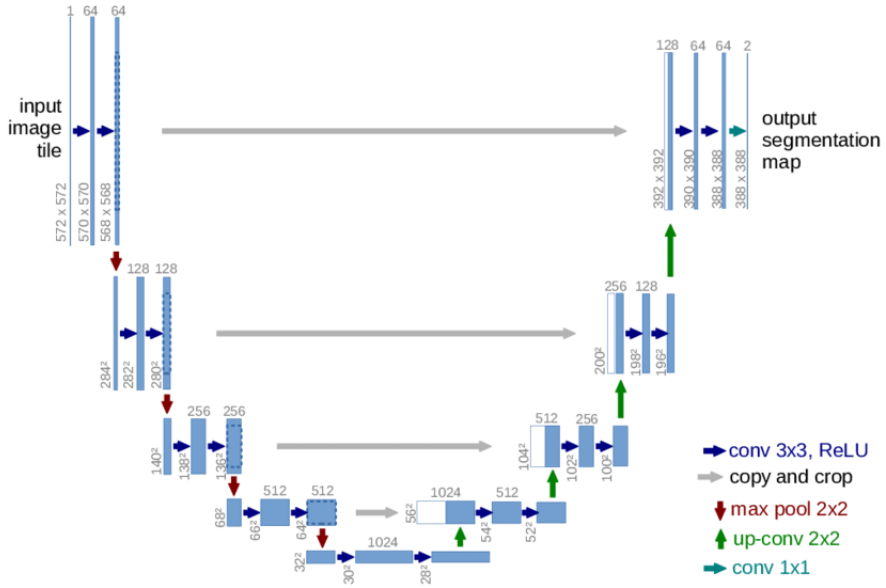


Figure 2.14: The architecture of a two-dimensional U-Net. The blue boxes correspond to newly calculated features, while the white boxes are copied features. Their dimensions are provided on the left side, and the numbers above give the number of feature maps. Credits: [47]

In 2015, a new CNN architecture named U-net was introduced by Olaf Ronnenberger et al. in [47]. It was initially designed for image segmentation tasks in the biomedical field, but it has since been adapted to perform other types of tasks and has also been extended to handle volumetric data [48][49]. This network architecture is characterized by its U-shaped structure, formed by an encoding and a decoding part. The encoder, also known as the contracting path, is the first part of the network. It extracts features from the data and reduces the spatial resolution of the images. This is achieved by repeatedly applying convolutional operations along with downsampling operations such as strided convolution or max pooling. With each downsampling step, the number of feature channels increases.
In the second part of the network, the decoder or expansion path increases the resolution of the high-level feature maps by performing upsampling operations instead of downsampling operations. Hence, the decoder consists of repeated upsampling and convolution operations, where the number of feature channels

is decreased at each level of the U-net. Additionally, the features from the contracting path can be concatenated with the corresponding features from the expansion path at each level. This allows the decoder to precisely localize objects and features by recovering spatial information lost during downsampling. As a result, the final high-resolution output of the network is capable of performing accurate localization and capturing contextual information.

2.3 Automated treatment workflow

Several steps of the treatment generation process can be automated in order to obtain a fully automated pipeline. As already mentioned, the segmentation step, during which the tumor and organs are delineated, can be done through a deep neural network. But the focus of this thesis is on the use of deep learning for the tuning of machine parameters.

2.3.1 Dose prediction and dose mimicking

In the first deep learning-based automated pipelines, deep learning was introduced to predict the optimal dose distribution (see Section 3.1.2). However, the device parameters needed to achieve that predicted distribution are not provided by the model and still need to be determined. Therefore, an additional step is required to provide parameters that achieve the predicted dose distribution as faithfully as possible. This step is called dose mimicking and requires an optimization algorithm. Then, with the beam parameters given by the dose mimicking algorithm, the final dose distribution can be computed (see Figure 2.15.

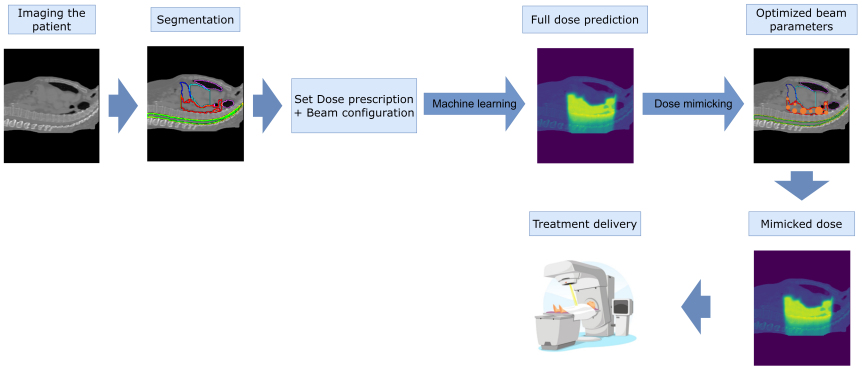


Figure 2.15: The full workflow to generate a proton treatment plan with deep learning to predict the full dose distribution and dose mimicking to find the beam parameters.

2.3.2 Dose and weights prediction

The new automated pipeline proposed in this thesis aims at providing a prediction of beam parameters thanks to deep learning. For that purpose, a prediction of the dose per beam distribution is computed by a first neural network based on the anatomical data of the patient. After that, this dose prediction, along with the anatomical data, is fed to a second deep neural network. This network is designed to predict the beam parameters, namely the weight of each spot (see Figure 2.16).

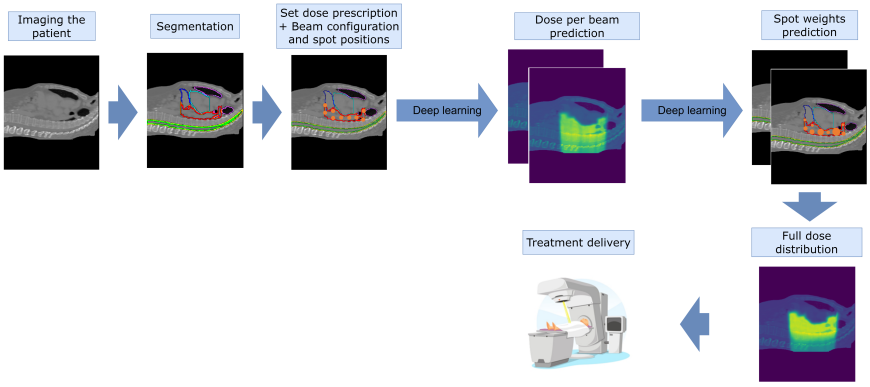


Figure 2.16: The full workflow to generate a proton treatment plan with deep learning to predict the beam dose distributions and the beam parameters. This workflow is the one implemented in this thesis.

The first difference with the pipeline described in the previous section is that the first network predicts an individual dose distribution for each beam rather than directly predicting the full dose.
The second difference is that dose mimicking is no longer required. Instead, the machine parameters will be provided by the second neural network. The spot weights associated with one beam will be predicted by providing the second network with the corresponding dose per beam prediction, along with spot positions and anatomical information. All the required machine parameters should therefore be provided beam by beam by the second deep neural network.

Chapter 3

Literature Review

The use of deep neural networks to automate the treatment planning process has been investigated during the past few years. However, many of those research studies target photon therapy treatments, while applications in proton therapy are scarcely explored. First, Section 3.1 reviews some of the state-of-the-art neural networks employed to predict the full dose distribution in photon and proton therapy. Since no pipeline like the one proposed in this thesis has been investigated for proton therapy, Section 3.2 will present articles presenting somehow equivalent pipelines for photon therapy treatments. Lastly, a study introducing a network designed to fine-tune spot weights for adaptive proton therapy will be presented, along with its limitations, in Section 3.3.

3.1 Full dose prediction

In the past few years, a growing number of articles have been published, exploring deep learning models predicting dose distributions in photon therapy and across various treatment sites [50][51][52]. Among them, most of the network's models fall into the categories of Convolutional Neural Network (CNN) or Generative Adversarial Network (GAN).

3.1.1 ResNet

Residual networks (ResNets) have been introduced by He et al. to tackle the challenge of training very deep neural networks [53]. The main idea is to add skip connections to the network in order to alleviate the vanishing gradient issue. CNNs based on the ResNet have been used to predict dose distributions slice by slice [54][55].

3.1.2 U-net based architectures

The majority of the dose prediction models that have been tested are variations of the U-net architecture (see Section 2.2.3). First modified to generate dose prediction distributions slice by slice [56], the 3D U-Net was then employed to directly handle volumetric data [57][58][59]. Variations of the baseline model include the introduction of a structure-focused loss, for instance [59] or the study of the impact of data normalization and inputs (usually a combination of the CT, the PTV, and the OAR contours)[58].

Even more complex variations of the architecture have been proposed, such as the 3D Dense Dilated U-Net [60], and the 3D U-Res-Net [61], which is based on both a 3D U-net and a residual network. The latter also integrates information about beam configurations into the inputs. Some other convolutional neural networks have taken their inspiration from the encoding-decoding architecture of the U-Net to predict the 3D dose distributions [62][63].

HD U-Net

Nguyen et al. proposed a Hierarchically Densely connected U-Net (HD U-Net) [64] that is based on the architectures of the U-Net and the DenseNet. The DenseNet was introduced by Huang et al. [65] and offers the advantages of mitigating the vanishing gradient problem, reducing the number of parameters, strengthening the propagation of features, and encouraging feature reuse. The main idea of DenseNet is to densely connect each convolutional layer to every other layer in a feed-forward fashion such that the current layer takes as input every preceding layer (Figure 3.1).

By combining the global and local information capabilities of U-Net with the efficient feature propagation and reuse of DenseNet, the HD U-Net architecture integrates the strengths of both models. To have the best of both architectures, dense operations are added to the standard U-Net.
The contracting path is composed of dense convolutions and dense downsampling. The dense convolution is equivalent to the densely connected computations of the DenseNet since new features are concatenated with previous features after the convolution and ReLU operations. The dense downsampling consists of a strided convolution that halves the resolution and is followed by a ReLU activation. This new set of features is concatenated with the old set of features that are downsampled with a max pooling operation.
The expansion path is formed by dense convolutions and classic U-Net upsampling. The U-Net upsampling upsamples the features to double their resolution. After applying a convolutional operation, these new feature maps are concatenated with the feature set at the same level of the encoder part of the "U".
All computations were performed using 3D operations since the network was designed to handle volumetric data.

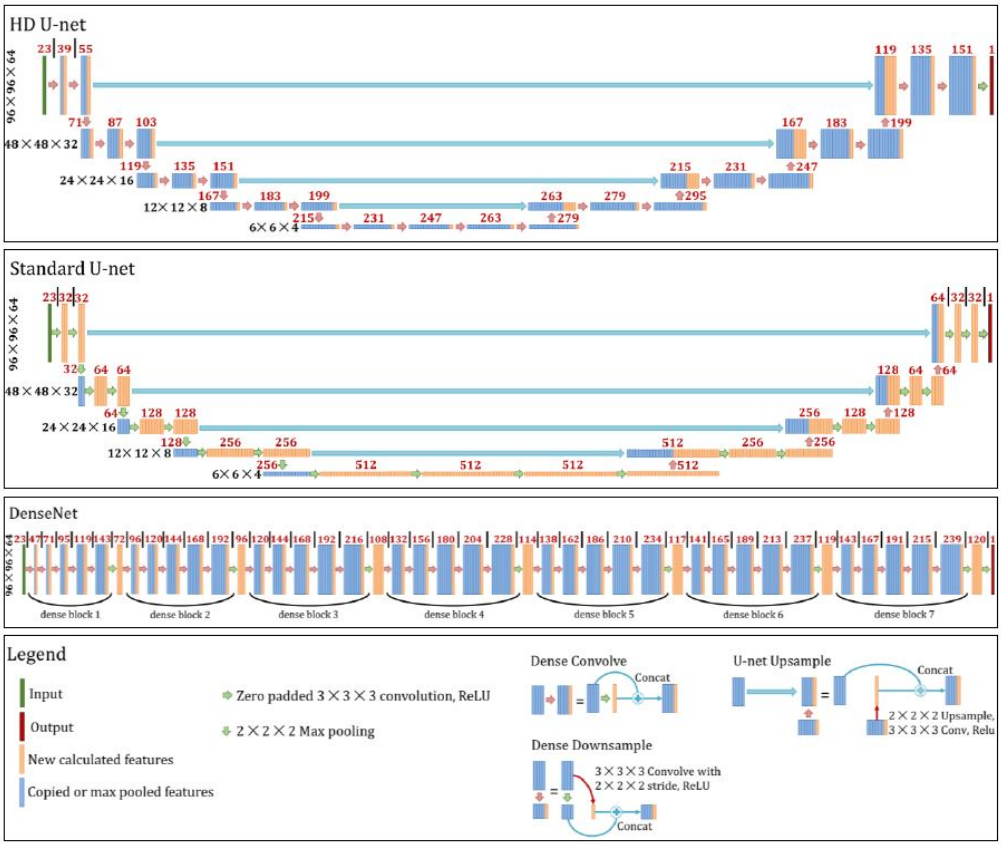


Figure 3.1: Different architectures that have been tested and compared to predict the full dose distributions. The red numbers above each layer give the number of feature maps. Their dimensions are provided by the black numbers on the left side for each hierarchy level. Credits: [64]

The HD U-Net architecture was originally tested for head and neck cancer patients treated with Volumetric modulated Arc Therapy (VMAT) and taking only anatomical information as input [64]. It was compared with the two architectures that inspired it, the U-Net and the DenseNet, and was found to outperform both of them. It was then extended to include beam information in the inputs in order to treat lung cancer with Intensity Modulated Radiation Therapy (IMRT) with heterogeneous beam configurations [66].
IMRT is somewhat analogous to IMPT, but in the context of conventional radiotherapy. With IMRT, multiple beam angles are employed, and the intensity of each beam can be modulated. Instead of having fixed field beam angles, VMAT delivers the dose by continuously rotating the radiation source along an arc trajectory, enabling treatment from a 360°beam angle [67].

Proton therapy

Fewer articles have been published concerning the prediction of dose distributions in proton therapy. In 2021, Guerreiro et al. used two 3D patch-based U-Nets to predict proton and photon dose distributions with the purpose of providing a guidance tool for modality selection [68]. The dataset included patients treated for pediatric abdominal tumors with heterogeneous prescribed doses. Planning was done for both VMAT and PBS. Even if the number and orientation of beams in PBS were patient-specific, the inputs did not include beam arrangement information as they only included CT and OARs together with ITV and vertebra mask contours set to the prescribed dose.

Eriksson et al. proposed a pipeline for robust automated treatment planning [69]. A first 3D U-Net predicted the nominal dose distributions. Then, the nominal dose is deformed into scenario doses that correspond to a given set of scenarios. These scenario doses are predicted thanks to a second 3D U-Net. This is followed by robust dose mimicking optimization in order to provide robust plans for patients treated with intensity-modulated proton therapy.

In [70], a U-net network has been trained to predict dose distributions for patients with oropharyngeal cancer. The impact on plan robustness of different dose mimicking alogrithms and of post-processing was assessed.
A HD U-Net is currently used and maintained at the MIRO lab. This architecture has been employed to predict dose distributions and generate both photon and proton plans for patients presenting with oropharyngeal cancer [71]. But other than the CT, the target volume with the dose prescription, and binary masks for the OARs, an additional input was provided to the network: prior knowledge about the dose distribution that was obtained by running a first automatic optimization in the TPS without any manual tuning.
In order to develop an automated decision support tool for modality selection in the case of esophageal cancer, two HD U-Nets were also trained to predict dose distributions for IMRT and PBS treatments [72]. Similarly, the CT, the body contour, the OARs, and the target volume are processed by the network. Prior knowledge about the dose distribution was also demonstrated to provide more accurate results.
The work done in this thesis was based on this existing network architecture.

3.1.3 GAN

GANs are well-known deep learning algorithms employed to generate artificial data by capturing the distribution of the training data. Initially designed for generating 2D images [73], they have been extended to generate 3D images [74] and have found a wide range of applications, including in the medical field [75][76]. A GAN is composed of two neural networks trained concurrently: a generator and a discriminator. The discriminator is a binary classifier whose goal is to identify whether a sample is coming from the real dataset or if it was

generated by the generator. On the other hand, the generator learns the training data distribution in order to generate synthetic data based on the estimated distribution.

In the context of dose prediction for conventional radiotherapy treatments, conditional GANs have been developed such that the generator is able to predict a dose distribution map corresponding to some given anatomical information. In a first time, Mahmood et al. proposed a 2D GAN to predict the dose distribution slice-by-slice by taking as input a 2D slice of the CT with contoured PTV, and OARs [77]. Subsequently, a 3D GAN outperforming the previous model was implemented by Babier et al. [78]. In [79], a 3D attention-gated GAN named DoseGAN was introduced (Figure 3.2). The network was fed with the CT, PTV, and OARs concatenated and it showed better results than the 3D GAN thanks to the attention gates added in the generator and discriminator networks. All previously mentioned GANs are derived from the pix2pix architecture proposed by Isola et al. [80] and rely on a U-net shaped generator. The pix2pix network is a general-purpose conditional GAN designed for image-to-image translation tasks and has found various applications in many fields.

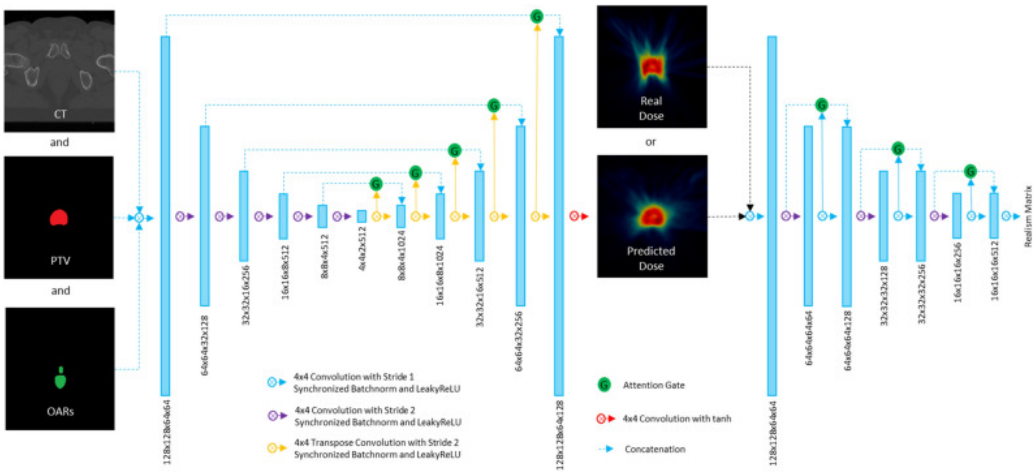


Figure 3.2: The architecture of the DoseGan. On the left side, the CT, PTV, and OARs are fed to the generator network. On the right, the discriminator network generates a realism matrix that aims at distinguishing the real dose distributions from the predicted ones. Credits: [79]

3.2 Fluence map prediction in conventional radiotherapy

In recent years, several papers investigating the prediction through deep learning of fluence maps in photon therapy have been published [81] [82] [83] [84]. Some studies have proposed a treatment workflow equivalent to the one investigated here in the sense that fluence maps are predicted based on anatomical information about the patient and dose per beam predictions.

Lee et al. [85] published a paper on the prediction of fluence maps with a U-net network. The network takes as inputs the 3D contours of the organs and of the target (PTV), together with the 3D beam dose distribution. All inputs are viewed from the Beam's Eye View (BEV) of a single beam, and the fluence map corresponding to that single beam is provided as an output by the network. Some additional residual connections are added between the central channel of the 3D dose distribution and the end of the network (Figure 3.3). However, the authors assumed that predictions of the dose per beam distributions were available, as they considered that previous works on dose prediction were good enough. However, predicting the dose per beam distribution has not yet been widely studied since most of the literature focuses on the prediction of the total dose distribution. The following papers fill this lack of dose per beam prediction by presenting an extended pipeline.

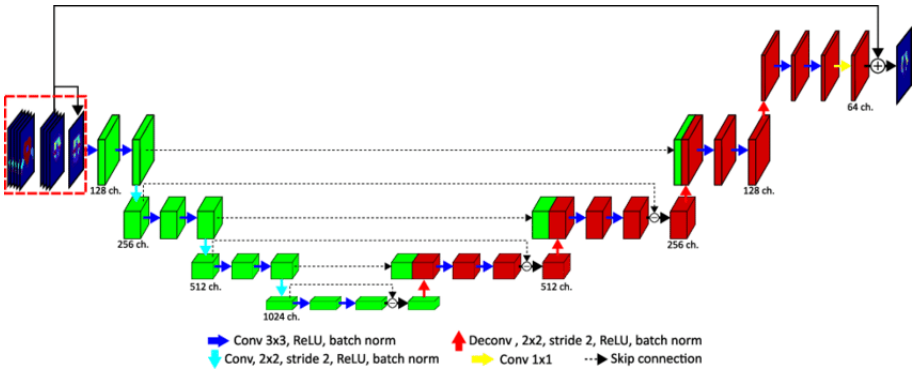


Figure 3.3: Architecture of the network employed by Lee et al. to predict the fluence maps from inputs rotated in BEV. Credits: [85]

Wang et al. published two papers proposing a complete workflow to predict fluence maps for pancreas Intensity Modulated Radiation Therapy (IMRT) with nine beams [86] [87]. In the first one, the network was tested for a single PTV description, and in the second paper, the research was extended to several PTV prescriptions. The first network of the pipeline is predicting the dose per beam distributions of the nine beams simultaneously, slice by slice (Figure 3.4.A). The

network is a CNN (named BD-CNN) with a downsampling block followed by an upsampling block. Information about the CT images is then added through the concatenation of beam templates before the last convolutional block. The loss function used is a weighted sum of the mean square error in the dose per beam predictions and the mean square error in the total dose prediction. The coefficient used in the weighted sum is a hyperparameter tuned during validation. Moreover, the loss is computed only on the ROI, defined as the PTV expanded by 1cm. The second network of the pipeline (called FM-CNN) is predicting the fluence maps and also has a U-net architecture (Figure 3.4.B). It takes as inputs the dose per beam and the PTV maps (single or primary and boost maps) that are projected along the beam's eye view to have 2D inputs. The FM-CNN outputs a 2D fluence map for this single beam. The loss function used for this network is a modified MAE that includes a regularization term in order to avoid overdosing and underdosing by balancing the over- or underestimated regions. This function has inspired the loss function that will be described in Section 4.6.1.

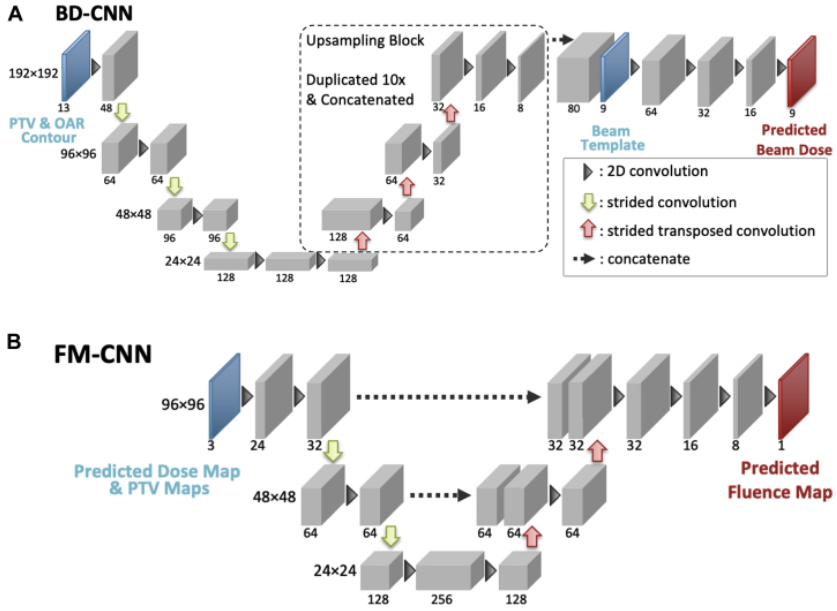


Figure 3.4: The two networks of the workflow proposed by Wang et al. **a)** An encoder-decoder network with 4 levels of resolution predicts slices of the 9 beam dose distributions. **b)** The reconstructed 3D beam distributions are projected along the BEV and fed with the PTV masks to a U-Net with 3 levels of resolution. This U-Net predicts the fluence map. Credits: [87]

Vandewinckele et al. developed a similar automated treatment workflow for the case of lung IMRT with nine beams [88]. Further investigation was made on the impact of the collimator angle and of the different input parameters. The 3D dose per beam predictions are made with a 3D U-net, taking as inputs a CT image, a PTV multi-label map, an OAR multi-label map, and the nine open beam dose distributions. Then, the fluence map is predicted with a 2D U-net using input data projected in the BEV. Various sets of inputs were considered. In addition to the beam dose, networks were trained with and without the CT image and with and without the PTV and OAR multi-label maps. Even if the difference was small, adding anatomical information in the form of a CT image and segmentation maps improved the results. The best combination is the beam dose combined with the CT image. Additionally, a ROI is defined as the PTV extended by a 25mm margin. This ROI binary mask is multiplied with the output before computing the loss function to improve the performance of the network. Compared to the fluence map predicted with clinical beam dose distributions, a deterioration in the fluence map prediction is seen when the dose prediction network and the fluence map network are used sequentially. Therefore, the full automated planning workflow does not provide usable results.

3.3 Fluence map prediction in proton therapy

To my knowledge, the following publication is the only one so far that is exploring the possibility of tuning spot weights with deep learning for proton therapy. Indeed, Zhang et al. proposed a deep learning framework for re-tuning spot weights as a step towards an online adaptive workflow for proton therapy [89]. Online adaptive proton therapy aims at reducing dose distortions due to anatomical variations between fractions by rapidly adapting the initial treatment plan while the patient is on the treatment couch.
For this purpose, Zhang et al. first generated a reference IMPT plan for a single head and neck cancer case. The number of beams used for the reference plan was not explicitly mentioned. Then, data augmentation is performed by modifying the spot weights of the reference plan with different levels of fluctuation. Therefore, the number of spots and their positions were the same across all samples in the dataset. The network was trained to predict the residual between the newly generated spot weights and the reference weights, utilizing only the residual between the new and reference dose maps as input. The architecture of the trained network is a 3D adaptation of the ResNet architecture [90]. The fully connected layer at the network's end had an output dimension equal to the number of spots since it remained constant across the dataset. The proposed framework is consequently patient-specific, and the whole process of data augmentation and training of the network should be repeated for another patient. In the limitations already acknowledged by the authors, it should be highlighted that the model was only tested on a single reference case. Additionally, due to the patient-specific nature of the framework, it lacks generalization possibilities beyond the individual cases, which is needed for our application.

Chapter 4

Methods

This chapter describes the methods that have been employed for this thesis. Even if the main focus was set on the network predicting the spot weights from the dose per beam distributions, the entire pipeline is detailed here. First, Section 4.1 provides information on the available data. Then, the architectures of both the dose per beam prediction network and the spot weight prediction network are explained in Section 4.2. Data pre-processing, the training process, and data post-processing until the final plan generation are described in Sections 4.3, 4.4, and 4.5, respectively. Several variations of the networks have been implemented and tested. Section 4.6 lists some of the main variations that have been explored.

4.1 Data

An anonymized dataset composed of 39 patients with esophageal cancer was available to train the DNNs. IMPT plans with PBS were generated for each patient according to the PROTECT protocol (see the supplementary material in [72] for more information). The dose prescribed was 50.4 Gy, and it was delivered in 28 fractions. Every plan was made with the same beam configuration : two oblique posterior beams with angles of 150°and 180°. The two beams may be referred to in the following sections as beam 1 and beam 2, respectively. Plans were done by an experienced dosimetrist in RayStation using robust optimization.

For each patient, the following information was available:

- The 3D Computed Tomography (CT) scan.
- The 3D mask of all 9 OARs and of the body contour together. The

organs at risk are the fundus, the heart, the left ventricle, the left and right kidneys, the liver, the lungs, the spinal canal, and the spinal canal extended by 3mm.

- The 3D mask of the CTV with the dose prescription in Gray (entries are either 50.4 or 0).
- The 3D full dose distribution in Gray units.
- The 3D dose distribution for beam 1 and beam 2 in Gray units.
- The 3 probability arrays to select slices, rows, and columns that contain the CTV (see Section 4.4.2).
- The spot positions for each beam and their weights. This is provided in two NumPy arrays. Each array has four columns and as many rows as there are spots for that beam. The first three columns contain the x, y, and z coordinates of the spots, and the last column corresponds to their respective weights. These arrays will be referred to as the coordinate array in the following sections.

In the previous list, all volumetric data (dose, CT, CTV, and OAR contours) was available with a voxel size of 1x1x1 mm^3. The 3D images are stored in nifty files, along with some meta data available in the header of the file. It includes information such as the voxel resolution or the number of voxels along each direction of the image. The file also contains information about the relationship between voxel indexes in the 3D array and coordinates in the reference space. This information will be useful for data pre-processing and post-processing (see Sections 4.3 and 4.5).

4.2 Network Architecture

The architecture of the networks employed for the prediction of spot weights and beam dose distributions is the HD U-Net architecture detailed in Section 3.1.2. Similarly, the networks trained for this thesis contain 4 layers (4 dense downsample and upsample operations). The max pooling operation was used to decrease the resolution each time by 2. ReLU was employed in the activation layers, and each 3D convolutional operation was performed with 16 filters. Unlike the original network published in [64], in this version, a normalization layer is added after each activation layer in the form of batch normalization. The dimensions of intermediate layers in the network are determined by the number and size of inputs and outputs, which are model-specific. The following sections describe the models more specifically.

4.2.1 Dose per beam prediction

The first model of the pipeline is the one predicting the dose per beam distributions (Figure 4.1). Since a full dose prediction for esophageal cancer has already been implemented with a HD U-Net [72], the parameters of that network were kept as default parameters. For more clarity, this network will be referred to as the U-Net Dose network from now on.

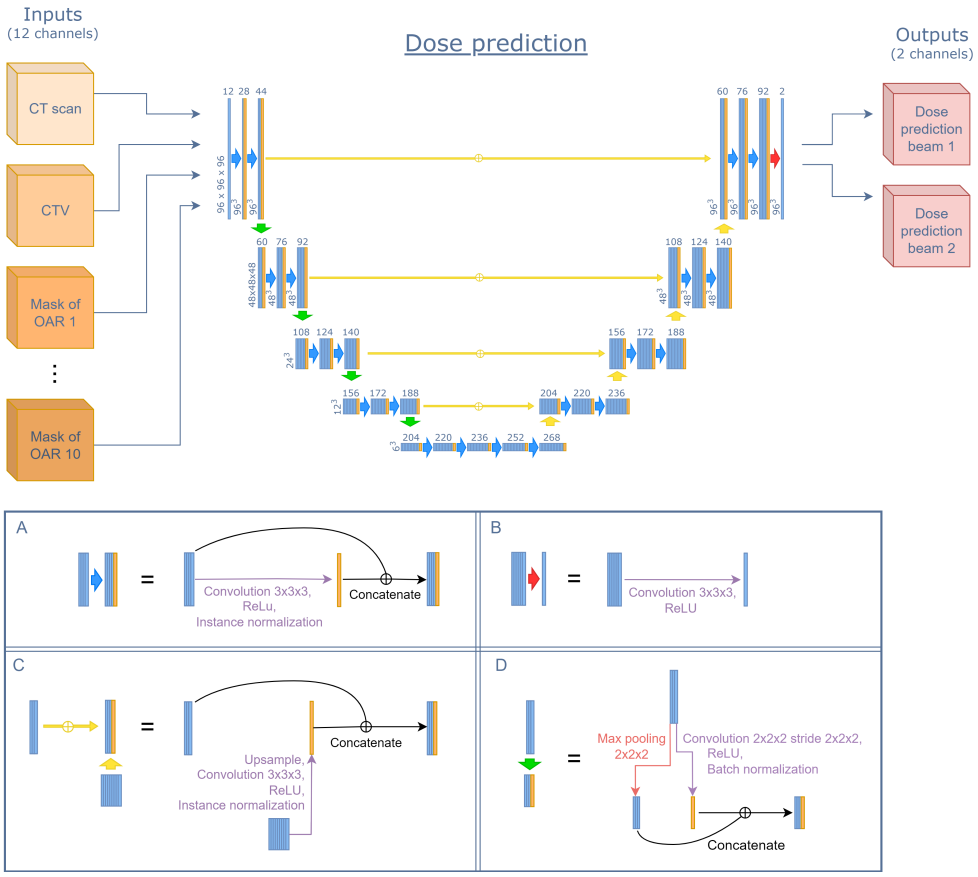


Figure 4.1: **Top**: the architecture of the HD U-Net network predicting the beam dose distributions. **Bottom**: the legend of the operations used in the network. A is the dense convolution, B is a simple convolutional operation, C is the upsampling operation, and D is the dense downsampling operation.

The network has 12 input channels that contain: the CT scan, the mask of the CTV with the dose prescription level, and the 10 boolean masks corresponding to the 9 OARs and the body contour.

The two output channels provide the 3D dose distribution in Gray corresponding to the beam with an angle of 150°and the distribution corresponding to the beam with an angle of 180°.

First, this network was trained on data with voxels of size 1x1x1 mm^3. In a second time, all data was rescaled to voxels of size 3x3x3 mm^3 to train another model.

4.2.2 Spot weight prediction

Figure 4.2: The architecture of the HD U-Net network predicting the weights of the spots. The legend for the different operations employed in the network is the same as above (Figure 4.1).

The second model of the pipeline and the main focus of this thesis is the one estimating the weights associated with each spot (Figure 4.2). To achieve this goal, the following information is provided in the 14 input channels:

- the CT scan;
- the boolean mask of the CTV;
- the 10 boolean masks corresponding to the 9 OARs and the body contour;
- the dose distribution in Gray corresponding to one beam;
- the spot map with unit weights (1 MU).

From this point forward, this network will be named the U-Net Weights network. The single output channel contains the 3D spot map with optimized weights. This matrix should have zeros at the same entries as the matrix provided as an input spot map, but the unitary entries should be replaced by the actual values of the weights associated with each spot. In order to ease the training of the network, the output matrix is directly multiplied by a binary mask matrix with unitary entries where spots are located. This binary mask matrix is similar to the one provided as input.

One of the inputs to this network is the dose distribution delivered by one beam. It also corresponds to the output of the previously described U-Net Dose network. The pipeline was designed such that the output of the first network, the U-Net Dose, should be directly fed to the second network, the U-Net Weights. However, the two networks were only tested individually for now by training the U-Net Weights network on the ground truth dose per beam distributions and not the predicted ones.

All data fed to this network had voxels of size 1x1x1 mm^3.

4.3 Pre-processing

4.3.1 From spot coordinates to 3D matrix

As mentioned in the section above, information about the spot positions must be provided to the U-Net Weights network in the form of a 3D matrix, similarly to other inputs. Additionally, the output of the network is also in the form of a 3D matrix. Entries corresponding to the location of one (or several) spot(s) are filled with the values of the weights, and other entries are set to zero. On the other hand, data about the spots is available in arrays containing their x, y, and z coordinates and their ground truth weight. Consequently, a first script is needed to convert the array of coordinates to the 3D matrix form. To achieve that, the meta data contained in the CT file for instance (see Section 4.1), is retrieved to build the matrix with the same orientation and voxel size as the

other input data. As a result, several spots can fall into the same voxel. To mitigate this issue, the data was scaled to a voxel size of 1x1x1 mm^3. This is relatively small compared to what is usually employed in the literature, with voxel sizes ranging typically from 3mm to 5mm. Despite that, some spots may still fall into the same voxel, but the number is lower. Several strategies were designed to deal with these spots.

First, two different strategies were chosen to adapt the initialization spot map matrix provided as input to the network. Normally, the entries in this matrix should be either 0 or 1, depending on the presence or absence of a spot at that place. If there are multiple spots for the same voxel, the entry can either be set to one or to the number of spots in that voxel. With the first option, the matrix remains binary, but it might be interesting to provide the network with the number of spots per voxel, as done with the second option.

Then, two strategies were also considered to adapt the optimized spot map matrix against which the output of the network will be tested. The entry in a voxel with one spot is directly set to the ground truth weight corresponding to the spot in that voxel. With several spots per voxel, the entry can be set to:

- the sum of the weights of every spot in that voxel;
- the average weight of spots in that voxel.

Two combinations of the previously presented options were considered. The first combination is named *one-add*; only ones are set in the initialization matrix, and where the weights are added in the optimized matrix. The second combination is named *add-mean* and adds the number of spots in the initialization matrix and takes the mean for the optimized matrix.

4.3.2 Dataset generation

Due to the small size of the voxels, the amount of data is substantial, even for only 39 patients, and dealing with such a volume of information is challenging. It requires making as many optimizations as possible in order to ensure reasonable training time and memory consumption.

For this reason, a dataset containing every piece of needed information for all patients is built in the form of a dictionary. Each entry in the dictionary corresponds to another dictionary holding data on a specific patient. By storing all the information in such a data structure, retrieving specific information about a given patient can be done efficiently once the dataset is loaded. Furthermore, since the model will pass over the same patient's data many times during training, it is efficient to pre-process the data as much as possible when creating the dataset. This way, some computations, such as normalizing the CT scan or building the masks of the OAR separately, are only done once. However, the dataset can still require a lot of memory. When creating the dataset

dictionary, every piece of information should be stored with the least amount of memory required. Storing only booleans for binary masks and using 32-bit floats instead of 64-bit floats are examples of optimizations that truly make a difference in obtaining a dataset with a reasonable size that can therefore be loaded in memory. Some additional changes have been tested with the objective of further improving the efficiency of data loading, but they will be discussed in Section 4.6.2.

4.4 Training

Among the 39 patients in the dataset, 30 were used for training, 4 for validation, and 5 for testing. The batch size was set to 1 due to the limited amount of memory available on the GPU. The first models were trained for 100 epochs, and the last models were trained for 150 epochs.

4.4.1 Optimizer

Optimizers are algorithms used to efficiently adjust the weights and biases of a neural network during training in order to minimize the loss function. The step size at which an optimizer updates the parameters is called the learning rate and is one of the most challenging hyperparameters to select due to its significant impact on the model's performance. One of the most widely employed optimizers today is the Adam optimizer [91], whose name is derived from "Adaptive Moment Estimation". The Adam optimizer provides individual adaptive learning rates for the different parameters, which are computed from the estimates of the first and second order moments of the gradients. The Adam optimizer has demonstrated robustness and efficiency across various deep learning applications, and it is particularly well-suited for tasks involving large amounts of data and parameters. It was therefore the optimizer used in this thesis.

4.4.2 Patches

When training the network, working with full matrices is not feasible with the resources currently available. To cope with the computational limitations, the network is fed with only a subset of each matrix at a time. The sub-matrices are called patches and should be as large as possible given the memory constraints in order to give enough context to the model. For the U-Net Weights network, the patches have a size of 128x128x160. And for the U-Net Dose network, the size of the patches is 96x96x128 for the 1x1x1 mm^3 resolution and 96x96x96 for the 3x3x3 mm^3 resolution.
Patches are selected according to the probability distributions available within the dataset (see Section 4.1). This allows the model to focus on the part of the images containing the CTV. Then, to reconstruct the entire 3D prediction, patch-wise predictions must be reassembled together by using a Gaussian filter element-wise and computing a weighted sum of the patches.

4.4.3 Computational power

GPUs are specific hardware components that were initially designed for graphics applications, such as video gaming. However, it was found that the performance characteristics needed in video games are similar to those required for neural network training. As a result, the use of GPUs has been generalized to neural network programming, benefiting from their high memory bandwidth and the parallelism of GPU computing.

Several Nvidia GPUs were available for training, including multiple generations : GeForce RTX 2080 Ti, GeForce RTX 3090, Tesla V100, Tesla A10, Tesla A100 40 GB, and Tesla A100 80 GB. These resources must be shared among the different users. For the computationally intensive task at hand, one of the latest generations of available GPUs was used whenever possible. The GPU employed was primarily chosen from either the 4 Nvidia Tesla A100 with 80 G or the 6 Nvidia Tesla A100 with 40 GB.

4.5 Post-processing and plan generation

The estimated values of the spot weights are provided patch-wise by the neural network in the form of 3D sub-matrices that must be reassembled to obtain the full weight prediction matrix. This section aims at describing the end of the proposed pipeline, from the predicted 3D matrix to the generation of a treatment plan.

4.5.1 From 3D matrix to spot coordinates

The U-Net Weights network outputs the predicted weights in the form of a 3D matrix, where the values of the weights are given in entries located at the spot position. The 3D matrix must be converted back into the coordinate array form in which it was originally provided. This is the inverse operation done during pre-processing (see section 4.3.1). The initial array with the x, y, and z spot coordinates and the associated weights is completed with the predictions of the weight values. When one voxel contains the value associated with several spots, the strategy employed during pre-processing will determine which value will be set in the coordinates array. If the values were added during pre-processing, the estimated weight will be divided by the number of spots in that voxel. If it was the mean of the weights that was computed during pre-processing, the predicted weight values are directly written back to the coordinate array.

4.5.2 Plan generation

From the updated coordinate array, a plan corresponding to the estimated spot weights can be generated. To achieve that, a new script was used to substitute the initial weights with the predicted values within the DICOM files where the plan parameters are stored. Digital imaging and communications in medicine

(DICOM) is an international standard for medical images and related information. The script changes the weights and updates all parameters accordingly, energy layer by energy layer. Finally, the resulting dose of the modified plan can be computed and visualized with RayStation.
With this last script ending the pipeline, a full workflow that can be integrated into RayStation has been developed.

4.6 Experiments

4.6.1 Loss functions

Dose per beam prediction

Two different loss function variations were tested for the U-Net Dose network. As previously explained, the doses of the two beams are returned at the same time by the network as the two output channels.

The first loss function tested was a MSE only including the error made on the total dose. For a ground truth total dose y_{total} , a dose prediction for beam 1 $\hat{y}_{beam_1}$, a dose prediction for beam 2 $\hat{y}_{beam_2}$, and for N voxels, the loss function can be written as:

$$MSE_{total} = \frac{1}{N}\sum_{N}(\hat{y}_{beam_1} + \hat{y}_{beam_2} - y_{total})^2 \tag{4.1}$$

The second loss function tested was a weighted sum of the total dose error and of the error in each individual beam dose. By taking into account the error made in each individual beam, it could encourage a balanced distribution of the total dose across each beam and thus lead to more accurate and realistic beam dose maps. The loss function can be written as:

$$\begin{aligned} MSE_{weighted} = 0.25 * \frac{1}{N}\sum_{N}(\hat{y}_{beam_1} - y_{beam_1})^2 \\ + 0.25 * \frac{1}{N}\sum_{N}(\hat{y}_{beam_2} - y_{beam_2})^2 \\ + 0.5 * \frac{1}{N}\sum_{N}(\hat{y}_{beam_1} + \hat{y}_{beam_2} - y_{total})^2 \end{aligned} \tag{4.2}$$

Spot weights prediction

Since the loss function used is a key component when training a DNN, several functions have been experimented for the U-Net Weights. As explained in Section 2.2.1, the main two loss functions employed in regression problems are the MSE (Equation 2.1) and the MAE (Equation 2.2) loss functions. However, we can also change the ROI over which the error is computed. Typically, the mean

error is computed over the entire output, but in this application, we are only interested in specific values of the output, i.e., the values at the spots' locations. This is why the mean error can be evaluated in a restricted region of the output. Different restricted regions have been tested. Hereafter, the loss functions that have shown the most significant results are described.

First, we define the MSE_{body} loss function, which is a MSE computed over the voxels contained in the body of the patient by using the corresponding binary mask available.

Then, a specific ROI is defined around the CTV. This ROI is the CTV that is extended in order to include all spot locations. The binary mask corresponding to that ROI is shown in Figure 4.3, together with the spot locations and the CTV delineation. A second loss function can be defined as the MSE_{ROI} function. This is a MSE computed over the ROI previously described.

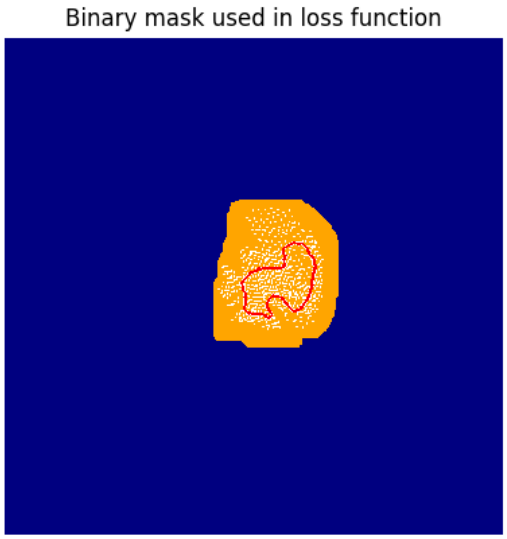


Figure 4.3: An axial slice of the binary mask of one patient used in the MSE_{ROI} loss function is represented in orange. The small white crosses indicate the location of spots, and the CTV is delineated in red, but they were added for the visual representation and are not part of the binary mask employed.

Finally, a modified MSE was tested. This MSE was inspired by the modified MAE loss function used by Wang et al. in [86] and mentioned in Section 3.2.

This loss function is defined as:

$$MSE_{regularized} = (1 + \lambda)\frac{\sum_{i=N_{ROI}}(y_i - \hat{y}_i)^2}{N_{ROI}} \tag{4.3}$$

The regularization term λ is introduced to prevent over- and underestimations by balancing the number of positive and negative errors. This term is defined as:

$$\lambda = \frac{|N(y - \hat{y} > 0.01) - N(y - \hat{y} < -0.01)|}{N_{ROI}} \tag{4.4}$$

In the above equations, N_{ROI} is the number of voxels in the ROI around the CTV described earlier. $N(y - \hat{y} > 0.01)$ is the number of voxels with an error superior to 0.01. Since y represents the predictions and $\hat{y}$ the ground truth values, this number is the amount of largely underestimated weights. Similarly, $N(y - \hat{y} < -0.01)$ corresponds to the number of weights largely overestimated.

4.6.2 Loading data

The first challenge encountered when training the network was handling the large amount of data. To obtain code that can run in a reasonable amount of time, several approaches have been considered. The DNNs were implemented with PyTorch, which is a machine learning framework. DataLoader and Dataset are two data primitives provided by PyTorch that allow loading and iterating over the data [92]. A custom Dataset was implemented to store the data, and DataLoader was used to access the data by warping an iterable around the Dataset. The Dataloader returns samples in batches of a given size, and it allows sample shuffling, custom sampling, and multi-processing. Even if using the Dataset and DataLoader classes provided by PyTorch did improve performance in a first time, the optimizations explained in Section 4.3.2 were the most efficient in the end.

Chapter 5

Results

The results obtained for the prediction of the dose per beam distributions are presented in Section 5.1. Section 5.2 details the outcomes of the spot weight predictions. A discussion of these results will be carried out in Section 6.

5.1 Dose per beam prediction

As mentioned earlier, the U-Net Dose was initially trained on data with a voxel size of 1x1x1 mm^3. However, this resolution gave poor results, including artifacts related to the patching system (see Appendix A). Therefore, the following results were obtained with a voxel size of 3x3x3 mm^3. Moreover, two loss functions were presented in Section 4.6.1 for this network: the MSE_{total} (Equation 4.1), which takes only the total dose error into account, and the $MSE_{weighted}$ (Equation 4.2), which includes the individual beam error. Since the MSE_{total} failed to provide realistic beam dose distributions (see Appendix A), only the results obtained with the $MSE_{weighted}$ loss function are shown in the following sections.

5.1.1 Evaluation metrics

First, Table 5.1 presents the mean error and its standard deviation of the dose predictions obtained over all test patients and over all folds. For each test patient, this error was computed by taking either the MSE or the MAE over each voxel of the dose distribution. It was done for each beam individually, and the predictions of the two beams were summed and compared to the true full dose distribution.

	MSE	MAE
Beam 1	3.63 ± 1.76	0.65 ± 0.32
Beam 2	3.90 ± 2.20	0.69 ± 0.35
Full dose	8.73 ± 7.52	1.08 ± 0.70

Table 5.1: Mean and standard deviation of the error over the dose distributions computed for all 25 test patients.

Table 5.2 shows similar metrics but calculated for specific ROIs this time. The MSE and MAE are computed by taking the difference between the ground truth full dose and the sum of the two dose beam predictions. The ROIs are the CTV and three organs that are often the most critical in esophageal cancer cases: the heart, the lungs, and the spinal canal.

ROI	MSE	MAE
CTV	7.4632 ± 19.3202	1.7778 ± 1.7386
Heart	12.2723 ± 6.7186	1.8252 ± 0.5567
Lungs	7.0855 ± 4.4761	1.3728 ± 0.5001
Spinal canal	12.8091 ± 18.9583	1.8874 ± 1.2267

Table 5.2: Mean and standard deviation of the full dose error over all 25 test patients and in 4 different ROIs: the CTV, the heart, the lungs, and the spinal canal.

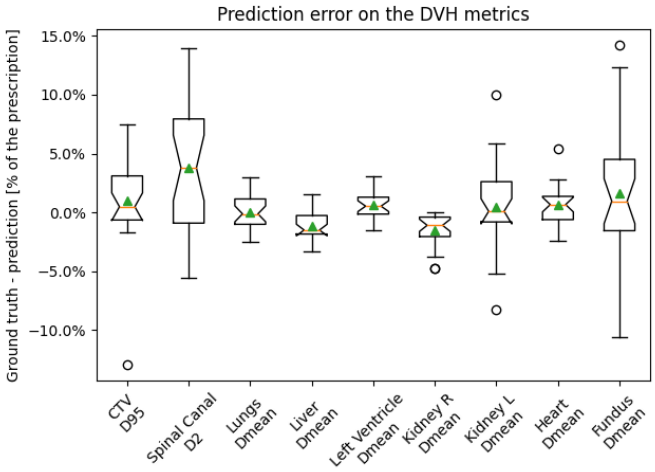


Figure 5.1: Boxplots of the prediction error on DVH metrics in percentage of the prescription dose (50.4 Gy). The box extends from the first to the third quartile of the data: the orange line is the median, and the green triangle is the mean. **From left to right** : the D95 dose for the CTV, the D2 dose for the spinal canal, and the average dose for all the other OARs.

In addition, the prediction errors on DVH metrics have been calculated for each test patient and are represented with boxplots in Figure 5.1. The maximal dose peak in the CTV is displayed in Figure 5.2. For each test patient, the maximal dose predicted by the network is compared to the one in the ground truth plan. The DVH of one patient is shown in Figure 5.3 with the CTV curves in yellow and with the curves of every OAR.

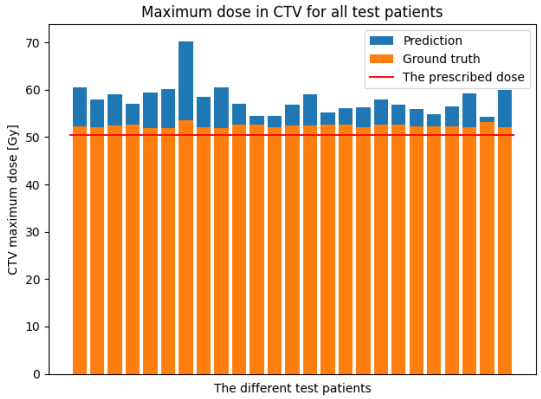


Figure 5.2: Comparison of the maximal dose in the CTV between the ground truth and the prediction across all 25 test patients.

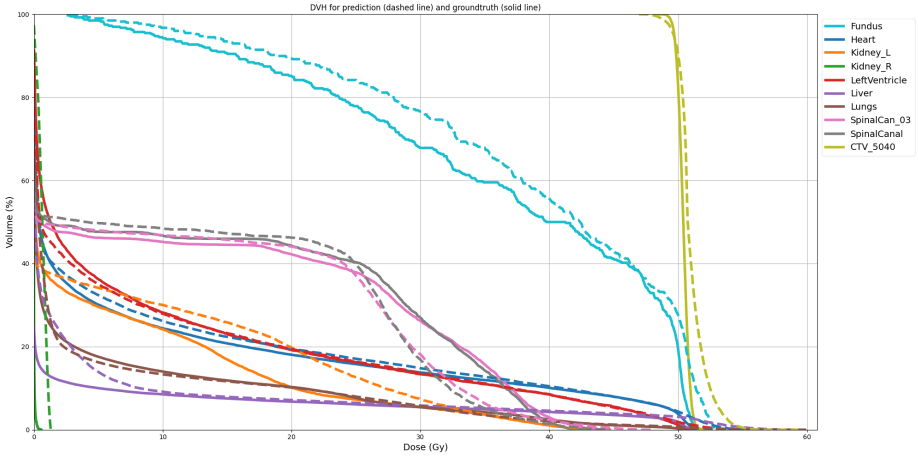


Figure 5.3: The Dose Volume Histogram for patient 45. The full lines are the ground truth, and the dashed lines are the prediction and were computed on the summed distribution of the two beams.

5.1.2 Visual comparison of dose distributions

Figure 5.4 displays dose maps for one of the test patients. For each image, the same axial slice of the patient was taken, and the CTV contour corresponding to that slice was added. The first column shows the true dose distribution, the second column is the prediction and the last column shows the difference between the two. In this last column, the red regions correspond to underdosed regions, and the blue regions correspond to overdosed regions.

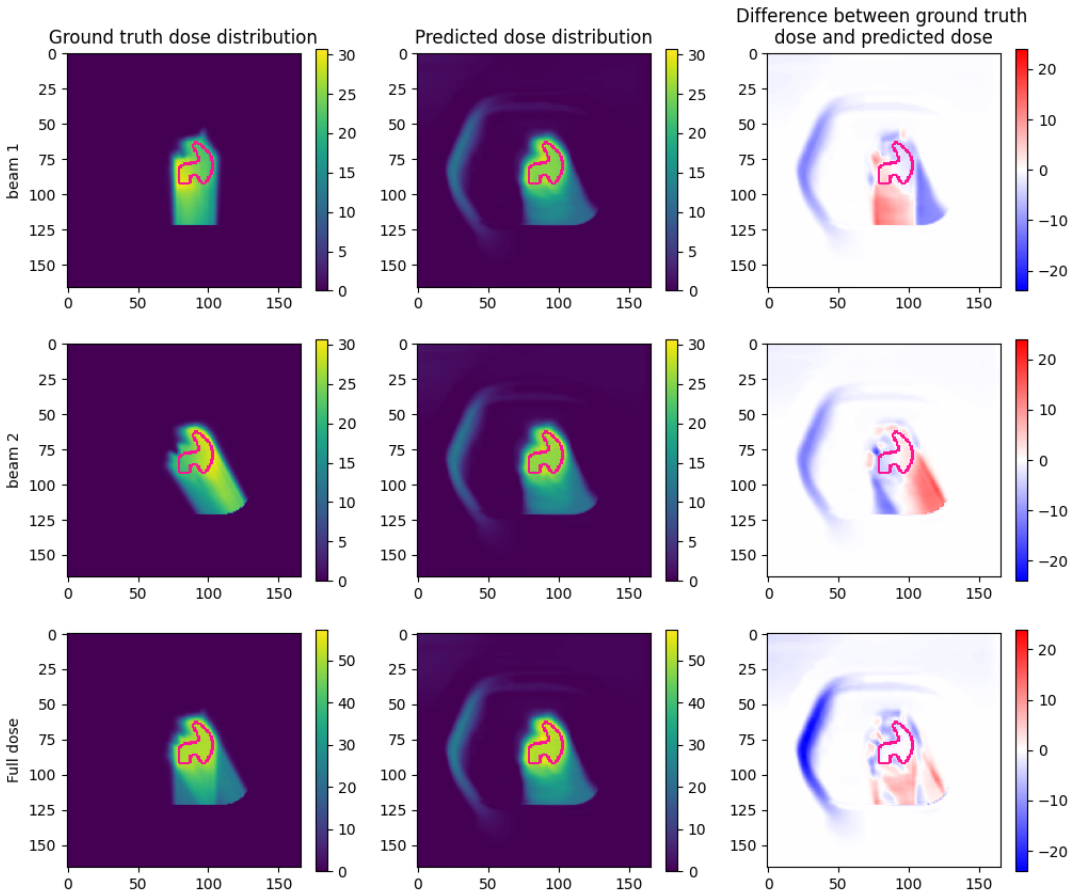


Figure 5.4: Dose maps to compare the predicted dose distributions and the ground truth distributions for patient 45 in Gy. The CTV is delineated in dark pink in each map. **From left to right** : the ground truth, the prediction, and the difference between the two. **From top to bottom** : beam 1 (180°), beam 2 (150°), and full dose with both beams. For better visualization, the dose scales differ between the beam dose maps and the full dose maps.

5.2 Spot weight prediction

Several models of U-Net Weights have been trained and tested. In particular, different loss functions have been employed, as described in Section 4.6.1. In the following sections, three models will be compared. Since their loss function is what differentiates them, the models will be named after their loss function: the MSE_{body}, the MSE_{ROI}, and the $MSE_{regularized}$. In Section 4.3.1, several strategies regarding the mapping of spots to the same voxels have been discussed. Due to a lack of satisfactory results, only one strategy was tested. All the results presented here were obtained by following the *one-add* strategy.

5.2.1 General metrics

Table 5.3 holds the mean errors of the predicted weights and their standard deviation for the three tested models. The Root Mean Square Error (RMSE) and the MAE were computed for each test patient in each fold and then averaged. For each patient, the error was computed over every predicted spot weight value.

Model	RMSE	MAE
MSE_{body}	0.1329 ± 0.0195	0.0745 ± 0.0106
MSE_{ROI}	0.1347 ± 0.0237	0.0793 ± 0.0142
$MSE_{regularized}$	0.1052 ± 0.0184	0.0644 ± 0.0105

Table 5.3: The mean and standard deviation of the error of the predicted weights across all 25 test patients.

The distributions of spot weight values for one patient are shown in Figure 5.5. The upper left histogram shows the distribution of values for the ground truth weights. The three other histograms were constructed with the weights predicted by the three trained models. Additionally, Table 5.4 provides information about how many spot weights have a value equal to zero for each model. This information is provided as a mean percentage of spots across all test patients. This table also gives the percentage of spots with weight values below 0.011 MU but not equal to zero since weights below this threshold value of 0.011 will be considered as zeros by the machine.

Model	weights $= 0$	$0 <$ weights < 0.011
MSE_{body}	$0.0\% \pm 0.0$	$1.9053\% \pm 2.1453$
MSE_{ROI}	$43.6900\% \pm 32.9856$	$26.7498\% \pm 8.7810$
$MSE_{regularized}$	$0.8476\% \pm 1.8173$	$7.3081\% \pm 8.1053$

Table 5.4: The percentage of predicted weight values that are equal to 0 MU and the percentage of weights with a predicted value inferior to 0.011 MU, but not equal to zero. Each percentage is averaged over all 25 test patients, and reported for the three different models.

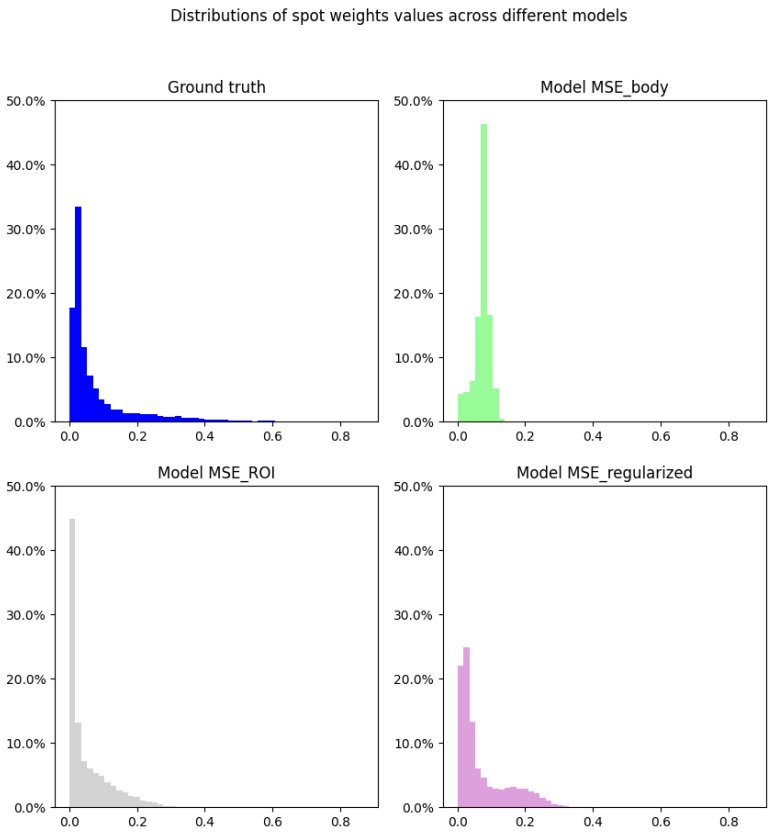


Figure 5.5: The distribution of weight amplitudes for beam 1 of patient 45. The X-axis represents the weight values. The Y-axis represents the percentage of spots with weight values within the given interval for that beam and that patient.

5.2.2 Distribution of the error in space

In order to have information about the spatial distribution of the error within each test patient's body, several rings were delineated in and around the CTV. As it can be seen in Figure 5.6, the CTV was divided into three regions, delineated by the contour of the CTV and contours obtained by eroding the CTV by a 5mm and 10mm margins. Similarly, three regions outside the CTV were defined by extending the CTV by 5mm and 10mm margins. Notice that Figure 5.6 shows only one axial slice of the 3D contours, and which is why the margins between the regions might not seem regular. To give an idea of the distribution of spots within these regions, Figure 5.6 also indicates the percentage of spots in each ring averaged over all patients.

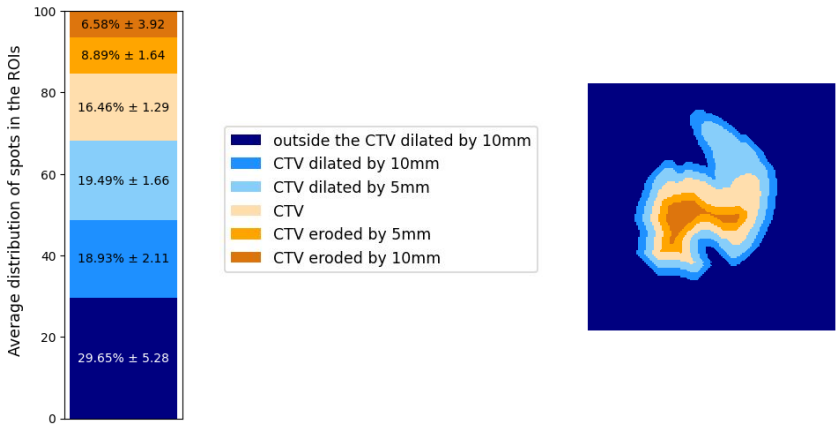


Figure 5.6: **Right** : an axial slice of the different rings around the CTV of patient 45. Orange parts are in the CTV, and blue parts are out of the CTV. **Left** : the average percentage of spots in each ring around the CTV across all patients.

The error made in each of these regions is depicted in Figure 5.7. For each model and each ring in and around the CTV, a boxplot shows the distribution of the MAE across all patients. As a reminder, the regions were defined as rings and are therefore disjoint.

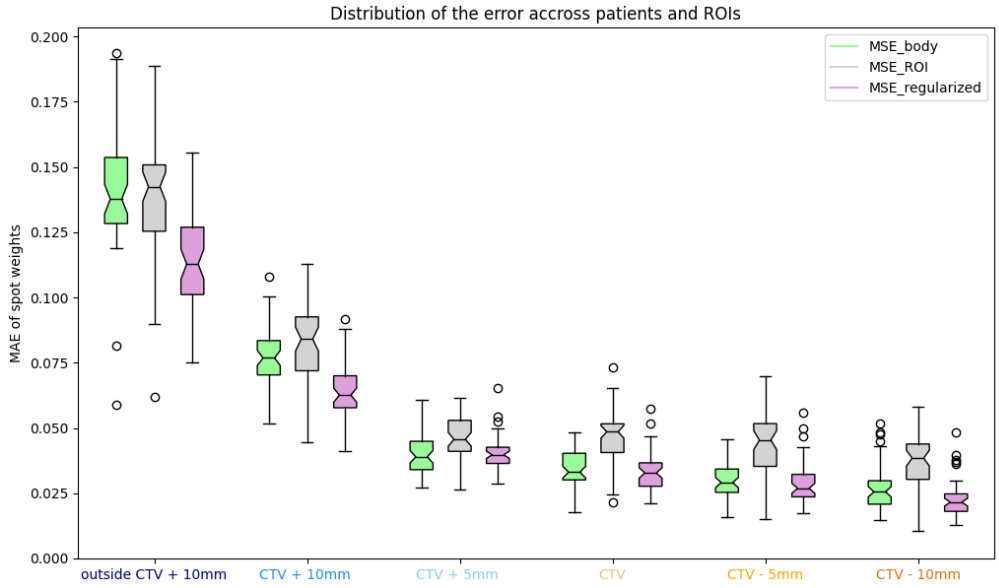


Figure 5.7: Spatial distribution of the MAE across all 25 test patients and for the 3 models

The amplitudes of spot weights and their spatial location can be observed in Figure 5.8. To ensure a sufficient number of spots for an improved visualization, each map is constructed by selecting spots from five consecutive axial slices. The CTV contour corresponding to the central slice out of the five was overlaid in red on each image. For each beam, the ground truth spot map and the spot maps of the three models are shown. The amplitude scales differ for each image in order to maintain a clear visualization.
The difference between the ground truth map and each model can be seen in Figure 5.9. Red spots are underestimated, and blue spots are overestimated.

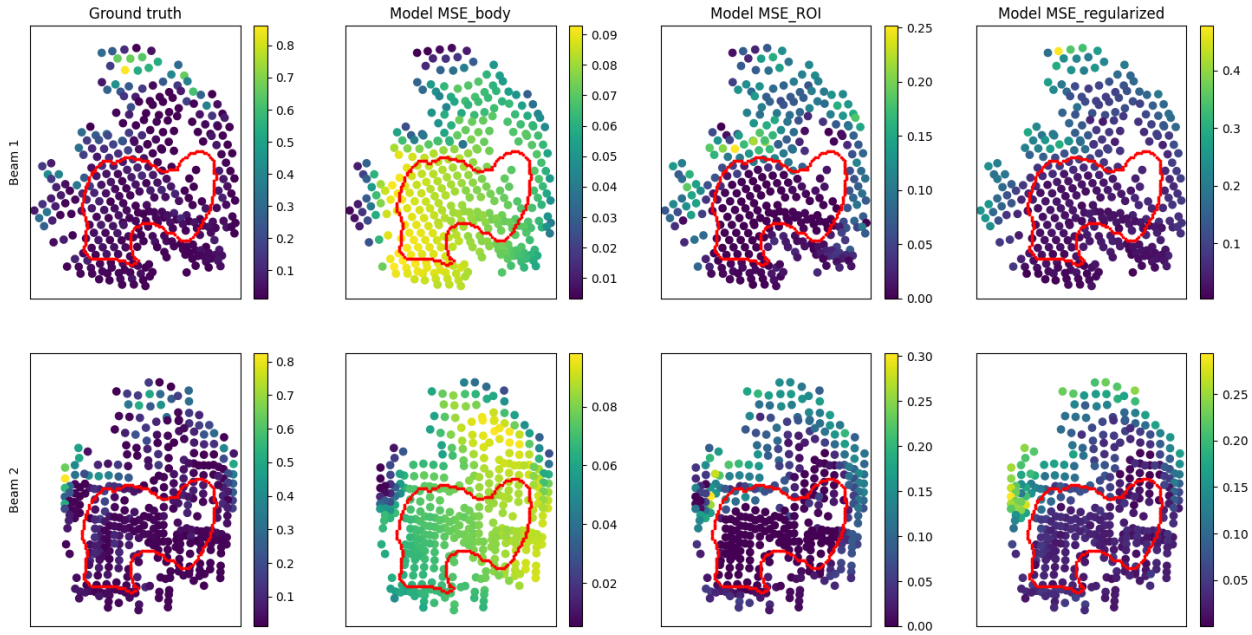


Figure 5.8: The amplitude of weights for different spots in patient 45. The CTV contour is in red. The **upper row** shows spot maps for beam 1 and the **lower row** for beam 2. **From left to right** : the ground truth spot map, the MSE_{body}, MSE_{ROI}, and the $MSE_{regularized}$ spot maps.

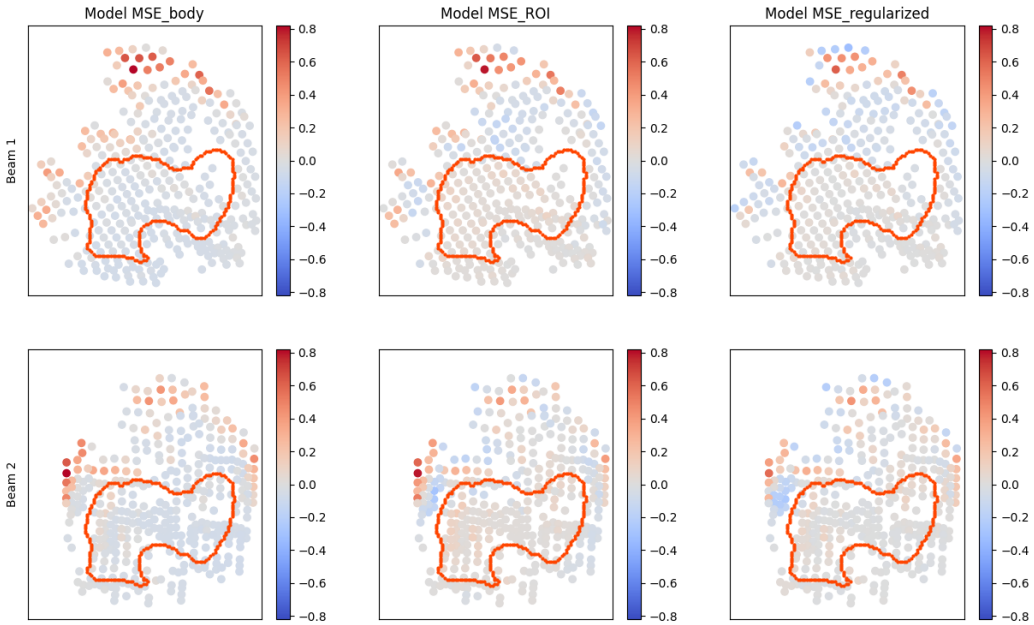


Figure 5.9: The difference between the ground truth and the predicted weights for different spots in patient 45. The CTV contour is dark orange. The **upper row** shows spot maps for beam 1 and the **lower row** for beam 2. **From left to right** : the MSE_{body}, MSE_{ROI}, and the $MSE_{regularized}$ spot maps

5.2.3 Resulting dose

Finally, this section presents the dose resulting from the predicted spot weights for a patient. The results in this section are presented as screenshots from the RayStation interface. On the right side of each image, isodoses resulting from the given weights are shown on an axial slice of patient 45. This axial slice is the same as in previous figures. On the left side of the image, the associated DVH is displayed, with one curve corresponding to each ROI delineated on the dose map. The CTV is in red, the lungs in blue, the heart in orange, and the spinal canal in pink. First, Figure 5.10 shows the dose corresponding to non-optimized weights that are all set to one. The three following screenshots, Figure 5.11, Figure 5.12, and Figure 5.13, present the doses associated with weights predicted with the MSE_{body}, MSE_{ROI}, and $MSE_{regularized}$ models, respectively. Figure 5.14 displays the optimized and ground truth dose.

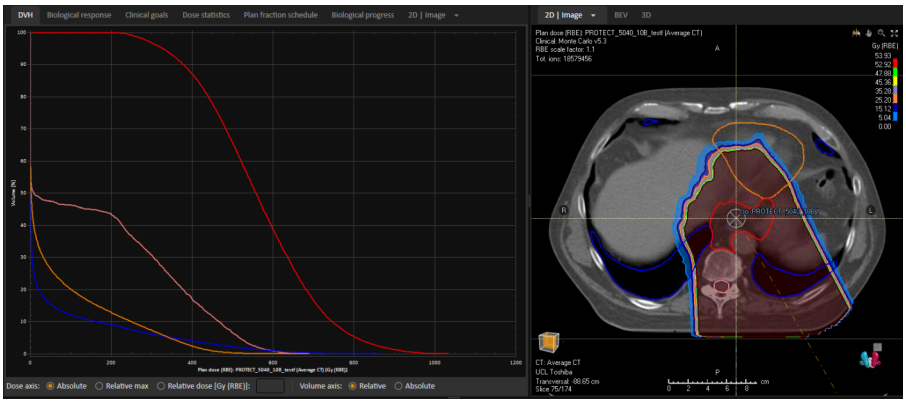


Figure 5.10: DVH and dose map [Gy] obtained with unitary weights for patient 45.

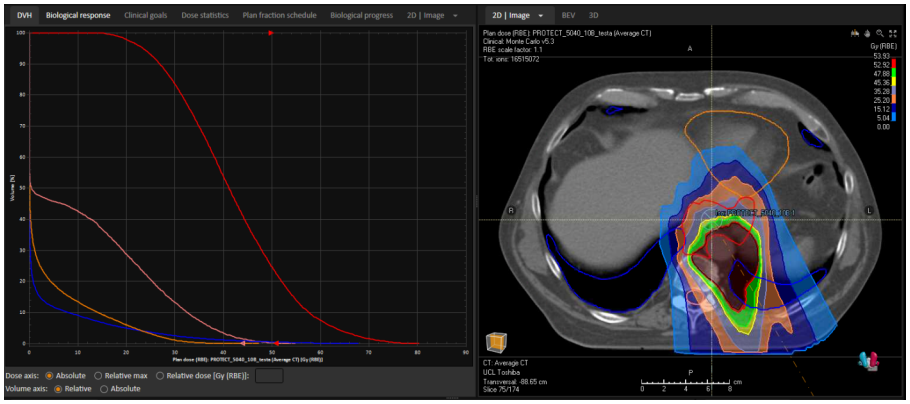


Figure 5.11: DVH and dose map [Gy] obtained with the MSE_{body} model for patient 45.

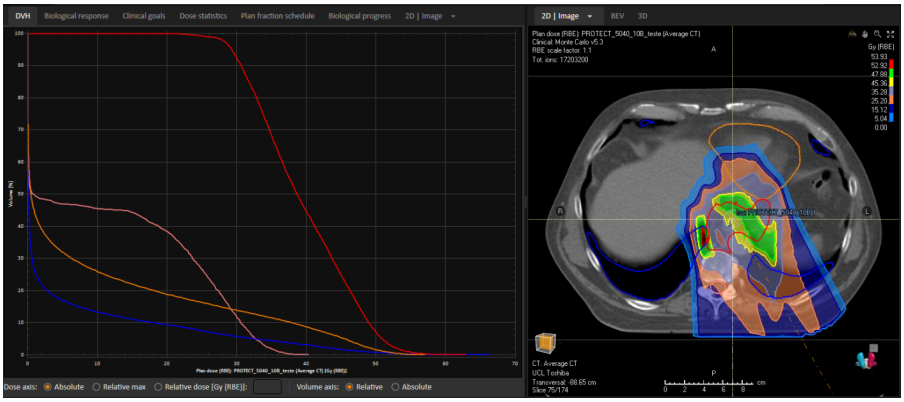


Figure 5.13: DVH and dose map [Gy] obtained with the $MSE_{regularized}$ model for patient 45.

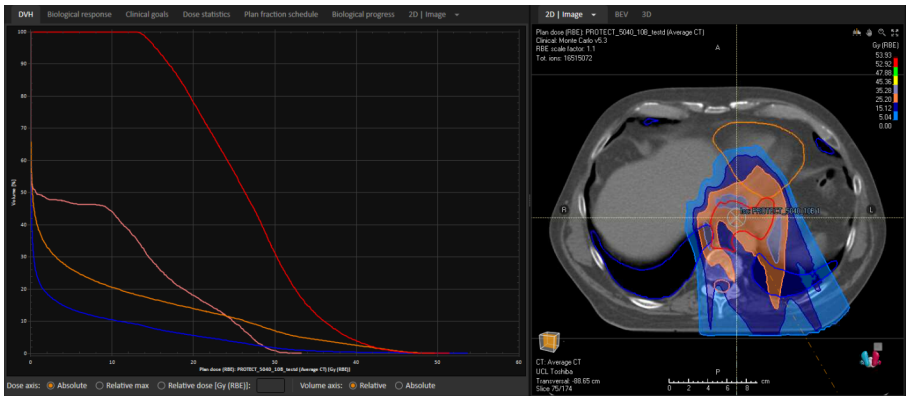


Figure 5.12: DVH and dose map [Gy] obtained with the MSE_{ROI} model for patient 45.

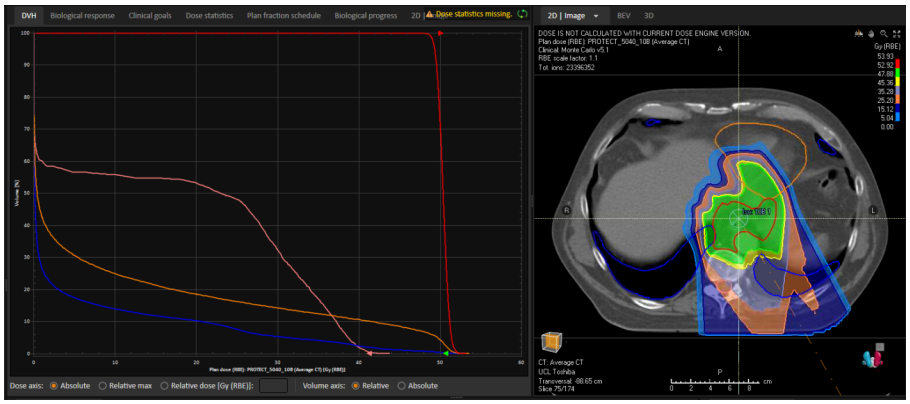


Figure 5.14: Optimized ground truth DVH and dose map [Gy] for patient 45.

5.3 Computation time

Training the U-Net Dose network with a voxel size of 3x3x3 mm^3 and over 150 epochs for the 5-fold cross-validation took about 3h56 with the Nvidia Tesla A100 GPU, and it takes about 30 seconds to evaluate the two beam dose distributions of one patient.

The training time of the U-Net Weights network for 150 epochs and the 5-fold cross-validation is about 6h55 for the $MSE_{regularized}$ model, 6h20 for the MSE_{ROI} model, and 6h25 for the MSE_{body} model, all trained on the Nvidia Tesla A100 GPU. For the two beams of one patient, it takes about 24 minutes to evaluate the spot weights and to make the conversion of the 3D matrices back to the vector form.

Chapter 6

Discussion

This discussion starts with the interpretation of the obtained results in Section 6.1. Several areas for improvement are provided in Section 6.2. The limitations of the pipeline and of the employed methods are acknowledged in Section 6.3. Finally, Section 6.4 presents perspectives on potential future applications.

6.1 Results interpretation

6.1.1 Dose per beam prediction

As it was explained earlier, the U-Net Dose network was trained on data having a voxel size of 1x1x1 mm^3 in a first time in order to have beam dose predictions of the same resolution as the ones needed for the U-Net Weights network. However, many artifacts were observed in the first predictions (see Appendix A). For instance, some lines appeared regularly on the dose predictions, and the contours of the body and of some organs were appearing. Consequently, the voxel size was increased to provide more context information in each patch.

General metrics

Two loss functions were tested for the training of the U-Net Dose network. As mentioned earlier, the results obtained with the MSE_{total} loss function, which only considers the error made in the total dose distribution, were not satisfactory. Indeed, the predicted distributions were very uneven between the two beams and therefore not realistic (see Appendix A). However, from Table 5.1, we can observe that the network trained with the second loss function, the $MSE_{weighted}$, seems not to be biased towards one beam. Taking into account the error made on the full dose and on each individual beam dose prediction provides similar MSE and MAE errors between the two beams, though a slight

increase in mean error and standard deviation can be noticed for beam 2 (150°).

Metrics on the CTV

First of all, we can see from Figure 5.1 that the D95 of the CTV is lower in the predictions than in the ground truth for the majority of the tested patients. Only one outlier patient has a much higher D95 than the ground truth. The mean error is 1% and the interquartile range is less than 5%. However, when looking at the maximal dose in the CTV in Figure 5.2, we can observe that this peak dose is higher than in the ground truth plan for all test patients. Compared to the ground truth, the maximal dose is increased on average by 5 Gy, which represents a 10% increase in terms of the prescribed dose. The DVH in Figure 5.3 shows a CTV curve for the prediction that is less steep and slightly right-shifted compared to the ground truth line. All these observations indicate that even if the dose predicted for 95% of the CTV's volume is not high enough for a number of patients, all of them suffer from the presence of hot spots in the predicted dose distributions.

Metrics on the OARs

We can observe in Figure 5.1 that the Dmean error for the fundus and, to a lesser extent, for the left kidney are highly varying across patients. There is also a higher variance for the D2 of the spinal canal, with a mean and variance of 3.8% and an interquartile range of 8.85%. This high variance in the spinal canal is also confirmed by Table 5.2. This table also reports the errors made in critical OARs such as the heart and the lungs. Even if the variability across patients is much smaller, the mean error made in the heart is in the same range of values as the one made in the spinal canal, contrary to the lungs, which have a smaller error. Finally, the DVH of patient 45 displayed in Figure 5.3 shows that the predicted plan delivers more dose to most OARs than the ground truth plan, especially for the fundus, the left kidney, and the liver.

Visual comparison of dose distributions

Two main observations can be added by analyzing Figure 5.4. First, a non-zero dose is predicted by the network at locations far away from the tumor and not in the beam path, which is abnormal. Especially a part of the body's contour is predicted with a dose over 20 Gy, while it should be predicted to zero. Second, we can clearly see in the beam dose maps that the predictions made for each beam individually do not take into account the fact that the beam comes from a specific angle. On the ground truth dose maps, we can distinctly observe at which angle the beam is, while the dose distributions are similar for both beams and are a mix of the two angles.
With these two observations, we can say that the predictions made by the U-Net Dose network are not realistic. Even if the difference in dose observed in the CTV is quite low, the results are not good enough to be used in the pipeline.

Since the goal of this network is to provide guidance information for the U-Net Weights network, unrealistic beam dose distributions could badly impact the performance of this second network and complicate the learning procedd. Added to the fact that CTV contains hot spots, this network should still be improved.

6.1.2 Spot weight prediction

General results

Among other things, the U-Net Weights was trained with three different loss functions. According to Table 5.3, the best model is the one with the $MSE_{regularized}$ loss function since it has the smallest overall error. RMSE and MAE values are more similar for the other two models, although the MSE_{body} has a slightly lower mean error and variance.

When looking at the distributions of weight amplitudes in Figure 5.5, we observe that the MSE_{body} has a distribution that is less similar to the one of the ground truth weights. Between the other two models, the distribution of MSE_{ROI} is more like the true distribution at first glance. However, we can see that the percentage of weights in the first left bin of the histogram corresponding to small weights is far higher for this model.
This is confirmed by the information available in Table 5.4. The percentage of weights with a zero value greatly varies between models. While all values predicted by the MSE_{body} are greater than zero, the $MSE_{regularized}$ model predicts 0.85% of zero weights on average. This is rather small compared to the 43.7% of zero weights predicted by the MSE_{ROI} model. The issue of having many weights set to zero has been encountered many times when training different types of models. In the case of the MSE_{ROI}, we can see that this percentage has a great variance across patients. Moreover, it has been observed that the percentage also varies manly across the different cross-validations. Either the percentage of zeros was very high or the percentage was more reasonable, although often higher than the two other models. The patient whose distribution is displayed in Figure 5.5 was in the last category.
The second column of the table indicates the percentage of non-zero weights smaller than 0.011. Since below this value, the weights will be considered equal to zero by the machine, they should be taken into account. Once again, the MSE_{ROI} is the one predicting the greatest percentage of such small weights, and, added to the zero-valued weights, it results in a high portion of weights that are considered as zeros. This is therefore important to take into account when evaluating the weight predictions of a specific patient since they might not be representative for this model due to higher inter-patient variability.

Distribution of the error in space

When observing the regions defined in Figure 5.6, we can see that a high percentage of spots are located outside the CTV, with almost 30% of spots in the

outer region. It should therefore be reminded that the ground truth plans in the dataset were made with robust worst-case optimization using a 7mm setup error, explaining the numerous spots outside the CTV.

Overall, we can see in Figure 5.7 that the error increases when the spots are farther outside the CTV for all models. More specifically, we can see that there is less variation in the MAE across the three regions inside the CTV for a given model. Inside the CTV, the MSE_{ROI} is the worst-performing model, while the two other models show more similar MAE but slightly better values for the $MSE_{regularized}$. However, when going away from the CTV, the MAE of the MSE_{body} increases more than the two other models. In the outer region, the $MSE_{regularized}$ has the lowest MAE of all three models.

The overall increase in error outside the CTV is confirmed when looking at Figure 5.8 and Figure 5.9. For this patient, we can see that the range of weight amplitudes varies a lot across models. The ground truth weights are smaller in the CTV, with values approximately below 0.2 MU. Only a few spots outside the tumor have higher weights, up to 0.8 MU. More precisely, these spots are located behind the tumor when looking in the BEV direction. The MSE_{body} map shows spots with really small weights that are even smaller where the higher spots should be. The maps of the two other models look more similar to the ground truth map, but spots with peak values are still underestimated. Overall, the error maps show that the error is indeed more important outside the CTV, with spots that are both underestimated and overestimated. However, the largest errors are coming from underestimated spots, and the weight errors made by the $MSE_{regularized}$ model have a lower amplitude. This confirms the previous observations made with the other metrics that put the $MSE_{regularized}$ model as the best-performing one overall.

Resulting dose

In general, the dose resulting from the predicted spot weights for all three models (Figure 5.11, Figure 5.12, and Figure 5.13) is far from what it should be (Figure 5.14). For each model, the DVH curve associated with the CTV is not steep enough, and a number of clinical goals are not met. On the dose distribution obtained with the MSE_{body} model, we can see that isodose lines associated with the highest doses are not centered around the CTV and are mainly outside it. This results in an averaged dose in the CTV that is too low (41.9 Gy). In addition, it also causes very large overdose peaks in the body, with a dose peak of over 82 Gy. The spinal canal is especially overdosed. The isodoses of the other two models are slightly more centered around the CTV, even if they are far from looking like the ground truth dose distribution. The MSE_{ROI} model results in a highly underdosed CTV with an average dose of 26.47 Gy. Finally, the $MSE_{regularized}$ also results in an underdosed CTV, but to a lesser extent (39.4 Gy). However, it comes at the cost of higher dose peaks in the body (over 65 Gy).

Even if previous analyses indicated the $MSE_{regularized}$ model as the most promising one, none of the tested models provided good enough results when it came to the resulting dose distribution and the clinical goals. The U-Net Weights network could not be used as it is to directly predict the spot weights and needs to be further improved.

6.2 Prospects for improvements

6.2.1 Dose per beam prediction

Since the current U-Net Dose network predicts dose at locations such as the body contour and does not take into account the trajectory of the beams for the individual beam dose predictions, it could be interesting to provide information about the beam trajectories to the model. This could be done by adding the beam mask to the inputs or by using it to compute a masked CT. By adding prior knowledge that there is dose deposition only on the beam's trajectory, the predictions could turn out to be more realistic.

Another solution that can be implemented independently or in complement to the previous one is to fine-tune the weighting factors of the $MSE_{weighted}$ loss function. In Equation 4.2, the 0.25 coefficient behind the individual beam dose error and the 0.5 coefficient behind the full dose error were chosen arbitrarily. Adding more weight to the individual beam errors could be a solution to obtain more realistic results. As done in [86], the weighting factor could be a hyperparameter tuned during validation.

6.2.2 Spot weights prediction

Given the observations made when analyzing the results of the U-Net Weights network, several ideas that could potentially improve the performance of the network are explained in the following paragraphs. First, some solutions involving data manipulation will be presented. Then, a discussion on the network's architecture will be held, adding broader perspectives. Finally, insights about the computation time will be provided.

But before that, one possibility of improvement could also be to fine-tune the value of the parameter in the $MSE_{regularized}$ loss function. In the expression of the regularization term λ (see Equation 4.4), different values than 0.01 could be tested. Even if a smaller value than 0.01 was indeed tested and did not show much improvement, a more thorough approach testing different ranges of values could be conducted.

Data

To begin with, the size of the data voxels could be increased, as done for the U-Net Dose network. For the first network, it has proven beneficial to add more context information to each patch of data. This could also be interesting for the U-Net Weights network, given the fact that the network seems to have a hard time predicting the values of the spot weights that are farther away from the CTV. And since the dose distributions are concentrated in the CTV but the spots with the highest weight values are located outside the CTV, increasing the amount of context information in each patch could improve the results.
A small voxel size was initially chosen in order to avoid, as much as possible, the problem of having several spots per voxel. Changing the resolution of the data will come at the cost of a higher number of such voxels. Since several strategies were proposed to deal with this issue, but only one was tested, this could also be a good opportunity to test their impact.

Adding more context information can be done not only by changing the resolution but also by directly changing the patch size. If the patch size were constrained by the actual memory limitations, these constraints might evolve in the future. For now, the patches were larger in the inferior-superior direction in order to encompass a larger portion of the tumor since the dataset contained patients with esophageal cancer. But increasing the patch size in other directions to provide more information along the direction of the beam's trajectory could prove interesting.

As the same network is used to predict the weights of both beams, another idea to ease the prediction could be to rotate all input data in the BEV.
The dataset consisted of 39 patients, which is quite small. Training neural networks often requires a large dataset in order to avoid overfitting and have good generalization capability. Some data augmentation strategies could therefore be tested to increase the size of the dataset. However, a larger dataset comes at the cost of increased resource needs, and it has proven to be hard to manage.

Network

By choosing to represent the spots and their weight with 3D matrices, the result is a representation with many zeros and only a few non-zero values, rendering the data highly sparse. One path of improvement could be to look for networks and data structures that handle sparse data more efficiently and appropriately. Nonetheless, this is challenging as most of the input data is dense, which means that both sparse and dense data must be treated adequately by the same network. Instead of rasterizing the spots into voxel grids, other types of 3D data representation could be considered, such as directly working with point clouds for instance [93] [94]. This could alleviate the problems of voluminous data and quantization artifacts inherent to volumetric data. But once again, the difficulty is adapting those approaches to be able to also feed the network with dense data such as the CT scan or the dose distributions. For instance, in [95], an architec-

ture combining a Graph Neural Network (GNN) and a CNN was employed to evaluate the error in 3D segmentations of OARs. The error is evaluated node-wise by leveraging the connectivity information of the segmentation structure and information provided in sub-volumes of the CT scan centered around the nodes.

Computation time

The time needed to compute the predicted weights of a patient can seem elevated. It is partly due to the large number of patches that are required to reconstitute the entire prediction for all weights. The need for a large number of patches is the consequence of two things. First, it is due to the small size of the voxels. Second, it is related to the value of the stride used when reconstructing the whole prediction matrix. By doubling the stride, the computation time passes from 24 minutes to around 5 minutes for one patient. One could argue that a small stride is not required since the data that is reconstructed is not continuous but sparse. However, it has been observed that increasing the stride for models such as the MSE_{ROI} reduces performance by increasing the number of zero-valued spot weights in the reconstructed full matrix. If the computation time must be decreased, it could still be interesting to assess the impact of increasing the stride for the $MSE_{regularized}$ model. This model predicts fewer zeros and might be less affected by this change. This consideration should be kept in mind if improved models are found in the future.

6.3 Acknowledging method and pipeline limitations

One difficulty of implementing such a pipeline is the use of two different networks in a row. However, since this type of spot weight prediction network is being tested for the first time, it was trained on the ground truth dose per beam distributions. Even if we cannot make the assumption that dose per beam distributions can easily and accurately be predicted by a DNN, the error propagated from one network to the next entails a lack of interpretability of the results. By testing the networks individually, it is possible to get a first idea of how accurate a spot weight network could be, provided that the beam dose predictions are good enough. But if the networks are improved, they should, of course, be tested consecutively by actually providing the output of the U-Net Dose network to the U-Net Weights network. Testing the pipeline as a whole is really important since the error made by the first network could impact the performance of the second network and no longer provide satisfactory end plans. If the results were to be improved, the robustness of the predicted plans should also be assessed. Even if the networks are trained on robust plans, an evaluation of the robustness of the predicted plans should be made to validate the pipeline.

To ease the training and obtain more accurate networks, some constraints were added. First, the dataset was homogeneous in the beam configuration and in the dose prescription. Such constraints reduce the flexibility of the network but often allow for greater accuracy. Second, the pipeline assumes that the number of spots and their location are provided to the U-Net Weights network. Even if these parameters are patient-specific, they are fixed at the beginning of the process and therefore decrease the flexibility of the model. The advantage is that it considerably decreases the search space, making the training more feasible.
For that reason, all these constraints were introduced to reach a trade-off between flexibility and accuracy.

6.4 Perspectives and applications

Given the poor results obtained so far, the search space could be further reduced by using the pipeline for SFUD plans instead of IMPT plans. Single-Field Uniform Dose (SFUD) is another approach to combining fields for PBS treatments. SFUD implies that each field is optimized independently to cover the target uniformly. With IMPT, it is the sum of each field that should result in a uniform target coverage. The different fields of IMPT plans are therefore optimized simultaneously, providing more flexibility but also resulting in individual fields that are no longer homogeneous. It might be easier to predict homogeneous fields, and the pipeline could consequently provide better results for SFUD plans.

Even if the projected values are not accurate enough to be used directly, the pipeline can also be used to provide a first guest for the spot weights. It would still require some further optimization, but it could potentially save some time by reducing the number of iterations needed to produce the final plan.

Chapter 7

Conclusion

The automation of the treatment planning process in radiotherapy has already been the focus of numerous research studies, many of which have implemented solutions based on the use of deep neural networks. However, automating the optimization of spot weights for proton therapy treatments with a deep neural network had never been done previously, to the best of our knowledge. In this thesis, a full end-to-end pipeline was implemented, going from the segmented data to the prediction of spot weights and integrating the corresponding plan into RayStation. This pipeline is composed of two different neural networks.

First, a U-Net predicts the 3D dose distributions of each individual beam from the CT scan of the patient and the masks of the OARs and the CTV. The results that have been obtained so far show some defects. The full dose distributions are not homogeneous enough, with more hot spots than ground truth plans, and the error made in some critical organs, such as the spinal canal, could be reduced. In terms of individual beam distributions, dose is predicted in locations that are too distant from the beam's trajectory. Solutions such as reweighting the loss function to put more emphasis on the error made in each individual beam and adding beam masks to the input information could provide more accurate predictions.

Second, another U-Net network predicts the spot weights. It takes as input the 3D beam dose distribution, the spot locations, as well as anatomical information in the form of the CT scan and masks of OARs and CTV. The spot locations were provided to the network by rasterizing the spots into voxel grids in order to work with volumetric data. The spots are also predicted through this volumetric representation and converted back to their original form as a list of spot coordinates and weights. Several models were tested in this thesis. Especially, the results of three models with different loss functions were presented. Several metrics showed the $MSE_{regularized}$ loss function, which is a modified

MSE including a regularization parameter, to be the most promising one.

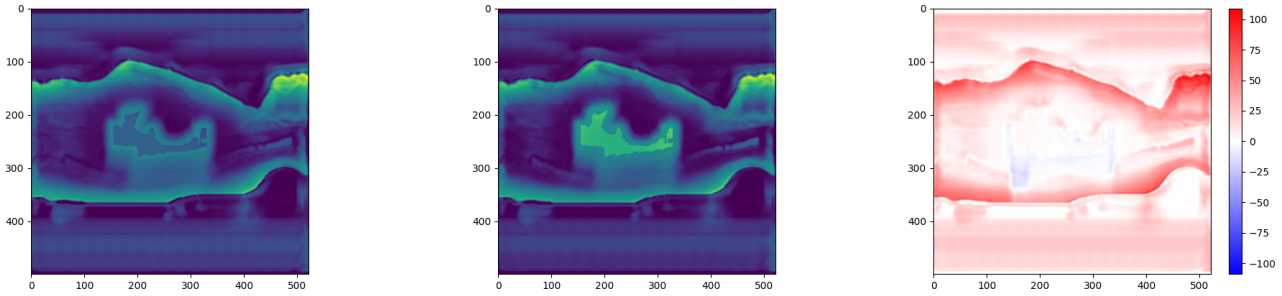
Finally, at the end of the pipeline, a script is used to integrate the predicted weights into the initial non-optimized plan. When observing the dose resulting from the weight predictions, none of the three analyzed models produced satisfactory results. Several ideas to improve the weight prediction were discussed such as adding more context information by changing the resolution of the data, rotating the data in BEV, or even considering other types of representation for the spot data.

This thesis was a first attempt at implementing an accurate end-to-end pipeline for the automation of proton therapy treatment plan generation, and it leaves plenty of room for improvement. Many ideas for improvement that are still to be explored were proposed. Meanwhile, if the weights predicted are not accurate enough to be used directly in the final plan, they could be used as an initial first guest. In this case, further manual optimization would still be needed but could require fewer iterations. This way, the proposed workflow could still save some time during the planning process.

Appendix A

Supplementary results

Dose per beam predictions with 1x1x1 mm^3 voxels and MSE_{total} loss function



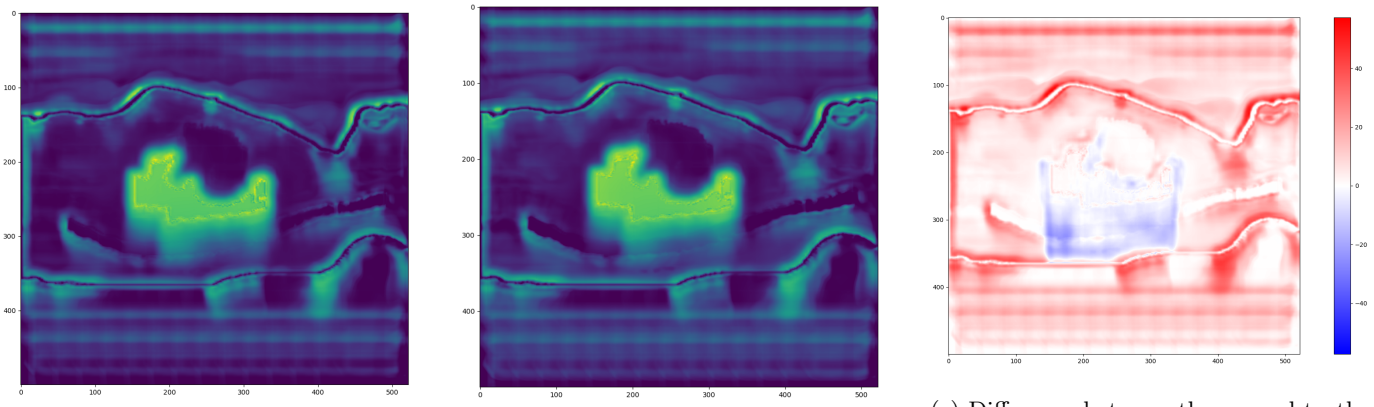
(a) The predicted dose for beam 1

(b) The predicted dose for beam 2

(c) Difference between the ground truth full dose and the sum of the predictions of the two beams [Gy]

Figure A.1: Dose error and dose predictions for patient 45 provided by the U-Net Dose network trained with voxels of resolution 1x1x1 mm^3 and with the MSE_{total} loss function

Dose per beam predictions with 1x1x1 mm^3 voxels and $MSE_{weighted}$ loss function



(a) The predicted dose for beam 1

(b) The predicted dose for beam 2

(c) Difference between the ground truth full dose and the sum of the predictions of the two beams [Gy]

Figure A.2: Dose error and dose predictions for patient 45 provided by the U-Net Dose network trained with voxels of resolution 1x1x1 mm^3 and with the $MSE_{weighted}$ loss function

Bibliography

[1] Jacques Ferlay et al. *Global cancer observatory: cancer tomorrow*. Lyon, France: International Agency for Research on Cancer, 2020. URL: https://gco.iarc.fr/tomorrow. Accessed: 16-07-2023.

[2] Rifat Atun et al. "Expanding global access to radiotherapy". In: *The lancet oncology* 16.10 (2015), pp. 1153–1186. DOI: 10.1016/S1470-2045(15)00222-3.

[3] Antony J Lomax et al. "A treatment planning inter-comparison of proton and intensity modulated photon radiotherapy". In: *Radiotherapy and oncology* 51.3 (1999), pp. 257–271. DOI: 10.1016/S0167-8140(99)00036-5.

[4] Kristin A Higgins et al. "National cancer database analysis of proton versus photon radiation therapy in non-small cell lung cancer". In: *International Journal of Radiation Oncology* Biology* Physics* 97.1 (2017), pp. 128–137. DOI: 10.1016/j.ijrobp.2016.10.001.

[5] Steven H Lin et al. "Randomized phase IIB trial of proton beam therapy versus intensity-modulated radiation therapy for locally advanced esophageal cancer". In: *Journal of Clinical Oncology* 38.14 (2020), p. 1569. DOI: 10.1200/JCO.19.02503.

[6] Xingzhe Li et al. "Toxicity profiles and survival outcomes among patients with nonmetastatic nasopharyngeal carcinoma treated with intensity-modulated proton therapy vs intensity-modulated radiation therapy". In: *JAMA Network Open* 4.6 (2021), e2113205–e2113205. DOI: 10.1001/jamanetworkopen.2021.13205.

[7] S Teoh et al. "Proton vs photon: A model-based approach to patient selection for reduction of cardiac toxicity in locally advanced lung cancer". In: *Radiotherapy and Oncology* 152 (2020), pp. 151–162. DOI: 10.1016/j.radonc.2019.06.032.

[8] *Proton Therapy : Advanced Radiation Therapy Treatment*. URL: https://provisionhealthcare.com/about-proton-therapy/advantages-of-proton/. Accessed: 10-04-2023.

[9] Benjamin E Nelms et al. "Variation in external beam treatment plan quality: an inter-institutional study of planners and planning systems". In: *Practical radiation oncology* 2.4 (2012), pp. 296–305. DOI: 10.1016/j.prro.2011.11.012.

[10] Diyana Afrina Hizam et al. "Evaluation of treatment plan quality for head and neck IMRT: a multicenter study". In: *Medical Dosimetry* 46.3 (2021), pp. 310–317. DOI: 10.1016/j.meddos.2021.03.003.

[11] Elisabetta Cagni et al. "Variations in head and neck treatment plan quality assessment among radiation oncologists and medical physicists in a single radiotherapy department". In: *Frontiers in Oncology* 11 (2021), p. 706034. DOI: 10.3389/fonc.2021.706034.

[12] Radhe Mohan and David Grosshans. "Proton therapy–present and future". In: *Advanced drug delivery reviews* 109 (2017), pp. 26–44. DOI: 10.1016/j.addr.2016.11.006.

[13] Stephanie Lim-Reinders et al. "Online adaptive radiation therapy". In: *International Journal of Radiation Oncology* Biology* Physics* 99.4 (2017), pp. 994–1003. DOI: 10.1016/j.ijrobp.2017.04.023.

[14] Birgit Sabine Müller et al. "Impact of interfractional changes in head and neck cancer patients on the delivered dose in intensity modulated radiotherapy with protons and photons". In: *Physica Medica* 31.3 (2015), pp. 266–272. DOI: 10.1016/j.ejmp.2015.02.007.

[15] Shervin Minaee et al. "Image segmentation using deep learning: A survey". In: *IEEE transactions on pattern analysis and machine intelligence* 44.7 (2021), pp. 3523–3542. DOI: 10.1109/TPAMI.2021.3059968.

[16] Ahmed Hosny et al. "Artificial intelligence in radiology". In: *Nature Reviews Cancer* 18.8 (2018), pp. 500–510. DOI: 10.1038/s41568-018-0016-5.

[17] Stanislav Nikolov et al. *Deep learning to achieve clinically applicable segmentation of head and neck anatomy for radiotherapy*. 2021. arXiv: 1809.04430.

[18] Risheng Wang et al. "Medical image segmentation using deep learning: A survey". In: *IET Image Processing* 16.5 (2022), pp. 1243–1267. DOI: 10.1049/ipr2.12419.

[19] Mohammad Hesam Hesamian et al. "Deep learning techniques for medical image segmentation: achievements and challenges". In: *Journal of digital imaging* 32 (2019), pp. 582–596. DOI: 10.1007/s10278-019-00227-x.

[20] Matthew M. Ladra and Torunn I. Yock. "Proton Radiotherapy for Pediatric Sarcoma". In: *Cancers* 6.1 (2014), pp. 112–127. DOI: 10.3390/cancers6010112.

[21] Harald Paganetti. *Proton therapy physics*. CRC press, 2018.

[22] Jaeman Son et al. “Development of optical fiber based measurement system for the verification of entrance dose map in pencil beam scanning proton beam”. In: *Sensors* 18.1 (2018), p. 227. DOI: 10.3390/s18010227.

[23] Manmadhachary Aiamunoori. “CT imaging parameters for precision models using additive manufacturing”. In: *Multiscale and Multidisciplinary Modeling, Experiments and Design* 2 (Sept. 2019), pp. 1–12. DOI: 10.1007/s41939-019-00046-1.

[24] Thorsten M Buzug. “Computed tomography”. In: *Springer handbook of medical technology*. Springer, 2011, pp. 311–342.

[25] Ervin B Podgorsak. *Radiation oncology physics*. 2005.

[26] Harald Paganetti. “Range uncertainties in proton therapy and the role of Monte Carlo simulations”. In: *Physics in Medicine & Biology* 57.11 (2012), R99. DOI: 10.1088/0031-9155/57/11/R99.

[27] Wei Liu et al. “Robust optimization of intensity modulated proton therapy”. In: *Medical physics* 39.2 (2012), pp. 1079–1091. DOI: 10.1118/1.3679340.

[28] D Pflugfelder, JJ Wilkens, and U Oelfke. “Worst case optimization: a method to account for uncertainties in the optimization of intensity modulated proton therapy”. In: *Physics in Medicine & Biology* 53.6 (2008), p. 1689. DOI: 10.1088/0031-9155/53/6/013.

[29] Risto Miikkulainen et al. “Evolving deep neural networks”. In: *Artificial intelligence in the age of neural networks and brain computing*. Elsevier, 2019, pp. 293–312. DOI: 10.1016/B978-0-12-815480-9.00015-3.

[30] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[31] Athanasios Voulodimos et al. “Deep learning for computer vision: A brief review”. In: *Computational intelligence and neuroscience* 2018 (2018). DOI: 10.1155/2018/7068349.

[32] Hendrik Purwins et al. “Deep learning for audio signal processing”. In: *IEEE Journal of Selected Topics in Signal Processing* 13.2 (2019), pp. 206–219. DOI: 10.1109/JSTSP.2019.2908700.

[33] Daniel W Otter, Julian R Medina, and Jugal K Kalita. “A survey of the usages of deep learning for natural language processing”. In: *IEEE transactions on neural networks and learning systems* 32.2 (2020), pp. 604–624. DOI: 10.1109/TNNLS.2020.2979670.

[34] Kai Fukami, Koji Fukagata, and Kunihiko Taira. “Assessment of supervised machine learning methods for fluid flows”. In: *Theoretical and Computational Fluid Dynamics* 34 (Jan. 2020), pp. 497–519. DOI: 10.1007/s00162-020-00518-y.

[35] Christian Janiesch, Patrick Zschech, and Kai Heinrich. “Machine learning and deep learning”. In: *Electronic Markets* 31.3 (2021), pp. 685–695. DOI: 10.1007/s12525-021-00475-2.

[36] Qi Wang et al. “A comprehensive survey of loss functions in machine learning”. In: *Annals of Data Science* (2020), pp. 1–26. DOI: 10.1007/s40745-020-00253-5.

[37] Samuel L. Smith et al. *Don’t Decay the Learning Rate, Increase the Batch Size*. 2018. arXiv: 1711.00489.

[38] Zewen Li et al. “A survey of convolutional neural networks: analysis, applications, and prospects”. In: *IEEE transactions on neural networks and learning systems* (2021). DOI: 10.1109/TNNLS.2021.3084827.

[39] Andrej Karpathy et al. “Large-scale video classification with convolutional neural networks”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

[40] Ossama Abdel-Hamid et al. “Convolutional neural networks for speech recognition”. In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545. DOI: 10.1109/TASLP.2014.2339736.

[41] Jiuxiang Gu et al. “Recent advances in convolutional neural networks”. In: *Pattern Recognition* 77 (2018), pp. 354–377. DOI: 10.1016/j.patcog.2017.10.013.

[42] Rikiya Yamashita et al. “Convolutional neural networks: an overview and application in radiology”. In: *Insights into Imaging* 9 (June 2018), pp. 611–629. DOI: 10.1007/s13244-018-0639-9.

[43] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. “Deep Sparse Rectifier Neural Networks”. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. PMLR, Nov. 2011, pp. 315–323. URL: https://proceedings.mlr.press/v15/glorot11a.html.

[44] Dingjun Yu et al. “Mixed pooling for convolutional neural networks”. In: *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings 9*. Springer. 2014, pp. 364–375. DOI: 10.1007/978-3-319-11740-9_34.

[45] Sergey Ioffe and Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. In: *International conference on machine learning*. PMLR. 2015, pp. 448–456. URL: https://proceedings.mlr.press/v37/ioffe15.html.

[46] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization*. 2017. arXiv: 1607.08022.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (2015), pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[48] Özgün Çiçek et al. "3D U-Net: learning dense volumetric segmentation from sparse annotation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Springer. 2016, pp. 424–432. DOI: 10.1007/978-3-319-46723-8_49.

[49] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation". In: *2016 fourth international conference on 3D vision (3DV)*. IEEE. 2016, pp. 565–571. DOI: 10.1109/3DV.2016.79.

[50] Shadab Momin et al. "Knowledge-based radiation treatment planning: A data-driven method survey". In: *Journal of Applied Clinical Medical Physics* 22.8 (2021), pp. 16–44. DOI: 10.1002/acm2.13337.

[51] Yang Sheng et al. "Artificial intelligence applications in intensity modulated radiation treatment planning: an overview". In: *Quantitative Imaging in Medicine and Surgery* 11.12 (2021). DOI: 10.21037/qims-21-208.

[52] Dan Nguyen et al. "Advances in Automated Treatment Planning". In: *Seminars in Radiation Oncology* 32.4 (2022), pp. 343–350. DOI: 10.1016/j.semradonc.2022.06.004.

[53] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[54] Xinyuan Chen et al. "A feasibility study on an automated method to generate patient-specific dose distributions for radiotherapy using deep learning". In: *Medical physics* 46.1 (2019), pp. 56–64. DOI: 10.1002/mp.13262.

[55] Jiawei Fan et al. "Automatic treatment planning based on three-dimensional dose distribution predicted from deep learning technique". In: *Medical physics* 46.1 (2019), pp. 370–381. DOI: 10.1002/mp.13271.

[56] Dan Nguyen et al. *A feasibility study for predicting optimal radiation therapy dose distributions of prostate cancer patients from patient anatomy using deep learning*. 2018. arXiv: 1709.09233.

[57] Tomohiro Kajikawa et al. "A convolutional neural network approach for IMRT dose distribution prediction in prostate cancer patients". In: *Journal of radiation research* 60.5 (2019), pp. 685–693. DOI: 10.1093/jrr/rrz051.

[58] Siri Willems et al. "Feasibility of ct-only 3d dose prediction for vmat prostate plans using deep learning". In: *Artificial Intelligence in Radiation Therapy: First International Workshop, AIRT 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 17, 2019, Proceedings 1*. Springer. 2019, pp. 10–17. DOI: 10.1007/978-3-030-32486-5_2.

[59] Yuhei Koike et al. "Patient-specific three-dimensional dose distribution prediction via deep learning for prostate cancer therapy: Improvement with the structure loss". In: *Physica Medica* 107 (2023), p. 102544. DOI: 10.1016/j.ejmp.2023.102544.

[60] Mary P Gronberg et al. “Dose prediction for head and neck radiotherapy using a three-dimensional dense dilated U-net architecture”. In: *Medical physics* 48.9 (2021), pp. 5567–5573. DOI: 10.1002/mp.14827.

[61] Jieping Zhou et al. “A method of using deep learning to predict three-dimensional dose distributions for intensity-modulated radiotherapy of rectal cancer”. In: *Journal of applied clinical medical physics* 21.5 (2020), pp. 26–37. DOI: 10.1002/acm2.12849.

[62] Vasant Kearney et al. “DoseNet: a volumetric dose prediction algorithm using 3D fully-convolutional neural networks”. In: *Physics in Medicine & Biology* 63.23 (2018), p. 235022. DOI: 10.1088/1361-6560/aaef74.

[63] Yan Shao et al. “Prediction of three-dimensional radiotherapy optimal dose distributions for lung cancer patients with asymmetric network”. In: *IEEE Journal of Biomedical and Health Informatics* 25.4 (2020), pp. 1120–1127. DOI: 10.1109/JBHI.2020.3025712.

[64] Dan Nguyen et al. “3D radiotherapy dose prediction on head and neck cancer patients with a hierarchically densely connected U-net deep learning architecture”. In: *Physics in medicine & Biology* 64.6 (2019), p. 065020. DOI: 10.1088/1361-6560/ab039b.

[65] Gao Huang et al. “Densely connected convolutional networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[66] Ana Maria Barragán-Montero et al. “Three-dimensional dose prediction for lung IMRT patients with deep neural networks: robust learning from heterogeneous beam configurations”. In: *Medical physics* 46.8 (2019), pp. 3679–3691. DOI: 10.1002/mp.13597.

[67] May Teoh et al. “Volumetric modulated arc therapy: a review of current literature and clinical use in practice”. In: *The British journal of radiology* 84.1007 (2011), pp. 967–996. DOI: 10.1259/bjr/22373346.

[68] F Guerreiro et al. “Deep learning prediction of proton and photon dose distributions for paediatric abdominal tumours”. In: *Radiotherapy and Oncology* 156 (2021), pp. 36–42. DOI: 10.1016/j.radonc.2020.11.026.

[69] Oskar Eriksson and Tianfang Zhang. “Robust automated radiation therapy treatment planning using scenario-specific dose prediction and robust dose mimicking”. In: *Medical Physics* 49.6 (2022), pp. 3564–3573. DOI: 10.1002/mp.15622.

[70] Elena Borderias-Villarroel et al. “Machine learning-based automatic proton therapy planning: Impact of post-processing and dose-mimicking in plan robustness”. In: *Medical Physics* (2023). DOI: 10.1002/mp.16408.

[71] Margerie Huet-Dastarac et al. “Patient selection for proton therapy using Normal Tissue Complication Probability with deep learning dose prediction for oropharyngeal cancer”. In: *Medical Physics* (2023). DOI: 10.1002/mp.16431.

[72] Camille Draguet et al. “Automated clinical decision support system with deep learning dose prediction and NTCP models to evaluate treatment complications in patients with esophageal cancer”. In: *Radiotherapy and Oncology* 176 (2022), pp. 101–107. DOI: 10.1016/j.radonc.2022.08.031.

[73] Ian Goodfellow et al. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[74] Jiajun Wu et al. “Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling”. In: *Advances in neural information processing systems* 29 (2016). URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/44f683a84163b3523afe57c2e008bc8c-Paper.pdf.

[75] Yizhou Chen et al. “Generative Adversarial Networks in Medical Image augmentation: A review”. In: *Computers in Biology and Medicine* 144 (2022), p. 105382. DOI: 10.1016/j.compbiomed.2022.105382.

[76] Nripendra Kumar Singh and Khalid Raza. “Medical image generation using generative adversarial networks: A review”. In: *Health informatics: A computational perspective in healthcare* (2021), pp. 77–96. DOI: 10.1007/978-981-15-9735-0_5.

[77] Rafid Mahmood et al. “Automated treatment planning in radiation therapy using generative adversarial networks”. In: *Machine learning for healthcare conference*. PMLR. 2018, pp. 484–499. URL: https://proceedings.mlr.press/v85/mahmood18a.html.

[78] Aaron Babier et al. “Knowledge-based automated planning with three-dimensional generative adversarial networks”. In: *Medical Physics* 47.2 (2020), pp. 297–306. DOI: 10.1002/mp.13896.

[79] Vasant Kearney et al. “DoseGAN: a generative adversarial network for synthetic dose prediction using attention-gated discrimination and generation”. In: *Scientific reports* 10.1 (2020), p. 11073. DOI: 10.1038/s41598-020-68062-7.

[80] Phillip Isola et al. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5967–5976. DOI: 10.1109/CVPR.2017.632.

[81] Lin Ma et al. “Deep learning-based inverse mapping for fluence map prediction”. In: *Physics in Medicine Biology* 65.23 (Nov. 2020), p. 235035. DOI: 10.1088/1361-6560/abc12c.

[82] Zengtai Yuan et al. “Accelerate treatment planning process using deep learning generated fluence maps for cervical cancer radiation therapy”. In: *Medical Physics* 49.4 (2022), pp. 2631–2641. DOI: 10.1002/mp.15530.

[83] Xinyi Li et al. “Automatic IMRT planning via static field fluence prediction (AIP-SFFP): a deep learning algorithm for real-time prostate treatment planning”. In: *Physics in Medicine Biology* 65.17 (Aug. 2020), p. 175014. DOI: 10.1088/1361-6560/aba5eb.

[84] Xinyi Li et al. “An artificial intelligence-driven agent for real-time head-and-neck IMRT plan generation using conditional generative adversarial network (cGAN)”. In: *Medical Physics* 48.6 (2021), pp. 2714–2723. DOI: 10.1002/mp.14770.

[85] Hoyeon Lee et al. “Fluence-map generation for prostate intensity-modulated radiotherapy planning using a deep-neural-network”. In: *Scientific Reports* 9 (Oct. 2019). DOI: 10.1038/s41598-019-52262-x.

[86] Wentao Wang et al. “Fluence Map Prediction Using Deep Learning Models – Direct Plan Generation for Pancreas Stereotactic Body Radiation Therapy”. In: *Frontiers in Artificial Intelligence* 3 (Sept. 2020), p. 68. DOI: 10.3389/frai.2020.00068.

[87] Wentao Wang et al. “Deep Learning–Based Fluence Map Prediction for Pancreas Stereotactic Body Radiation Therapy With Simultaneous Integrated Boost”. In: *Advances in Radiation Oncology* 6 (Feb. 2021), p. 100672. DOI: 10.1016/j.adro.2021.100672.

[88] Liesbeth Vandewinckele et al. “Treatment plan prediction for lung IMRT using deep learning based fluence map generation”. In: *Physica Medica* 99 (2022), pp. 44–54. DOI: 10.1016/j.ejmp.2022.05.008.

[89] Guoliang Zhang et al. “SWFT-Net: a deep learning framework for efficient fine-tuning spot weights towards adaptive proton therapy”. In: *Physics in Medicine & Biology* 67.24 (2022), p. 245010. DOI: 10.1088/1361-6560/aca517.

[90] Kaiming He et al. “Deep residual learning for image recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016, pp. 770–778.

[91] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980.

[92] *toch.utils.data*. PyTorch 2.0 documentation. URL: https://pytorch.org/docs/stable/data.html. Accessed: 08-08-2023.

[93] Eman Ahmed et al. *A survey on Deep Learning Advances on Different 3D Data Representations*. 2019. arXiv: 1808.01462.

[94] Yulan Guo et al. “Deep learning for 3d point clouds: A survey”. In: *IEEE transactions on pattern analysis and machine intelligence* 43.12 (2020), pp. 4338–4364. DOI: 10.1109/TPAMI.2020.3005434.

[95] Edward GA Henderson et al. “Automatic identification of segmentation errors for radiotherapy using geometric learning”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2022, pp. 319–329. DOI: 10.1007/978-3-031-16443-9_31.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain
Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/epl**