



**Hochschule Darmstadt**

**Department of Natural Sciences and Mathematics**
**&**
**Information Science and Computer Science**

# Artificial intelligence guided positioning of active voxels in particle therapy treatment planning

Thesis for the award of the academic degree

Master of Science (M. Sc.)

in the study program Data Science

submitted by:

Anastasiia Quarz

First supervisor: Prof. Dr. Jutta Groos, Department of Mathematics

Second supervisor: Prof. Dr. Elke Hergenröther, Department of Computer Sciences

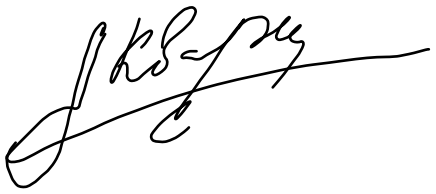
Issue date: 27/09/2021

Submission date: 11/04/2022

**Erklärung**

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Viernheim, den 11.04.2022



## ABSTRACT

Heavy ion beam therapy is a highly effective cancer treatment. Unlike conventional X-ray therapy, the dose deposited per unit length by charged particles increases toward the end of the range in tissue (Bragg peak), and effect in the brag peak is further enhanced by the increased relative biological effectiveness (RBE) of slow particles. The treatment planning in heavy ion therapy optimizes treatment fields to maximize the RBE-weighted dose (calculated in every sub-volume, voxel) to the target while minimizing the dose to the surrounding organs at risk (OAR). Because of the complex physics and of the correction for the variable RBE, treatment plan calculation can be long and requires powerful computers. Moreover, it is affected by numerous uncertainties (e.g., patients positioning or anatomical changes) that require computation of many different scenarios (robust optimization).

The aim of this work is to accelerate treatment planning through a deep learning algorithm that reduces the number of voxels considered in the optimization without reduction of the plan quality. While previous attempts used a random sampling algorithm, here we implemented a convolutional neuronal network (CNN) based on P-Net architecture, with a loss function penalizing a high number of selected voxels and target underdosage. Training on the small database (30 patients) showed stability of the results during the testing (20 patient). Target coverage with a lower number of involved voxels than in the random sampling algorithm was achieved consistently in three independent runs for 20 epochs.

More epochs seem to be necessary for the algorithm to converge towards a stationary optimal value. This was beyond the scope of this work, as computation time per epoch was around 7-9 hours. A GPU-based implementation of the CNN and the dose calculation could greatly facilitate this goal. The results are a promising step toward a full CNN selection of the critical voxels to minimize computation of robust plans. A larger dataset and increased computation time is needed to assess if critical properties of selected voxels exist and whether there is a minimum number and location of selected voxels whilst still maintaining target coverage in patients.


---


**Keywords**—*radiotherapy, optimisation, artificial intelligence, treatment planning system*

## ABSTRACT (GERMAN)

Die Ionenstrahltherapie stellt eine hocheffektive Art der Tumorbekämpfung dar. Im Gegensatz zu konventioneller Photonenstrahlung nimmt ihre Dosisabgabe gegen Ende ihrer Reichweite zu (Bragg Peak), wo zusätzlich ihr Effekt durch die hohe relative biologische Wirksamkeit (RBW) langsamer Ionen verstärkt wird. In der Bestrahlungsplanung wird die RBW-gewichtete Dosis im Ziel maximiert (in jedem Volumenelement, Voxel), während sie in umliegende Risikoorganen (OAR) reduziert wird. Aufgrund der komplexen physikalischen und biologischen Wirkung dauert die Optimierung von Plänen lange und erfordert eine hohe Rechenleistung. Überdies unterliegt die Therapie Unsicherheiten wie beispielsweise der Patientenpositionierung, die in zusätzlichen Szenarien berücksichtigt werden müssen (Robuste Optimierung).

Das Ziel dieser Arbeit ist es, die Bestrahlungsplanung mit Hilfe Künstlicher Intelligenz zu beschleunigen, indem die Anzahl der in der Optimierung berücksichtigten Voxel reduziert wird, ohne dabei die Planqualität zu reduzieren. Existierende Ansätze basierten auf einem zufälligen Sampling, während hier ein Neuronales Netzwerk (CNN) mit P-Netz Architektur verwendet wird. Eine Kostenfunktion bestraft eine hohe Anzahl von Voxeln sowie eine Unterdosierung des Ziels. Ein Training auf einem kleinen Datensatz aus 30 Patienten zeigte in einem Testdatensatz (20 Patienten) stabile Ergebnisse. Die erforderliche Dosis im Ziel wurde in 3 unabhängigen Experimenten mit 20 Epochen konsistent erreicht, wobei weniger Voxel als in der zufälligen Auswahl verwendet wurden.

Mehr Epochen scheinen trotzdem notwendig zu sein um Konvergenz zu erzielen, waren aber im Rahmen dieser Arbeit aus Zeitgründen nicht möglich, da jede Epoche 7-9h benötigte. Eine GPU-basierte Implementierung des CNN und der Planoptimierung würde die Berechnung wesentlich beschleunigen. Insgesamt wurde ein wesentlicher Schritt zur Implementierung vollständiger CNN erreicht, die kritische Voxel für eine robuste Optimierung auswählen können. Auf einem größeren Datensatz mit mehr Rechenzeit kann abschließend beantwortet werden, ob sich kritische Eigenschaften der ausgewählten Voxel identifizieren lassen und ob es eine optimale Anzahl und Lage der Voxel gibt, mit denen eine adäquate Dosisabdeckung im Patienten garantiert werden kann.


---


**Keywords**—*radiotherapy, optimisation, artificial intelligence, treatment planning system*

**ACKNOWLEDGEMENTS**

Firstly, I would like to thank Professor Durante for opening the world of biophysics to me, where everything I am interested in is blended. I am also grateful to Professor Groos for showing me that statistics can be not only scaring, but also beautiful, Professor Hergenröther for giving me first knowledge of computer vision and deep learning. Without the combination of these fields of knowledge, I would never be brave enough to start my carrier in biophysics.

Moreover, I would like to express my gratitude to Professor Graeff for his patience and being supportive all the time, when implementing parts of the project by my request and careful listening to thousands of new ideas regularly. He gave me a lot of feedback and hints, so that I could look next for possible solutions for my master thesis. And a special thanks to the whole biophysics department at GSI for broadening my knowledge and helping me to overcome the difficulties caused by many facets of the field.

Last but not least, I am thankful to my mother Elena, husband Jochen, daughter Emilia and the whole family for their patience and support. It was a long way to the final stage of my scientific project we had to go through together.

## TABLE OF CONTENTS


**INTRODUCTION** ..... **19**

**1. PARTICLE RADIOTHERAPY** ..... **22**

1.1. PHYSICS OF CHARGED PARTICLE TRANSPORT (CPT) ..... 23
1.2. BEAM DELIVERY ..... 25
1.3 BIOLOGICALLY WEIGHTED DOSE ..... 26
1.4 TREATMENT PLANNING ..... 28
1.5 TRIP98 ..... 30

**2 ARTIFICIAL INTELLIGENCE IN MEDICINE** ..... **33**

2.3 BASIC CONCEPTS OF CONVOLUTIONAL NEURAL NETWORKS ..... 34
2.4 AI IN PARTICLE THERAPY ..... 38

**3 PREVIOUS METHOD. SELECTION OF THE VOXELS BY RANDOM MASK [13]** ..... **42**

**4 DATA** ..... **44**

**5. CURRENT METHOD** ..... **48**

5.1 ARCHITECTURE ..... 50
*5.1.1 Concatenation of the feature maps* ..... *51*
*5.1.2 Dilation* ..... *52*
*5.1.3 Loss function* ..... *54*
5.2 EVALUATION METRICS ..... 58
5.3 EVALUATION METHODS ..... 60

**6. IMPLEMENTATION** ..... **63**

6.1 HARDWARE ..... 63
6.2 SOFTWARE ..... 63
*6.2.1 SSH* ..... *64*
*6.2.2 TRiP98* ..... *64*
*6.2.3 Communication between TRiP98 and Python: FIFO* ..... *67*
*6.2.4 Python environment* ..... *67*
6.3 PIPELINE ..... 67
*6.3.1 Epoch details* ..... *67*
*6.3.2 System pretraining* ..... *68*
*6.3.3 System training* ..... *68*

**7. RESULTS** ..... **72**

**8. DISCUSSION** ..... **93**

**9. OUTLOOK** ..... **96**

**10. APPENDICES** ..... **98**

10.1 PYTHON ENVIRONMENT ..... 98
10.2 ADDITIONAL MATERIAL ..... 98

**11. BIBLIOGRAPHY** ..... **102**

*LIST OF ABBREVIATIONS*

| | |
|---|---|
| 3D | three-dimensional |
| 4D | four-dimensional |
| A3C | Asynchronous Advantage Actor Critic |
| AI | artificial intelligence |
| CNN | convolutional neural network |
| CPT | charged particle transport |
| CSDA | continuous slowing down approximation |
| CT | computed tomography |
| CTV | clinical target volume |
| DVH | dose-volume histogram |
| GTV | gross tumour volume |
| ICRU | International Commission on Radiological Units |
| IMRT | intensity modulated radiotherapy |
| FM | full mask |
| LET | linear energy transfer |
| LQ | linear quadratic |
| MS | Monte Carlo |
| NN | neural network |
| NTCP | normal tissue complication probability |
| OAR | organ at risk |
| PB | pencil-beam |
| RM | random mask |
| PTV | planning target volume |
| RBE | relative biological effectiveness |
| ReLu | Rectified Linear Unit |
| RL | reinforcement learning |
| SBRT | Stereotactic Body Radiation Therapy |
| SPR | stopping power ratio |
| SV | selected voxels |
| TCP | tumour control probability |
| TPS | treatment planning system |
| TRiP98 | a research software for Treatment Planning for Particles |
| VMAT | volumetric modulated arc therapy |
| VOI | volume of interest |

*LIST OF FIGURES*

*Figure 0.1 Radiation therapy workflow and expected artificial intelligence applications in it. The workflow starts with a decision that the patient needs the radiation therapy, then the optimal parameter of imaging to gain quality and reduce radiation is chosen, followed by preparing the data for the treatment planning (segmentation). After the treatment planning is created, it should be approved with quality assurance (QA). After all the steps are done, AI is supposed to help with online therapy by controlling the position, motion of the patient, adjusting the plan accordingly, to deliver the best possible result. At the end toxicity and other measurements needed for a follow-up should be foreseen [10]. ___ 19*

*Figure 1.1 Physical advantages of particle therapy. (A) Depth-dose distributions of high-energy X-rays and monoenergetic beams of protons or carbon ions. At the same range, C-ions have lower straggling than protons, but a tail of fragments is visible beyond the Bragg peak. In clinical applications, the Bragg peak must be extended to cover all the tumour (B). This can be done by overlapping different pristine beams at different energy and intensity. Figures from GSI Helmholtz Centre library. _____ 22*

*Figure 1.2 Charged particles produce a reduced integral dose to normal tissue. Treatment of a lung cancer by stereotactic body radiation therapy (left) or carbon ions (right). Only 3 beams can be used with particles, as also shown in the 3D image below. Plans from Krjstian Anderle, Ph.D. thesis, Technical University of Darmstadt 2015. _____ 23*

*Figure 1.3 Pencil beam scanning concept. Figures from GSI Helmholtz Centre library. __ 25*

*Figure 1.4 Cell survival of hamster cells exposed to X-rays or heavy ions of different LET. CHO-K1 is the wild-type, radioresistant strain; xrs-5 is a DNA-repair deficient mutated cell line. RBE is large and LET dependent in the radio resistant strain but approximately 1 for radiosensitive cells. [36] _____ 27*

*Figure 1.5 Schematic of the target volume (TV) strategy in radiotherapy. The macroscopically visible tumour forms the gross tumour volume (GTV), with an extension to treat possible microscopic disease in the clinical target volume (CTV). A further extension to the planning target volume (PTV) is supposed to cover uncertainties in delivery to ensure full target dose in the CTV. Often, the target extensions overlap with organ at risk (OAR), creating a conflict in the planning targets, located in the irradiated volume.[37] ____ 28*

*Figure 1.6 Example of DVH for robust plan optimization. Multiple lines represent different scenarios use for the optimization of the plan. Red lines mark 95% of the scales. Figures made by Moritz Wolf from GSI Helmholtz Centre. _____ 29*

*Figure 2.1 Subfields of AI. [47] _____ 33*

*Figure 2.2 Scheme of feedforward artificial neuronal network with error back propagation [57] _____ 35*

*Figure 2.3 Example of value calculation in a convolutional layer [58] _____ 36*

*Figure 2.4 Example of the effect of different filters [60] _____ 37*

*Figure 2.5 Example of dilation rate parameter [46] _____ 37*

*Figure 2.6 Development of prostate cancer radiotherapy 1935–2010. It demonstrates the efficiency of different techniques to deliver the dose to the prostate without compromising the healthy tissues. The cold colours (like blue) represent low dose and the red, dark red the high dose region. Abbreviations: 3D-CRT, 3D conformal radiotherapy; IMRT, intensity modulated radiotherapy; RT, radiotherapy. [63] _____ 39*

*Figure 2.7 Radiation therapy workflow and expected artificial intelligence applications in it. The workflow starts with a decision that the patient needs the radiation therapy, then the optimal parameter of imaging to gain quality and reduce radiation is chosen, followed by preparing the data for the treatment planning (segmentation). After the treatment planning is created, it should be approved with quality assurance (QA). After all the steps are done, AI is supposed to help with online therapy by controlling the position, motion of the patient, adjusting the plan accordingly, to deliver the best possible result. At the end toxicity and other measurements needed for a follow-up should be foreseen [10]. ___ 39*

*Figure 3.1 Average D99 values from RA for constant boundary parameter $x_B$=1 and varying both $x_i$ and boundary widths* 42
*Figure 3.2 Randomly sampled voxels assigned for optimization (a). Resulting nominal dose distribution (b)[13]* 43
*Figure 4.1 3D view of the segmented organs and the tumour in a patient from the database* 45
*Figure 4.2 Relative frequency of OARs per patient in the database. The median number of OARs in the whole dataset is shown with the dashed red line.* 45
*Figure 4.3 Distribution of the number of voxels in target* 45
*Figure 4.4 Example of a mask (left) and distance map (right). The colours of the voxels show the distance (the whiter the higher the distance). The contour shows the original VOIs, used in the plan.* 46
*Figure 4.5 Relative frequency of voxels with different distance* 47
*Figure 5.1 P-Net architecture for 3D image input. The parameters of the convolution layers (dark blue rectangles) are kernel size, output channels, dilation. The information from Blocks 1-5 is concatenated before Block 6.* 50
*Figure 5.2 Final architecture used in the thesis. The parameters of the convolution layers (dark blue and blue rectangles) are kernel size, output channels, dilation. The information from Blocks 1-5 is concatenated before Block 6.* 51
*Figure 5.3 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified binary mask* 53
*Figure 5.4 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified binary mask applied in 2 layers in a row* 53
*Figure 5.5 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified randomly sampled binary mask applied in 2 layers in a row* 54
*Figure 5.6 Loss function for D95≥95% (top), D95<95% (bottom).* 58
*Figure 5.7 Example of accuracy applied to an image in segmentation task [75]* 59
*Figure 5.8 Relative dose vs percentage of selected voxels in target. The data points are medians of the datasets. The numbers represent the sampling parameter in the interior. Semi-transparent labels denote the datapoints used for the training dataset.* 60
*Figure 6.1 Representation of the communication between TRiP98 and Python environment* 64
*Figure 6.2 Flowchart of the processes in TRIP98* 64
*Figure 6.3 Example of 3D representation of CT data (left) and segmented organs (right)* 65
*Figure 6.4 Example of an optimized plan (left) and refined binary mask (right)* 66
*Figure 6.5 Dataflow of interaction between TRiP98 and Python* 69
*Figure 7.1 Loss (top) and percentage of SV out of VOI (misselected voxels) (bottom) in the pretraining stage* 72
*Figure 7.2 Loss (top) and percentage of misselected voxels (bottom) for Experiment 0* 73
*Figure 7.3 Distribution of the SV (top) and D95 (bottom) value in epochs, full mask (FM) and random mask (RM) in Experiment 0* 74
*Figure 7.4 History of the values produced during the training for patient CBS303* 75
*Figure 7.5 History of the values produced during the training for patient CFW221* 75
*Figure 7.6 Density plot of distribution to select a voxel for CBS303 (top) and CFW221 (bottom)* 76
*Figure 7.7 Distribution of probabilities in different bins for epoch 6 (top) and 20 (bottom)* 77
*Figure 7.8 Loss in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)* 78
*Figure 7.9 Percentage of SV out of VOIs (misselected voxels) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)* 79
*Figure 7.10 Dynamic of the median D95 during the training* 80
*Figure 7.11 Distribution of the D95 value in epochs, full mask (FM) and random mask (RM) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)* 81
*Figure 7.12 Dynamics of the median percentage of the selected voxels in the target during the training* 82

*Figure 7.13 Distribution of the percentage of the selected voxels in the target in epochs, full mask (FM) and random mask (RM) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom). Red dashed line is the median of random mask medians for both datasets.* 83
*Figure 7.14 Representation of the median percentage of the selected voxels (X-axis) vs the median of D95 (Y-axis) for each epoch in Experiments 1-3. The grey line indicates the threshold of 95% for D95.* 84
*Figure 7.15 Density of probabilities to select voxel in epochs 6 (top), 12 (middle), 18 (bottom)* 86
*Figure 7.16 Examples of all 3 types of mask behaviour for a patient (1-vertical, 2-horizontal, 3-curved)* 88
*Figure 7.17 Visualisation of the dose maps for the vertical change in the mask for the patient CUM330 (D95, SV). The colour scale on the left represents the dose and the white spots are SV. Black contour-target, orange one -brainstem (Produced with Slicer 8).* 88
*Figure 7.18 Visualisation of the dose maps in different steps of the training for the patient CBP381 (D95, SV). The colour scale on the left represents the dose and the white spots are SV. Black contour - target, orange - eyes, red - brainstem (Produced with Slicer 8). RM stays for random mask.* 90
*Figure 7.19 Comparison of the random mask (left) with the mask, produced with our CNN after epoch 6 (right) slice 16 (top), 17 (middle) and 18 (bottom) for CZW434 patient. The colour scale on left represents the dose and the white spots are SV. Black contour-target, dark blue -optic nerve, orange -eyes, red -brainstem (Produced with Slicer 8)* 91
*Figure 9.1 Illustration of the architecture used in [8], the numbers on top of each block are dimensions.* 97
*Figure 10.1 Relative frequency of OARs in the dataset* 98
*Figure 10.2 IteR-MRL architecture. At each step it gets the probability of the previous step and interaction map with regions which are either should be selected or removed from the output mask.* 99
*Figure 10.3 Example of TRiP98 log file* 99
*Figure 10.4 Loss in pretraining stage with full mask as label* 99
*Figure 10.5 D95 difference between random mask and the mask produced with AI for each patient.* 100
*Figure 10.6 History of changing number of voxels in the tumour during the training for experiment 1-3 (top-bottom)* 101

*LIST OF TABLES*

*Table 1 Comparison of the percentage of SV and D95 for patient CBP381 in different steps of the training* _____ *89*
*Table 2 Comparison of values for full, random and AI mask for the CZW434 patient* _____ *92*

# Introduction

The influence of Artificial Intelligence (AI) is growing in many fields of human activity and in the medical field in particular. Different areas of medicine can benefit from techniques such as convolutional neural networks (CNN). The possible increase in the degree of automation and thereby faster workflow of analysis and diagnostics play a key role in cancer, where any delay plays against the patient chance to be optimally cured. AI is used already diagnostics [1], [2], prognostics [3], [4], segmentation [5]–[7], and treatment planning system (TPS) [8].

In cancer treatment, a combination of surgery, chemotherapy and radiotherapy is usually applied [9]. Radiotherapy offers a non-invasive method to kill the tumour cells and spare its surrounded normal tissue using high-energy rays (X-rays) or particles. Radiotherapy can be applied after surgery to prevent tumour recurrence, caused by residual cancer stem cells (adjuvant treatment). For instance, in breast cancer radiotherapy patients are routinely performed following quadrantectomy. For patients with large tumours, radiotherapy is given before surgery (neoadjuvant) to shrink the tumour, which will be easier to remove. Finally, in other cases definitive radiotherapy is given with curative intent as a replacement of surgery, possibly in combination with chemotherapy or targeted therapy.

Figure 0.1 shows the radiation therapy workflow and some of possible AI applications.

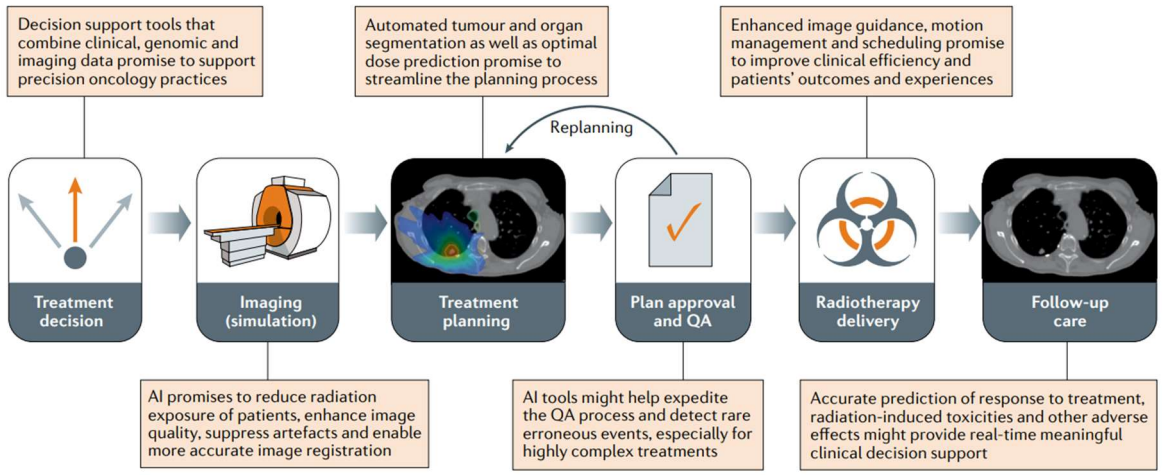



*Figure 0.1 Radiation therapy workflow and expected artificial intelligence applications in it. The workflow starts with a decision that the patient needs the radiation therapy, then the optimal parameter of imaging to gain quality and reduce radiation is chosen, followed by preparing the data for the treatment planning (segmentation). After the treatment planning is created, it should be approved with quality assurance (QA). After all the steps are done, AI is supposed to help with online therapy by controlling the position, motion of the patient, adjusting the plan accordingly, to deliver the best possible result. At the end toxicity and other measurements needed for a follow-up should be foreseen [10].*

Even though cancer diagnostics and image segmentation are now often exploiting AI techniques, AI applications in treatment planning are still limited.

The goal of the radiotherapy is to deliver a prescribed dose to the tumour with minimal possible damage of the normal tissue. Sometimes tumours are located very close to critical organs (organ at risk, OARs), increasing the risk of severe morbidity. Particle therapy exploits the special physical properties of charged particles and allows a much more precise dose deposition at a pre-defined depth. With charged particles, normal tissue is spared much better than with X-rays, thus theoretically reducing the toxicity. Treatment planning calculates the optimal parameters to irradiate the tumour (target) and spare the neighbouring organs. Variable beam parameters including angle, intensity, range, and beam size.

In this work, we apply deep learning within the particle therapy treatment planning system to turn down the computational time by reducing the number of voxels involved into the plan calculation. In fact, notwithstanding the benefits of the particle therapy, it is very expensive and treatment planning is time consuming. Yet very few investigations about the applications of machine learning in heavy ion therapy have been performed worldwide. AI in treatment planning may substantially reduce the time for treatment plan thus improving patient workflow and reducing the delay between diagnosis and treatment.

Not only complete task as segmentation or dose calculation can be taken as the basis for AI applications. For example, the TPS TRiP98 developed at GSI by Dr. Michael Krämer [11], [12] and currently used as basis for Siemens Syngo TPS for treatment with carbon ions, goes through multiple stages before the final optimization and can be very time consuming when robust optimization is used. Reduction of the computational time can be achieved through preprocessing of the input data at the optimization stage. One excellent solution is the selection of only those voxels (**vo**lumetric pi**xel**) that are highly relevant for the final result. It reduces the resolution of the dose correlation matrix and the time needed to optimize the number of particles which should be delivered in each voxel. Assuming that a homogeneous dose at the target boundary implies a homogeneous dose inside the target, the random algorithm [13] uses all voxels within a user-defined distance to the target boundary, but only a fraction of randomly selected voxels in the interior (random mask).

In this work we selected the voxels using a CNN algorithm instead of the previous random sampling. The current random sampling method forces users to input the two parameters (boundary shell and interior probability) manually, check the DVH and, if necessary, adjust the variables. The new system should be able to deliver a stable dose distribution independently of

the size of the tumour, a precise selection of the voxels in the tumour volume and the best possible optimization result. With this aim, we have built a dedicated loss function and trained the system on the treatment plans of the patients treated with GSI during the pilot trial 1997-2007 for head-and-neck malignancies, mostly chordomas or chondrosarcomas of the skull base.

In first chapter we provide some information on particle therapy, such as physics of the process, beam delivery and the treatment planning itself, needed for the further understanding of the work. Second chapter provide an overview of use of artificial intelligence in medicine and some basics of convolutional neural networks. It is followed by the description of the current method used for reduction of computational power and time, spent on the optimisation of the fluence. In chapter 4, the information about data, test and train datasets is provided. In chapter 5, we introduce the used method for position and selection of the voxels. Detailed information about the architecture, its benefits as well as evaluation metrics and methods. Chapter 6 provides detailed information on implementation. Chapter 7 shows the received results, which are discussed in Chapter 8. Chapter 9 is an outlook for further research.

# 1. Particle radiotherapy

Radiotherapy is an essential component of cancer therapy. The combination of surgery, chemotherapy and radiotherapy is becoming a standard for most cancer patients. Out of the approximately 2/3 of cancer patients receiving radiotherapy, over 80% are irradiated with X-rays produced at linear electron accelerators (Linacs). The goal of external radiotherapy (teletherapy) is delivering a dose as high as possible to the tumour (target volume) minimizing the dose to the normal tissue, especially to the organs at risk (OAR) close to the target. The most advanced X-ray delivery techniques is intensity modulated radiotherapy (IMRT), an advanced form of 3D conformal X-ray therapy, where the intensity of the beams is modulated to achieve a higher degree of conformality of the resulting dose distribution within the tumour target volume. The intensity of the beams that cross sensitive organs is reduced, while the intensity of those beams that see primarily the target is increased. The resulting inhomogeneities (hot/cold spots) are compensated by the beams coming from different directions. The other patients receive specialized treatments such as gamma knife or brachytherapy. Only about 0.8% of the radiotherapy patients are treated with high-energy charged particles, but their number is rapidly increasing. The rationale for using accelerated ions in therapy comes from the depth-dose distribution (Figure 1.1), and was originally proposed by Robert Wilson [14] at the University of California in Berkeley (CA, USA).

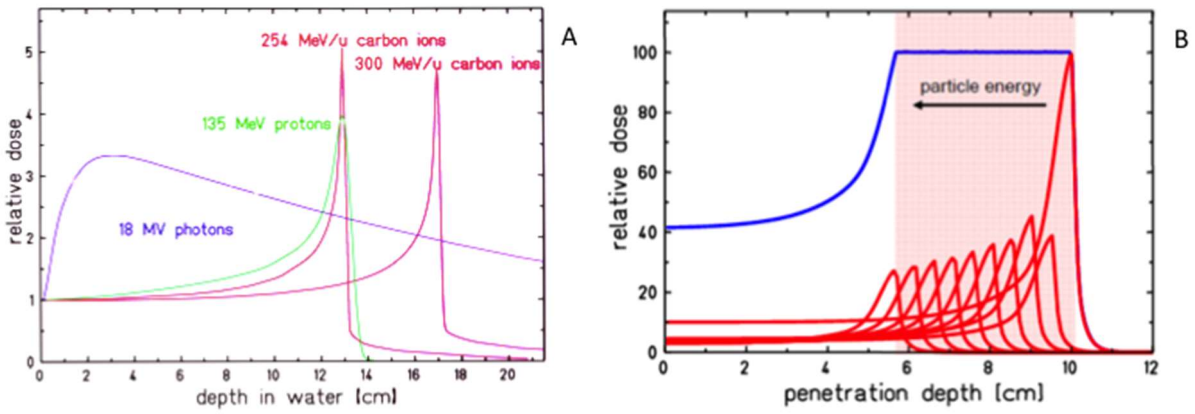



*Figure 1.1 Physical advantages of particle therapy. (A) Depth-dose distributions of high-energy X-rays and monoenergetic beams of protons or carbon ions. At the same range, C-ions have lower straggling than protons, but a tail of fragments is visible beyond the Bragg peak. In clinical applications, the Bragg peak must be extended to cover all the tumour (B). This can be done by overlapping different pristine beams at different energy and intensity. Figures from GSI Helmholtz Centre library.*

The advantages of the Bragg peak shown in Figure 1.1 A are quite obvious: unlike X-rays, the energy deposited per unit track increases with depth, therefore for a single beam the dose to the normal tissue will be much lower for ions than for photons when delivering the same dose to

the tumour. While in X-ray therapy it is necessary to crossfire the tumours from many different angles to increase the ratio between the dose to the tumour and normal tissues, only a few beams are necessary if charged particles are used (Figure 1.2). Thus, the same radiation dose to the tumour (and therefore the same tumour control probability, TCP) can be achieved with lower integral dose to the normal tissue (lower normal tissue complication probability, NTCP); or the dose to the tumour can be increased (higher TCP) keeping the same NTCP as expected for X-rays. IMRT is almost unbeatable in terms of conformity of the high dose region, but the cost is an even larger "dose bath" where the patient is subjected to a larger volume of low dose exposure.

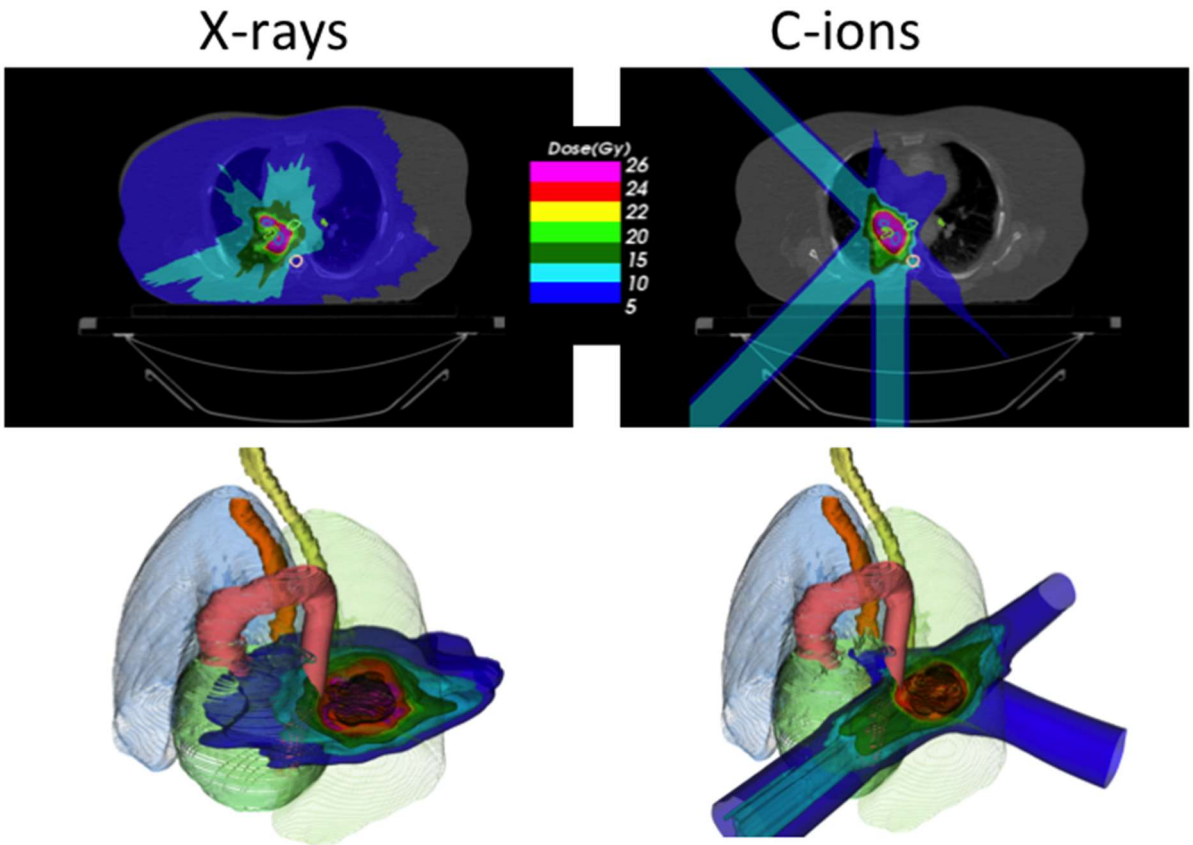



*Figure 1.2 Charged particles produce a reduced integral dose to normal tissue. Treatment of a lung cancer by stereotactic body radiation therapy (left) or carbon ions (right). Only 3 beams can be used with particles, as also shown in the 3D image below. Plans from Krjstian Anderle, Ph.D. thesis, Technical University of Darmstadt 2015.*

### 1.1. Physics of charged particle transport (CPT)

Both the longitudinal and lateral dose profiles resulting from the interaction of charged particles with the human tissues are important in CPT. The longitudinal profile is dominated by the inelastic electromagnetic interaction with atomic electrons, leading to a slowdown of the primary particles. Lateral profile is mostly caused by the elastic scattering on target nuclei and

leads to a broadening of the beam. Nuclear interactions reduce the intensity of the primary beam and contribute to both longitudinal and lateral profiles.

For moderately relativistic particles, the main energy loss channel is ionization of the atomic shell electrons. The usual continuous slowing down approximation (CSDA) is used by neglecting the higher-energy moments [15]. In CSDA, assuming that the mean energy of the ion is reduced less than 5% in an absorber thickness *d*, the mean energy loss is proportional to the stopping power *S* (or LET), adequately described for a large energy range in terms of mean energy loss and shell corrections in the Bethe formula, including Barkas-Anderson-Bloch corrections:

$$S = \frac{2\pi^2 N_A e^4}{mc^2}\frac{Z_p^2}{\beta^2}\rho\frac{Z_T}{A_T}\left[\ln\left(\frac{2mc^2\beta^2\gamma^2}{I}\right) - \beta^2 - \frac{C(\beta)}{Z_T} + Z_p L_1(\beta) + Z_p^2 2(\beta) + L_3(\beta)\right] \quad (1.1)$$

where *e* is the electronic charge, $N_A$ the Avogadro number, *m* the mass of the electron; $Z_p$ and $\beta$ the charge and relative velocity of the projectile, respectively; $Z_T$, $A_T$, and $\rho$ the atomic number, mass number and density of the target material, respectively; and *I* is the mean excitation energy. The various terms are the shell correction *C,* Barkas correction $L_1$, Bloch term $L_2$, and Mott and density corrections $L_3$. Eq. 1.1 is generally known as Bethe-Bloch formula, and it is generally considered accurate at high energies.

The CSDA range, which is essential for irradiating the tumour, and not the surrounding organs at risk (OAR) in the Bragg peak region, can be calculated by the stopping power:

$$R_{CSDA} = \int_0^L dx = \int_E^0 \frac{dE}{dE/dx} \quad (1.2)$$

where E is the initial energy and L the maximum range. Integration of the Bethe-Bloch equation (Eq. 1.1) is not a simple task and typical approximations is the Bragg-Kleeman formula [16]:

$$R_{CSDA} = \int_0^L dx = \int_E^0 \frac{dE}{dE/dx} \approx AE_0^p \quad (1.3)$$

Lateral scattering for thick targets is dominated by multiple Coulomb scattering and is well described by Molière's theory of multiple scattering [17]. At small scattering angles, Moliere's theory approximates the scattering distribution as a Gaussian with standard deviation σ that can be described as:

$$\sigma_\theta = \frac{14.1\ \text{MeV}}{\beta pc} Z_\text{p} \sqrt{\frac{d}{L_\text{rad}}} \left[1 + \frac{1}{9}\log_{10}\left(\frac{d}{L_\text{rad}}\right)\right] \qquad (1.4)$$

where $d$ is the total mass thickness and $L_\text{rad}$ is the radiation length, which depends on the atomic number $Z$ of the target material. Eq. 1.4 shows that multiple coulomb scattering increases for thick and heavy targets (because $L_\text{rad} \approx Z^{-2}$), whereas it decreases with particle velocity and, at the same range, with particle mass. For instance, protons have a lateral scattering approximately 3 times greater than C-ions at a range of 15 cm [18].

Nuclear interactions generate slow target fragments, which give a small contribution to the dose but can have high biological effectiveness [19]. If particles heavier than protons are used, projectile fragmentation produces fast fragments with a mean velocity similar to the velocity of the primary ion. These fragments have lower mass and therefore higher range than the primary ions, thus generating a longitudinal tail in the Bragg curve (Figure 1.1). The angular distribution of the fragments is narrow in the forward direction, but the spread of the lighter fragments (protons and helium) contributes to the lateral widening of the beam.

## 1.2. Beam delivery

As shown in Figure 1.1 B, the narrow pristine Bragg peak must be extended to cover all the tumour area (spread-out-Bragg-peak, SOBP). This can be done either by passive modulation of the primary beam, or by changing the energy while raster scanning tumour slices with a pencil beam (Figure 1.3). New proton therapy centres all deliver the beam using pencil beam scanning (PBS), in which the whole tumour volume is scanned in 3D using a narrow pencil beam [16].

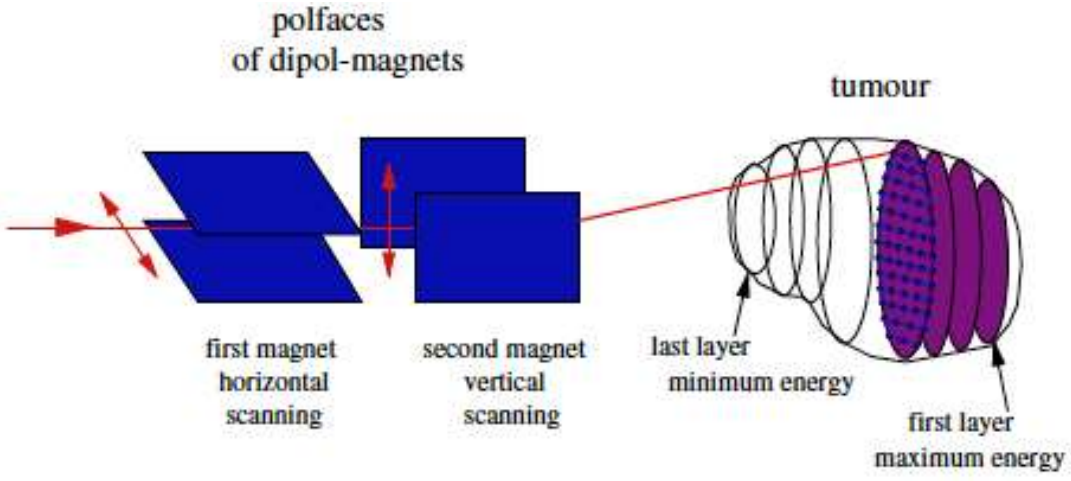



*Figure 1.3 Pencil beam scanning concept. Figures from GSI Helmholtz Centre library.*

PBS is also used in the majority of C-ion centres, with only a few centres in Japan still using the old passive modulation systems. PBS provides an unsurpassed conformity, but is problematic for moving targets, especially thoracic and abdominal tumours that move with breathing. The problem is caused by the interplay between beam and tumour movements, resulting in poor dose distributions [20]. The problem is tackled with different motion mitigation techniques [21]–[23], but some simple methods such as gating leave the problem of the residual motion, whereas accurate 3D tumour tracking with the beam requires complex on-line fast movement and range adaptation. A simpler way to handle range changes and complex motion patterns is 4D-treatment planning, in which the plan is optimized from a full 4D computed tomography (CT) scan of the tumour, thus including the motion. This technique is ideal for particle therapy [24], and in particular for therapy with heavy ions [25], in which the interplay between beam scanning and target motion produces poor target coverage.

### 1.3 Biologically weighted dose

Radiotherapy is based on the observation that radiation kills cells in a dose-dependent manner. Cell survival is generally described by the linear–quadratic (LQ) model [26]:

$$S = e^{-\alpha D - \beta D^2} \tag{1.5}$$

where $S$ is the fraction of cells surviving after irradiation with a dose $D$. The cell radiosensitivity is determined by the fitting parameters a and b, generally using their ratio. At high LET, $\alpha$ tends to increase and $\beta$ to decrease [27], resulting in survival curves that are almost exponential. The ratio $D_X/D_p$ of the doses of the reference radiation X (X-rays) and particle radiation p producing the same survival is the relative biological effectiveness (RBE) for cell killing, which Eq. 1.5 shows, is higher at low doses than at high doses, especially when a/b ratio is low. Even if the LQ model is only applicable in the low-dose range, typical of conventional fractionation in radiotherapy, it can be extended to high dose per fraction such as procedures used in Stereotactic Body Radiation Therapy (SBRT) [28].

The RBE is higher for heavy ions and increases the SOBP peak/plateau ratio in heavy ion therapy compared to proton therapy, because it is higher in the target region (peak; high LET) than in the normal tissue (entrance; low LET). The quantity used to account for the RBE in the physical dose is called *RBE-weighted dose* defined by the International Commission on Radiological Units (ICRU) as [29].

$$D_{RBE} = \mathrm{RBE}(E, D, a, b, c) \cdot D \ (\mathrm{Gy}) \tag{1.6}$$

where the physical dose in gray (1 Gy= 1 J/kg) is corrected by the dimensionless RBE factor, which is a function of the dose, the particle energy *E,* and several other factors (*a*, *b*, *c*…) such as dose rate, oxygen concentration, intrinsic radiosensitivity, among others. Given its dependence on so many parameters, RBE can only be calculated by a biophysical model such as the microdosimetric kinetic model[30], or the local effect model[31], both based on the LQ model (Eq. 1.6).

Among the various parameters, notable for radiotherapy is the RBE dependence on the intrinsic radiosensitivity (that is, the $\alpha/\beta$ ratio, derived from the X-ray dose-response curve; Eq. 1.6) and on the dose per fraction. The RBE decreases when the intrinsic radiosensitivity of the tissue or the dose per fraction are increased. Accordingly, the maximum RBE advantage is observed for radioresistant tumours (Figure 1.4). The fractionation dependence is intertwined with the radiosensitivity, because the RBE decrease at high doses is steeper for radioresistant than for radiosensitive tissues, that is, the RBE in hypofractionation decreases [32], [33] more sensitively for normal tissue ($\alpha/\beta \approx 2$ Gy) than for the tumour ($\alpha/\beta \approx 10$ Gy) [34]. Therefore, even if the RBE is low at high dose per fraction, hypofractionation is possible and indeed is often pursued in clinical treatments with heavy ions[35].

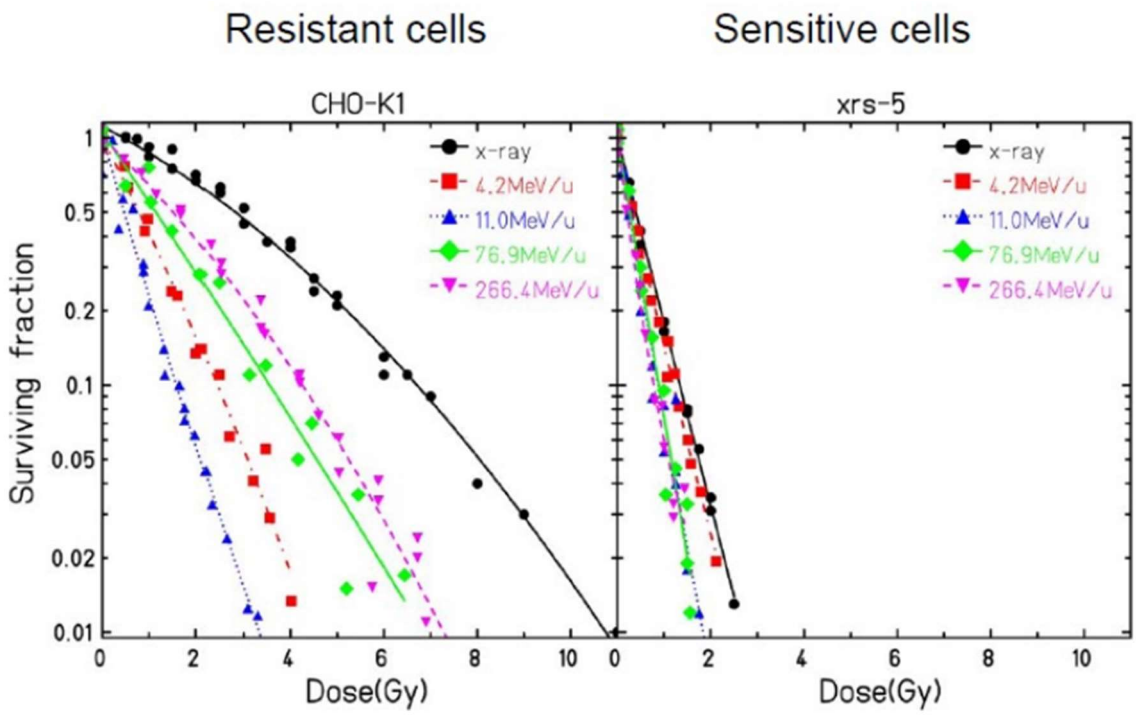



*Figure 1.4 Cell survival of hamster cells exposed to X-rays or heavy ions of different LET. CHO-K1 is the wild-type, radioresistant strain; xrs-5 is a DNA-repair deficient mutated cell line. RBE is large and LET dependent in the radio resistant strain but approximately 1 for radiosensitive cells. [36]*

## 1.4 Treatment planning

Between the diagnosis of cancer and the treatment some important steps should be done. The treatment should be planned to simulate the best settings of the beam to spare as much as possible of normal tissue and cover the tumour with the prescribed dose. A computed tomography (CT) scan is usually used to create a virtual version of the patient. The tumour and OARs should be segmented in each slice to reproduce 3D image, needed for the planning.

Segmentation of the tumour is divided into the following steps:

- Gross tumour volume (GTV) represents the tumour observed by imaging,
- Clinical target volume (CTV) covers in addition to GTV the area which should be removed with the primary tumour,
- Planning target volume (PTV) is another margin added to the CTV considered to assure a delivery of prescribed dose to the entire CTV.

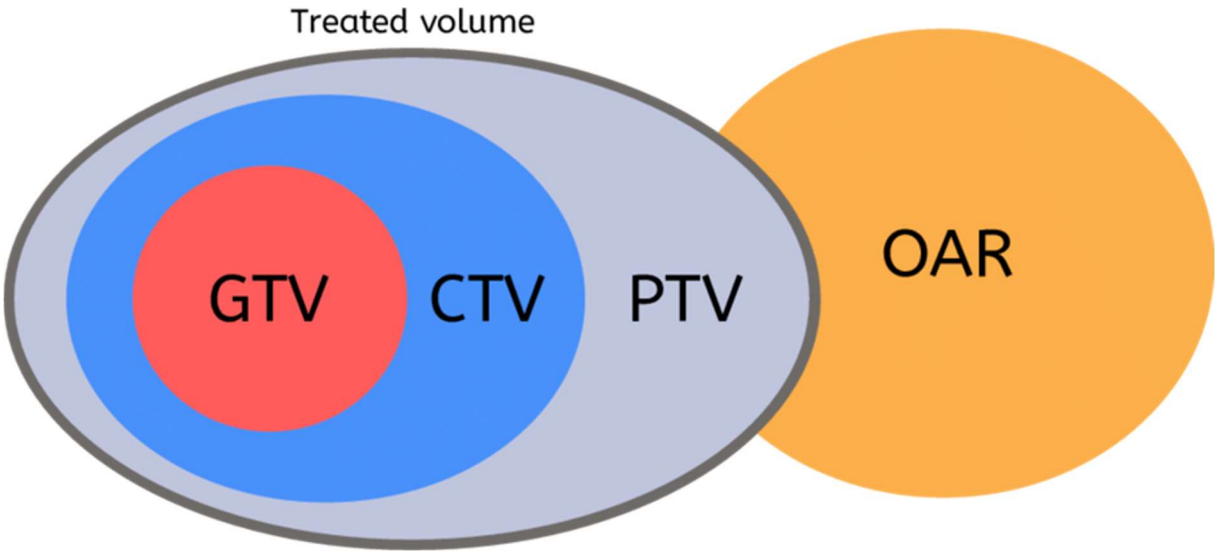



*Figure 1.5 Schematic of the target volume (TV) strategy in radiotherapy. The macroscopically visible tumour forms the gross tumour volume (GTV), with an extension to treat possible microscopic disease in the clinical target volume (CTV). A further extension to the planning target volume (PTV) is supposed to cover uncertainties in delivery to ensure full target dose in the CTV. Often, the target extensions overlap with organ at risk (OAR), creating a conflict in the planning targets, located in the irradiated volume.[37]*

The exact calculation of beam intensity and directions to cover the tumour with the maximum possible dose while sparing the OAR is called treatment planning. The calculation for every kind of radiation relies on base data. The treatment planning produces dose-volume histograms (DVH). In a differential DVH, each column represents the volume of an organ receiving the dose in x-axis bin. More common are cumulative DVH, where each point represents the volume of a given organ receiving ≥dose than the one given in the x-axis (Figure 2.3). With very fine

(small) bin sizes, the cumulative DVH takes on the appearance of a smooth line graph. The lines always slope and start from top-left of the graph (100% of the volume receives ≥ 0 Gy) to bottom-right (0% of the volume receives ≥ of the maximum dose). DVH summarizes 3D dose distributions in a graphical 2D format and are most commonly used as a plan evaluation tool and to compare doses from different plans or to structures.

The metrics used for evaluation of the treatment planning are specified as Dn, where D is responsible for the dose and n for the target volume in percent of volume (like cc), or Vn, where V is the volume and n is dose in the volume in percent. For example, D95 indicates the dose in 95% of the volume, which can be seen in Figure 1.6. The purpose is to deliver at least 95% of the prescribed dose [38].

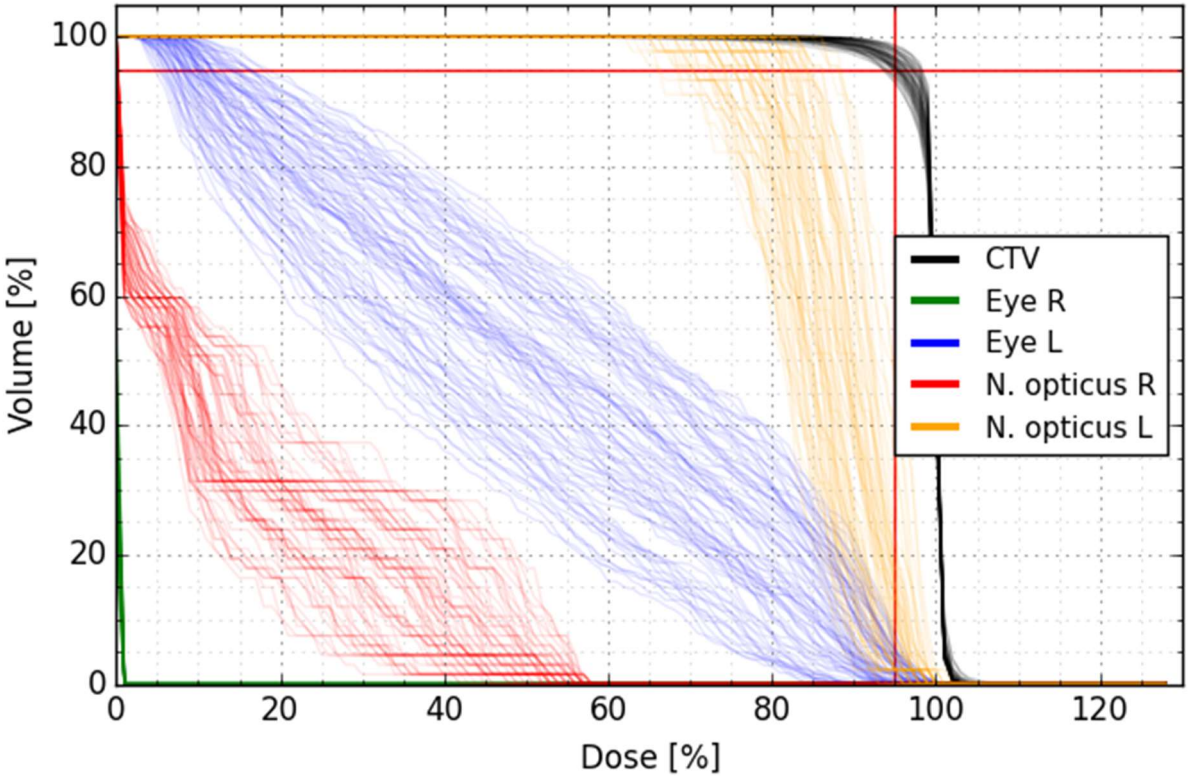



*Figure 1.6 Example of DVH for robust plan optimization. Multiple lines represent different scenarios use for the optimization of the plan. Red lines mark 95% of the scales. Figures made by Moritz Wolf from GSI Helmholtz Centre.*

The complexity of interactions of ion beams with living matter makes it difficult to provide purely experimental base data sets for treatment planning. Hence one has to rely on sufficiently accurate – and fast – calculations to obtain base data like depth dose distributions and particle spectra. The requirements of a physical model can be summarized as follows:

- handle all ions of therapeutical interest, i.e., $Z \leq 6$,

- work on a three-dimensional grid, i.e., the millions of voxels of a computed tomography (CT),
- the position of single Bragg peaks should be reproduced within 0.5 mm (1/2 CT voxel),
- the calculation of complete treatment plans based on a physical model should be reasonably fast, a few seconds or minutes per patient plan for a single CPU.

A further complication is that the position of the patient cannot be reproduced exactly as it was during the imaging. The possible shifts of the patient's position or changes in their anatomy should be taken into consideration during the planning to achieve a more accurate result, which is close to the real treatment. This is achieved in so-called robust optimization, where multiple uncertainty scenarios are considered. This increases plan robustness, but also drastically increase problem size.

### 1.5 TRiP98

The treatment planning system (TPS) used in this thesis was developed at GSI by Dr. Michael Krämer toward the end of the past century and is called TRiP98 [11], [12]. TRiP98 has a physical beam model for the beam transport; a biological beam model (Local effect Model, LEM [39]) to compute RBE-weighted doses; and an optimization algorithm to calculate the best intensity and direction of the $^{12}$C-beam. Siemens Syngo was derived in part from this TPS and is used in clinics for treating patients with $^{12}$C.

The physical model calculates electromagnetic and nuclear interactions, generating energy loss and straggling (longitudinal and lateral). With these ingredients, depth dose profiles can be generated numerically by convolving distributions as the calculation propagates from one depth interval to the next. The TPS is reasonably fast and allows to compute and tabulate ion pencil beams within a few minutes.

The absorbed dose, $D_{abs}$, generated by the superposition of such pencil beams at locations $x_b$ with number of particles $N = \{N_b\}$ at the location x of an irradiated voxel is

$$D_{abs}(x,N)[Gy] = 1.6 \cdot 10^{-8} \sum_b d(E_b, x_b) \left[\frac{MeV}{g \cdot cm^{-2}}\right] G(x, x_b) N_b \qquad (1.7)$$

where $d(E_b, x_b)$ is the planar-integrated dose per incident ion of primary energy $E_b$ in pencil beam b with number of particles $N_b$. The function:

$$G(x, x_b) = \frac{1}{2\pi\sigma_b^2} e^{-\frac{r^2}{2\sigma_b^2}} \qquad (1.8)$$

is the profile of the beam with width $\sigma_b$ at lateral distance $r^2 = |x-x_b|^2$. G may be restricted to the initial beam spot profile, or it may be modified by adding a second Gaussian to account for lateral scattering effects. If necessary, passive devices such as range shifters can be included in this scheme by simply adding their respective water-equivalent thickness to the depth coordinate.

From the physical dose in Eq. 1.7 it is now necessary to calculate the RBE-weighted dose (Eq. 1.6). This is done in TRiP98 with the local effect model (LEM) [40], [41]. The basic assumptions are that on the local level the radiation damage by sparsely ionizing photon radiation is the same as for particle radiation, and that the cell nucleus is the sensitive target. These assumptions allow to separate the difficult biological aspect of the problem from the purely physical one. The former is represented by the empirical photon dose response curve (Eq. 1.5), the latter by the microscopic radial dose distribution, D(r). The a and b parameters in Eq. 1.5 usually can be derived from a rich set of patient and laboratory data. D(r) can be calculated with approximated formulas [42]. Combining these main ingredients by integrating the dose response over the cell nucleus yields the response of a cellular system when exposed to ions of a particular type and energy, $-\ln S = \alpha_z D_z + \beta_z D_z^2$, where $D_z$ is the specific energy deposited in the cell nucleus. LEM only provides intrinsic αz, βz for single monoenergetic particle traversals in the first place. Computation of biological effects in complex radiation fields, as they are common in radiation therapy, is the task of the TPS. This task, i.e., to determine the survival in each voxel of the irradiated tissue, provides the RBE. The biological dose distribution then becomes

$$D_{bio}(x,N) = D_{abs}(x,N) \cdot RBE(x,N) \tag{1.9}$$

A typical single treatment field may comprise as many as $5 \cdot 10^4$ individual beam spots organized in up to 50 different energy slices. The TPS has to determine the number of particles, N, within each pencil beam so that the resulting RBE-weighted dose (Eq. 1.9) distribution matches the medical prescription. The latter includes not only the prescribed target (tumour) dose, $D_P$, but also the constraint that excessive dose values in OAR should be avoided. This can be formulated as a least-squares minimization problem:

$$\chi^2(x) = \sum_{x_{target}} \left[D_p(x) - D_A(x)\right]^2 + \sum_{x_{OAR}} \left[D_p(x) - D_A(x)\right]^2 \cdot \theta[D_A(x) - D_{OAR}(x)] \tag{1.10}$$

where $D_A(x)$ is the actual dose distribution calculated according to Eq. 1.9, and $D_{OAR}(x)$ is the maximum allowed biological dose in the OAR. θ denotes the Heaviside function, which

evaluates to one if the argument is positive, otherwise it is zero. Its purpose is to impose a penalty if $D_{OAR}(x)$ is exceeded, but to do nothing if the actual dose is below that limit. The sum runs over all voxels in the target and the OAR, respectively. The free parameters, N, to be determined are implicitly included in $D_A(x)$ via Eq. 1.7 and 1.9. The solution of Eq. 1.10 is not trivial due to the nonlinear dependence on N and the constraint term but can be achieved by appropriate iterative algorithms implemented in the latest version of the TRiP98 TPS, in particular methods of steepest descent (plain gradient) and conjugate gradients.

The method of steepest descent directly follows the negative gradient of the multidimensional $\chi^2$-function in Eq. 1.10 in order to find the minimum. Conjugate gradient methods use gradients too but take the previous iteration step into account to accelerate the convergence. Up to $10^5$ voxels and up to $7 \cdot 10^4$ different pencil beams may be required for a multifield treatment plan.

The size of tumours can vary from 2 ml to more than a litre. The bigger tumours contain more voxels. Moreover, the number of beams is different from patients to patient. All said above has high impact on the computational time. With the currently most efficient minimization algorithm, acceptable biological optimization can be achieved within 3-5 minutes for 3D non robust optimisation, on a computer with 16 cores and more that 64GB of RAM If the optimization is supposed to be robust or considers motion, both RAM requirements and computation times increase significantly.

# 2 Artificial Intelligence in medicine

"*Artificial intelligence is a branch of computer science capable of analysing complex medical data. Their potential to exploit meaningful relationship within a data set can be used in the diagnosis, treatment and predicting outcome in many clinical scenarios.*" [43]

AI is a very broad field (Figure 2.1), which is inspired by human beings and their behaviour. Even though the algorithms try to imitate us, they learn in a different way. For some tasks, which even children learn after a few attempts, like classification [44], moving objects [45] etc., a machine can struggle or needs a lot of data to learn the connection between the data and its label. At the same time, despite the drawback it is able to learn to distinguish many classes at the same time [46] and not to forget them, as people tend to do, and spend seconds instead of hours on repetitive tasks, like segmentation.

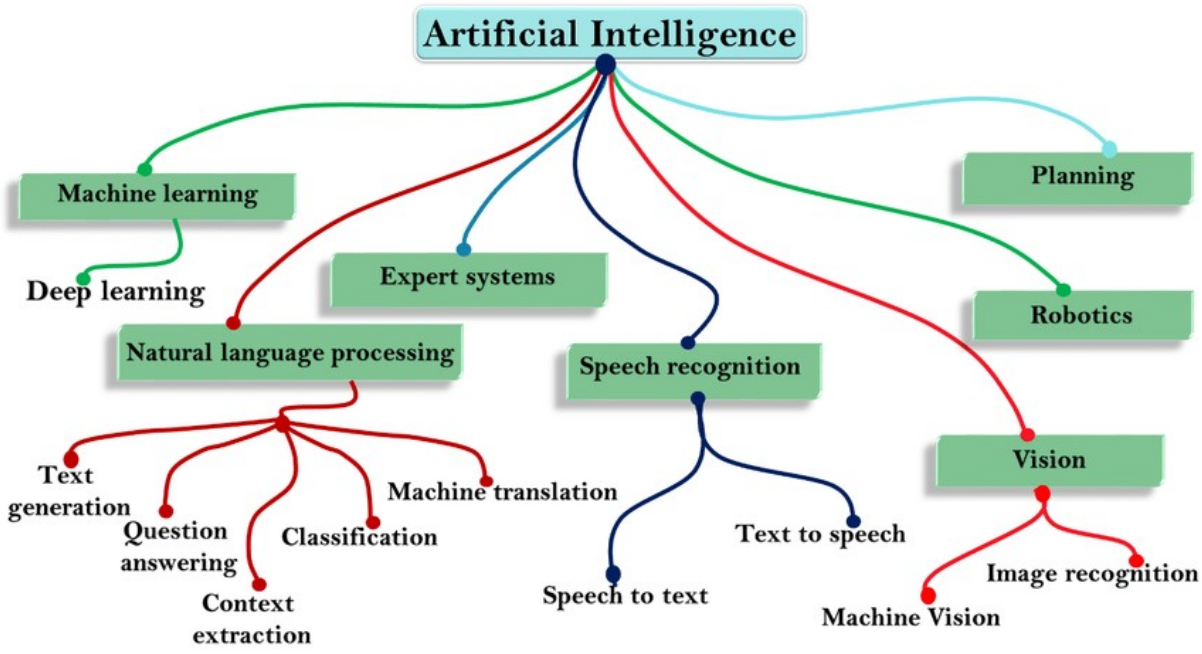



*Figure 2.1 Subfields of AI. [47]*

Further we observe some fields of medical research, where AI can be applied. Machine learning, and its subfield deep learning in particular, is used for the research described further in this work.

**Diagnosis**, especially at an early stage can save the patient's life. Some diseases are often diagnosed at last stages, when they start to show symptoms and sometimes are incurable [48], [49]. In our high-tech era, there are distinct application points for DL. For example, there is a high number of smartphone owners. They can benefit from a DL algorithm, because it might

be integrated in a smartphone as an application to classify an image for skin cancer diagnostics [1]. The DL architecture offered by Esteva et al. performed better than an average expert for identification of keratinocyte carcinomas (most common cancer) and malignant melanomas (most fatal cancer). Some diagnoses can be done even at DNA level [2].

**Prognosis** helps to forecast whether the patient will have a remission or has a high probability to get a disease withing the following years. For example, models that predict breast [3] or lung [4] cancer risk over 1 year outperformed medical doctors.

**Imaging**. Segmentation is an essential step in data analysis for many areas of biomedical research, such as detection [50], classification [51] or treatment planning [8]. Manual segmentation is a very long process, which strongly depends on a specialist who does it. Automatic systems allow to save hours needed for the manual procedure. U-Net was developed for biomedical image segmentation [5]. It and its variants [6] are successfully applied in many pieces of research. Some of reinforcement algorithms allow even a user interaction, where the missegmented areas are marked, which increases the quality of the final result [7], [52]

**TPS** are also a very tempting area for research. Monte Carlo simulation is widely used in TPS. It allows physically accurate dose calculation, and therefore more accurate planning of the treatment with high time cost. There are many interesting pieces of research, made for TPS in photon therapy, for example IMRT [8] and VMAT [53]

Of course, there are many more other pieces of research, which are definitely interesting but go beyond the thesis [54]

### 2.3 Basic concepts of Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of Artificial Neural Networks (ANN). CNNs are aimed at solving tasks when image input is involved. In comparison with ANNs, where all layers are fully connected (which can cause overfitting) and require regularization techniques, like dropout, CNNs are a regularized version of multilayer perceptrons by definition.

First of all, to start working with NNs the following main points should be considered:

- Dataset. The structure and the format of the data are very important for such algorithms. The result can be corrupted by a badly prepared dataset. Some examples of a data impact can be found in [55], [56]
- Define the desirable output shape. It is necessary to choose the last layer of the model correctly. For example, for the binary output, where the answer is only “yes” or “no” it is enough to have one perceptron as output
- Choose the model: number of layers, filters, kernels etc.
- Define your loss functions, which is a metric, used by NN to update the weights during backpropagation
- Choose an optimizer and its parameters
- Choose an evaluation metric

In the chapter we briefly touch upon some of the points above, which provides the surface understanding of the processes needed for the work.

Apart from the data there are some other basic elements of the main workflow, The next part of the algorithm is feedforward propagation, where the sample goes through the whole architecture and depending on the shape of the last layer comes out as the result. Then with the help of the output, the predefined loss function (also called error function) is calculated. The function allows the system to understand how far it is from the label (ground truth). Then the derivative of the function is calculated to update the weights in the system. This step is called backpropagation (Figure 2.2). But more detailed information exceeds the scope of this thesis and is not further explained.

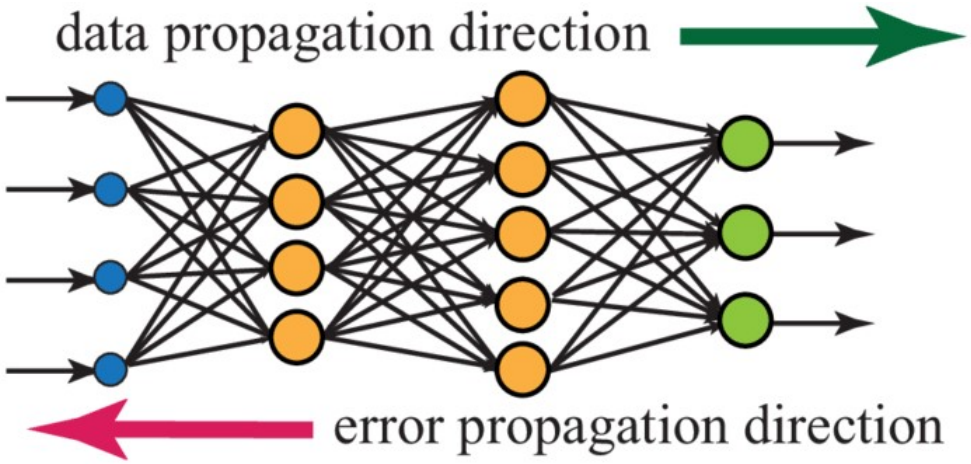



*Figure 2.2 Scheme of feedforward artificial neuronal network with error back propagation [57]*

After briefly touching the main concepts, we go into more detail of convolutional layers. They are the core block of a CNN. They consist of a set of kernels with a small receptive field. The kernel is a matrix usually with odd number of pixels in each dimension (3x3, 5x5). The kernel is convolved with the input image, i.e., takes the element-wise product at each pixel (Figure 2.3). The result is an activation map, which detects a specific class of feature at some spatial position in the input. Each layer consists of a predefined number of filters, which usually grows with the depth.

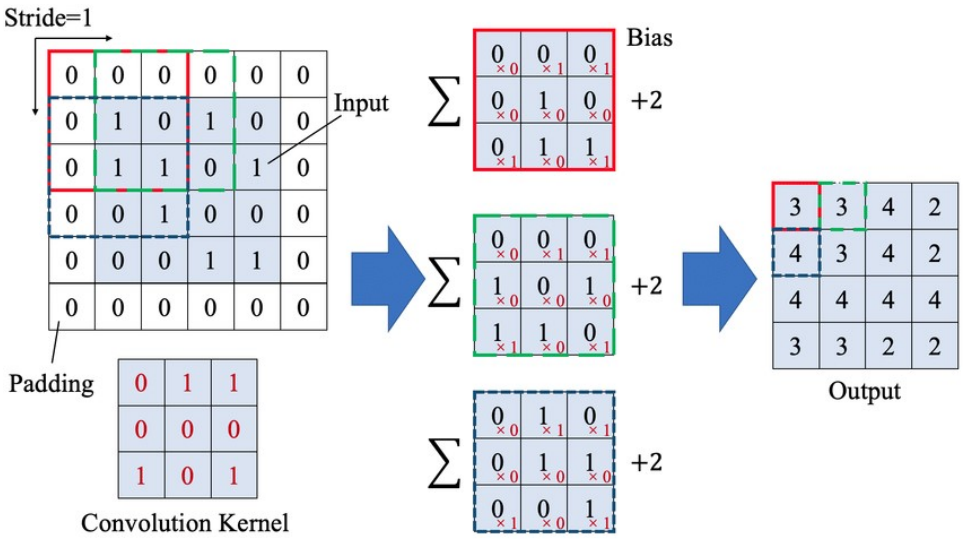



*Figure 2.3 Example of value calculation in a convolutional layer [58]*

The numbers in the kernels are learnable parameters, that means that during training the numbers are updated due to the back propagation to extract the most relevant features.

The convolutional layers also have hyperparameters, which should be defined before the training. For example,

- kernel size: number of pixels processed together. Usual the size is 3x3 or 5x5 for two-dimensional space, but can also be 1x1 for dimensional reduction or a combination of 3x1 and 1x3 to reduce the computational power needed for 3x3 kernels like in [59] (Figure 2.3)
- padding: a technique, which adds zero-valued pixels on the borders of an image. It allows to keep the input resolution in the output and not undervalue the information from the borders (Figure 2.3)
- stride: notes a number of pixels, which define the sliding step for the kernel window. For example, a stride of two means that each kernel is offset by 2 pixels from its predecessor (Figure 2.3)

- number of filters: number of feature maps to get as output after a convolutional layer (Figure 2.4). Each filter gets a specific feature, but a big number of filters can also cause overfitting. Usually, the number increases with the depth of the layers during the first dimensions, related to the original input decrease.
- Dilation: a technique, which allows to ignore some pixels within a kernel without significant loss of information. For example, a 3x3 kernel expands to a 7x7 kernel by applying a dilation rate of 2, but still processes only 9 more widely spaced pixels (Figure 2.5). For more information see 5.1.2.

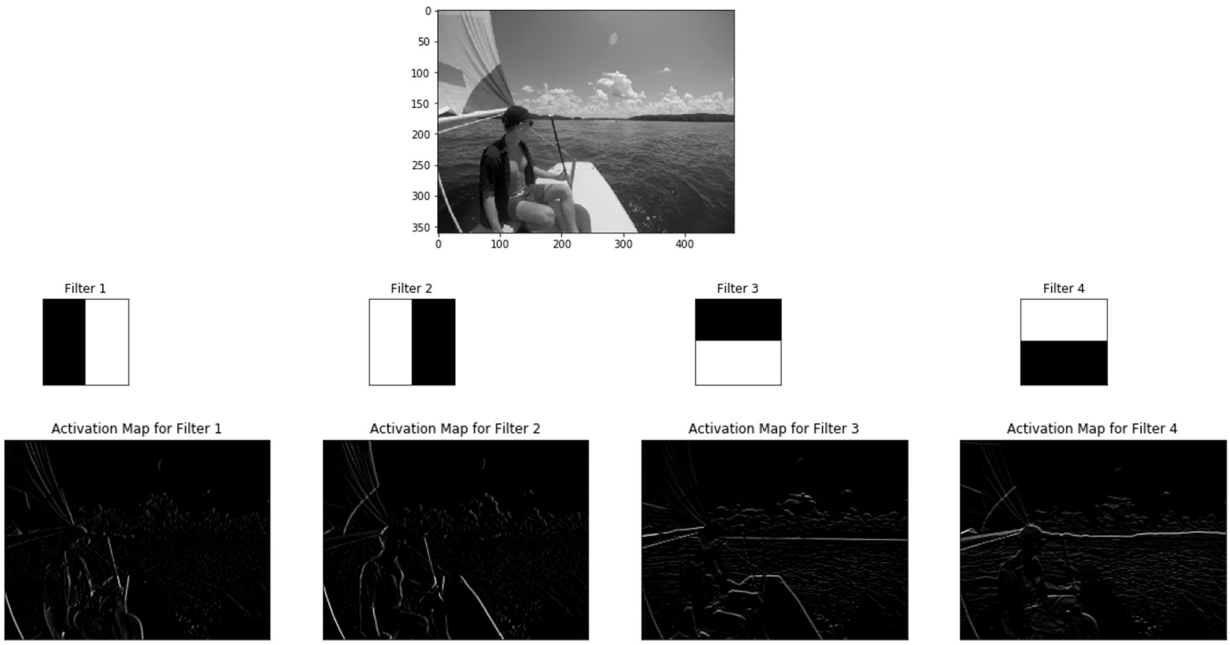



*Figure 2.4 Example of the effect of different filters [60]*

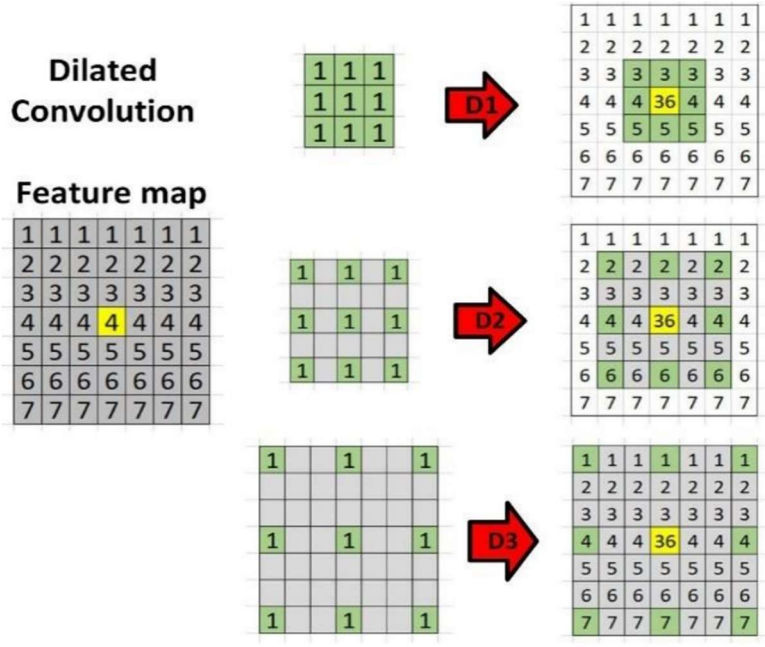



*Figure 2.5 Example of dilation rate parameter [46]*

After each layer an activation function is applied. The function transforms the linear output into the non-linear one and makes it easy for the model to generalize or adapt with a variety of data. Rectified Linear Unit (ReLu) (Eq. 2.1) and its versions (leaky ReLu) are typically used in CNNs.

$$f(x) = \begin{cases} 0, if\ x < 0 \\ x, if\ x \geq 0 \end{cases} \quad (2.1)$$

where x is an input value.

In the cases, where the probability values are needed as output, either the sigmoid activation function (multi-label classification) (Eq. 2.2)

$$f(x) = \sigma(x) = \frac{1}{1 + e^x} \quad (2.2)$$

or the softmax function (multi-class classification) (Eq. 2.3) is applied in the last layer and to each of the nodes.

$$f(x)_i = \sigma(\vec{x})_i = \frac{e^{x_i}}{\sum_{j=1}^{C} e^{x_j}} \quad (2.3)$$

where x is an input value, $\vec{x}$ is an input vector for a label in NN or CNN, $x_i$ is an i$^{th}$ element of the vector, C is number of classes.

Even though the NNs and CNNs in particular are extremely powerful, their main drawback is that they require a huge amount of prelabelled data.

## 2.4 AI in particle therapy

The area of AI application has been growing for the several years. Healthcare is not an exception. The era of the internet enables an easy access to the open source medical data such as [61], [62]. Increasing performance of CPU and GPU, also on cloud, foster accessibility of the deep-learning algorithms for many research groups. Kaggle competitions are also raising interest in the field. Such competitions not only provide data for free, but also give some prizes for the best result withing the competition time.

AI systems have a big potential in radiotherapy, which should be explored. The complexity of the radiotherapy techniques (see Figure 2.6) is growing together with the time and computational cost. AI algorithms can provide high accuracy and efficiency of the workflow in

radiotherapy, due to their ability to build a connection between machines and humans. For example, in image segmentation task user can give hints, where the segmentation was incorrect [7] and the system develops itself further to be more precise.

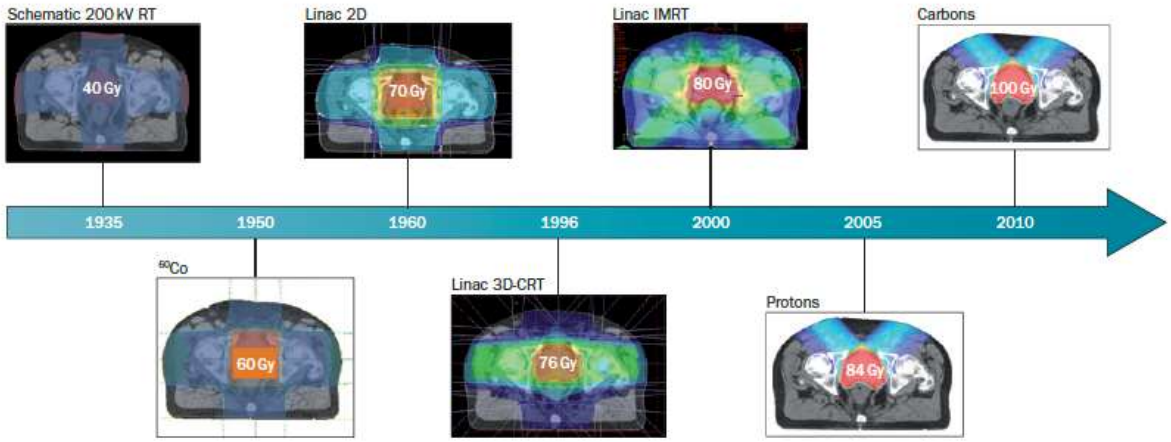



*Figure 2.6 Development of prostate cancer radiotherapy 1935–2010. It demonstrates the efficiency of different techniques to deliver the dose to the prostate without compromising the healthy tissues. The cold colours (like blue) represent low dose and the red, dark red the high dose region. Abbreviations: 3D-CRT, 3D conformal radiotherapy; IMRT, intensity modulated radiotherapy; RT, radiotherapy. [63]*

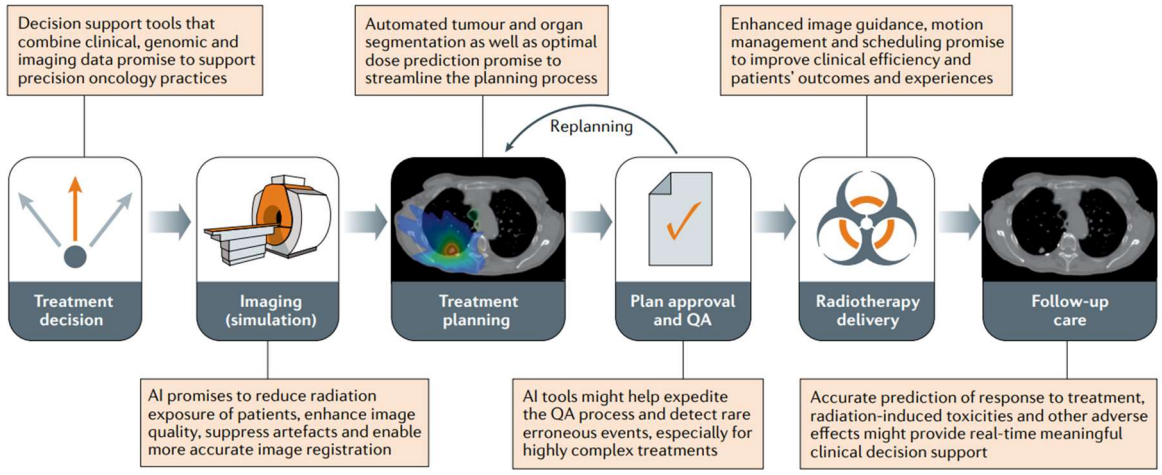



*Figure 2.7 Radiation therapy workflow and expected artificial intelligence applications in it. The workflow starts with a decision that the patient needs the radiation therapy, then the optimal parameter of imaging to gain quality and reduce radiation is chosen, followed by preparing the data for the treatment planning (segmentation). After the treatment planning is created, it should be approved with quality assurance (QA). After all the steps are done, AI is supposed to help with online therapy by controlling the position, motion of the patient, adjusting the plan accordingly, to deliver the best possible result. At the end toxicity and other measurements needed for a follow-up should be foreseen [10].*

Figure 2.7 describes the whole workflow of radiotherapy potentially based on AI. The imaging and segmentation steps are an important part that influences the final output of the dose calculation in the treatment planning.

Among the most studied applications of AI in treatment planning is segmentation. Segmentation is an important input for the treatment planning, since even with the best plan, one can only treat the tumour as good as it was originally outlined on the CT. High quality manual segmentation is time consuming and depends heavily on the experience and patience of Radiologists. Segmentations by different physicians vary often significantly [64], and even the same physicians segment organs different on different days. But the result of the TPS is correlated to the segmented organs, especially if the tumour is located very close to OARs. The recent progress in computer processing power_allows to speed up such stages of treatment planning just as imaging in general [65] And the deep learning techniques can not only reduce the time spent on the manual segmentation significantly [5], but also improve and standardize the quality of segmentation.

Most of the work in AI based TPS was done for photons [8], [66]. The studies showed that Convolutional Neural Networks (CNN) can compete in dose calculation for IMRT and VMAT with conventional simulation techniques.

On the other hand, particle therapy still remains a relatively unexplored field for such improvements. Multiple reasons contribute to this: one of them is that photon therapy is more common than particle therapy, meaning it has a bigger market for advanced applications and more available datasets of treated patients. Another one is that the physics behind those therapies is different. The particle therapy is more calculation demanding, requires a bigger number of parameters and is more sensitive to uncertainties than the photon one. These challenges make particle therapy depending on not only accuracy of the TPS, but also on relatively fast methods to calculate the delivered dose. AI would be a prime candidate to improve particle therapy TPS. However, the challenges mentioned above make implementation of AI methods more complicated. It is difficult to gain lots of data due to the specifics of the field. Especially heavier ion therapy is limited to just few centres worldwide, with the largest state of the art facilities having treated just a few thousand patients total each [cite PTCOG statistics here; www.ptcog.ch]. This represents patients treated for a number of different tumours with varying protocols, reducing the subset of data useful for training in each specific application. Since sharing of patient data between centres is not easily possible, this presents a bottleneck in developing new AI models and applications.

There are some pieces of research for the proton therapy. No AI-based TPS for heavy ions were found

The first trial to use AI for 3D proton dose calculation in volume was made by Nomura at al. using three-dimensional convolutional neural network (3D-CNN) [67]. A dataset of 193 head and neck squamous cell carcinoma patients was used to train and evaluate the model. Three following inputs data were used to calculate volumetric proton dose distribution: a binary surface mask, which gives a voxel the value of 1, if its inside the irradiated target, spot beam data (initial beam energy, spot weight and spot position along perpendicular axes to beam direction) and a 3D representation of stopping power ratio (SPR) relative to water.

The model demonstrated promising results of calculation the 3D proton dose distribution spending around 0.8 seconds for a plan and using 1500 spots with a consumer grade GPU and a mean absolute error of 0.778 cGyE. Once the model is trained it can be fine-tuned for proton dose distribution with another calculation method or beam parameters using a small database via transfer learning. The limitation of the model is the voxel size of 4 mm instead of 2 mm for clinical usage caused by the GPU capacity. Further innovative approaches were offered by Neishabouri et al. [68], who proposed to calculate the dose distribution for each single pencil beam using Last-Short Term Memory (LSTM) network, and Zhang, X. et al. [69], who offered to use a discovery cross-domain generative adversarial network (DiscoGAN).

Recently Wu [70] et al. proposed to use a combination of U-Net [5] and DenseNet [71] called HD U-Net to make the result of the pencil-beam (PB) dose calculation closer to the Monte Carlo (MC) results. Even though Monte Carlo simulation improves all the time, for example using GPU, it still cannot process the information in a few seconds. Nevertheless, it still stays the most precise technique for dose calculations [72]. The PB method is faster, but the precision of the dose calculation suffers from approximations used. The offered U-Net architecture has provided significant improvement of the dose and average gamma passing rate obtained the result over 89 times faster than MC.

# 3 Previous method. Selection of the voxels by random mask [13]

From empirical studies it was concluded that the separate subsampling of the voxels from the boundary shell and interior volume considerably reduces the computational load and time needed for plan optimization.

A Euclidian distance transformation in three-dimensional space [73] is used for separation of the boundary shell from the interior volume. The randomization parameter for each voxel of both subvolumes can be described with the following probability for a specific sampling parameter x:

$$Px = 1/x \cdot 100\% \tag{3.1}$$

The width of the boundary shell is set as 2mm. The efficiency of the choice is proved on a lung patient using robust optimization (see Figure 3.1) with 9 uncertainty scenarios. 21 uncertainty scenarios were used for the robust analysis to calculate D99 as the evaluation variable.

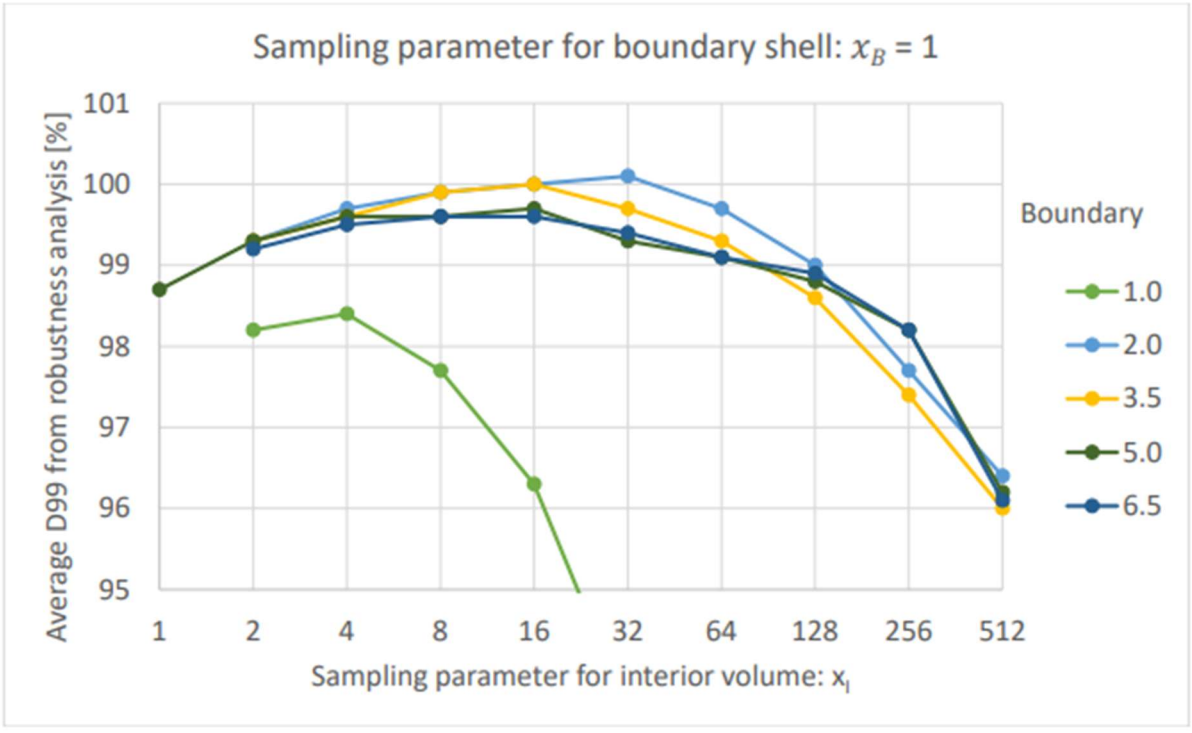



*Figure 3.1 Average D99 values from RA for constant boundary parameter $x_B$=1 and varying both $x_i$ and boundary widths*

The example of the best D99 average value of 100.1% of the prescribed dose in the target is shown in Figure 3.2. The result was achieved with boundary width of 2 mm and sampling parameter for interior volume 32 (appx. 3% of the voxels)

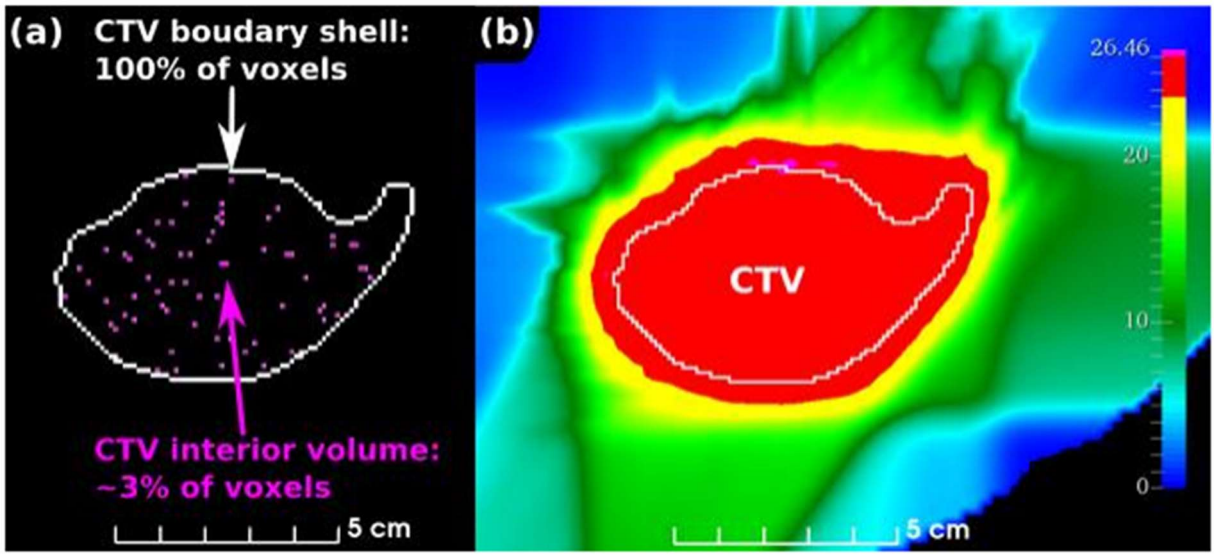



*Figure 3.2 Randomly sampled voxels assigned for optimization (a). Resulting nominal dose distribution (b)[13]*

The random sampling method for the lung cancer patient reduced the computation time by a factor of about 30 (317min to 10.6min) and RAM use by approximately 8 times (91.1Gb to 11.3Gb) compared to the nominal case. This is highly beneficial, especially in the context of daily adaptive treatment workflow, where the irradiation is adapted to the patient anatomy as seen on the day of the actual treatment. However, a setback of this method is the need for manually optimizing the sampling parameters for different cases in a try-and-error approach. Moreover, a more specific distribution of the active voxels in the subsample (e.g., more active voxels close to organs at risk) could potentially provide a more reliable result compared to the random voxel selection and enable further reduction of the number of active voxels. In this thesis, a novel CNN approach for voxel selection is developed in order to tackle the first point, I.e., to reduce the need for manual parameter tuning.

# 4 Data

The treatment planning data is taken from a cancer patient database (450 patients), treated with carbon ion at GSI from 1997 until 2008 [18], [74], [75].

For this work, 50 head and neck cancer patients' plans were selected without taking the specific location of the tumour into account. 30 patients were selected randomly for the training and the rest 20 ones were used for testing.

The plans were adapted to the latest TRiP98 version (see more in 1.5 and 6.2.2). for patients treated with an outdated treatment protocol, the treatment plans were updated, such that the same plan optimization protocol was used for all patients. All plans in the dataset have two personalized beams from different angles, to be optimized. The following parameters were used to create a random mask for subsampling the voxels in each volume-of-interest (VOI): the boundary shell thickness was set to 2 mm, the boundary shell sampling parameter was set to 1 and for the interior to 8 (see 3 for more details). Thus, all voxels associated with the shell were used during plan optimization, while only 12.5% of voxels from the interior contributed.

The voxel optimization is influenced not only by the voxels located in the area which belongs to their VOIs, but also by the voxels from the other closely located VOIs if those are involved into the treatment. For example, if the tumour is located close to the left eye and its optic nerve (Figure 4.1), then the optimization in the contiguous optical nerves will be compensated by the nearest voxels in tumour. The bar plot (Figure 10.1 in 10.2 Additional material) displays information about relative frequency of different OARs appearance in the dataset.

The number of OARs (Figure 4.2), which are located close to the tumour, also has an impact on the optimization process in the target. This way the selected voxels (SV) on the border of the OARs are also involved into the calculation process for the target.

The median of the number of OARs in each plan for the whole dataset is 5. As we can see the extremes with only 1 or 7 OARs in patients are represented only in the test dataset while the patients with 3 OARs on the treatment plan appear only during the training. At the same time both datasets include all represented OARs (Figure 10.1 in 10.2 Additional material)
The number of voxels in target varies from $9 \cdot 10^3$ to $130 \cdot 10^3$ voxels. Such big variety can corrupt a CNN if not enough samples are represented. We decided to keep in in our dataset to check the robustness of our system explained in the next chapter

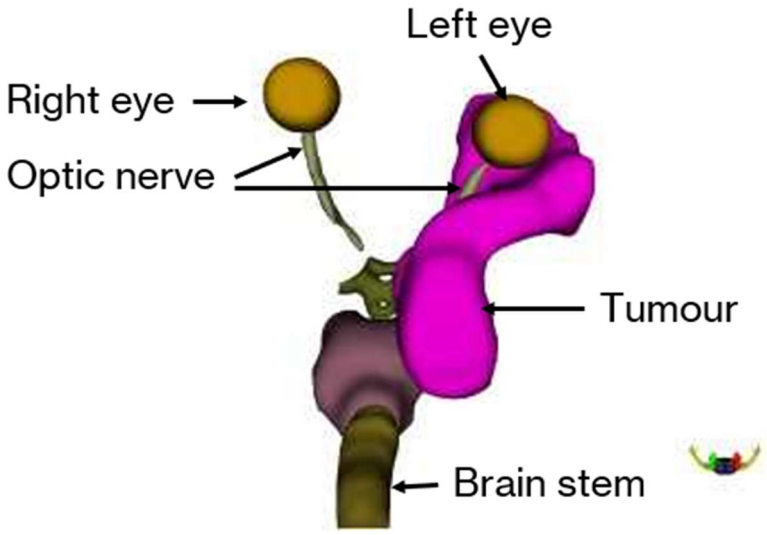



*Figure 4.1 3D view of the segmented organs and the tumour in a patient from the database*

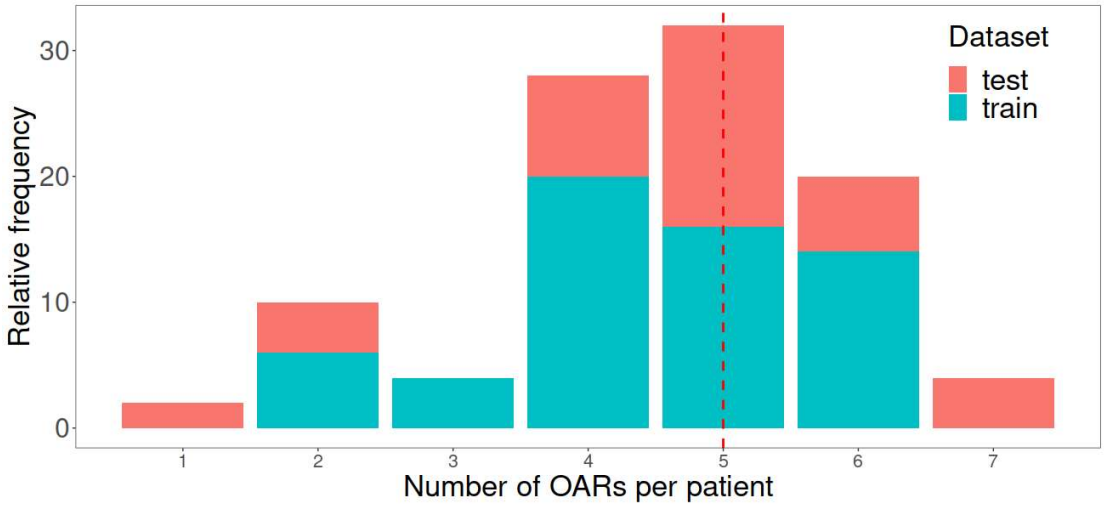



*Figure 4.2 Relative frequency of OARs per patient in the database. The median number of OARs in the whole dataset is shown with the dashed red line.*

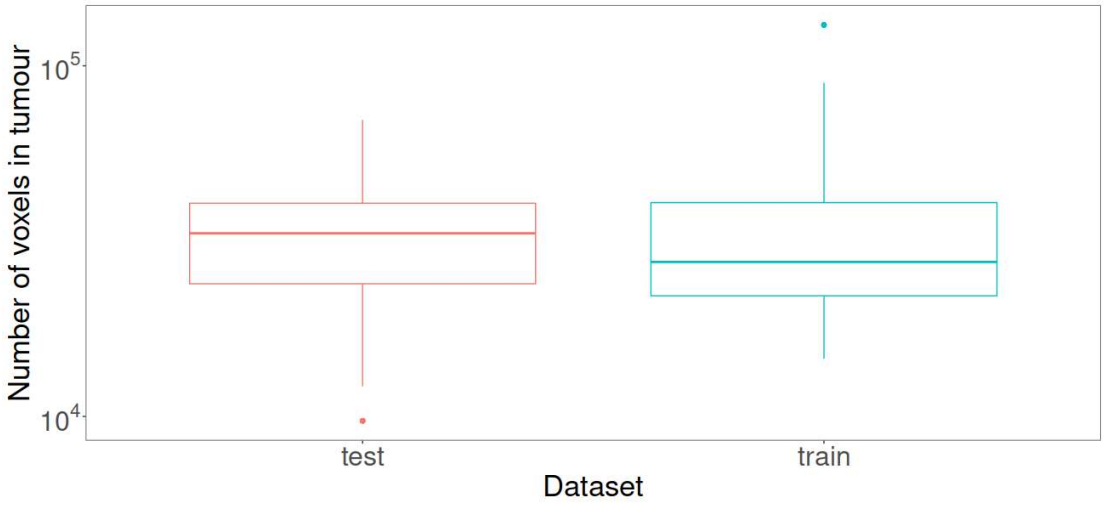



*Figure 4.3 Distribution of the number of voxels in target*

For each patient the following data was produced with TRiP98 (see 1.5) from CT data and the treatment plan:

- Binary mask: within VOIs relevant for the optimization (Figure 4.4), it indicates which voxel will be included in the optimisation. it depends on the sampling parameters set in the TPS (for more information see 3). The binary mask with the following sampling parameters was used as labels for the loss function (see more in 5.1.3): 1 for the boundary shell and 8 for the interior shell (see more in 3). Small VOIs, like optic nerves and chiasm were not sampled.
- Label map: assigns each VOI a numerical label ($2^n$, where n is the serial number of the VOI in the related script). The target volume is always the first one. It allows not only to count the voxels in different VOIs, but also to apply a loss function (see 5.1.3) to the specific voxels.
- Distance map: shows the Euclidean distance in millimetres from the shell towards the interior volume for each of the VOIs present (Figure 4.4 (right)). The distance calculation also takes 3D structure of VOIs into account.

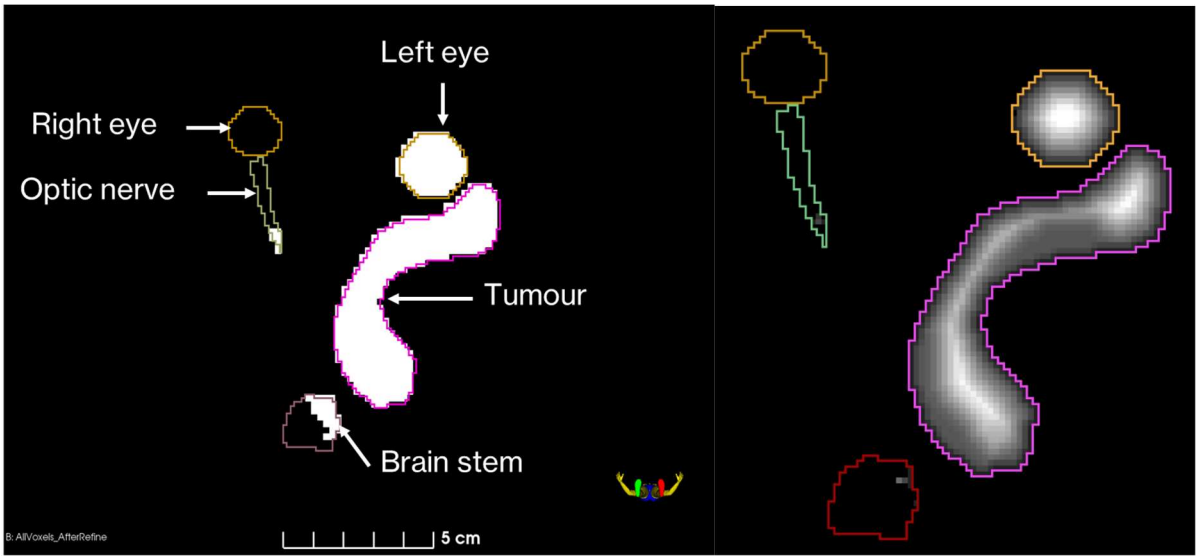



*Figure 4.4 Example of a mask (left) and distance map (right). The colours of the voxels show the distance (the whiter the higher the distance). The contour shows the original VOIs, used in the plan.*

Each voxel in the target has a value, which displays its Euclidian distance transformation in three-dimensional space [73]. The relative frequency of all voxels depending on the distance can be seen in Figure 4.5. The first is corresponded to the boundary shell where currently all voxels are selected.

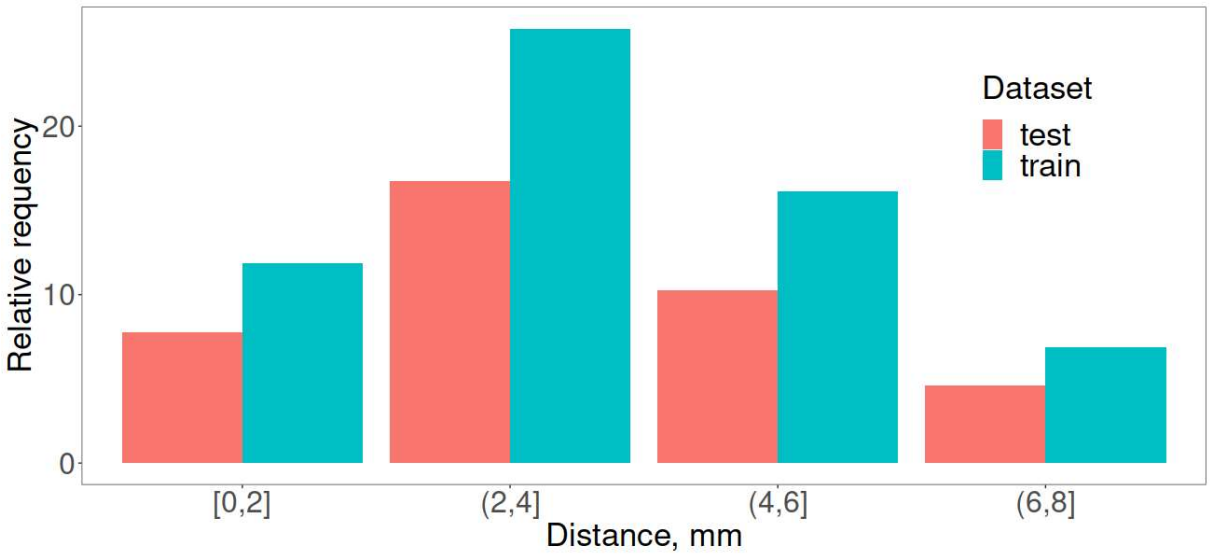



*Figure 4.5 Relative frequency of voxels with different distance*

# 5. Current method

In this work, a new AI based model for the task of VOI voxel selection in carbon ion therapy treatment planning is proposed. There were no similar models found in the literature, that could be transferred to our problem of automatic voxel mask subsampling. Hence, in search for the ideal solution, at first, we went through plain ideas trying to keep the system as simple as possible. Starting from high complexity causes difficulty in error recognition.

The randomly sampled mask has some analogy to segmentation. But in comparison with it, the sampling has more constraints and, therefore, higher complexity. The biggest obstacle arises from the fact that the fidelity of a particular voxel mask can only be evaluated, by executing the complete TPS chain, i.e., dose calculation, plan optimization and evaluation of its dosimetric viability. There is no a-priori way to determine, whether a particular voxel mask will result in an acceptable plan. One possible outcome of AI based voxel selection hence could be the detection of patterns which voxel areas are most relevant for the plan optimization. Naively, one could for example expect that areas where target and organs at risk are very close should require more attention in the optimization, for the plan to fulfil the target dose coverage while adhering to organ at risk dose limits.

Some of the medical tasks, for example (segmentation, photon treatment planning) allow to separate the three-dimensional CT data into a list of subsequent two-dimensional slices. But the physics behind particle therapy TPS is more complicated (see 1), so, the voxels in different slices are interdependent. For the same reason patching cannot be applied. Consequently, the algorithm should be able to work in three-dimensional space.

We started with mathematical metaheuristic optimizers, such as Particle Swarm Optimization [76] and Simulated Annealing [77], because it was intuitive to try to move the SV as particles in the volume, restricting them to remain in the VOIs. Usually, such optimizers are not used for a high number of variables as presented in the work in three-dimensional space. So, the algorithms were overwhelmed with the number of variables and dimensions. Moreover, the value of one moved voxel becomes invalid when the surrounded voxels are moved. So, it could only be applied to searching for the labels, which can be used in addition to the RM to train the CNN. This algorithm is not able to learn from its experience, which makes it time consuming (for robust optimisation 10 to 60 minutes), because each patient has to be explored from scratch. Of course, it also means that the processes for each patient can be parallelized. But there is no computational possibility to open too many TRiP98 (See 1.5 and 6.2.2) instances at the same

time, because some plans for patients with a big tumour volume even without all robust calculations require tens of gigabytes of RAM. Nevertheless, none of these methods is able to control the number of SV and gets no TRiP98 response of the dose distribution.

We therefore decided to use CNNs, which are broadly used for image processing. The convolutional layers with their kernels allow not only to get the most relevant information from the input, but also make CNN more robust for the shifting in comparison with NNs.

The next naïve CNN-based approach is to take a voxel and crop a cube around with some information about the environment and position of the neighbouring voxels. The approach has some weak points, which convinced us not to use it. Even though it allows to get rid of the cube size limitation, the system cannot recognize the full 3D distribution of voxels in the volume. The number of voxels only in the tumour itself can reach 130 000 and it takes very long computational time to process them. Besides, the approach cannot be parallelized, because a change in one cube causes the output data change in the others. And it is computationally expensive to recalculate the dose after each change, and those little changes would be insignificant.

The next level implies using the matrix, which contains the target and OARs, produced with TRiP98 as input. The matrix is a subvolume of the whole CT image, which is typically 512x512 in x and y. They represent a layer of the CT scan, their number (z dimension) is variable from patient to patient depending on the shape and the volume of the tumour. The subvolume contains only the target and the OARs relevant for the treatment planning.

The shape of subvolume varies strongly for all dimensions (from 17x59x45 and 38x102x54 to 10x56x112), which is not acceptable for CNN as input. We changed the shape of all subvolumes to 40*104*114 by centring it in the new matrix and adding zeros at the borders used for both training and testing.

After all, the data used in the training should be concatenated by the last axis. We used “Channel last” architecture and axis 0 should be added for the batch size (equals 1 in our case). At the end the data is driven to the following shape: (1, 40, 104, 114, 3)

Of course, for three-dimensional input the size can impact the training time dramatically, especially without usage of GPU. So, such architectures as 3d U-Net and V-Net [78] are not flexible in terms of the input shape changes because of the max-pooling layers.

We tried to find an architecture, which could have a good potential for improvement, further usage in the project and could be able to handle so many constraints and dependencies.

## 5.1 Architecture

The systems based on changing dilation demonstrated an impressive result in semantic segmentation [79].

In this work we tried to use P-Net [80], used as a part of DeepIGeoS for segmentation with the further improvement after a user interaction R-Net. Instead of using decoding and then encoding, as V-Net [78], which loses some information in the feature maps, it uses dilation to expand information about the environment of each voxel.

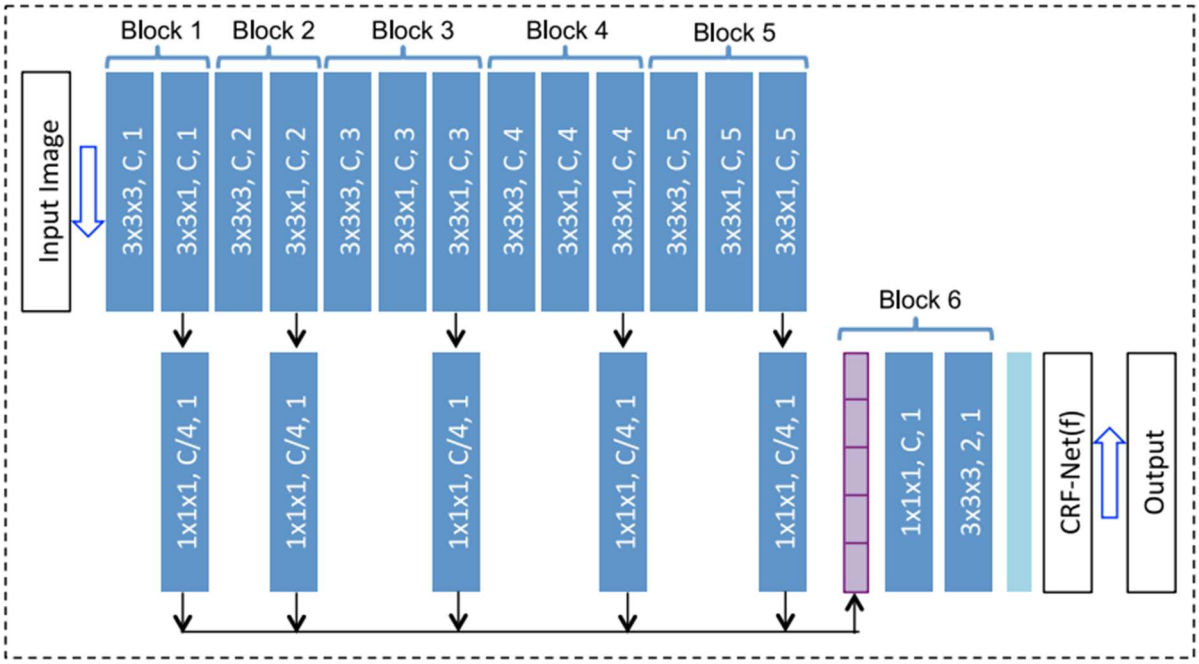



*Figure 5.1 P-Net architecture for 3D image input. The parameters of the convolution layers (dark blue rectangles) are kernel size, output channels, dilation. The information from Blocks 1-5 is concatenated before Block 6.*

IteR-MRL (Iteratively-Refined interactive 3D medical image segmentation via Multi-agent Reinforcement Learning) [52] used a combination of PixelRL [81] and P-Net in the context of reinforcement learning (Asynchronous Advantage Actor Critic (A3C)) in medical segmentation. The main idea is that the naïve multi-agent reinforcement learning (MARL) approach can be avoided using dilation in the system. It allows the voxels to communicate between each other without having a separate architecture for each one.

IteR-MRL allows the system to improve the segmentation within n-steps through interaction with the user. The user gives the system some hints of which areas should be corrected. As input, the original 3D image, previous segmentation probability and the hint map were used.

BS-IRIS (Boundary-aware Supervoxel-level Iteratively Refined Interactive Segmentation) [7] (Figure 10.2 in 10.2 Additional material) improved the Dice Coefficient [82] of the results by deleting the down- and upsampling and improving the reward system.

To check whether the architecture is working for our purposes the idea of the BS-IRIS was simplified by taking only the policy network (actor) from the A3C. This approach allowed to simplify the complexity of the reward shaping [83] needed in RL systems to the level of the loss function reduction used in CNNs. The initial parameters of the BS-IRIS architecture were saved, except the fact that a linear activation function was used instead of ReLu in the last convolutional layer Figure 5.2.

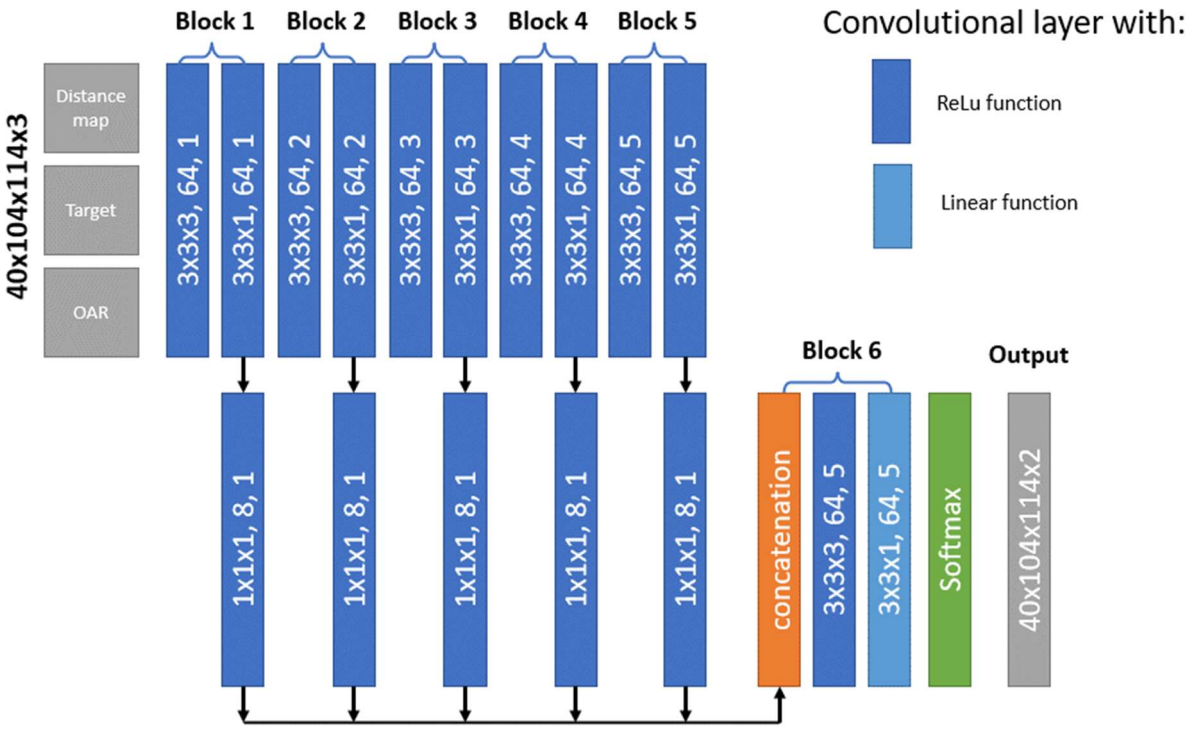



*Figure 5.2 Final architecture used in the thesis. The parameters of the convolution layers (dark blue and blue rectangles) are kernel size, output channels, dilation. The information from Blocks 1-5 is concatenated before Block 6.*

### 5.1.1 Concatenation of the feature maps

Due to the same output size from the layers the chosen architecture allows to concatenate the results of the feature maps before the last block.

In general, the research of the feature map output is a separate direction. The complexity of the information grows with the number of convolutional layers. The complexity is divided into low-, middle- and high features. In that way, the first layers look for the information about edges, spots, then with the growing depth the following layers identify pattern, parts and objects [84]. Different operations with feature maps are met also in U-Net for the reconstructions of an image in upsampling layers [5], in DenseNet [71] as substitution of the residual blocks in ResNet [85].

Considering the above, the benefit of the concatenation of the features from all convolutional blocks should be the most completed information about the input data. In conventional architecture each following layer gets only the input from the previous one. If we benefit from the method above compared to a conventional one, where only the output of the last block is relevant, should be investigated in this particular case.

### 5.1.2 Dilation

Systematic usage of dilated convolutional layers in semantic segmentation was offered by Yu [86]. The architecture allows to expand the perspective field of each point without losing resolution or coverage of images.

In our work it is important to save the information about the distribution of the SV inside the VOIs. The main purpose was to teach the system to understand that the number of SV should be minimized, and their position plays a significant role for the result.

In such systems max-pooling is typically used, especially in segmentation tasks due to its ability to reduce the resolution and to save the most important information at the same time. In our case it could cause the loss of the volume distribution. While beyond the scope of this work, it would be an interesting point for further research.

Use of dilation showed significant results in segmentation tasks [87]. The idea behind it was that the growing dilation rate helps a voxel to communicate/get information from more remote voxels as with traditional kernel when dilation rate equals one.

Padding “SAME” is used to keep the resolution of the masks, otherwise the final mask will be smaller than the primary one. The padding is implemented by adding zero levels to the image.

Further we are trying to understand what information is received by different dilation rates.

The first example (Figure 5.3) shows the same binary mask proceeded with different dilations, where all pixels (for simplification no 3rd dimension) with value equal 1 are corresponded to the optimization points (selected voxels in the work)

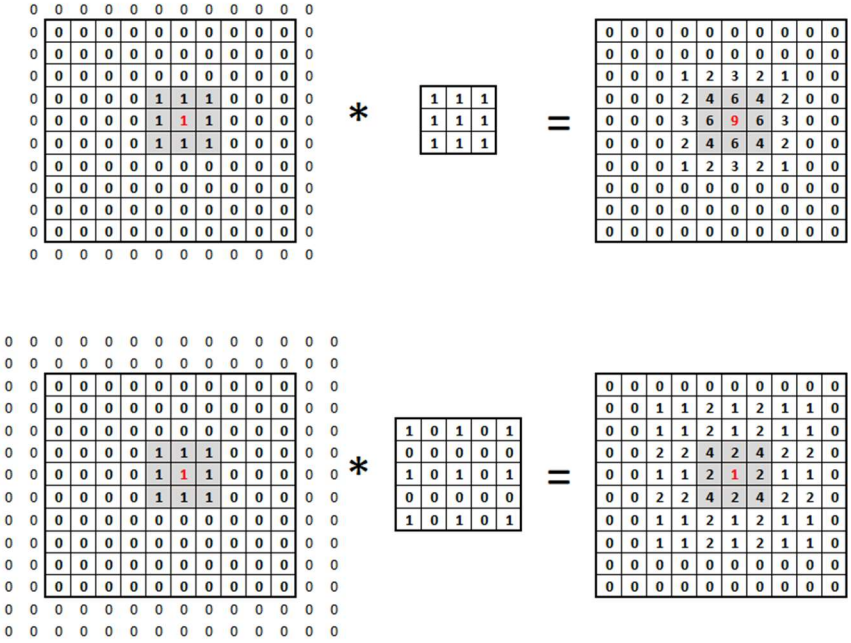


*Figure 5.3 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified binary mask*

If we take the kernel with only ones, which is simple to understand, it visualizes the information about each pixel neighbours. For example, the central pixel (marked red) gets the value of 9, which is the number of the closest neighbouring SV including the central pixel itself. If the kernel of ones with dilation rate equal two is applied to the same matrix, it is obvious that the pixel can "see" further neighbours but become "blind" for the nearest ones.

The second example demonstrates a chain of growing dilation for one single mask and an example of a random mask with dilation in a row. It allows to combine the information from both layers. So, the green pixel receives the information that its remote neighbour has 2 closest neighbours

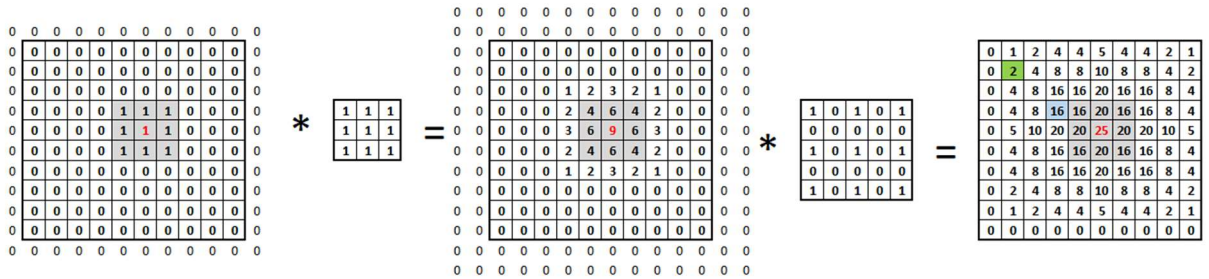


*Figure 5.4 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified binary mask applied in 2 layers in a row*

To see the changes in the output in the case of the random mask (only some pixels are selected) we reproduce the experiment like the one, which is shown in Figure 5.4. The received value is lower due to the lower number of the selected voxels, which are the information carrier for the selected kernel. So, the information allows to estimate the distribution of the voxels around each point.

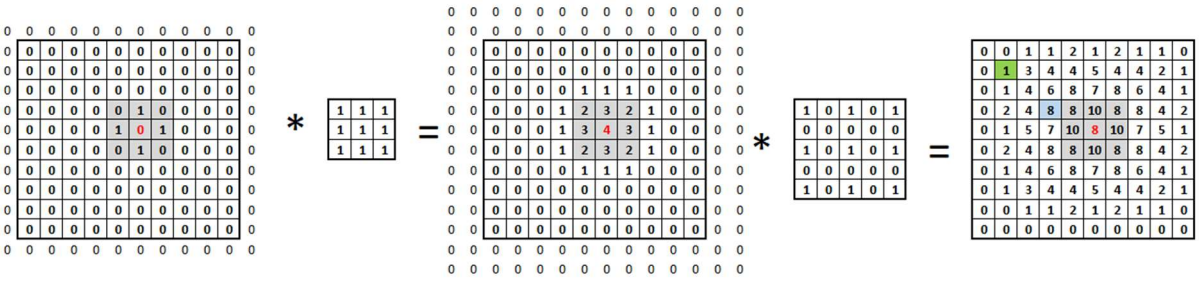


*Figure 5.5 Two-dimensional example of dilation rate (1 for the top figure and 2 for the bottom one) on a simplified randomly sampled binary mask applied in 2 layers in a row*

The combination of the 64 filters should allow the system to get more information about the position of surrounding SV that were already selected. Besides, the voxels from organs at risk can affect the optimization if they are located close enough to the target.

### 5.1.3 Loss function

The sampling parameter of the voxels for the optimization has no ground truth in the traditional way. That means that the same number of SV can be evaluated differently depending on the distribution of those voxels in the volume. So, the pre-implemented versions of the typically used loss functions, such as Mean Squared Error (MSE), and Cross-Entropy (CE), cannot be applied in this work. Information about the loss should contain two main conditions:

1. The number of SV should be reduced
2. The dose constrains for the target D95 should stay above 95% of the prescribed dose (3 Gy)

The masking technique used in [67] is very promising, as it allows to remove unnecessary connections and penalize only the relevant area. The effect is reached by setting the loss of the irrelevant areas to 0. For example, the area out of VOIs has no SV by definition. However, in our particular case, where we want to select as few voxels in OARs as possible, using this without adaption, results in the system to reduce the probability for voxel selection rapidly until no voxels are selected. Therefore, we did not pursue it within the scope of this work.

Creation of a loss function without having reference data is a very challenging task. There is some research, which created new loss functions like [88], but they also need a point for comparison (label), which we do not have in our case. The systems, which were the basis for our method [7], [52], [81], were applied both for segmentation and in reinforcement learning context (reward and punishment instead of the loss function). That is why we had to try a function that would reflect the response from TRiP98 and motivate our system to explore and exploit. The next issue we had to face was how to apply the function. We tried many combinations such as: on all voxels, only on selected ones, etc. We use the same approach as [7], [52], [81], where the mean loss was used for gradients calculation.

To overcome the disadvantage of not using masking we pretrained the system with cross-entropy (CE) loss, where the RM was used as the true label. The pretraining reduces the number of combinations, which the system should try before it understands what causes the highest loss of the system.

$$L = CE = -\sum_{i=1}^{C} y_i \cdot log\hat{y}_i \qquad (5.1)$$

where $y_i$ and $\hat{y}_i$ are the ground truth and the CNN score for each class $i$ in C. The values correspond to probabilities of classes, that means that $\sum_{i=1}^{C} y_i = \sum_{i=1}^{C} \hat{y}_i$=1. To get probability as output activation function (softmax or sigmoid) (see more in 2.3) is usually applied. The metric represents well how the true and predicted probabilities are different from each other.

To reduce the complexity of the work we restricted the evaluation only to the target. It allowed us to understand whether the offered system could find the best probabilities for the masks, without tuning the loss functions of the VOIs simultaneously. Because different organs have different sensitivity to radiation, individual constrains for maximum dose, shape, and distance from the target.

For the main part of the training the following loss function was used:

The loss in each voxel calculated

$$\mathrm{Lx,y,z} = \begin{cases} if\ voxel(x,y,z) is\ target: \begin{cases} If\ voxel\ was\ selected: p_{x,y,z} \\ else: 0 \end{cases} \\ CE\ (Eq. 5.1) \end{cases} \qquad (5.2)$$

where x, y, z are coordinates of the voxel and $p_{x,y,z}$ is probability, taken from softmax (Eq. 2.3) for the $i^{th}$ class

The approach is based on the following ideas:

- The optimization in the voxels in the shell is sensitive not only to the position of the neighbouring voxels in the target, but also in the OARs, located closer to the it. So, the continuous learning of the random mask allows to evaluate the target in different environments.
- The output mask provides only probabilities for a voxel to be selected. So, it is a challenge to give the information about the selected voxel back to the system. The use of probabilities only in SV allows to tune it slowly.

Final loss is calculated by:

$$L = C * \frac{1}{N} \sum_{1}^{N} L_{x,y,z} \tag{5.3}$$

where C, a CNN independent variable (further called independent variable), depends on the number of SV in the target and planning target values, received from TRiP98.

To avoid meaningless calculations with the mask, which has less than 3% of SV the independent variable was set to $10^6$ and those masks were not sent to TRiP98. In other cases:

$$C = \begin{cases} If\ D95 \geq 95\%: \%\ of\ selected\ voxels \\ else: \begin{cases} if\ \%\ SV > 30: |\%\ of\ selected\ voxels\ - 0.3| * (95 - dose\ in\ \%) + 1 \\ else: \exp(|\%\ of\ selected\ voxels\ - 0.3|) * (95 - dose\ in\ \%)/10^6 + 1 \end{cases} \end{cases} \tag{5.4}$$

The independent variable penalizes the total Loss in case of underdosage and is a regulator of the loss impact in case the minimum is reached.

To derive the independent variable, the following ideas were used:

1. To simplify the problem, we did not use any reward (loss reduction in the context of CNN) for the cases, where the planning target variable D95 was higher than 95%, which means that the system could not distinguish any benefits of a particular voxel mask above that dosimetric threshold. The first part expresses the threshold for the dose in the volume: $D_{95}\% \geq 95\%$ (95% of the target volume should be more or equal to 95% of the prescribed dose). As a result, the total loss was strongly reduced by the percentage of the SV. For example: we assume that we have an ideal system with no loss in the cross-entropy part of the matrix and, hence, it has no impact on the mean. The target has 100 voxels, in the first case the system selects 20 voxels with 25% probability, in the second one 10 voxels but with 50%

probability in those voxels (the other voxels can have a different probability, but they were not selected with the sampling method). C in the first case equals 0.2 (20% of the voxels) and in the second one 0.1. The mean in both cases is 0.05 ($\frac{1}{100} * (0.25 * 20) = \frac{1}{100} * (0.5 * 10)$), because the voxels, which were not selected get 0 value. This way, the final loss L will be reduced in 5 times in the first case (L=0.2*0.05) and in 10 times in the second one. The approach helps to provide the system with the information that the percentage of SV must be reduced, which is reached by reduction of probabilities in voxels.

2. We also had to create a loss function for the cases, where the target is underdosed. The value is critical for the whole treatment planning. The CNN has no direct connection to the TRiP98 calculation of the dose, so the loss has to be set very high to teach the system that the expected result was not reached. Besides, the function had to have a minimum in a "safe area", in which the constraints can be kept, but still leave a good starting point for the further steps. We set it to 30%. This was done because the decrease of percentage of SV caused a decrease of the dose coverage (D95), so there is a trade-off how low the percentage should be to still be able to fulfil the constraints. The independent constant consists of two parts, because it is more critical for the TRiP98 optimisation process to have only a few voxels in the mask than to have more SV in comparison with the random mask. Let us have a closer look at the independent variable for the case of underdose in the target. The main part of the equation is $(95 - dose\ in\ \%)$, where 95 expresses the threshold for the dose relative to the target dose value we receive after the optimization in TRiP98. The lower the dose, the higher the part of the equation. The first term |% of SV - 0.3| regulates the dependency of the loss function from the percentage of SV with the minimum at 0.3. For the percentage lower than 30% we use an exponential term, which rapidly increases the loss by reducing the percentage. It had to be divided by 10^6 not to let it be too big, because the difference between small steps would have been big and the system could have missed the part of the loss related to the rest of the cube.

   The addition of 1 at the end is used to avoid having the minimum of the function at 0, which could confuse the system because it is not the final goal of the process.

A graphical representation of the loss function can be seen in Figure 5.6.

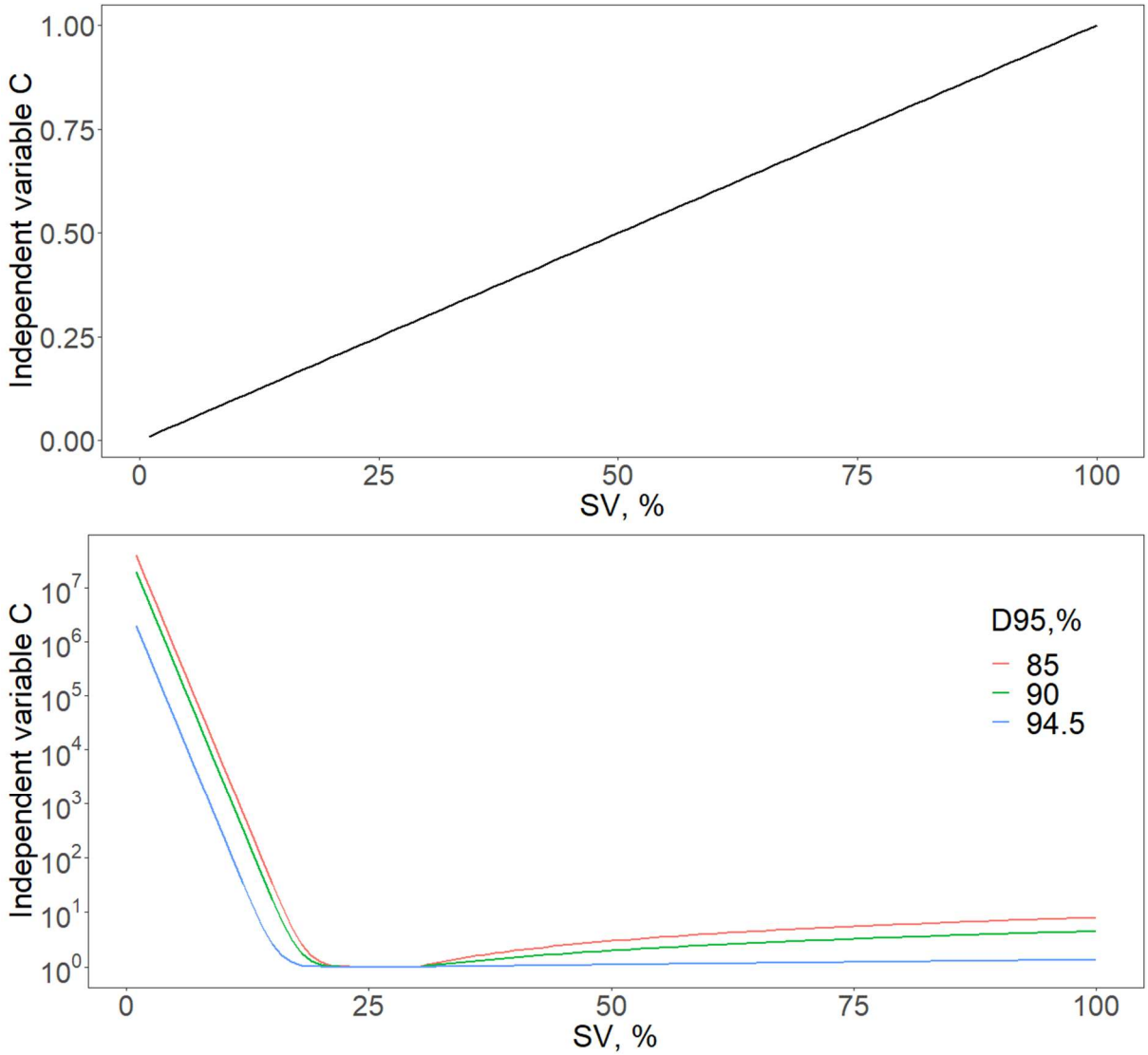



*Figure 5.6 Loss function for D95≥95% (top), D95<95% (bottom).*

## 5.2 Evaluation metrics

The question which evaluation metric reflects the best relationship between the estimated and true results is challenging and has a significant impact on the outcome. Especially when it is applied to two- or three-dimensional output. In that case accuracy ($\frac{true\ positive+true\ negative}{total\ number\ of\ samples}$), which is often used in classification tasks, can lead to the opposite result during the training. In Figure 5.7 Example of accuracy applied to an image in segmentation task [75] the accuracy of 95% for the masking task is shown.

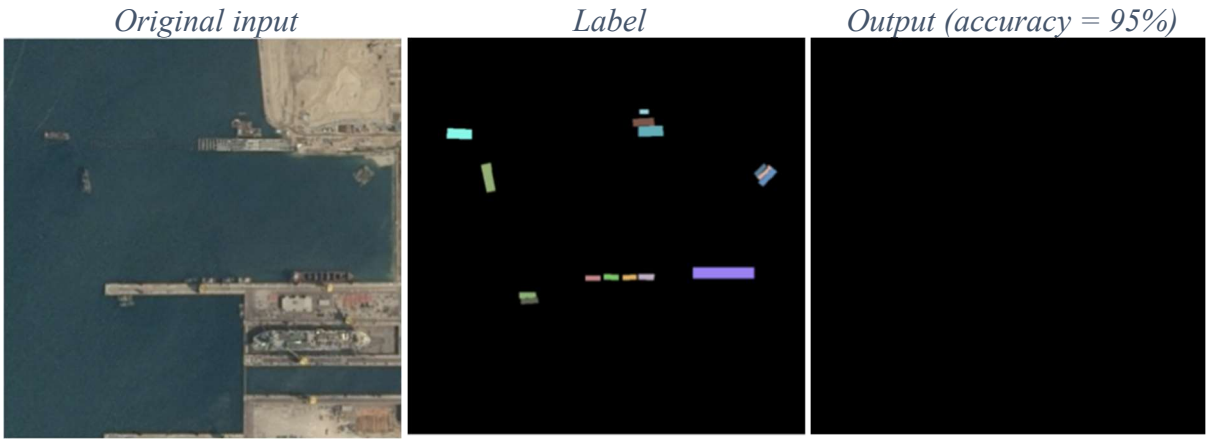



*Figure 5.7 Example of accuracy applied to an image in segmentation task [75]*

In segmentation, the Dice coefficient is often used. It allows to measure the similarity between two masks or segments in the image. However, even though we have the three-dimensional image output we cannot use such metrics because the main goal of the algorithm is the best possible positioning of voxels in the VOIs for optimization of the dose. In our case the evaluation requires metrics that can be calculated from the DVH (See 1.4). The description of our evaluation metrics is represented below.

For the target volume (tumour) the following metric was applied: D95 > 95%[38]. It means that 95% of the volume should receive at least 95% of the prescription dose. Of course, the higher the percentage is, the better, but we need to face the trade-off between the speed and accuracy of the calculations. The metric is important, because underdose would compromise tumour control and promote local recurrence.

In this work, we do not concentrate on the metrics of each of OARs. They depend on whether the organ is parallel (many or all disabled subunits cause the organ failure, like a kidney), or serial (disabling of any subunit causes failure of the entire organ, for example, a spinal cord). So, for parallel organs it is important to know that the dose in the certain volume does not exceed a certain limit, while for serial organs the maximum dose in a small volume like 0.03 cc is crucial.

Even though OARs are not a part of the optimization in the work, it was decided to use $D_{0.03cc} \leq 2.46$ Gy for all OARs as the estimator of the treatment plans. It equals the $D_{0.03cc} \leq 55$ Gy over the full treatment course delivered in XX fractions. The limits are used for chiasm, optic nerve and brainstem [89], which are the most frequent OARs in our plans.

## 5.3 Evaluation methods

In order to provide a fair comparison to the behaviour of the random mask, when changing the number of the voxels involved in the optimization, we run the whole dataset with the following parameters: 1, 2 and 3 for the shell and 1, 2, 3, 4, 6, 8, 10, 12 and 14 for the interior volume (see more in 2).

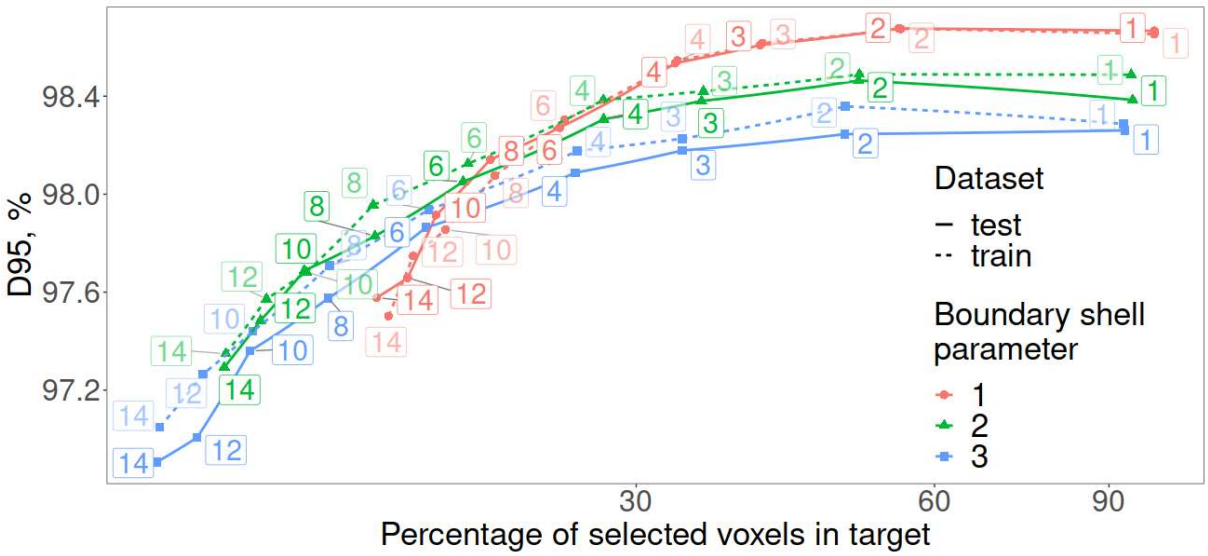



*Figure 5.8 Relative dose vs percentage of selected voxels in target. The data points are medians of the datasets. The numbers represent the sampling parameter in the interior. Semi-transparent labels denote the datapoints used for the training dataset.*

Figure 5.8 shows importance of the sampling parameter of the boundary shell for the optimization process. Interestingly, the red curve (boundary shell parameter = 1) at an interior sampling parameter of 8 is as good as the green curve (boundary shell parameter = 2) with an interior parameter 6. The two curves cross, with the boundary shell sampling parameter of 2 providing equal or better dose accuracy at lower percentage of SV than when a boundary shell sampling parameter of 1 is chosen.

Each patient's data was processed (see more in 6.3.1) during both training and testing. We use all the data for representation because in terms of training each sample, even repeated 5 times, is still independent; and during the testing, we prove the stability of the system. The reason for that is that as output we receive probabilities for the voxels to be selected. The sampling method is not stable and leaves the mask changing depending on the random seed.

Two types of training were accomplished. Both of them start with the model, pretrained for 5 epochs with CE loss, Adam optimizer and learning rate of $10^{-4}$. As a label, the random mask with the following sampling parameters was used: 1 for the boundary shell and 8 for the interior shell (see more in 3). For the trainings, the learning rate was set as $10^{-5}$.

The first type of the training demonstrates the workability of our own loss function and the system in general. We trained the model from epoch 6 to 20 with only CE loss (hereafter denoted 'Experiment 0'). The second one is the model trained from epoch 6 to 20 with our own loss function. This experiment was repeated three times (hereinafter Experiments 1-3) to check the reproducibility of the model. Checking reproducibility is important for this work, since different percentages of selected target voxels can result in acceptable plans, meaning the stable convergence of the model is not a-priori clear. At first, we would like to demonstrate how efficiently the pretraining stage can reduce the number of SV out of the VOIs (misselected voxels) (the reasons are described in 5.1.3). Figure 7.1 shows the learning process in the pretraining stage, aimed at reducing the number of the voxels selected outside the VOIs

$$Misselected\ voxels = \frac{Number\ of\ selected\ voxels\ in\ outer\ space}{Number\ of\ voxels\ in\ outer\ space} \qquad (5.5)$$

The value depends on the random seed for the current run of the algorithm because the voxel is selected with a certain amount of probability. Hence, the tracked value is approximate.

To evaluate the results, the following data was tracked in each step:

- The value of the total loss function
- Percentage of SV in each VOI and outside of them
- Evaluation metrics for each VOI (D95 for the target and D0.03 cc for the OARs) (excluded in the pretraining stage for time consuming reasons)
- Probability to select a voxel (softmax layer output)

Each experiment is evaluated in the way described below.

For each model it is important to evaluate the changes of the loss function to see, whether the system converges or not. We also track the dynamic of the misselected voxels to see the impact of the chosen loss function on the general improvement of the model.

The dynamic of the changes of D95 and percentage of SV is represented by two boxplots (train und test dataset), in x-axis the epochs are displayed as numbers, the full mask (with all voxels in VOIs) as FM and the random mask as RM. For better comparison of the data between the experiments all plots were scaled in the same way: from 0 to 50 for the percentage of SV and from 75 to 100 for D95. That means that the samples with D95 being less than 75% will not be shown.

For better visualisation we also plot the median D95 compared to the median of the percentage of SV for each epoch.

To understand the differences between the random mask and the mask produced with our system we investigated the probability change in voxels in the $6^{th}$ and $20^{th}$ epochs in the test dataset. For that, we discretized the distance map into 2 mm bins and checked the output of the softmax layer for action 1 (select voxel) for each bin. That way we obtained a density of probabilities in each distance bin.

# 6. Implementation

This chapter contains description of the main hardware and software components, which were used for the work. Besides the implementation, the chapter provides more details on data preparation, architecture nuances and difficulties faced during the work.

## 6.1 Hardware

All calculations were processed on a 16 core Intel Xeon CPU E5-2689 0 processor with ta clock rate of 2.6 GHz, and 128 GB RAM. No GPU was used. Usage of the cloud servers such AWS was not considered for regulatory reasons in terms of the patient's data. The main operation system was Ubuntu 20.04 LTE.

## 6.2 Software

The mask, generated with the offered CNN is a part of the treatment planning system TRiP98 workflow and should get some input information before the mask can be generated and give the mask back to the system to finish the calculations. The program is written in the programming language C.

This made direct implementation of the tools developed in this work to TRiP98 difficult, since suitable frameworks are more widely available in other programming languages. Python is widely used in the context of CNNs, since it offers a lot of ready-made solutions for building Neural Networks or Convolutional Neural Networks without going deep into the coding of each layer. The common packages like TensorFlow [90], [91] and Pytorch [92] get updates and bug fixation constantly. It makes them more trustworthy than self-implemented layers.

In order to get around this limitation, the following pipeline was created to provide communication between the two programming languages.

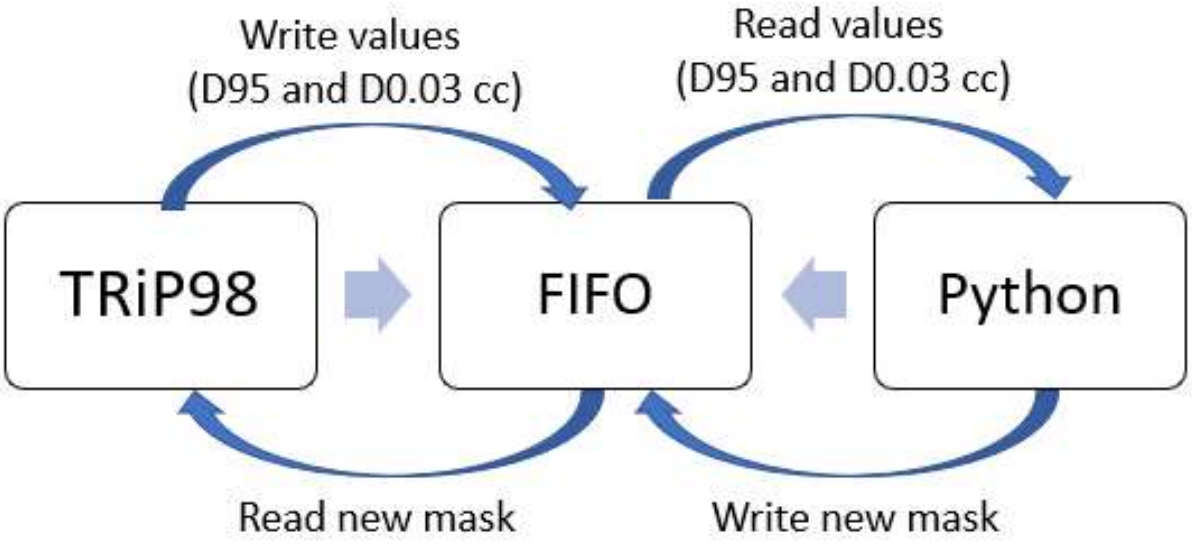



*Figure 6.1 Representation of the communication between TRiP98 and Python environment*

### 6.2.1 SSH

The data is located on a GSI server for the regulatory reasons. To enable access to the used dataset, Linux package SSHFS [93] was used. It uses SFTP to mount a remote filesystem. It allows to use the data from a remote serve as if it is located on your own PC. Hence, to read the data in Python script no additional packages were needed, the traditional path was enough to make all the necessary operations.

### 6.2.2 TRiP98

TRiP98 is a research software for Treatment Planning for Particles [11], [12], which is written in C and constantly upgraded. The main workflow can be described with the following flowchart in Figure 6.2.

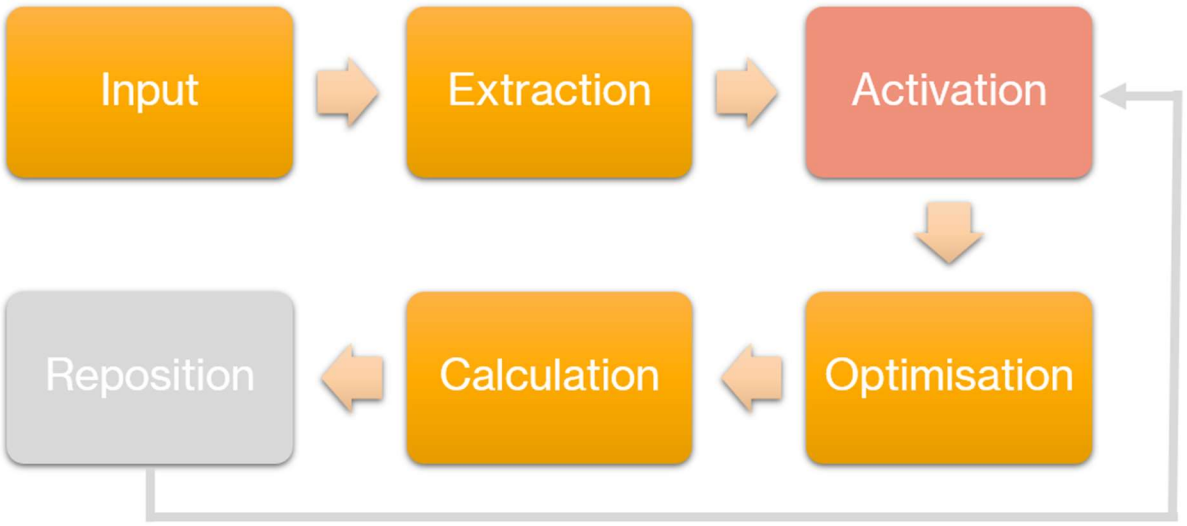



*Figure 6.2 Flowchart of the processes in TRIP98*

There are fixed parameter inputs to TRiP98, i.e., those which are the same for all patients, and parameters that vary for each patient. As fixed input, TRiP98 requires some information on the beam delivery setup (for example, the distance between pencil beam scanning magnets and the isocenter), and a set of basic data for depth dose profiles and pencil beam lateral profiles in water at different beam energies to compute the dose, particle spectra for LET and RBE, as well as RBE tables for different tissues. As patient specific input it requires the planning CT scan data and, separately, data which contains information about segmented organs and their offset in the CT scan data.

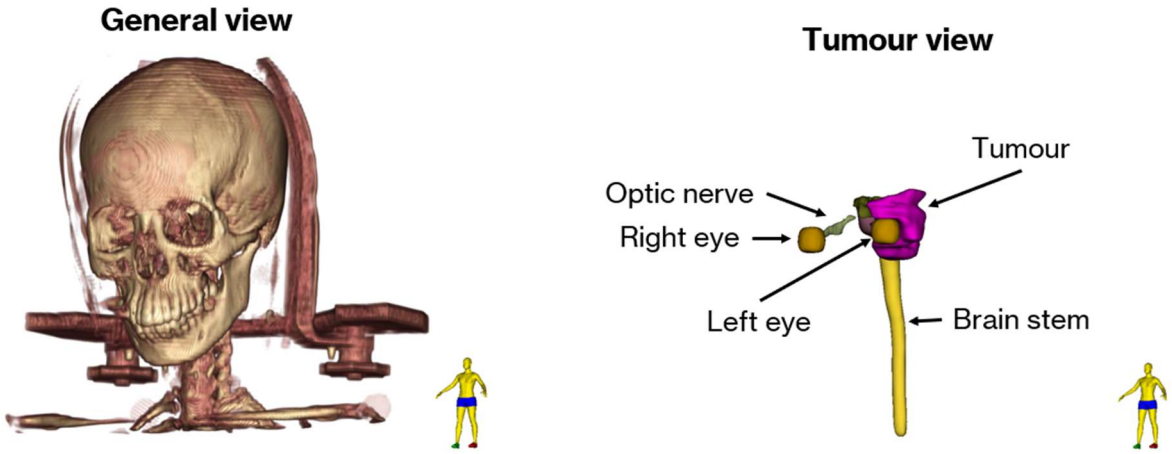



*Figure 6.3 Example of 3D representation of CT data (left) and segmented organs (right)*

The number of treatment fields and their incident angle, which is defined by position of the couch and the gantry, are also required, just as the lateral and longitudinal sampling steps between individual pencil beams used in the plan optimization. The user also has to set the target and the planned target dose, as well as additional parameters like sampling density for each VOI (see 3) and the weight factor and maximum dose, which define the importance of OARs in the plan optimization.

After all data is provided, a raster grid (see more in 1.2) is generated, which determines the lateral extent of the treatment field and the range of beam energies needed to cover the tumour in depth. Then, the influence of each grid spot position to the target and OAR voxels is calculated.

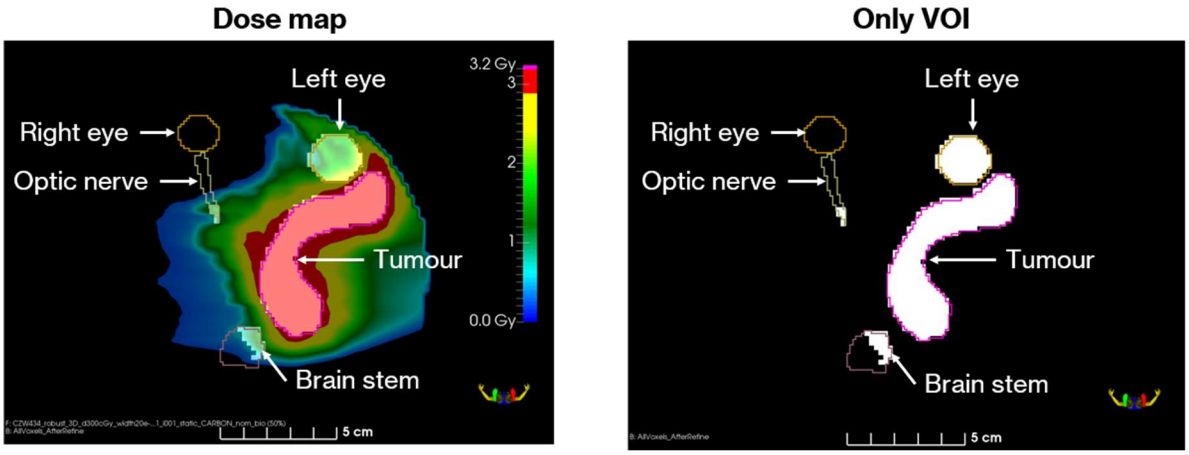



*Figure 6.4 Example of an optimized plan (left) and refined binary mask (right)*

Voxels which receive no or insignificant dose are removed to increase the computational efficiency of the plan optimization. In Figure 6.4 we can see that the right eye and a part of its optic nerve as well as a part of the brainstem are out of the beam range. Therefore, the associated voxels can be excluded from the matrix. Afterwards, the random subsampling of the voxels in each VOI with user defined parameters is applied (see 3).

In the optimisation stage, the number of particles to be delivered at each raster point which would result in the optimal target dose is iteratively calculated using gradient descent methods.

While the reduction of voxel numbers is efficient for optimisation, it prohibits to accurately calculate the target and OAR dose at the end of the process. In the standard TRiP98 workflow, a so-called forward dose calculation is used to evaluate the dosimetric outcome. This essentially recalculates the entire dose matrix, which would be inefficient for an iterative process as AI training. Therefore, the entire dose matrix, including the refined voxels, is kept in memory. This permits to rapidly calculate the dose. This is especially efficient if multiple optimisations on different subsets of voxels are carried out.

To make the software able to interact with the used deep learning method written in Python and speed up the time, spent on the optimisation, Professor Graeff implemented this new option for TRiP98. The whole matrix which is needed for the dose calculation is kept in RAM. A valid sub-matrix containing only the selected, active voxels for a specific instance of either the random mask or a CNN input can be rapidly extracted. The full matrix is furthermore needed to calculate the target and OAR doses. Storing the full mask therefore saves time both for an iterative optimization of different sets of SV as well as the required output calculation to evaluate the given set (Figure 10.3 in 10.2 Additional material)

### 6.2.3 Communication between TRiP98 and Python: FIFO

Since TRiP98 cannot easily feature a direct implementation of the network, as outlined above, communication and synchronization between two independent processes is necessary. A FIFO was chosen in this work. FIFO is an acronym for “First-In First-Out”, sometimes also called a “named pipe”. The main idea is that the system allows two programs to communicate through a pipe. To start the communication one side should open the FIFO file only for writing, and the other one only for reading. If one of the partners is missing, the system allows neither reading nor writing. For that reason, that very system was chosen. It gives the guarantee that during the calculation processes the systems can synchronize the transfer of all necessary information without using an additional file for counting. Besides, it excludes the risk of reading outdated information. And the last but not the least is that both Python and TRiP98 (written in C) can use the file.

### 6.2.4 Python environment

Python version 3.8.10 was used to build the CNN and interact with TRiP98. The subprocess module [94] allows either to wait until the started process finishes or run it as a background process. It was used to start TRiP98 calculations for each patient automatically without using bash scripts or similar scripts to synchronize the actions between TRiP98 and CNN. The python module os [95] provides the main operation related to the creation of FIFO and new folders. Numpy [96], [97] is used to manipulate the masks from TRiP98 and related statistics is operated with Pandas package [98].

To build the architecture the package TensorFlow with Keras [90], [91] was chosen. For automatic tracking of the variables needed for the gradient calculation, GradientTape(), and its function gradient() to calculate the gradients of trainable variables was used.

The full list of used packages can be found in 10.1.

## 6.3 Pipeline

### 6.3.1 Epoch details

Because of the TRiP98 nuances described in 0 the dataflow in an epoch was built in the following way:

1. Shuffle all training data before start
2. Train the system for each patient as described in Pipeline

3. Switch to the next patient
4. Repeat step 2
5. Repeat steps 1-4

This concept was chosen to benefit from the streamlined plan generation and dose calculation when a patient is used several times in a single TRiP98 call as described above. Similar strategies are used in reinforcement learning, where each episode can be repeated either n times or until the algorithm terminates.

This approach increases the risk of overfitting the data, which is also inherent in the small dataset. As the training process was constrained by the duration of each TRiP98 calculation, this trade-off was deemed necessary.

### 6.3.2 System pretraining

It is important that the system can distinguish the VOI interior from the rest of the CT, where no voxels should be selected.

The system was pretrained first. The reason is that the masking technique cannot be used in our case (see more details in 5.1.3). The pretraining is aimed to reduce the number of voxels, which are selected in the area outside the VOIs. This area makes up between 71.3% to 97.3% of the entire volume. It is an inherent property of the complex geometry of the input as well as the need for a uniform input image size to the CNN.

The pretraining also allows the system to learn from the empirically received result of the optimal voxel selection.

The model was pretrained for 5 epochs using CE (Eq. 5.1) loss. As a label, the random mask with the following parameters was used: 1 for the shell and 8 for the interior volume (see more in 3). The model was trained with Adam optimizer [99] and the learning rate equals $10^{-4}$ for the pretraining stage and $10^{-5}$ for the training one.

### 6.3.3 System training

The algorithm is built to minimize the interaction with the other parts of the software setup (like TRiP98) and the complete workflow can be started with a single script. Each patient in each epoch follows the procedure described below:

1. Start TRiP98 to calculate the full dose influence matrix and the target dose on the initial binary mask

2. Wait for response from TRiP98
3. Expand the binary mask and distance map to image size acceptable for the convolutional neural network
4. Get D95 and D0.03 cc
5. Create a new mask and postprocess it back to the original resolution and format
6. Pass the new mask to TRiP98 to calculate a new plan and dose
7. Wait for the TRiP98 response and calculate gradients
8. Update the weights in the CNN

Repeat the steps 5 to 8 five times for each patient (except step 3, which is done only once per patient). This is depicted in Figure 6.5.

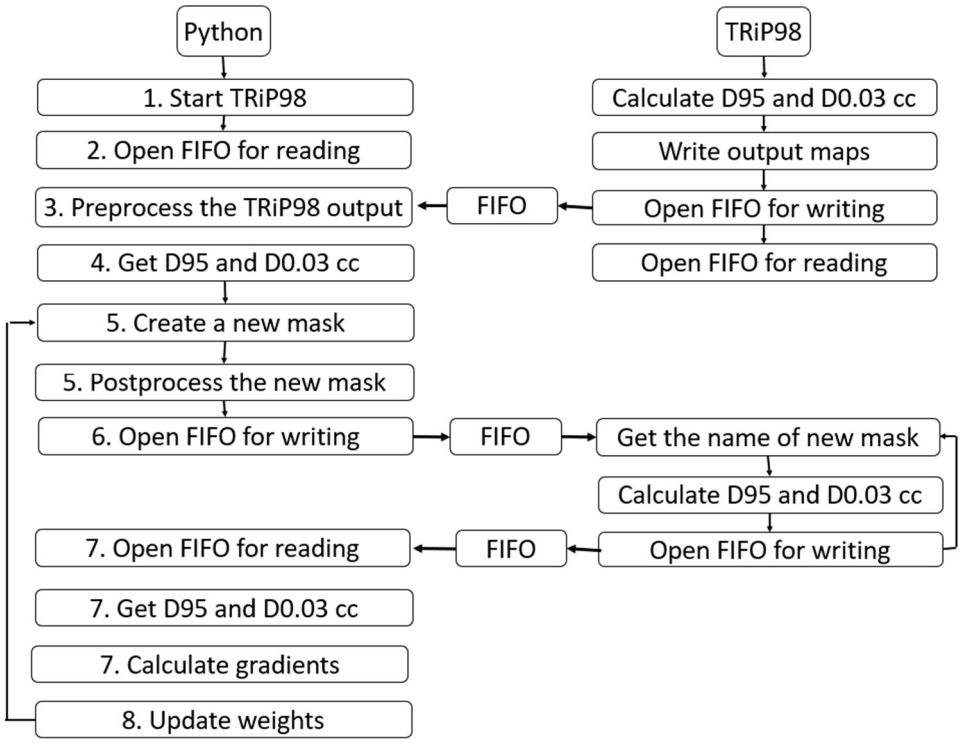



*Figure 6.5 Dataflow of interaction between TRiP98 and Python*

Each step listed in the process is explained in detail below.

**Step 1**. TRiP98 is started with the python script (further Python) for each patient separately. It calculates the full dose influence matrix and the target dose on the initial binary mask and write it into the FIFO file.

**Step 2**. Python opens the FIFO file for reading and waits for the planning target values (D95 and D0.03cc). It is necessary, because TRiP98 produces the data used for as input.

**Step 3**. The 3D-images ("cubes") are written in a special format consisting of two files: .ctx and .hed. The .ctx file contains a binary representation of the data and the .hed file provides the description of the parameters needed to interpret the binary data. For example, the length of the three sides of the cube, the data type (integer, float) and the order of bytes (big-endian or little-endian). The data is read and transformed into numpy nd-array [96], [97]. After reading the data has ZYX axis order, which is the typical order in the C language. The data should have the same order before being sent to TRiP98 for calculate D95. In the current work we did not change the order for the training process.

The label mask (see 4 for more details) is divided into two images: one contains only the target and the other one the OARs. The separation is realized with a label mask, where the voxels in the target always have a value of 2. After the separation the values higher than 0 set to 1.

At the stage of data preprocessing (see more in 5) all cubes were mapped to the same resolution 40x104x114.

**Step 4.** Read D95 calculated for the initial script (random mask)

**Step 5-8**. After the input data is prepared, it is used to get the new mask from CNN. The output has TensorFlow tensor format, and it is important to save it for calculation of the gradients.

The output has the shape of (1,40,104,114,2) where the first number represents the batch size (1 in our case), 2-4 positions represent the preprocessed shape of the input data and the last one is responsible for the action. Value 2 means that only 2 actions are possible: 0 or 1. The values in each voxel are probabilities to select one of the represented actions: 0 - the voxel will not be selected and 1 – the voxel will be selected. That representation of the data was chosen for two reasons:

- Firstly, it allows to extend the system in case the number of actions in the future is increased. Binary representation does not allow to change it quickly.
- Secondly, it permits to use the TensorFlow Sample() method to get the mask from probabilities.

We had to use sampling for the following reasons: the probability to select a voxel was never higher than 0.5 (see 7 for more detail) that means that argmax could not be used; the second reason was that the current method is a simplified Actor Critic system, which uses sampling. Sampling is necessary only after the pretraining process, where the TRiP98 interaction is started and there are no labels for the loss function.

After the output cube is sampled, it is necessary to postprocess the resolution back to the original one and remove the misselected voxels outside the VOIs, otherwise TRiP98 cannot use it for further calculations.

Finally, after the output is written into .ctx and .hed data format the FIFO file opens to write the name of the new mask. TRiP98 calculates new D95 values, which are used for loss calculation (see more in 5.1.3). Then the weights in the CNN are adjusted accordingly.

Then the process repeats 5 times in total and after that “STOP” is written into FIFO instead of the name of the mask to stop the loop calculation for the patient.

# 7. Results

The duration of one epoch of the pretraining stage takes about 2 hours. The pretraining of the system showed that it was able to reduce the misselection within the first epoch from 50 to 1-2 per cent and up to appx. 0.2 per cent after the fifth epoch. It also demonstrated that the system learnt to distinguish whether the voxel was related to the VOI or not. Moreover, it means that the system has potential to operate in various VOIs differently if the system is adapted accordingly.

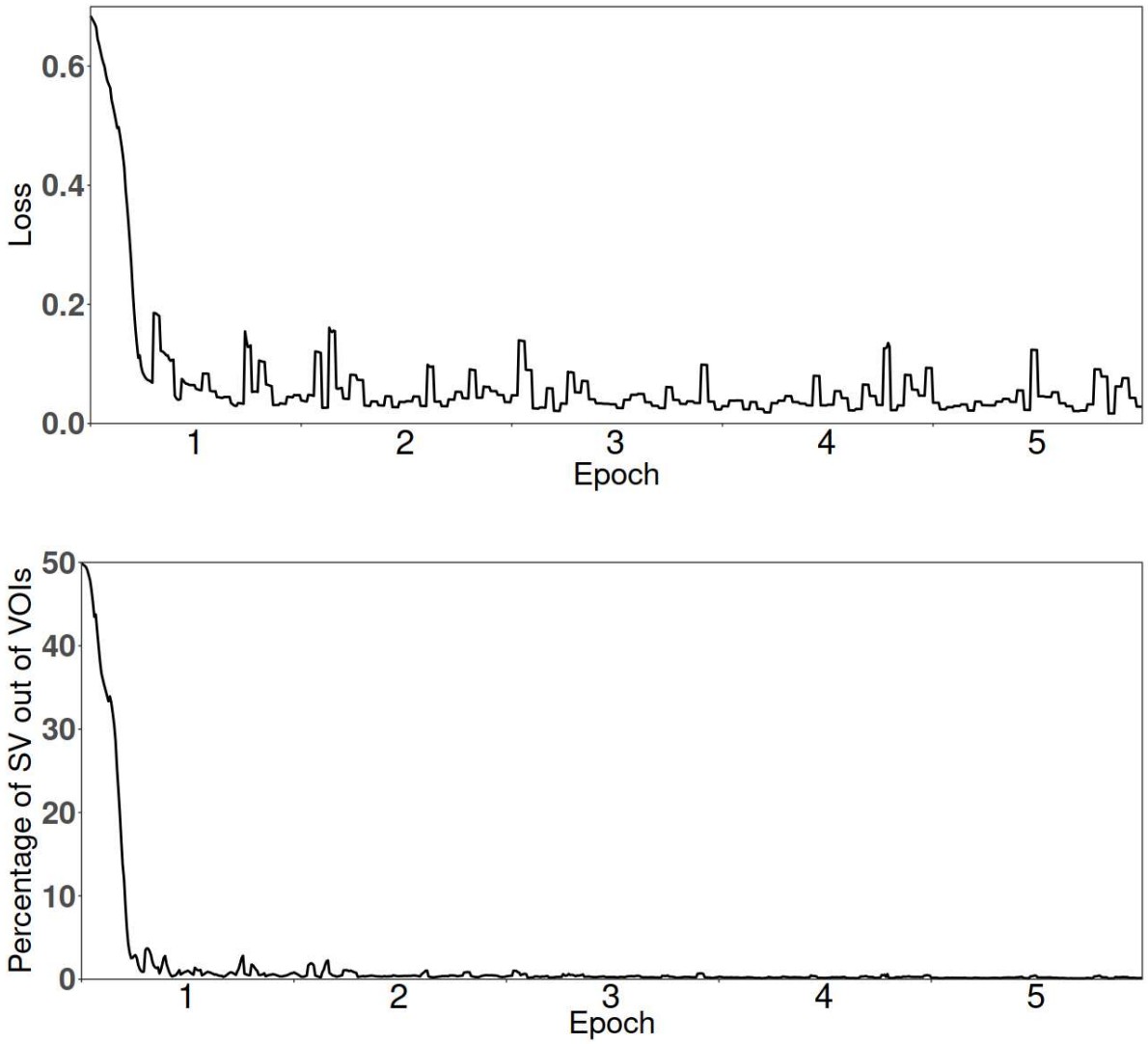



*Figure 7.1 Loss (top) and percentage of SV out of VOI (misselected voxels) (bottom) in the pretraining stage*

The reduction of the loss in the pretraining stage is partly correlated to the misselection. The jumps of the loss function (Figure 7.1 (top)) demonstrate that the system cannot find the pattern in the random masks, for example the selection of all voxels in the boundary shell. In Figure 10.4 (10.2 Additional material) we demonstrate the loss function of the pretraining done with

the full mask as label, where the loss function decreases to appx. 0.003 at the end of the pretraining.

After seeing the effect of the pretraining, we will have a look at Experiment 0 to assess whether further training of the model with the same loss function could improve the results received. One epoch takes on average between 7 and 9 hours for the training and 1 hour 15 minutes for the testing. The training time strongly depends on the following factors: the connection speed to the server, where all patient's data (CT scans, information about the VOIs) is saved, and the number of voxels in the mask.

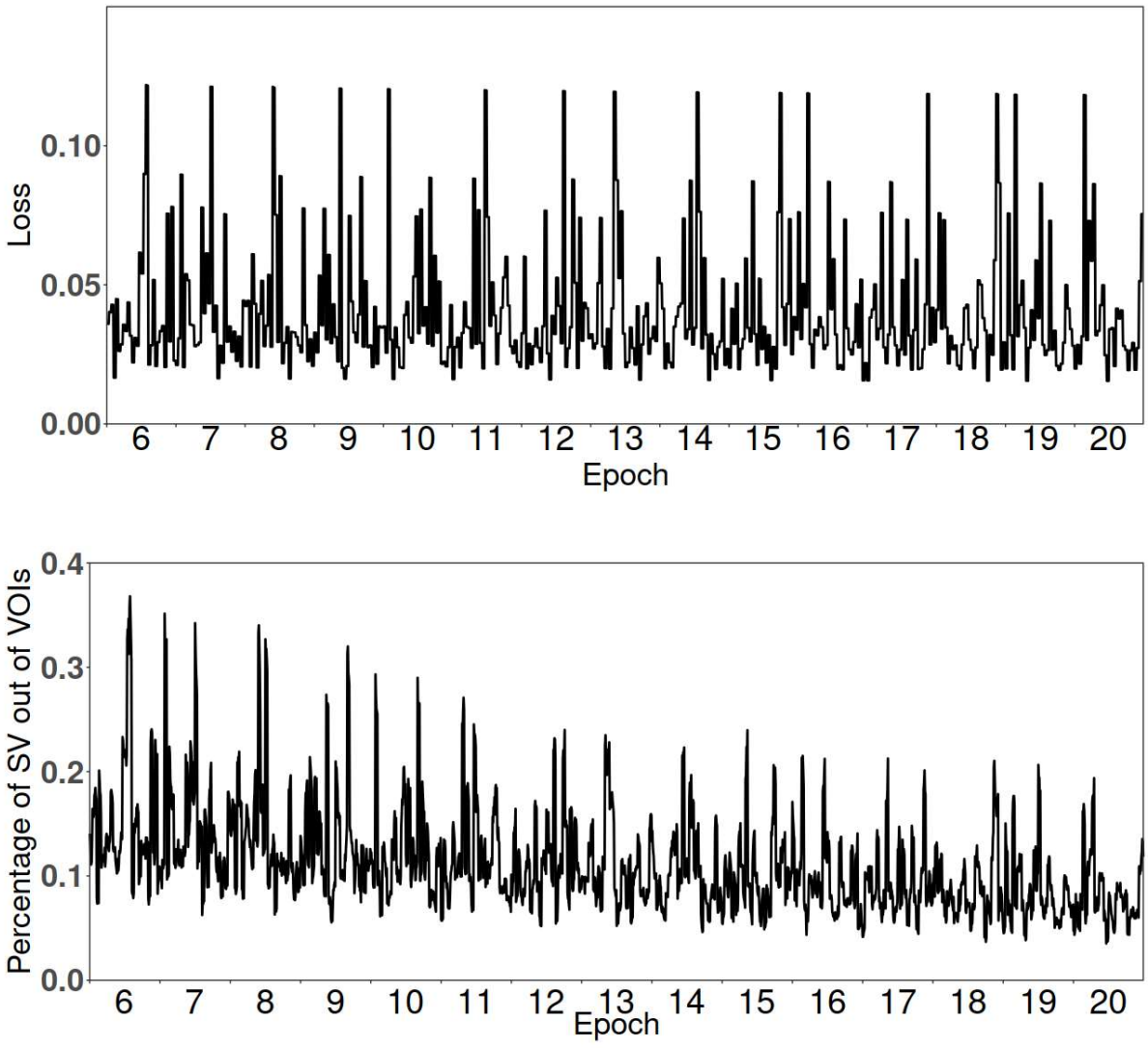



*Figure 7.2 Loss (top) and percentage of misselected voxels (bottom) for Experiment 0*

In Figure 7.2 (bottom) we can see a slow improvement in the distinguishing of the VOIs from the space outside the VOIs, while the loss function (Figure 7.2 (top)) does not show improvement with the time. The data on the same patients caused the highest loss in all the

epochs. The same patients had the lowest percentage of SV in random mask in the train dataset. Hence, it is questionable, whether a significantly longer pretraining of the model could deliver a better final result.

Figure 7.3 represents the data distribution for percentage of SV (top) and D95 (bottom) received from Experiment 0. The system selects less voxels than in the random mask during the whole experiment, while the range of D95 stays close to the values of the random mask. The difference between the values is in the range from -0.5 to 1% (Figure 10.5 in 10.2 Additional material)

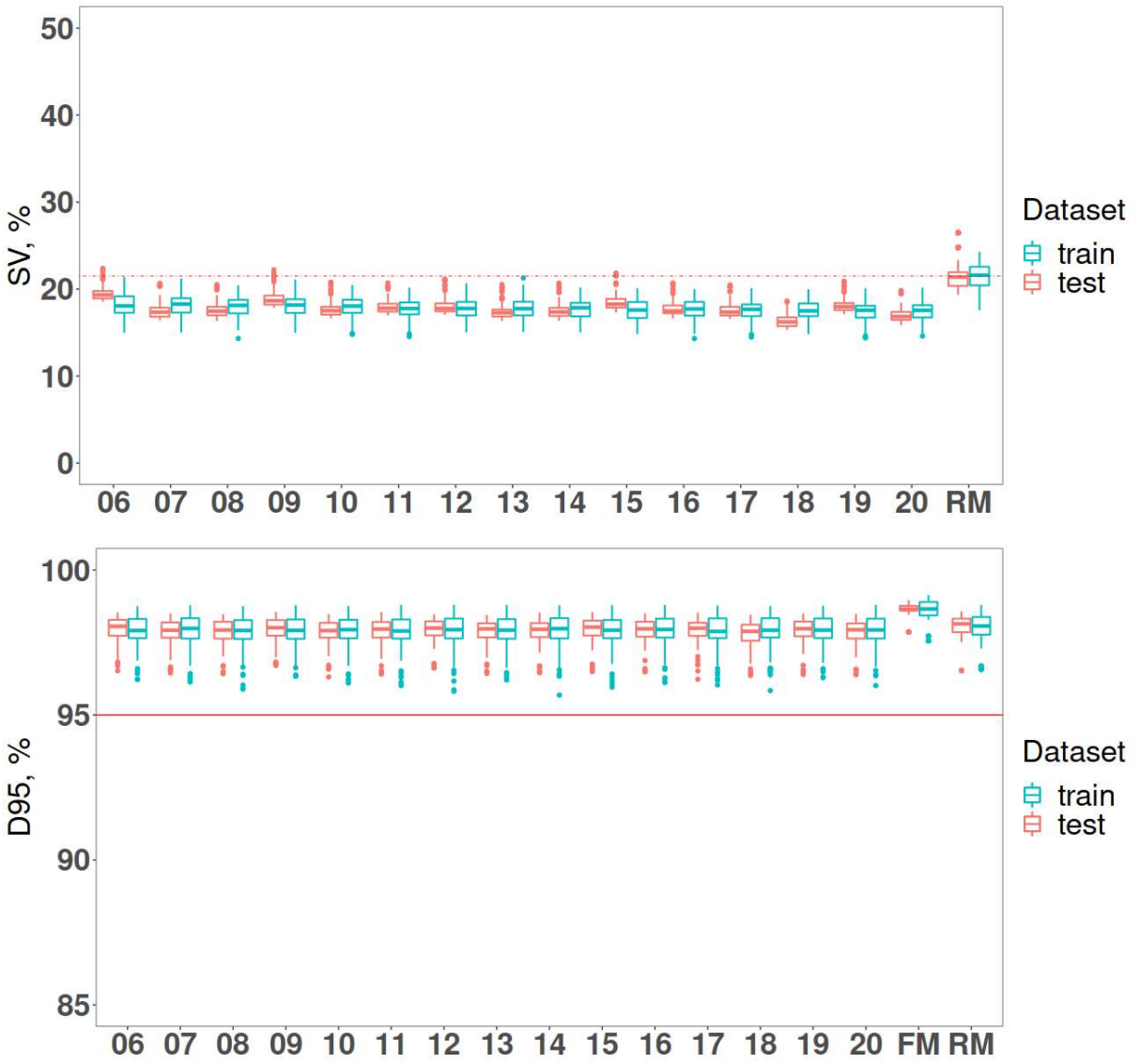



*Figure 7.3 Distribution of the SV (top) and D95 (bottom) value in epochs, full mask (FM) and random mask (RM) in Experiment 0*

The outliers in D95 from the random mask training database correspond to two patients who also appear as outliers during the training in the other epochs. Figure 7.4 and Figure 7.5 show the data from a more detailed analysis for these two patients. D95 and the percentage of SV is displayed with the grey dashed line; the red solid line shows the threshold for D95. As we can

see, both patients have a similar "starting point", but for patient CBS303 a lower number of voxels was selected, and D95 remained slightly underestimated.

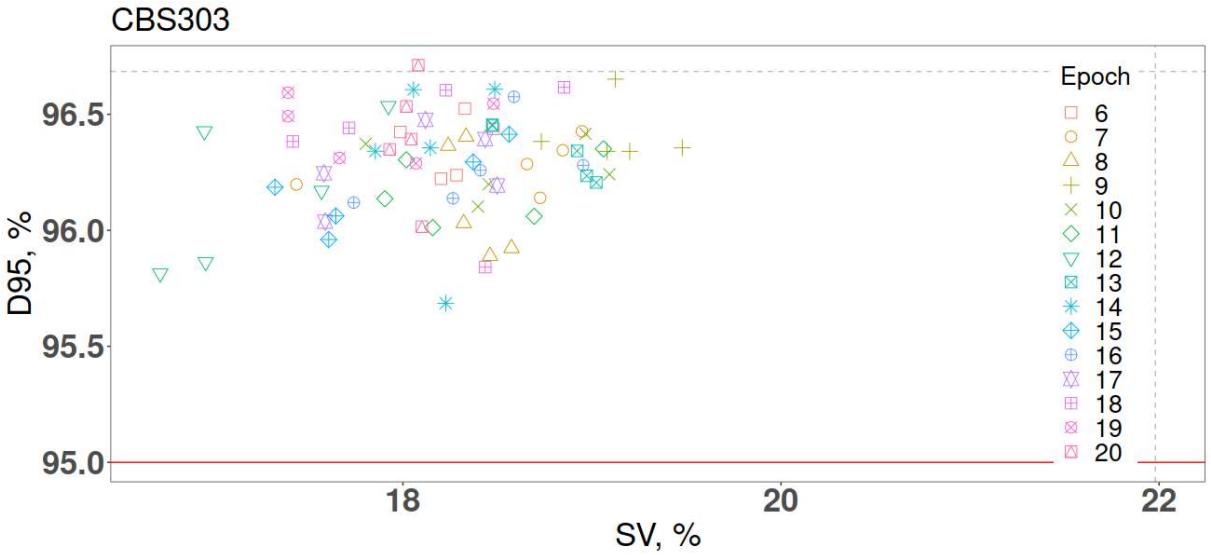



*Figure 7.4 History of the values produced during the training for patient CBS303*

At the same time patient CFW221 (Figure 7.5) demonstrates better results with lower percentage of SV in comparison with the random mask and better dose distribution in the target.

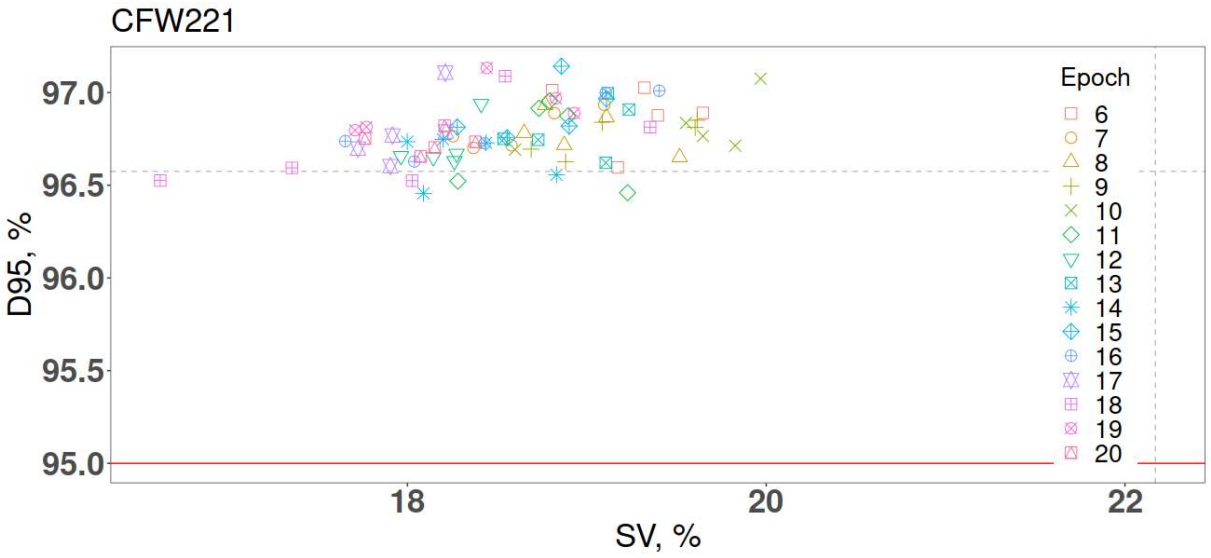



*Figure 7.5 History of the values produced during the training for patient CFW221*

The patients have different number of voxels in the tumour volume: 15823 for CBS303 and 25485 for CFW221. It can be caused by a lower probability for patient CBC303 than for patient CFW221 to select a voxel in the boundary shell (Figure 7.6).

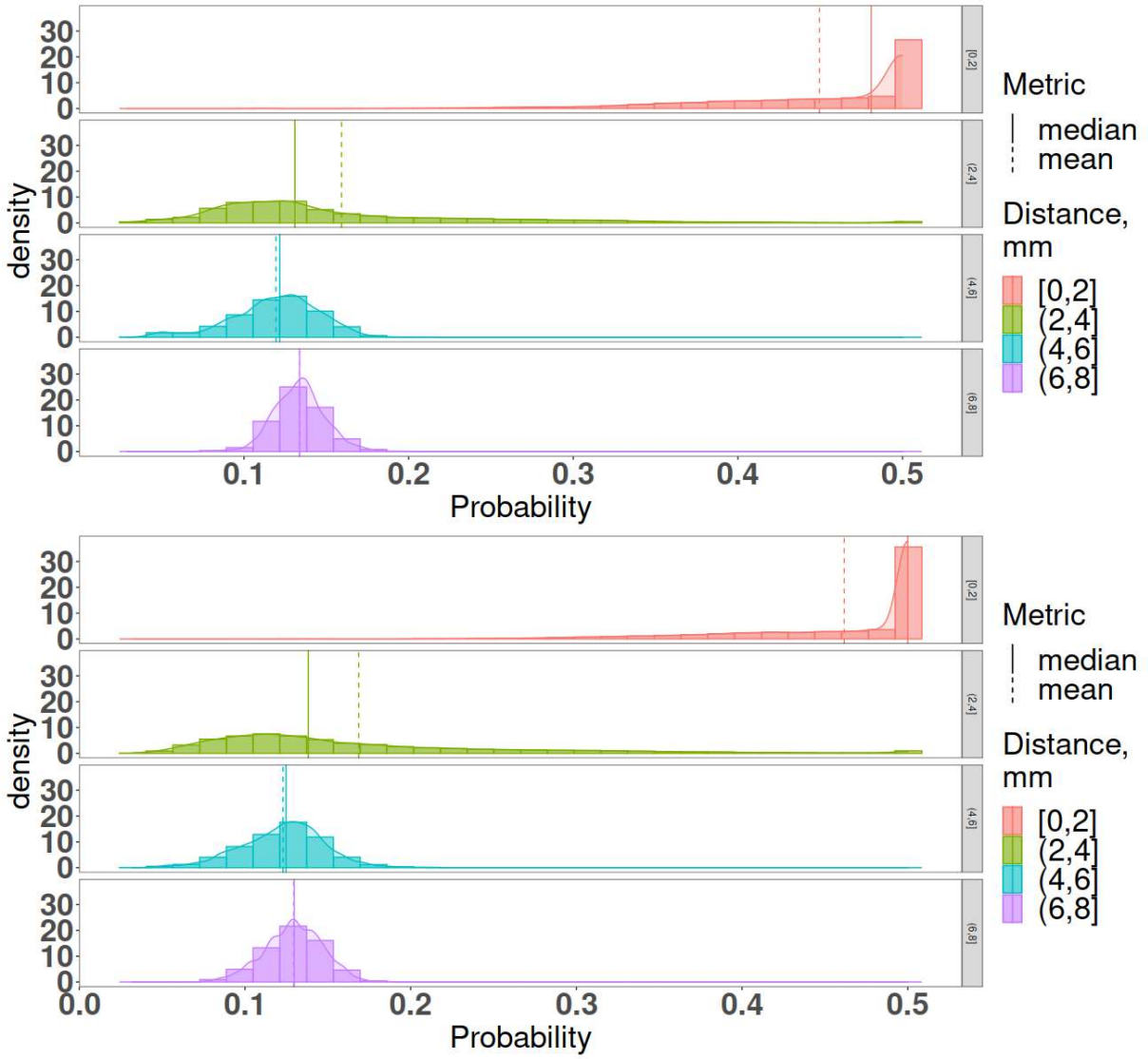



*Figure 7.6 Density plot of distribution to select a voxel for CBS303 (top) and CFW221 (bottom)*

Experiment 0 demonstrated a good approximation of the random mask and absence of an overfitting effect in the test dataset even though we had only 30 patients, each involved 5 times. It is represented by similar results between test and train dataset for all the epochs (Figure 7.3).

For better understanding of the inner process of the system the distribution of the probabilities depending on the distance was plotted. In the first bin up to 2 mm in depth we can see a shift of the mean towards 0.5. That means that the system starts to understand the importance of the boundary shell. We can clearly see that the probability has the maximum value at 0.5, which is not the same for the full mask. It can be caused by the fact that the boundary shell is thin (in our case only a one voxel layer) and the system needs either more time for the training or some changes in the architecture to work better. At the same time, the variance in the other bins decreases and the mean and median move to the value of the random mask (0.125).

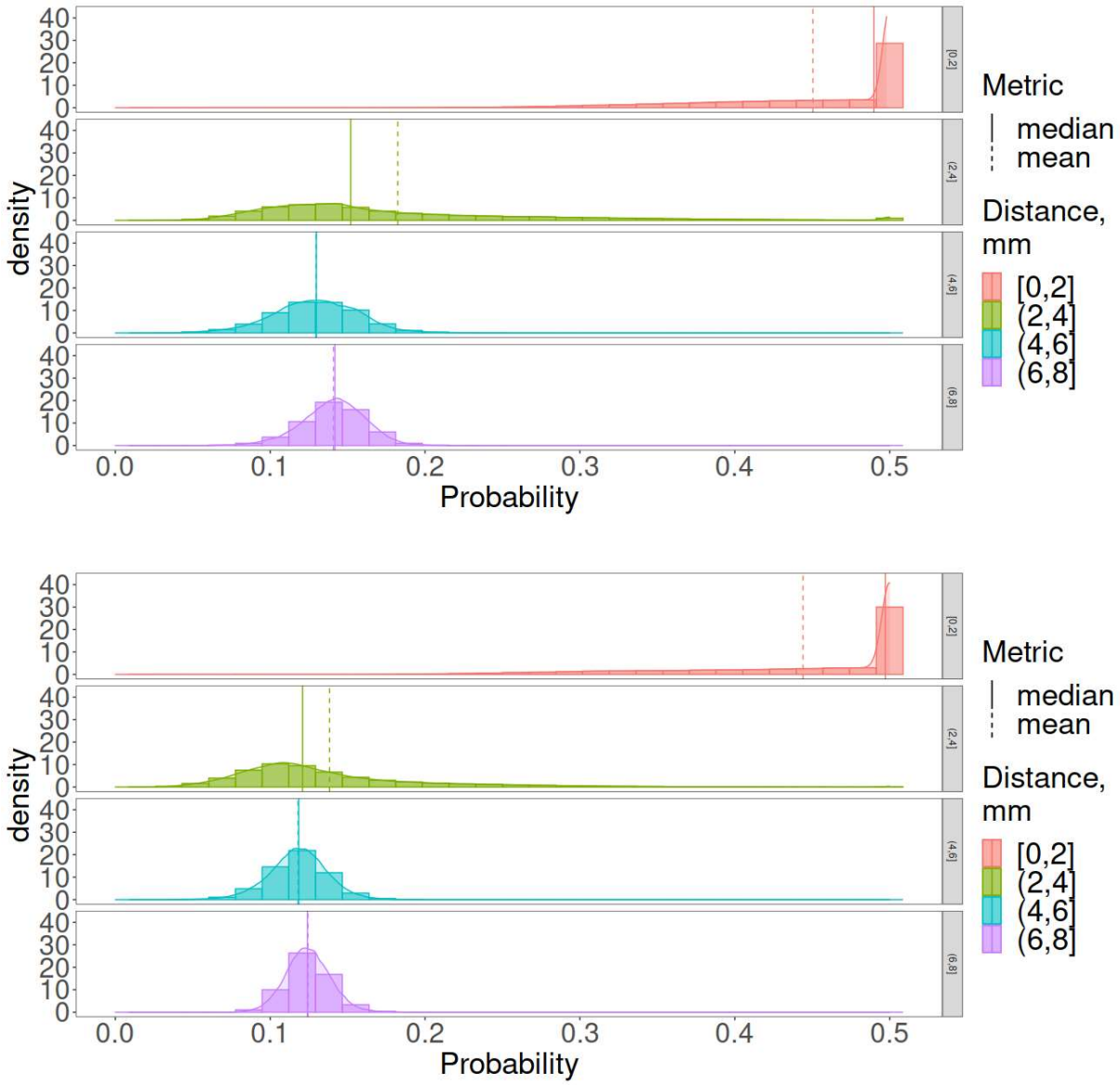



*Figure 7.7 Distribution of probabilities in different bins for epoch 6 (top) and 20 (bottom)*

To check the reproducibility of our own method the training from 6$^{th}$ to 20$^{th}$ epochs with our own loss function (see 5.1.3) was performed 3 times.

The final loss of the function depends on two parts: the CE loss, which also correlates with the number of misselected voxels and the positioning of the voxels in the tumour. Positioning has the highest impact on the loss, because the independent variable is correlated to the dose distribution and percentage of SV (see more in 5.1.3).

There are no perceivable similarities in the loss values over all the experiments (Figure 7.8). This is partly an effect of the fact that each experiment started with a different order of patients for the epoch (Figure 10.6 in 10.2 Additional material) because some of patients' data is less sensitive for undersampling than other. The high peaks are caused by high underdose of the target, as it was set in the loss function to penalize strongly if D95 is much lower than 95%.

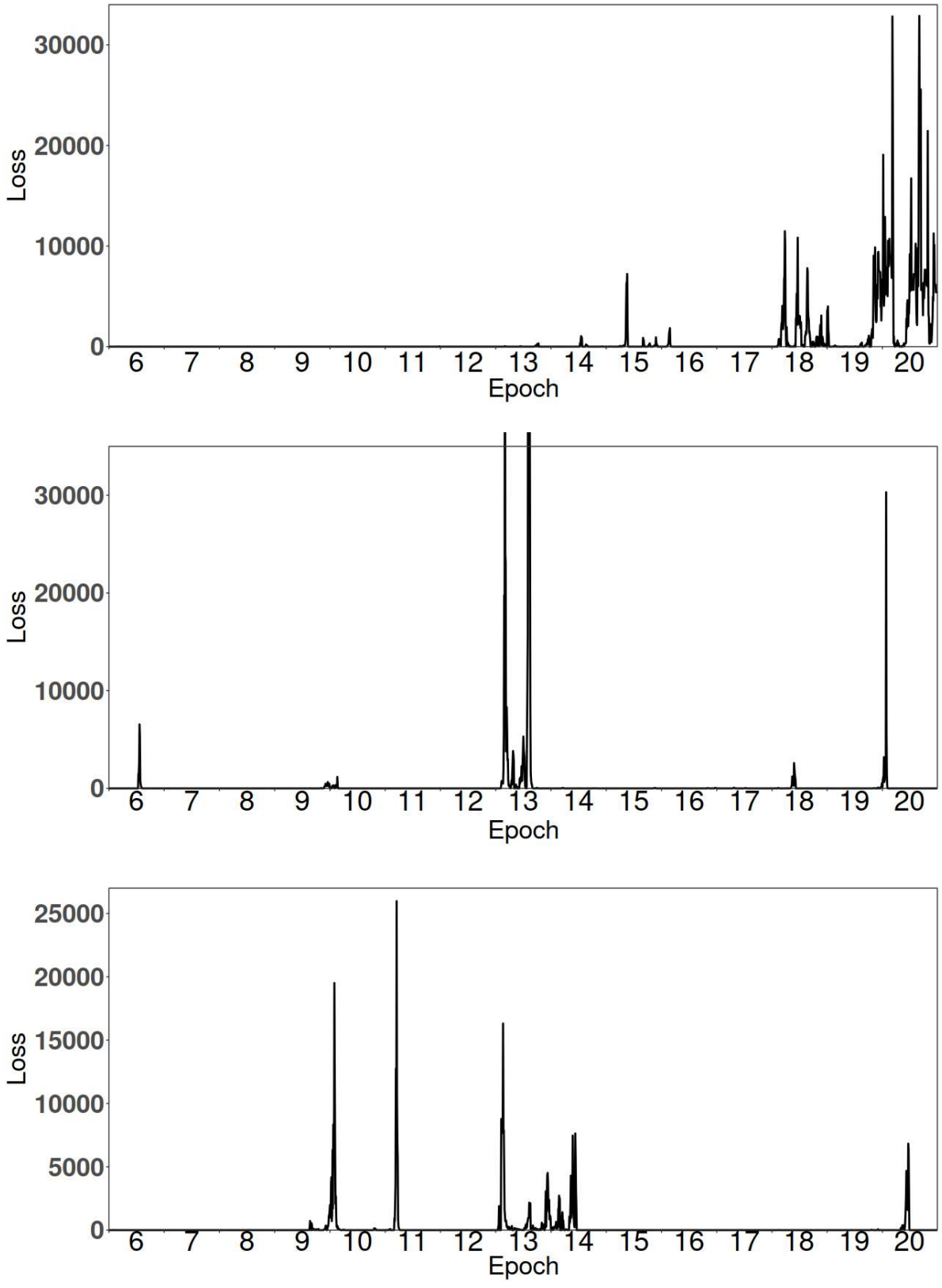



*Figure 7.8 Loss in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)*

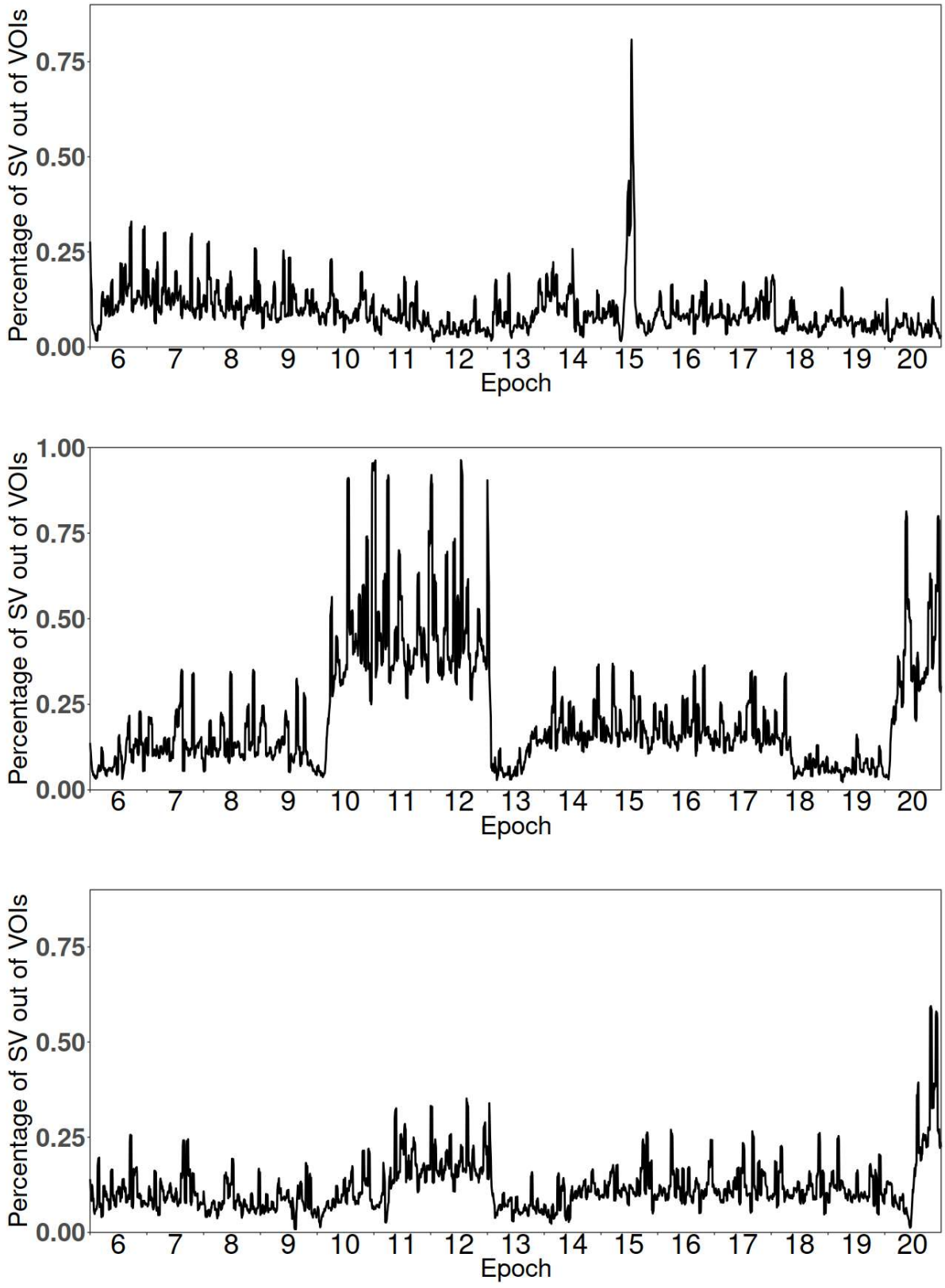



*Figure 7.9 Percentage of SV out of VOIs (misselected voxels) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)*

There is not enough data to claim that the system improves the misselection with the training. For example, only the first experiment showed the same trend as the experiment with only CE loss (Figure 7.9). In contrast, Experiments 2 and 3 showed a different behaviour compared to Experiment 0 and Experiment 1 but were similar to each other. For example, both exhibit jumps in the percentage of misselected voxels for epochs 10-12 and 20, although scaled by a factor of 2.

Figure 7.11 represents the median of D95 distribution in the three experiments for better visualisation close to each other. If we look only at the behaviour of the dose distributions in the three experiments, we can find the following pattern: 2nd and 3rd experiments have similar behaviour. The median decreases in the 9th epoch, then rises again just to decrease again in the 13th epoch, which repeats in the 19th epoch. Experiment 1 shows different behaviour of the data from the other two, but it also has a "wave"-like pattern and drops in the last three epochs towards high target underdosage.

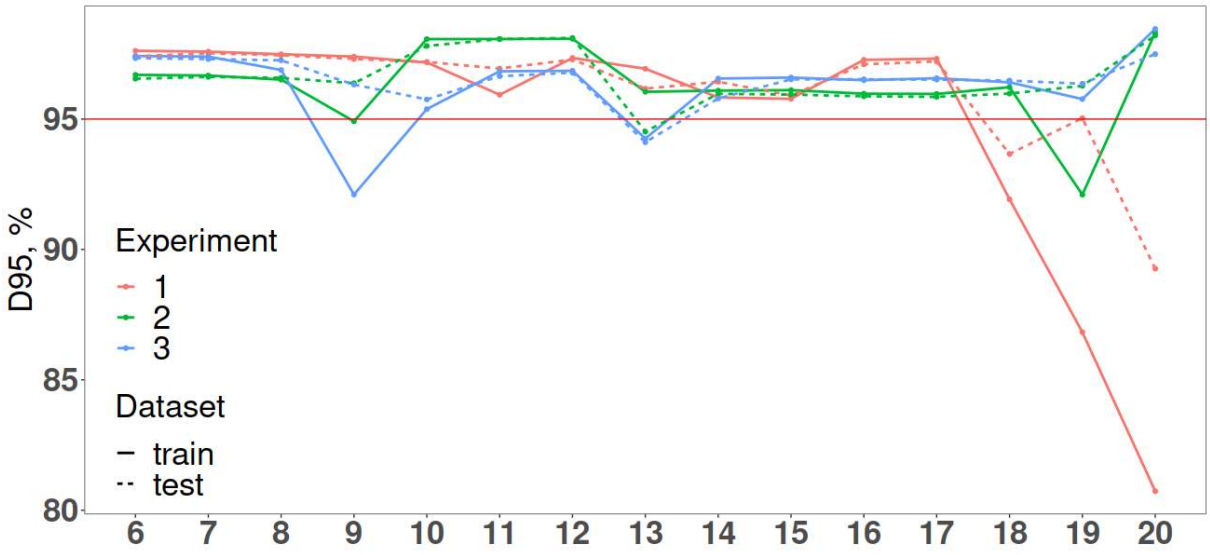



*Figure 7.10 Dynamic of the median D95 during the training*

Full data can be seen in Figure 7.11. All the experiments showed D95≥95% in most of the epochs for at least 75% of the data. We can observe a difference between train and test dataset in the last epochs of Experiment 1: The test is based on the last weights of an epoch, in which the system got updated after the last step. The difference is therefore pronounced when the training shows dynamic changes within the epoch, for example at the end of Experiment 1. This is typically the case when the number of selected voxels changes drastically as it can be seen in Figure 7.9.

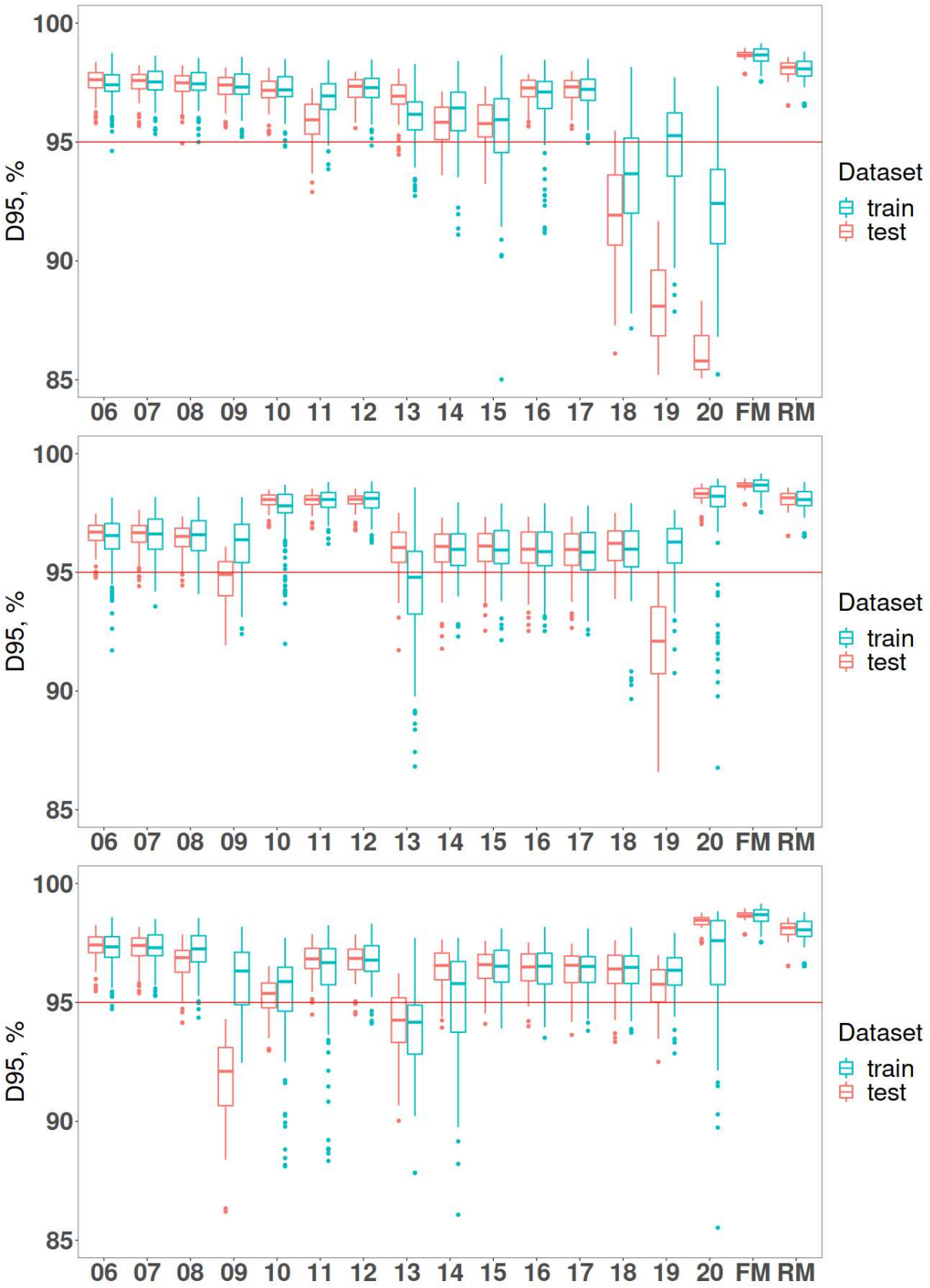



*Figure 7.11 Distribution of the D95 value in epochs, full mask (FM) and random mask (RM) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom)*

Next, we investigated the behaviour of the distribution of the percentage of SV as a function of epochs. Figure 7.12 shows again a similar behaviour for the 2nd and 3rd experiments in the training process.

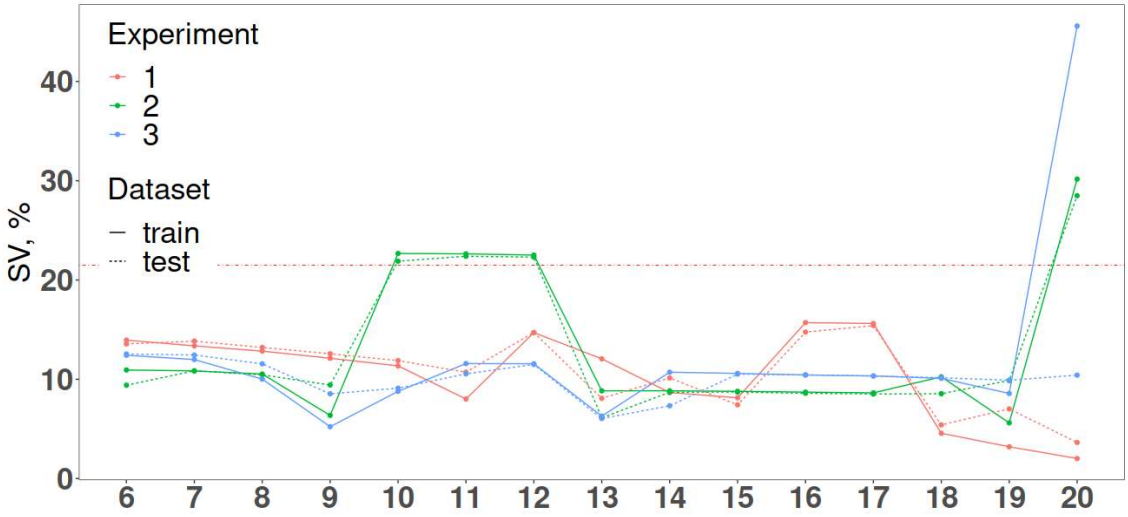



*Figure 7.12 Dynamics of the median percentage of the selected voxels in the target during the training*

Detailed representation of the data can be seen in Figure 7.13. All the experiments show the trend to reduce the percentage of SV in comparison with the random mask. The system shows stability after applying it to the test dataset. There is no big difference between the percentage of SV in the train and test database. The stability is related to the probabilities resulted after the softmax layer. But in comparison with the previous method the system is more flexible to change the probability for different regions. Hence, without practical results, it was not obvious that the system could demonstrate such stability especially when it was trained and tested on a small and very inhomogeneous dataset.

To see the whole picture of the result it is necessary to see the correlation of changes in D95 and percentage of SV. As we could see in Figure 5.8 the lower percentage of SV, the lower D95. It becomes more difficult for the optimizer in TRiP98 to negotiate the gaps in the mask and fulfil the constraints set in the script.

In Figure 7.14, we provide some details on the "history" of the median of each value per epoch for the test and train datasets to interpret the model convergence behaviour better. We can see that the system preformed similarly to the random mask for the D95 value, having less SV in comparison with it. We do not compare the results of the system with the full spectra of the random mask results for different sampling parameters, because this work was aimed to find a system which can understand the final goal (D95 !< 95%). We could see the desired results in most of the epochs in all the experiments.

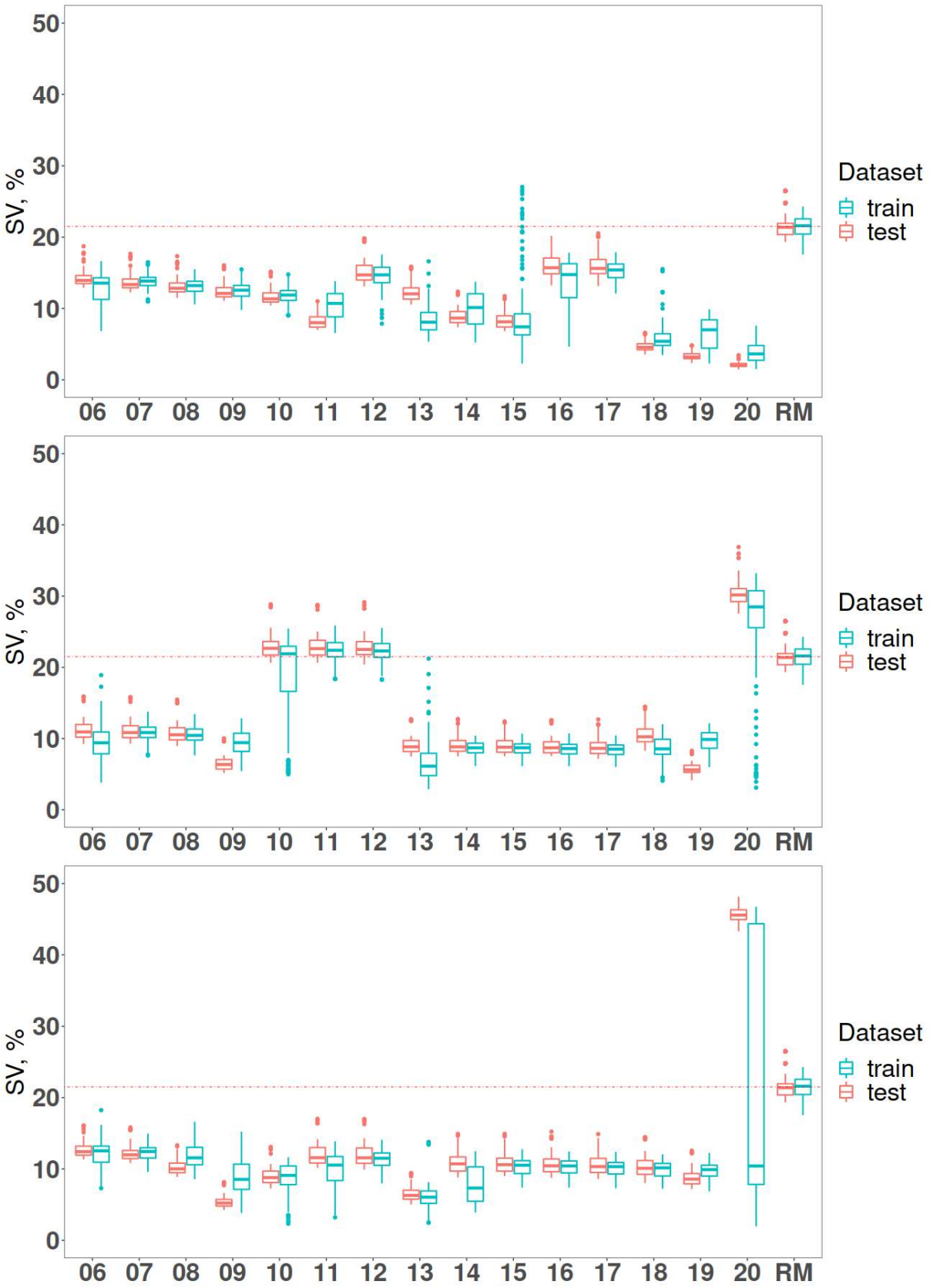



*Figure 7.13 Distribution of the percentage of the selected voxels in the target in epochs, full mask (FM) and random mask (RM) in Experiment 1 (top), Experiment 2 (middle) and Experiment 3 (bottom). Red dashed line is the median of random mask medians for both datasets.*

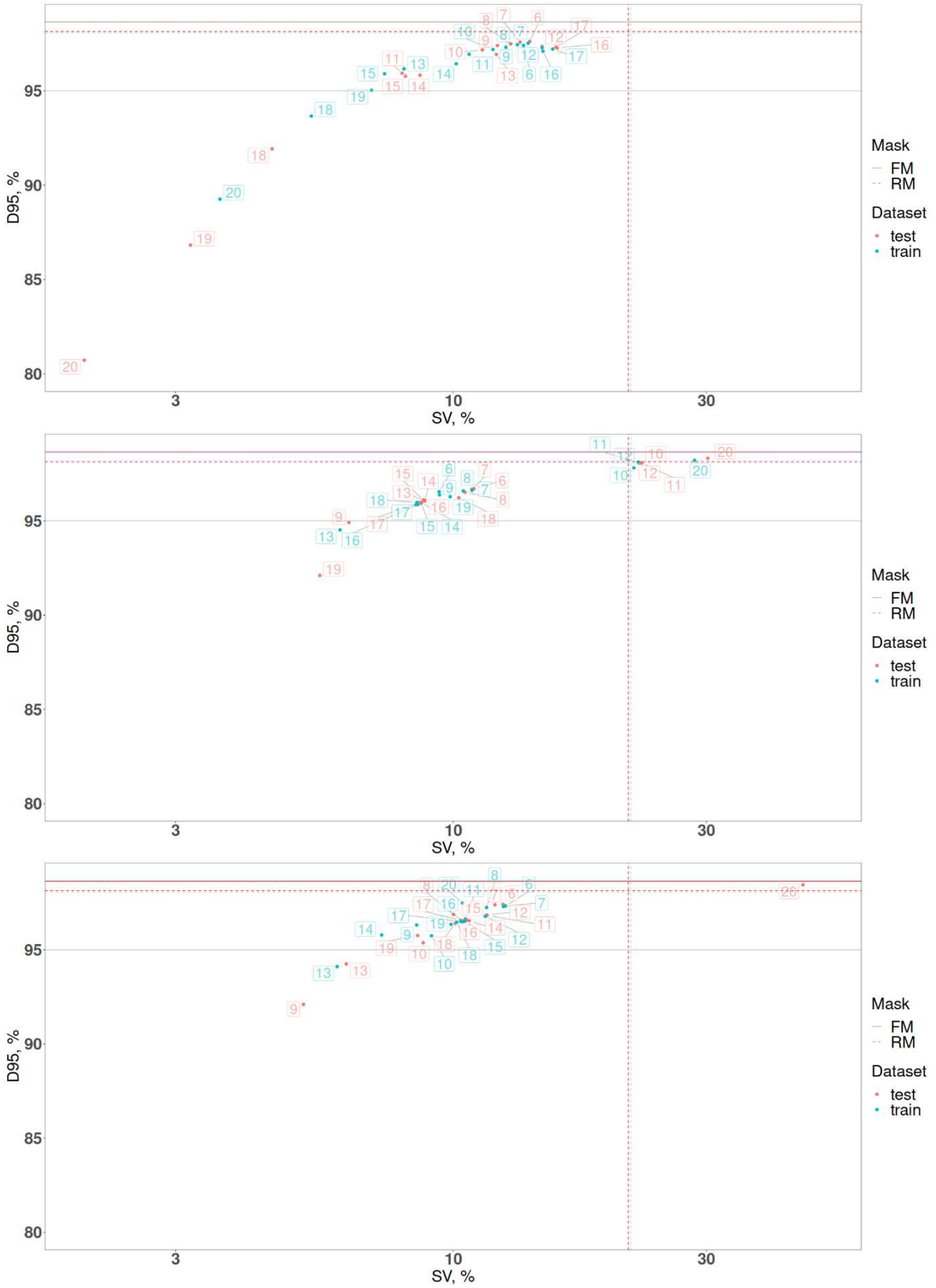



*Figure 7.14 Representation of the median percentage of the selected voxels (X-axis) vs the median of D95 (Y-axis) for each epoch in Experiments 1-3. The grey line indicates the threshold of 95% for D95.*

For a detailed investigation of the observed behaviour during the training we take a closer look at the 3rd experiment. In some epochs (like epoch 9) there is a big difference between the train

and test dataset for both (percentage of SV and D95), because at the end of the training for the epoch the system got a patient, for whom the tumour was underdosed because of the suboptimal positioning and number of SV in it. So, the last update of the weights in the architecture was only a few steps forward or back from a better result. The whole test dataset was calculated with the final weights of the epoch, which were considerably different from some of the weights used in the training steps. This caused such a large difference in the outcome.

It is clearly seen that the system recovers in the following epochs until epoch 13, where during the whole training process the median percentage of SV is around 6%. This is the lowest result for the experiment among all epochs and the target dose coverage in the epoch is also the lowest. The reason could lie within the modelling of the loss function. It has the minimum at 1% on condition that the D95≥95%, that forces the system to reduce the loss with the reduction of the voxels. After some training the system increases the probabilities to select the voxels and the threshold for the dose is reached in over 75% of the train dataset in epochs 14-18. The system decreases towards a 10 % sampling parameter but is perturbed again in epoch 19. One possible explanation for this behaviour is that the data shuffling for the epoch sets patient CBC303, who we previously identified as an outlier to the training data, as the last one into the training set (it is worth to keep in mind that each patient is gone over 5 times in each epoch). This causes the system to jump to higher selection probabilities at epoch 20, i.e., the observed behaviour.

During the training it was also noticed that the system often tends to reduce selection probabilities in all the voxels, i.e., reduces the sampling parameter, if it receives a high loss value from the unmet dose constraints. This is desired in the case when the dose constraints are fulfilled but this does not work in case they are not. Due to the design of the loss function, if the target dose coverage is lower than the dose threshold, the function has the minimum around 30%, that forces the system to increase the probability again and start the search for a better solution one more time.

Figure 7.15 shows the distribution of the voxel selection probability in different distance bins in epochs 6, 12 and 18. It can be seen that the tails of the distributions are wider for the later epochs. In epoch 6 we can still see the “effect” of the pretraining, so most of the data is concentrated around a small range, but in the $12^{th}$ epoch the distributions become skewed with a lower probability in all the bins. However, after the $18^{th}$ epoch the probabilities still stay more spread than in the epoch 6, while the values of the mean and median are very close.

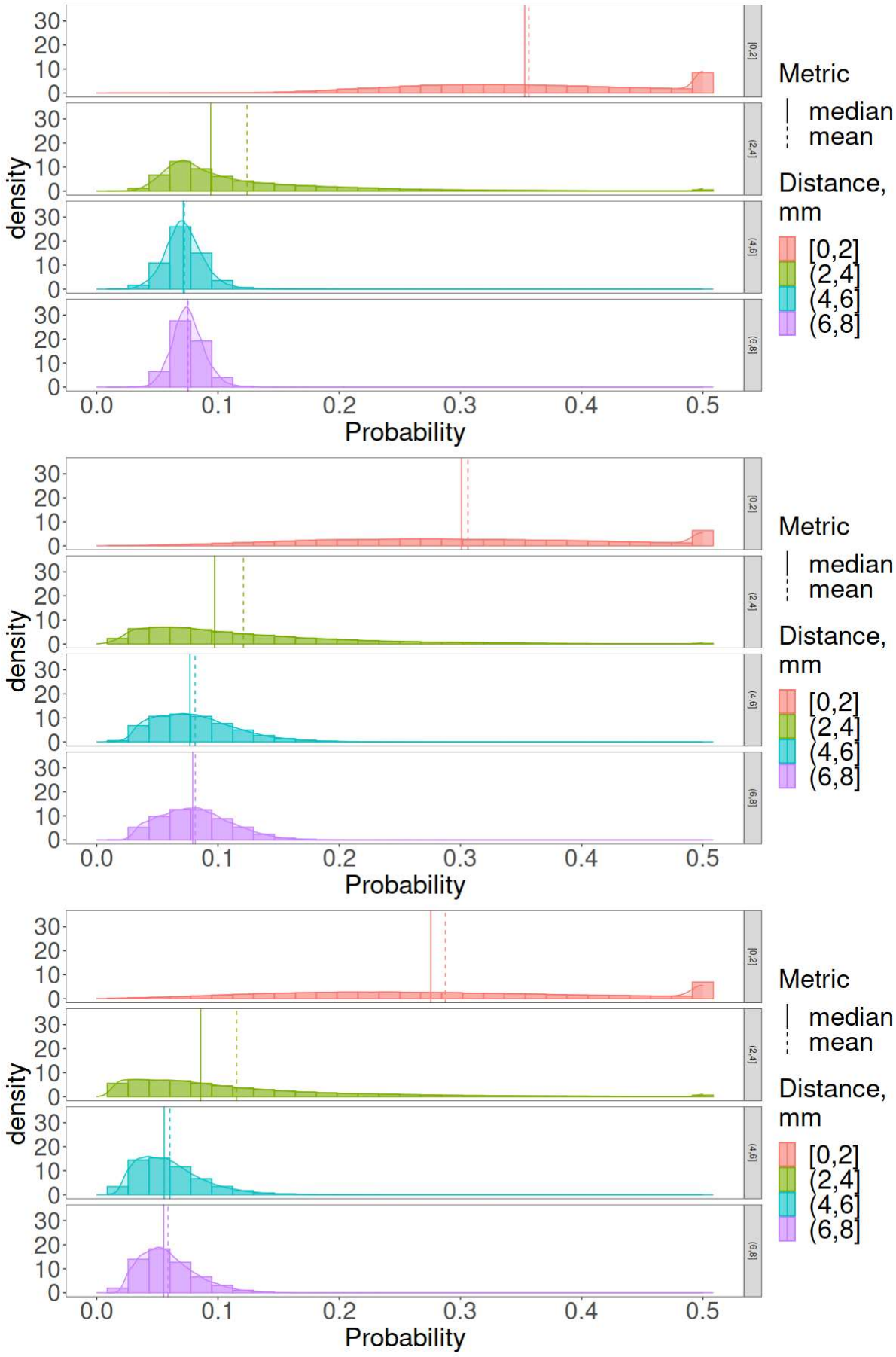



*Figure 7.15 Density of probabilities to select voxel in epochs 6 (top), 12 (middle), 18 (bottom)*

A longer training would be necessary for a better representation of the data. But the high variance seen in the probability distribution is a hint at a possible underlying pattern, identified by the model, which results in higher selection probabilities in some areas seemed more important than the others for plan optimization. While in-depth investigation of such possible patterns is beyond the scope of this thesis, it should be studied in future works.

The visualization of the median dose vs median percentage of SV in an epoch for all the experiments shows that our loss function was able to get the dose coverage in the target similar to the results of the random mask in some epochs keeping the percentage of SV between 10 and 15. Remarkably, this presents only half of the voxels as used for the random mask, which is a great advantage in terms of computational efficiency of the plan optimization.

As mentioned above, training with only CE loss (Experiment 0) also showed the trend to reduce the number of voxels slowly but kept a high similarity to the random mask. Hence, the reduction of the sampling parameter at similar dosimetric accuracy shown above serves to highlight the validity of our self-designed loss function to further optimize the voxel selection process.

The "best" results for Experiments 1-3 though appear in the first epochs after the pretraining stage. As best result we define those where the percentage of SV is less that in the random mask and D95≥95%. Then the system starts to explore the environment and demonstrates a fluctuating behaviour in the dose accuracy and percentage of SV. Apart from that, there was no obvious reason found to describe the behaviour of the system in Experiment 1 in the last 3 epochs, where the system does not recover quickly after getting a high penalty as in the other cases.

With a closer look at individual patients' results we could distinguish some classes of the mask changes during the training period (Figure 7.16):

1. Horizontal, where approximately the same number of voxels results in a different dose outcome
2. Vertical, where approximately the same dose in a patient can be reached by a different number of SV
3. Curved as random mask, the behaviour is similar to the changes in the random mask: less voxels-lower dose

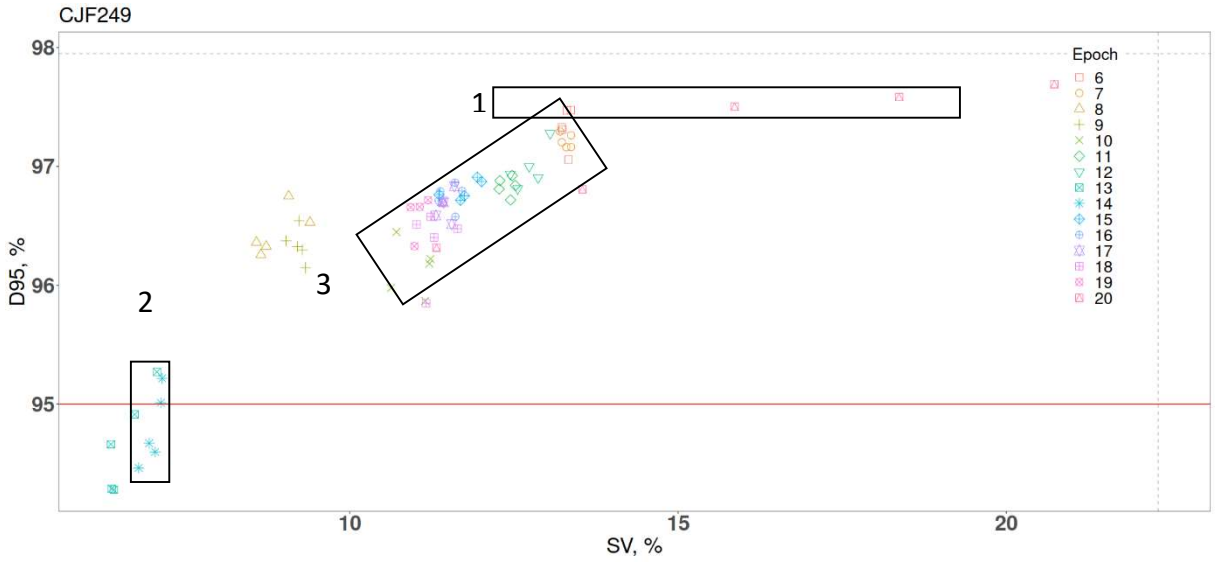



*Figure 7.16 Examples of all 3 types of mask behaviour for a patient (1-vertical, 2-horizontal, 3-curved)*

A strong variation of voxel activation during the epochs is illustrated in Figure 7.17. A strong decrease in the number of activated voxels leads only to a moderate decrease in the target dose.

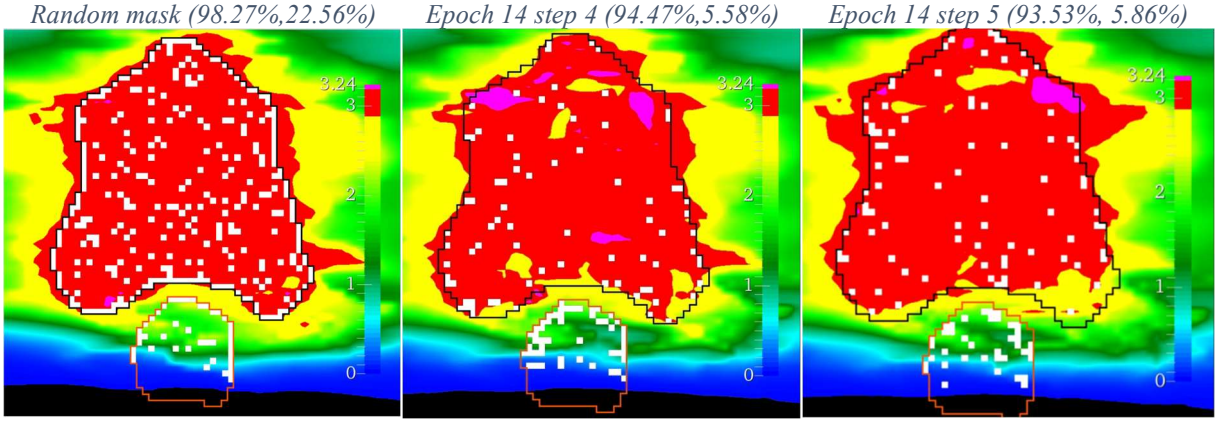



*Figure 7.17 Visualisation of the dose maps for the vertical change in the mask for the patient CUM330 (D95, SV). The colour scale on the left represents the dose and the white spots are SV. Black contour-target, orange one -brainstem (Produced with Slicer 8).*

Figure 7.17 demonstrates the vertical change of the mask. In step 4 (middle figure) of epoch 14 5.58% of the target voxels were selected, in the following step (right figure) - 5.86%. In the figure we can see the difference in positioning of these voxels which has a noticeable effect on the dose accuracy: despite selecting 53 more voxels in step 5, the resulting D95 was reduced by almost 1% (see more in 5.1.3). In this case, the target comprised a total of 18643 voxels. As the baseline comparison, the RM for the case is demonstrated on the left. The impact of the positioning can be clearly seen in the demonstrated example. The bottom part of the target (right Figure) has only very few voxels selected in the slice. The underdosed part has no contouring in the areas in the previous slice and no selected voxels in the following one. It causes the underdose of such a big area of the target, what is also proved by the decreased D95 value.

Figure 7.18 shows an example of the generated mask in different epochs of the training. We can see the same wave effect as described above. It is hard to estimate the similarity of the data in 3D value manually to say which part of the target had a lack of voxels and which was overcrowded. The underdosed area is clearly visible in the random mask, and the system is trying to learn the impact with the further training epochs. For example, there are some overdose areas (pink) in epoch 9 and epoch 20 close to the underdose ones. And in epoch 20 in the slice there are many overdose areas close to the boundary shell.

Table 1 shows the same value as written next to the Figure 7.18 for their better comparison of them between the epochs.

*Table 1 Comparison of the percentage of SV and D95 for patient CBP381 in different steps of the training*

| Type | SV | SV, % | Dose, Gy | D95, % |
|---|---|---|---|---|
| Full mask | 39114 | 100 | 2.94877 | 98.29 |
| Random mask | 7761 | 19.84 | 2.93062 | 97.69 |
| epoch 6 step 4 | 4617 | 11.80 | 2.91251 | 97.08 |
| epoch 9 step 2 | 2594 | 6.63 | 2.83213 | 94.40 |
| epoch 12 step 4 | 4118 | 10.53 | 2.90429 | 96.81 |
| epoch 15 step 3 | 3878 | 9.91 | 2.8885 | 96.28 |
| epoch 20 step 5 | 2957 | 7.56 | 2.87274 | 95.76 |

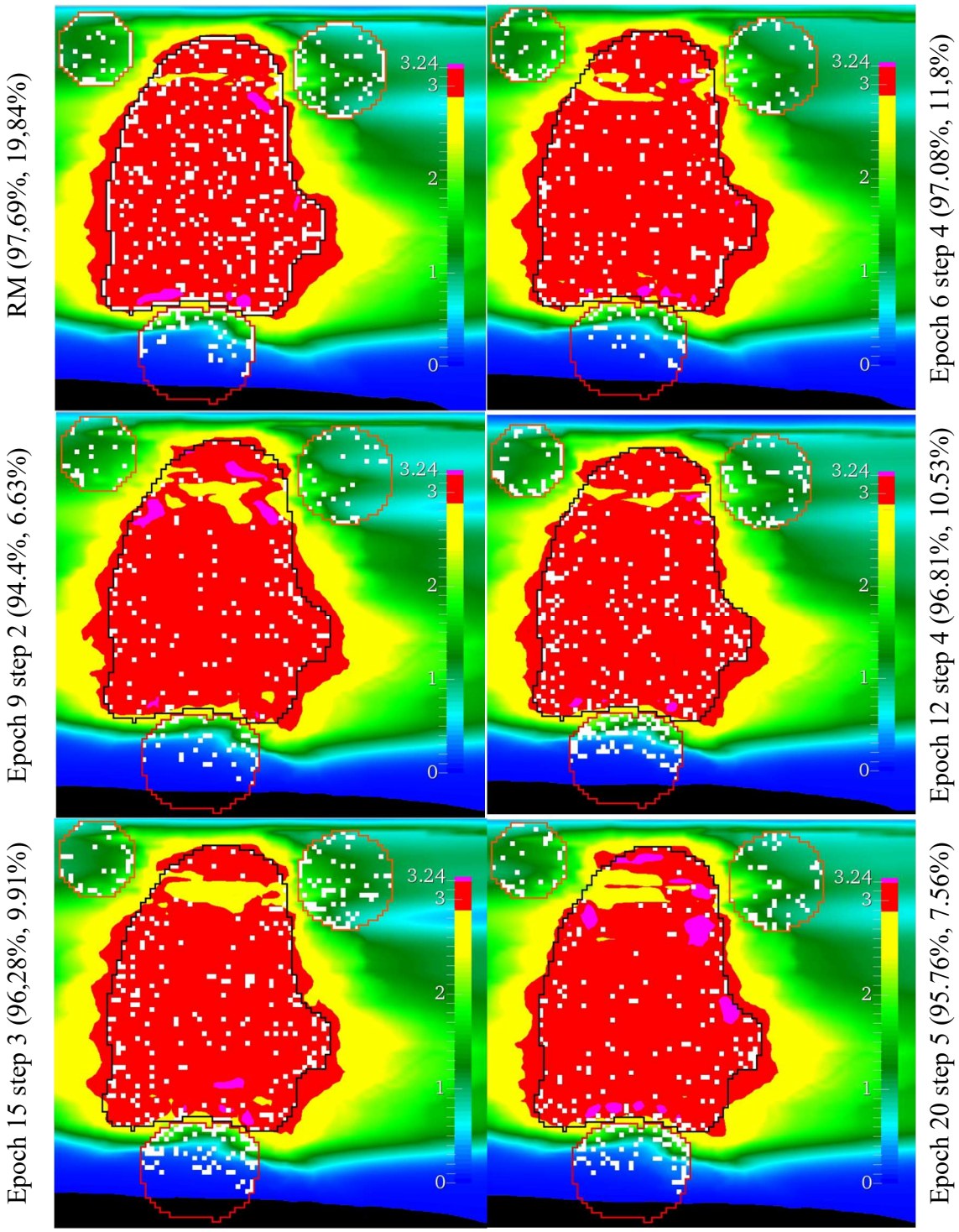



*Figure 7.18 Visualisation of the dose maps in different steps of the training for the patient CBP381 (D95, SV). The colour scale on the left represents the dose and the white spots are SV. Black contour - target, orange - eyes, red - brainstem (Produced with Slicer 8). RM stays for random mask.*

Figure 7.19 shows an example, produced with our CNN model after epoch 6 (AI mask), in comparison with the random mask for three slices (16-18). In slices 16 and 18 we can clearly see the underdose for both methods. Those areas are presented as yellow spots inside the black contoured target. The AI mask in slice 17 reproduces a clear underdose area close to the orange contoured eye and the dark contoured optic nerve.

*Random mask (97,52%, 26%)* *AI mask (96,57%, 15%)*

16

17

18

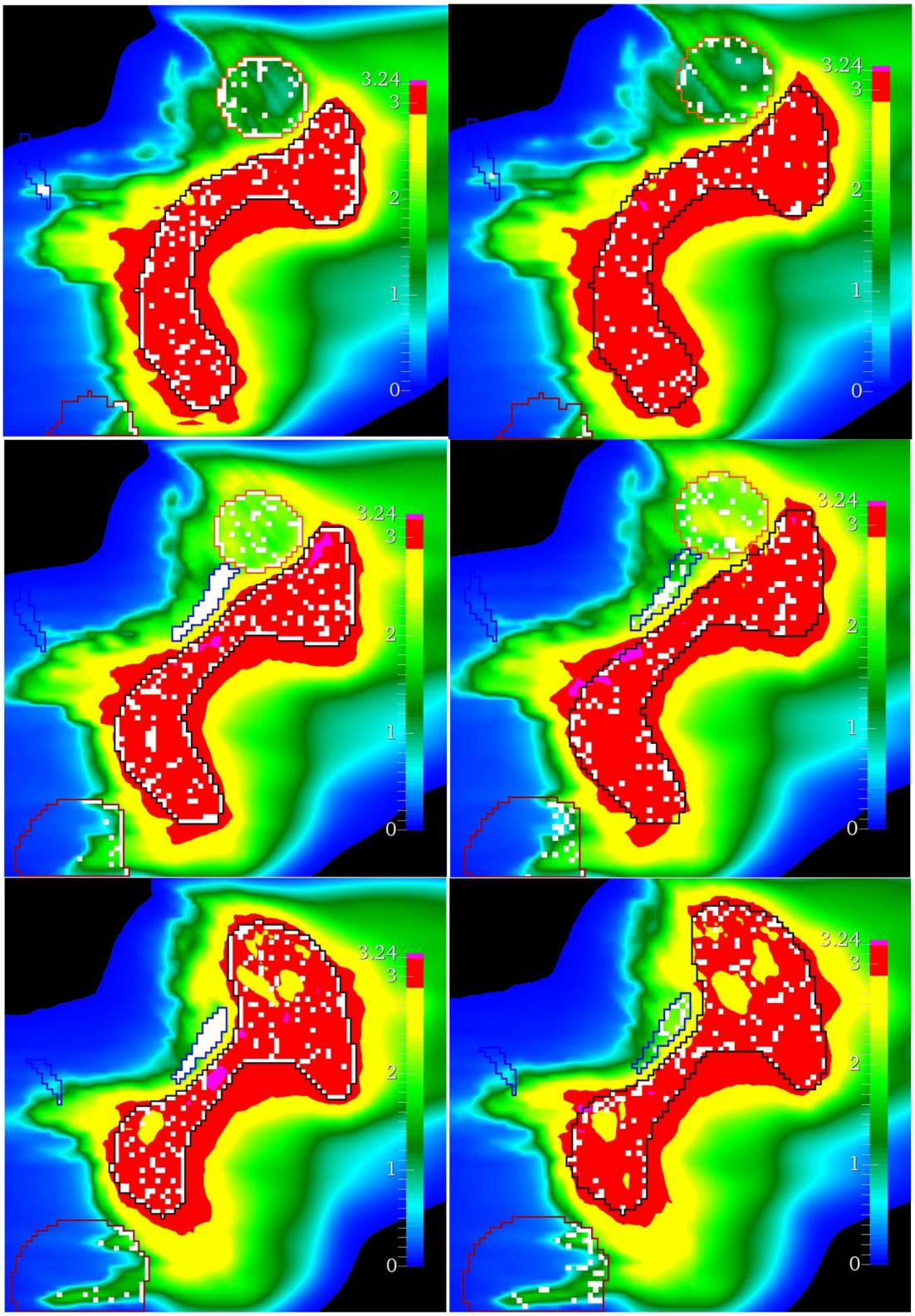



*Figure 7.19 Comparison of the random mask (left) with the mask, produced with our CNN after epoch 6 (right) slice 16 (top), 17 (middle) and 18 (bottom) for CZW434 patient. The colour scale on left represents the dose and the white spots are SV. Black contour-target, dark blue - optic nerve, orange -eyes, red -brainstem (Produced with Slicer 8)*

In the AI mask we see less SV than in the random mask. The target values for both masks are shown in Table 2. The random mask has 1% less than the full mask of the dose coverage after reducing the percentage of SV by a factor of 4, while the AI mask has 2% less after reducing the percentage of SV by appx. factor of 7.

*Table 2 Comparison of values for full, random and AI mask for the CZW434 patient*

| Type of the mask | D95, % | SV, % |
|---|---|---|
| Full | 98.58 | 100 |
| Random | 97,52 | 26 |
| AI | 96,57 | 15 |

# 8. Discussion

In this thesis, a novel framework to select a subsample of voxels in a VOI for treatment plan optimization in heavy ion therapy was presented. Subsampling of the target voxels has a beneficial effect for the computational efficiency and speed of the TPS, which is relevant for efficient workflow in heavy ion therapy facilities. This is especially the case when we consider daily adaptive, and, in particular, robust optimization, where a large number of scenarios coupled with a small-time frame between daily image guidance and treatment necessitates the TPS to be highly efficient. Subsampling the target voxels to be considered during treatment plan optimization has the effect of reduction in both memory consumption and the number of floating-point operations, required to complete the TPS, contributing towards this goal.

The new CNN framework was built on the heuristic approach, previously presented. The original framework considered different sampling parameters for the VOI boundary shell and interior, which had to be set manually and adjusted in case of unacceptable results. Our new framework offers automatic selection of the optimal parameters which deliver the promising results and reduce the computational time.

Optimisation is a complex task. Our system had to find not only the most optimal volumetric distribution of the voxels, which is already challenging because of the number of possible solutions, but also minimize the number of selected voxels keeping the dose coverage constraints in the target. The model showed promising results after only a few epochs of training with a small dataset. For comparison, the system in [7] had around 200 epochs of pretraining and 500 further epochs of training. Besides, the test results demonstrated no overfitting of the model, which is usually expected from small datasets. We could reach acceptable results by reduction of the percentage of SV by a factor of 2.

The simplification of the model allowed us to concentrate only on the target to see the system response to the changes we had made to both the architecture and the loss function. Despite the fact that the model could not get the advantage of the original model, where reinforcement learning was applied, our model is a good starting point to continue the research looking for the relevant input and constraints to improve the results. Considering the simplicity of the model and absence of labels, which are usually used, the received results showed good stability.

To check the robustness of the CNN model more data and training epochs are required. Switching to GPU can speed up the processes. Taking into consideration the limitation of the

time we decided to combine a validation dataset (10 patients) and a test one (10 patients) into a single test dataset. It saved us 75 minutes/epoch and we tuned our system only to the train dataset. That means that the requirement to the test dataset be seen only once and containing only unseen data was fulfilled.

Given the achieved results, the choice of the CNN architecture appears validated. Nevertheless, a number of further improvements and exploration should be investigated, which was not possible in the time frame of this thesis. Firstly, the dose map can give each voxel more information about local dose coverage. That means that the underdosed areas can be incentivized to activate more voxels. We kept the idea to our further research due to the complexity of the problem. It is necessary to understand how to scale the values so that the system can learn from it in connection with the problem of minimization of the percentage of SV. Besides, the dose constraints are different for different OARs (see more in 5.2). Secondly, the reaction of the system to our loss function remains partially unclear. The system shows dynamic swings, where a different loss function might have supported more stable iterations. Single outlier patients appear to have a significant impact on the system. The complexity of the implemented gradient calculations for back propagation, even with the preimplemented function from TensorFlow, is very high and should be understood further. The implemented loss function is not aimed to improve D95 to values higher than 95%, so, the target coverage is typically lower than for the random map. OAR doses are currently not taken into account. The main purpose was to keep D95 above the threshold, so that parts of the loss were turned on and off, which might have caused the fluctuations around 95%. Finally, the system should retain memory of the actual position of the selected voxels, which currently impacts the selection probability only indirectly. The memory of the previous decisions and its results could be achieved through a reinforcement learning system.

It also remains necessary to investigate in more detail the different behaviours (horizontal and vertical) in masks to find the reasons for that and improve the system. The OARs constraints were touched only briefly in the work because the loss function provided improvements related to the random mask only and each case needs to be checked manually.

The following part highlights the difficulties, which we had to face during our work and could help followers to avoid them.

Users, who want to use GPU in Python's TensorFlow must bear in mind that it is exclusively compatible with GPUs by the NVIDIA brand [100]. Some of the AMD GPUs can be used in WLS (Windows Linux system) or with external packages, like ROCm in Linux (tutorial can be

found [101]). But these have an issue with support of new Linux kernels. It is recommended to install them straight after Linux has been setup.

In our case it was not possible to use FIFO for communication if the processes were started on different servers with a shared hard disk drive, so all the processes had to be executed on a single machine. The cause is not entirely clear and should be investigated in future implementations.

Apart from that, in further work if FIFO file is used for the interaction between two processes it is necessary to implement a function, which can check, whether one of them has unexpectedly been closed or does not respond anymore and gives an error. It can be achieved either with the multiprocessing approach or with a simple timer, which also has some drawbacks.

# 9. Outlook

In the future research we are planning to switch the whole system to GPU, that reduces the computational time. This will permit a higher throughput of more epochs on a larger training dataset.

The effect of the following modifications of the input is going to be investigated:

- Downsampling the resolution for a faster calculation.
- Trying the approach with other combinations of the sampling parameters as the starting point. For example, the full mask could offer an opportunity to find a solution independently from the random mask approach.
- Rotating the current version of the input matrix ZYX (40x104x114) to get the new version XYZ (114x104x40). It can improve the activation maps resulted from the convolutional layers, so the layers with 3x1 kernels could potentially get more information.
- Tuning the learning rate, for example, with the learning rate schedule and trying out another optimizer such as Stochastic gradient descent (SGD).
- Trying to use the dose map as a part of the loss function instead of the target planning values.
- Removing the distance map from the model. It is possible that the system can estimate the depth without additional data, or the dependency is not so obvious as it seems (for example, “first layer” of the voxels is less relevant than the second one).
- Creating a separate input channel for each OAR to enable the system to follow the dependencies between different VOIs better.
- Adding some information about the beam parameters. There is a possibility that the voxel sampling depends on the spacing and lateral width of the pencil beams. As such, adding this information could make the system more generalized and accurate.
- It is also not clear whether the chosen architecture can see the distribution of the voxels in the entire volume. We suspect the dilation could work better if the system had a starting point, like a random mask to be refined. Then, the positioning of the voxels can be tuned to be more optimal. [7], [52] used the previous result of the system as a part of input.
- Changing the point where the target and OARs are summarized in the system (in our work, in the first layer) allows the system to explore the data first separately and then in

combination. An example of the approach can be seen in Figure 9.1 done by [8] for an automatic TPS for IMRT.

- Updating the system to A3C architecture.

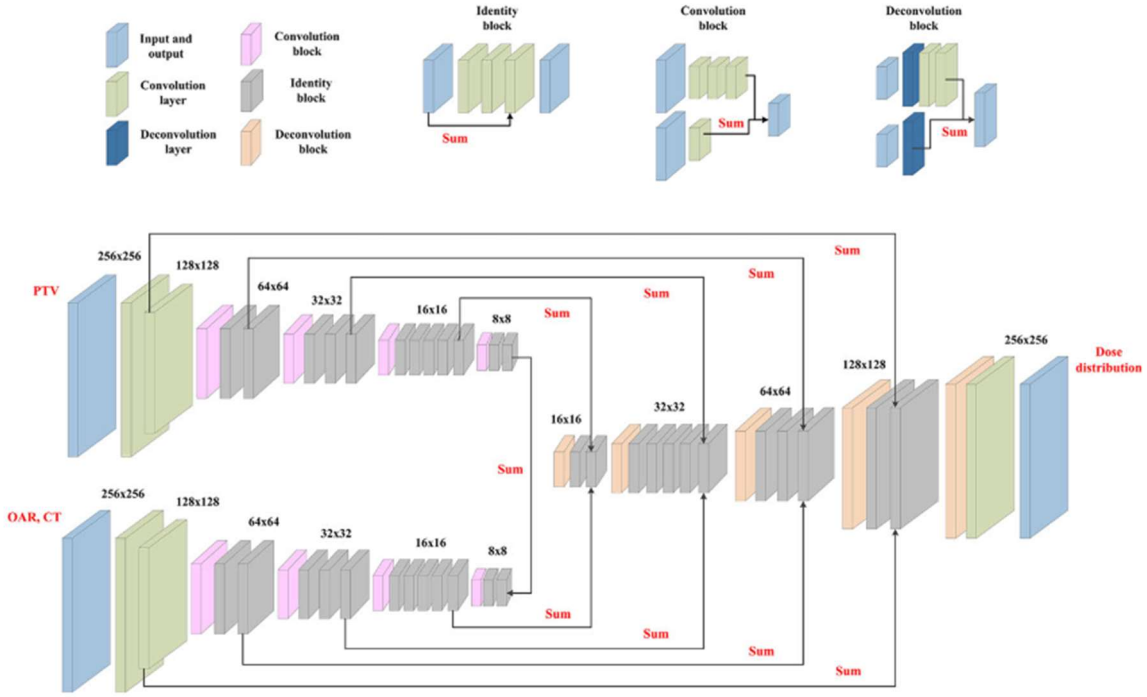



*Figure 9.1 Illustration of the architecture used in [8], the numbers on top of each block are dimensions.*

For deeper understanding of the weaknesses of the algorithm it is necessary to find a mathematical algorithm, which can be applied to the whole dataset and convert the distribution of the SV into volumetric clusters, where the subvolume either overcrowded or with lack of SV will be identified. This can be done by analysis of the underdosed areas of the dose map in combination with the distance map. The method should help to analyse the results better and give a hint to a further improvement. For example, if it shows that each sample has a low dose value mostly close to OARs, then the loss function can be updated to penalize those areas more than the others.

The promising results of this study guarantee an extension of the model training to a larger portion of the pilot project dataset, of which we currently used only 50 of approximately 300 possible Head and Neck cases. Moreover, different tumour locations, also outside the head, will be addressed to prove the generalisability of the approach. This will also require incorporating target motion into the treatment planning process. Finally, we will include robust optimization scenarios, where we expect to find synergies across similar scenarios that a CNN could identify with methods derived from this work.

# 10. Appendices

## 10.1 Python environment

Functional setup in Python 3.8.10:

| Package | Version |
|---|---|
| numpy | 1.22.2 |
| pandas | 1.4.1 |
| tensorflow | 2.8.0 |
| tensorflow_probabilities | 0.16.0 |

## 10.2 Additional material

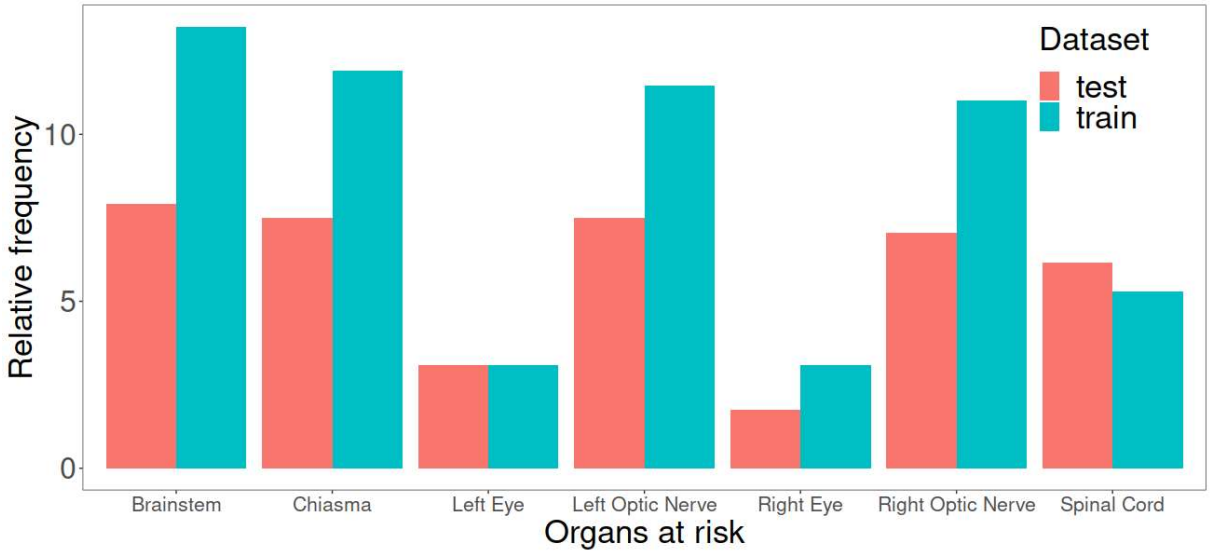



*Figure 10.1 Relative frequency of OARs in the dataset*

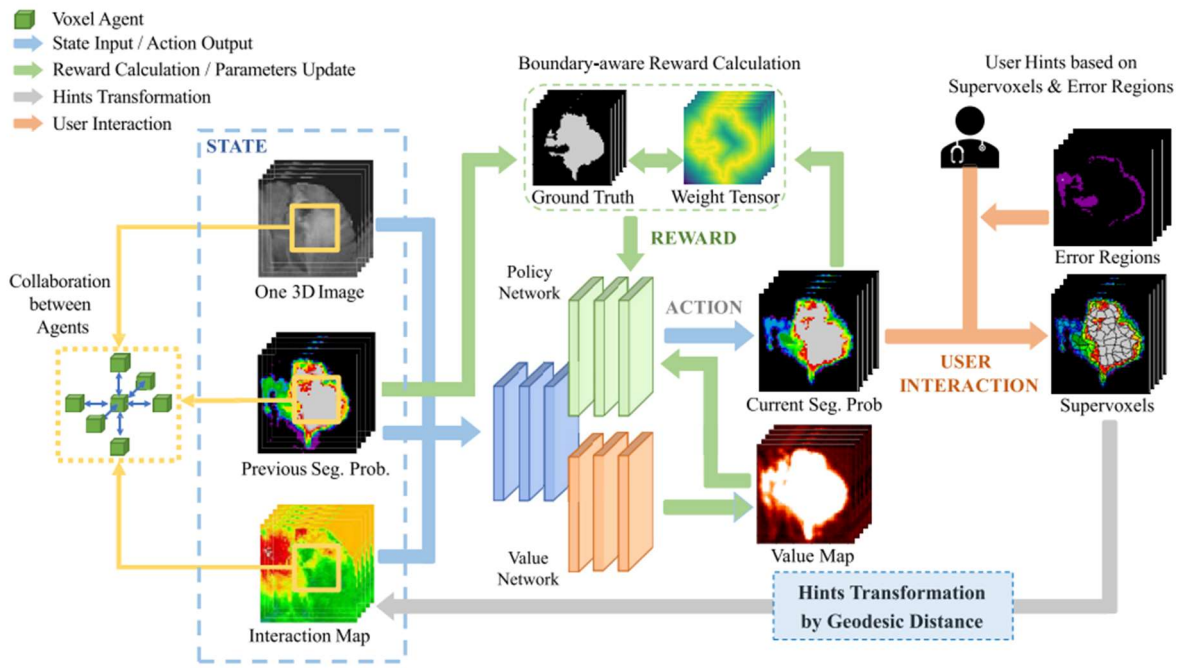



*Figure 10.2 IteR-MRL architecture. At each step it gets the probability of the previous step and interaction map with regions which are either should be selected or removed from the output mask.*

```
Iter #  1/100 time:    0.349 residual:   38.75 % (old:   34.82 %), dChi2Change:      1, dChi2: 6096.92, dV
Iter #  2/100 time:   0.2373 residual:   19.31 % (old:   17.35 %), dChi2Change: 0.751703, dChi2: 1513.85,
Iter #  3/100 time:   0.2486 residual:   8.568 % (old:   7.698 %), dChi2Change: 0.80312, dChi2: 298.047, d
Iter #  4/100 time:   0.2501 residual:   5.524 % (old:   4.964 %), dChi2Change: 0.584209, dChi2: 123.925,
Iter #  5/100 time:   0.2399 residual:    4.56 % (old:   4.097 %), dChi2Change: 0.318683, dChi2: 84.4323,
Iter #  6/100 time:   0.2852 residual:   4.092 % (old:   3.676 %), dChi2Change: 0.194808, dChi2: 67.9842,
Iter #  7/100 time:   0.2435 residual:   3.717 % (old:    3.34 %), dChi2Change: 0.174786, dChi2: 56.1015,
Iter #  8/100 time:   0.2394 residual:   3.423 % (old:   3.075 %), dChi2Change: 0.152153, dChi2: 47.5655,
Iter #  9/100 time:   0.2434 residual:   3.208 % (old:   2.882 %), dChi2Change: 0.121659, dChi2: 41.7788,
Iter # 10/100 time:   0.2345 residual:   3.023 % (old:   2.716 %), dChi2Change: 0.11167, dChi2: 37.1133, d
Iter # 11/100 time:   0.2733 residual:   2.847 % (old:   2.558 %), dChi2Change: 0.113187, dChi2: 32.9126,
```

*Figure 10.3 Example of TRiP98 log file*

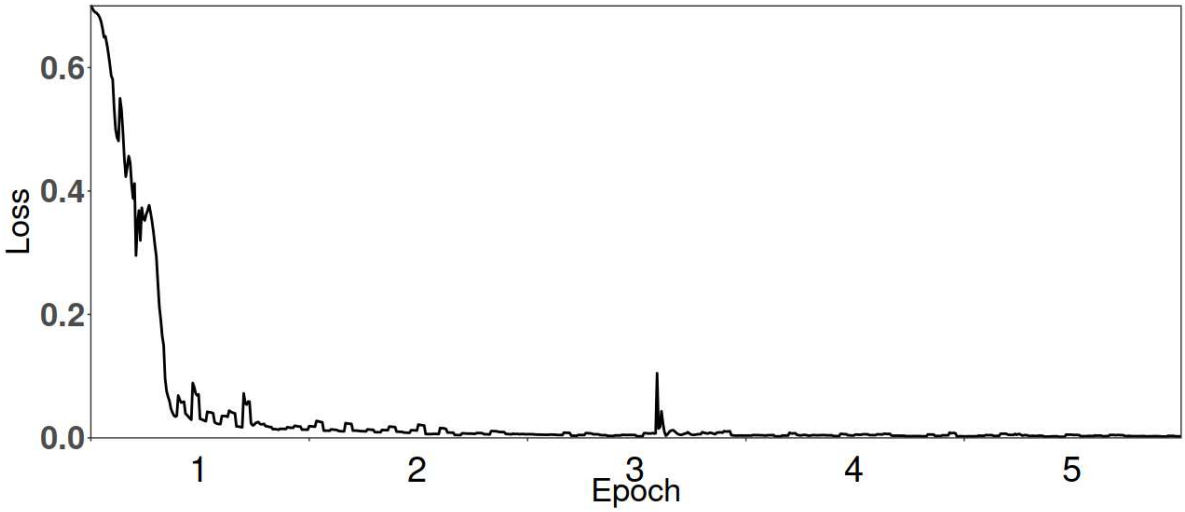



*Figure 10.4 Loss in pretraining stage with full mask as label*

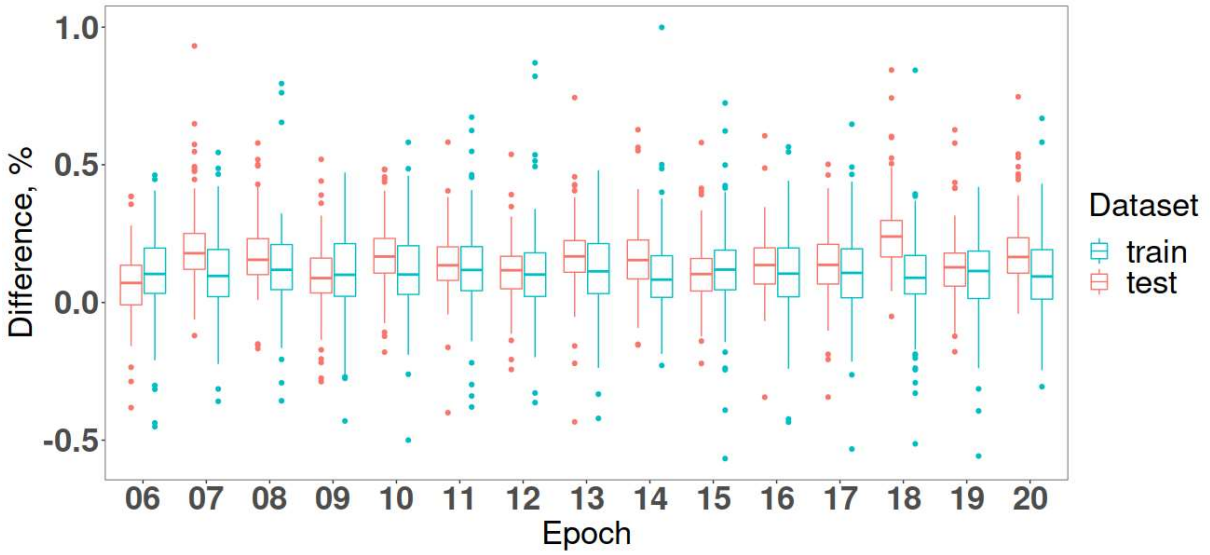



*Figure 10.5 D95 difference between random mask and the mask produced with AI for each patient.*

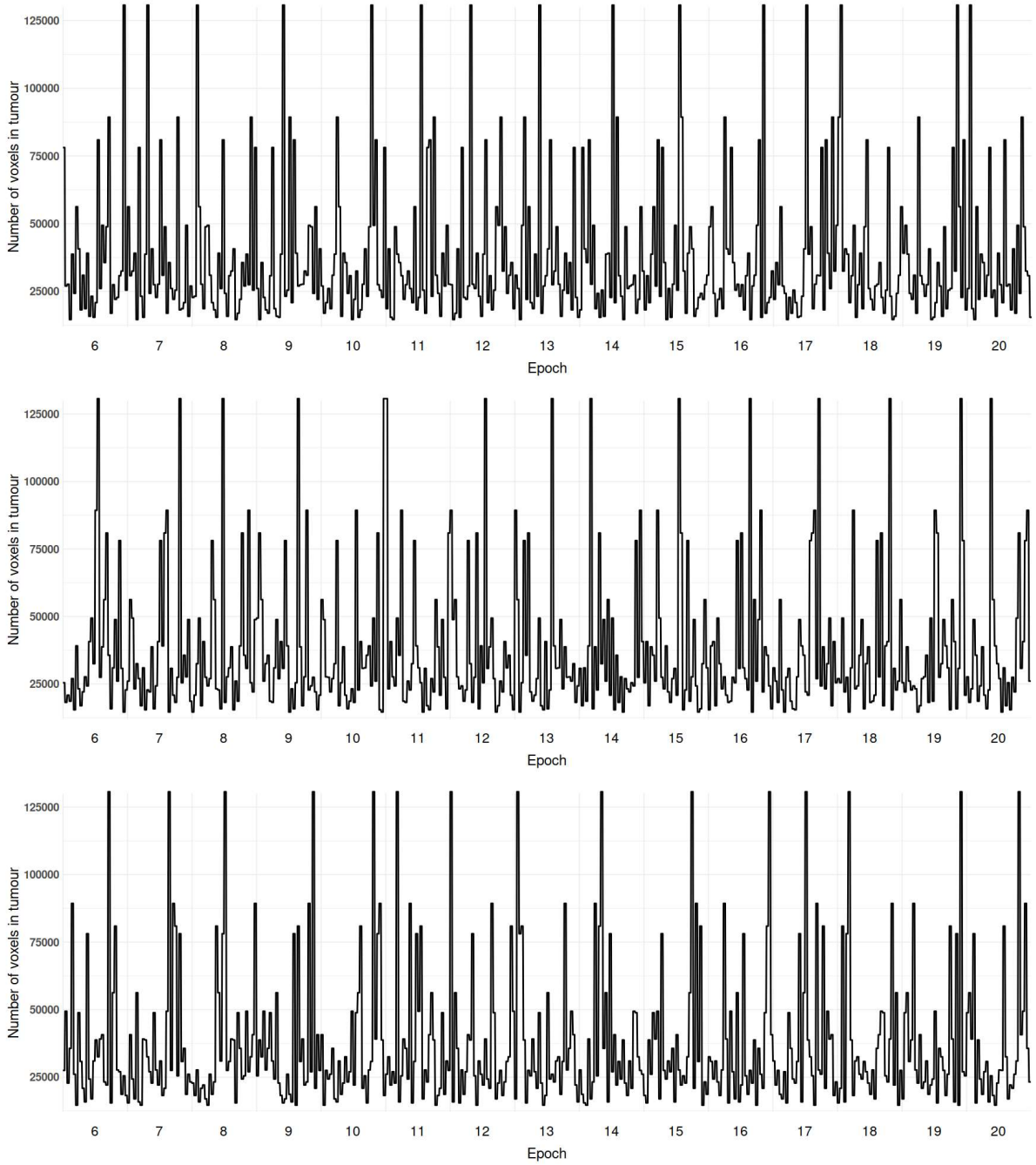



*Figure 10.6 History of changing number of voxels in the tumour during the training for experiment 1-3 (top-bottom)*

# 11. Bibliography


[1] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017, doi: 10.1038/nature21056.

[2] D. Capper *et al.*, "DNA methylation-based classification of central nervous system tumours," *Nature*, vol. 555, no. 7697, pp. 469–474, 2018, doi: 10.1038/nature26000.

[3] S. M. McKinney *et al.*, "International evaluation of an AI system for breast cancer screening," *Nature*, vol. 577, no. 7788, 2020, doi: 10.1038/s41586-019-1799-6.

[4] D. Ardila *et al.*, "End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography," *Nat. Med.*, vol. 25, no. 6, pp. 954–961, 2019, doi: 10.1038/s41591-019-0447-x.

[5] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, doi: 10.1007/978-3-319-24574-4_28.

[6] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, 2021, doi: 10.1109/ACCESS.2021.3086020.

[7] C. Ma *et al.*, "Boundary-Aware Supervoxel-Level Iteratively Refined Interactive 3D Image Segmentation with Multi-Agent Reinforcement Learning," *IEEE Trans. Med. Imaging*, vol. 40, no. 10, 2021, doi: 10.1109/TMI.2020.3048477.

[8] J. Fan, J. Wang, Z. Chen, C. Hu, Z. Zhang, and W. Hu, "Automatic treatment planning based on three-dimensional dose distribution predicted from deep learning technique," *Med. Phys.*, vol. 46, no. 1, 2019, doi: 10.1002/mp.13271.

[9] A. Recht *et al.*, "The Sequencing of Chemotherapy and Radiation Therapy after Conservative Surgery for Early-Stage Breast Cancer," *N. Engl. J. Med.*, vol. 334, no. 21, 1996, doi: 10.1056/nejm199605233342102.

[10] E. Huynh *et al.*, "Artificial intelligence in radiation oncology," *Nat. Rev. Clin. Oncol.*, vol. 17, no. 12, pp. 771–781, 2020, doi: 10.1038/s41571-020-0417-8.

[11] M. Krämer and M. Scholz, "Treatment planning for heavy-ion radiotherapy: calculation and optimization of biologically effective dose," *Phys. Med. Biol.*, vol. 45, no. 11, pp. 3319–3330, Nov. 2000, doi: 10.1088/0031-9155/45/11/314.

[12] M. Krämer, O. Jäkel, T. Haberer, G. Kraft, D. Schardt, and U. Weber, "Treatment planning for heavy-ion radiotherapy: physical beam model and dose optimization," *Phys. Med. Biol.*, vol. 45, no. 11, pp. 3299–3317, Nov. 2000, doi: 10.1088/0031-

9155/45/11/313.

[13] M. Wolf, K. Anderle, M. Prall, and C. Graeff, “Random voxel subsampling enables large scale robust optimization in heavy ion therapy,” in *50. Jahrestagung der Deutschen Gesellschaft für Medizinische Physik (DGMP) e. V.*, 2019, pp. 252–253, [Online]. Available: https://cdn.dgmp.de/media/document/3147/DGMP2019-Abstractband.pdf.

[14] R. R. Wilson, “Radiological use of fast protons,” *Radiology*, vol. 47, pp. 487–491, 1946.

[15] H. Bichsel, “Stochastics of Energy Loss and Biological Effects of Heavy Ions in Radiation Therapy,” in *Advances in Quantum Chemistry*, 2013, pp. 1–38.

[16] M. Durante and H. Paganetti, “Nuclear physics in particle therapy : a review,” *Reports Prog. Phys.*, vol. 79, p. 096702, 2016, doi: 10.1088/0034-4885/79/9/096702.

[17] A. Embriaco, E. V Bellinzona, A. Fontana, and A. Rotondi, “On Molière and Fermi–Eyges scattering theories in hadrontherapy,” *Phys. Med. Biol.*, vol. 62, no. 15, pp. 6290–6303, Jul. 2017, doi: 10.1088/1361-6560/aa7a94.

[18] D. Schardt, T. Elsässer, and D. Schulz-Ertner, “Heavy-ion tumor therapy: Physical and radiobiological benefits,” *Rev. Mod. Phys.*, vol. 82, no. 1, pp. 383–425, 2010, doi: 10.1103/RevModPhys.82.383.

[19] M. Rovituso and C. La Tessa, “Nuclear interactions of new ions in cancer therapy: impact on dosimetry,” *Transl. Cancer Res.*, vol. 6, no. S5, pp. S914–S933, 2017, doi: 10.21037/tcr.2017.06.46.

[20] C. Bert, S. O. Grözinger, and E. Rietzel, “Quantification of interplay effects of scanned particle beams and moving targets,” *Phys. Med. Biol.*, vol. 53, no. 9, 2008, doi: 10.1088/0031-9155/53/9/003.

[21] T. Kubiak, “Particle therapy of moving targets—the strategies for tumour motion monitoring and moving targets irradiation,” *Br. J. Radiol.*, vol. 89, no. 1066, p. 20150275, Oct. 2016, doi: 10.1259/bjr.20150275.

[22] C. Bert and M. Durante, “Motion in radiotherapy: particle therapy,” *Phys. Med. Biol.*, vol. 56, no. 16, pp. R113–R144, Aug. 2011, doi: 10.1088/0031-9155/56/16/R01.

[23] M. Riboldi, P. R. Orecchia, and P. G. Baroni, “Real-time tumour tracking in particle therapy : technological developments and future perspectives,” *Lancet Oncol.*, vol. 13, no. 9, pp. e383–e391, 2012, doi: 10.1016/S1470-2045(12)70243-7.

[24] K. Czerska *et al.*, “Clinical practice vs. state-of-the-art research and future visions: Report on the 4D treatment planning workshop for particle therapy – Edition 2018 and 2019,” *Phys. Medica*, vol. 82, pp. 54–63, Feb. 2021, doi: 10.1016/j.ejmp.2020.12.013.

[25] C. Graeff, R. Lüchtenborg, J. G. Eley, M. Durante, and C. Bert, “A 4D-optimization concept for scanned ion beam therapy,” *Radiother. Oncol.*, vol. 109, no. 3, pp. 419–424,

2013.

[26] J. F. Fowler, “21 Years of biologically effective dose,” *Br. J. Radiol.*, vol. 83, no. 991, pp. 554–568, Jul. 2010, doi: 10.1259/bjr/31372149.

[27] T. Friedrich, U. Scholz, T. Elsässer, M. Durante, and M. Scholz, “Systematic analysis of RBE and related quantities using a database of cell survival experiments with ion beam irradiation,” *J. Radiat. Res.*, vol. 54, no. 3, pp. 494–514, 2013, doi: 10.1093/jrr/rrs114.

[28] J. Z. Wang, Z. Huang, S. S. Lo, W. T. C. Yuh, and N. a Mayr, “A generalized linear-quadratic model for radiosurgery, stereotactic body radiation therapy, and high-dose rate brachytherapy.,” *Sci. Transl. Med.*, vol. 2, no. 39, pp. 39ra48-39ra48, 2010, doi: 10.1126/scitranslmed.3000864.

[29] ICRU, “Report 93 - Prescribing, Recording and Reporting Light Ion Beam Therapy,” *J. ICRU*, vol. 16, no. 1–2, 2016, doi: 10.1093/jicru/ndy025.

[30] T. Inaniwa *et al.*, “Treatment planning for a scanned carbon beam with a modified microdosimetric kinetic model.,” *Phys. Med. Biol.*, vol. 55, no. 22, pp. 6721–6737, Nov. 2010, doi: 10.1088/0031-9155/55/22/008.

[31] R. Grün *et al.*, “Impact of enhancements in the local effect model (LEM) on the predicted RBE-weighted target dose distribution in carbon ion therapy,” *Phys. Med. Biol.*, vol. 57, no. 22, pp. 7261–7274, 2012, doi: 10.1088/0031-9155/57/22/7261.

[32] T. Friedrich, U. Scholz, M. Durante, and M. Scholz, “RBE of ion beams in hypofractionated radiotherapy (SBRT),” *Phys. Medica*, vol. 30, no. 5, pp. 588–591, 2014.

[33] Y. Yoshida *et al.*, “Evaluation of therapeutic gain for fractionated carbon-ion radiotherapy using the tumor growth delay and crypt survival assays,” *Radiother. Oncol.*, vol. 117, no. 2, pp. 351–357, Nov. 2015, doi: 10.1016/j.radonc.2015.09.027.

[34] J. D. Chapman, “Can the two mechanisms of tumor cell killing by radiation be exploited for therapeutic gain?,” *J. Radiat. Res.*, vol. 55, no. 1, pp. 2–9, 2014, doi: 10.1093/jrr/rrt111.

[35] A. M. Laine *et al.*, “The Role of Hypofractionated Radiation Therapy with Photons, Protons, and Heavy Ions for Treating Extracranial Lesions,” *Front. Oncol.*, vol. 5, Jan. 2016, doi: 10.3389/fonc.2015.00302.

[36] W. K. Weyrather, S. Ritter, M. Scholz, and G. Kraft, “RBE for carbon track-segment irradiation in cell lines of differing repair capacity,” *Int. J. Radiat. Biol.*, vol. 75, no. 11, 1999, doi: 10.1080/095530099139232.

[37] T. Pavel, “Feasibility of magnetic resonance imaging-based radiation therapy for brain tumour treatment,” School of Science, 2017.

[38] ICRU, “ICRU report 50: Prescribing, recording and reporting photon beam therapy (ICRU report 50),” *J. ICRU*, 1993.

[39] M. Krämer and M. Scholz, “Rapid calculation of biological effects in ion radiotherapy,” *Phys. Med. Biol.*, vol. 51, no. 8, pp. 1959–1970, 2006, doi: 10.1088/0031-9155/51/8/001.

[40] T. Elsässer *et al.*, “Quantification of the relative biological effectiveness for ion beam radiotherapy: Direct experimental comparison of proton and carbon ion beams and a novel approach for treatment planning,” *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 78, no. 4, pp. 1177–1183, 2010.

[41] T. Friedrich, R. Grün, U. Scholz, T. Elsässer, M. Durante, and M. Scholz, “Sensitivity analysis of the relative biological effectiveness predicted by the local effect model.,” *Phys. Med. Biol.*, vol. 58, no. 19, pp. 6827–49, 2013, doi: 10.1088/0031-9155/58/19/6827.

[42] G. Kraft, M. Krämer, and M. Scholz, “LET, track structure and models - A review,” *Radiat. Environ. Biophys.*, vol. 31, no. 3, pp. 161–180, 1992, doi: 10.1007/BF01214825.

[43] A. N. Ramesh, C. Kambhampati, J. R. T. Monson, and P. J. Drew, “Artificial intelligence in medicine,” *Ann. R. Coll. Surg. Engl.*, vol. 86, no. 5, pp. 334–338, 2004, doi: 10.1308/147870804290.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.

[45] A. Hundt *et al.*, “‘Good Robot!’: Efficient Reinforcement Learning for Multi-Step Visual Tasks with Sim to Real Transfer,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, 2020, doi: 10.1109/LRA.2020.3015448.

[46] M. A. Deif and R. E. Hammam, “Skin Lesions Classification Based on Deep Learning Approach,” *J. Clin. Eng.*, vol. 45, no. 3, pp. 155–161, 2020, doi: 10.1097/jce.0000000000000405.

[47] K. Seetharam, S. Raina, and P. P. Sengupta, “The Role of Artificial Intelligence in Echocardiography,” *Curr. Cardiol. Rep.*, vol. 22, no. 9, 2020, doi: 10.1007/s11886-020-01329-7.

[48] P. M. Ellis and R. Vandermeer, “Delays in the diagnosis of lung cancer,” *J. Thorac. Dis.*, vol. 3, no. 3, 2011, doi: 10.3978/j.issn.2072-1439.2011.01.01.

[49] A. Kleppe, O. J. Skrede, S. De Raedt, K. Liestøl, D. J. Kerr, and H. E. Danielsen, “Designing deep learning studies in cancer diagnostics,” *Nat. Rev. Cancer*, vol. 21, no. 3, pp. 199–211, 2021, doi: 10.1038/s41568-020-00327-9.

[50] P. Schelb *et al.*, “Classification of cancer at prostate MRI: Deep Learning versus Clinical

PI-RADS Assessment," *Radiology*, vol. 293, no. 3, 2019, doi: 10.1148/radiol.2019190938.

[51] F. I. Tushar *et al.*, "Classification of Multiple Diseases on Body CT Scans Using Weakly Supervised Deep Learning," *Radiol. Artif. Intell.*, vol. 4, no. 1, 2022, doi: 10.1148/ryai.210026.

[52] X. Liao *et al.*, "Iteratively-refined interactive 3D medical image segmentation with multi-agent reinforcement learning," 2020, doi: 10.1109/CVPR42600.2020.00941.

[53] D. Nguyen *et al.*, "3D radiotherapy dose prediction on head and neck cancer patients with a hierarchically densely connected U-net deep learning architecture," *Phys. Med. Biol.*, vol. 64, no. 6, 2019, doi: 10.1088/1361-6560/ab039b.

[54] E. Yom-Tov, G. Feraru, M. Kozdoba, S. Mannor, M. Tennenholtz, and I. Hochberg, "Encouraging physical activity in patients with diabetes: Intervention using a reinforcement learning system," *J. Med. Internet Res.*, vol. 19, no. 10, 2017, doi: 10.2196/JMIR.7994.

[55] C. Chen *et al.*, "OoDAnalyzer: Interactive analysis of out-of-distribution samples," *arXiv*. 2020, doi: 10.1109/tvcg.2020.2973258.

[56] M. Koziarski, "Two-Stage Resampling for Convolutional Neural Network Training in the Imbalanced Colorectal Cancer Image Classification," *arXiv*, Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.03332.

[57] A. E. Runnova, V. Musatov, A. Andreev, and M. Zhuravlev, "Brain states recognition during visual perception by means of artificial neural network in the different EEG frequency ranges," 2018, doi: 10.1117/12.2314829.

[58] J. Shi *et al.*, "Improvement of damage segmentation based on pixel-level data balance using vgg-unet," *Appl. Sci.*, vol. 11, no. 2, pp. 1–17, 2021, doi: 10.3390/app11020518.

[59] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," 2017.

[60] J. Jordan, "Convolutional neural networks.," Jun. 26, 2017. https://www.jeremyjordan.me/convolutional-neural-networks/ (accessed Apr. 04, 2022).

[61] "OpenfMRI." https://openfmri.org/ (accessed Mar. 28, 2022).

[62] "CT Medical Images | Kaggle." https://www.kaggle.com/datasets/kmader/siim-medical-images (accessed Mar. 28, 2022).

[63] J. Thariat, J. M. Hannoun-Levi, A. Sun Myint, T. Vuong, and J. P. Gérard, "Past, present, and future of radiotherapy for the benefit of patients," *Nat. Rev. Clin. Oncol.*, vol. 10, no. 1, pp. 52–60, 2013, doi: 10.1038/nrclinonc.2012.203.

[64] K. Tingelhoff *et al.*, "Analysis of manual segmentation in paranasal CT images," *Eur.*

*Arch. Oto-Rhino-Laryngology*, vol. 265, no. 9, 2008, doi: 10.1007/s00405-008-0594-z.

[65] E. Alcaín *et al.*, "Hardware architectures for real-time medical imaging," *Electron.*, vol. 10, no. 24, 2021, doi: 10.3390/electronics10243118.

[66] M. Ma, N. Kovalchuk, M. K. Buyyounouski, L. Xing, and Y. Yang, "Incorporating dosimetric features into the prediction of 3D VMAT dose distributions using deep convolutional neural network," *Phys. Med. Biol.*, vol. 64, no. 12, 2019, doi: 10.1088/1361-6560/ab2146.

[67] Y. Nomura, J. Wang, H. Shirato, S. Shimizu, and L. Xing, "Fast spot-scanning proton dose calculation method with uncertainty quantification using a three-dimensional convolutional neural network," *Phys. Med. Biol.*, vol. 65, no. 21, 2020, doi: 10.1088/1361-6560/aba164.

[68] A. Neishabouri, N. Wahl, A. Mairani, U. Köthe, and M. Bangert, "Long short-term memory networks for proton dose calculation in highly heterogeneous tissues," *Med. Phys.*, vol. 48, no. 4, 2021, doi: 10.1002/mp.14658.

[69] X. Zhang, Z. Hu, G. Zhang, Y. Zhuang, Y. Wang, and H. Peng, "Dose calculation in proton therapy using a discovery cross-domain generative adversarial network (DiscoGAN)," *Med. Phys.*, vol. 48, no. 5, 2021, doi: 10.1002/mp.14781.

[70] C. Wu *et al.*, "Improving proton dose calculation accuracy by using deep learning," *Mach. Learn. Sci. Technol.*, vol. 2, no. 1, 2021, doi: 10.1088/2632-2153/abb6d5.

[71] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January, doi: 10.1109/CVPR.2017.243.

[72] H. Paganetti, "Range uncertainties in proton therapy and the role of Monte Carlo simulations," *Physics in Medicine and Biology*, vol. 57, no. 11. 2012, doi: 10.1088/0031-9155/57/11/R99.

[73] C. R. Maurer, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 2, pp. 265–270, 2003, doi: 10.1109/TPAMI.2003.1177156.

[74] M. Horcicka, "Master-Thesis Theorie und Numerik zur nichtlinearen," 2011.

[75] M. Uhl *et al.*, "Highly effective treatment of skull base chordoma with carbon ion irradiation using a raster scan technique in 155 patients: First long-term results," *Cancer*, vol. 120, no. 21, pp. 3410–3417, 2014, doi: 10.1002/cncr.28877.

[76] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective

continuous space problems: A review," *Evolutionary Computation*, vol. 25, no. 1. 2017, doi: 10.1162/EVCO_r_00180.

[77] D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon, "Optimization by simulated annealing. An experimental evaluation. Part I. Graph partitioning," *Oper. Res.*, vol. 37, no. 6, 1989, doi: 10.1287/opre.37.6.865.

[78] F. Milletari, N. Navab, and S. A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," 2016, doi: 10.1109/3DV.2016.79.

[79] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *4th Int. Conf. Learn. Represent. ICLR 2016 - Conf. Track Proc.*, 2016.

[80] G. Wang *et al.*, "DeepIGeoS: A Deep Interactive Geodesic Framework for Medical Image Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, 2019, doi: 10.1109/TPAMI.2018.2840695.

[81] R. Furuta, N. Inoue, and T. Yamasaki, "PixelRL: Fully Convolutional Network with Reinforcement Learning for Image Processing," *IEEE Trans. Multimed.*, vol. 22, no. 7, 2020, doi: 10.1109/TMM.2019.2960636.

[82] K. H. Zou *et al.*, "Statistical Validation of Image Segmentation Quality Based on a Spatial Overlap Index," *Acad. Radiol.*, vol. 11, no. 2, 2004, doi: 10.1016/S1076-6332(03)00671-8.

[83] M. Grześ and D. Kudenko, "Theoretical and empirical analysis of reward shaping in reinforcement learning," *8th Int. Conf. Mach. Learn. Appl. ICMLA 2009*, no. December, pp. 337–344, 2009, doi: 10.1109/ICMLA.2009.33.

[84] C. Olah, A. Mordvintsev, and L. Schubert, "Feature Visualization," *Distill*, vol. 2, no. 11, 2017, doi: 10.23915/distill.00007.

[85] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, vol. 2016-December, doi: 10.1109/CVPR.2016.90.

[86] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2016.

[87] R. Hamaguchi, A. Fujita, K. Nemoto, T. Imaizumi, and S. Hikosaka, "Effective Use of Dilated Convolutions for Segmenting Small Object Instances in Remote Sensing Imagery," in *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, 2018, vol. 2018-January, doi: 10.1109/WACV.2018.00162.

[88] Q. Zhu, P. Zhang, Z. Wang, and X. Ye, "A New Loss Function for CNN Classifier Based on Predefined Evenly-Distributed Class Centroids," *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2019.2960065.

[89] M. Lambrecht *et al.*, "Radiation dose constraints for organs at risk in neuro-oncology;

the European Particle Therapy Network consensus," *Radiother. Oncol.*, vol. 128, no. 1, pp. 26–36, 2018, doi: 10.1016/j.radonc.2018.05.001.

[90] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," 2016, Accessed: Mar. 10, 2022. [Online]. Available: https://www.tensorflow.org/.

[91] F. Chollet, "Keras: The Python Deep Learning library," *Keras.Io*, 2015.

[92] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, vol. 32.

[93] "GitHub - libfuse/sshfs: A network filesystem client to connect to SSH servers." https://github.com/libfuse/sshfs (accessed Apr. 05, 2022).

[94] "subprocess — Subprocess management — Python 3.10.4 documentation." https://docs.python.org/3/library/subprocess.html (accessed Apr. 10, 2022).

[95] "os — Miscellaneous operating system interfaces — Python 3.10.4 documentation." https://docs.python.org/3/library/os.html?highlight=os#module-os (accessed Apr. 10, 2022).

[96] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, no. 7825. 2020, doi: 10.1038/s41586-020-2649-2.

[97] S. Van Der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Comput. Sci. Eng.*, vol. 13, no. 2, 2011, doi: 10.1109/MCSE.2011.37.

[98] W. McKinney and P. D. Team, "Pandas - Powerful Python Data Analysis Toolkit," *Pandas - Powerful Python Data Anal. Toolkit*, 2015.

[99] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 2015.

[100] "GPU support | TensorFlow." https://www.tensorflow.org/install/gpu (accessed Apr. 10, 2022).

[101] D. Ngo, "Install Tensorflow 2 & PyTorch for AMD GPUs | by Dat Ngo | Analytics Vidhya | Medium," Dec. 06, 2020. https://medium.com/analytics-vidhya/install-tensorflow-2-for-amd-gpus-87e8d7aeb812 (accessed Apr. 10, 2022).