UNIVERSITÀ DEGLI STUDI DI PALERMO
SCUOLA DI DOTTORATO
Technologies and Sciences for Human Health
Department of Biological, Chemical and Pharmaceutical Sciences and Technologies
SSD PHYS-06/A

Artificial Intelligence methods for Physics applied to Medicine

Director:
Prof. VALERIA VETRI

Tutor:
Prof. MAURIZIO MARRALE

PhD Student:
MATTIA ROMEO

XXXVIII CYCLE

This thesis was produced while attending the PhD programme in Technologies and Sciences for Human Health at the University of Palermo, Cycle XXXVIII, with the support of a scholarship co-financed by the Ministerial Decree no. 352 of 9th April 2022, based on the NRRP - funded by the European Union - NextGenerationEU - Mission 4 "Education and Research", Component 2 "From Research to Business", Investment 3.3, and by the company UPMC Italy s.r.l. - ISMETT

We may hope that machines will eventually compete with men in all purely intellectual fields. But which are the best ones to start with? Even this is a difficult decision. Many people think that a very abstract activity, like the playing of chess, would be best. It can also be maintained that it is best to provide the machine with the best sense organs that money can buy, and then teach it to understand and speak English. This process could follow the normal teaching of a child. Things would be pointed out and named, etc. Again I do not know what the right answer is, but I think both approaches should be tried.

We can only see a short distance ahead, but we can see plenty there that needs to be done.

Alan Turing, Computing machinery and intelligence, 1950

These words were written in 1950, and although many things have been accomplished, much still remains to be done.

CONTENTS

Abstract **9**

Introduction **11**

1 Artificial Intelligence and Machine Learning **17**
1.1 General overview on Artificial Intelligence 18
1.1.1 Historical results achieved by Artificial Intelligence 19
1.2 General overview on Machine Learning 21
1.2.1 Machine Learning tasks 23
1.2.2 Machine Learning Experience 24
1.2.3 Machine Learning Performance 25
1.2.4 Loss Functions and Optimization 28
1.3 Classical Machine Learning Models 30
1.3.1 Linear Models 31
1.3.2 Maximum Likelihood Estimation 34
1.3.3 Logistic Regression 35
1.4 Models regularization 36
1.4.1 Model capability 36
1.4.2 Hyper-parameters and Validation set 37
1.4.3 Regularization techniques 39

2 Deep Learning **43**
2.1 Fundamentals of Deep Learning 44
2.1.1 Perceptrons and Multi-layer Perceptrons network 45
2.1.2 Backpropagation algorithm 50
2.1.3 Universal Approximation Theorem 55
2.2 Computer vision 57
2.2.1 Convolution 59
2.2.2 Convolutional Neural Network 62

2.3 Image segmentation . . . 68
2.3.1 Classical methods . . . 69
2.3.2 DL approaches . . . 70
2.3.3 UNet . . . 72
2.4 Physics Informed Neural Network . . . 73
2.4.1 PINNs for solving Boundary Value Problems . . . 74

3 A segmentation model for Ventral Intermediate nucleus identification 77
3.1 Brain MRI and Transcranial Magnetic Resonance guided Focused Ultrasound treatments for Essential tremor patients . . . 79
3.1.1 Brain Magnetic Resonance Imaging . . . 79
3.1.2 TcMRgFUS treatment for ET patients . . . 81
3.2 General tractography pipeline for tcMRgFUS treatment planning . . . 84
3.2.1 T1 preprocessing . . . 84
3.2.2 DTI preprocessing . . . 85
3.2.3 Probabilistic tractography and connectivity maps . . . 85
3.2.4 Probabilistic tractography timing . . . 86
3.3 DeLTA-BIT version 1 . . . 87
3.3.1 Materials and Methods . . . 88
3.3.2 Results . . . 94
3.3.3 Discussion . . . 99
3.4 DeLTA-BIT version 2 . . . 100
3.4.1 Materials and Methods . . . 101
3.4.2 Results . . . 106
3.4.3 Discussion . . . 112
3.5 Conclusion and future perspectives . . . 113

4 Validation and clinical implementation of an AI-driven segmentation tool and design of a quality assurance pipeline 115
4.1 Radiotherapy and Image Contouring . . . 117
4.1.1 Overview of Radiotherapy Modalities . . . 117
4.1.2 Impact of Radiation Modality on Organ-at-Risk Contouring . . . 118
4.1.3 ROIs contouring in radiotherapy treatment planning . . . 120
4.2 Materials and Methods . . . 121
4.2.1 Setting . . . 121
4.2.2 Metrics definition and Tolerances . . . 123
4.2.3 TheraPanacea ARTPlan configuration . . . 125
4.2.4 QA protocols design . . . 126
4.2.5 Protocols acceptance criteria . . . 127
4.2.6 QA pipeline . . . 129
4.3 Results . . . 130

4.3.1 Acceptance QA results . . . 131
4.4 Discussion . . . 136
4.5 Conclusion and future perspectives . . . 141

5 Physics Informed Neural Network in Fricke Dosimetry 143
5.1 Fricke gel dosimetry . . . 146
5.1.1 Gel compound . . . 146
5.1.2 Radiation chemistry in Fricke gel . . . 146
5.1.3 Fricke Gel readout techniques . . . 147
5.1.4 Diffusion in Fricke Gels . . . 150
5.1.5 Auto-oxidation in Fricke Gels . . . 152
5.2 Numerical Simulation of Fricke Gel System Evolution . . . 152
5.2.1 Boundary Value Problem definition . . . 152
5.2.2 Forward in Time Centered in Space numerical scheme . . . 153
5.2.3 1D Numerical Simulations . . . 154
5.2.4 2D Numerical Simulations . . . 155
5.2.5 3D Numerical Simulation . . . 157
5.3 General PINN model algorithm for Fricke Gel . . . 158
5.3.1 Backward in time diffusion BVP . . . 159
5.3.2 PINN algorithm for distribution de-blurring . . . 160
5.3.3 Network architecture and training parameters . . . 162
5.3.4 Solution stability and reproducibility . . . 163
5.3.5 Characteristic time scale . . . 164
5.4 Evaluation methods and metrics . . . 165
5.5 Preliminary results . . . 166
5.5.1 1D distributions: feasibility study . . . 166
5.5.2 Higher dimension extension . . . 168
5.6 PINN method on MTB-PVA gel: application to FGs affected by auto-oxidation . . . 174
5.6.1 Gel preparation . . . 175
5.6.2 Irradiation . . . 175
5.6.3 Measurements System . . . 175
5.6.4 Data processing . . . 177
5.6.5 AO and diffusion coefficients estimation . . . 178
5.6.6 PINN training design . . . 182
5.6.7 Results . . . 183
5.7 Discussion . . . 184
5.8 Conclusions and Future perspectives . . . 186

Conclusions and Future perspectives **187**

5.9 Scientific publications 189

5.9.1 A segmentation model for Ventral Intermediate nucleus identification 189

5.9.2 Validation and clinical implementation of an AI-driven segmentation tool and design of a quality assurance pipeline 190

5.9.3 Physics Informed Neural Network in Fricke Dosimetry 191

Acknowledgments **193**

Bibliography **199**

List of Acronyms **221**

A Acceptance QA report **223**

Abstract

The aim of the present PhD project is the development and validation of Artificial Intelligence (AI) models for applications in the medical field, specifically within the domain of Medical Physics. Three main applications are addressed: (i) the development and validation of a segmentation model for the Ventral Intermediate Nucleus (VIM), designed to support Essential Tremor treatments, enable real-time treatment planning, and enhance personalized medicine; (ii) the development of three Quality Assurance (QA) protocols, along with a dedicated software pipeline, for AI-powered auto-contouring tools in Radiotherapy; and (iii) the development of Physics-Informed models capable of mitigating the blurring effect in dosimetry applications involving Fricke hydrogels.

Essential Tremor (ET), the most common movement disorder in adults, significantly impacts daily activities due to involuntary tremors affecting the hands, head, voice, or limbs, thereby reducing quality of life. Although its exact pathogenesis remains unclear, when pharmacological treatments fail, surgical interventions targeting the VIM nucleus of the thalamus can effectively suppress tremor. Several treatment techniques are available, varying in invasiveness; regardless of the technique adopted, treatment planning based on Magnetic Resonance imaging is mandatory. The objective is to identify a Region of Interest (ROI) encompassing the VIM. Existing methods for VIM identification are primarily atlas-based or tractography-based, each with specific advantages and limitations. The aim of application (i) was to develop, train, and validate a tractography-informed Deep Learning model capable of identifying the VIM in a short time while preserving patient specificity.

The second application concerns Radiotherapy treatment planning, where the absorbed dose of ionizing radiation must be accurately evaluated prior to treatment delivery. A comprehensive evaluation of treatment efficacy and associated risks requires accurate contouring of both tumor volumes and organs at risk (OARs), a task that is typically time-consuming as it is performed manually by clinicians. Following the Deep Learning revolution, several AI-powered tools have been introduced to assist in this process. In this framework, the author's main contribution in application (ii) was the design and development of three QA protocols for a clinically implemented auto-contouring software.

Fricke hydrogels belong to the class of chemical dosimeters and are of great interest in Radiotherapy dosimetry due to their ability to map spatial dose distributions. Despite their advantages, they are affected by ion diffusion, which leads to a blurring effect that limits their temporal applicability. Most existing research has focused on optimizing chemical compositions to reduce diffusion. In contrast, the work presented in application (iii) explores a post-measurement AI-based approach to mitigate this effect in the framework of the Physics-Informed Machine Learning.

Overall, the results obtained during this PhD program support the clinical translation of AI in the medical field, while emphasizing the necessity of rigorous validation procedures and the continuous human oversight required to ensure model reliability and safety.

INTRODUCTION

Nowadays, AI has permeated almost every aspect of our lives: we continuously use well-known generative chatbots to seek help in writing emails or checking the grammar of our texts; we read about new discoveries and research projects that make use of AI; and, on the other hand, we are constantly subjected to cookie requests that allow technology companies to collect data.

AI is pervasive across many domains. Most people are not fully aware of what AI actually is, and for professionals in the sector it is often difficult to explain both its mechanisms and its limitations. As described in literatures [1], expectations on AI - driving investments worth billions of dollars - frequently exceed its actual capabilities. This trend has gone so far that some analysts, such as Michael Burry, think that AI may be creating a significant *economic bubble*, to the point that certain investors are betting against the most prominent AI companies (Osmond Chia. "Trader who inspired The Big Short bets against AI as tech shares fall". BBC News, 05/11/2025).

AI crises are nothing novel. The so-called *AI winters* -periods in which economic investments in AI sharply declined - have characterized the field since its birth. In general, these winters occurred when applications reached the limits of the technology and knowledge available at the time. Today, this may be happening once again.

To mitigate the effects of a potential new crisis, it is essential that scientific researches should prioritize enhancing the *reliability, accountability, and traceability* of developed models, while ensuring *safety, realism*, and *robustness*. At the same time, it is imperative that institutions undertake greater efforts to establish clear regulatory frameworks for AI, so as to prevent misplaced trust in the technology from fostering ignorance and disbelief. At present one of the most comprehensive regulatory plan in place is the European Union's AI Act, which entered into force on 1 August 2024 and will become fully applicable in July 2026, with certain provisions enforced earlier (February 2025) and others later (up to 2027).

AI applications in the medical field became particularly significant during the 1990s, when many countries started national mammography screening programs. Managing large amounts of imaging data became increasingly challenging, thus motivating the

development of computational systems to support diagnosis. Since then, an important line of research has focused on developing AI models for mammography classification. Many of the imaging models developed in the broader computer vision field were later adapted to mammography tasks, as in the case of the most popular Convolutional Neural Network (CNN) such as AlexNet, VGG, ResNet, and DenseNet [2–4].
Still within the medical field, automatic lesion and organ segmentation has represented another major driving force for AI research. Particularly noteworthy is the case of nnU-Net, developed by Fabian Isensee *et al.* [5]. Rather than introducing a new network architecture, nnU-Net is a deep learning framework that automatically adapts preprocessing, training, and hyperparameter configurations for biomedical 2D and 3D segmentation tasks. Thanks to its flexibility and robustness, nnU-Net achieved state-of-the-art performance across several international medical image segmentation challenges, becoming one of the most widely adopted frameworks in medical imaging AI research [5].
More recently, the rapid increase in computational power and the availability of large annotated datasets have further accelerated the integration of AI into clinical practice, extending its applications beyond diagnosis to prognosis prediction, treatment planning, radiomics, and personalized medicine [6–8].

In the medical field, the AI impact is almost a reflection of broader societal dynamics. Companies are producing and commercializing AI-powered software with ambitious capabilities, yet their actual performance on real-world or in-house data is often left unvalidated. Such a lack of rigorous assessment and empirical validation inevitably results in financial waste for Health Institutes and, more critically, undermines the quality of clinical practice and patient care. Addressing these shortcomings requires systematic evaluation frameworks and a stronger emphasis on evidence-based integration of AI technologies into healthcare. These tasks require a well-defined professional role, which could be associated with the Medical Physicist. In fact, this profession is closely related to safety-related concepts, thanks to its expertise in radioprotection, quality control of imaging systems, and quality assurance of treatment protocols, such as those applied in radiotherapy. In this context, the Italian Association of Medical Physics (AIFM) promotes this idea both through the issuance of guidelines and through the organization of educational programs on the application of AI in the medical sector.

To understand how concepts such as *accuracy, reliability, and robustness* can be applied in clinical practice to ensure patient safety, and how they were addressed in this PhD research, it is necessary to take a step back in time—before AI was invented and even before the term itself was coined. One must return to the person who first asked the fundamental question: *"Can machines think?"*. This question was first raised by Alan Turing in his famous work entitled *"COMPUTING MACHINERY AND INTELLIGENCE"* [9]. Alan Turing proposed exploiting the *imitation game* as a criterion for establishing whether a machine can exhibit intelligent behavior. In this

game, there are three subjects: a man (A), a woman (B), and an interrogator (C). The object of the game for the interrogator is to determine which of the other two is the man and which is the woman, based on the answers that A and B provide. A machine can be considered "intelligent" if, after either A or B has been replaced by the machine, the number of times the interrogator loses the game remains unchanged.

Later, Edward Feigenbaum introduced the *Expert Turing Test*, also known as Feigenbaum test, a Turing Test performed by an expert in the domain for which the AI system produces outputs [10, 11].

In the medical field, these aspects assume great importance, since AI-generated outputs have direct implications for patient care. First, any output—whether diagnostic or therapeutic in nature—cannot be blindly accepted without human verification. Second, any AI application in medicine should undergo a rigorous validation process, carried out by means of the "Expert Turing Test". As an example, evaluating the performance of an AI-powered human organ auto-contouring system requires comparison with manual contouring performed by a domain expert, such as a radiologist. At the same time, the validation process must be conducted using appropriate metrics, in accordance with established legal guidelines and the relevant scientific literature.

The main aim of this Thesis is the development and validation of AI models to be applied in the medical field, specifically within the domain of Medical Physics. Three applications are reported: (i) the development and validation of a segmentation model for the Ventral Intermediate Nucleus, designed to assist Essential Tremor treatments, enable real-time treatment planning, and enhance personalized medicine; (ii) the development of a Quality Assurance protocol, along with a software pipeline, for AI-powered auto-contouring tools to be applied in Radiotherapy, specifically dedicated to the segmentation of organs at risk; (iii) the development of a Physics-Informed model capable of mitigating the blurring effect in dosimetry applications involving Fricke hydrogels.

ET, the most common movement disorder in adults, significantly impacts daily activities due to involuntary trembling of the hands, head, voice, or limbs, affecting quality of life [12,13]. The exact pathogenesis remains unclear [14], and when medication fails, surgical intervention targeting the VIM nucleus of the thalamus can suppress tremor [15]. There are several treatment techniques available for this purpose, more or less invasive. Although Deep Brain Stimulation is the gold standard [16], Transcranial Magnetic Resonance guided Focused UltraSound (TcMRgFUS) treatment has emerged as a non-invasive alternative offering precise, incision-free brain lesioning [17, 18].
Regardless of the adopted technique, a treatment planning, performed on Magnetic Resonance images, is always mandatory. Here the aim is to identify a ROI, which includes the VIM. It must be said that VIM is a very small area, and at the moment precise identifications require specific acquisition protocols and scanner (7T) normally

not performed and not available in clinical practices. However, during the treatment, a precise indication of the VIM location is provided by direct feedback of the patient. So, in order to reduce the treatment duration, it is essential reducing the region to explore. There are different methods to identify ROI containing VIM, mostly based on atlas or tractography. Both methods have advantages and disadvantages: atlas-based methods are fast but not patient specific; tractography-based methods are very patient specific but very time consuming. The author's aim in (i) was to develop, train and validate a Deep Learning model tractograph-informed able to identify the VIM very rapidly, making its outputs patient specific. The full work will be discussed in Chapter 3.

The second application regards the radiotherapy treatment planning, where the Absorbed Dose of ionizing radiation must be evaluated before treatment delivery. It is mandatory to ensure that the dose delivered to the tumor mass is sufficient to induce cell necrosis. At the same time, it is equally mandatory to verify that the dose delivered to healthy tissues—specifically to the Organs at Risk (OARs)—does not exceed a tolerance threshold, beyond which ionizing radiation could cause severe injuries. A proper evaluation of both treatment efficiency and the associated risks must therefore be conducted prior to delivery, which requires contouring of tumor masses and OARs. This operation is generally very time-consuming, as it is performed manually by doctors and technicians. In recent years, companies have commercialized AI-powered tools to assist operators in manual contouring. *ArtPlan* by *TheraPanacea* (Paris, France) is an auto-contouring software designed for OARs. This software has been acquired by the MedAustron ion beam therapy center, located in Wiener Neustadt, Lower Austria.
The author carried out a six-month research experience abroad at MedAustron and was involved in a project aimed at implementing ArtPlan into the clinical workflow. The author's main task was to design and develop a protocol for the Quality Assurance of ArtPlan. At present, there is a lack of literature specifically addressing this topic: most works refer only to general guidelines, and no clearly defined protocols have been adopted. For this reason, the (ii) author's work, described in Section 4, represents the current *State of the Art*.

Fricke hydrogels belong to the family of chemical dosimeters. They consist of an aqueous solution of ferrous sulfate embedded in a gel matrix. When ionizing radiation interacts with the gel molecules, ferrous ions are oxidized into ferric ions. This chemical change can be mapped providing three-dimensional measurements—rather than point measurements. Since measurements are based on ferric ions position, blurring effects due to ions diffusion limit their applicability in the radiotherapy field. Most of the research on Fricke gels has focused on exploring chemical compositions to limit diffusion. In contrast, the author's research was conducted by developing a post-measurement AI-powered technique designed to mitigate the diffusion effects. After a failed attempt to recover the corrected distribution using a *denoising* technique, the author explored the possibility of incorporating physical laws into the training process of an AI model.

The ion motion responsible for the blurring effect can in fact be described as a *Boundary Value Problem* governed by the *Diffusion equation*. By exploiting this information, it is possible to train models that are more robust and accurate. This approach is known as Physics-Informed Machine Learning. A neural network trained with this method is referred to as a Physics-Informed Neural Network (PINN). The third application focuses on the development of PINN models capable of recovering the un-blurred distribution—also referred to as the *initial distribution*—given the blurred one - also referred to as the *final distribution* - along with the physical parameters of the gel system. This approach has several implications, including mathematical ones, which are fully described in Section 4.

CHAPTER 1

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

This chapter aims to introduce the reader to the fundamental concepts of AI, which are essential for understanding the analysis conducted by the author.
Artificial Intelligence is a scientific discipline, and therefore a complete description cannot be provided within the scope of a single chapter of a PhD thesis. Instead, only an overview will be presented, with a particular focus on Deep Learning and Neural Network.

Any educational program concerning AI in general begins with the explanation of a scheme such as the one reported on Figure 1.1.

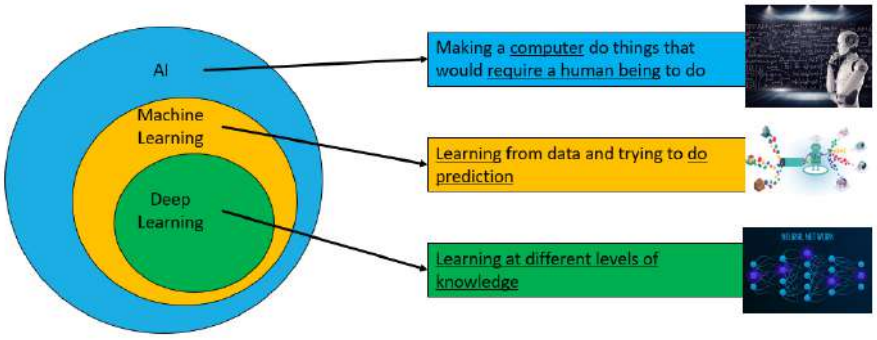


Figure 1.1: Traditional representation of the hierarchy of AI subfields.

Following the most common introduction, we can say that AI is the most general ensemble of computational applications that enable a computer to perform tasks normally carried out by humans. From this definition, computation itself could be regarded as AI. Indeed, looking back in time, this represents the foundational level of Turing's vision.
AI has several subfields, one of which is known as Machine Learning (ML). ML refers to learning from data. We may have a collection of data (a *dataset*), organized in pairs of input and output. As an example, we may consider a set of pairs consisting of voltage

and current intensity measurements, where voltage values are treated as inputs and current intensity values as outputs. ML can be seen as the process of discovering the unknown relation between input and output. In the previous example, this relation could correspond to Ohm's law.
Of course, this approach requires that a "zero-state model" (to be trained) is defined and used as an approximator for the relation we want to identify. In the voltage–current example, this could be a *linear model* with non-initialized parameters. When we use a *Neural Network* (NN) as an approximator, we enter the domain of Deep Learning (DL). It must be underlined that the AI revolution we are experiencing in this and last decades concerns mostly DL applications.

In the following sections AI and ML will be briefly introduced. DL will be discussed in detail in the next chapter.

1.1 General overview on Artificial Intelligence

A simple search for the term "Artificial Intelligence" on the web can provide several definitions, such as the one offered by Wikipedia (Italian version):

> The ISO/IEC 42001:2023 standard Information technology – Artificial intelligence Management System (AIMS) defines artificial intelligence as "the capability of a system to exhibit human abilities such as reasoning, learning, planning, and creativity." [19]

Another result is (Wikipedia, English version):

> Artificial intelligence (AI) is the capability of computational systems to perform tasks typically associated with human intelligence, such as learning, reasoning, problem-solving, perception, and decision-making. It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals [20].

The author believes that the best way to understand AI clearly is to start from the event in which the term was coined: the Dartmouth Conference (1956). At this event, personalities such as Claude Shannon, John McCarthy, Nathaniel Rochester, and Marvin Minsky took part, and the well-known *Proposal for the Dartmouth Summer Research Project on Artificial Intelligence* [21] was defined. This proposal had the main aim of:

> *We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so*

> *precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.* [21]

Specifically, at this event John McCarthy introduced the term "Artificial Intelligence" for the first time.

Regardless of the exact definition, AI research aims to realize Turing's vision: machines that execute tasks on behalf of humans. This idea was not Turing's own; it was already known before his work. He cites Lady Lovelace (Augusta Ada King, née Byron, 10 December 1815 – 27 November 1852), who is known to be the first programmer in the History. She wrote:

> [The Analytical Engine] might act upon other things besides number, were objects found whose mutual fundamental relations could be expressed by those of the abstract science of operations, and which should be also susceptible of adaptations to the action of the operating notation and mechanism of the engine...Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and of musical composition were susceptible of such expression and adaptations, the engine might compose elaborate and scientific pieces of music of any degree of complexity or extent [22, 23].

It is clear that up to this point the author has provided only historical notes, which have guided humankind toward AI as we know it today. The remainder of the section will be dedicated to the historical results achieved by AI. It remains for the reader to develop an idea of what artificial intelligence actually is.

1.1.1 Historical results achieved by Artificial Intelligence

As mentioned by Professor Giosuè Lo Bosco (University of Palermo), in his lectures on AI, the first generation of AI scientist made the subsequent predictions:

- H. A. Simon and Allen Newell (1958): *"within ten years a digital computer will be the world's chess champion and within ten years a digital computer will discover and prove an important new mathematical theorem."*
- A. Simon (1965): *"machines will be capable, within twenty years, of doing any work a man can do."*

- Marvin Minsky (1967): *"within a generation, the problem of creating "artificial intelligence" will substantially be solved."*
- Marvin Minsky (1970): *"in from three to eight years we will have a machine with the general intelligence of an average human being."*

Of these predictions, only one eventually came true, though with a delay of two decades. This is the case of *Deep Blue*, which defeated Garry Kasparov, the world chess champion at that time (1997). A photograph of the event is shown in Figure 1.2.

Figure 1.2: Photo depicting Garry Kasparov losing to Deep Blue.

Another prediction was only partially achieved: "within ten years a digital computer will discover and prove an important new mathematical theorem". The *Logic Theorist* (1956) was able to prove and even find new demonstrations for already known theorems, but none of them was referred to as a *new theorem*. For the other predictions, AI has made significant progress, but what is known today as *Artificial General Intelligence* (AGI), which companies such as OpenAI aim to achieve, has not yet been achieved and we are far from that result.

An important result, often overlooked, was achieved in the *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), which began in 2010. This challenge is based on the well-known dataset of images called *ImageNet* [24]. The task consists of developing AI models capable of performing object detection in images. The first edition (2010) included a dataset of approximately 20,000 images, with 1,000 object categories. In 2012, for the first time, a convolutional neural network named *AlexNet* won the challenge, demonstrating the superiority of Deep Learning models over classical ones, such as *Support Vector Machines*. In 2014, Google's model *GoogleNet* achieved performance close to that of humans, and in 2015 these results were surpassed by a model developed by a Microsoft team, *ResNet*. It should be emphasized that, as Olga Russakovsky pointed out, the ILSVRC covers only 1,000 categories; humans can recognize a far larger number of categories and, unlike programs, can also interpret the context of an image [25].

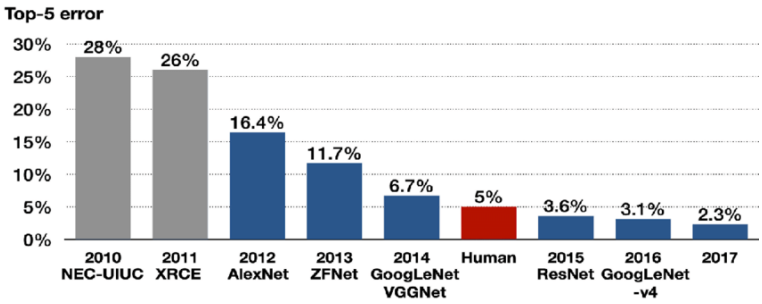


Figure 1.3: Algorithms that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2010–2017. The algorithms with blue graph are convolutional neural network. Image taken from [26].

As the latest result, the author wishes to mention the birth of *ChatGPT* (Chat Generative Pre-trained Transformer), which was released in November 2022. The chatbot developed by OpenAI is credited with accelerating the *AI boom*, thanks to its extraordinary performance, which has profoundly changed the world, shifting economic attention toward AI.

1.2 General overview on Machine Learning

As already mentioned, ML refers to learning from data. Before we embark on a discussion of what is meant by learning, it is useful to consider why ML is substantially different from general AI. In programming it is possible to distinguish two different paradigms: on the one hand, we have rule-based models; on the other hand, the rules must emerge from models *training* process.

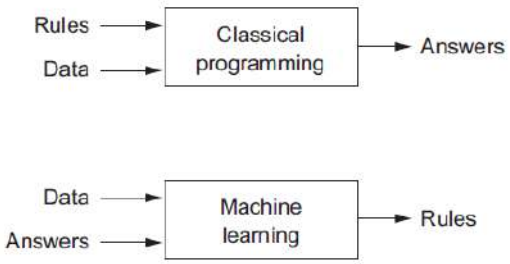


Figure 1.4: Scheme illustrating the difference between general AI and ML: a general AI model makes decisions based on an input and predefined rules, whereas an ML model is trained by being provided with questions and answers, with the aim of discovering the underlying rules. Figure adapted from [27].

Now we are ready to face the learning question. Lets us start with a very simple example proposed by the reference [27]. Suppose we have a planar points distribution, defined by providing the x and y-coordinates as couples (x, y). Suppose some of these points are labeled as "black" and the others as "white", as depicted in figure 1.5. Now suppose we aim to train an ML model capable of predicting a label ("black" or "white") given a new point (x_1, y_1). In ML terminology, this is called a *classification* task. The reader may now ask: *"What should the model actually learn?"*. Conceptually, we would like the model to analyze the distribution and identify *patterns* that can, at least,

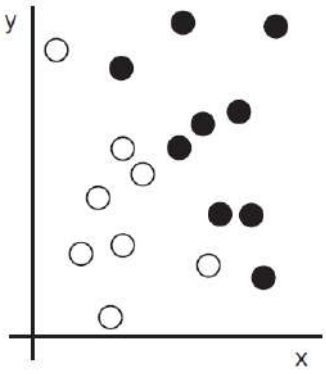


Figure 1.5: Points distribution: points labeled as "black" and "white" [27].

provide a rationale for why some points are labeled "black" and others "white". At first glance, one must admit that there seems to be no apparent reason for a point to be "black" or "white". What happens if we apply a geometrical transformation to our system of points, as depicted in Figure 1.6?

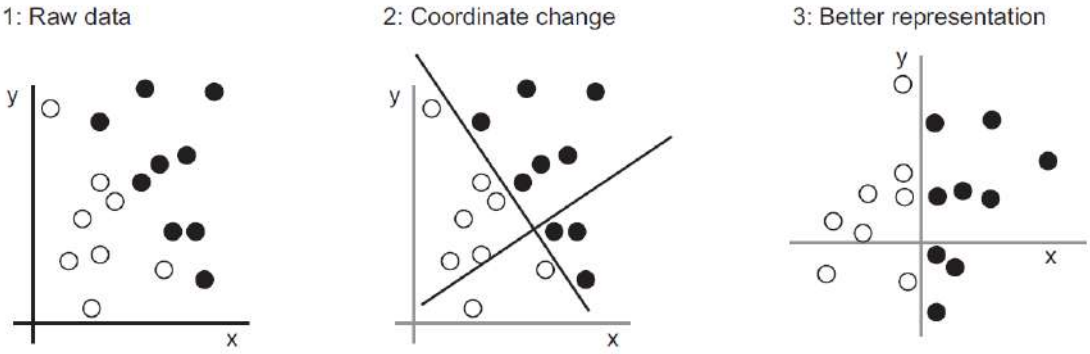


Figure 1.6: Applying geometric transformations to obtain improved representations [27].

In the new representation, the overall picture becomes clear: points with $x > 0$ are labeled "black", whereas those with $x < 0$ are labeled "white". The essence of the machine learning problem lies in discovering the transformation that most effectively represents the data. In the same way, in an ML classification task the problem lies in finding the *hyperplane* that separates the data. At this point, it is necessary to understand how this transformation can be found.

Understanding whether a machine can exhibit cognitive processes may lead to conjectures as questionable as those raised by the *"can a machine think?"* question. Analogously to Turing's approach, Tom M. Mitchell in his book proposed a definition of the learning process that reformulates the question in an experimental way, allowing us to verify whether a machine has actually learned. This definition is:

Definition 1. *A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its*

performance at tasks in T, as measured by P, improves with experience E. [28]

Lets us say something about this definition.

1.2.1 Machine Learning tasks

In machine learning, a task specifies what we want the model to learn. Two broad categories can be identified: **classification** and **regression** tasks. In classification tasks, the model's outputs are restricted to a discrete set of possible values, whereas in regression tasks the output is continuous. Both categories encompass a wide range of specific tasks.
Here, some examples of classification tasks are presented:

- **Simple classification**: the model is provided with inputs, which may be more or less complex and structured, and must assign each input to one class within a predefined set. When only two classes are present, the task is referred to as *binary classification*. Examples include classifying patients as healthy or unhealthy based on clinical outcomes or diagnostic images.

- **Object detection**: the model receives images and must identify the presence of one or more objects (belonging to predefined classes) and output bounding boxes localizing them. Examples include text-detection tools and perception modules in autonomous-driving systems.

- **Anomaly detection**: the model learns recurrence patterns from a dataset and issues an alert when a new observation is incompatible with the learned distribution. Examples include bank fraud-detection mechanisms and beam-quality assurance systems in CERN Large Hadron Collider experiments.

- **Image segmentation**: the model is provided with images and must detect and delineate one or more objects within them. This task can be interpreted as a simple classification problem applied at the pixel level.

Some examples of regression tasks include:

- **Simple regression task**: the model is provided with inputs, which may be more or less complex and structured, and must predict new values associated with new inputs. Typical examples include stock price estimation and the prediction of missing values in a dataset.

- **Denoising**: the model is provided with pairs of clean and noisy data corresponding to the same measurements, and must learn to produce the clean version when given the noisy one. The main difference from simple regression is that input and output share the same data type.

- **Machine translation**: the model is provided with a text in one language and must produce a text in another language with the same meaning.
- **Image generation**: the model is provided with a textual description and must produce a corresponding image.
- **Image translation**: the model is provided with an image in one representation and must produce an image, in the same context, but in a different representation. Examples include transforming a zebra into a horse or generating deepfakes.
- **Transcription**: the model is provided with images and must generate a textual description of their content.

A further ML task, not mentioned above, is **clustering**. It is a classification task in which the number of classes is generally not known *a priori*. Moreover, all the tasks discussed so far require *supervised* training, meaning that the model must be provided with the correct output associated with each input during training. Clustering, from this perspective, is fundamentally different, as it relies on *unsupervised* training, meaning that unlabeled inputs are provided during the training.

1.2.2 Machine Learning Experience

A first point to clarify is what the experience represents for a machine. In ML, the experience corresponds to the dataset, that is, the collection of data, inputs and outputs, provided to the model for training and testing. Once the task is defined, the type of data that can be used is necessarily constrained. For example, if we want to train a model to classify whether lung images contain some cancer sign or not, we need lung images paired with labels indicating the presence or absence of the pathology. Similarly, if we aim to segment OARs in Computed Tomography (CT) images, we require a dataset of labeled images, e.g. CT scans associated with binary masks corresponding to each OAR.
In ML terminology, the term *label* typically refers to the ground truth. One of the major challenges in modern ML applications is the lack of labeled datasets. In fact, in many research domains it is possible to collect very large datasets, but these are often unlabeled. This represents a significant limitation, to the point that labeled datasets in advanced applications are sometimes considered the *gold (or oil) of the 21st century*. Every time we accept cookie requests on websites or social networks, we are effectively giving away our *gold* for free, data on which AI companies can build value and generate profit.

In ML we are interested in generalizing the model's capability, i.e. extending what the model learns during training to *unseen* data. This requires a proper evaluation of the *generalization error*, which must be performed on data that are not fed to the model during training. For this reason, when designing an ML project, the dataset must first

be split into a *training set*, containing the data used during training, and a *test set*, containing the data used for the final evaluation of the model. Under no circumstances should the test set be mixed with the training set: once the split is performed, the test set must be kept completely separate from the model and can only be used at the end, during the evaluation phase.
It may happen that we have only a small dataset, for which training and evaluation cannot rely on sufficiently robust statistics. In such cases, one may choose to apply *k-fold cross-validation*. This technique consists of dividing the entire dataset into k subsets (or *folds*) and training the model k times. In each iteration, $k-1$ folds are used for training and the remaining fold is used for testing, with the folds being permuted at each step. At the end, the overall performance and the selection of the best model can be obtained by computing an ensemble average over the validation results.

1.2.3 Machine Learning Performance

Once the tasks and the experience have been defined, we can move on to the concept of **performance**. In the first instance, performance measurements rely on the evaluation process. Assuming that we have trained an ML model, we want to estimate how far the model is from the desired target. This can be done by adopting metrics that quantify the error committed by the model. Examples of such metrics include the Mean Squared Error (MSE), the Accuracy (ACC), the Dice Similarity Index (DSI), and so on.
Here it is worth noting that these metrics belong to different task domains: MSE is typically used in regression tasks, ACC in classification tasks, and DSI in segmentation tasks. The MSE is defined as:

$$MSE = \frac{1}{N}\sum_{i}(y_{i,true} - y_{i,pred})^2 \tag{1.1}$$

where N is the number of test samples, $y_{i,true}$ is the true label of the i-th input, and $y_{i,pred}$ is the label predicted by the ML model for the same input. It is evident that this type of metric provides global information about how far the model is from the true label. In general, this metric can be used for any task, but one must acknowledge that the information it provides is rather compressed; for example, in classification tasks it does not offer a clear measure of the model's error.
For classification tasks, ACC is a better choice, since it measures the ratio between correct predictions and all predictions.

Definition 2 (Classification Notation)**.** Suppose we have a binary classification problem; let us define:

- True Positive (TP), the number of samples for which the model correctly predicts a positive outcome;

- True Negative (TN), the number of samples for which the model correctly predicts a negative outcome;
- False Positive (FP), the number of samples for which the model predicts a positive outcome while the true label is negative;
- False Negative (FN), the number of samples for which the model predicts a negative outcome while the true label is positive.

With these definitions, it is easy to verify that the ACC ratio can be expressed as:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{\text{all data}}. \tag{1.2}$$

When we have more than two classes, we must compute the ACC for each class and then take the average over all classes. Although ACC can represent ML results better than simple MSE, some considerations must be taken into account. The first concerns the class balance of the dataset. It is very common for datasets to exhibit different class distributions; in such cases, we say that the dataset is *unbalanced*. As an example, consider a binary classification dataset with 20% positive outcomes and 80% negative outcomes. This situation occurs frequently in medical applications. In a case like this, a model that (incorrectly) predicts only negative outcomes would still achieve an ACC of 80%, even though it has learned nothing meaningful.
In such scenarios, it is essential to compute accuracy by weighting according to the class distribution; otherwise, the resulting performance estimates may be heavily overestimated. This type of mistake is among the most common in the AI community. For this reason, whenever we design an ML model, a preliminary statistical analysis of the dataset is strongly recommended.
The previous example implicitly shows why ACC cannot be used for segmentation evaluation. When we want to compare two Regions of Interest (ROIs) - for example, the contouring of the same organ, one manually delineated and one predicted - we can use the same notation introduced above (Definition 2). We call TP the number of pixels or voxels (the 3D extension of a pixel) that belong to the organ in both contourings. We call TN the number of pixels (or voxels) that belong to the background in both contourings. Finally, FP and FN denote the pixels that are misclassified, in accordance with the classification notation. In general, when the area (or volume) ratio $|ROI|/|Background|$ ($|\cdot|$ denotes the size of the region) is small, it means that the number of negative pixels (or voxels) is large. In this condition, which is also the most frequent, a model that does not detect any organ would still obtain a high ACC. In fact, we would have $TP = 0$, $TN = |Background|$, $FP = 0$, and $FN = |ROI|$, which implies:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TN}{TN + FN} = \frac{|Background|}{|Background| + |ROI|}. \tag{1.3}$$

Since, by assumption, the ratio $|ROI|/|Background| \to 0$, we would obtain $ACC \to 1$. In segmentation tasks, overlap metrics are much more useful and representative. These metrics provide a measure of the area (or volume) of the intersection relative to the ROI's area (or volume). Suppose we have again two ROIs to compare; let us call ROI A the ground truth (GT) and ROI B the prediction. We define the DSI as:

$$DSI = 2\frac{|A \cap B|}{|A| + |B|}. \tag{1.4}$$

Often, DSI is also referred to as Dice Similarity Coefficient (DSC). It is easy to verify that a perfect overlap yields $DSI = 1$, whereas zero overlap yields $DSI = 0$. Another overlap metric is the Jaccard Index (Jacc), defined as:

$$Jacc = \frac{|A \cap B|}{|A \cup B|}. \tag{1.5}$$

Once again, a perfect overlap gives $Jacc = 1$ and zero overlap gives $Jacc = 0$. However, Jacc is generally considered more stringent; for this reason, DSI is usually preferred. For completeness, adopting the classification notation (Definition 2), DSI and Jacc can be rewritten in a more handy form. Since TP represents the intersection, and $|A \cup B| = TP + FP + FN$, $|A| = TP + FN$, and $|B| = TP + FP$, we obtain:

$$DSI = 2\frac{TP}{2TP + FP + FN}\ , \qquad Jacc = \frac{TP}{TP + FN + FP}. \tag{1.6}$$

Going back to the considerations regarding the use of ACC in classification tasks, the author has so far mentioned only the aspect related to class balance. Another important aspect to take into account concerns the type of information that ACC can provide. ACC describes, in general, the percentage of correctly classified inputs, but it does not convey any information about the *precision* or the *sensitivity* of the trained model. These two concepts can be expressed through the following metrics.
Precision concerns the model's ability to provide positive predictions that are truly positive. This can be represented by the ratio between TP and the number of inputs classified as positive by the model, i.e. $TP + FP$. Conversely, sensitivity concerns the model's ability to identify positive outcomes among all actual positives. Analogously, this is represented by the ratio between TP and all positive samples, i.e. $TP + FN$. Summarizing, we can introduce two additional metrics that provide very useful information:

$$Precision = PPV = \frac{TP}{TP + FP}\ , \qquad Sensitivity = TPR = \frac{TP}{TP + FN} \tag{1.7}$$

where PPV stands for Positive Predictive Value and TPR for True Positive Rate. Another common name for sensitivity is *Recall*. Other metrics, such as *Specificity* or *Miss Rate*, can be easily derived using similar considerations.

What has been discussed so far concerns the evaluation process. It is evident that, in principle, we can use any metric, as long as it captures the information we intend to represent. However, metrics also play an important role during the training process. Due to the conceptual differences between these uses, the role of metrics during training will be discussed in the next subsection.

Once tasks, experience, and performance have been introduced, the author suggests that readers who are not familiar with the AI field read again the Definition 1.

1.2.4 Loss Functions and Optimization

At this point, it is important to address the delicate question of "how a model can be trained." Taking inspiration from early learning theories, such as *Behaviorism*, we might consider training a model using a system of reinforcement and punishment. This forces the author to introduce the concept of a **loss function**. A loss function is a measure of the error committed by the model during the training process; ideally, a perfectly trained model would yield a loss equal to zero. Now, let us assume our ML model depends on a set of parameters θ_i. For example, if our model is a line, it would have two parameters (for 1D inputs and outputs): the slope and the intercept. The **Optimization** is the process of finding the parameters θ_i that make a pre-established loss function as close to zero as possible.

There are different ways to optimize a model; the conceptually simplest is by *exploring the parameter space*: trying all possible combinations and selecting the set which yields the smallest loss value. Although this seems straightforward from a conceptual point of view, one must consider that this approach has strong limitations, the most problematic being the dimensionality of the parameter set. In fact, even for the simplest model (a line model), we would need to explore the plane $\mathbb{R}^2$. If we add a new parameter, the space to explore becomes $\mathbb{R}^3$. In general, if we have N parameters, the space to explore is $\mathbb{R}^N$. Since a common NN can have millions of parameters, such an approach would be totally intractable.

A mathematical concept that can make the problem of minimizing a loss function more manageable is the *Gradient*. Most readers should be familiar with the derivative of a function: from high school, we learn that a positive derivative indicates an increasing function. This means that if we consider a function $\mathcal{L}$ of a single variable at a point x_0, a positive derivative implies that taking a step to the right along the real line results in a value for $\mathcal{L}$ that is greater than before. Conversely, taking a step to the left results in a smaller value. For higher-dimensional functions, the derivative generalizes into the gradient, which represents the *direction of maximum increase* (or steepest ascent). As an example, if we consider the elevation of the Earth's surface as a function of two independent variables, the Earth's coordinates (see Figure 1.7), computing the gradient near Catania would yield a direction that points toward the peak of Mount Etna. In ML training, suppose we have a model $M(x; \theta_i)$ that accepts an input x and is defined

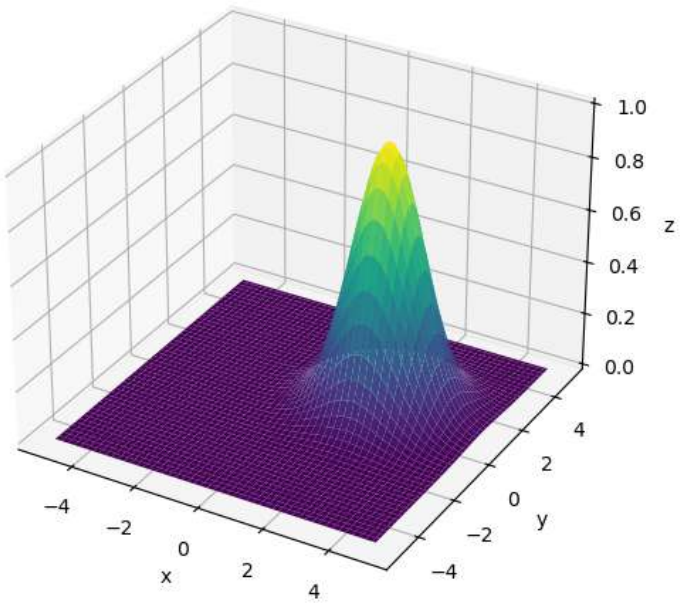


Figure 1.7: A 3D surface plot of a 2D function. Near the peak, the gradient vector points toward the coordinates of the maximum.

by a set of parameters θ_i. Given a loss function $\mathcal{L}(y_{true}, M(x;\theta_i))$ that compares the true label of the input x (y_{true}) with the model output, by computing the gradient with respect to all parameters θ_i, we obtain the direction in the parameter space along which $\mathcal{L}$ increases most rapidly. If we "take a step" in the opposite direction, we obtain a new set of parameters that yields a smaller value for $\mathcal{L}$ than before. Iterating this operation we can solve the optimization problem. This mechanism constitutes the fundamental *update rule* for the model's parameters, which can be expressed mathematically as follows:

$$\theta_{i,t+1} = \theta_{i,t} - \epsilon \nabla_{\theta_i} \mathcal{L} \tag{1.8}$$

where $\theta_{i,t+1}$ and $\theta_{i,t}$ indicate the i-th parameter at the $(t+1)$-th and t-th iterations, respectively. In ML terminology iterations are called *epochs*. $\nabla_{\theta_i}\mathcal{L}$ represents the derivative of $\mathcal{L}$ with respect to the i-th parameter θ_i, and ϵ is called the *learning rate*, representing the amplitude (or size) of the "step".
To emphasize the author's background in Physics, the most general formulation of Newton's second law is given by:

$$m\ddot{\mathbf{r}} = -\nabla U \tag{1.9}$$

where m is the mass, $\ddot{\mathbf{r}}$ indicates the second derivative of the position vector, and U is the potential energy. This expression is so closely related to the update rule explained above that some optimization algorithms, such as *Momentum*, add a velocity term to the algorithm. This makes the optimization process similar to the *equation of motion of a body falling off a cliff*.
Regardless of the optimization method chosen, the parameters must be initialized. This is typically achieved by sampling from random distributions. While, in principle, any distribution can be used, specific ones such as the *Normal* or *Uniform* distributions are often more suitable depending on the particular task and architecture.
In summary, training a ML model via gradient-based methods is analogous to releasing a stone from a precipice, allowing it to descend toward the lowest point of a valley. Although physical descent is governed by only two independent variables (the spatial coordinates x and y), whereas ML models often operate within a high-dimensional parameter space, the fundamental principles of the gradient remain applicable. In both scenarios, the gradient provides the direction of steepest descent, allowing to solve the optimization problem.

To conclude this subsection, the author wishes to mention two essential aspects of optimization problems. In such contexts, the ultimate goal is to find the *global minimum* of the loss function. Generally, only specific models, such as linear ones, guarantee that a global minimum exists and is unique. For more complex models, this is not guaranteed; therefore, the best result we can achieve is often a *local minimum*. It is important to underline that the optimization of a model using a gradient descent approach is in general a *stochastic process*. To make general models more robust, one should perform an exploration of the parameter space by training the model several times with different initializations.
The second aspect relates to computational cost: computing the gradient for each parameter can be extremely expensive in terms of both computational resources and time. The algorithm that enabled this calculation is *Backpropagation*, which was proposed in 1962 by Frank Rosenblatt, although he could not apply it at the time. Paul Werbos later used backpropagation to train *Multi-Layer Perceptrons* in 1982, yet timing issues persisted. For this algorithm to become truly efficient, it had to wait until the last two decades, when these calculations were translated into *parallel computation on Graphics Processing Units* (GPUs).

1.3 Classical Machine Learning Models

The landscape of classical Machine Learning models is vast, ranging from traditional linear models to decision trees and random forests. Since this thesis focuses exclusively DL architectures, and the preceding ML treatment serves primarily to introduce fundamental concepts, the discussion in this section is restricted to linear and logistic

models. For a more comprehensive overview of other Machine Learning paradigms, the reader is referred to Chapter 5 of [29].

1.3.1 Linear Models

At this point, it is important to introduce the simplest class of ML models: linear models. Often, the term "linear model" is mistakenly used to indicate only straight-line relationships, due to the linear correspondence between inputs and outputs. In ML, however, a model is defined as linear whenever the output has a linear relationship with the model **parameters**. In other words, a linear model is any model that can be expressed in matrix form as:

$$\mathbf{y}_{pred} = \mathbf{X}\mathbf{w} \tag{1.10}$$

where:

- $\mathbf{X}$ is the **design matrix**, which contains the input data. If the inputs are scalars (1D), $\mathbf{X}$ is a column vector of size N (the number of samples). If each input consists of multiple *features*, $\mathbf{X}$ is a matrix of size $N \times M$, where M is the number of features.

- $\mathbf{y}_{pred}$ represents the model predictions for each input. If the output is a single scalar, $\mathbf{y}_{pred}$ is a vector of size N; otherwise, it is a matrix of size $N \times L$, where L is the dimension of the output space.

- $\mathbf{w}$ is the **parameter matrix** (or **weight matrix**), which defines the behavior of the model. If a specific entry is larger than others, the corresponding feature has a greater influence on the output. To ensure dimensional consistency, $\mathbf{w}$ must have the size $M \times L$.

Based on this definition, if our goal is to find the best-fitting line for a set of 2D points, the model is expressed as:

$$y_{pred} = mx + b. \tag{1.11}$$

To maintain consistency with our matrix notation, we define:

- $\mathbf{X} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{bmatrix}$;

- $\mathbf{y}_{pred} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$;

- $\mathbf{w} = \begin{bmatrix} b \\ m \end{bmatrix}$.

This specific configuration is known as **Linear regression**. If the inputs consist of multiple features, say M, the design matrix extends to:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \dots & x_{1,M} \\ 1 & x_{2,1} & x_{2,2} & \dots & x_{2,M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N,1} & x_{N,2} & \dots & x_{N,M} \end{bmatrix}. \tag{1.12}$$

Now, considering a general polynomial of degree d:

$$y = \sum_{i=0}^{d} w_i x^i, \tag{1.13}$$

we observe that monomials of different degrees do not combine (they are linearly independent) and can thus be treated as distinct components of a vector. Exploiting this property, a polynomial model of degree d can be framed as a linear ML model, where the design matrix is given by:

$$\mathbf{X} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^d \\ 1 & x_2 & x_2^2 & \dots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_N & x_N^2 & \dots & x_N^d \end{bmatrix}. \tag{1.14}$$

Ultimately, linear models can be utilized to fit polynomials of any degree and dimensionality, providing the flexibility to approximate a vast range of functions. Finding the polynomial that best fits a set of points is known as **Polynomial regression**.

Suppose we have a set of inputs $\mathbf{X}$ associated with outputs $\mathbf{y}_{true}$ and we wish to construct a linear model. From the definition of a linear model (Eq. 1.10), it is evident that our goal is to determine the parameters within the weight matrix $\mathbf{w}$ that minimize a specific loss function $\mathcal{L}(\mathbf{y}_{true}, \mathbf{y}_{pred})$. Given that this is a regression task, we select the Mean Squared Error (MSE), defined as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_{pred,i} - y_{true,i})^2 = \frac{1}{N}\|\mathbf{y}_{pred} - \mathbf{y}_{true}\|_2^2, \tag{1.15}$$

where $\|\cdot\|_2^2$ denotes the squared Euclidean norm. By substituting $\mathbf{y}_{pred}$ with the expression from Eq. 1.10, we obtain:

$$MSE = \frac{1}{N}\|\mathbf{X}\mathbf{w} - \mathbf{y}_{true}\|_2^2 = \frac{1}{N}\left[(\mathbf{X}\mathbf{w} - \mathbf{y}_{true})^T(\mathbf{X}\mathbf{w} - \mathbf{y}_{true})\right], \tag{1.16}$$

where we utilize the fundamental linear algebra identity $\|\mathbf{a}\|_2^2 = \mathbf{a}^T\mathbf{a}$ (the superscript T

denotes the transpose). To find the minimum, we compute the gradient with respect to $\mathbf{w}$ and set it to zero:

$$\begin{aligned} \nabla_{\mathbf{w}} MSE = \nabla_{\mathbf{w}} \left[\frac{1}{N} (\mathbf{X}\mathbf{w} - \mathbf{y}_{true})^T (\mathbf{X}\mathbf{w} - \mathbf{y}_{true}) \right] = 0 \\ \rightarrow \nabla_{\mathbf{w}} \left[(\mathbf{X}\mathbf{w} - \mathbf{y}_{true})^T (\mathbf{X}\mathbf{w} - \mathbf{y}_{true}) \right] = 0, \end{aligned} \tag{1.17}$$

where $\nabla_{\mathbf{w}}$ represents the gradient with respect to each weight in $\mathbf{w}$. Equation 1.17 admits a closed-form analytical solution known as the **Normal Equation**:

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}_{true}, \tag{1.18}$$

where $(\mathbf{X}^T \mathbf{X})^{-1}$ is the inverse of the matrix product $\mathbf{X}^T \mathbf{X}$. The complete derivation is available in Section 5.1.4 of [29].
In general, Equation 1.17 provides a set of *critical points*, which may include maxima, minima, or saddle points. However, we know that the MSE by definition takes values in the range $[0, +\infty)$, making it *positive definite*. Furthermore, from Equation 1.10, we can deduce that the MSE is a *quadratic form* with respect to the parameters $w_{i,j}$. In other words, for linear models, the MSE surface is a **convex hyper-paraboloid**. Consequently, the unique critical point found is guaranteed to be a global minimum, corresponding to the vertex of this geometric structure.

The Normal Equation tells us that, when we want to use a linear model to solve an ML task, exploring the parameter space is unnecessary. In fact, under these specific conditions (linear models and MSE as the loss function), we already have the optimal solution in closed form. This result is highly significant and implies that, if we know *a priori* that inputs and labels are polynomially related, we should not use any other ML model.

To conclude this subsection, it is important to address an aspect that has been neglected so far in this treatment. In ML, the primary objective is prediction, according to the definitions introduced in Sections 1.2.2 and 1.2.3. While this section has approached the problem as a fitting task, the essential difference between ML and general curve fitting lies in the generalization capability on unseen data. This distinction directly translates into the need to evaluate the generalization error of the linear model. As is standard practice in ML, this is achieved by partitioning the initial dataset, composed of the design matrix (Eq. 1.12) and the corresponding true labels $\mathbf{y}_{\text{true}}$, into a training set $(\mathbf{X}^{\text{train}}, \mathbf{y}^{\text{train}}_{\text{true}})$ and a test set $(\mathbf{X}^{\text{test}}, \mathbf{y}^{\text{test}}_{\text{true}})$. In this way, we can apply the normal equations to the training set and subsequently evaluate the performance, and thus the generalization error, on the test set.

1.3.2 Maximum Likelihood Estimation

Linear models have been treated as approximators of an underlying function, where their outputs are seen as predictions of the true labels. Another approach interprets model outputs as the *probability of a value*. Consider a dataset consisting of m examples $\mathbb{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$ drawn independently from a true but unknown real-world distribution $p_{\text{data}}(\mathbf{x})$. Let $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$ be a parametric family of probability distributions over the same space, defined by the parameters $\boldsymbol{\theta}$. In other words, $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$ maps any configuration $\mathbf{x}$ to a real number estimating the true probability $p_{\text{data}}(\mathbf{x})$.
The *Maximum Likelihood Estimation* (MLE) principle asserts that the best estimation for $\boldsymbol{\theta}$ is defined as:

$$\begin{aligned} \boldsymbol{\theta}_{\text{MLE}} &= \arg\max_{\boldsymbol{\theta}} p_{\text{model}}(\mathbb{X}; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}(\mathbf{x}_i; \boldsymbol{\theta}). \end{aligned} \tag{1.19}$$

Since the product of many probabilities can lead to computational issues, such as **underflow**, we observe that the location of the maximum does not change by applying the logarithm or scaling by a constant:

$$\begin{aligned} \boldsymbol{\theta}_{\text{ML}} &= \arg\max_{\boldsymbol{\theta}} \frac{1}{m} \log \prod_{i=1}^{m} p_{\text{model}}(\mathbf{x}_i; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \frac{1}{m} \sum_{i=1}^{m} \log p_{\text{model}}(\mathbf{x}_i; \boldsymbol{\theta}) \\ &= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x} \sim \hat{p}_{\text{data}}} \log p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta}), \end{aligned} \tag{1.20}$$

where $\mathbb{E}$ denotes the expectation (the mean value). This formulation is closely related to two fundamental metrics in *Information Theory*: the Kullback-Leibler divergence, which measures the information lost when one distribution approximates another, and the *Cross-entropy*. Specifically, minimizing the Kullback-Leiber divergence between the data distribution and the model is equivalent to maximizing the log-likelihood. The Cross-entropy $H(P, Q)$ is defined as:

$$H(P, Q) = -\mathbb{E}_{\mathbf{x} \sim P} \log Q(\mathbf{x}). \tag{1.21}$$

In this context, maximizing Equation 1.20 is equivalent to minimizing the cross-entropy between the empirical distribution $\hat{p}_{\text{data}}$ and the model p_{model} [29].

Based on these concepts, it is straightforward to verify that applying the MLE principle to a linear model, assuming its outputs follow a Gaussian distribution, leads directly to the same result provided by the Normal Equation (Equation 1.18). For a

complete derivation, the reader is referred to Section 5.5.1 in [29].

1.3.3 Logistic Regression

The motivation for establishing the mathematical foundation in the preceding subsection is to develop an approach for classification tasks that is analogous to the one used for linear models. In binary classification, the main distinction from regression is that labels are discrete, usually restricted to the set $\{0, 1\}$. Since standard linear models are not naturally constrained to this range, we can re-frame the task as finding the *probability of a positive outcome given a certain input*. To this end, we need a model whose outputs are confined to the interval $[0, 1]$. The simplest approach involves applying a *Sigmoid* function to the output of a linear model. The Sigmoid function is defined as:

$$\sigma(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}. \tag{1.22}$$

By defining the underlying linear model as the product of the design and weight matrices (Equation 1.10), the **logistic model** can be expressed as:

$$\mathbf{y}_{pred} = \sigma(\mathbf{X}\mathbf{w}). \tag{1.23}$$

The process of finding the optimal parameters $\mathbf{w}$ in Equation 1.23 that maximize the likelihood (or minimize the cross-entropy) is known as **Logistic Regression**. Unlike linear models, logistic regression does not admit a closed-form analytical solution; instead, it is optimized using iterative methods such as **Gradient Descent**. Nevertheless, this model shares a critical property with the linear one: its combination with the cross-entropy loss function produces a **convex cost surface**, ensuring convergence to a global minimum.

The extension to multi-class classification problems is obtained by replacing the Sigmoid function with the *Softmax* function:

$$Soft(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{1.24}$$

where $\mathbf{z} = (z_1, z_2, \ldots, z_K)$ is the vector of *logits* (raw scores) for the K possible classes. In other words, $\sigma(\mathbf{z})_i$ represents the predicted probability that a given input belongs to class i, ensuring that the sum of probabilities across all classes equals one. Consequently, the underlying model must produce an output vector with a dimensionality equal to the number of classes K.

In conclusion, it should be emphasized that this framework is valid regardless of the specific architecture of the underlying model. In this section, a linear structure was proposed as the underlying model because it shares fundamental mathematical properties with the linear regression models previously discussed. However, any other

model architecture, including NNs, can be employed within this same framework to address classification tasks.

1.4 Models regularization

1.4.1 Model capability

In general, in the ML context, the relationship between inputs and outputs is not known *a priori*. Therefore, even though we aim to use a linear model, we must address the problem of determining the specific structure of the model. In the context of linear models, this translates into deciding the degree of the polynomial used as an approximator. If no additional information is provided alongside the dataset, the only way to find the best model is through *trial and error*. However, we must consider an aspect that regards polynomials but also other models: if we fit a set of points with a polynomial of degree d - a model with a certain capability - then a polynomial of degree $d + 1$ - another model with higher capability - can fit the data just as well, or even better. For instance, given two points, we know that we can find a line that passes perfectly through both. At the same time, we can also find a parabola that fits those two points perfectly. In general, a polynomial of any order $d \geq 1$ can perfectly fit two points. This aspect conducts us in facing the problem of the correct **model capability**. As mentioned, this problem regards any ML models. For the sack of simplicity, the author will keep the discussion focused on linear models.
To better discuss the problem raised above, we consider an example from [29].

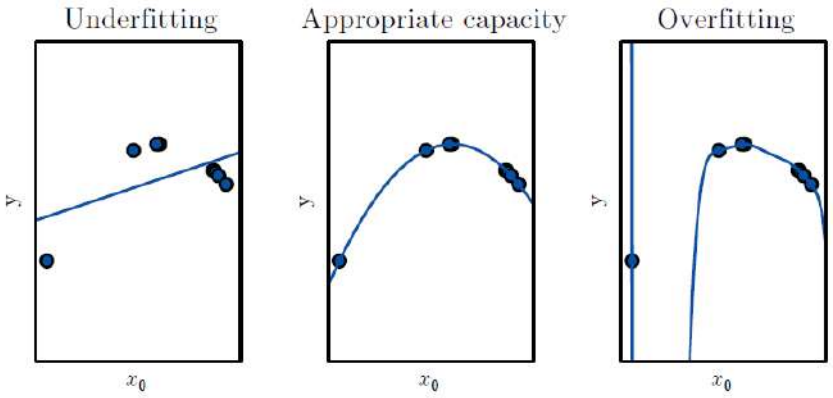


Figure 1.8: Comparison of fitting capabilities on a fixed set of points. The plots show, from left to right, a linear fit, a quadratic fit, and a degree-9 polynomial fit. Figure adapted from [29].

Figure 1.8 illustrates a comparison between different models fitting a fixed set of points. Specifically, the models shown are, from left to right: a linear model, a quadratic model (parabola), and a 9th-degree polynomial.

On the left, it is evident that the line fails to capture the underlying trend of the data. In this scenario, the likelihood of a new data point falling on this line is very low.

This case represents an **underfitting regime**. Conversely, on the right, the 9th-degree polynomial fits the training points perfectly. However, by observing the oscillation of the curve, we can deduce that the probability of a new point lying on this path is also extremely low. This phenomenon is known as the **overfitting regime**. The 2nd-degree polynomial (center) provides a balanced fit; while it may have less precision on the training points than the high-degree polynomial, it is much more likely to accurately predict the location of new, unseen data points.
While this example does not explicitly dictate how to choose the ideal model capacity, we can derive an empirical rule that provides a rationale for model selection: one should choose the *simplest model that fits the data well*. Once a satisfactory representation of the distribution is achieved, increasing the model's capability is generally discouraged. The risk, as depicted on the right side of Figure 1.8, is that the model may predict values that deviate significantly from expected behavior. Furthermore, if the evaluation is performed on points closely related to those used in training - a common occurrence - the model may produce outputs that are entirely meaningless when applied to real-world data. Overfitting remains the **worst enemy** of any ML model.

1.4.2 Hyper-parameters and Validation set

The previous subsection addressed the issue of selecting the appropriate model capacity. In modern ML, and particularly in DL, models are typically over-parameterized. Although this might seem contradictory to the principles discussed above, typical DL applications involve millions of parameters. Consequently, the concept of model capacity becomes crucial when comparing models whose number of parameters differs by orders of magnitude, rather than merely by units or tens. This shift in scale relaxes the strict constraints traditionally imposed on model capacity, such as those encountered in simple linear models, thereby justifying the adoption of over-parameterized regimes in practical applications. At the same time, this necessitates alternative approaches for identifying the model's operating regime, particularly for determining whether the model is overfitting.
Recalling that the primary objective of ML is the ability to make accurate predictions, we are generally interested in the generalization error. The error produced by the model on the training data, while essential for guiding the optimization process, does not provide direct information regarding generalization performance.
Assuming we are training a model with an appropriately chosen capacity, we can track the loss values across successive iterations, also called *epochs*. This allows us to visualize the *training curve*. Typically, this curve exhibits a rapid initial decrease, as the randomly initialized parameters begin to learn the data structure, and subsequently tends toward zero asymptotically. If we evaluate the generalization error at the end of each epoch, we observe a distinct behavior: the initial trend closely follows the training curve, but eventually reaches a minimum before starting to increase again.

This characteristic evolution reveals the operating regime of the model. Initially, the model is in the underfitting regime, having not yet captured a robust representation of the data. When the minimum is reached, the model achieves its optimal configuration and peak generalization performance. Beyond this point, the model begins to overfit the training data, causing the generalization error to rise. In this final state, the model no longer represents the true underlying distribution; instead, it has merely converged on a function that fits the noise or specificities of the training set, failing to reproduce the real-world behavior. A graphical representation is depicted in figure 1.9.

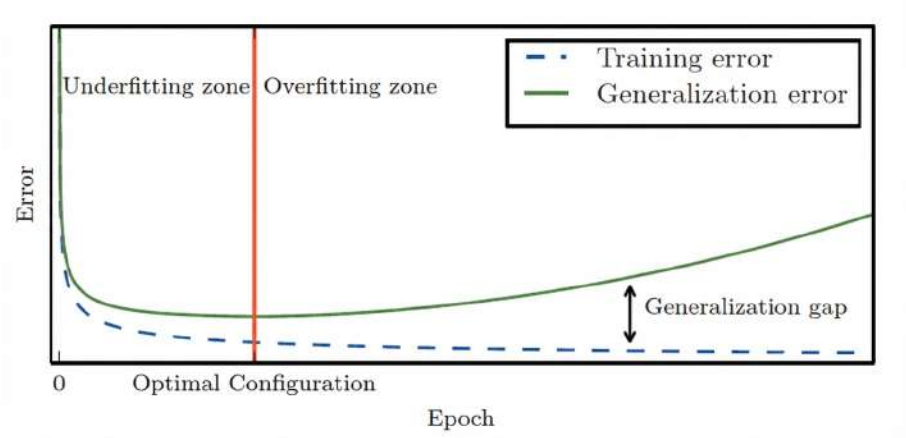


Figure 1.9: Trends of training and generalization errors as a function of training epochs. The plot illustrates the transition from underfitting to overfitting, highlighting the optimal configuration at the minimum of the generalization error curve. The generalization gap represents the difference between the two error types. Figure adapted from [29]

The behavior described above is contingent on the proper configuration of *hyper-parameters*. Unlike model parameters, hyper-parameters are not trainable; they are defined by the user prior to the training process and remain fixed. Examples of hyper-parameters include the learning rate, the model architecture, and the choice of the loss function. In essence, hyper-parameters define the model capacity, and their configuration determines the level of performance the model can achieve. In typical ML applications, models are trained multiple times with varying hyper-parameters to identify an optimal configuration. In other words, hyper-parameters can be viewed as the "control knobs" that the user adjusts to fine-tune the results.
Since hyper-parameters are decisive in the validation process, according to Section 1.2.2, the "Pandora's box" of the test set cannot be opened to determine their best configuration. Doing so would risk the same fundamental issue: overfitting. To mitigate this, it is useful to introduce an additional data subset: the **validation set**. From the original dataset, an initial split into training and test sets is performed. A small portion of the training data is then reserved as a validation set to serve as a proxy for the generalization error. While the validation set is not used to optimize the model's parameters directly, it provides a measure of the error on unseen data. Although this is only an estimate, it allows us to evaluate the model regime without compromising the integrity of the test set.

1.4.3 Regularization techniques

Regularization in ML is fundamental to prevent model overfitting. While many techniques can be applied during training, and users can even develop *ad hoc* solutions, certain methods are widely adopted as best practices.

Regularization based on training curve

Building on the concepts introduced in the previous subsection and using the training curves as a basis, two simple techniques can be introduced. The first involves saving **model checkpoints**, which are essentially "snapshots" of the model's parameter configuration at specific epochs. While one could save a checkpoint at every epoch, this becomes memory-intensive for very large models. Alternatively, a checkpoint can be saved only when a new best performance is achieved on the validation set. By analyzing the training curves at the end of the process, we can identify different model regimes and select the optimal model based on the recorded minimum of the *validation curve*. An extension of this approach is **Early Stopping**. Since training can be highly time-consuming, to avoid wasting computational resources, the training process is terminated when the validation error stops improving. Specifically, training is halted if no new minimum is reached on the validation curve after a user-defined number of epochs, a hyper-parameter commonly referred to as *patience*.

Loss-based Regularization

Other regularization techniques involve modifying the loss function directly. Once a base loss function is selected for training, additional terms can be added to enforce specific behaviors. For example, in a regression task using Mean Squared Error (MSE), we can add a penalty term that discourages the parameters from reaching excessively large values. This approach, known as **L2 regularization** (or Squared Weight Decay), defines the training loss as:

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{base}} + \lambda \|\mathbf{w}\|_2^2 \tag{1.25}$$

where $\mathcal{L}_{\text{train}}$ is the effective loss, $\mathcal{L}_{\text{base}}$ is the task-specific loss (e.g., MSE), λ is a regularization hyper-parameter, and $\|\mathbf{w}\|_2^2$ represents the squared Euclidean norm of the parameter vector. This approach can be justified by considering the nature of real numbers. While allowing parameters to take any real value leads to an *uncountably infinite* search space, the interval $[0, 1]$ is also an *uncountably infinite* set. However, by penalizing large weights, we effectively constrain the model's search to a smaller, more "compact" region of the parameter space. From an optimization perspective, exploring a restricted range of possibilities is often more efficient and less prone to capturing noise in the data than searching across the entire real line.
A variant of L2 regularization is **L1 regularization**, which replaces the squared Euclidean norm in Equation 1.25 with the **L1 norm** (the sum of absolute values).

This method forces less significant weights to become exactly zero, thereby inducing a *sparse representation* of the parameters. Such sparsity is particularly useful for *model interpretability*; since the model effectively performs automatic feature selection, the resulting weight distribution directly reveals which features are most relevant to the task.
In general, the addition of terms to the loss function can significantly aid model training. As previously mentioned, optimization translates into the exploration of a parameter space that is often high-dimensional. If certain information about the task is known *a priori*, it can be incorporated into the loss function as a **constraint** or penalty. This method is highly effective, as its effect is to discourage or entirely exclude regions of the parameter space where these constraints are not respected. This principle forms the basis of **Physics-Informed Neural Networks**, which will be discussed in the next chapter.

Adaptive algorithms

One of the most important hyper-parameters is the learning rate, ϵ. This parameter defines the "step size" for the optimization rule (Equation 1.8); consequently, it must be carefully tuned. In general, the loss surface is highly complex, and convergence to a global minimum is not guaranteed. Furthermore, reaching even a "good" local minimum is often uncertain, a condition largely influenced by the choice of learning rate.

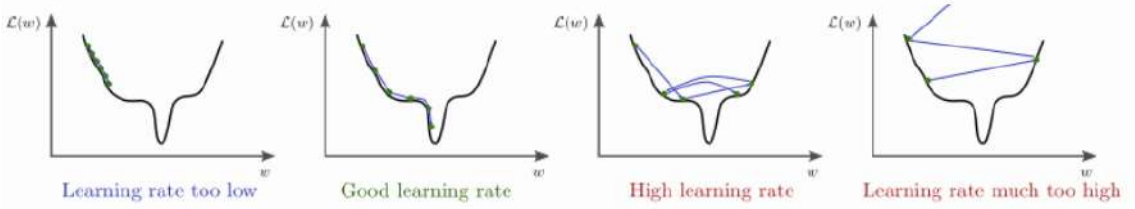


Figure 1.10: Comparison of training behaviors for different learning rates. An excessively small learning rate leads to slow convergence, whereas a large learning rate may cause the optimizer to overshoot the global minimum.

Figure 1.10 illustrates the various behaviors induced by different learning rate values. An excessively small learning rate leads to slow convergence; in the worst-case scenario, the model may get **stuck** in a local minimum far from the global optimum. Conversely, a learning rate that is too large allows the optimizer to escape local minima but may cause it to **overshoot** the most favorable regions of the parameter space. There is no universal rule for setting the optimal learning rate and, due to the **stochastic nature** of the optimization process, even identical settings can lead to different outcomes.
To address these issues, **adaptive optimization algorithms** and **learning rate schedulers** are often employed. These methods adjust the learning rate dynamically during training. The underlying intuition is that at the beginning of the process,

parameters require large updates to explore the space; as the optimization reaches a flatter region (or plateau), a finer search is required to locate a good minimum. Consequently, one can implement a **learning rate decay** as a function of the epoch number. Among the most popular methods is the *polynomial decay* (poly-decay) function.
Other advanced methods maintain a history of the gradients' values and directions from previous epochs to identify and follow trends in the loss landscape. Notable examples include *Momentum*, *AdaGrad*, *RMSprop*, and *Adam*. For further information regarding these algorithms, the reader is referred to Section 8.5 in [29]. Specifically, *Adam* is widely considered the state-of-the-art optimizer as it combines the advantages of both momentum and adaptive scales.

Data augmentation

Definition 1 states that a model's performance depends heavily on the amount of experience, provided as data—used during training. In many ML scenarios, datasets are limited, making it difficult to improve results simply by following this definition. A common strategy to overcome this limitation is the application of data transformations, a technique known as **Data Augmentation**.
While data augmentation can be applied to various tasks, its utility is most evident in Computer Vision. If the task is to detect cats in images, for instance, a rotated cat is still a cat; similarly, a translated or mirrored cat remains a cat. The core idea is to expand the training set by applying transformations that preserve the essential information the model is intended to learn.
It should be noted that data augmentation does not truly increase the number of independent samples -applying a transformation is not equivalent to collecting entirely new data, which might contain novel features. However, it encourages the model to focus on more robust features. Furthermore, this method makes the model *invariant* to the specific transformations applied, which significantly enhances its generalization capabilities on unseen data.
Common transformations used for data augmentation include: translations, rotations, mirroring, flipping, rescaling, and the injection of Gaussian noise.

CHAPTER 2

DEEP LEARNING

DL has propelled AI to become perhaps the most significant innovation of the current century to date. This statement naturally invites two fundamental questions: *"how?"* and *"why now?"*.

The *Perceptron*, the artificial counterpart to the biological neuron, was modeled as early as the *1950s*. The backpropagation algorithm was proposed around the same period and famously applied in 1972. Furthermore, the first version of the *Universal Approximation Theorem* - the mathematical cornerstone of neural network functionality - was proven in 1989, and Physics-Informed Neural Network for solving Partial Derivative Equations were introduced in the *1990s*. Despite these early breakthroughs, DL only emerged as the dominant paradigm in AI after the first decade of the 21st century. Why did this transition take so long? What finally made the "dream" a reality?

There were two primary limitations: the hardware required to train these models and the availability of data for the training process. For a long time, DL remained a theoretical pursuit; the conceptual framework was ready, but computers lacked the necessary performance. Starting in the *1990s*, the landscape began to shift: the Internet as we know it took shape, and computational power increased. As data collection grew, storage capabilities began to follow the exponential trend observed today. However, a crucial piece of the puzzle was still missing.

If one were to assign a specific date and location to the beginning of the modern AI revolution, it would arguably be between June 16th and 21st, 2013, in Atlanta, USA. On the occasion of the 30th International Conference on Machine Learning, Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Andrew Ng presented a landmark paper [30] stating:

> *"Scaling up deep learning algorithms has been shown to lead to increased performance in benchmark tasks and to enable discovery of complex high-level features. Recent efforts to train extremely large networks (with over 1 billion parameters) have relied on cloud-like computing infrastructure*

and thousands of CPU cores. In this paper, we present technical details and results from our own system based on Commodity Off-The-Shelf High Performance Computing (COTS HPC) technology: a cluster of GPU servers with Infiniband interconnects and MPI. Our system is able to train 1 billion parameter networks on just 3 machines in a couple of days, and we show that it can scale to networks with over 11 billion parameters using just 16 machines." [30]

This work catalyzed the shift in DL from Central Processing Unit (CPU) to GPU architectures. It demonstrated that the massive scale achieved by the *2012 Google Brain project* [31], which famously required 16,000 CPU cores, could be replicated, and even surpassed, by a much smaller cluster of GPU-accelerated off-the-shelf servers. Around that same period, **NVIDIA Corporation** began its ascent toward becoming the most valuable companies in the world. This meteoric growth was driven by the realization that their GPUs, originally designed for the video game industry, were uniquely suited for the massive parallel computations required by Deep Learning.

2.1 Fundamentals of Deep Learning

A point that deserves a discussion is what fundamentally distinguishes DL from general ML. While the primary difference lies in the use of NNs as function approximators, this is only a part of the story. In general ML, regardless of the task, models typically operate on well-established, "hand-crafted" features. The inputs are usually data that have undergone specific pre-processing stages designed to extract *a priori* defined mathematical descriptors. As an example, without loss of generality, in ML-based image classification, models are often fed with features extracted from ROIs that describe geometric properties, gray-level distributions, and textures. When applied to clinical imaging, this framework is known as **Radiomics** [32]. A similar approach is common across all **omics** disciplines (Genomics, Proteomics, etc.). Today, one of the foremost challenges in AI is **Multi-omics integration**, which aims to train models by effectively combining features derived from different omic layers.
In DL, the paradigm is that features must emerge from raw data. This aspect represents a *"double-edged sword"*: on the one hand, models are free to learn knowledge unknown to humans; on the other hand, this affects model *explainability* [33]. To address this issue, additional algorithms, such as **Grad-CAM** (Gradient-weighted Class Activation Mapping) for image classification, have been developed. These techniques generate heatmaps highlighting the specific anatomical regions that the model prioritized to reach its decision. In clinical practice, this allows for a direct comparison between the model's focus and the clinician's expertise, effectively "grounding" the emergent features of DL back into human-interpretable anatomy and ensuring that the prediction

is based on relevant biological markers rather than computational noise. However, model explainability is still an open issue and considerable work remains to be done [34].

The following subsections will be dedicated to the introduction of the perceptron, multi-layer perceptron models, the backpropagation algorithm, and an overview about the universal approximation theorem.

2.1.1 Perceptrons and Multi-layer Perceptrons network

The fundamental unit of Neural Networks (NNs) is the **perceptron**, which is inspired by the functioning of biological neurons. A biological neuron receives multiple inputs via its dendrites and emits a single electro-chemical impulse through the axon. It is important to underline that this process is non-linear: the neuron remains inactive until the input signals reach a specific threshold value. Only once this threshold is met does the neuron "fire," sending a relatively strong impulse. This mechanism is known as the **neuron activation process**.

To translate this biological mechanism into a computational model, the perceptron abstracts the input-output process through a weighted sum and an activation function. In this mathematical framework, the signals received by the dendrites are represented by an input vector $x = [x_1, x_2, \ldots, x_n]$, where each input is assigned a specific weight (w_i). These weights reflect the "strength" or importance of each connection, mimicking the synaptic plasticity found in biological systems.

The core of the perceptron operation consists of calculating the linear combination of these inputs, adding a bias (b) to shift the decision threshold, and finally passing the result through a non-linear **activation function** (f). Mathematically, the output y is expressed as:

$$y = f\left(\sum_{i=1}^{n} w_i x_i + b\right) \tag{2.1}$$

This function f acts as the mathematical equivalent of the biological "threshold," determining whether the artificial neuron should "fire" or remain silent based on the accumulated evidence from its inputs.

The specific expression of the activation function f is crucial, as it determines the perceptrons' ability to capture complex patterns. It is interesting to notice that Equation 2.1 becomes exactly equal to Equation 1.23 when a sigmoid is used as an activation function. The most common activation functions include:

- **Heaviside Step Function:** This is the simplest form of activation, directly mimicking the biological threshold. The output is 1 if the input is positive and 0 otherwise. While intuitive, its main drawback is that it is not differentiable at zero, making it unsuitable for modern gradient-based learning algorithms.

- **Sigmoid Function:** To overcome the limitations of the step function, the Sigmoid

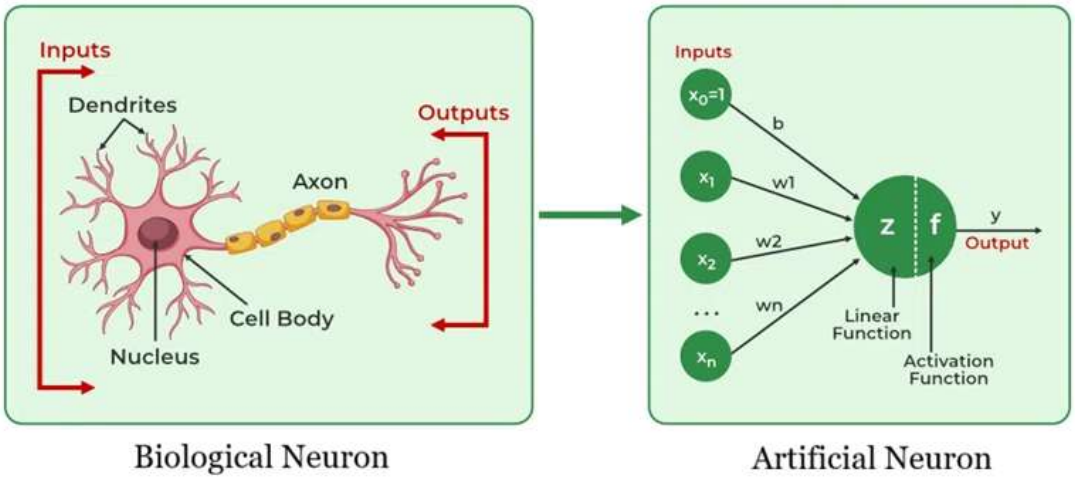


Figure 2.1: Comparison between biological and artificial neuron.

(or logistic) function was introduced. It maps any real-valued input into a range between 0 and 1, providing a smooth, differentiable transition. However, it suffers from the *vanishing gradient problem* for very high or very low input values.

- **Hyperbolic Tangent (tanh):** Similar to the Sigmoid, the tanh function is S-shaped, but it maps inputs into a range between -1 and 1. Being zero-centered, it often allows for faster convergence during the training process compared to the Sigmoid. This activation function is the most widely used in PINN applications for solving Partial Derivative Equations.

- **Rectified Linear Unit (ReLU):** Currently the standard in Deep Learning, ReLU is defined as $f(x) = \max(0, x)$. It effectively addresses the vanishing gradient issue and reduces computational cost, allowing for the training of much deeper architectures.

- **Leaky ReLU:** As an extension of the ReLU, the Leaky ReLU introduces a small slope for negative values (e.g., $f(x) = \alpha x$ for $x < 0$, where α is a small constant). This modification is designed to prevent the "dying ReLU" problem, ensuring that neurons continue to learn even when their input is negative.

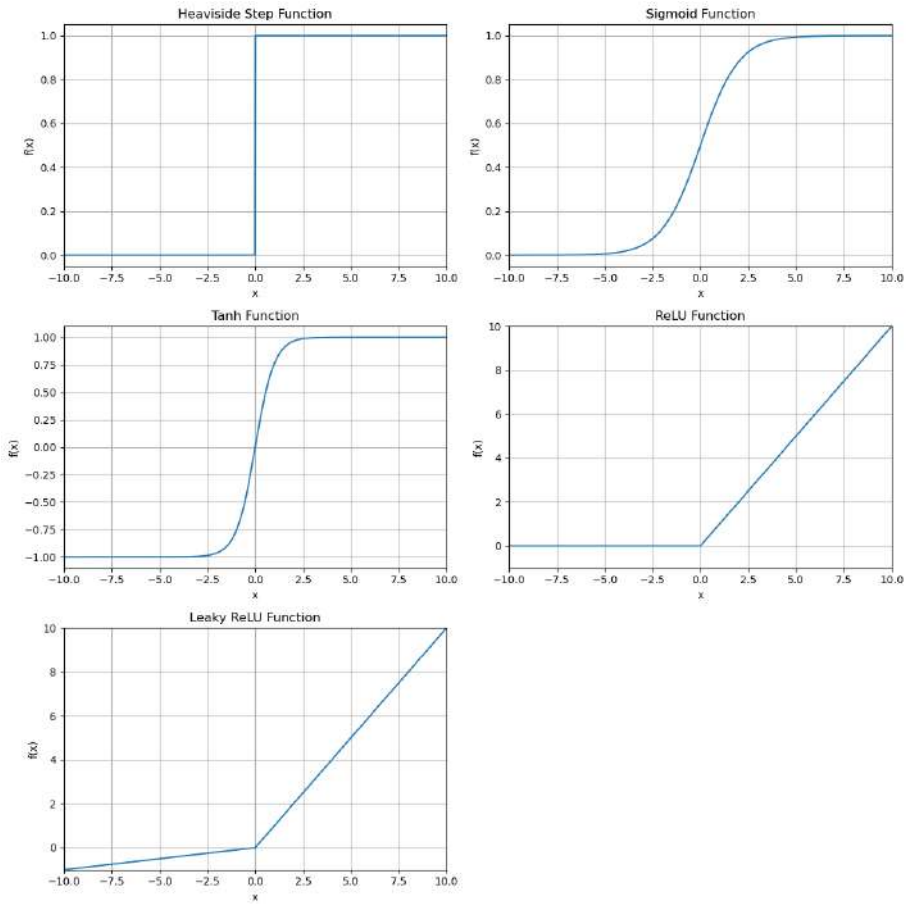


Figure 2.2: Plots of different activation functions. Specifically, the Heaviside step, Sigmoid, Hyperbolic tangent (tanh), ReLU, and Leaky ReLU are depicted.

Drawing inspiration from the biological brain, the natural extension of a perceptron is its integration with others to form a connected structure: what we define as a *Neural Network* (NN). The most straightforward way to construct such a network is by feeding multiple perceptrons the same input vector and aggregating their outputs to achieve the desired mathematical output structure. This configuration is commonly referred to as a *Single-Hidden layer Network*. Figure 2.3 illustrates the schematic of this network architecture. Although a simpler model could be obtained using only a single layer, the number of units would be constrained by the dimensionality of the output we aim to predict. Such a network does not meet, even in principle, the requirements to be a universal approximator.

In DL, units at the same level (or *depth*) of interaction are referred to as a *layer*. In Figure 2.3, we can distinguish three types of layers: the input layer, which contains the inputs fed to the network; the hidden layer, which contains the network units - "hidden" indicates that the user does not have direct control over its operations; and the output layer, which represents the model's prediction. The output of the hidden layer, in accordance with the previous definitions, is a vector of the form $\boldsymbol{L} = [L_1, L_2, \ldots, L_M]$, where M is the number of units in the layer. By applying the equation that defines a

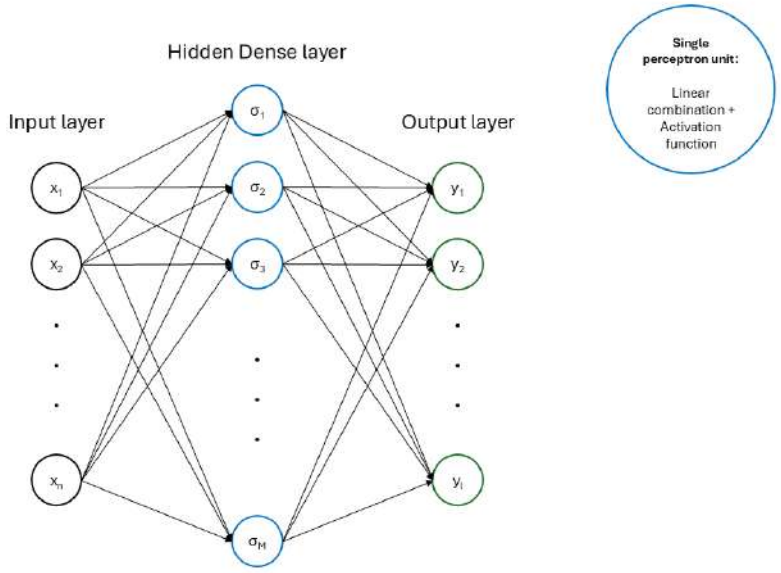


Figure 2.3: Schematic of a network with a single hidden layer. The network input is a vector with n components, which constitutes the input layer. In the middle, there are M perceptrons, also referred to as *units*. Each unit computes a linear combination of the inputs it receives and then applies an activation function to produce an output. This array of units constitutes a **Dense layer**. In DL, layers that are neither inputs nor outputs are termed *hidden*, since the operations they perform are a function of the training process; in other words, the user does not have direct control over the specific parameters of these layers. In this example, the final output is a vector of l components, constituting the output layer.

perceptron (Equation 2.1), each component i of the layer's output can be expressed as:

$$L_i = f\left(\sum_{j=1}^{n} w_{i,j}x_j + b_i\right) \tag{2.2}$$

where n is the number of inputs. It is straightforward to deduce that the total number of trainable parameters in a layer is given by:

$$N_{\text{free_par}} = N_{\text{units}} \times (N_{\text{inputs}} + 1) \tag{2.3}$$

where the addition of 1 accounts for the bias of each unit. In the network shown in Figure 2.3, the number of parameters for the hidden layer is $M \times (n + 1)$. Similarly, for the output layer, which has l units receiving M inputs from the hidden layer, the number of free parameters is $l \times (M + 1)$.
Applying Equation 2.2 to the output layer to obtain the components of the predicted vector $\boldsymbol{y}_{\text{pred}} = [y_{\text{pred},1}, y_{\text{pred},2}, \ldots, y_{\text{pred},l}]$, we derive:

$$y_{\text{pred},i} = f\left(\sum_{k=1}^{M} w'_{i,k}\ L_k + b'_i\right) = f\left(\sum_{k=1}^{M} w'_{i,k}\ f\left(\sum_{j=1}^{n} w_{k,j}x_j + b_k\right) + b'_i\right) \tag{2.4}$$

where $(w'_{i,k}, b'_i)$ represent the parameters of the output layer. While Equation 2.4 may

appear complex, its notation can be significantly simplified by employing matrix algebra. As done with linear models, the summation $y_i = \sum_{j=0}^{n} w_{i,j} x_j$ can be represented as a matrix product $\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}$, where $\boldsymbol{y}$ is the output vector, $\boldsymbol{W}$ is the weight matrix, and $\boldsymbol{x}$ is the input vector. The requirement to use the transpose depends on the convention adopted for the weight matrix: in this notation, the i-th column of $\boldsymbol{W}$ contains the weights associated with the i-th unit. To maintain mathematical consistency with matrix-vector multiplication, the transposed matrix $\boldsymbol{W}^T$ must be used. Using this notation, Equation 2.4 can be rewritten as:

$$\boldsymbol{y}_{\text{pred}} = f\left(\boldsymbol{W}^{(2)T} \boldsymbol{L}^{(1)} + \boldsymbol{b}^{(2)}\right) = f\left(\boldsymbol{W}^{(2)T} f\left(\boldsymbol{W}^{(1)T} \boldsymbol{x} + \boldsymbol{b}^{(1)}\right) + \boldsymbol{b}^{(2)}\right) \tag{2.5}$$

where the superscripts (1) and (2) refer to the first and second layers, respectively. Equations 2.4 and 2.5 clearly highlight the most significant difference between NNs and linear models. If f were the identity function, Equation 2.5 would reduce to a series of simple matrix products. Regardless of the number of layers, since matrix multiplication is a linear operation, the composition of multiple linear transformations is mathematically equivalent to a single linear transformation. In other words, even with hundreds of layers, the entire architecture would collapse into a single operation, as if only one layer existed. In NNs, non-linear activation functions break this linearity, allowing for the effective combination of different layers to learn complex representations. Furthermore, a non-linear mapping cannot be represented by a linear one. This is why NNs differ fundamentally from linear models: the former represents a class of non-linear approximators, while the latter is restricted to linear mappings.
Although the architecture presented here is sufficient to approximate any *Borel measurable function* - as demonstrated by the Universal Approximation Theorem [35, 36], which will be introduced in the following subsections - empirical studies have shown that better performance and faster convergence can be achieved by increasing the number of hidden layers. The rationale behind this approach is that each layer allows the model to learn new features, whose complexity increases as the network becomes *deeper*. This expansion in depth is precisely what gives *Deep Learning* its name. A way to interpret these empirical results is provided by Goodfellow et al. [29]:

> Choosing a deep model encodes a very general belief that the function we want to learn should involve the composition of several simpler functions. This can be interpreted from a representation learning point of view as saying that we believe the learning problem consists of discovering a set of underlying factors of variation that can in turn be described in terms of other, simpler underlying factors of variation. Alternately, we can interpret the use of a deep architecture as expressing a belief that the function we want to learn is a computer program consisting of multiple steps, where each step makes use of the previous step's output.

An NN with multiple hidden layers is commonly referred to as a *Multi-Layer Perceptron* (MLP). Since the outputs of the layers in these architectures are never fed back into previous stages (unlike in *Recurrent Neural Networks*), they belong to the broader category of *Feed-Forward Neural Networks*, which also includes other architectures such as *Convolutional Neural Networks*.

2.1.2 Backpropagation algorithm

The foundation for optimization algorithms based on gradient descent is the update rule presented in Equation 1.8. As previously discussed, MLP networks often employ a large number of parameters, which makes the optimization process computationally expensive. A fundamental technique to address this is the *chain rule* for calculating derivatives of *composite functions*. Before embarking on a discussion of how backpropagation is derived, the author intends to present a representation for mapping computational operations onto a graph, which simplifies the analysis. This representation is known as a **computational graph**. Each node in the graph represents a variable, which may be a scalar, a vector, or more generally, a tensor. Conversely, each edge represents a mathematical operation. The graph for the function $z = xy$ is depicted in Figure 2.4a, while Figure 2.4b illustrates the computational graph for the single hidden layer network presented in the previous section.
With this notation, we are now ready to introduce the *chain rule* for the gradient calculation of composite functions. Consider simple functions of scalar variables, such as $z = g(y)$ and $y = f(x)$. We can define the composite function that takes the value x and produces the number z ($x \longmapsto z$) as:

$$z = g(y) = g\left(f(x)\right). \tag{2.6}$$

To compute the derivative with respect to x, we apply the rule for composite functions:

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}. \tag{2.7}$$

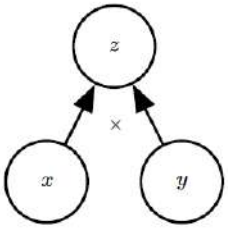
This mathematical rule extends to functions of higher-dimensional variables. Let us transform Equation 2.6 into the vector form $z = g(f(\boldsymbol{x}))$, where z remains a scalar; this can be represented by the computational graph in Figure 2.5.

In this case, the partial derivative of z with respect to the components of the vector $\boldsymbol{x}$ is given by:

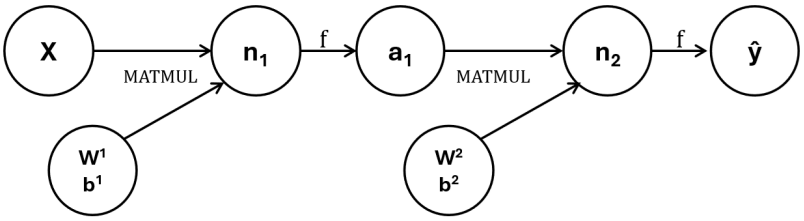
$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i}. \tag{2.8}$$

As previously discussed, this expression can be converted into vector form as:

$$\nabla_{\boldsymbol{x}} z = \left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right)^T \nabla_{\boldsymbol{y}} z \tag{2.9}$$



(a)



(b)

Figure 2.4: Computational graphs for: (a) the function $z = xy$; (b) the neural network presented in Figure 2.3. In (b), the symbol $\hat{\boldsymbol{y}}$ stands for the prediction for $\boldsymbol{y}$ and f represents a generic activation function. Nodes represent variables, while operations are indicated along the edges.

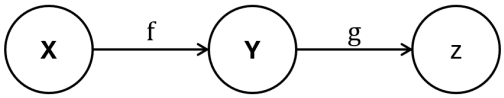


Figure 2.5: Computational graph for $z = g(\boldsymbol{y}) = g(f(\boldsymbol{x}))$.

where $\left(\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}\right)^T$ represents the transposed Jacobian matrix and $\nabla_{\boldsymbol{y}} z$ is the gradient of z with respect to the vector $\boldsymbol{y}$.

A typical MLP extends the computational graph depicted in Figure 2.5 to the one shown in Figure 2.6, where the computation of a generic loss $J(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is also included. This computational graph shows that each layer l receives the activation from the previous layer, $\boldsymbol{a}_{l-1}$, as input. It then applies the linear transformation (pre-activation) $\boldsymbol{n}_l = \boldsymbol{W}^{(l)T} \boldsymbol{a}_{l-1} + \boldsymbol{b}^{(l)}$ and, finally, applies the activation function to produce the l-th activation: $\boldsymbol{a}_l = f(\boldsymbol{n}_l)$.

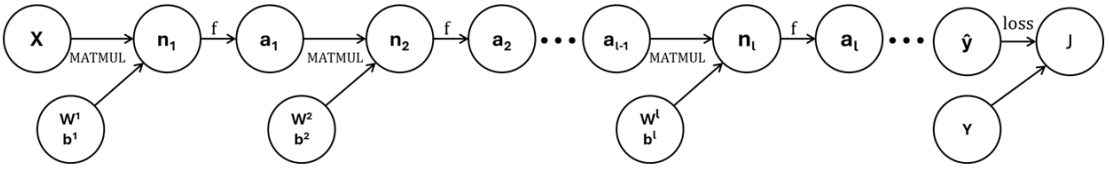


Figure 2.6: Computational graph for the computation of a generic loss $J(\hat{\boldsymbol{y}}, \boldsymbol{y})$ for a generic MLP. $\boldsymbol{y}$ represents the true label vector.

Now using Equation 2.9 we can easily deduce that

$$\nabla_{\boldsymbol{n}_l} J = \left(\frac{\partial \boldsymbol{n}_{l+1}}{\partial \boldsymbol{n}_l} \right)^T \nabla_{\boldsymbol{n}_{l+1}} J \tag{2.10}$$

Since our aim is to compute gradients for each network weight we can deduce that:

$$\begin{cases} \nabla_{\boldsymbol{W}^{(l)}} J &= \left(\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{W}^{(l)}} \right)^T \nabla_{\boldsymbol{n}_l} J = \left(\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{W}^{(l)}} \right)^T \left(\frac{\partial \boldsymbol{n}_{l+1}}{\partial \boldsymbol{n}_l} \right)^T \nabla_{\boldsymbol{n}_{l+1}} J \\ \nabla_{\boldsymbol{b}^{(l)}} J &= \left(\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{b}^{(l)}} \right)^T \nabla_{\boldsymbol{n}_l} J = \left(\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{b}^{(l)}} \right)^T \left(\frac{\partial \boldsymbol{n}_{l+1}}{\partial \boldsymbol{n}_l} \right)^T \nabla_{\boldsymbol{n}_{l+1}} J \end{cases} \tag{2.11}$$

The backpropagation algorithm consists of *forward-propagating* the inputs through the network, mapping all operations according to the computational graph, and then *back-propagating* the error gradient. This process allows for the layer-by-layer calculation of the weight gradients by leveraging the chain rule.
Figure 2.7 provides a schematic overview of how this algorithm operates: x_i represents the input components, while h_i denotes the activations of the hidden units for a single-layer network.

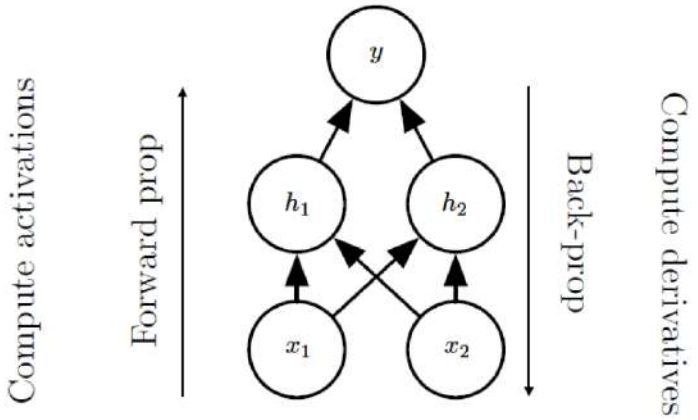


Figure 2.7: Schematic representation of the backpropagation algorithm: the input, defined by its components x_i, is first forward-propagated through the network; subsequently, an output is produced and used to compute the loss; finally, the gradients with respect to each unit h_i are back-propagated.

The algorithm can be summarized by two steps: forward (Algorithm 1) and backward (Algorithm 2) propagation. The computation of the Jacobian $\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{n}_{l-1}}$ yields:

$$\frac{\partial \boldsymbol{n}_l}{\partial \boldsymbol{n}_{l-1}} = \frac{\partial}{\partial \boldsymbol{n}_{l-1}} \left(\boldsymbol{W}^{(l)T} f(\boldsymbol{n}_{l-1}) + \boldsymbol{b}^{(l)} \right) = \boldsymbol{W}^{(l)T} \odot f'(\boldsymbol{n}_{l-1}) \tag{2.12}$$

where the operator $\odot$ indicates an element-wise multiplication and f' is the derivative of the activation function. For instance, for the ReLU function (by convention):

$$f'(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{2.13}$$

Algorithm 1: Forward Propagation

Data: Network depth L; weight matrices $\boldsymbol{W}^{(k)}$ and bias vectors $\boldsymbol{b}^{(k)}$ for $k \in \{1, \ldots, L\}$; input $\boldsymbol{x}$; target $\boldsymbol{y}$

Result: Loss $J(\hat{\boldsymbol{y}}, \boldsymbol{y})$ and maps of the activations $\boldsymbol{a}_k$ and pre-activations $\boldsymbol{n}_k$

Initialize activation: $\boldsymbol{a}_0 \longleftarrow \boldsymbol{x}$
for $k = 1$ **to** L **do**
    $\boldsymbol{n}_k \longleftarrow \boldsymbol{W}^{(k)T} \boldsymbol{a}_{k-1} + \boldsymbol{b}^{(k)}$
    $\boldsymbol{a}_k \longleftarrow f(\boldsymbol{n}_k)$
end
Assign prediction: $\hat{\boldsymbol{y}} \longleftarrow \boldsymbol{a}_L$
Compute loss: $J \longleftarrow J(\hat{\boldsymbol{y}}, \boldsymbol{y})$

Algorithm 2: Backward Propagation

Data: Activations $\boldsymbol{a}_k$, pre-activations $\boldsymbol{n}_k$, weights $\boldsymbol{W}^{(k)}$, input $\boldsymbol{x}$, target $\boldsymbol{y}$

Result: Gradients $\nabla_{\boldsymbol{b}^{(k)}} J$ and $\nabla_{\boldsymbol{W}^{(k)}} J$ for all layers

Perform Forward Step to obtain all $\boldsymbol{a}_k$ and $\boldsymbol{n}_k$

Compute output gradient: $\boldsymbol{g} \longleftarrow \nabla_{\hat{\boldsymbol{y}}} J(\hat{\boldsymbol{y}}, \boldsymbol{y})$

Compute gradient with respect to last pre-activation: $\boldsymbol{g} \longleftarrow \boldsymbol{g} \odot f'(\boldsymbol{n}_L)$

for $k = L$ **to** 1 **do**

  $\nabla_{\boldsymbol{b}^{(k)}} J \longleftarrow \boldsymbol{g}$

  $\nabla_{\boldsymbol{W}^{(k)}} J \longleftarrow \boldsymbol{a}_{k-1}^T \boldsymbol{g}$

  $\boldsymbol{g} \longleftarrow (\boldsymbol{W}^{(k)} \boldsymbol{g}) \odot f'(\boldsymbol{n}_{k-1})$ `// Update gradient for previous layer`

  `// Corresponds to:` $\frac{\partial \boldsymbol{n}_k}{\partial \boldsymbol{n}_{k-1}}$

end

The concept of backpropagation was initially proposed by Frank Rosenblatt (1962); however, its practical implementation was hindered by his focus on discrete variables. In 1982, Paul Werbos successfully applied the method to MLPs in his PhD studies. The modern formulation, which serves as the primary inspiration for this thesis, was derived independently and popularized in 1986 by David E. Rumelhart [37].

Neural networks rely on symbolic representations, where algebraic expressions and computational graphs define operations on variables rather than specific values. When performing back-propagation, deep learning frameworks generally follow one of two paradigms to compute gradients [29]:

- Symbol-to-Number Differentiation: This approach computes the numerical value of the gradient directly from the graph and a given set of input values. Frameworks such as Torch [38] and Caffe [39] utilize this method, focusing on providing immediate numerical outputs for optimization.
- Symbol-to-Symbol Differentiation: Instead of returning a value, this method extends the existing computational graph by adding nodes that represent the symbolic derivatives of the operations. This allows for a more flexible, graph-based description of the gradient, a technique adopted by libraries like Theano [40] and TensorFlow [41].

To conclude this subsection, it is important to underline why backpropagation represents a significant advantage in gradient computation. Algorithm 2 visits each edge of the computational graph only once, computing gradients solely for simple operations. In other words, the back-propagation algorithm is designed to reduce the number of common subexpressions without regard to memory overhead. Consequently, backpropagation avoids the exponential explosion of repeated subexpressions.

2.1.3 Universal Approximation Theorem

Section 1.3.1 showed how linear models can be used as approximators. This approach has several advantages, such as the direct estimation of parameters which leads to a global minimum for the MSE. However, the range of functions we can approximate with these models is limited and consists mostly of polynomial functions. Unfortunately, in ML we are often interested in finding approximations for non-linear problems. This requires extending the ML approach to a broader family of approximators: the *FeedForward Neural Network (FFNN)*. FFNN is a family of NN larger than MLP and includes Convolutional Neural Networks as well.

As mentioned previously, early studies on NN began in the *50s*; however, significant time passed before theoretical and experimental results were achieved. The first important theoretical results were published by Hornik in the first months of 1989 [35], and by Cybenko in December 1989 [36]. Both works introduced the well-known **universal approximation theorem**. Although Cybenko's work was published later, he had already presented it at conferences; furthermore, his version is more restricted than Hornik's. The proofs of these theorems are beyond the scope of this thesis; however, the principal outcomes will be presented here.

The Universal Approximation Theorem [36] (Cybenko's version) states that a MLP with at least one hidden layer, an **unspecified** number of units, a sigmoid activation function, and a linear output layer can approximate any *Borel function* with arbitrary precision. The definition of Borel functions will not be provided; however, it is sufficient to know that any continuous function on a closed and bounded subset of $\mathbb{R}^n$ belongs to this category. Hornik's version [35] extends this result to any "squashing" activation function, i.e., functions f that satisfy the following conditions: $\lim_{x\to\infty} f(x) = 1$; $\lim_{x\to-\infty} f(x) = 0$; f is non-decreasing; and the image of the function is limited to $[0, 1]$. Later, the property of being a universal approximator for MLPs was extended to a wide range of activation functions, including ReLU [42].

It is worth noting that although in their papers Hornik and Cybenko refer to FFNN, they focused on MLP. In particular, they proved the theorem by assuming that the NN could arbitrarily increase in "width" to reach the desired precision. Ever since then, it has been experimentally clear that there are more advantages in deep networks than in wide ones, but arbitrary "depth" was theoretically taken into consideration only later. Among the most important results, Zhou's work [43] represents a milestone, since it states that a deep convolutional NN can be used to approximate any continuous function to an arbitrary accuracy when the depth of the neural network is large enough. With this result, the whole class of FFNN has been demonstrated to be universal approximators. However, this theoretical framework establishes just an ***existence*** condition. It does not provide any information about the architecture a NN must have to be able to approximate a specific function.

The Universal Approximation Theorem is a fundamental theoretical result and

provides at least a rationale for using NNs in ML problems. However, several considerations must be made. The first concerns optimization. The theorem states that given a (Borel) function, there exists a MLP with a single hidden layer, arbitrarily wide, able to approximate it. However, even if we find the optimal number of units, nothing guarantees that the optimization process will find the correct parameters configuration. In other words, we may have a complex loss surface that traps the model in a local minimum, or even worse, the model may fall into an overfitting regime.
Secondly, achieving high precision may require a very large number of units. In fact, considering a binary classification task with a single hidden layer, it is known that the number of free parameters required to correctly classify n examples can be as high as 2^n. This introduces a common problem in DL: the number of parameters needed to fit a function can become prohibitively large.

Another issue concerns the definition of Borel functions. The Universal Approximation Theorem is often misunderstood or overestimated. In the common view, held especially by students new to the field, this theorem is perceived as implying the ability to approximate any human skill or task. This is, of course, not the case. First, the theorem has been proven in a strictly mathematical sense: it makes use of functions as defined in mathematical terms. A human's ability to classify clinical images, for instance, is not a Borel function.
The mathematical definition of a function states that for any given value x within the domain, there exists one, and only one, corresponding value $f(x)$. In clinical practice, however, various factors such as stress, working hours, and image quality, can influence a doctor's performance. Consequently, the same image may be classified differently by the same clinician at different times, leading to *intra-observer variability*. Similarly, disagreement between different clinicians results in *inter-observer variability*. Both measures suggest that clinical classifications often violate the fundamental definition of a mathematical function. This leads to a critical question for AI researchers: *'Does the function we aim to approximate with NN actually exist?'*
Naturally, Maximum Likelihood allows us to reframe the problem as finding the probability of an input being assigned a specific label. Other techniques involve injecting random noise during training to simulate these behaviors, although this is merely a simulation of a far more complex reality. Nevertheless, the question raised remains relevant and should be considered when designing an AI project.

To conclude this subsection, the "take-home" message is that FFNNs have been proven to be universal approximators of mathematical functions. This justifies their use in ML, but it must also be clear that NNs are not artificial brains.

2.2 Computer vision

One of the most famous image databases is the well-known *MNIST* (Modified National Institute of Standards and Technology database) [44], a dataset of handwritten digits. This database was created from the NIST dataset [45] and is now used as a benchmark for AI algorithm development.

The MNIST database contains 60,000 labeled images, whose sizes have been normalized to 28×28 pixels. To classify these digits using an MLP, the images must first be linearized, a process also known as *flattening*. This operation maps matrices from the space $\mathbb{R}^{m \times n}$ into vectors in the space $\mathbb{R}^{mn}$ [1], effectively making each pixel an input node for the network (Figure 2.8a). In the case of the MNIST dataset, the 28×28 images are converted into flat vectors of size 784 (Figure 2.8b).

Figure 2.8: Flattening process: (a) generic flattening of a matrix; (b) flattening of an image from the MNIST dataset.

At this point, it is necessary to design the network architecture. We can start with a simple structure consisting of four hidden layers and an output layer, denoted by the vector $[100, 200, 500, 200, 10]$[2]. We employ the ReLU activation function for the hidden layers and a Softmax activation for the output layer, whose size must match the number of classes (in this case 10 digits).

Since this is a ML task, we partition the dataset into training, validation, and test sets. This division allows us to train the MLP on the training set using, for example, Cross-Entropy as the loss function. We can then determine the optimal configuration by monitoring performance on the validation set. Finally, we evaluate the model on the test set to measure its generalization capability, for instance, in terms of ACC.

A typical MLP achieves a test ACC around 95% [27]. Performance can be improved by introducing *dropout layers* between the hidden layers. Dropout is a regularization technique designed to prevent neuron co-adaptation, where different units in the same layer learn redundant features. It consists of randomly "dropping out" (deactivating) a fraction of the connections or units during each training iteration. A typical dropout

[1] In the space $\mathbb{R}^{m \times n}$, each element is a matrix with m rows and n columns; conversely, elements of $\mathbb{R}^{mn}$ are vectors with $l = mn$ components.

[2] This is a typical notation for an MLP structure: the number of elements specifies the total number of layers (including the output layer), while each entry indicates the number of units per layer.

rate is approximately 10%.

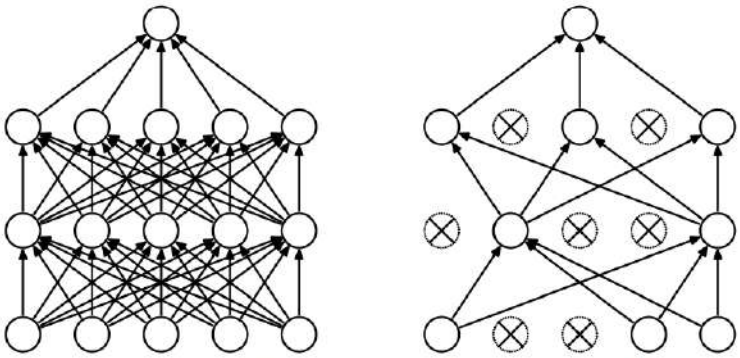


Figure 2.9: Illustration of dropout. The left panel shows a fully connected MLP. The right panel shows the network with dropout applied: randomly selected neurons are deactivated during training to encourage the learning of independent features and prevent overfitting. Image adapted from [46].

By adding dropout layers, the ACC on the test set generally increases; however, for this specific architecture on MNIST, it rarely reaches the 96-97% threshold. This naturally raises the question: what causes this limitation?

First, we should examine the number of parameters. By iteratively applying Equation 2.3, it is easy to derive that the configuration proposed above involves $301,410$ free parameters. This raises two important questions. First, could this number be excessive? Second, since we have worked with very small images (28×28 pixels), what happens if we attempt to process high-definition or multi-channel (3D) images? There is evidently an explosion in the number of parameters, leading to significant scalability issues.

Secondly, we must reflect on how we feed the images into the network. As mentioned, we apply a flattening operation. With this operation, every pixel is treated as having a potential correlation with every other pixel, regardless of their spatial proximity. Does this reflect the way humans read a digit? Generally, no. Our visual perception is primarily based on detecting local discontinuities and patterns. This is why we typically prefer reading black letters on white pages, or vice-versa: there is a sharp contrast (or discontinuity) between the *foreground* (the object of interest) and the *background* of the image.
The mathematical operation on 2D distributions that most closely mimics human vision is *Convolution*, which also offers the advantage of being a linear operation.

The integration of convolution into NNs has been one of the most significant successes of the last decade. Indeed, CNN address both previously mentioned issues, spatial information loss and parameter explosion, by providing high-performance models. For the same MNIST task, a CNN can be shown to reach an ACC of 99% with only $92,322$ free parameters, a significant reduction compared to the MLP.

In this section, convolution and classical computer vision will be presented first. Subsequently, CNN for classification and segmentation tasks will be introduced.

2.2.1 Convolution

Convolution is the fundamental operation underpinning CNNs. Originally developed in the fields of signal and image processing for filtering and feature extraction, this operation describes how the shape of one function is modified by another. Before exploring its computational applications and discrete implementation, it is necessary to present the mathematical definition in the continuous domain.

Definition 3. Let f be a function of n variables $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$. Given another function w, the convolution $s(\boldsymbol{x})$ of f and w is defined as:

$$s(\boldsymbol{x}) = (f * w)(\boldsymbol{x}) \equiv \int_{\mathbb{R}^n} f(\boldsymbol{a}) w(\boldsymbol{x} - \boldsymbol{a})\, d\boldsymbol{a}. \tag{2.14}$$

In this context, f is referred to as the *input*, w as the *kernel* (or filter), and s as the *feature map*.

In the context of computer vision, an image is a 2D distribution defined on a 2D *discrete grid of points*, which are commonly referred to as pixels[3]. To apply Definition 3 to an image for feature extraction, a discrete reformulation of the definition is required.

Definition 4. Let f be a function representing an image on a grid of size $N \times M$. Let w be a kernel of size $K \times L$, where $K < N$ and $L < M$. Given an integer S referred to as the *Stride*, the resulting feature map s, defined on a grid of size $\lfloor \frac{N-K}{S} + 1 \rfloor \times \lfloor \frac{M-L}{S} + 1 \rfloor$, is given by:

$$s(x, y) = \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} f(S \cdot x + i, S \cdot y + j) w(i, j) \tag{2.15}$$

where (x, y) are the coordinates in the feature map.

From Definition 4, it is evident that the extension to higher-dimensional inputs is straightforward. The definition also shows that, in general, the output size is smaller than that of the input. In this context, before applying the convolution operation, we can expand the input by adding zero values along the borders. This operation, which does not inherently change the existing feature values, allows for control over the output size and is called *zero-padding*. As an example, if we aim to obtain an output size equal to the input size, it is sufficient to solve the equations:

$$\frac{(N + 2P_1) - K}{S} + 1 = N \quad \text{and} \quad \frac{(M + 2P_2) - L}{S} + 1 = M, \tag{2.16}$$

where P_1 and P_2 indicate the amount of zero-padding added in each edge. In the common case where $S = 1$ and the kernel has a square size $(K \times K)$, the result is $P = \frac{K-1}{2}$. For instance, in Figure 2.10, where $K = 3$, an output size equal to the input

[3]The term *pixel* (a portmanteau of *picture element*) originally refers to the individual sensing elements in a Charge-Coupled Device (CCD).

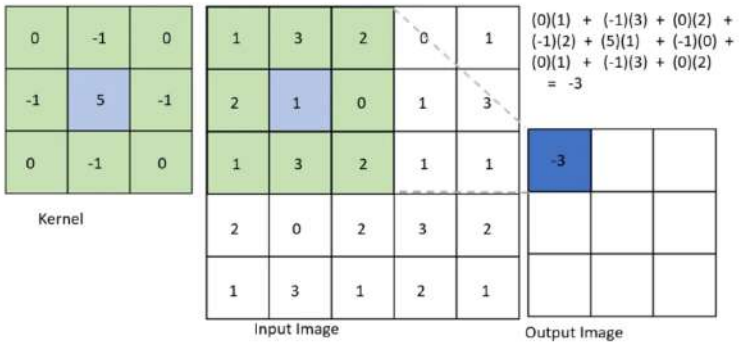


Figure 2.10: Graphical explanation of convolution. In this case, the input has a size of 5×5 and the kernel has a size of 3×3 with a stride $S = 1$. By applying the discrete definition, the resulting feature map has a size of 3×3. The first element of the output, indexed as $(0, 0)$, is the result of the element-wise product and summation obtained by overlapping the kernel center on the input pixel at coordinates $(1, 1)$. The mathematical operations are indicated in the top-right corner. To obtain the remaining output pixels, the kernel is shifted to the right by a number of positions equal to the stride S; once a row is completed, the process moves to the next row.

is obtained by adding one row and one column of zeros to each edge ($P = 1$). This specific configuration is commonly referred to as "same" padding (see figure 2.11).

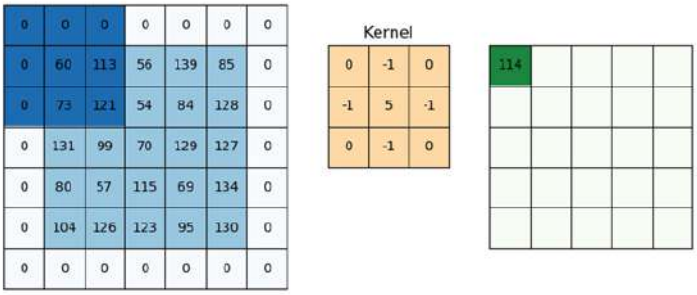


Figure 2.11: Zero-padding for output size equal to the input.

Before the revolution of the last decade, the entire field of AI-based computer vision was focused on finding specific kernels (or filters) able to extract particular features.

Mean filter

Averaging operations in signal analysis are widely used for noise reduction. For instance, most techniques in astrophotography are based on capturing multiple images and then, after proper registration (to account for the motion of the *celestial sphere*), applying an averaging operation. This process across different pictures can be referred to as a *temporal mean*.

When only a single image is available, we can instead calculate a *spatial mean* by averaging pixel values within a small local neighborhood. This is achieved by assigning to each pixel the mean value of its neighbors, for example, the eight pixels surrounding a central pixel. Specifically, we calculate the sum of pixel values in a 3×3 window (including or excluding the center) and then divide by the total number of considered

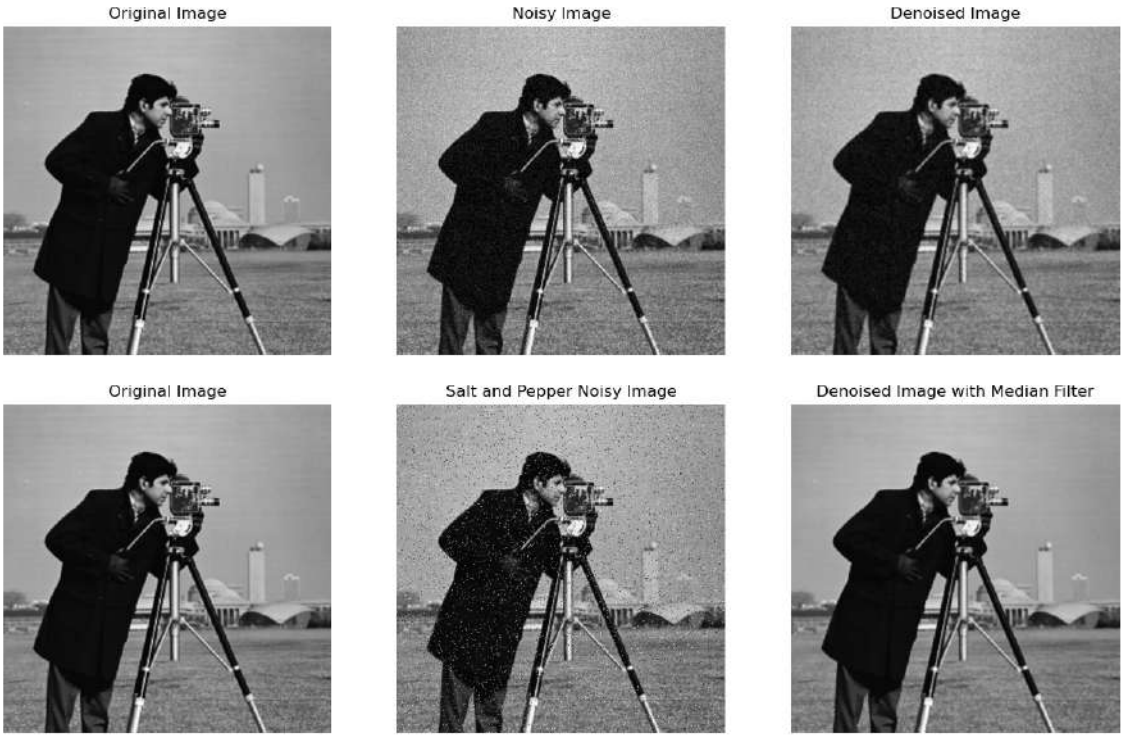


Figure 2.12: Mean filters and denoising applied to the standard *Cameraman* test image. Upper panel: white noise obtained by sampling from a zero-centered normal distribution with a standard deviation equal to 8% of the maximum intensity, followed by a mean filter (kernel size = 3, including the center). Lower panel: salt and pepper noise, with a noise density of 5%, and the application of a median filter (kernel size = 3).

pixels (8 if the center is excluded, 9 otherwise). This operation can be efficiently performed using convolution with the following kernels:

$$k = \frac{1}{8}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{or} \quad k' = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{2.17}$$

While this operation is effective against white noise (noise normally distributed around zero), it is less suitable for other types of interference. A common example is *salt and pepper noise*, which consists of random pixels being replaced by either zero (black) or maximum (white) intensity values. This characteristic gives the noise its name. We can observe this phenomenon, for example, in old printed photographs where the emulsion has physically peeled off.

In such cases, a mean filter performs poorly. A better alternative is the *median filter*, which replaces each pixel with the median value of its neighborhood. It is important to note that the median filter is not a convolutional filter, as the median is obtained through an order-based operation rather than a linear combination. Figure 2.12 compares the action of the mean filter on an image with white noise (upper panel) and the median filter on an image with salt and pepper noise (lower panel). In both cases, the standard *Cameraman* test image was used with simulated noise.

Edge detection

The human perception of edges is based on strong discontinuities between uniform regions. For example, when looking at a desk, we clearly perceive the edges because of the sharp discontinuity between the desk surface and the region beyond it. Since images are 2D distributions, this effect can be evaluated in terms of gradients: we expect high gradients near edges and low gradients in uniform regions. When considering gradients, direction must also be taken into account; specifically, we expect high horizontal gradients at vertical edges and high vertical gradients at horizontal ones.

Let us recall the mathematical definition of the partial derivative along the x-direction for a 2D function at the point (x_0, y_0):

$$\frac{\partial}{\partial x} f(x_0, y_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x, y_0) - f(x_0, y_0)}{\Delta x}. \tag{2.18}$$

Since images lie on a 2D discrete grid of points, we can define the *finite difference* derivative (specifically the central difference) as:

$$\begin{aligned} \Delta_x f(x_0, y_0) &= f(x_0 + 1, y_0) - f(x_0 - 1, y_0) \qquad \text{along } x, \\ \Delta_y f(x_0, y_0) &= f(x_0, y_0 + 1) - f(x_0, y_0 - 1) \qquad \text{along } y. \end{aligned} \tag{2.19}$$

Taking inspiration from Equations 2.19, we can define the *Prewitt* and *Sobel* filters for gradient calculations:

$$\text{Prewitt:} \quad \Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \qquad \Delta_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.20}$$

$$\text{Sobel:} \quad \Delta_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \qquad \Delta_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.21}$$

This approach was further developed to provide highly effective solutions for edge detection. The gold standard of these methods is the Canny algorithm, which utilizes Sobel filters along with several other preprocessing stages. Detailed information can be found in [47].

2.2.2 Convolutional Neural Network

We have seen that convolution is a powerful way to extract meaningful patterns from images, partly because it resembles how the human visual system processes visual information. Before DL, computer vision mainly relied on hand-crafted convolution kernels designed to detect specific shapes or textures. This led researchers to wonder whether NNs could learn these filters on their own, discovering features that humans

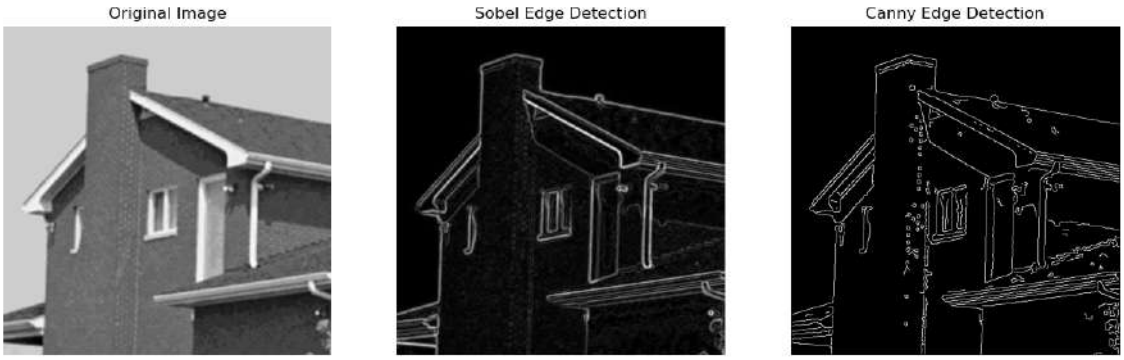


Figure 2.13: Example of edge detection performed using convolutional filters. The central panel shows the application of Sobel filters to the original image (left panel); the result is computed as the *magnitude* of the gradients obtained in both horizontal and vertical directions. The right panel displays the output of the Canny algorithm. The standard *House* test image was used for this comparison.

might not even think of. In deep models, this idea goes further: complex patterns can be built step by step from simpler ones.

A CNN is a NN that exploits convolution as its fundamental linear operation instead of matrix multiplication. In other words, in a CNN a neuron is no longer defined as the linear combination of inputs followed by an activation function. Rather, the neuron corresponds to a convolutional kernel applied to an input arranged in a grid format, such as an image, and its activation depends on the local pattern it detects. Unlike in an MLP, the NN parameters to learn are the entries of kernels, and since the same kernel is applied across the entire image, the operation is translationally invariant. This property, which is an additional advantage of convolutional architectures, is known as *parameter sharing*: it allows the same filter to operate over the whole spatial domain of the input, drastically reducing the number of learnable parameters and improving generalization. A representation is depicted in figure 2.14.

It is instructive to examine the implications of translational invariance and parameter sharing. Once a convolutional kernel has learned to detect a given feature within an image, the same set of weights can be applied uniformly across all spatial locations, ensuring that the feature is recognized independently of its position in the input. This spatial equivariance, combined with weight sharing, enables convolutional layers to generalize feature detectors across different images and spatial contexts. Such behavior is intrinsic to CNNs, whereas MLPs, due to their fully connected structure and absence of shared parameters, lack this form of inductive way to learn.

As written above, in deep learning performance typically improves with increasing network depth. The key challenge, however, is to make the network deeper while avoiding an explosion in the number of parameters. Additionally, we should consider that the most relevant features and the window size required to detect them depend on the specific task we aim to solve. This leads us to the concept of the *Receptive Field*, i.e., the region of the input that a unit "looks at" to extract features. In a single convolutional layer, this corresponds to the kernel size. It is clear that we can

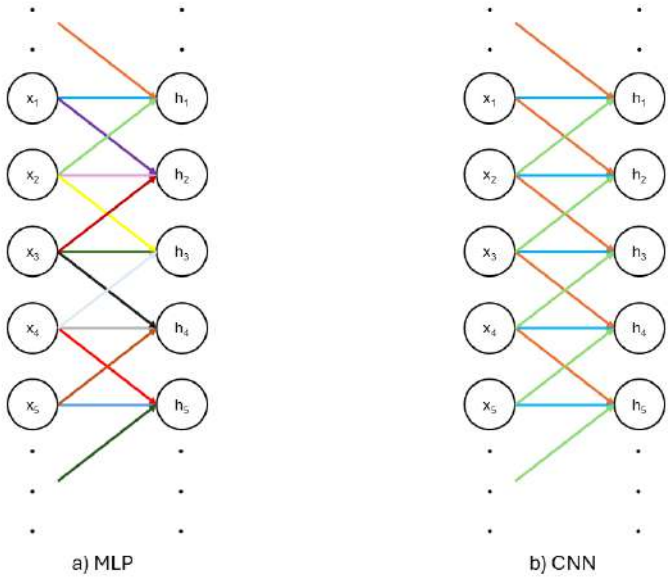


Figure 2.14: Parameter dependence in a) MLP and b) CNN. Each color denotes a distinct learnable weight. In a), every connection between input and hidden units carries an independent parameter, reflecting the fully connected structure of an MLP. In b), hidden units share the same set of kernel weights across different spatial positions, so identical colors highlight the reuse of the same parameters over multiple local receptive fields, exemplifying convolutional weight sharing.

only detect features whose spatial extent fits within the kernel: in other words, we cannot detect a feature that manifests itself over a region larger than the kernel size. A straightforward solution would be to use large kernels; however, the number of learnable parameters grows quadratically with the kernel size. For instance, a kernel as large as the entire image—guaranteeing the ability to detect features of any size—would be equivalent to a dense layer in a MLP. How can we address this problem? The solution is simpler than one might expect: using small kernels (e.g.,3×3) and stacking many layers. The intuition is that, as we go deeper in the network, the feature maps progressively integrate information from earlier layers. For example, to obtain a receptive field of 5×5, we may either use a single convolutional layer with a 5×5 kernel (25 parameters) or two consecutive layers with 3×3 kernels (9 parameters each). Figure 2.15 illustrates this behavior. This simple example shows the reason why general CNNs exploit convolutional layer with 3×3 kernel size.

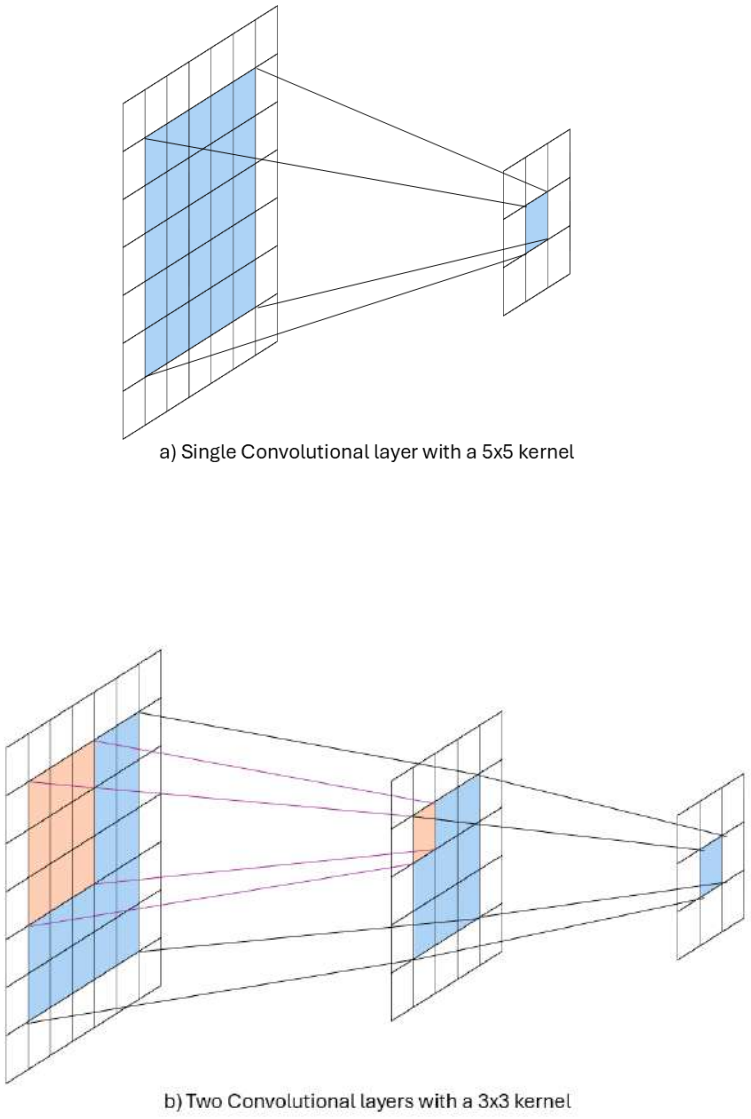


Figure 2.15: Comparison between a single 5×5 convolutional layer (a) and two stacked 3×3 convolutional layers (b). While both configurations achieve an equivalent receptive field, the latter requires significantly fewer parameters, illustrating the efficiency of deep architectures built from small kernels.

CNN general architecture

The typical CNN architecture consists of a convolutional component stacked on top of a relatively small MLP. In the DL community, the convolutional part is often referred to as the feature extractor: convolutional operations extract hierarchical features that are then passed to an MLP classifier as a one-dimensional vector. The overall network exhibits a tapered shape, where the spatial resolution is progressively reduced—first down to a manageable number of feature maps (end of the convolutional stage), and then to the final number of output units—equal to the number of classes—through the MLP.

The great success of CNNs began with *AlexNet* [48], proposed by Alex Krizhevsky and Ilya Sutskever under the supervision of Geoffrey Hinton (awarded the Nobel Prize in Physics in 2024). It consists of eight layers, the first five being convolutional and the last three fully connected. Starting from an input layer of $227 \times 227 \times 3$ (the RGB image size in ImageNet), the first hidden layer is a convolutional layer with 96 filters of size 11×11 and stride $S = 4$. By applying Definition 4, we obtain an output size of $55 \times 55 \times 96$. As we can see, the reduction in spatial resolution is compensated by an increase in the number of feature maps. Subsequently, a max-pooling layer is applied (kernel size $K = 3 \times 3 \equiv 3$, $S = 2$). This type of layer does not contain learnable parameters and is typically applied after a convolution to compute a summary statistic, such as the maximum or the average value within a window. The output size becomes $27 \times 27 \times 96$.

The second convolutional layer has 256 filters with $K = 5$, $S = 2$, and zero padding $P = 2$ (same padding), followed by a max-pooling layer ($K = 3$, $S = 2$). The output size is $13 \times 13 \times 256$. Subsequently, there are three convolutional layers ($K = 3$, $S = 1$, $P = 1$), the first two with 384 filters and the last one with 256. The output size remains $13 \times 13 \times 256$. A max-pooling layer ($K = 3$, $S = 2$, output size $= 6 \times 6 \times 256$) is applied before a flattening layer (output size = 9216). The network then includes an MLP with two fully connected layers, each with 4096 units, followed by an output layer with 1000 units, corresponding to the number of classes. All layers use ReLU activation functions, except for the last one, which employs a softmax activation. A schematic representation of the AlexNet architecture is depicted in figure 2.16

Visual Geometry Group Networks

A Visual Geometry Group Network (VGG) network extends the AlexNet approach by reducing the number of parameters while enabling substantially deeper architectures. In general, the VGG name is followed by a number indicating the total number of hidden layers. The most widely used models are VGG16 — comprising 13 convolutional layers and 3 fully connected layers — and VGG19 — with 16 convolutional layers and 3 fully connected layers [49].

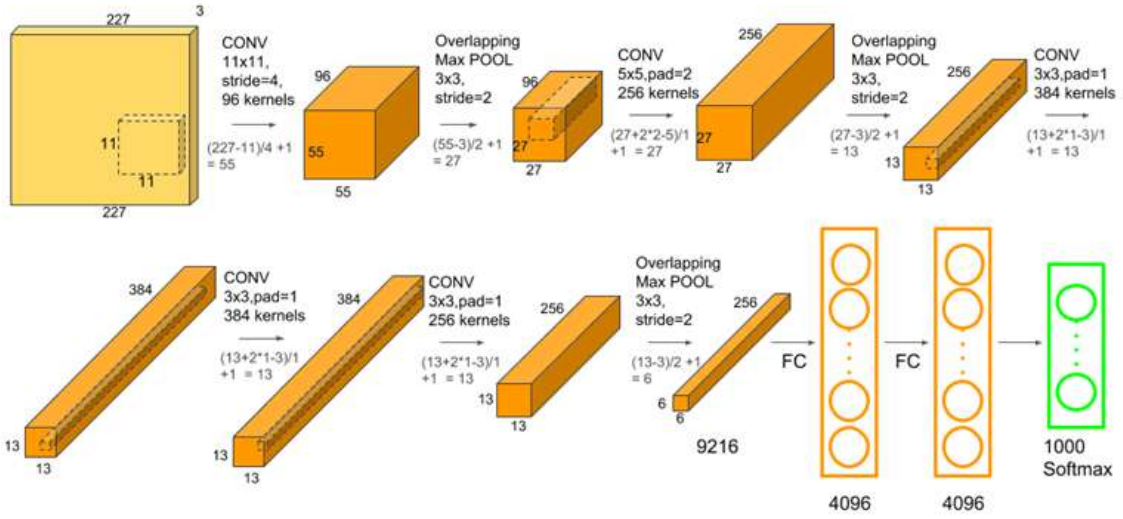


Figure 2.16: AlexNet architecture scheme.

The main difference with AlexNet lies in the exclusive use of small convolutional kernels — typically 3×3 — throughout the architecture (a schematic representation is depicted in figure 2.17). As shown in the previous section, this design choice reduces the number of parameters while preserving the same receptive field. VGG16 and VGG19 contain approximately $138M$ and $144M$ parameters, respectively, making their training feasible today even in in-house facilities (including for 3D tasks). Due to these characteristics, VGG networks became the state of the art in 2014 and remained a consolidated reference for 2D tasks until 2021, when they were progressively replaced by more efficient and higher-performing architectures, such as Visual Transformers [50,51]. In 3D tasks VGGs are still widely used.

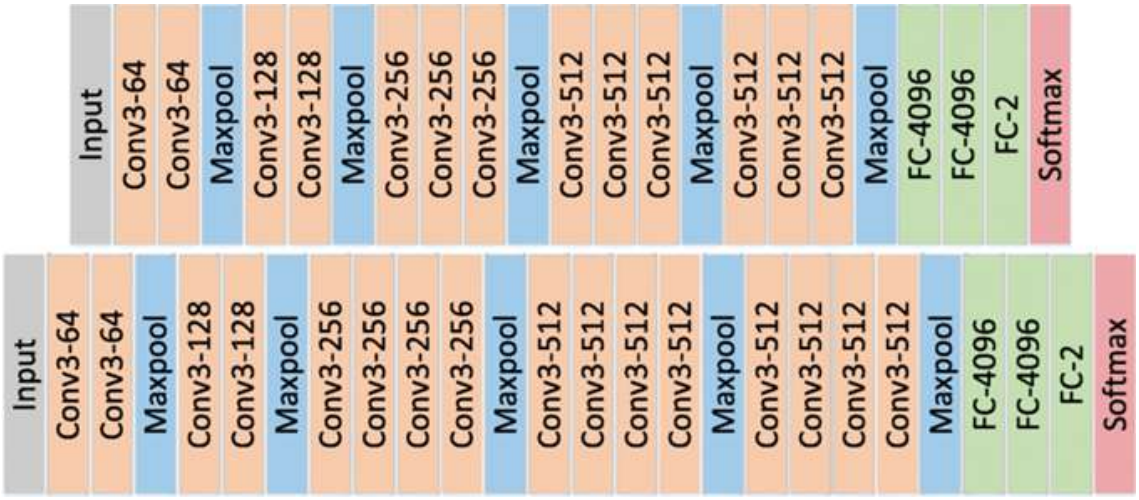


Figure 2.17: Schematic representation of VGG16 (top) and VGG19 (bottom). Conv3 stands for convolutional layer with kernel 3×3 in size, while FC stands for fully connected layer. The number following the layer type indicates the number of feature maps in the Conv case and the number of hidden units in the FC case. In this case the model performs a binary classification, as the final layer has size 2. Image has been adapted from reference [52].

Although CNNs exhibit extraordinary performance on image-related tasks, the risk of overfitting is still present. General regularization techniques are commonly adopted; however, CNN-specific methods have also been introduced. In architectures such as VGG, which are very deep, the most frequent issue is the *evanescent gradient*, i.e., optimization gradients that vanish due to the progressive multiplication of functions occurring in the back-propagation algorithm. This effect becomes more severe as the network depth increases (introducing more layers translates into more function multiplications). A Batch Normalization (BN) layer can mitigate this issue. Applied after a convolutional layer, it normalizes the outputs (over the minibatch, i.e., different examples or images) so that their mean is zero and their standard deviation is one. It is important to say the activation function should be always applied after a BN. Another method is based on the Spatial Dropout (SD) layer, a dropout variant designed for convolutional layers. In this case, entire feature maps are "switched off" during training. In general, when building a CNN, convolutional blocks—containing convolutional, BN, activation, and SD layers—are used instead of pure convolutional layers (or instead of listing the full sequence of layers each time). Empirical evidence and established good practices indicate that BN layers applied after each convolution tend to improve performance, whereas SD layers should be applied only in the deeper parts of the network.
Finally, another regularization technique is based on avoiding max-pooling layers. In fact, this layer forces the computation of a summary statistic for spatial downsampling which can limit the NN in its learning capability. A variant which avoids this constraint is the use of the stride directly in the convolutional layer, thus obtaining a downsampling without using any pre-established function (see figure 2.18).

2.3 Image segmentation

Image segmentation refers to the process of separating a set of pixels/voxels from the rest of the image based on a given rule or aggregating property. In principle, the aggregating property can be anything; however, in most practical cases, we are interested in segmenting objects, such as people in a street image—as done by an autonomous driving system—or organs and tissues in biomedical images. In these situations, the underlying rule is not always straightforward, and for this reason, DL has represented a great opportunity to improve results. Nevertheless, it must be underlined that when a classical technique has been proven to be effective, the use of DL should be properly motivated. This concept applies to most DL applications.

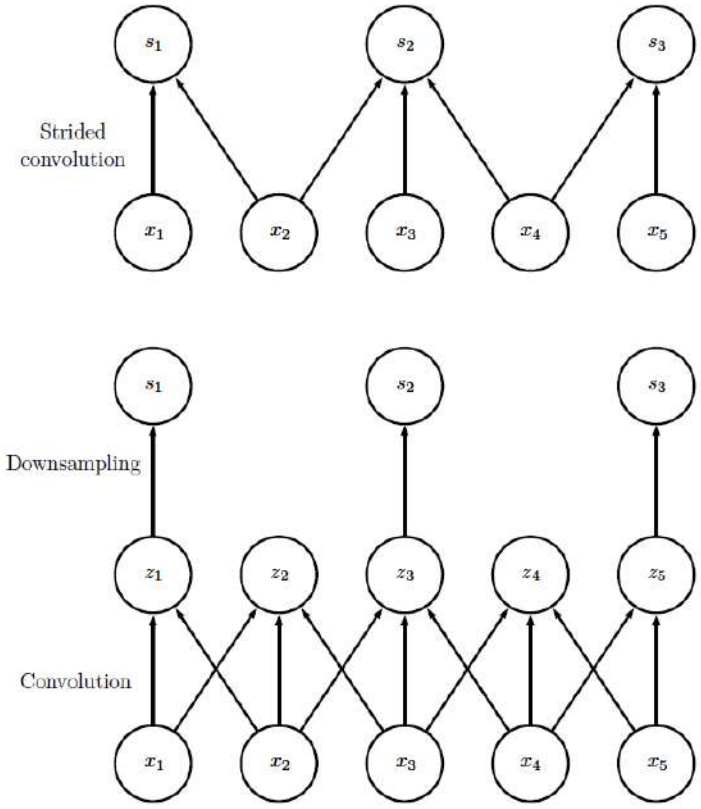


Figure 2.18: Convolution with a stride. In this example, we use a stride of two. (Top) Convolution with a stride length of two implemented in a single operation. (Bottom) Convolution with a stride greater than one pixel is mathematically equivalent to convolution with unit stride followed by downsampling. Obviously, the two-step approach involving downsampling is computationally wasteful, because it computes many values that are then discarded. Additionally, it constrains the summary operation.Image adapted from reference [29].

2.3.1 Classical methods

One of the most well-known classical segmentation techniques is based on the simple idea that the ensemble of pixels/voxels constituting the object of interest shares a common gray-level range. In other words, the aggregating property is a gray-level rule, such as: the pixels/voxels belonging to the object lie within the range $[G_{min}, G_{max}]$, where G_{min} and G_{max} define a gray-level visualization window. This method is known as *thresholding*. Thresholding works particularly well on organ segmentation in CT images. In CT images, the gray-level values are converted into Hounsfield Units (HUs), transforming an arbitrary intensity scale into a physical measure that relates to matter density. In general, density is almost constant within an organ, so thresholding becomes the perfect technique to separate it from the background.

In the thresholding method, two main limitations must be considered. The first concerns the identification of the gray-level window: the image histogram should allow a clear separation between classes; ideally, it should be at least bimodal. Moreover, the result strongly depends on the choice of the threshold values. A solution to this issue was proposed by Otsu with the *Otsu's algorithm* [53], which determines optimal thresholds by minimizing the intra-class variance while maximizing the inter-class variance.

The second limitation arises from the fact that thresholding is a global method, meaning that parts of the background may share the same gray-level values as the objects of interest. Consequently, thresholding can only be reliably applied to restricted regions of the image.

Another technique can be obtained by imposing that the pixels/voxels within an object of interest share a large range of intensity values—generally also shared with the background—but within the object the contrast between neighboring pixels/voxels—i.e., the gray-level difference—is lower than a fixed threshold. This approach is referred to as the *contrast-based method*, whose main algorithm is *Region Growing*.
Region Growing works by starting from a seed point and including neighboring pixels/voxels if the gray-level difference with respect to a contrast threshold is satisfied. Each time a new pixel/voxel is added to the region, it is also included in the list of points to be explored, where the contrast check is performed again. The process stops when the list of points to explore becomes empty and no more pixels/voxels can be added to the region. The name "growing" arises because the region starts from a single point and progressively expands.
Compared to thresholding, region growing has the advantage of being a local method, thus overcoming one of the issues mentioned above. However, the choice of the contrast threshold strongly affects the results: in the worst cases, the region may either fail to grow or expand to include the entire image.

Segmentation classical methods can also be derived from clustering techniques or from edge detection algorithms, such as those presented in Section 2.2.1. Particularly in bioimaging, and especially in brain imaging, a very popular technique is known as the *atlas-based method*. In this case, a universal prototype containing the required ROIs—hence the name atlas-based—is known *a priori*. Segmentation is performed by finding the transformation that best aligns the image with the prototype and then applying the inverse transformation to the ROIs. In brain imaging, the most widely used prototype is the *MNI152 template*, on which most atlases are defined.

2.3.2 DL approaches

All the segmentation methods mentioned in the previous section are very interesting and powerful; however, they all have limitations that can generally be summarized into two recurring issues: the need for human intervention and processing time.
The human component in classical segmentation methods is not negligible: most of the time, users must input parameters and use a trial-and-error approach to find an optimal solution. Indeed, these methods are often referred to as *semi-automatic methods*.
Processing time in segmentation is also crucial, as it is the main reason motivating the use of AI-based methods. Classical algorithms, such as region growing or atlas-based methods, can require very long computational times, often leading users to resort to manual segmentation.

DL models, when properly trained, provide an appropriate trade-off between speed and reliability, virtually eliminating the human component.
Historically, the major successes of DL in image segmentation occurred after the introduction of CNNs. Chronologically, object detection models were developed before segmentation models, due to the higher complexity of performing pixel-wise classification. In fact, in object detection the output consists of simple rectangular regions, whereas the geometry of a segmentation can be much more complex.
First object detection models worked by dividing images into patches—mostly using classical methods—and passing them to a CNN trained to classify the presence of an object. The main drawback of this approach was the computational cost: the model had to analyze many patches from the same image before identifying a bounding box.

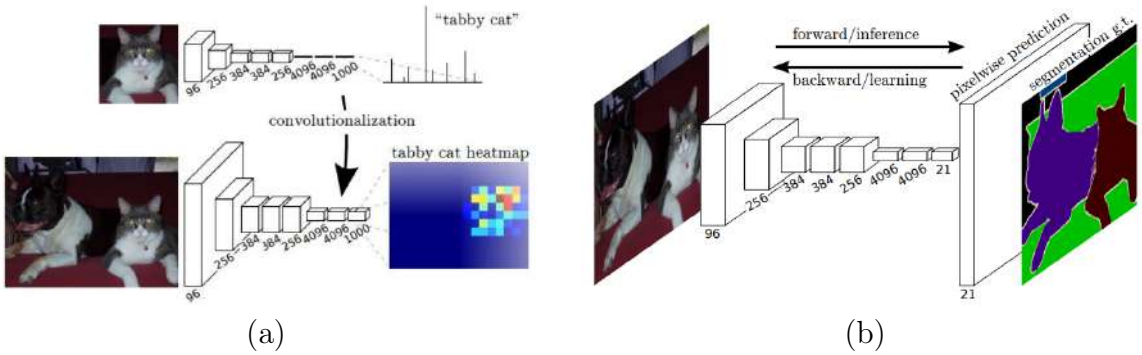


Figure 2.19: From a CNN to a FCN. (a) Class activation map highlighting the detected object. This representation lies in a latent space whose dimensionality is smaller than that of the original image. Each "pixel" in this space corresponds to a patch of the input image. (b) FCN architecture producing pixel-wise predictions. Figure adapted from [54].

Later, researchers realized that feature extraction in CNN models could also be exploited for object detection. This idea was inspired by the observation that the last feature maps of a convolutional classifier contain *activation patches* (see figure 2.19 a), i.e., local activation patterns produced when convolutional filters detect particular structures in the image[4]. In other words, suppose a CNN is able to classify the presence of a particular object in an image (e.g., cats, dogs, people, tumors). The activation patches are related to the regions of the image containing that object, and therefore they can be used to estimate bounding boxes or perform image segmentation. The remaining challenge concerns how to resize or project an activation patch so that it matches the original image shape.
In object detection, this approach led to the development of architectures such as *YOLO v1* (You Only Look Once, [56, 57]), which effectively overcame the limitations of earlier object detectors. This model processes the entire image and projects the output into a

[4]Activation patches should not be confused with the activation maps produced by the *Grad-CAM* algorithm [55]. In the former case, neuron activations are directly visualized, whereas in the latter the activation map is obtained from the gradient of the predicted class with respect to the last convolutional layer.

space of size $7 \times 7 \times 5$. The idea is to predict the position of the bounding box center on a grid of 49 patches of the image, along with the coordinates of the center within each patch, the height and width of the bounding box, and a value representing the confidence score—i.e., the probability that the center lies within the considered patch. In a single step, the model performs both patch classification and bounding box size estimation. Hence the name "You Only Look Once".
In image segmentation, Fully Convolutional Networks (FCNs) were developed. The basic idea is that the convolutional part of a CNN can be seen as an *encoder*, which transforms the original image into activation maps, i.e., representations of the input in a latent space. Under this assumption, segmentation can be obtained by applying a decoder model, which operates in the opposite direction of the encoder, projecting the latent representation back into the original image space (see figure 2.19 b). The solution was found by introducing the *transposed convolutional layer*, a convolutional layer that expands the spatial size of its input by applying learnable kernels. The first work was proposed by Jonathan Long, Evan Shelhamer, and Trevor Darrell in *Fully Convolutional Networks for Semantic Segmentation* [54], where the authors achieved strong results in terms of the Intersection over Union metric on the PASCAL VOC, NYUDv2, and SIFT Flow datasets [58–60].

2.3.3 UNet

In 2015, Olaf Ronneberger, Philipp Fischer, and Thomas Brox published the paper *U-Net: Convolutional Networks for Biomedical Image Segmentation*, in which an innovative FCN architecture is introduced [61]. In this architecture, the decoder part is nearly symmetric to the encoder—i.e., the convolutional part of the CNN. Moreover, *skip connections*—i.e., alternative computational paths—are included to mitigate the vanishing gradient problem. In fact, while sequential layer applications translate into derivative multiplication in the backpropagation algorithm, alternative paths translate into derivative summation, compensating for the multiplication of very small numbers in deeper layers.

The U-Net architecture can be divided into levels, where the spatial resolution remains constant within each level. Thanks to the symmetric structure, skip connections link the encoder and decoder at corresponding levels. The deepest level, also called the bottleneck, represents the main connection between the encoder and decoder parts. Here, the network projects the input into the latent space before starting the decoding stage, which converts the representation into the corresponding segmentation. The last layer typically has a sigmoid activation, since segmentation can be seen as a pixel-wise (or voxel-wise) classification task. When the aim is to segment more than one ROI, the same architecture can be used by replacing the final activation with a softmax function. In this way, the model can segment multiple ROIs by processing the input only once. Additionally, U-Nets have also been used in generative tasks by replacing

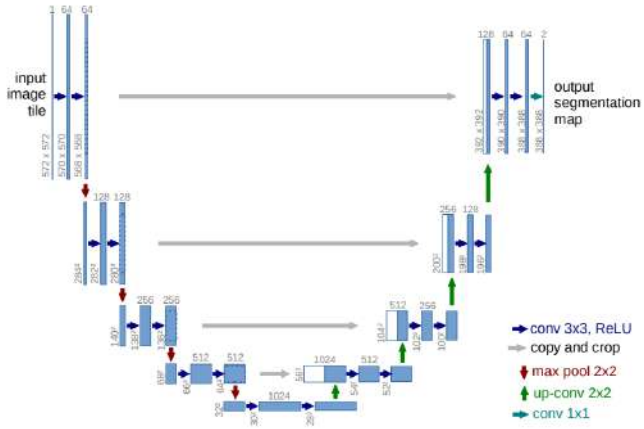


Figure 2.20: UNet architecture as proposed in the original paper [61]: the left side corresponds to the encoder, and the right side to the decoder. The network is organized into levels, each containing two convolutional blocks. Each convolutional block consists of a 2D convolution layer (kernel size 3) followed by a ReLU activation. The numbers on the left of each block indicate the input shape, while the numbers on top indicate the number of feature maps. The encoder uses max pooling with stride 2 after each level. At the bottom of the network is the bottleneck, from which the decoder starts. The decoder upsamples the feature maps and concatenates them with the corresponding encoder outputs. In the last layer, a softmax or sigmoid activation function is applied to perform pixel-wise classification.

the final activation with a linear one. In other words, the U-Net architecture is very versatile for tasks that transform an image from one representation to another while preserving the spatial structure.

The 3D UNet, also known as *VNet*, was introduced in 2016 by the same team [62]. It is obtained by replacing 2D convolutional layers with their 3D counterparts. Since configuring 3D UNet hyperparameters can be very challenging, Fabian Isensee introduced *nnUNet* in 2021 [5]. nnUNet is often seen as a modified UNet architecture, but in reality, the acronym stands for "no new UNet". This misconception arises because nnUNet made 3D UNet applications simple and accessible. Thanks to this, nnUNet gained great popularity, which led many people to believe it was a completely new architecture. Beyond the misconception about it being a different architecture, Isensee's work is of extraordinary importance in the medical field, as nnUNet has greatly increased the number of practical applications, significantly advancing knowledge in the domain.

2.4 Physics Informed Neural Network

In this and in the previous chapter we have seen that the loss function used for optimization is defined over a highly dimensional space. Exploring the parameter space, in the strict sense, is in general unmanageable. However, in Section 1.4.3 we saw that additional loss contributions can be used to bound portions of this space, thus improving optimization. When physical equations are used for this purpose, the approach is called **Physics-Informed Machine Learning**. A NN trained to satisfy physical laws is referred to as a **Physics-Informed Neural Network (PINN)**.
This approach, in addition to improving optimization, has the advantage of making

NNs more reliable. However, in most cases, incorporating physical laws as loss terms is not trivial. During NN optimization, differentiability and parallelization must always be guaranteed, which may make the inclusion of certain terms in the loss function impossible.
To understand how this approach can be used, a practical example is presented. Suppose the aim is the segmentation of a ROI whose size is *a priori* known. Let the ROI volume $V(y_{true})$ be known. For a segmentation task we could use a composite loss given by the sum of the binary cross-entropy CE and the DSI loss $\mathcal{L}_{DSI}$[5].
In this case, the physical information about the volume V, since it corresponds to the sum of the predicted binary mask multiplied by the voxel volume—an operation that is clearly differentiable and parallelizable—can be incorporated by defining the loss $\mathcal{L}$ as:

$$\mathcal{L} = \alpha CE(y_{true}, y_{pred}) + \beta \mathcal{L}_{DSI}(y_{true}, y_{pred}) + \gamma \left(\sum y_{pred} - V(y_{true}) \right)^2 \tag{2.22}$$

where for simplicity the voxel volume is assumed to be equal to one, and α, β, and γ are weighting factors. The squared difference is used to ensure that only positive terms are summed.

A class of PINN models has become particularly relevant due to the tasks they address and the physical terms incorporated into the loss function. This is the case for neural networks used to solve Boundary Value Problems (BVPs), which will be discussed in the next subsection.

2.4.1 PINNs for solving Boundary Value Problems

A BVP is a physical–mathematical problem concerning Partial Differential Equations (PDEs). Many physical laws are expressed as PDEs. Examples include diffusion (heat equation), electrostatics (Poisson or Laplace equation), and wave propagation (wave equation), but also the Einstein field equations, the Navier–Stokes equations, Maxwell's equations, the Schrödinger equation, and many others. A PDE describes the general behavior of a system, often governed by some physical principle. However, to solve a specific problem, additional equations—known as Boundary Conditions (BCs)—are required. A PDE together with the corresponding BCs defines a BVP.
Mathematical frameworks define when a BVP is well-posed; in such cases a unique solution is guaranteed. On the other hand, we have already seen that NNs are universal approximators. By combining these two observations, the solution of a BVP can be approximated with a NN.
This approach was first proposed by Lee Hyuk and Kang In Seok in 1990 [63], for discretized numerical equations. Later, Isaac Elias Lagaris, Aristidis Likas, and Dimitrios I. Fotiadis demonstrated that NNs could be used to solve ordinary and partial differential equations [64], providing solutions in *closed analytical form*—i.e., solutions are

[5]The DSI loss is defined as $\mathcal{L}_{DSI} = 1 - DSI$.

provided in the continuum and remain differentiable without the need for interpolation algorithms. However, the approach became popular only between 2010 and 2020 due to computational limitations [65]. In 2021 a review on *Nature*, formalized the birth of Physics-Informed Machine Learrning [66].

The PINN approach to solve BVP is conceptually very simple. The underlying idea is the following: *If deep learning frameworks are very efficient at computing gradients with respect to parameters, why should they not be equally efficient at computing them with respect to the inputs themselves?* With this observation, the method becomes straightforward. Given a BVP—i.e., a PDE together with its boundary conditions—we can construct *ad hoc* loss functions to optimize a NN so that it satisfies these constraints. The starting elements are the domain Ω, a mathematical representation of the physical system; a PDE describing the phenomenon; an initial condition—i.e., the mathematical description of the system at the initial time t_0; and an equation describing the behavior at the boundary of the domain $\partial\Omega$. With these ingredients it is possible to define three loss functions.

Given a PDE, we can define a differential operator D. As an example, consider the 1D heat equation $\frac{\partial T}{\partial t} - k\frac{\partial^2 T}{\partial x^2} = 0$ describing the distribution of temperature T along a rod. In this case, the PDE operator is defined as $D = \frac{\partial}{\partial t} - k\frac{\partial^2}{\partial x^2}$ and the PDE can be rewritten as $DT = 0$. With this definition we can construct the *residual loss* $\mathcal{L}Res$. Suppose that N_1 points $\{(x_i, t_i)\}$ are randomly sampled from the interior of Ω. The residual loss is defined as

$$\mathcal{L}_{Res} = \frac{1}{N_1}\sum_{i=1}^{N_1}\left[D\left(NN(x_i, t_i; \boldsymbol{\theta})\right)\right]^2 \tag{2.23}$$

where $NN(x_i, t_i; \boldsymbol{\theta})$ denotes the output of the neural network with parameters $\boldsymbol{\theta}$ evaluated at the point (x_i, t_i). In the case of the heat equation, the residual loss becomes

$$\mathcal{L}_{Res} = \frac{1}{N_1}\sum_{i=1}^{N_1}\left[\frac{\partial}{\partial t}NN(x_i, t_i; \boldsymbol{\theta}) - k\frac{\partial^2}{\partial x^2}NN(x_i, t_i; \boldsymbol{\theta})\right]^2. \tag{2.24}$$

The same approach can be applied to the boundary conditions. In general, the behavior of the system at the boundary is described by Dirichlet, von Neumann, or mixed boundary conditions. In the case of the heat equation, Dirichlet conditions can be written as $T(x = x_0, t) - T_1 = 0$ and $T(x = x_L, t) - T_2 = 0$, where L is the length of the rod. Here x_0 and x_L define the boundary $\partial\Omega$. Let BC denote the operator describing the boundary equations. Suppose that N_2 points $\{(x_i, t_i)\}$ are randomly sampled from $\partial\Omega$. The boundary loss is defined as

$$\mathcal{L}_{BC} = \frac{1}{N_2}\sum_{i=1}^{N_2}\left[BC\left(NN(x_i, t_i; \boldsymbol{\theta})\right)\right]^2 \tag{2.25}$$

Finally, assuming that the initial condition can be provided either as an analytical expression or as a set of measurements, let us denote it as $y_{\text{true}}(x, t_0)$. The corresponding loss function $\mathcal{L}_{IC}$ is defined as:

$$\mathcal{L}_{IC} = \frac{1}{N_3} \sum_{i=0}^{N_3} \left[NN(x_i, t_0) - y_{\text{true}}(x_i, t_0) \right]^2, \tag{2.26}$$

where N_3 points $\{(x_i, t_0)\}$ are randomly sampled. Given the randomness of the sampled points, when the initial condition is provided as measurements, an interpolating function is required to provide a reference value at each spatial position.
In conclusion, optimizing a neural network to minimize the total loss, defined S:

$$\mathcal{L} = \mathcal{L}_{Res} + \mathcal{L}_{BC} + \mathcal{L}_{IC}, \tag{2.27}$$

directly translates into finding an approximate solution for the BVP under consideration.

The PINN method for solving boundary value problems (BVPs) has several distinctive features compared to a standard NN approach. First, since a PINN is trained to satisfy the equations of the BVP, it offers a degree of explainability. Second, the training is not purely supervised. Indeed, from equation 2.27, we can see that only the last term corresponds to a *data-driven component*. The terms $\mathcal{L}_{Res}$ and $\mathcal{L}_{BC}$ provide information in an unsupervised manner: random points are sampled from the domain, and their corresponding labels are given by Physics rather than measurements. Finally, the optimized model cannot generalize to other PDEs or systems, unlike typical DL approaches. In this context, the NN literally represents the **solution of the BVP in the domain** Ω, and it is valid only for that specific problem. Therefore, the PINN method for BVPs has both advantages and limitations and is particularly well suited for specific types of tasks, in contrast to general NN approaches, which in principle can be applied to a wider range of problems.

CHAPTER 3

A SEGMENTATION MODEL FOR VENTRAL INTERMEDIATE NUCLEUS IDENTIFICATION

In the last decade, the advent of biomedical imaging has significantly improved anatomical knowledge and the quality of both diagnostic and therapeutic practices. The capability of "observing" within tissues has enabled medical research to achieve results that were once considered almost miraculous. Before the development of imaging techniques, brain anatomy could only be studied *post mortem*, which severely limited the knowledge that could be gained through observation. Following the discovery of X-rays and, subsequently, the establishment of radiology as a medical discipline, the goal of visualizing *in vivo* three-dimensional structures inside the human body was achieved on October 1, 1971 by Godfrey Hounsfield. Together with Allan Cormack, he was awarded the Nobel Prize in Physiology or Medicine in 1979 for the invention of Computed Thomography (CT). The first CT scan was performed on the brain and, already demonstrating its diagnostic power, revealed the presence of a tumor. It is particularly intriguing that Hounsfield was able to conduct his research while working for EMI Company[1]. At that time, EMI was generating substantial profits from the success of the Beatles, which made it possible to finance Hounsfield's work. Even more fascinating is that Paul McCartney reportedly expressed interest in supporting this research, stating that *whatever could be discovered might be of great help to science.*
Hounsfield indirectly contributed to the development of Magnetic Resonance Imaging (MRI) too, another major diagnostic technique for visualizing three-dimensional structures within the human body. MRI originated from the discovery of Nuclear Magnetic Resonance (NMR) independently performed by Felix Bloch e Edward Purcell. In 1973, Paul Lauterbur published in *Nature* the first NMR images, obtained from simple objects such as water-filled tubes. Subsequently, in 1974, images of small animals were reported. In 1977 Peter Mansfield developed a technique able to scan rapidly bodies, producing

[1]Electric and Musical Industries (EMI) was a British multinational conglomerate founded in March 1931 in London, also well known as a major record label.

an image of a human finger.
Compared to CT, MRI required more time to be adopted in routine clinical practice. This delay can be attributed to several factors. First, the technology is significantly more complex than that used in CT, particularly due to the need to generate very strong magnetic fields, typically on the order of a Tesla. Additionally, another relevant factor concerns the terminology associated with the underlying physical phenomenon, namely *Nuclear Magnetic Resonance*. NMR involves interactions with nuclear spins—specifically their precession in a magnetic field—and the associated energies lie in the radiofrequency range, far below those of ionizing radiation, such as X-rays or gamma rays, produced by nuclear reactions or decay processes. However, in the 1970s, the term "nuclear" generated public concern, negatively affecting the perception of the technique and, to some extent, slowing its clinical and financial development. This is one of the reasons why the term MRI was later preferred over NMR in medical applications. Nevertheless, despite these difficulties, MRI became a routine clinical practice in the 2000s. Paul Lauterbur and Peter Mansfield were awarded the Nobel Prize in Physiology or Medicine in 2003.

In the context of personalized medicine and real-time treatment planning systems, AI has assumed a role of great importance. Auto-segmentation tools—whose outputs must still be validated by clinicians—are often integrated into clinical workflows, and AI-powered systems capable of adapting treatments based on patient-specific features help reduce analysis time to just a few minutes. The combination of high-quality anatomical imaging and AI systems has significantly improved both diagnostic findings and patient care. This is the case of Transcranial Magnetic Resonance guided Focused UltraSound (TcMRgFUS) treatment. In this technique, MRI supports the procedure in real time, making it non-invasive, safer, and more efficient compared to alternative approaches. Treatment planning based on patient imaging is almost always mandatory; however, intraoperative adjustments—required due to patient-specific changes—are often limited by time constraints. In such scenarios, AI provides valuable support.
In this chapter, an application of AI to support TcMRgFUS treatment for Essential Tremor will be presented and discussed in terms of background, methodologies, and results. First, brain MRI and TcMRgFUS will be introduced, with a focus on their relevance to this work. Subsequently, the development process of the AI model will be outlined, highlighting the challenges encountered, current limitations, and ongoing improvements. Despite the results obtained so far, this work is still in progress, and further efforts are required to fully demonstrate the reliability and efficacy of the developed models.

3.1 Brain MRI and Transcranial Magnetic Resonance guided Focused Ultrasound treatments for Essential tremor patients

3.1.1 Brain Magnetic Resonance Imaging

Nowadays, MRI drives a large portion of medical and anatomical discoveries concerning the brain[2]: early diagnosis, personalized medicine, the study of pathologies, structural and functional connectivity, surgical planning, and many others.
In contrast to other radiological techniques, MRI offers a wide range of applications. In particular, scan parameters can be tuned to investigate different tissue properties as well as functional aspects. It is well known that structural images can be obtained from signals that depend on *longitudinal and transverse relaxation times* or on *proton density* within tissues–corresponding to T_1, T_2, and ρ acquisitions, respectively. At the same time, water diffusion can be exploited to explore neuronal fiber connections, while blood oxygenation (due to the paramagnetic properties of deoxyhemoglobin) can be used to map brain activity. The number of physical markers that can be probed through MRI is very large and is often limited only by the available scanner time. Some brain MRI examples are reported in figure 3.1.

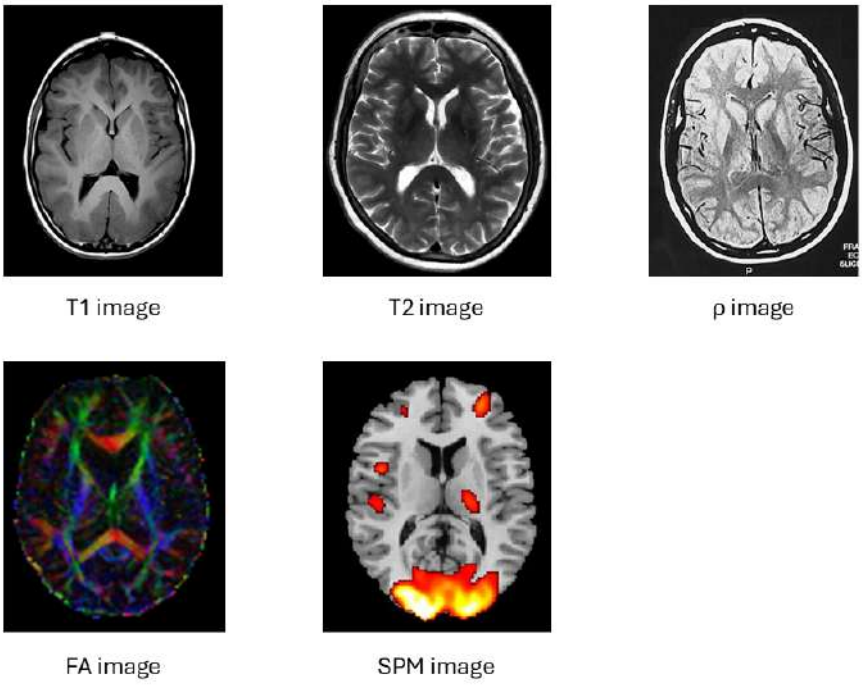


Figure 3.1: Example of different types of brain MRI. Top row: T_1, T_2, and ρ structural images. Bottom row: on the left, an example of an Fractional Anisotropy image derived from diffusion-weighted imaging; on the right, a Statistical Parametric Map obtained by overlaying the results of a statistical analysis of functional MRI onto a T_1 image.

[2]MRI has significantly impacted the study of all organs; however, its applications in brain research are particularly prominent.

Diffusion-Weighted Imaging and Fiber Tracking

In the work presented in this chapter, Diffusion Weighted Imaging (DWI) images are particularly effective for identifying areas not discernible through structural imaging alone. DWI is based on a specific MRI acquisition protocol, designed to investigate water diffusion properties within the brain. It consists of applying a perturbation—through a radiofrequency pulse—to protons within the tissue, followed by an opposite perturbation. As a result, stationary structures exhibit no net effect, whereas signal attenuation is observed in moving structures. By applying perturbations from different directions, the diffusion tensor can be estimated voxel-wise. Mathematically, six directions are sufficient for diffusion tensor estimation; however, typical clinical acquisitions involve approximately 64 perturbations, also referred to as gradients. In high-quality DWI datasets, such as those of the Human Connectome Project (HCP) [67, 68], the number of gradients can exceed 200.

DWI is motivated because water diffusion is strongly related to brain anatomy. In particular, in the absence of obstacles, water motion is isotropic, whereas strong anisotropy is observed near neuronal fibers, which constrain water diffusion along their principal directions. Studying the diffusion tensor in the brain allows the visualization of the so-called *brain structural network*. It is important to underline that neuronal fibers are generally smaller than the typical image resolution; however, studying water diffusion in the brain allows their *tracking*.

Fiber tracking is a method used to reconstruct structural connectivity in the brain. Two main methodologies are commonly employed for this purpose: ***deterministic tractography*** and ***probabilistic tractography*** (Figure 3.2). The former was historically based on modeling diffusion using a tensor represented as an ellipsoid and estimating fiber orientations along its principal eigenvector. The latter can be interpreted as a weighted *random walk*, in which propagation directions are sampled from the local *fiber Orientation Distribution Function* (fODF), a function describing the probability of white matter fibers being oriented in a particular direction within a voxel. Fiber tracking is then performed by allowing a *walker* to move between voxels, taking random steps according to the fODF. Over the years, more sophisticated algorithms have been developed for both approaches, substantially improving the quality and reliability of tractography results.

In both approaches, it is possible to emphasize specific neuronal pathways by selecting starting (seed) regions—from which tracts originate—target regions (i.e., regions to be reached, typically connected to the seed), and exclusion regions (e.g., brain ventricles or cerebrospinal fluid areas, where no white matter is present). Additionally, waypoint regions can be defined to constrain tractography, forcing tractograms to pass through selected areas. The latest is particularly useful in probabilistic tractography.

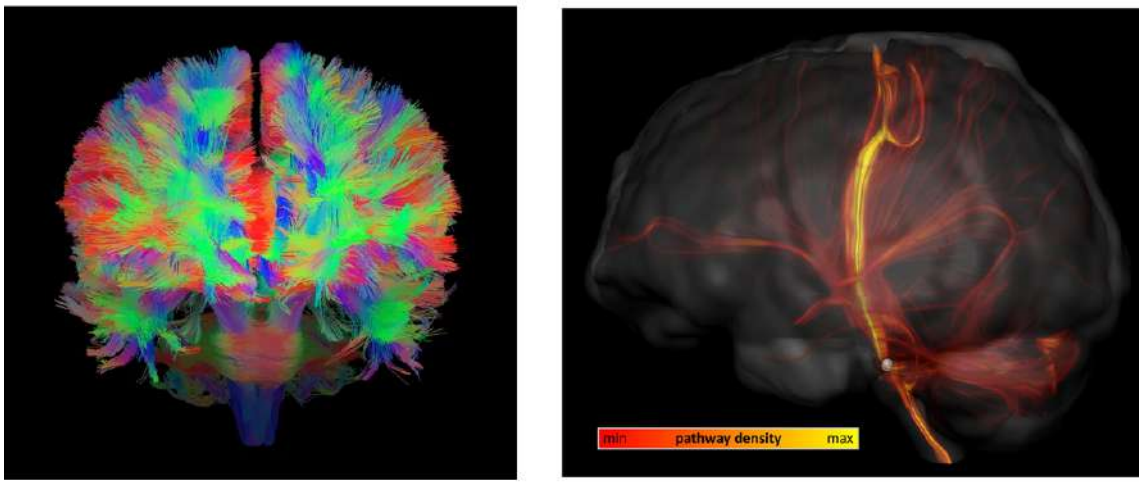


Figure 3.2: Fiber tracking examples: deterministic (left) and probabilistic (right) tractography.

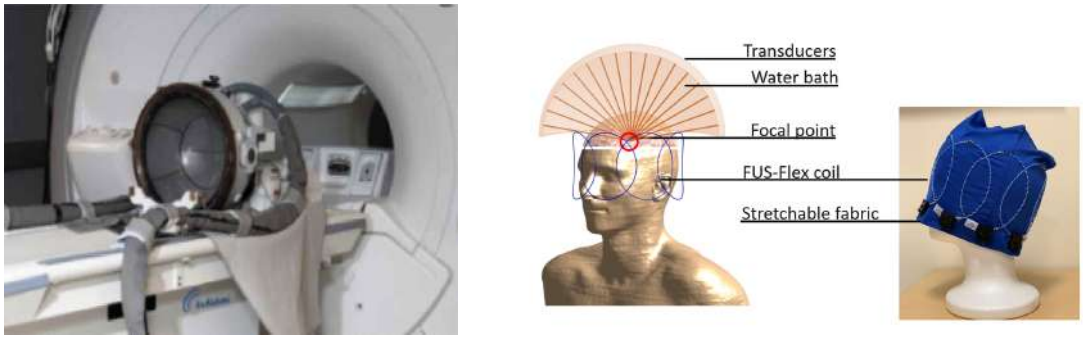
3.1.2 TcMRgFUS treatment for ET patients

As we have already seen, MRI is a radiological technique able to visualize a wide range of anatomical characteristics. In recent years, it has also found applications in neurosurgical treatment. On the one hand, it is an extraordinary technique capable of providing highly valuable information for treatment planning; in this context, fiber tracking represents an example. On the other hand, it has also become a guiding tool for supporting surgery and monitoring patients *in vivo*. This is the case for tcMRgFUS treatment, in which MRI assists both in targeting and in visualizing the biological effects of focused ultrasound beams.

TcMRgFUS is a non-invasive neurosurgical technique aimed at inducing thermal ablation within brain tissue by exploiting focused ultrasound beams. The use of ultrasound for therapeutic purposes dates back to the early 1940s, when Lynn and Putnam first performed targeted, well-demarcated, and controlled ablations in cortical and subcortical areas of mammals [69, 70]. However, at that time, reaching subcortical areas required craniotomy. With the development of devices consisting of multiple phased-array transducers, it became possible to deliver sufficient thermal energy to a small focal point through the intact skull, directly targeting deep brain structures and inducing controlled thermal ablation [70, 71]. The combination with MRI, exploiting protocols capable of performing thermometry, has made tcMRgFUS brain ablation a more accurate and safer technique [70, 72].

Ultrasound waves are generated by piezoelectric crystals (transducers), which convert electrical pulses into high-frequency vibrations, typically up to several hundred kHz [71]. A large number of transducers (typically around 1000 or more) are arranged in a hemispherical structure—forming the so-called helmet—which is placed over the patient's

head. To minimize wave dispersion, the patient's scalp must be shaved. Moreover, the helmet is filled with degassed, circulating, cooled water, which serves both as an acoustic coupling medium and for tissue cooling. The entire system is fixed using a stereotactic frame and is built to be MR compatible. Graphial representation of tcMRgFUS system are dipicted in figures 3.3.

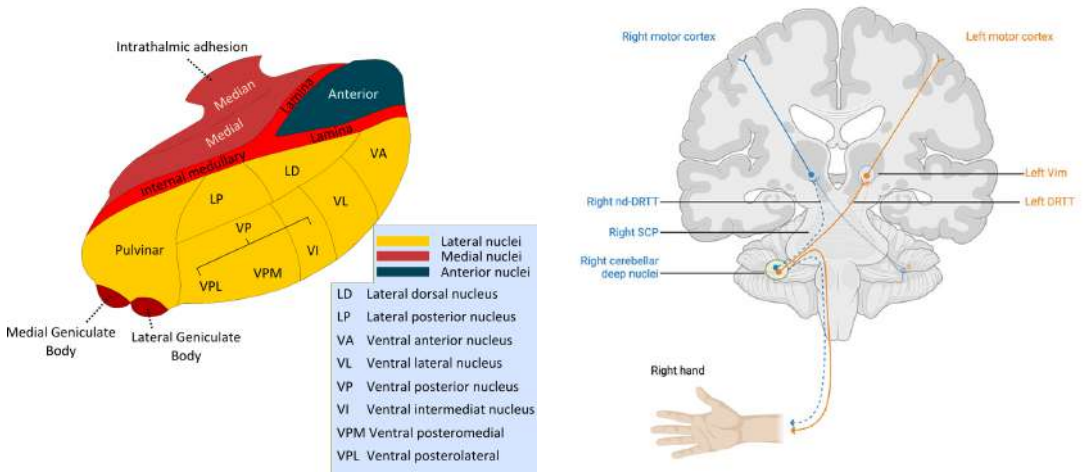


(a) (b)

Figure 3.3: TcMRgFUS systems. a) Commercial example: Exablate 4000, Insightec Ltd. b) System scheme: transducers produce ultrasound; the semi-circular shape allows for focusing; the water bath improves coupling with tissues and acts as a cooler; the FUS-Flex coil is used as a local MR coil to improve imaging.

Essential Tremor and the role of the Ventral Intermediate Nucleus in its symptoms

TcMRgFUS has gained particular interest in Essential Tremor (ET) treatment, the most common movement disorder in adults. It significantly impacts daily activities due to involuntary trembling of the hands, head, voice, or limbs, affecting quality of life [12,13]. A potential cause is abnormal activity in the central tremor network, though its exact pathogenesis remains unclear [14]. Treatment depends on symptom severity and patient needs, ranging from behavioral therapy to medications and surgery [73]. When medication fails, surgical intervention such as thalamotomy can help patients. In literature the ventral intermediate nucleus (VIM) is indicated as target to suppress tremor [15]. The VIM is a thalamic subregion, known as a nucleus (Figure 3.4a). Its dimensions are very small, approximately $4 \times 4 \times 6$ mm^3 [74]; for this reason, its precise targeting is challenging. The VIM is part of the motor control network and acts as a hub for sensorimotor integration [75, 76]. It is also part of the Dentato-Rubro-Thalamic tract (DRTT), a tract that originates in the dentate nucleus, follows the ipsilateral superior cerebellar peduncle, and later decussates, reaching the contralateral red nucleus and the contralateral thalamus [77] (Figure 3.4b).
In latest years, several techniques have been proposed. Although Deep Brain Stimulation—a highly invasive technique that consists of implanting a pacemaker-like device within a brain region [79, 80]—is considered the gold standard in ET care, tcMRgFUS [81] has emerged as a non-invasive and safe alternative. However, regardless on the surgery technique, treatment planning is mandatory in both cases.



(a) (b)

Figure 3.4: a) Thalamus parcellization: subdivision of the thalamus into regions called nuclei. The VIM is part of the lateral nuclei, indicated here as VI. b) Representation of the DRTT (image adapted from reference [78]).

In tcMRgFUS, the exact location of the VIM is identified through direct patient feedback. During an MRI session, with the patient awake, the neuroradiologist explores thalamic regions by inducing temporary lesions. When the VIM is located and stimulated with a low-intensity pulse, ET symptoms cease, allowing the doctor to perform permanent ablation. In this context, treatment success depends on the duration of the procedure, as the patient—who experiences tremor—is required to remain immobile inside the MRI scanner. Consequently, the high-stress conditions may lead the patient to request interruption of the treatment. For this reason, it is important to reduce the region to be explored as much as possible. Ideally, the VIM position should be identified *a priori*; however, its localization remains an open issue [82], mainly due to limitations in image resolution. In fact, it must be underlined that the VIM is not visible in routine clinical imaging acquisitions.

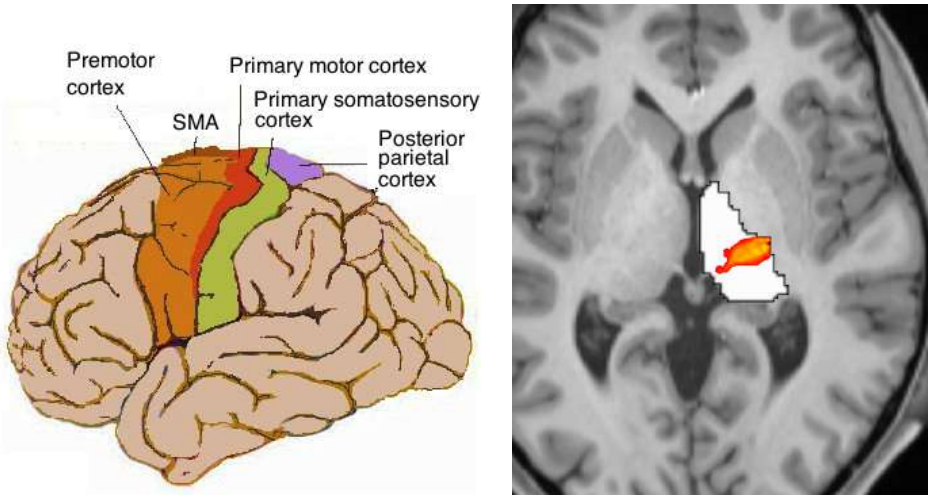
Currently, the VIM position is estimated using atlas-based approaches. Atlases are detailed brain maps derived from *post mortem* studies. The method consists of building labeled brain templates and finding the best affine or nonlinear transformation that aligns the patient's brain image with the template. Once this transformation is obtained, its inverse is applied to the atlas to map brain ROIs into the *patient's space*[3]. As an example, Thalamus Optimized Multi Atlas Segmentation (THOMAS) [83] has been proposed as a thalamus-specific atlas for VIM identification.

Atlas-based methods offer significant advantages; however, they are generally based on the anatomy of only a few subjects. Their main limitation is the loss of patient-specific features.

As a patient-specific technique, tractography is capable of highlighting a region that contains the VIM with high probability. The simplest way to obtain such a region is by evaluating the connectivity map within thalamus with the *Primary Motor Cortex (M1)*

[3]The representation space of the original patient image.

(figure 3.5).



(a) (b)

Figure 3.5: a) Topography of human motor cortex. Pancratderivative work: Iamozy - Own work, This file was derived from: Cortex sensorimoteur1.jpg:, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=33981076`.b) M1-thalamus connectivity map obtained by probabilistic tractography. Tractography was performed by setting the left thalamus as the seed region and the left M1 cortex as the target region. Ventricles were used as exclusion (avoidance) regions. The connectivity map was obtained by counting the number of times each voxel contributed to a streamline ending in the target region. Figure adapted from reference [84].

Although VIM is much smaller than the region here highlighted this is one of the most used approach, originating from the work proposed by Behrens in reference [85], which will be discussed in the next section.

3.2 General tractography pipeline for tcMRgFUS treatment planning

The VIM region estimation through probabilistic tractography requires multiple steps performed using *FreeSurfer* [86] and *FSL* tools [87–89] and python code suitably developed using the numpy, nibabel and scikit-learn libraries [90, 91].

First, MRI data must be converted in NIfTI format. Subsequently, T_1 and DWI images are processed separately. Results coming from both processing steps are then combined to run the PROBTRACKX routine, as shown in Figure 3.6.

3.2.1 T1 preprocessing

T1 images are essential for generating binary masks of brain regions requested for probabilistic tractography. Prior to brain region extraction, preprocessing can be conducted using the *Freesurfer* software. The complete pipeline, executed via the `recon-all` command, involves noise and motion correction, intensity normalization, and labeling of both cortical and subcortical brain regions. Subsequently, binary masks

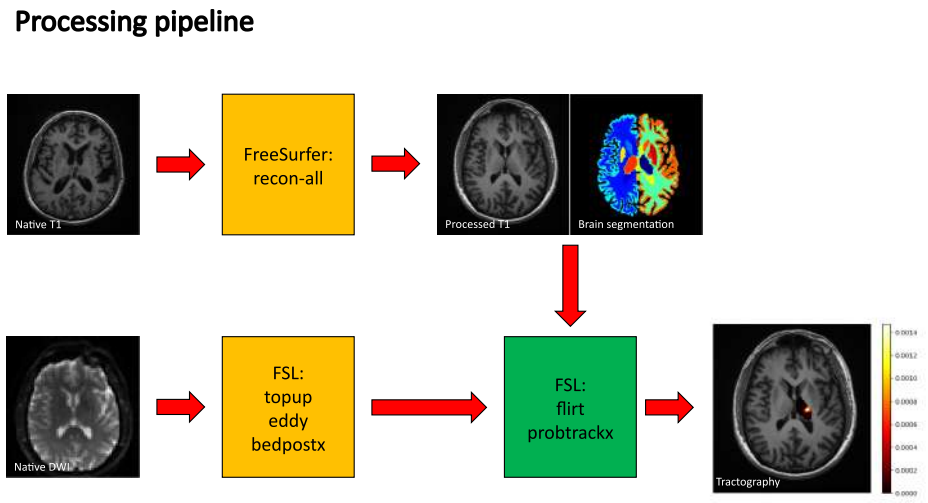


Figure 3.6: Schematic representation of the probabilistic tractography pipeline.

for both cortical and subcortical regions can be derived using additional FreeSurfer commands: `mri_annotation2label`, `mri_label2vol`, and `mri_binarize`, this last utilizing the segmentation files produced by `recon-all` to create binary masks. Lastly, given FreeSurfer's unique data representation, a data reorientation to MRI standard can be performed using the FSL command `fslreorient2std`.

3.2.2 DTI preprocessing

Since DTI images are prone to noise and artifacts, DTI preprocessing is mandatory. *FSL* tools (`topup`, `eddy`) are employed for motion and eddy current corrections, as well as noise reduction. After these first steps, the Orientation Fiber Distribution (OFD) at each voxel was estimated using `BEDPOSTX` (Bayesian Estimation of Diffusion Parameters Obtained using Sampling Techniques, where X stands for modelling Crossing Fibres, [92]).

3.2.3 Probabilistic tractography and connectivity maps

Probabilistic tractography can be performed using `probtrackx2_GPU` (the GPU version of `probtracx`, [85, 93]). This tool produces sample streamlines, drawing an orientation from the voxel-wise BEDPOSTX distributions and taking a step in this direction. In other words, these streamlines follow a random walk exploiting the information of the water diffusion tensor for each voxel.
The process starts from a seed region and ends when a termination criterion is reached. A termination criterion consists in reaching a target area, e.g. a cortical region, or arresting the process when the streamline ends in an avoided region, e.g. brain ventricles. Additionally, a region can be selected as a way-point, filtering out unuseful streamlines.

This pipeline was first proposed by Behrens in [92], with the main aim of thalamic parcellation based on its connectivity with cortical areas. Connectivity maps can be obtained by counting the number of times each voxel contributes to a streamline ending in a specific target region. The results can be interpreted as *the probability of a voxel being connected to a given cortical area* [92]. Since this approach produces a connectivity map for each target area, a segmentation can be performed by applying a maximum probability criterion, assigning each voxel to one, and only one, cortical area. By using the thalamus as the seed region, cortical regions as targets, and the ventricles as exclusion regions, thalamic parcellation can be achieved. Figure 3.7 illustrates thalamic parcellation through probabilistic tractography (adapted from [92]).

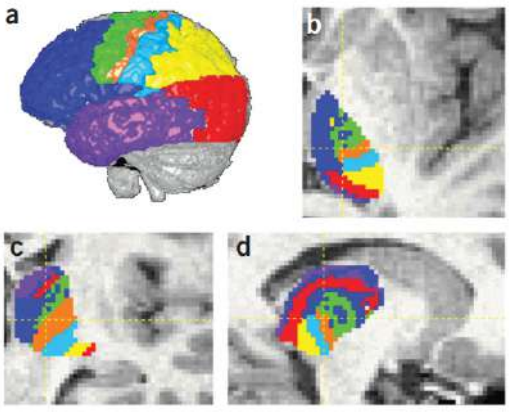


Figure 3.7: Connectivity-based segmentation of the thalamus. a) Division of the cortex according to anatomical landmarks after cortex. (b-d) Classification of thalamic voxel based on prabable connections to these cortical areas. Figure adapted from reference [92].

3.2.4 Probabilistic tractography timing

Performing the probabilistic tractography pipeline for thalamus parcellation is computationally demanding. On a Dell workstation (Intel Xeon Platinum 8268 CPU @ 2.90GHz, 48 cores, 128GB RAM, NVIDIA RTX A5000 GPU), the total processing time was approximately 2.42 hours for a single subject. See table 3.1 for a detailed time analysis.

Clinical data acquired on a 1.5T MRI scanner	
Standard pipeline	
Processing description	**Time (HH:MM:SS)**
FREESURFER	02:00:00
Post-FREESURFER	00:05:00
DWI preprocessing	00:20:00
BEDPOSTX	00:05:00
DWI to T1 Registration	00:05:00
PROBTRACKX	00:02:00
ACPC registration	00:05:00
Total	**02:42:00**

Table 3.1: Computational time needed for the entire flowchart for probabilistic tractography clinical data acquired on a 1.5T MRI scanner, using a Dell workstation with Intel Xeon(R) Platinum 8268 CPU @ 2.90GHz, 48 cores, 128GB of RAM, NVIDIA RTX A5000 GPU with 24GB of memory.

3.3 DeLTA-BIT version 1

In the previous sections, it has been shown that probabilistic tractography can be used to label the thalamus into its subregions, each projecting bundles of fibers that connect the thalamus to various cortical brain regions, thus providing valuable non-invasive in vivo thalamic parcellation [94–96]. This approach can be particularly useful in tcMRgFUS treatment planning, since the VIM nucleus receives fibers from the cerebellum and is mainly connected to the primary motor cortex [15, 97, 98]. However, the probabilistic tractography approach is very time-consuming, requiring computational analysis times of many hours per patient, and the analysis pipeline is not physician-friendly. From Table 3.1, it is clear that most of the required time arises from T1 image preprocessing. It should be noted that approximately 2 hours are required for the full "recon-all" analysis, although several steps can be skipped. Moreover, it is important to note that this time estimation was obtained using a high-performance workstation with GPU acceleration, which significantly speeds up computations. In general, hospitals do not have access to such computational resources. Access to dedicated workstations is limited, and the availability of high-performance GPUs is even more rare. For these reasons, online treatment re-planning remains impractical.

Nowadays, use of ML and DL techniques could help to tackle complex problems, also in medical field [99], exploiting a data-driven approach rather than model-based algorithms.

Recently, white matter tractography tasks were performed through DL approaches using Diffusion Tensor Imaging (DTI) scans–imaging of the diffusion tensor within brain tissue derived from DWI images–as input and directly providing white matter tractography streamlines as output.

First applications were aimed at finding deterministic tractography maps. In particular, Neher et al. adopting a random forest classifier that uses raw diffusion data was able to provide direction proposals for streamline tracking [100]. Successively, DL

models for fiber tractography making use of a fully-connected and a recurrent neural network architectures [101, 102] were developed as a regression task to carry out deterministic tractography providing the streamline directions in each tracking step. Another approach followed is the estimation of the fODFs by means of DL networks and using these results to reconstruct both deterministic and probabilistic tractograms. Deep CNNs were used for obtaining discrete fODFs [103] as well as for predicting spherical harmonics coefficients for continuous fODF estimation [104]. Local fiber orientations along tractography streamlines were also predicted by means of recurrent neural networks by solving a multi-class classification problem instead of the conventional regression approach. [105]. Other DL models (including CNN, fully CNN or CNN + Transformer,etc) developed for fiber tracking are reported in literature [106–111]. Recently, a general framework for tractography with deep reinforcement learning was proposed [108].

In this study, I propose DeLTA-BIT (acronym of Deep-learning Local TrActography for BraIn Targeting), a novel open-source DL framework aimed at both reconstructing probabilistic tractography maps inside the thalamus and realizing thalamic parcellation starting from diffusion-weighted MR data. This model is able to provide the connectivity maps, as those obtainable with the pipeline described in section 3.2, between each voxel inside the thalamus and any user-specified cortical or subcortical masks. Our models were trained for different cortical regions and more than 1,000 subject from HCP database [67, 68] were considered for training and validation. As it will be shown in next section, in DeLTA-BIT version 1 validation concerns only the model capability in reproducing connectivity map deriving from tractography, for this reason it can be seen only as an internal validation.
In the next sections, materials and methods will be presented, followed by the results. In the discussion section, the limitations will be addressed and the rationale for the need for an improved version will be introduced.

3.3.1 Materials and Methods

Dataset

The working dataset for designing the DL models should be composed of a large set of DWI acquisitions whose DTI images can be extracted from. In addiction, also T_1 images are needed to extract binary masks for optimal identification and segmentation of the different brain regions of interest. Moreover, since this task is supervised, the probabilistic tractography maps should be used as labeled images for ground truth of the model. The dataset chosen that satisfies some of these requests is the HCP dataset [68, 112–114], with approximately 1200 young adult subjects. For each subject considered in this work only T_1 structural images, DWI images as well as results of BEDPOST analyses were downloaded. Each subject image subset underwent to the

pipeline presented in section 3.2. Estimation of connectivity produces a number of maps equal to the number of selected target regions. In our case, these are cortical regions; specifically, the frontal, occipital, parietal, post-central, pre-central (M1), and temporal cortices were used as targets.

To build our dataset to use fo DL training subjects with anomaly data, such as missing data or different acquisition protocols, were discarded. The final size of the dataset is 1064 subjects, whose 800 were included in the training set and 264 in the test set.

Bounding box

Before training the CNNs an analysis about computational complexity was performed. A statistical study about thalamus physical features, such as volume and surface, showed that the thalamus of a hemisphere has a mean volume of about 8600 mm^3, i.e. approximately 0.05% of a standard MRI image (typically in size $256 \times 256 \times 256$ mm^3). For this reason a bounding box was chosen. Since all HCP images were registered on the MNI152 standard, they share an equal frame of reference. To find the voxel coordinates of the bounding box, a hundred of subjects were randomly chosen and their thalamus binary masks were loaded; after this the smallest box which contains the largest thalamus among all chosen subjects was selected. Ten voxels of tolerance for each directions were added and then it was checked that all thalamus data lie inside the box. A bounding box of $38 \times 60 \times 48$ voxels (Figure 3.8) was identified reducing the computational complexity to train CNNs.

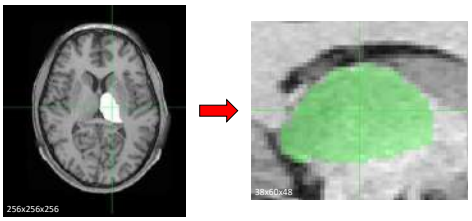


Figure 3.8: Bounding box containing the left side hemisphere thalamus of all dataset subjects. This allows to work with input size of $\mathbf{38 \times 60 \times 48}$ voxels.

AI models

In order to reduce the time required for probabilistic tractography, we developed two types of CNNs. Looking at figure 3.6, the idea is to use a CNN for brain segmentation and several CNN for direct tractography prediction. In the following we refer to the CNN which works on brain segmentations as the segmentation model, whereas the models which predict tractographies are referred to as the regression models. We have trained one segmentation model for thalamus segmentation, and six regression models

(one for each cortical region considered). In this way the computational time required for tractography evaluation can be strongly reduced. However, a minimal preprocessing is still necessary: registration between structural and DWI images and DWI artifacts corrections are needed.
The network architecture used for each model is a 3D U-Net (Figure 3.9), the truly difference between them depend only on the input and output layers.
The network input comes projected through the network to produce the output. The network is subdivided in levels, for each of them our model has two convolutional blocks. A convolutional block is made by a 3D convolutional layer with kernel size equal to 3, a LeakyRelu activation layer ($\alpha = 0.01$) and a Batch Normalization layer.
In the network coding part (left side of the network in figure 3.9), at every level except the first, convolutional blocks reduce the spatial size of the layer inputs and add new feature maps. In a typical U-Net the rule is: at each level halve the input spatial sizes and add twice as many feature maps. This goes on until the fourth level, where the layer input becomes really small in spatial size, $5 \times 8 \times 5$ voxels, but it has a large number of feature maps, 128. At this stage the network has done the encoding process and now is ready for decoding (right side of the network in figure 3.9). In the decoding process the network does exactly the opposite: it increases the layer input in spatial size and reduces the number of feature maps by *3D Transposed convolutional layers*, until the layer input gets the same spatial dimensions as the network input. This is the reason why in the first level we add feature maps but we keep the convolutional stride of the first block equal to 1.
Networks were implemented using the Python's library *Keras* [115].

Training models

The segmentation model was trained using the native T_1 images as inputs and the binary masks produced by *FreeSurfer* [86] as targets. The images were cut into the above defined bounding box.
The regression models have been trained using same inputs but different connectivity maps as target. This choice is justified by assumption that the tractographies are indipendent each other. The inputs of the regression models consist of 11 3D volumes arranged to form a 4D array of size $38 \times 60 \times 48 \times 11$, where each volume has the size of the bounding box. The first volume is the binary mask of the left side hemisphere thalamus and the second one is FA image. The other volumes are obtained multiplying each components of eigenvector of the diffusion tensor by its eigenvalue; so, since eigenvectors have 3 components, 9 3D volumes were obtained. This choice is based on the physical consideration that the most of the information about probabilistic tractography derives from the values of diffusion tensor inside thalamus, i.e. from its eigenvectors and eigenvalues in this region. Of course, under this assumption an approximate version of the probabilistic tractography can be obtained.

In general, each of the developed network has a total number of parameters equal to $1,466,433$, of which $1,465,025$ are trainable and randomly initialized (using a normal distribution). They have been trained using *Adam algorithm* as optimizer. The loss functions we use are the *combo loss* (the sum of the crossentropy and the dice loss) for the segmentation task, and the *Mean squared error* (MSE) for regression models. Since an enough number of subjects to train the network was available, a single validation split was carried out to monitor the training of each models, so for all of them the training set became of 704 subjects in size and a validation set of 96 subjects (randomly selected) was used. In addiction, as the training set is still large in size and requires a very large amount of memory, we used a data generator that provides the networks with input batch of 16 subjects (batch size) for regression models and 64 for the segmentation one. The data generator does also other operations: indeed, in this case it was adopted for implementing data normalization, online data augmentation and random shuffling at the end of each epoch. The network needs data normalization to have inputs taking values between 0 and 1. This is typical procedure in DL because it facilitates the training process. Data augmentation was applied to improve model generalizability and to guarantee that a model is independent for some data transformation. The transformations we used for data augmentation are: small angle rotation, translation, flipping, scaling and adding of random Gaussian noise. When an input is loaded a transformation is chosen randomly (transformation and its parameters) and is applied to it, so at each epoch the network sees different data. This type of data augmentation is called *online data augmentation*. Lastly, at the end of each epoch the data generator applies a random shuffling on the data list, to avoid that the network always sees the same sequence of data which can cause overfitting or underfitting problems.
During the training phase a `model checkpoint` was exploited and this saves a copy of the model for each iteration the model with the best performances on the validation set. Analysis of the training curve allows to select the best model being careful not to choose an overfitted model.
The hardware we used to train the network is a `NVIDIA RTX A5000` GPU with $24GB$ of memory. We set 2000 epochs for each tractograpy, which takes about of 50 hours to complete the training of one regression model and 11 hours to train the segmentation model.

The AI pipeline

The AI pipeline is different from the original one, although DWI pre-processing is still mandatory. In order to obtain a set of tractographies we need first to register structural and DWI images on the MNI152 standard. Then the thalamus mask is predicted using the segmentation model and 11-volume inputs are constructed for tractographies predictions. The time required is about of 3-4 minutes.

The code of these models is freely available on Github at the link `https://github.`

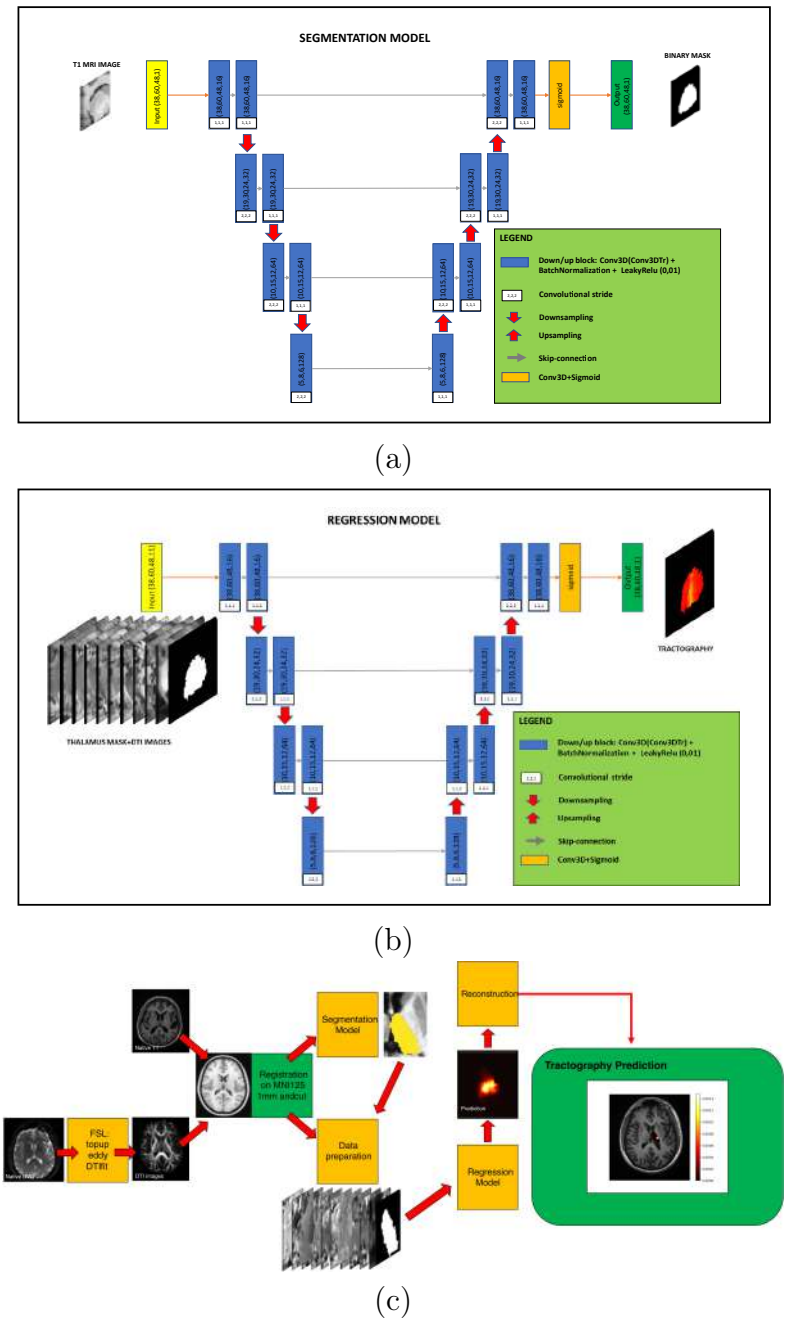


Figure 3.9: Schematic representation of the segmentation network (a) and the regression network (b) and the new pipeline with AI models. The regression network is a four levels 3D U-Net architecture. The network has a symmetric shape and it can be seen as an encoder-decoder: on the left side there is the encoder part and on the right side there is the decoder part. The network is subdivided in levels, for each of them our model has two convolutional blocks. A convolutional block is made by a 3D convolutional layer with kernel size equal to 3, a LeakyRelu activation layer and a Batch Normalization layer. The numbers reported inside blocks, in parentheses, refer to the size of the block outputs: from bottom, the first three numbers refer to the spatial size and the last one to the number of feature maps. The numbers on bottom of blocks refer to the convolutional stride, when they are 1 the layer output maintain the same spatial size when they are 2 the layer output is spatially halved. For each level the first convolutional block has a convolutional stride equal to 2, except for the first one, this reduces the size of the input by half for all directions. The compaction is offset by a double number of feature maps. The output layer is made by a convolutional block with sigmoid activation function, this gives an output with values between 0 and 1.

`com/mromeo1992/delta-BIT`.

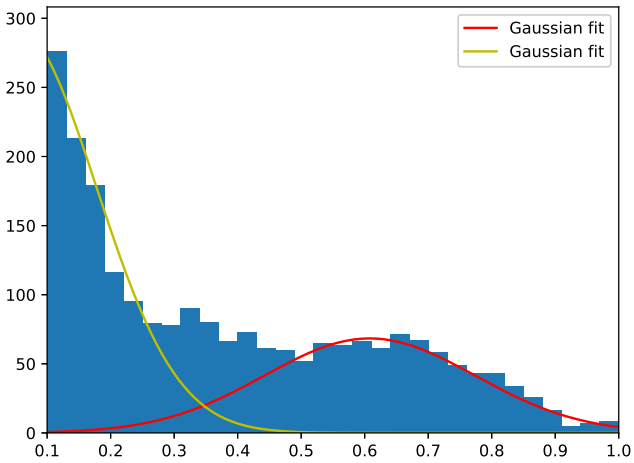


Figure 3.10: Histogram of gray level of a tractography: a bimodal distribution is evident and the background and foreground voxels can be distinguished according to their intensity values.

Evaluation methods

Segmentation model The predition ability of the segmentation model have been evaluated on the test set. The metric we used is the Similarity Dice Coeffiient (DSC).

Regression models MSE was solely chosen to train the network and, although it has many advantages, it should be used carefully for a proper evaluation (after training in models validation). Indeed, probabilistic tractographies have many features and a single value, such as MSE, is not able to summarize them. When the MSE is calculated, first the absolute error for each voxel is evaluated, and then it is averaged over the whole image and at the end averaged over all subjects. In such a way the information about where the model is mismatching the desired output is lost. For this reason, two different approaches were employed to evaluate the models. The first is based on the distance between the centers of gravity calculated from the ground truth and predicted tractographies. This evaluation considered both the Euclidean distance and the absolute distances along each axis to account for potential directional bias.
The second method exploits the bimodal distribution of gray levels typical of probabilistic tractographies (see Figure 3.10). Based on this observation, thresholding algorithms can be used to define ROIs, from which various discrete metrics can be estimated.
In TcMRgFUS treatment planning, neuroradiologists typically apply a threshold to the tractography to identify a Region of Interest (ROI). Inspired by this clinical practice, and considering that each probabilistic tractography exhibits a bimodal intensity distribution, voxels can be classified as background or foreground based on their intensity values. This method requires a threshold value, which in this study was determined using Otsu's algorithm [53]. Once the ROIs were identified DSC were calculated.

3.3.2 Results

In this section the results obtained through the developed networks are presented and the prediction abilities of these models are quantified and discussed. First, the predictive capabilities of the segmentation model (thalamus ROI) and the precentral cortex region model (containing the VIM) are illustrated. Figure 3.11 shows an example comparison between the connectivity maps obtained through FSL tools and that predicted by the proposed model. Although some differences can be noted, these results can be obtained in approximately **8 minutes** on clinical images (T1 1mm isotropic, DWI 3mm isotropic with 64 gradients, both acquired using a 1.5T scanner, see section 3.4).
Subsequently, we present the results for models developed and trained for other cortical regions—specifically the frontal, occipital, and parietal cortices, the postcentral gyrus, and the temporal cortex (graphical representations are shown in Figure 3.12). Finally, the results of the thalamus parcellation analysis are reported.

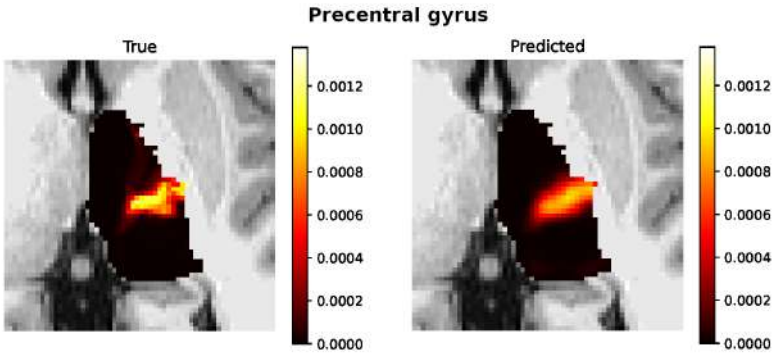


Figure 3.11: Comparison of predicted (DL approach) and true probabilistic tractography maps (from PROBTRACTX pipeline) representing thalamic connections to the precentral gyrus.

The segmentation model

The segmentation model has to find the binary mask of the thalamus from T_1 images. This is the first step of AI pipeline and, consequently, all successive results depend on its prediction abilities. This model is able to reach a mean DSC about of 0.93 on the test set, with a standard deviation of about 0.01 (figure 3.13), which represents a very good performance with small data dispersion.

Precentral gyrus model

Even if the pathophysiology of essential tremor is nowadays only partially understood, the presumed neuronal pathways involved are not the subject of controversy and two main circuits implicated in tremor generation ("tremor networks") are recognised: the cortico–ponto–cerebello–thalamo–cortical loop and the Guillain–Mollaret triangle (dentate nucleus to red nucleus to inferior olivary nucleus to dentate nucleus circuit) [116, 117]. For this reason the VIM nucleus is today considered one of the possible

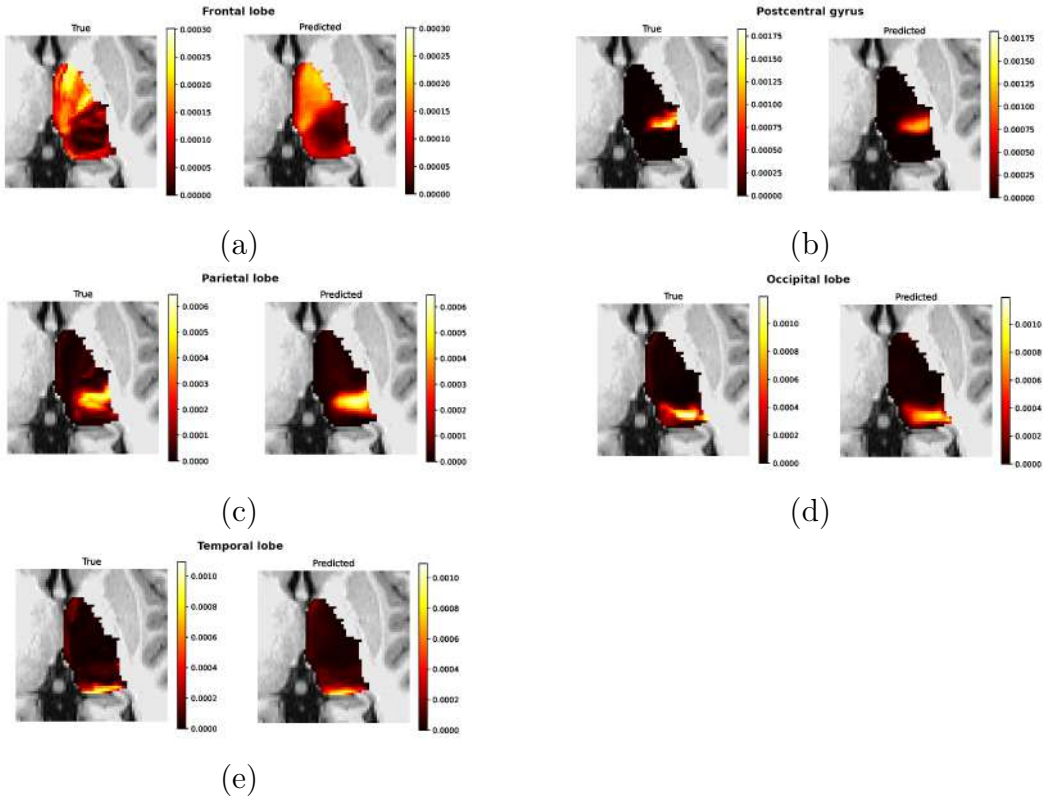


Figure 3.12: Comparison of predicted (DL approach) and true tractography maps (from PROBTRACTX pipeline) for various target brain regions: a) frontal lobe (without the precentral gyrus), b) postcentral gyrus, c) parietal lobe (without the postcentral gyrus), d) occipital lobe, e) temporal lobe.

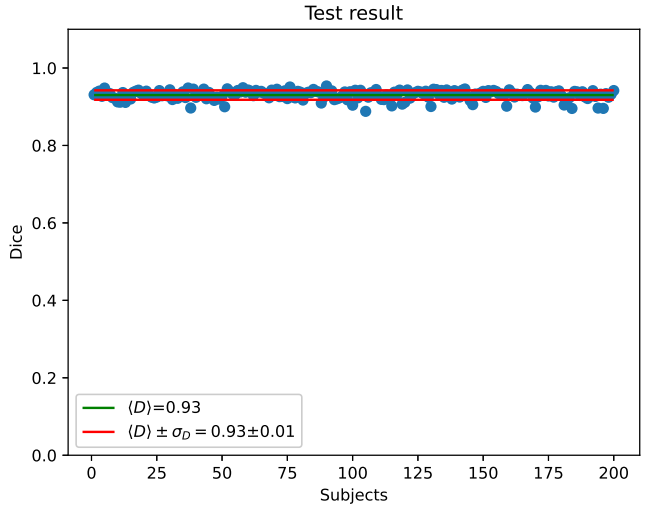


Figure 3.13: Dice similarity coefficient between the predicted thalamus and the binary mask produced by *FreeSurfer*. Average score is 0.93 ± 0.01.

targets to interrupt the cortical hyperexcitable state that results from a reduction in the cerbellar GABAergic tone. By the use of probabilistic tractography is it possibile to identify the area of the thalamus that is mostly connected with the precentral gyrus (i.e. primary motor cortex) of the frontal lobe thus the presumed anatomical location of the VIM. So, we focus in this tractography map due to its importance for treatment planning for neurofunctional disorders as ET.

Centre of gravity Figure 3.14 shows the coordinates of the predicted center of gravity are plotted as a function of the real ones (in millimeters). It is evident that the most of the scatter points stay within the region delineated by straight lines with slope equal to 1 and with y-intercept equal to ± 1 *mm*. This result means that if the predicted distributions were used rather than the real ones, the error on the center of mass would be equal to ± 1 mm (which is equal to the image resolution). In fact, by calculating the average euclidean distance between the two centers of gravity, we have obtained 0.3 ± 0.5 mm. We calculated the Pearson's correlation coefficient for each coordinates and we found 0.92 along the x-direction, 0.94 along the y-direction and 0.99 along the z-direction.

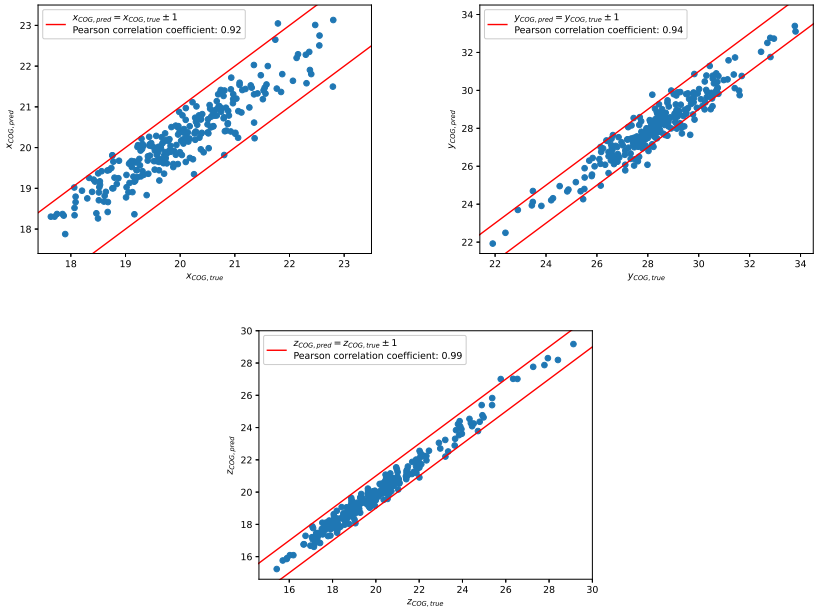


Figure 3.14: Coordinates of the centre of gravity (in millimeters) for the true and the predicted tractography maps.

Another way to evaluate the distances between the center of gravity for the true and predicted distributions is shown in Figure 3.15. The distances along the three axes are centered around a zero-value with a standard deviation smaller than 1 mm.

Dice Similarity Coefficient For each subject, a ROI was derived from the ground truth tractography and another from the predicted tractography using Otsu's algorithm. Subsequently, the DSC between them was calculated using Equation 1.4. The results are summarized in Figure 3.16.

The high DSC value (i.e., 0.80 ± 0.06) highlights that, on average, the high-probability regions of the predicted tractography maps significantly overlap with the corresponding regions in the ground truth. This indicates a strong spatial consistency across subjects.

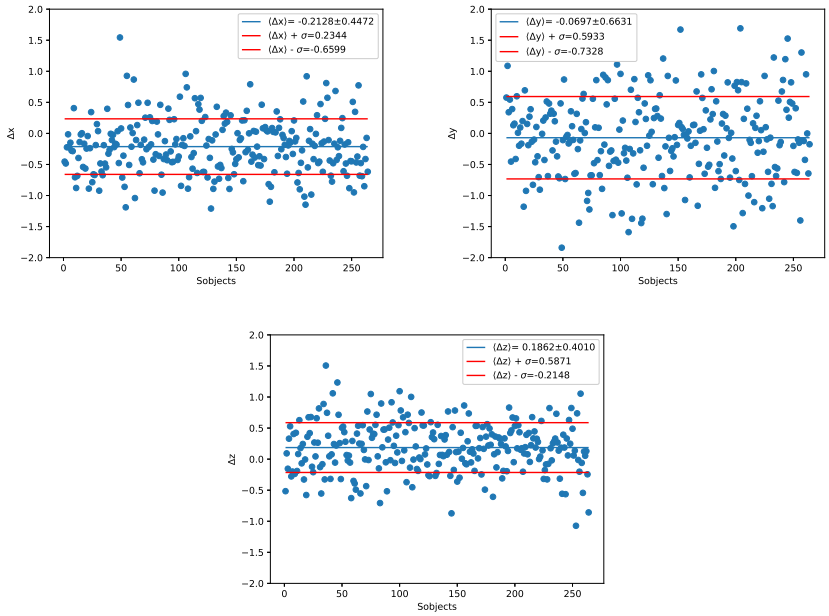


Figure 3.15: Absolute distance, along each direction, between the centre of gravity of the true and the predicted tractography.

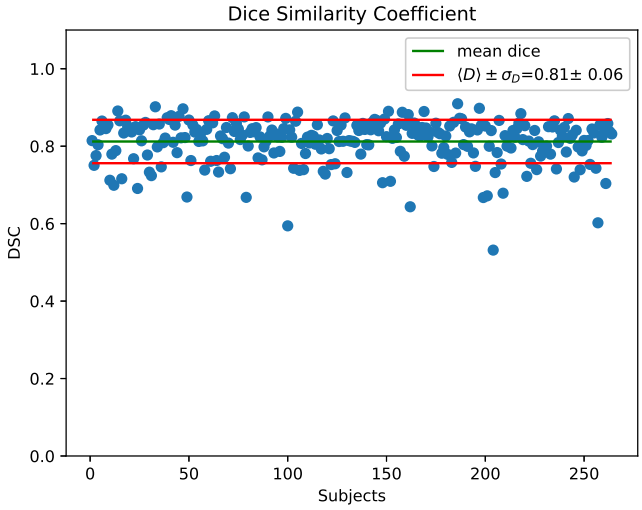


Figure 3.16: Dice similarity index between predicted and true ROIs. Average score is about of 0.80 with a standard deviation of 0.06.

Other cortex regions

DL models, analogous to the one developed for the precentral gyrus, were trained and validated for other cortical regions, including the frontal, occipital, parietal, and temporal cortices, as well as the postcentral gyrus. The results for these regions are plotted in Figure 3.12 and summarized in Table 3.2.

All models produced predicted tractography maps with centers of gravity that were, on average, within 1 mm of the ground truth (with a standard deviation smaller than or comparable to the spatial resolution of the images). Furthermore, the DSI was found to be high (≥ 0.70) for most regions. The region with the lowest DSI value (≈ 0.5) was

associated with the temporal cortex; this result can be explained by the fact that the temporal ROI is significantly smaller than those of the other regions. Consequently, even small fluctuations in a few voxels can significantly impact the overlap metrics.

Brain Region	Δx(mm)	Δy (mm)	Δz (mm)	Δr(mm)	DSC
Frontal	0.4 ± 0.5	0.4 ± 0.6	0.2 ± 0.2	0.6 ± 0.8	0.85 ± 0.03
Occipital	-0.1 ± 0.7	0.3 ± 1.0	0.02 ± 0.7	0.3 ± 1.2	0.70 ± 0.09
Parietal	-0.2 ± 0.4	0.6 ± 0.8	-0.1 ± 0.4	0.5 ± 0.9	0.81 ± 0.05
Postcentral	-0.4 ± 0.4	0.4 ± 0.6	0.2 ± 0.6	0.6 ± 0.9	0.77 ± 0.06
Precentral	-0.2 ± 0.5	-0.07 ± 0.7	0.2 ± 0.4	0.3 ± 0.8	0.80 ± 0.06
Temporal	-0.06 ± 0.800	0.4 ± 1.0	0.3 ± 0.5	0.5 ± 1.2	0.5 ± 0.2

Table 3.2: Models performance over several metrics.

Thalamus parcellation

Model performance was also evaluated based on the capability to emulate thalamic parcellation, as described in Section 3.2.3. The overlap between the ground truth (GT) and predicted ROIs was quantified using the DSI. These results are reported in Table 3.3.

	Frontal	Occipital	Parietal	Postcentral	Precentral	Temporal
DSC	0.90 ± 0.03	0.70 ± 0.06	0.80 ± 0.05	0.60 ± 0.15	0.70 ± 0.1	0.60 ± 0.08

Table 3.3: Dice Similarity Coefficient (DSC) of different thalamus areas.

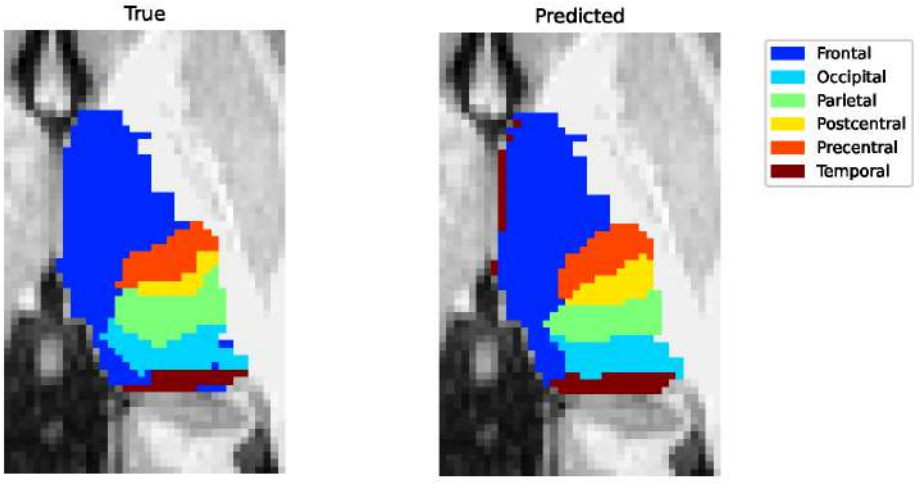


Figure 3.17: Comparison between thalamic parcellation obtained from the true (PROBTRACKTX pipeline) and the predicted tractography (DL approach).

3.3.3 Discussion

The tractography pipeline described in Section 3.2 and summarized in Figure 3.6 has been shown to be effective for thalamic parcellation and to provide useful information for VIM targeting in treatments such as tcMRgFUS or DBS [95]. Despite these advantages—which remain relevant for treatment planning—probabilistic tractography cannot be performed during interventions due to its time requirements (over 2 hours for the full pipeline) [85, 94]. Although several steps can be skipped to enable faster analysis, DWI preprocessing still requires considerable time. Moreover, the analysis becomes feasible only with access to high-performance workstations, a requirement that is not easily satisfied in clinical settings.

DeLTA-BIT, the AI pipeline proposed here, aims to overcome these limitations. As previously shown, the thalamic region primarily connected to the primary motor cortex (M1, precentral gyrus cortex) is associated with the VIM nucleus. Specifically, the connectivity map with the precentral cortex highlights a ROI that contains the VIM. DeLTA-BIT is designed to predict this region. It is based on a segmentation model—aimed at predicting the thalamus mask—and on regression models—aimed at estimating connectivity maps. The model developed for thalamus segmentation from T_1-weighted structural images achieved a Dice coefficient of approximately 0.91, which is comparable to the state of the art in thalamus segmentation [118, 119]. The results on precentral gyrus connetivity map (section 3.3.2) show good agreement between the M1 connectivity ROI and the one predicted using DeLTA-BIT ($DSC = 0.81 \pm 0.06$, with a center-of-mass distance of less than one millimeter).

Additionally, the pipeline was designed to be adaptable to any tractography-based segmentation. As proof of concept, DeLTA-BIT was trained to estimate connectivity maps with other cortical regions, showing good agreement. Finally, it was demonstrated that applying thalamic parcellation to DeLTA-BIT outputs, and comparing it with that obtained using the probabilistic tractography pipeline, yields a DSI higher than 0.6 for all ROIs.

Although promising results have been obtained, DeLTA-BIT presents some issues and limitations. First, it was designed with the primary goal of assisting in tcMRgFUS treatment re-planning during the intervention. While it aims to predict the VIM location, the ROI highlighted by the M1 connectivity map is broader than the VIM alone. Indeed, the VIM is reported in the literature to have dimensions of approximately $4 \times 4 \times 6\ mm^3$, whereas the M1 connectivity map identifies a volume of about 400 mm^3. Consequently, its prediction is insufficient to provide exact targeting information.

Secondly, the model was trained and validated exclusively on HCP data. The HCP dataset is renowned for its high-quality MRI imagery, including both T1-weighted and DWI scans. Specifically, its DWI images were acquired using a protocol with over 200 gradients—enabling highly precise tractography—whereas typical clinical protocols involve no more than 64 gradients. Furthermore, the HCP dataset consists solely

of healthy young adults, while clinical subjects are generally older and affected by pathologies, such as Essential Tremor. Both the superior data quality and the specific subject demographics introduce a significant bias into the developed models, which persists despite the application of data augmentation during training. Equally important, DeLTA-BIT results miss a comparison with other methods.
Finally, although DeLTA-BIT can predict connectivity maps in clinical data in approximately 8 minutes, it still requires DWI preprocessing, which can be computationally demanding for high-quality data—such as the HCP dataset. The regression models were designed under the underlying assumption that a learnable function exists between diffusion tensor information and connectivity maps. While this appears reasonable, it must be noted that diffusion data carries structural connectivity information globally; conversely, operating locally on a small bounding box may limit this estimation. As demonstrated in [120], thalamic parcellation depends more on a voxel's geometric surroundings than on the local fiber orientation within the thalamic gray matter. From this perspective, the DL models developed here may simply be learning geometric statistics from the tractography labels, potentially rendering the use of DTI data unnecessary.
Although the latter assertion fundamentally challenges the current design of DeLTA-BIT, it provides a path to substantially enhance the project's methodology. Indeed, since the models primarily learn statistical patterns, it may be possible to train them using T1-weighted images alone. This would reduce not only the time required for DWI processing but also the acquisition time itself, further accelerating the workflow while potentially maintaining the same level of performance.

The limitations discussed above necessitated the development of a second version of DeLTA-BIT, which will be presented in the following sections.

3.4 DeLTA-BIT version 2

The issues raised in the previous section—concerning both VIM definition and validation methodology—led the author to redesign the entire DeLTA-BIT pipeline. First, a new literature research was conducted to identify a well-established, patient-specific method for VIM localization. Several studies, such as [82, 121], highlight the difficulties of precisely localizing the VIM using traditional MRI. A common practice involves evaluating the VIM position through post-operative imaging of ET or Parkinson's patients who experienced long-term clinical benefits. This approach is generally used to demonstrate that positive outcomes correlate with the overlap between the Volume of Tissue Activation (VTA) (the target in DBS treatment) and a tractography-based region. Specifically, reference [121] identifies the contralateral dentate nucleus ROI—estimated via connectivity maps—as the most reliable proxy for the VIM. However, this study was conducted on a small cohort of only nine patients (six male, three female). Furthermore, the estimation was performed using high-angular-resolution

diffusion imaging (128 directions, 1.5,mm isotropic voxels, $b = 1500\,\mathrm{s/mm^2}$, 3T scanner, 60 minuets for the acquisition) while utilizing the entire contralateral cerebellum as the ROI for the contralateral dentate nucleus. The setup proposed in [121] presents some limitations, primarily concerning its applicability in a clinical setting.
Reference [121] also introduces a methodology to estimate the Dentato-Rubro-Thalamic Tract (DRTT) — the tract linking the contralateral dentate nucleus with the M1 cortex and crossing the thalamus at the VIM level. However, the intersection of the tract with the thalamus is still considered too broad for precise VIM localization. Despite these limitations, this method served as the starting point for the new target definition in DeLTA-BIT. Using an intuitive approach, the VIM can be localized by applying inclusion and exclusion criteria. It is well-established in the literature that the VIM is located within the thalamus, specifically in the region highlighted by the connectivity map with the M1 cortex, and that it is an integral part of the DRTT [122–125]. By calculating the intersection of these regions, the VIM can be accurately localized.
Nevertheless, this process requires high-quality DWI data, which is not typically acquired in standard clinical settings. While the HCP dataset meets these quality requirements, no post-operative data can be derived from it as it only contains images from healthy subjects, thus preventing the validation of the proposed method. To overcome this issue, DeLTA-BIT_v2 was first trained on the HCP dataset, using T1-weighted images as input and the ROI introduced above as target. Successively, two validation steps have been carried out: i) *internal validation*, where it was tested the model capability in predicting ROIs obtained by probabilistic tractography on the HCP data; ii) *external validation*, where it was tested the model capability in predicting the VIM location on clinical data, along with a comparison with an atlas-based method (THOMAS, [83]).

These methodology faced all the issues raised in DeLTA-BIT_v1.

3.4.1 Materials and Methods

Data source

T1-weighted and DWI data (including BEDPOSTX analysis) from HCP have been used for VIM ROI estimation and for model training and internal validation in the same manner proposed in DeLTA-BIT_v1. Addtionally, The SUIT atlas [126–129] has been exploited for the right dentate nuclei segmentation.

Clinical data from 7 ET patients who underwent unilateral MR-guided focused ultrasound thalamotomy (targeting the presumed Vim) for tremor control were used for the external validation analysis. Clinical outcome assessments were performed by a neurologist with $\geq$ 20 years of experience in movement disorders. Only patients achieving an optimal clinical outcome (defined as a $\geq 80\%$ reduction in the treated-side at The Essential Tremor Rating Assessment Scale (TETRAS) score) assessed at least one year after treatment were included, under the assumption that such sustained,

marked improvement reflects accurate targeting and effective ablation of the VIM nucleus. Patients included in these analyses were retrieved from a previously conducted study at the University-Hospital "Paolo Giaccone" of Palermo funded by the Italian Ministry of Health ("Ricerca Finalizzata 2016", grant no. GR-2016-02364526; principal investigator: Dr. Cesare Gagliardo). The study protocol received approval from the local Institutional Review Board (approval code N. 04/18). All participants provided written informed consent prior to enrollment, and all procedures were conducted in accordance with the principles of the Declaration of Helsinki. In particular axial 3D T1-weighted Inversion Recovery (IR) BRAin Volume (BRAVO) pulse sequence acquired using a 1.5T scanner (GE Signa HDxt with an 8-channel head coil) with a 1mm isotropic resolution were used (scan parameters: slice thickness 1mm; repetition time 12.4ms; echo time 5.2ms; inversion time 1866ms; matrix 256x256; number of excitation 1; filed of view 25.6x25.6). Each subject underwent both a pre-treatment screening MRI, routinely used for treatment planning, and a 48-hour post-treatment follow-up scan. The screening and follow-up images were subsequently rigidly co-registered, and binary lesion masks were segmented from the follow-up scans.

Lesion segmentation was manually performed on 48-hour postoperative 3D T1 IR BRAVO pulse sequences. Segmentation strictly included the well-defined T1 hypointense core corresponding to the previously characterized Zone II (cytotoxic edema), while explicitly excluding the surrounding Zone III (vasogenic edema) [130, 131] (see figure 3.18). All segmentations were generated in ITK-SNAP version 4.2.2 [132] using the "pre-segmentation thresholding" tool by first defining upper and lower intensity thresholds, after which a seed bubble was placed within the lesion to initialize the differential-equation–based contour-evolution algorithm. All delineations were performed by a neuroradiologist with >12 years of experience in neuroanatomical segmentation and >10 years of experience as primary operator in tcMRgFUS procedures.

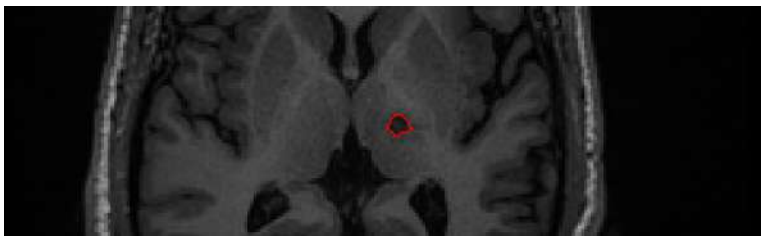


Figure 3.18: 24-hour post-treatment clinical image of an ET patient who underwent tcMRgFUS treatment. The thalamotomy lesion can be observed as a hypointense area within the left thalamus. The manual segmentation is highlighted in red.

Correlation between post-treatment clinical outcomes and the performance of the evaluated algorithms was assessed using the TETRAS score, considering only the hemi-score corresponding to the treated side.

External validation was performed by using the screening images as input for both the NN and the THOMAS pipelines, while the manually generated lesion masks served as GT

THOMAS Atlas

Thalamus-Optimized Multi-Atlas Segmentation (THOMAS) is a thalamus specific atlas-based pipeline able to identify 12 nuclei and sub-nuclei (22 ROIs plus thalamus ROI for each hemisphere), using just T1 images, as described in [83]. It was run in a docker container with operating time of 15-30 min per patient on our workstation (DELL Precision Tower, 128 GB RAM, Intel® Xeon(R) Platinum 8268 CPU @ 2.90GHz × 48). The VLPv ROI (i.e., the inferior portion of the ventral lateral posterior nucleus, as defined by Su et. al in [83]) was used as a surrogate for the VIM in the clinical dataset. In contrast, THOMAS-derived left thalamus segmentation was employed in the HCP dataset for probabilistic tractography, as THOMAS is a thalamus-specific atlas that provides more accurate thalamic delineation.

Probabilistic tractography and connectivity maps

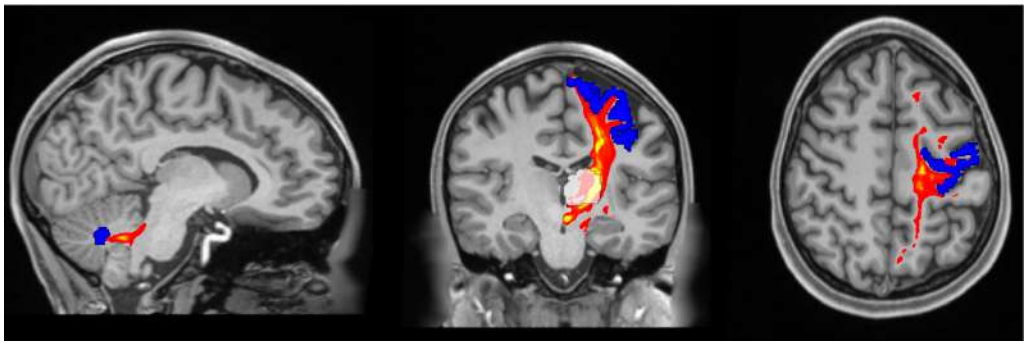
For each subject, tractographies were similarly performed as described in section 3.2.3. Specifically, three different tractographies were obtained, by varying seed and target regions: i) left thalamus-to-left M1 cortex; ii) right dentate nucleus-to-left M1 cortex including the left thalamus as a way-point; iii) left M1 cortex-to-right dentate nucleus including the left thalamus as a way-point. The last ones were used to identify the DRTT [121], see figure 3.19. Furthermore, the left thalamus-to-left M1 cortex tractography was exploited to evaluate the correspondent connectivity map, shown in Figure 3.20.

VIM identification

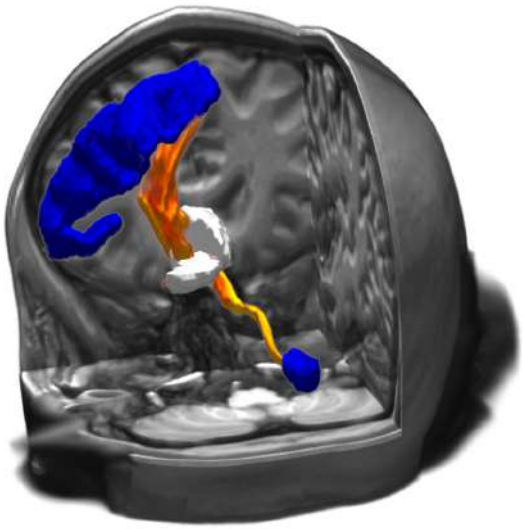
It is known that the VIM receives input from the contralateral dentate nucleus and projects to cortical areas of the ipsilateral motor network, primarily the primary motor cortex (M1) [122–125]. For this reason, the VIM location was identified by intersecting the thalamus region, the M1 connectivity map, and the tractogram obtained by merging the dentate-thalamo-M1 tractographies in both directions (see section 3.2.3). Since probabilistic tractography is affected by statistical noise, a mean spatial filter was applied to the resulting distribution. In order to obtain a binary mask of the VIM location, a threshold (10% of the maximum intensity) was applied. Figure 3.21 shows the obtained distribution and the contour of the VIM.

Revised design of DeLTA-BIT

DeLTA-BIT_v2 is constituted by a solely 3D Unet and its new architecture is depicted in figure 3.22. The network expects a 3D image as input — T1-normalized MR image in our case — and attempts to predict the spatial localization of the VIM, as defined in Section 3.4.1. The same bounding box defined in section 3.3.1 was used to reduce computational complexity.



(a)



(b)

Figure 3.19: Probabilistic tractography streamlines for the DRTT. In blue the left-M1-cortex and the right dentate nucleus ROIs. In white the left thalamus. In red-to-yellow scale the DRTT tractogram obtained by merging the dentate-thalamus-M1 tractography in both directions. Panel a, from left to right: sagital, coronal and axial projections. Panel b: rendering of the 3D image.

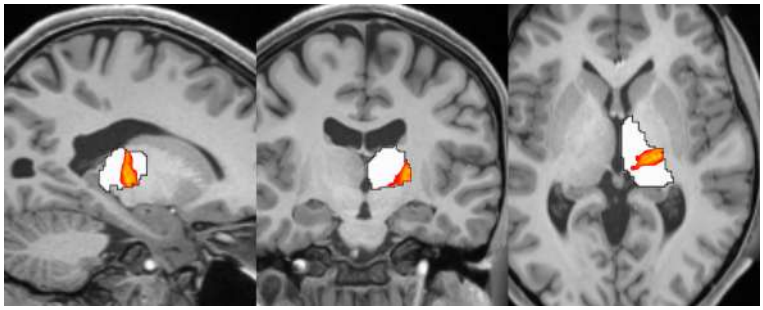


Figure 3.20: M1 connectivity map obtained setting the left thalamus as seed and the M1 as target region. As is well known, the VIM is located within this area.

The NN was trained using the same set-up proposed in section 3.3.1 and the same hardware.

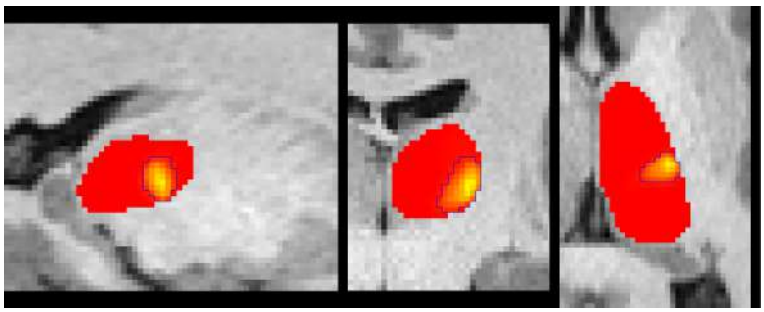


Figure 3.21: Identification of the VIM region. The VIM ROI was obtained by applying a mean filter (kernel size $= 5 \times 5 \times 5$) to the raw distribution obtained by intersection. The contour of the obtained binary mask is highlighted in blue.

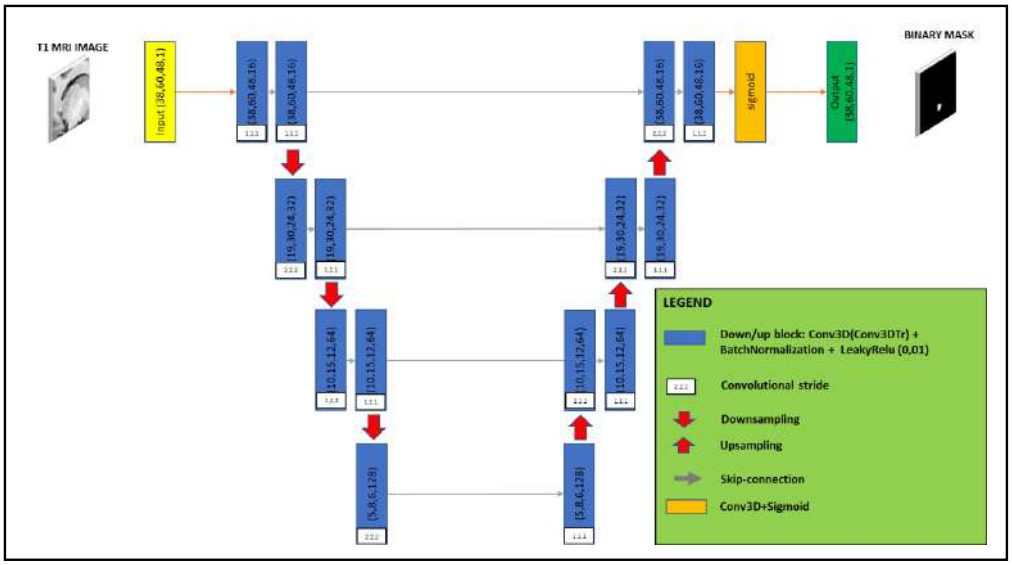


Figure 3.22: Schematic representation of the U-Net neural network used in DeLTA-BIT_v2. As it can be notice, the input consist on T1-weighted images only, cropped in accordance with the bounding box defined in section 3.3.1. The target to predict is given by the VIM ROI definition described in section 3.4.1

Evaluation methods

The prediction ability of the model has been evaluated on the test sets (HCP test set and clinical data). The metric we used is the DSC. Additionally, since the VIM region is very small compared to the size of the image, we included two more metrics in model evaluation: i) the surface Dice Similarity Coefficient (sDSC) and ii) the distance between the centers of the mass of the true and predicted VIM (D_{COM}). The sDSC has a similar definition to the DSC, or *Volume* DSC, except that only the voxels along the surface are considered. This requires a tolerance value to detect voxels located along the surface of a volume, which was set to 1 mm.

Finally, in the external validation, two additional metrics were considered. The first is the True Positive Rate (TPR), also referred to as *Recall*. Racalling definition 2, given a predicted ROI X and a GT ROI Y, we define TP (*True Positives*) as the voxels classified as positive (belonging to the ROI) in both X and Y, FP (*False Positives*) as the voxels classified as positive in X but not in Y, and FN (*False Negative*) as the

amount of positive voxels in Y missed in X. Based on these definitions, the TPR is expressed as:

$$\text{TPR} = \frac{TP}{TP + FN}. \tag{3.1}$$

In other words, Recall can be seen as the portion of GT volume recovered in the prediction.

In similar way, the *Precision* of a segmented region is evaluated as:

$$Precision = \frac{TP}{TP + FP}. \tag{3.2}$$

Precision corresponds to the proportion of the predicted volume that falls inside the GT volume. It is important to evaluate these two metrics jointly, since a predicted region whose volume is much larger than the GT and fully contains it would yield a TPR equal to 1, but a very low Precision (given by the ratio GT volume/predicted volume). Conversely, a region whose volume is entirely contained within the GT would result in high Precision but low TPR (given by the ratio predicted volume/GT volume).

3.4.2 Results

Internal validation

In this section, results on the test set (264 samples) are presented. Figure 3.23 depicts the overlap between the reference ROI (red contoured region) and the prediction (green contoured region) for three representative subjects, in the sagittal, coronal and axial projections. Although the figure shows only one slice per projection, a good agreement between the GT and the model output is observed.

Table 3.4 summarizes the results of the evaluation analysis with different metrics utilized as described in Section 3.4.1, performed over the whole test set. We obtained a mean DSC of 0.62 ± 0.15, which is a good result considering the difficulty of localizing such a small area (mean volume of 396 mm^3). To perform a more comprehensive evaluation, we also considered the sDSC. This metric assesses the overlap between the surfaces of two ROIs, considering a thickness of $1mm$. Thus, it allows for evaluating the agreement between the GT and the prediction along the border. For this metric, we obtained a mean value of 0.76 ± 0.17.

	DSC	sDSC	$D_{COM}[mm]$
mean	0.62	0.76	1.74
std	0.15	0.17	0.83

Table 3.4: Evaluation analysis with different metrics utilized.

In Figure 3.24a and b we show the dispersion of the results around the mean values.

Finally, as the last evaluation test, we considered the distance between the centers of mass (D_{COM}) of the reference and predicted ROIs. We obtained a mean distance

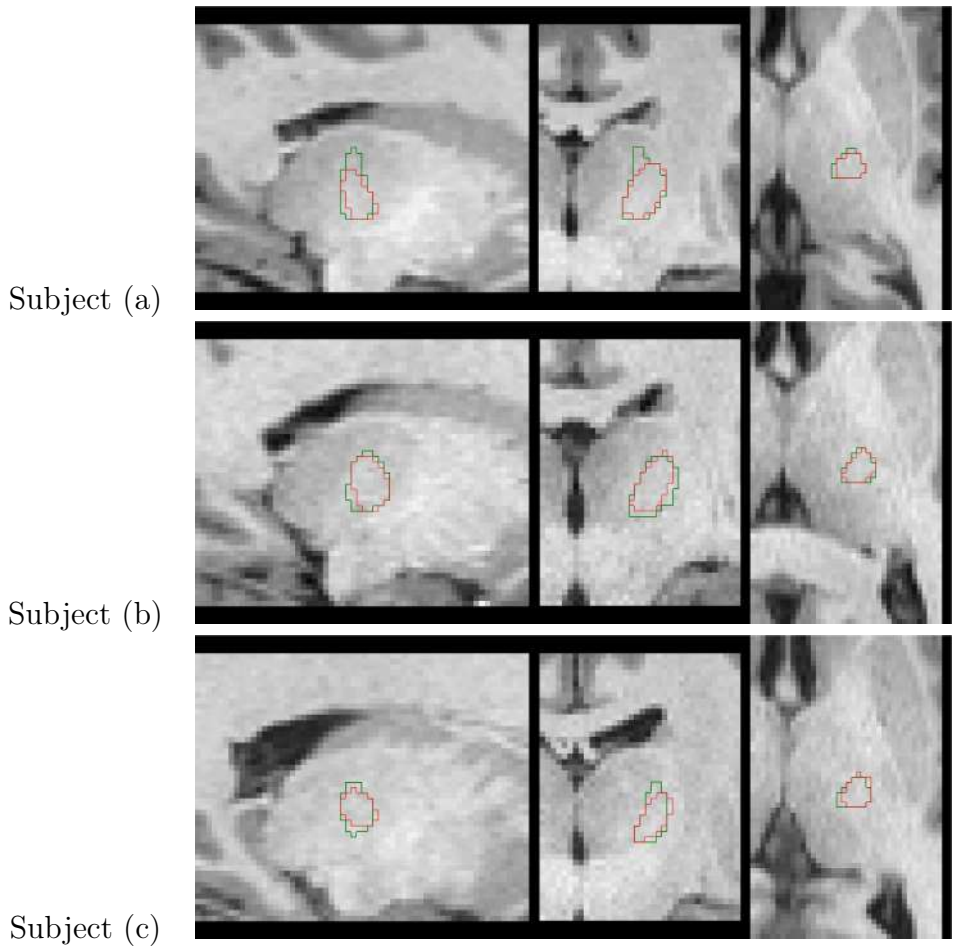


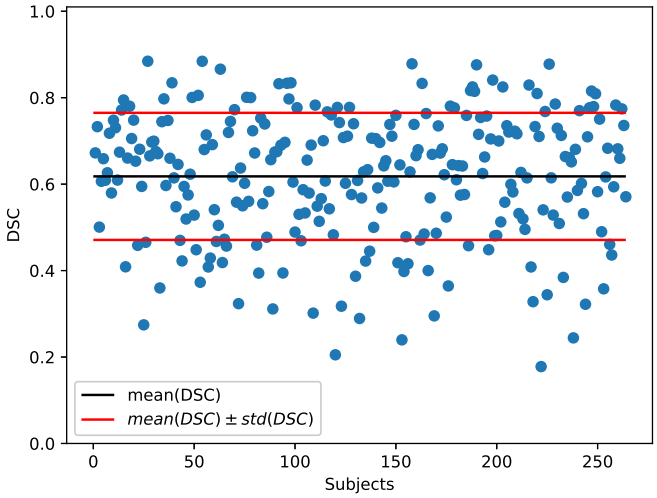
Figure 3.23: Comparison of VIM ROIs for three representative subjects. From left to right: sagittal, coronal and axial projection. Green: predicted ROI; red: reference ROI. Subject (a): $DSC = 0.78$, $sDSC = 0.98$; subject (b): $DSC = 0.80$, $sDSC = 0.99$; subject (c): $DSC = 0.88$, $sDSC = 0.99$.

of 1.74 ± 0.83 mm, which is comparable to the size of voxels along a direction. The correlation plots for each direction are shown in Figure 3.25.

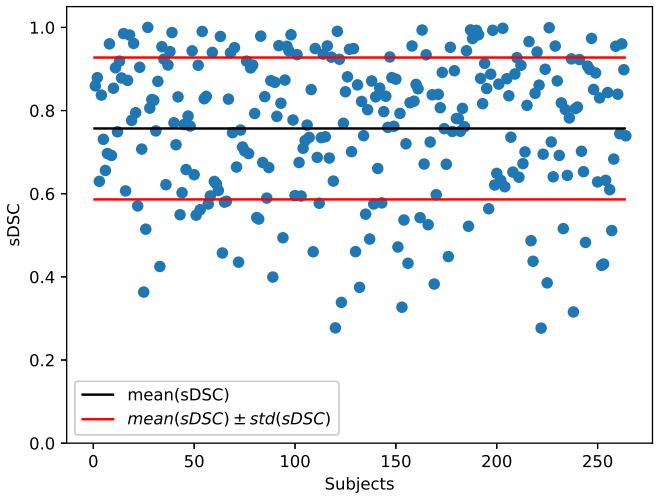
Besides the model's performance, it is important to evaluate the gain in terms of time required. The standard pipeline requires approximately 2.5 hours for T_1 pre-processing and between 2 minutes and 4 hours for DTI pre-processing, depending on the number of gradient acquisitions. Our model requires only a T_1 MR image to predict the VIM location, eliminating the need for DTI acquisition and processing. The time required to predict the VIM ROIs for the whole test set (264 subjects) was 5.9 seconds on our hardware. This result is of great importance, as it enables VIM localization during surgical procedures—something not feasible with the standard MR pulse sequences—and the remarkably short computation times make it particularly promising for clinical translation, where lengthy processing often drives the choice toward alternative targeting strategies.

External validation

To evaluate the generalization capability of our AI model, we used clinical data from 7 ET patients who underwent FUS treatment. T1 screening images have been used



(a)



(b)

Figure 3.24: Volume (upper panel) and surface (lower panel) Dice similarity coefficient between the predicted and the reference VIM binary masks. Average scores: $DSC = 0.62 \pm 0.15$ and $sDSC = 0.76 \pm 0.17$.

as input for both the NN and the THOMAS pipeline. T1 follow-up images, where the edema induced by thermal ablation appears as a hypo-intense region, were used to derive reference ROIs (see section 3.4.1). These ROIs can be seen as GT for the screening images after appropriate co-registration, as they are supported by positive clinical outcomes. Scores obtained by NN and THOMAS are reported and compared in this section.

Figure 3.26 illustrates the spatial correspondence between the reference ROI (red contour), the NN prediction (green contour), and the THOMAS prediction (blue contour) for three representative subjects, displayed in the sagittal, coronal, and axial planes.

Table 3.5 and Figures 3.27 report the results for both the NN and THOMAS pipelines. Overall, the two methods exhibit comparable performance: NN attains

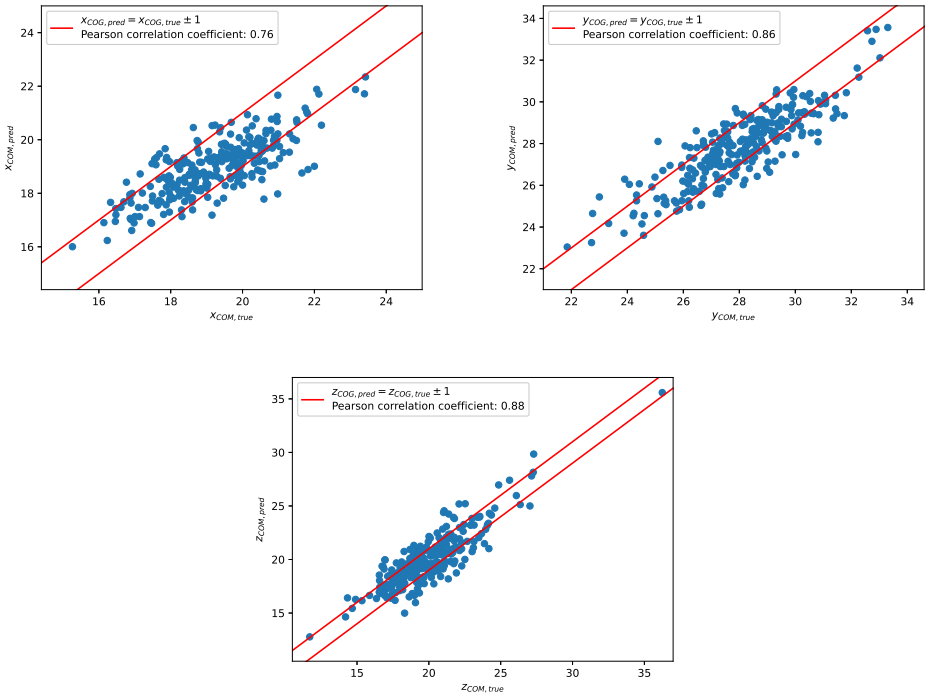


Figure 3.25: Correlation plot of the coordinates of the centre of mass for the true and the predicted VIM ROIs.

marginally higher DSC and sDSC values than THOMAS (Figure 3.27a,b). In terms of TPR, which represents the portion of GT volume recovered by the predictions, THOMAS performs better than NN (Figure 3.27c). It is noteworthy, however, that THOMAS produces ROIs with an average volume of 371 mm^3, substantially larger than the 205 mm^3 predicted on average by NN (in this clinical dataset, the average GT volume is 102 mm^3). This discrepancy suggests that the superior performance of THOMAS, in terms of TPR, may be influenced by a volume-related effect (see section 3.4.1), as further supported by the observed *Precision* values (Figure 3.27d).

method	DSC	sDCS	TPR	Precision	$D_{COM}[mm]$
NN	0.23 ± 0.11	0.25 ± 0.07	0.36 ± 0.12	0.20 ± 0.13	4.5 ± 1.5
THOMAS	0.21 ± 0.07	0.20 ± 0.05	0.57 ± 0.19	0.13 ± 0.05	4.0 ± 1.1

Table 3.5: Evaluation analysis with different metrics utilized.

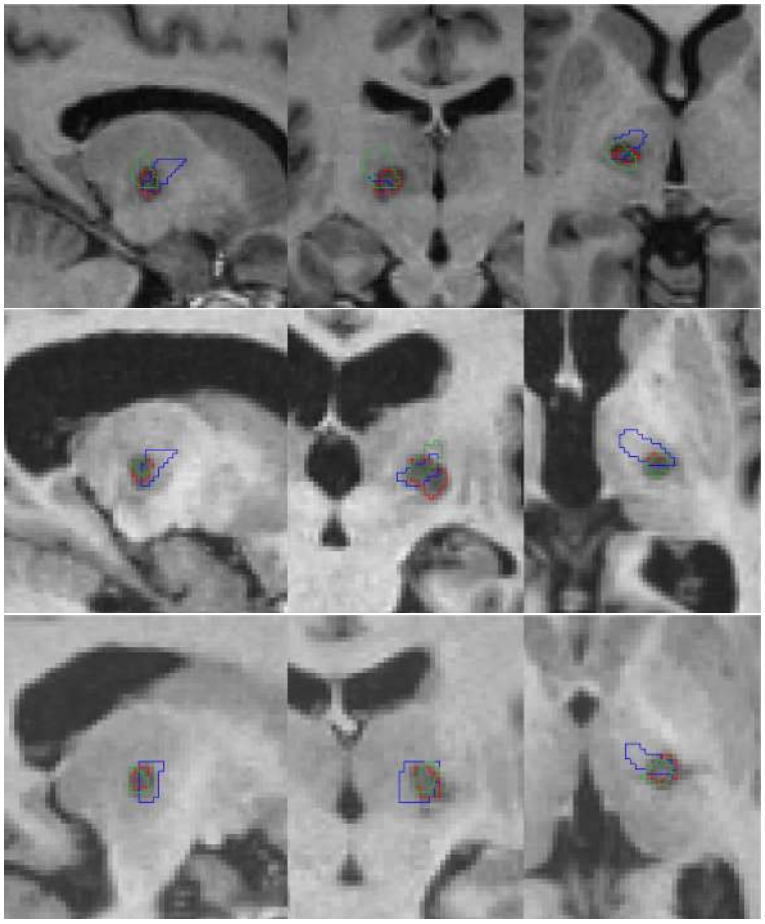


Figure 3.26: Comparison of VIM ROIs for three representative subjects. From left to right: sagittal, coronal and axial projection. Green: predicted ROI; red: reference ROI. Blue: THOMAS prediction. Subject (a): NN $DSC = 0.26$, $sDSC = 0.23$, $TRP = 0.53$, $Precision = 0.17$; THOMAS $DSC = 0.18$, $sDSC = 0.21$, $TRP = 0.45$, $Precision = 0.12$. Subject (b): NN $DSC = 0.30$, $sDSC = 0.32$, $TRP = 0.25$, $Precision = 0.38$; THOMAS $DSC = 0.18$, $sDSC = 0.15$, $TRP = 0.30$, $Precision = 0.13$. Subject (c): NN $DSC = 0.42$, $sDSC = 0.38$, $TRP = 0.51$, $Precision = 0.36$; THOMAS $DSC = 0.31$, $sDSC = 0.27$, $TRP = 0.58$, $Precision = 0.21$. Note that patient a) was treated on right thalamus. The NN was trained on the left side only. However, thanks to data augmentation applied during the training, the NN is able to detect the VIM in the right side with similar performance as for patients b) and c).

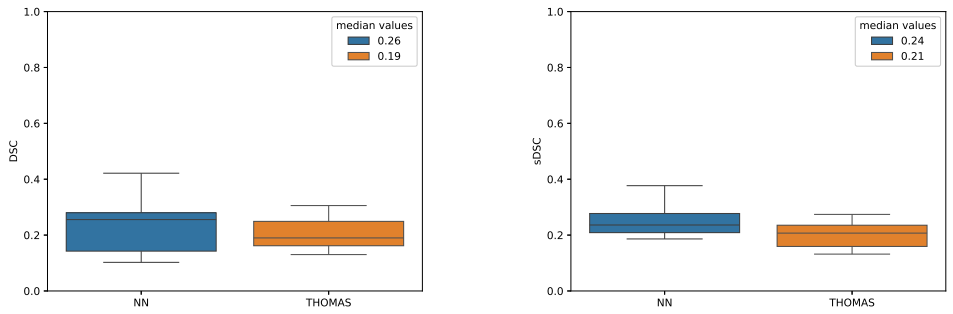


Figure 3.27: Metrics comparison between NN and THOMAS prediction ability. a) DSC distribution; b) sDSC distribution; c) TPR distribution; d) Precision.

3.4.3 Discussion

DeLTA-BIT_v2 overcame issues raised in its first version. In particular, a new procedure—based on well-established evidence [122–125]—was developed to train a model that requires T1-weighted images only. This new approach not only solves the issues related to VIM definition but also allows for fast estimation during treatments, eliminating the need for DWI acquisitions. External validation on clinical data—using lesion-based VIM estimation from patients with positive outcomes—offered the possibility to study DeLTA-BIT's performance on low-quality, clinical real-world data (images acquired with a 1.5 T scanner) of ET patients. Additionally, lesion-based VIM estimation in responding patients represents a more reliable GT. Finally, a comparison of the results with a widely-used atlas (THOMAS) provided benchmarks for the performance. Despite these improvements, DeLTA-BIT still presents some limitations.

First, the model is not provided with any information on the diffusion. The tractography data are fed to the model only at the training stage to identify the desired region, while at the querying stage the model produces an output based on T_1 images only. As it is well known, DL models are able to learn correlations between input and target, in our case a T_1 image and the corresponding tractography-based region. Given the size of the dataset utilized, one could be confident that these correlations are well achieved.

Secondly, most functional neurological disorders are more common in aged people in which the brain typically shows more evident signs of atrophy with larger subarchnoid and ventricular spaces. The enlargement of the third ventricle in particular may result in significant difference of thalamic targets location compared to AC-PC references. The HCP dataset contains images from young adult subjects only, acquired at 3T, which introduces a bias in our trained model. This issue has been taken into account during the training by data augmentation. Moreover, the NN was trained in predicting tractography-based ROIs, i.e. regions that overlaps with the VIM location with high probability, even though they are not exactly the same. The protocol for VIM location defined in section 3.4.1, based on sectioning the DRTT estimated as reported in literature [121], is not error-free. In ET surgical treatments the target is provided by a direct feedback of the patient. We must consider that the VIM location is still an open issue [82] and our definition is just an approximation. In this respect, our work is meant as a support to reduce the region to inspect.

All these issues become evident in the external validation. Indeed, the results on DSC and sDSC are considerably worse compared to those obtained on the HCP data. This is certainly due to the differences in data quality and patient anatomy variability. However, it must be considered that in the external validation we used as GT the segmented region of the edema induced by thermal ablation. This has two implications. First, this ROI also does not correspond exactly to the VIM, although it includes it. Second, the ROI is extracted from the follow-up image and it is used as GT for the screening images after appropriate co-registration. Notwithstanding co-registration, follow-up and

screening images are different; this introduces an uncertainty that cannot be avoided and affects the results.

All the factors mentioned above also affect the results obtained with the THOMAS pipeline. The Dice coefficients are slightly higher for NN than for THOMAS, but these differences are not large enough to clearly determine which pipeline performs better. However, from a timing perspective, NN is able to provide predictions in a significantly shorter computational time compared to THOMAS (a few seconds per patient versus tens of minutes), providing a clear advantage. Additionally, DeLTA-BIT has another potential advantage over THOMAS: it can be fine-tuned on clinical data. Indeed, in CNNs, pre-training on large and relatively generic datasets (as the one of the HCP) followed by fine-tuning on smaller domain-specific datasets (such as real-world clinical datasets) has consistently demonstrated substantial performance improvements. This transfer learning strategy enables networks to learn general low-level and mid-level features that can later be adapted to specialized tasks, while also reducing overfitting and improving convergence in data-limited domains such as medical imaging [133–135]. A preliminary test, conducted using ET patients clinical data from the Policlinico Hospital of Palermo, has demonstrated the effectiveness of the transfer learning approach, restoring performance to levels consistent with those observed during internal validation and significantly outperforming the THOMAS atlas ($DSC = 0.46 \pm 0.14$ vs $DSC = 0.31 \pm 0.07$, respectively).

3.5 Conclusion and future perspectives

Recent literature reports that tractography-based targeting has important benefit for tcMRgFUS and DBS treatment planning and treatment [95, 121]. First, this is a personalized method as it takes into account the individual variability rather than using atlas-based indirect targeting, which, although standardized, has evident weakness. The choice of targeting through probabilistic tractography approach allows to consider the inherent variability in brain anatomy across patients and to improve outcomes of thalamotomy [94, 95]. The probabilistic tractography approach shows reduced uncertainty in dense areas with crossing fibers and is less prone to sampling limitations with respect to deterministic tractography. However, probabilistic approach could be computational demanding and time consuming [85, 94]. Indeed, since DWI images are prone to be affected by noise and motion artifacts, a pre-processing of diffusion data is mandatory. Then, BEDPOSTX and PROBTRAKX are very time consuming. In recent years GPU versions of these tools have been developed to significantly accelerate processing; however, the time required for the entire protocol is still too long for a real-time application in clinical practice. Furthermore, powerful GPU-based architectures are not easily available to clinicians for this purpose, and DWI acquisitions are not feasible during interventions.

A deep learning approach has been developed to help targeting the ventral intermediate nucleus (VIM) in neurosurgical treatments of neurofunctional disorders as ET. This is achieved by predicting the VIM location using a CNN trained and validated on subject specific probabilistic tractography data, by exploiting structural (T_1-weighted images) and diffusion-weighted acquisition of more that 1,000 subjects of the HCP dataset. In its first version, DeLTA-BIT was designed to predict the connectivity maps typically used in tcMRgFUS treatment planning, demonstrating the NN's capability to learn features relevant to this task. The issues raised by this pipeline led the author to enhance both the model and its design.

In DeLTA-BIT_v2, an additional validation was performed on real-world clinical images from a small sample of patients who underwent tcMRgFUS thalamotomy, improving the assessment of the model's generalization capability. Despite the relatively modest performance, the achieved scores are in line with those reported for the widely used THOMAS atlas, placing DeLTA-BIT on the same level. However, it is evident that further fine-tuning of the model should be considered for effective clinical translation.

It is important to underline that DeLTA-BIT enables VIM prediction in a very short time (a fraction of a second per subject), making the proposed method promising for real-time targeting in MR-guided procedures.

Future work will focus on investigating the benefits of further fine-tuning the model, with the aim of enhancing DeLTA-BIT applicability in real-world clinical settings.

CHAPTER 4

VALIDATION AND CLINICAL IMPLEMENTATION OF AN AI-DRIVEN SEGMENTATION TOOL AND DESIGN OF A QUALITY ASSURANCE PIPELINE

Radiotherapy (RT) is the clinical practice that consists of delivering a sufficient dose of ionizing radiation to tumor cells in order to induce necrosis. As with any clinical procedure, RT must be carefully planned; in particular, it is necessary to maximize energy deposition in tumors while preserving, as much as possible, healthy tissues. For this reason, RT planning can be seen as an optimization task in which many variables and constraints must be considered. We can identify: radiation quality factors—such as the type of radiation, its energy, delivery system, etc.; geometrical factors—such as beam directions, entry points/surfaces, tumor location, patient positioning, etc.; anatomical factors—such as the anatomical site of the tumor (head & neck, thorax, abdomen, pelvis), tumor environment (e.g., organs at risk), presence of prostheses, patient's health status, and many others. In this context, oncologists and medical physicists work jointly in order to guarantee both patient safety and treatment effectiveness.

RT was introduced a few months after Wilhelm Röntgen's discovery of X-rays in 1895. Emil Grubbe treated the first patient in Chicago, USA [136]. Major advances were achieved thanks to the pioneering work on radioactivity by Marie Curie. In fact, until the 1950s–1960s—when linear accelerators (LINACs) were introduced into clinical practice—radioactive substances were used as sources of ionizing radiation. In some parts of the developing world, this methodology is still in use.

Over the last 70 years, research has focused on improving both Radiotherapy Treatment Planning (RTP) and the production and delivery of ionizing radiation. Hounsfield's invention of CT not only provided the possibility to localize tumors within the body, but also enabled the voxel-wise estimation of the linear attenuation coefficient—a physical parameter that describes how ionizing radiation energy is absorbed in matter. In this way, CT contributed to translating RT toward a personalized treatment paradigm.

Efforts have also been devoted to developing collimation systems capable of reducing irradiation of healthy tissues. At the same time, research has translated discoveries in material and nuclear physics—such as the way high-energy accelerated hadrons release energy at depth, a phenomenon known as the *Bragg peak*—into the RT field. The latter application has led to a radiotherapy technique known as hadron therapy, which is particularly effective in treating localized tumors in the vicinity of radiosensitive organs.

RTP is a complex, patient- and tumor-specific procedure that includes several steps. The most important one concerns the estimation of the Dose Distribution (DD) within the body. This describes how energy is deposited in tissues and allows oncologists and medical physicists to assess whether (i) the dose delivered to tumor cells will induce necrosis and (ii) the dose absorbed by organs at risk (OARs) will be safe for the patient. This step is crucial, although it involves several challenges. In fact, DD cannot be directly measured within tissues. As a consequence, it can only be estimated numerically and then compared with experimental measurements performed in water or in tissue-equivalent media.

Numerical estimation of DDs is performed using Monte Carlo or pseudo-analytic[1] methods. Both approaches require knowledge of the beam geometry and a patient model—typically provided by CT imaging.

Once the DD is available, all clinical objectives and constraints—such as doses to the tumor and to OARs—can be evaluated. At this stage, the contouring of the tumor and OARs on CT images plays a crucial role and is typically performed by radiologists and trained technicians.

To manage the entire RTP workflow, dedicated software—known as Treatment Planning System (TPS)s—has been developed. TPSs handle patient registration, medical imaging, platforms for DD calculation, and provide tools to assist operators in tumor and OAR contouring. As can be easily anticipated, this latter phase is particularly time-consuming. Moreover, the results are subject to inter-operator variability, which can introduce uncertainties in RTP.

In this context, AI has emerged as a prominent focus of research due to its capacity to automate complex processes, including organ and tumor segmentation. As already discussed, CNNs have made auto-segmentation a nearly solved problem[2]. In recent years, AI-driven tools have evolved into off-the-shelf products designed to optimize diagnosis and radiotherapy treatment planning. Today, these tools are increasingly being integrated into clinical workflows, enabling automated segmentation of tumors and organs at OARs and significantly reducing time requirements.

To date, although well-established evaluation criteria have been developed to assess the performance of AI models, standardized Quality Assurance (QA) protocols to

[1]Semi-analytic numerical methods primarily based on the Bethe–Bloch equation [137, 138]

[2]The advent of U-Net in both 2D and 3D tasks, along with results obtained across a wide range of applications, has provided a well-established methodology for auto-segmentation in biomedical imaging. In most cases, limitations arise primarily from data availability.

ensure the stability, reproducibility, and traceability of AI systems in medical settings are still lacking. In particular, for approved or validated software, there are no clear guidelines on how to preserve the integrity of the clinical workflow in the event of system upgrades—a gap that could potentially lead to errors in the treatment chain, with serious consequences for patient safety and legal implications in regions such as the European Union.

This chapter presents the author's work on the development of robust QA protocols aimed at addressing potential changes and ensuring the reliability of TheraPanacea ART-Plan, an AI-based tool for the automatic segmentation of OARs in radiotherapy, recently integrated into the clinical workflow at the MedAustron ion therapy center in Austria. These research activities were conducted during a six-month research stay abroad at the MedAustron center.

The chapter begins with an overview of RT modalities, providing the necessary clinical and technological background. It then focuses on the design and implementation of the proposed QA protocols, highlighting the methodological choices adopted to ensure stability and reproducibility of the system. Finally, a comparative analysis between different software versions (ART-Plan 2.1.0 and 2.2.1) is presented and discussed in order to evaluate the impact of updates on segmentation performance and overall workflow reliability.

Concluding it must be underline that the main achieved result was demonstrating for the first time that models upgrades should not be blindly accepted, as claimed improvements were not always reflected in clinical data.

4.1 Radiotherapy and Image Contouring

4.1.1 Overview of Radiotherapy Modalities

In RT, ionizing radiation is used to induce necrosis in tumor cells. In combination with chemotherapy, it represents one of the most common therapies in cancer care [139, 140]. Both therapies, when possible, are preferred to surgery due to their non-invasive nature [141]. The World Health Organization confirms that RT is used in 50% of cases [142]. The American National Cancer Institute distinguishes two main types of RT: external beam radiation therapy and internal radiation therapy. The latter, also known as brachytherapy, consists of placing small radioactive sources in close proximity to or within the tumor, allowing the treatment of a localized region of the body. It is typically used for tumors located in the pelvic region.

In external beam RT, ionizing radiation is delivered from an external source. In this case, the radiation delivery system plays a fundamental role in the treatment. Since radiation must traverse healthy tissues before reaching the tumor, the type of radiation—and consequently its production mechanism—determines how energy is deposited

within tissues. This behavior is described by the Depth Dose Profile (DDP). Based on DDP and the *Linear Energy Transfer (LET)*—a physical quantity used to characterize radiation quality—two main categories can be identified: low-LET and high-LET radiation. Low-LET radiation, such as photons and electrons, is characterized by a relatively high energy deposition in the entrance region followed by an approximately exponential attenuation with depth (blue curve in figure 4.1). In conventional beam configurations—typically low dose rate and without specialized collimation systems such as those used in microbeam radiation therapy—low-LET radiation constitutes *conventional RT*.

High-LET radiation, including protons and heavier ions, exhibits a markedly different depth-dose behavior. It is characterized by a relatively low and nearly constant energy deposition along the entrance path, followed by a pronounced peak—known as the Bragg peak—and a rapid fall-off beyond the peak, resulting in negligible dose deposition (orange curve in figure 4.1).

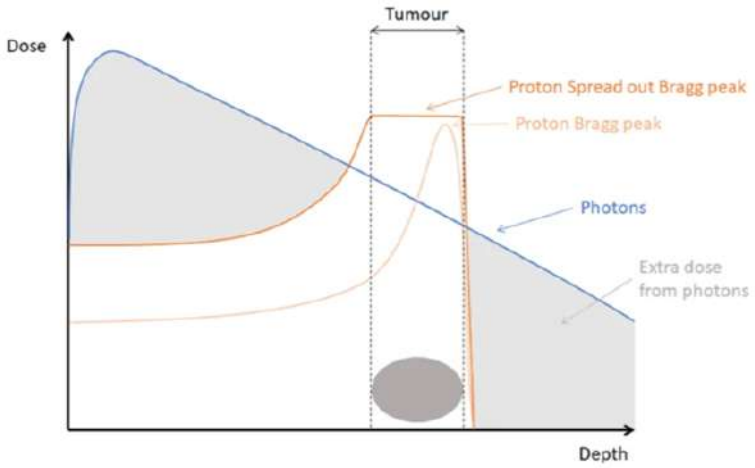


Figure 4.1: Comparison of DDPs in water for different types of radiation. The blue curve represents the low-LET behavior of photon radiation, with a dose peak located near the entrance region, typically around $1.5 \sim cm$ depth for a $6 \sim MV$ photon beam. The orange curves represent proton beam behavior. The thin orange curve shows the DDP for monoenergetic protons, characterized by the Bragg peak, typically located at approximately $15 \sim cm$ depth for $150 \sim MeV$ protons. The thick orange curve represents a Spread out Bragg peak, obtained by superposition of proton beams with different energies, used to achieve a uniform dose distribution across the tumor volume. Image adapted from [143].

4.1.2 Impact of Radiation Modality on Organ-at-Risk Contouring

The reason why radiation LET distinction was introduced arises from the way high-LET radiation affects patient positioning. In conventional RT, the LINAC head, from which the beam is delivered, can be easily rotated around the patient. In more advanced techniques, such as tomotherapy, full rotation around the patient is allowed, together with beam intensity modulation and a CT-guided system (see figure 3.2). This is reflected

in the RTP, since dose distribution estimation and contouring can be performed in a natural patient position (typically supine positioning).

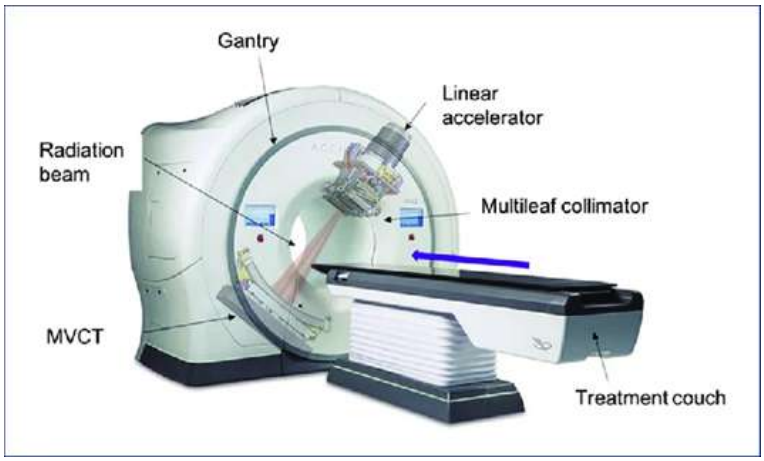


Figure 4.2: Functional scheme of a commercial tomotherapy unit. As can be observed, the LINAC head is able to rotate around the patient. At the same time, a detector system acquires images of the patient, providing information for image-guided treatment. Figure adapted from reference [144].

In ion beam therapy, where particles are accelerated by synchrotrons and delivered through a sophisticated beam transport system consisting of vacuum beamlines and electromagnetic elements for beam steering (see figure 4.3), beam directions are typically fixed (e.g., horizontal and vertical beamlines). Some centers, such as MedAustron, have developed gantry systems for beam rotation; however, these are limited to a restricted number of angles. In RTP, this translates into the need for patient positioning in less natural configurations (e.g., prone position or with neck hyperextension). This has consequences particularly in organ contouring, making the use of AI-based auto-contouring tools more challenging compared with conventional RT as reported by Chen et al. in reference [145].

Figure 4.3: Layout of the MedAustron facility with accelerator components, and clinical and nonclinical research irradiation rooms indicated. Figure adapted from reference [146]

4.1.3 ROIs contouring in radiotherapy treatment planning

Contouring target volumes and neighbouring OARs is a core step in RTP. While these structures have traditionally been manually delineated, semi-automated methods are increasingly integrated as assistive tools; such approaches have proven particularly effective to asses contouring consistency [145, 147]. Nevertheless, human validation and manual adaptations remain necessary.

Despite advances in semi-automated methods, manual contouring particularly in the abdomen and pelvis remains time-consuming and highly variable, comparing with target volumes compared and OARs in other anatomical sites (e.g., head & neck and thorax). This variability is driven by multiple sources of uncertainty, including patient positioning (e.g., prone, supine, rotated, upright, etc.) and organ filling, inter- and intra-observer differences, imaging quality and resolution, and the lack of standardized contouring guidelines across centers. Such inconsistencies may compromise treatment plan quality and adversely affect patient outcomes, with several studies associating inadequate contouring with reduced disease control and increased toxicity [148–150].

Before DL methods were introduced in RTP, atlas-based approaches were widely used. This strategy relies on deformable registration to transfer labels from pre-contoured reference patients, making it relatively simple to deploy but highly dependent on anatomical similarity and prone to registration-driven errors that often require substantial manual correction [151]. In contrast, DL models learn organ boundaries directly from large annotated datasets and therefore tend to deliver higher geometric accuracy, better robustness to anatomical variability, and significantly reduced manual editing time in clinical workflows. Research has demonstrated high-performance semantic segmentation of medical images using DL, achieving DSI (equation 1.4) values of up to 0.90 [5, 54, 61, 152–154]. Commercial solutions such as ART-Plan/Annotate (Therapanacea, Paris, France) further illustrate the clinical maturity of these methods, offering off-the-shelf DL-powered tools for automated segmentation of organs and lymph nodes in CT and MRI data used in treatment planning.

While atlas-based methods remain easier to implement and less data-intensive, their performance degrades in complex or atypical anatomies, whereas DL models once properly trained and validated provide faster, more consistent, and clinically preferred auto-contours across anatomical sites sites. Importantly, these DL-based tools are not designed to function as stand-alone decision-making systems but rather as complementary instruments that enhance workflow efficiency and consistency. Implementing AI in clinical practice is a complex, multidisciplinary challenge involving technological, ethical, and regulatory considerations.

Although rapid advances and the increasing availability of data have accelerated development of AI software, most of the literature focuses on the use of AI for RT quality assurance (QA) [155–159], while there is a lack of studies addressing QA frameworks for the algorithms themselves. To date, only limited guidance has been published [160].

In the European Union, AI systems in radiation therapy are classified as high-risk and regulated under the AI Act and Medical Device Regulation (MDR). Under Section 3 (Articles 16–19) of the AI Act, clinical users hospitals and radiotherapy departments must ensure proper deployment, human oversight, and accessible documentation and logs for traceability and evidence-based decisions. Combined with MDR requirements for safety and post-market surveillance, these rules emphasize user responsibility for transparency, accountability, and patient safety. These aspects have motivated the authors' research team to develop a QA protocol specifically designed for quality assessment and monitoring of the outputs of a commercially available AI-based auto-contouring software, already integrated into the clinical workflow at the MedAustron ion beam radiotherapy center.
This work presents the first QA protocol for auto-contouoring software such as ART-Plan/Annotate (TheraPanacea, Paris, France).

4.2 Materials and Methods

4.2.1 Setting

This study was conducted at MedAustron, the ion beam therapy center in Lower Austria, using ART-Plan/Annotate (TheraPanacea, Paris, France) versions v2.1.0 (2024) and 2.2.1 (2025) in conjunction with RayStation TPS (RaySearch Laboratories, Stockholm, Sweden) versions v2024A-SP1 and v2025-SP1 respectively. The datasets included planning CTs acquired with a Brilliance Big Bore Oncology CT scanner.
The study design can be divided into two major components: (i) establishment of baseline and tolerance limits based on the previously validated software, ART-Plan 2.1.0 and ii) the design development and deployment of QA protocols to commission and monitor ART-Plan 2.1.0 prior and during clinical implementation and the evaluation of ART-Plan 2.2.1.
In the first component, baseline values and tolerance limits were established for the selected OARs during the validation of ART-Plan 2.1.0. OARs were previously selected based on their agreement with MedAustron's in-house manual contouring. Specifically, in a quantitative analysis based on DSI and other geometrical metrics, contours were preselected and subsequently evaluated qualitatively by three clinicians. The qualitative assessment was conducted using the following scale: 0 = not usable, 1 = major correction required, 2 = minor correction required, and 3 = no correction required. At the end of this analysis, a set of 40 OARs was defined. The results of this analysis are shown in Figure 4.4 and Table 4.1. Based on this set of OARs, baseline values along with minimum acceptable limits—which must be maintained following any workflow changes—were established in accordance with the metrics designed for QA protocols (see next section).

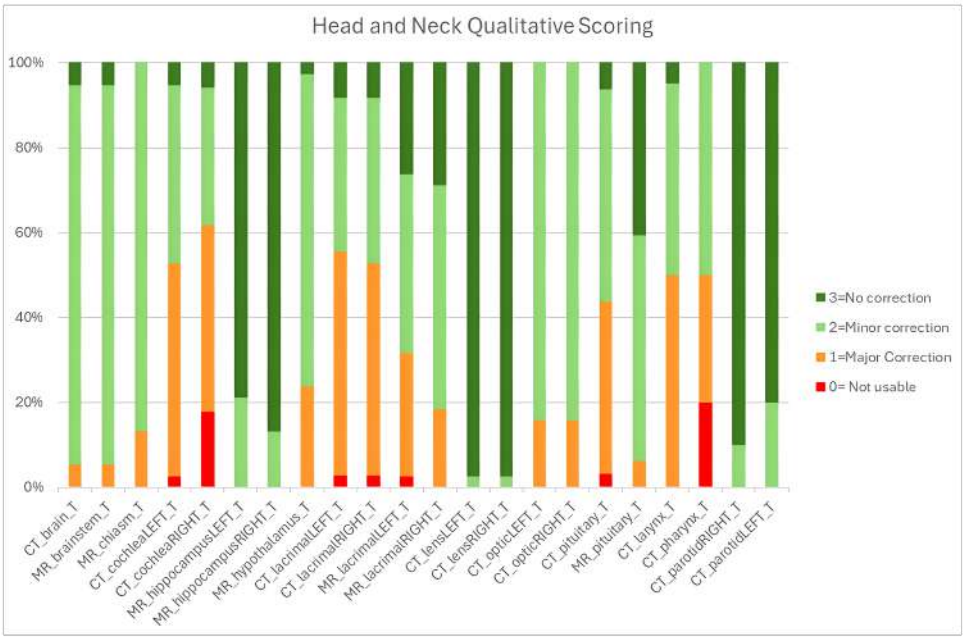
For the second component, three QA protocols were developed and tested. These protocols use Python-coded pipelines integrated with the TPS and are designed to ensure: Seamless and automated integration that supports routine routine functionality, referred to as *Clinical End-to-End QA*; long-term stability of automatic contouring referred to as *Periodic QA*; and performance consistency of new model releases with the established baseline, referred to as *Acceptance QA*.

Head and Neck (18)	**Thorax (7)**	**Abdomen (9)**	**Pelvis (6)**
brain	heart	bowelbag	anal canal
brainstem	lung L	bowelloops	bladder
bullbus L	lung R	colon	femoral head L
bullbus R	oesophagus	kidney L	femoral head R
chiasm	spinalcord	kidney R	rectum
cochlea L	thyroid	liver	sigma
cochlea R	trachea	spleen	
Larynx		stomach	
lacrimal gland L		spinal cord	
lacrimal gland R			
lens L			
lens R			
mandible			
opticus L			
opticus R			
parotis L			
parotis R			
pituitary			

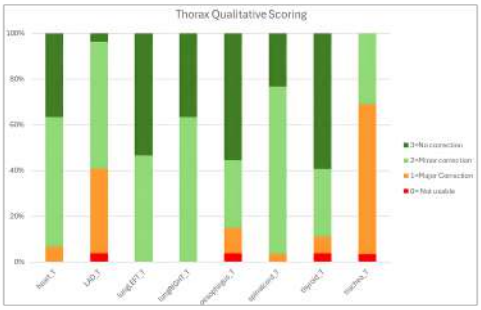
Table 4.1: Selected OARs per anatomical region.

Patient Data Preparation

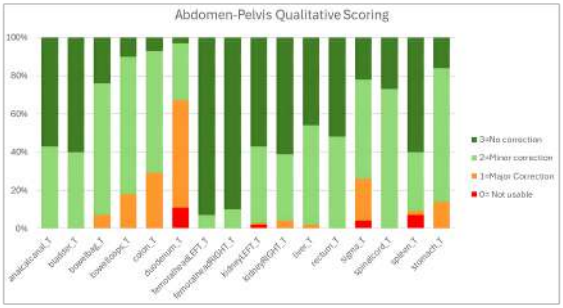
A dataset, consisting of in-house MedAustron data, was built to reflect auto-contouring performance on real-world clinical data. Five patient cases were created for each anatomical site (head & neck, thorax, abdomen, and pelvis) by combining and anonymizing twenty different patient CT scans from adult patients, both male and female, in which structures had been manually delineated by clinicians (serving as the ground truth). The aim was to obtain five different cases for each OAR. During data curation, only head-first supine (HFS) patient geometries were included, as the model was unable to segment organs in rotated or prone positions. The dataset was anonymized and stored in the clinical research database to prevent collisions and duplication within the clinical database. No further data or image processing methods were applied.



(a) Head and neck structures. For this anatomical site, ART-Plan/Annotate was also available for the MRI images, allowing the structures to be evaluated as well.



(b) Thorax structures.



(c) Abdomen and pelvis structures.

Figure 4.4: Qualitative evaluation of the structures for the different anatomical sites.

4.2.2 Metrics definition and Tolerances

Geometric Evaluation Metrics

The primary metric used was the DSI [161,162] defined in equation 1.4, which quantifies the overlap between auto-contours generated by TheraPanacea and reference segmentation done by clinicians.

Baseline has been set for each OAR as the mean value across five patients (see Section 4.2.1). A tolerance of one standard deviation (SD) from the baseline was used to define acceptable values, i.e. DSI acceptable limit = baseline - 1 SD. For the periodic QA, we defined a numerical error tolerance of 0.01 for DSI.

DSI was selected as the main evaluation metric.For borderline cases, where DSI might not be sufficient, dosimetrical metrics were added.

Dosimetric Evaluation

The primary evaluation metric was D_2 [163, 164], a near-maximum dose indicator for OARs. Accuracy was quantified using the relative absolute difference ΔD_2, defined as the percentage deviation between the baseline and automated contours:

$$\Delta D_2 = \frac{|D_{2,\text{manual}} - D_{2,\text{auto}}|}{D_{2,\text{manual}}} \cdot 100 \tag{4.1}$$

where $D_{2,\text{manual}}$ and $D_{2,\text{auto}}$ denote the D_2 values computed on the manual and automated contours, respectively. In this context, values closer to zero indicate better agreement. As done for the geometrical metrics, the baseline was defined as the average across patients. A tolerance threshold of 1% was adopted.

As a complementary measure, the mean squared error of the Dose Volume Histogram (DVH) bins (MSE_{DVH}) [163, 164] was used to evaluate the agreement between dose distributions. This metric was defined as follows.

For each OAR, the DVH was extracted from both the manually delineated contour and the automatically generated contour. Let $\text{DVH}_{\text{manual}}$ denote the reference DVH obtained from the manual segmentation, and DVH_{auto} the DVH derived from the automatic contour. To ensure comparability, both DVHs were discretized using identical dose binning. The dose range was divided into $K = 100$ bins with uniform width Δd. For each bin k, the corresponding volume fraction was computed. Prior to statistical comparison, DVHs were normalized to relative volume:

$$\sum_{k=1}^{K} e_k = 1, \quad \sum_{k=1}^{K} o_k = 1, \tag{4.2}$$

where e_k and o_k represent the relative frequencies (volume fractions) of the manual and automatic DVHs, respectively.

Similarity between $\text{DVH}_{\text{manual}}$ and DVH_{auto} was assessed using the chi-square (χ^2) goodness-of-fit statistic, under the null hypothesis that the manual DVH represents the reference (true) distribution. The test statistic is defined as:

$$\chi^2 = \sum_{k=1}^{K} \frac{(o_k - e_k)^2}{e_k}. \tag{4.3}$$

Under the null hypothesis, and assuming sufficiently large expected frequencies in each bin, the statistic approximately follows a chi-square distribution with $K - 1$ degrees of freedom.

Also in this case, the baseline was defined as the average value across patients. A tolerance level of 2% was defined for this metric.

The tolerance thresholds used for the dosimetric metrics were established through a sensitivity analysis. A tailored subset of OARs (based on DD) was selected, ensuring

that each structure received a clinically relevant dose ($D_2 > 1\,\mathrm{Gy}$). For each selected organ, a geometric transformation consisting of a 1 mm surface expansion was applied to the contours to assess its impact on D_2 and the DVH bins. The resulting variations were consistently below +1% for D_2 and +2% for MSE_{DVH}, supporting the chosen tolerance values.

Dosimetric metrics are inherently context-dependent, as the results strongly depend on the patient, the OAR, and the RT plan. For example, a non-irradiated OAR may yield zero dosimetric error even if the contour significantly differs from the GT. Their inclusion is motivated by the need to evaluate the dosimetric impact of using predicted contours instead of the GT. In some cases, even when a good overlap between prediction and GT is observed, large deviations in dose estimation may occur, particularly for OARs located close to the Planning Target Volume (PTV).
For this reason, these metrics should be interpreted as complementary to the geometrical analysis. Furthermore, since strong deviations were observed, Anal canal contour is not included in dosimetrical analysis.

4.2.3 TheraPanacea ARTPlan configuration

TheraPanacea ARTPlan (Paris, France) is a software platform dedicated to the development and commercialization of advanced, off-the-shelf solutions powered by DL for the automated segmentation of organs and lymph nodes in CT and MRI data used in treatment planning. Furthermore, the company is a certified Health Data Host (HDH) and holds ISO 27001 certification for Information Security Management and ISO 13485 for Quality Management Systems, as well as CE marking (Class IIb) and FDA/TGA clearance. The software, through its *Annotate* tool, automatically generates contours for more than 200 OARs and lymph nodes in approximately 3 minutes (full-body delineation). It was designed to be compatible with PACS-RIS [3] and DICOM[4] standards. In clinical use, user verification of the generated results is mandatory.

TheraPanacea ARTPlan is installed on a dedicated server, which users access through a web browser. Direct access to the underlying models is not permitted. CT queries[5] can be performed either manually—by uploading DICOM CT images via the platform—or semi-automatically by pushing DICOM data from the TPS, provided that a DICOM node has been configured between the TPS-PACS and the TheraPanacea dedicated server.

[3]In a medical and hospital setting, it is the IT system dedicated to the storage, transmission, and visualization of diagnostic images (such as X-rays, CT scans, and MRIs). While DICOM is the "language" (the file format) used by the machines to communicate, PACS serves as the "warehouse" and the "delivery system" that manages those files. Generally, PACS works in conjunction with the RIS (Radiology Information System) to link images with patient demographics and diagnostic reports.

[4]DICOM (Digital Imaging and Communications in Medicine) is a standard that ensures the interoperability of medical imaging devices by combining image data and patient metadata into a single unified file format.

[5]CT queries refer to the process of interrogating the model by providing CT image data as input, on which the auto-contours are generated.

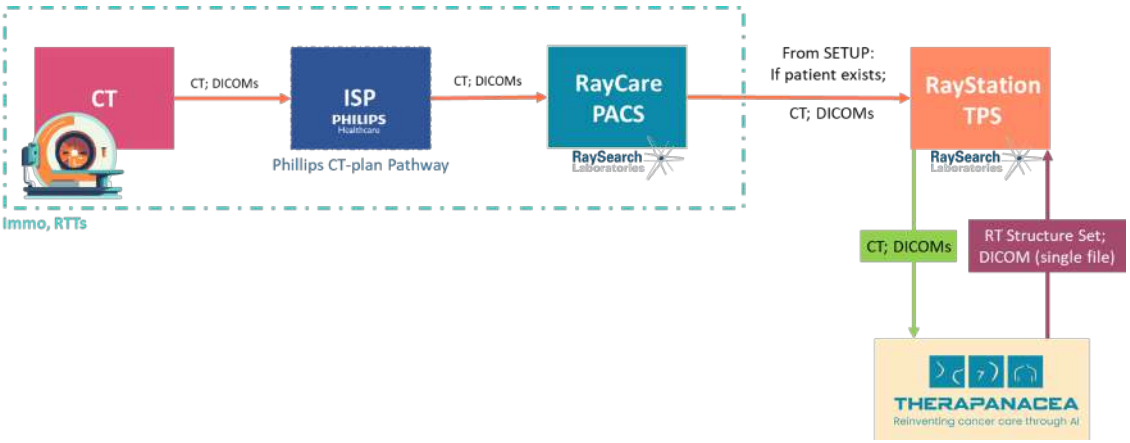


Figure 4.5: Process diagram of the MedAustron clinical TheraPanacea ARTPlan installation.

Figure 4.5 illustrates the TheraPanacea ARTPlan connectivity configuration installed at the MedAustron center. The process begins with the acquisition of a planning CT scan, which is subsequently stored in the Image Service Provider (ISP) CT portal planning pathway. The outputs are unidirectional and consist of DICOM data frames and sets. These data are then transferred to the RayCare PACS, from which it can be loaded into the RayStation treatment planning system. Patient data can only be accessed if the individual already exists in the database, ensuring that all ID tags are correctly assigned to each case. Through a DICOM node, the user can then push a CT dataset and retrieve an RT structure set from the ARTPlan server, where a bot automatically performs contouring using the annotate module. Additionally, usage statistics have been performed, since its deployment in October 2024. Between November and December, 91 usage cases were reported at MedAustron, followed by 66 cases from January to February 2025. The average time saved per immobilization is estimated at approximately 25%, based on a typical manual contouring time of 4.6 hours per CT scan. Based on this data, a projection would indicate the total manual contouring time reduction will continue to be reduced each year.

4.2.4 QA protocols design

To monitor ARTPlan clinical stability and the performance of AI models over different time intervals, three independent protocols were developed:

- **Clinical end-to-end:** Verifies the integrity of ARTPlan connectivity with RayStation. It serves as a debugging tool to distinguish user-side errors from those originating at the ARTPlan server or the DICOM node. It consists of running the auto-contouring workflow with privileged access—typically not available to standard users. If the semi-automatic CT query fails, a privileged user can attempt a manual query to determine whether the issue originates from the ARTPlan server or the DICOM node.

- **Periodic:** Evaluates the reproducibility of auto-contours generated by ARTPlan through DSI. It is intended to be performed quarterly (three times per year) and compares the current ARTPlan model against itself over time.
- **Acceptance:** Ensures completeness, consistency, and traceability through geometric DSI and dosimetric evaluations (ΔD_2, MSE_{DVH}) prior to the deployment of new ARTPlan models to the clinical server. To preserve the clinical workflow, TheraPanacea was requested to install new models on a dedicated test server.

4.2.5 Protocols acceptance criteria

Acceptance criteria apply for Periodic and Acceptance QAs only. They do not apply on Cllinical end to end QA since, as it was already mentioned, it serves as debugging tool.

Periodic QA

The following action levels for the periodic QA protocol define the severity of the QA outcomes and outline the corresponding actions to be taken.

1. **Pass:** ARTPlan Annotate produces identical contours (DSI = 1):
   (a) The software demonstrates reproducibility and stability.
      i. No further action is required.
2. **Warning:** ARTPlan produces contours that slightly differ from the baseline (acceptable limits not exceeded):
   (a) Possible numerical or export-related issues, while results remain consistent with the baseline.
      i. The user should check for potential data export errors.
3. **Fail:** ARTPlan produces contours that significantly differ from the baseline (acceptable limits exceeded):
   (a) Stability and reproducibility are not guaranteed.
      i. The user must investigate potential violations and errors in the AI-based auto-segmentation. The use of ARTPlan must be suspended.

Acceptance QA

Acceptance QA was designed to verify whether previously approved standards are maintained after models upgrade. This choice is motivated by generalization considerations. As discussed in Chapters 1 and 2, model validation—typically performed on an independent test set, different from the training set but still limited in size and variability—can

only provide an estimate of the generalization error. Therefore, although companies perform models validation prior to release, the reported improvements may not directly translate into better performance on real-world clinical data.
In this context, Acceptance QA aims to assess the impact of new models on in-house data. At the same time, it should not be interpreted as a full internal validation process, which would require in-depth analysis on sufficiently large datasets and consequently longer approval times. Rather, Acceptance QA is intended as a rapid check to ensure that quality standards are maintained. This requirement led to the adoption of several heuristic design choices, summarized as follows:

- **Limited test set:** A dataset consisting of only five patients is used to reduce the complexity and analysis time of the protocol.
- **Hierarchy among metrics:** Although the protocol employs multiple metrics, both geometrical and dosimetric, the final outcome must be objective and unambiguous. The use of multiple metrics, which may not always agree, can hinder this objective. Moreover, dosimetric metrics are inherently context-dependent, as described at the end of Section 4.2.2.
  This consideration motivated the definition of a primary metric—the DSI, which measures the overlap between prediction and GT—and supporting metrics, namely dosimetric metrics (due to their context-dependent nature). Accordingly, the Acceptance QA automatically provides an outcome for each OAR based on DSI, while the user performing the test is required to review the supporting metrics in borderline cases and make the final decision. In the latter case, when both dosimetrical metrics (ΔD_2 and MSE_{DVH}) provides better performance a positive score can be assigned to the relative contour.

The following action levels for the acceptance QA protocol clearly define the severity of the results and outline the corresponding consequences to be taken.

1. **Pass:** The new model is performing better than the baseline if:
   (a) DSI $>$ baseline
      i. No further action is required.
2. **Warning:** The new model is not performing better than the old one but not exceeding limits if:
   (a) acceptable limit $<$ DSI $<$ baseline
      i. The user must check supporting metrics.
3. **Fail:** The new model is performing worse than the baseline if:
   (a) DSI $<$ acceptable limit

i. The auto-segmentation performed by the AI bot must be revised by an expert for violations of clinical standards regarding contouring guidelines or plan constraints.

Acceptance QA applies to all OARs; therefore, a global decision must be made by one or more expert clinicians.

4.2.6 QA pipeline

In developing the QA protocols described above, significant effort was devoted to (i) implementing pipelines to translate the protocol designs into computational procedures integrated with the TPS, and (ii) making these pipelines accessible to non-expert users. This was not applied to the clinical end-to-end protocol, which essentially consists of running ARTPlan with privileged settings.
Pipelines were developed for both Periodic and Acceptance QA. For each protocol, two sub-pipelines can be identified: pre-processing and post-processing. Both rely on Python code, using libraries such as NumPy, Matplotlib, and pandas [90, 165–167], along with the TPS-integrated Python module. The ensemble of pipelines have been loaded as a repository on the MedAustron dedicated GitHub account, accessible under request.

Pre-processing pipeline

This pipeline consists of three nested loops over anatomical site, patient, and OAR. At each iteration, metric calculation is performed within the TPS. In Acceptance QA, contours generated by the new models are compared against the GT; in Periodic QA, following a new ARTPlan CT query, the generated contours are compared against those obtained from a previous query. At the end of the process, for each anatomical site, an Excel file is produced containing the results for each patient and each OAR (DSI, ΔD_2, and MSE_{DVH} for Acceptance QA; DSI only for Periodic QA). Figure 4.6 depicts the developed GUI.

The pipeline was developed with a simple and user-friendly Graphical User Interface (GUI). Upon execution, the user is prompted to provide the TPS version, anatomical site, and output folder path.

Post-processing pipeline

This pipeline consists of processing the raw results generated by the pre-processing pipeline and verifying the acceptance criteria described in the previous section. In the case of Acceptance QA, it additionally produces summary statistics and graphical outputs, which are then collected and used to automatically generate a report in Word format (see Figure 4.7 and Appendix A for more details). In this case, the pipeline can be executed using a pre-configured Jupyter notebook [168].

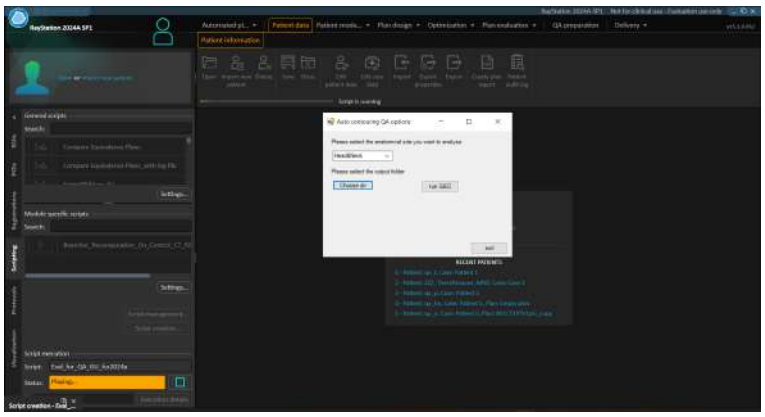


Figure 4.6: GUI developed inside the RayStation TPS for the pre-processing pipeline.

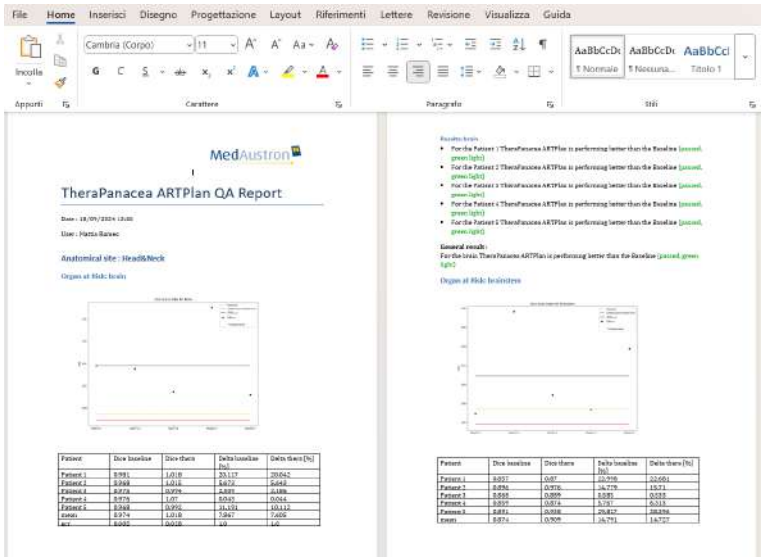


Figure 4.7: QA report file example produced by the post-processing pipeline in Acceptance QA.

4.3 Results

In this section, the results of the Acceptance QA, performed by comparing ARTPlan v2.2.1 against the previously approved ARTPlan v2.1.0, are reported. For the other QA protocols, only a brief discussion is provided below.

The clinical end-to-end QA was applied several times; however, it proved particularly useful during the installation of ARTPlan v2.2.1 on the test server. The installation was carried out remotely by TheraPanacea, with constant support from the in-house IT team. During the installation, an error caused a system crash, and the clinical end-to-end protocol was used as a debugging tool to support MedAustron's IT team.

Periodic QA was applied 4–5 times, consistently passing the stability and reproducibility tests. However, it should be noted that the tolerance thresholds—related to Periodic QA—described in the Materials and Methods section were defined following the observation of slight differences (DSI < 0.01) during the protocol development phase. In particular, these deviations were observed after a RayStation TPS upgrade (from v11 to v2024a). Given their very small magnitude, these deviations can be attributed to

numerical effects; nevertheless, they highlight that not only changes in the AI software, but also updates to the TPS, may lead to fluctuations in auto-contouring performance.

4.3.1 Acceptance QA results

DSI comparison of ART-Plan versions 2.1.0 and 2.2.1 (Tables 4.2, 4.3, 4.4, and 4.5) showed that the updated models: i) improved performance for 10 out of 40 OARs, ii) showed no change for 20 out of 40 OARs, and iii) performed worse for 10 out of 40 OARs. Additionally, the thyroid was not detected in one QA patient dataset.
ΔD_2 comparison showed that the updated models (Tables 4.6, 4.7, 4.8, and 4.9): i) improved performance for 13 out of 39 OARs, ii) showed no change for 6 out of 39 OARs, and iii) performed worse for 20 out of 39 OARs.
MSE_{DVH} comparison showed that the updated models (Tables 4.6, 4.7, 4.8, and 4.9): i) improved performance for 14 out of 39 OARs, ii) showed no change for 2 out of 39 OARs, and iii) performed worse for 23 out of 39 OARs.
Among the 20 ROIs showing no change in the DSI analysis, three OARs (right bulbus, pituitary, and left kidney) exhibited improved performance in both dosimetric metrics. Among the 10 ROIs with improved DSI, only three OARs belonged to the low DSI range (baseline DSI < 0.7).

These findings indicate that ART-Plan 2.2.1 does not demonstrate consistent improvement over the previous version and requires further investigation. A closer examination of how Annotate interprets the received anatomical site revealed that this version applies stricter constraints than the previous release. As a result, organs located at the edges of the planning CT are not contoured unless they are fully captured by the imaging protocol, which explains why the thyroid was missing.

Head and Neck OARs	DSI baseline	DSI Lowest acceptable limit	DSI ART-Plan v2.2.1 vs. Manual
brain	0,974	0,969	0,979
brainstem	0,874	0,858	0,858
bulbus L	0,942	0,936	0,937
bulbus R	0,947	0,930	0,943
chiasm	0,451	0,369	0,379
cochlea L	0,626	0,555	0,585
cochlea R	0,649	0,491	0,589
larynx	0,857	0,833	0,779
lacrimal gland L	0,494	0,382	0,470
lacrimal gland R	0,612	0,573	0,610
lens L	0,865	0,817	0,847
lens R	0,883	0,868	0,849
mandible	0,915	0,892	0,898
opticus L	0,772	0,746	0,743
opticus R	0,755	0,704	0,722
parotis L	0,888	0,867	0,863
parotis R	0,896	0,889	0,865
pituitary	0,573	0,504	0,541

Table 4.2: Head & Neck Acceptance QA results. Green = pass, Yellow = warning, Red = fail. The results for model v2.2.1 were the same as v2.3.2.

Thorax OARs	DSI baseline	DSI Lowest acceptable limit	DSI ART-Plan v2.2.1 vs. Manual
heart	0,941	0,915	0,943
lung L	0,975	0,971	0,963
lung R	0,968	0,951	0,961
oesophagus	0,696	0,519	0,775
spinalcord	0,795	0,761	0,757
thyroid	0,694	0,546	0,650
trachea	0,562	0,329	0,697

Table 4.3: Thorax Acceptance QA results. Green = pass, Yellow = warning, Red = fail.The results for model v2.2.1 were the same as v2.3.2.

Abdomen OARs	**DSI baseline**	**DSI Lowest acceptable limit**	**DSI ART-Plan v2.2.1 vs. Manual**
bowel bag	0,896	0,867	0,895
bowel loops	0,799	0,742	0,800
colon	0,840	0,766	0,840
kidney L	0,930	0,903	0,928
kidney R	0,904	0,828	0,770
liver	0,960	0,951	0,944
spleen	0,944	0,929	0,946
stomach	0,904	0,888	0,904
spinal cord	0,741	0,642	0,781

Table 4.4: Abdomen Acceptance QA results. Green = pass, Yellow = warning, Red = fail.The results for model v2.2.1 were the same as v2.3.2.

Pelvis OARs	**DSI baseline**	**DSI Lowest acceptable limit**	**DSI ART-Plan v2.2.1 vs. Manual**
analcanal	0,305	0,077	0,319
bladder	0,928	0,874	0,927
femoralhead L	0,952	0,922	0,942
femoralhead R	0,957	0,932	0,943
rectum	0,745	0,623	0,748
sigma	0,710	0,597	0,711

Table 4.5: Pelvis Acceptance QA results. Green = pass, Yellow = warning, Red = fail.The results for model v2.2.1 were the same as v2.3.2.

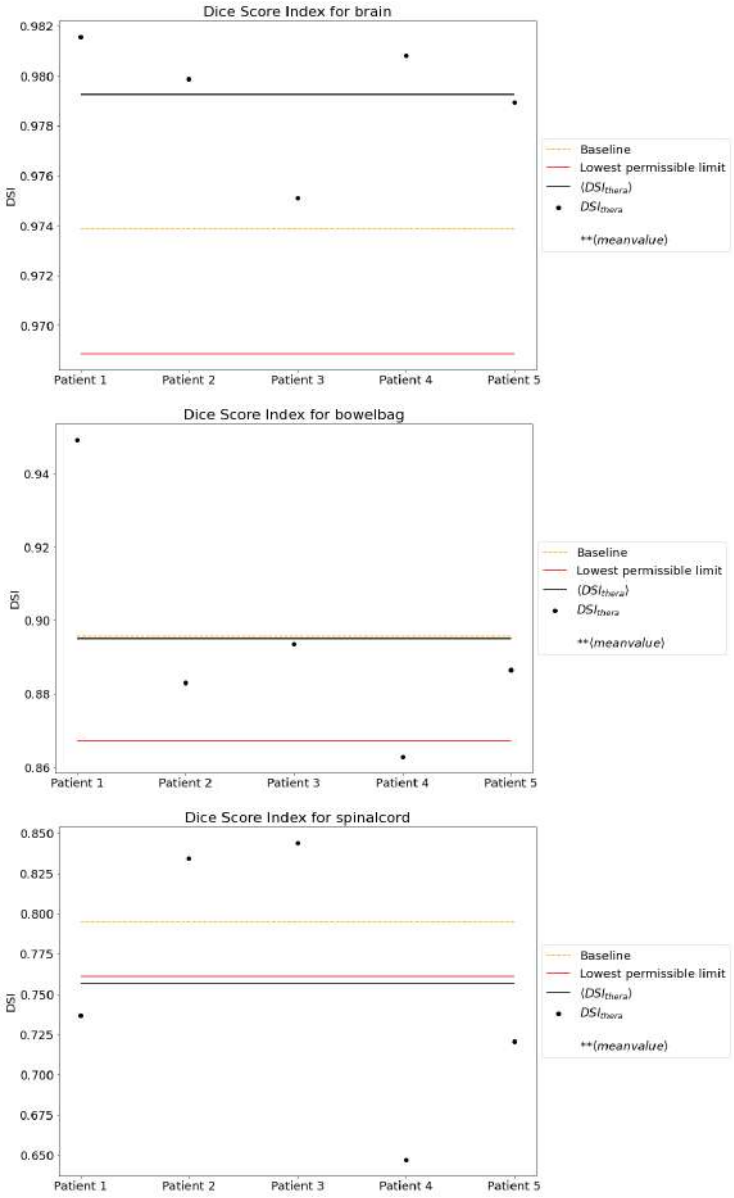


Figure 4.8: Examples of results, from top to bottom pass, warning and fail cases. In yellow the baseline values ($DSI = 0.974-0.896-0.795$) and in red the lowest acceptable limit ($DSI = 0.969-0.867-0.761$), derived from the initial evaluation. In black the mean DSI score obtained using ART-Plan 2.2.1 ($DSI = 0.979 - 0.895 - 0.757$).

Head and Neck OARs	ΔD_2[%] **baseline**	ΔD_2[%] **Highest acceptable limit**	ΔD_2[%] **ART-Plan v2.2.1**	MSE_{DVH} **baseline**	MSE_{DVH} **Highest acceptable limit**	MSE_{DVH} **ART-Plan v2.2.1**
brain	7.867	8.867	8.071	1.66e-05	1.69e-05	1.18e-05
brainstem	14.792	15.792	19.729	2.99e-04	3.05e-04	1.09e-03
bulbus L	1.721	2.721	1.092	7.27e-05	7.41e-05	7.73e-05
bulbus R	1.46	2.46	1.39	1.10e-04	1.12e-04	9.77e-05
chiasm	38.287	39.287	23.805	1.42e-02	1.45e-02	2.95e-02
cochlea L	6.067	7.067	6.889	1.10e-03	1.12e-03	3.46e-03
cochlea R	2.052	3.052	11.115	6.33e-03	6.45e-03	9.91e-03
larynx	3.693	4.693	47.666	1.54e-04	1.57e-04	1.95e-03
lacrgland L	39.134	40.134	55.445	1.85e-02	1.88e-02	2.23e-02
lacrgland R	31.743	32.743	36.424	1.68e-02	1.72e-02	1.50e-02
lens L	3.369	4.369	6.275	1.07e-03	1.09e-03	1.08e-03
lens R	1.43	2.43	2.201	4.25e-04	4.33e-04	9.97e-04
mandible	2.942	3.942	4.742	4.12e-04	4.20e-04	2.36e-04
opticus L	10.279	11.279	16.348	8.07e-04	8.24e-04	1.54e-03
opticus R	5.272	6.272	10.164	3.85e-03	3.93e-03	3.13e-03
parotis L	15.453	16.453	17.79	4.87e-04	4.97e-04	7.92e-04
parotis R	4.918	5.918	3.776	4.13e-04	4.21e-04	4.27e-04
pituitary	5.721	6.721	4.128	7.35e-03	7.50e-03	7.22e-03

Table 4.6: Head & Neck Acceptance QA results. Green = pass, Yellow = warning, Red = fail.

Thorax OARs	ΔD_2[%] **baseline**	ΔD_2[%] **Highest acceptable limit**	ΔD_2[%] **ART-Plan v2.2.1**	MSE_{DVH} **baseline**	MSE_{DVH} **Highest acceptable limit**	MSE_{DVH} **ART-Plan v2.2.1**
heart	9.19	10.19	7.30	2.84e-05	2.89e-05	1.85e-05
lung L	2.05	3.05	2.94	2.14e-06	2.18e-06	4.93e-06
lung R	0.92	1.92	2.19	4.76e-05	4.86e-05	7.59e-05
oesophagus	4.13	5.13	2.92	1.10e-02	1.12e-02	4.73e-04
spinal Cord	3.92	4.92	7.32	1.77e-03	1.81e-03	2.92e-03
thyroid	13.26	14.26	16.02	1.97e-02	2.01e-02	2.56e-02
trachea	30.43	31.43	18.49	9.11e-03	9.29e-03	5.91e-03

Table 4.7: Thorax Acceptance QA results. Green = pass, Yellow = warning, Red = fail.

Abdomen OARs	ΔD_2[%] **baseline**	ΔD_2[%] **Highest acceptable limit**	ΔD_2[%] **ART-Plan v2.2.1**	MSE_{DVH} **baseline**	MSE_{DVH} **Highest acceptable limit**	MSE_{DVH} **ART-Plan v2.2.1**
bowelbag	81.952	82.952	80.925	9.92e-04	1.01e-03	9.96e-04
bowelloops	14.534	15.534	15.443	2.22e-04	2.27e-04	2.16e-04
colon	8.525	9.525	8.941	6.19e-05	6.31e-05	6.03e-05
kidney L	4.301	5.301	2.24	4.69e-05	4.79e-05	3.24e-05
kidney R	2.074	3.074	9.724	2.61e-04	2.66e-04	1.13e-03
liver	5.66	6.66	7.943	1.78e-05	1.82e-05	4.08e-05
spleen	56.987	57.987	20.047	2.15e-05	2.19e-05	1.15e-05
stomach	17.959	18.959	17.659	4.16e-05	4.24e-05	4.35e-05
spinalcord	14.703	15.703	14.024	2.58e-02	2.63e-02	2.31e-02

Table 4.8: Abdomen Acceptance QA results. Green = pass, Yellow = warning, Red = fail.

Pelvis OARs	ΔD_2[%] **baseline**	ΔD_2[%] **Highest acceptable limit**	ΔD_2[%] **ART-Plan v2.2.1**	MSE_{DVH} **baseline**	MSE_{DVH} **Highest acceptable limit**	MSE_{DVH} **ART-Plan v2.2.1**
bladder	1.686	2.686	21.655	4.85e-04	4.94e-04	1.30e-02
femoralhead L	0.986	1.986	21.078	1.10e-03	1.12e-03	5.26e-02
femoralhead R	0.372	1.372	20.348	5.06e-04	5.16e-04	2.41e-02
rectum	4.745	5.745	24.418	6.74e-03	6.88e-03	9.01e-02
sigma	28.936	29.936	32.953	6.26e-03	6.39e-03	8.84e-03

Table 4.9: Pelvis Acceptance QA results. Green = pass, Yellow = warning, Red = fail. As it was already mentioned at the end of the Section 4.2.2, anal canl ROI is not included in dosimetrical analysis.

4.4 Discussion

The clinical implementation of AI solutions is a complex challenge that must take into account technological, ethical, and regulatory considerations. Currently, there is a noticeable gap in the literature on this topic, largely due to the rapid growth of AI in recent years.

The use of AI tools in medical environments requires that specialists be able to evaluate and test their outcomes. In the field of OAR segmentation, incompatibilities can arise from various causes, primarily mismatches between in-house guidelines and those

adopted in the dataset used to train the AI model. As an example, in the initial evaluation of ARTPlan v2.1.0 it was found that improvements could be achieved by merging the trachea and carina ART-Plan labels to obtain the corresponding trachea MedAustron label. Additionally, the datasets and regularization techniques (e.g. data augmentation) used during training have a strong impact on the contours produced by an AI tool. A preliminary study on the software showed a significant reduction in the ART-Plan's performance for OAR contouring in rotated patients, indicating that non-standard patient geometries were not included during model training. While this issue may be negligible in conventional RT, in particle therapy—where beam lines are fixed and patients must be positioned in a rotated orientation—the time savings offered by the software becomes minimal. This issue could be addressed using transfer learning strategies. In particular, fine-tuning the model on rotated patients could significantly improve performance.

It should be underlined that, before implementing an AI tool in the clinics, a feasibility study must be carefully conducted—comparing in-house needs with the software's capabilities—before implementing an AI solution in clinical practice, even if it has already obtained all required certifications. European AI-act, outlines just the requirements for hospitals, clinics, and healthcare professionals using such systems in patient care. In the next future, the inclusion of feasibility study requirements may also be introduced.

Although AI serves as a valuable tool for organ segmentation in radiotherapy, its outputs cannot be accepted without scrutiny. A robust traceability framework is essential to monitor variations and ensure the stability of segmentation results. To the authors' knowledge, this is the first fully automated and specialized pipeline designed for QA of auto-segmentation tools in RT. In its design, the author emphasized the need for QA protocols that address the most critical aspects of ART-Plan's functionality.

Access to ART-Plan for clinicians depends on the status of the DICOM node, which connects TheraPanacea's server with the TPS systems. Any issue with the DICOM node may compromise the functionality of the tool, preventing users from utilizing auto-contouring. The *Clinical end to end* test was developed to enable experts—who typically have more security privileges than clinicians—to distinguish between issues caused by local failures and those associated with the DICOM node. From another perspective, the *Clinical end to end* test also functions as the first debugging tool: when it fails, technicians must intervene to resolve the issue.

Result stability must be ensured—an auto-contouring system cannot produce different outputs for the same input. The *Periodic QA* test was designed to guarantee the stability and reproducibility of results. The import/export procedure can introduce small variations; therefore, a tolerance value is incorporated to account for these. When this tolerance is exceeded, technicians must intervene to investigate potential issues.

Finally, model upgrades are often perceived as improvements. However, this is not always the case, as demonstrated in our comparison of ART-Plan versions v2.1.0 and

v2.2.1. Model training is performed on datasets that are limited in size, and the declared performance may not necessarily translate to clinical settings. Unfortunately, this limitation cannot be fully eliminated. It is precisely for this reason that the *Acceptance QA* test was developed. Moreover, upgrades can lead to software failures. The author proposed the use of an additional test server to preserve the clinical workflow during upgrade.
TheraPanacea claimed that ART-Plan v2.3.1 is significantly faster than the previous versions due to the use of multi-output models. Indeed, a reduction in processing time was observed in MedAustroun; however, the models' performance were not sufficient to justify approval of the upgrade.
From the *Acceptance QA* test we performed, we conclude that model upgrades cannot be blindly accepted. A rapid test must be conducted before any changes to the clinical workflow are implemented. If the test fails, experts must intervene and initiate a new evaluation process.
An interesting fact was retrived successively, when TeraPanacea released a new ARTPlan version, v2.3.1. *Acceptance QA* was repeated and exactly the same results achieved in by v2.2.1, indicating that upgrades were not regarding AI models supporting the companies declaration. This finding, once again showed the importance of a tool such as Acceptance QA.

Regarding the evaluation metrics authors used for the Acceptance QA something must be clarified. From the point of view of the RT planner a good geometrical overlap between 2 ROIs is not enough to allow exchange between them. When OARs have small sizes, order of some millimeters, even small differences can result in big dose differences, especially if the organ is close to the PTV. On the other hand, dose metrics are RT-plan dependent. Patients underwent to different treatments, which were patient-specific planned. This aspect led to subjective evaluation metrics that cannot be used in automatic pipelines. For this reason the automatic decision system in Acceptance QA protocol we developed make use only of DSI metric, which is an objective metric. Other metrics must be considered when new model performance tests return warning and, however, they requires a human interpretation.

The clinical implementation of AI solutions is a complex challenge that must take into account technological, ethical, and regulatory considerations. Currently, there is a noticeable gap in the literature on this topic, largely due to the rapid growth of AI in recent years.
The use of AI tools in medical environments requires specialists to be able to evaluate and validate their outputs. In the field of OAR segmentation, inconsistencies may arise from various sources, primarily mismatches between in-house guidelines and those adopted in the dataset used to train the AI model. For example, in a previous study (ARTPlan v2.1.0 initial evaluation), it was observed that improvements could be achieved by merging the trachea and carina ART-Plan labels to match the corresponding

MedAustron trachea label.
Additionally, the datasets and regularization techniques (e.g., data augmentation) used during training have a strong impact on the contours produced by an AI tool. A preliminary study of the software showed a significant reduction in ART-Plan performance for OAR contouring in rotated patients, indicating that non-standard patient geometries were not included during model training. While this issue may be negligible in conventional RT, in particle therapy—where beam lines are fixed and patients must be positioned in rotated orientations—the time savings offered by the software become minimal. It should be emphasized that, before implementing an AI tool in clinical practice, a feasibility study must be carefully conducted to compare in-house needs with the software's capabilities, even if the tool has already obtained all required certifications. The European AI Act outlines requirements for hospitals, clinics, and healthcare professionals using such systems in patient care. In the future, the inclusion of mandatory feasibility study requirements may also be introduced.

Although AI represents a valuable tool for organ segmentation in RT, its outputs cannot be accepted without scrutiny. A robust traceability framework is essential to monitor variations and ensure the stability of segmentation results. To the authors' knowledge, this is the first fully automated and specialized pipeline designed for QA of auto-segmentation tools in RT. In its design, particular emphasis was placed on QA protocols addressing the most critical aspects of ART-Plan functionality.

Access to ART-Plan for clinicians depends on the status of the DICOM node, which connects TheraPanacea's server with TPS systems. Any issue with the DICOM node may compromise the functionality of the tool, preventing users from performing auto-contouring. The Clinical end-to-end test was developed to enable experts—who typically have higher access privileges than clinicians—to distinguish between issues caused by local failures and those associated with the DICOM node. From another perspective, the Clinical end-to-end test also functions as a primary debugging tool: when it fails, technical intervention is required.

Result stability must be ensured—an auto-contouring system should not produce different outputs for the same input. The Periodic QA test was designed to guarantee the stability and reproducibility of results. The import/export procedure may introduce small variations; therefore, a tolerance value is incorporated to account for these. When this tolerance is exceeded, technical investigation is required.

Finally, model upgrades are often perceived as improvements. However, this is not always the case, as demonstrated in our comparison of ART-Plan versions v2.1.0 and v2.2.1. Model training is performed on datasets that are limited in size, and the declared performance may not necessarily translate to clinical settings. This limitation cannot be entirely eliminated. For this reason, the Acceptance QA test was developed. Moreover, upgrades may introduce software failures. The use of an additional test server was proposed to preserve the clinical workflow during upgrades.

TheraPanacea reported that ART-Plan v2.2.1 is significantly faster than previous versions due to the use of multi-output models. A reduction in processing time was indeed observed at MedAustron; however, the model performance was not sufficient to justify approval of the upgrade.

Based on the performed Acceptance QA tests, we conclude that model upgrades cannot be accepted without verification. A rapid validation test must be conducted before introducing any changes into the clinical workflow. If the test fails, experts must intervene and initiate a new evaluation process.

An additional observation was made when TheraPanacea released ART-Plan v2.3.1. The Acceptance QA test was repeated, yielding results identical to those obtained with v2.2.1, indicating that the upgrade did not involve changes to the AI models, in accordance with the company's claims. This finding further highlights the importance of a QA tool such as the Acceptance QA.

Regarding the evaluation metrics used for the Acceptance QA, some clarification is needed. From the perspective of the RT planner, a good geometrical overlap between two ROIs is not sufficient to allow interchangeability. When OARs are small (on the order of a few millimeters), even minor differences can result in significant dose variations, especially when the organ is close to the PTV. On the other hand, dose-based metrics depend on the specific RT plan, which is patient-specific. This aspect introduces subjectivity and limits their use in automated pipelines. For this reason, the automatic decision system within the Acceptance QA protocol relies solely on the DSI metric, which is objective. Other metrics should be considered when performance tests raise warnings; however, their interpretation requires expert evaluation.

Finally, it is important to emphasize the need for a specialized professional figure responsible for the clinical implementation of AI tools. At MedAustron, the TPS is typically upgraded twice per year. Although QA protocols have been defined and pipelines have been developed, compatibility must be continuously ensured. This requires professionals with expertise in both clinical and informatics aspects. At present, this role can generally be fulfilled by a medical physics specialist, who should also have competencies in artificial intelligence and machine learning methodologies.

Concluding, it is the author's personal opinion that AI software of this kind should be commercialized with the possibility of fine-tuning the model based on in-house data. In fact, as already mentioned, one of the main limitations of DL-based auto-contouring software is the variability of guidelines across different centers. Expecting to train a single perfect model capable of satisfying all commercial needs is not feasible. Instead, offering pre-trained models that can be fine-tuned according to in-house guidelines would provide greater reliability.

4.5 Conclusion and future perspectives

The present study highlights the importance of developing and implementing QA protocols for AI-powered tools in clinical practice. One of the most relevant findings is that model upgrades cannot be blindly accepted without proper verification. Reliability, stability, and traceability must be continuously ensured in clinical environments, which further motivates research in this field. In this context, QA frameworks play a crucial role in guaranteeing that AI-based solutions remain consistent and safe when deployed in real clinical workflows. In particular, the proposed Acceptance QA approach demonstrated its usefulness in detecting inconsistencies between different software versions and in supporting informed clinical decisions regarding model updates.

Future work may focus on the development and integration of in-house models specifically tailored to unconventional radiotherapy setups, where standard training datasets may not fully capture patient variability. Moreover, further investigations could explore adaptive fine-tuning strategies and continuous learning approaches, aiming to improve robustness across different clinical protocols and institutions.

CHAPTER 5

PHYSICS INFORMED NEURAL NETWORK IN FRICKE DOSIMETRY

Dosimetry is a branch of physics that aims to quantify the damage induced by ionizing radiation in both materials and biological matter. The physical quantity that best describes this phenomenon is the *Absorbed Dose*, defined as the mean *imparted energy* per unit mass. Its unit is the Gray ($\mathrm{Gy} = \mathrm{Jkg}^{-1}$).
In human dosimetry, radiation weighting factors are introduced to account for the fact that different types of radiation produce different biological effects for the same absorbed dose, leading to the definition of the *equivalent dose*. In addition, to account for the different sensitivities of tissues to ionizing radiation, the equivalent dose is weighted using tissue weighting factors. The resulting quantity is the *effective dose*, measured in Sieverts (Sv).

Measurements of absorbed dose require an estimation of the energy deposited in matter. This can be achieved by exploiting radiation–matter interactions. One of the most common outcomes of these interactions is charge production in the medium. Therefore, a simple detector operates by measuring the amount of charge produced by radiation within a small volume, referred to as the detector active volume [169]. If a system is first established to collect the charge once it has been produced—for example, by applying an electric field between two electrodes—and a system is then used to measure this charge—for example, by measuring the resulting current—the deposited energy can be evaluated, provided that the mean energy required to produce an ion pair is known. This is the case for detector systems such as ionization chambers and semiconductor detectors. In both cases detectors are referred to as *active* or *real-time*. Other detector systems are based on the trapping of radiation-induced charge carriers. In these systems, the crystal structure of solid-state materials is engineered in such a way that excited electrons (and holes) can be captured in metastable energy levels, often referred to as trapping centers. Upon the subsequent supply of external energy—either thermal or optical—these charge carriers are released from the traps and recombine,

emitting light. The intensity of this luminescence signal is proportional to the absorbed dose [170]. This is the case for thermoluminescent dosimeters (TLDs) and optically stimulated luminescence (OSL) dosimeters. In this case, dosimeters are referred to as *passive*, since the measurement is performed subsequently using an external readout system.
Similar to TLDs and OSLs, scintillators operate by converting excitation energy into a light signal; however, in this case the process occurs promptly.
Another class of dosimeters is represented by calorimeters, in which the absorbed energy is evaluated through the measurement of a temperature increase. This approach is experimentally challenging [171], since a dose of 1 Gy produces a temperature rise of only about 0.00024 °C in $1g$ of water.
Finally, another class of dosimeters exploits radiation-induced chemical changes. In particular, the production of free radicals in matter is proportional to the absorbed dose. In radiochromic film dosimeters, free radicals induce polymerization, leading to a variation in optical density that is proportional to the absorbed dose. In electron paramagnetic resonance (EPR) dosimetry, such as alanine dosimetry, the concentration of free radicals is quantified through EPR measurements.

Dosimeters can be characterized through different metrological parameters, such as accuracy and precision, linearity, sensitivity, energy and dose-rate dependence, tissue equivalence, reusability, and fading (for passive dosimeters), among others.
An important characteristic of a dosimeter concerns its capability to map dose distributions. Most dosimeters belong to the class of point detectors, meaning that the dose is estimated only locally. As discussed in Chapter 4, in radiotherapy (RT) applications the goal is the estimation of spatial dose distributions. Although ionization chambers are considered the standard in many applications, they do not allow direct measurement of dose distributions. A possible solution is to employ arrays of dosimeters; this approach has been widely implemented using ionization chamber arrays, as well as alanine and silicon-based detectors (e.g., diode arrays) [172–174]. Conversely, radiochromic films allow for two-dimensional (2D) dose measurements, but do not provide information on the dose distribution in depth. In contrast, Fricke gel (FG) dosimeters are intrinsically three-dimensional (3D) detectors, enabling the measurement of volumetric dose distributions.

FGs belong to the class of chemical dosimeters and were introduced following the work of Hugo Fricke [175]. A Fricke solution consists of an aqueous solution of ferrous sulfate. Ionizing radiation interacts with this solution by oxidizing Fe^{2+} into Fe^{3+} ions, either directly or via intermediate water-derived free radicals. The solution can be gelled using matrices such as gelatin, agarose, or polyvinyl alcohol (PVA) [171]. As a result, the absorbed dose is proportional to the concentration of Fe^{3+} ions, and this relation holds locally.
Since Fe^{3+} ions exhibit paramagnetic properties, the gels can be imaged using MRI,

enabling the mapping of three-dimensional (3D) dose distributions [176]. Alternatively, the ferric ion concentration can be evaluated through spectrophotometric absorbance measurements when a suitable chelating agent is added to the gel. In this case as well, 3D dose distributions can be reconstructed, for example, using optical tomography techniques [177].
A key advantage of gel dosimetry is that the dosimeter itself also acts as a phantom; moreover, Fricke gels exhibit tissue-equivalent properties, making them particularly promising for complex clinical scenarios in which gels can be shaped into anthropomorphic phantoms. [171].
Despite their advantages, FG dosimeters suffer from a major limitation: the continuous post-irradiation diffusion of ions, which results in a blurring of the recorded dose distribution [171]. Additionally, especially in the presence of a chelating agent—such as xylenol orange (XO)—FGs undergo auto-oxidation (AO) processes, in which Fe^{2+} is spontaneously oxidized to Fe^{3+} even in the absence of irradiation, due to the greater stability of the ferric state [178]. These phenomena have significantly limited the use of FGs in RT, restricting their application to a relatively narrow range of scenarios [179]. This limitation is further exacerbated by practical constraints related to measurement logistics. MRI scanners may not be readily available, and irradiation and measurement facilities are often located in different places. Moreover, Olsson *et al.* reported that the usable time window of FG dosimeters—respecting the RT standards—is on the order of approximately 2 hours, due to diffusion effects (diffusion coefficient $\sim$ 1–2 mm^2 h^{-1}). These limitations highlight the need for advanced correction and modeling strategies to mitigate diffusion and auto-oxidation effects, thereby enabling a more reliable use of FG dosimeters in complex radiotherapy applications.

While most research on FGs has focused on materials science, aiming to identify chemical compositions with reduced ion mobility, the author's work explored the use of AI as a post-processing method to recover dose distributions from their blurred counterparts. In an initial, unsuccessful attempt, a denoising approach (see Section 1.2.1) was investigated, exploiting synthetic datasets generated through numerical simulations.
Preliminary results showed that the trained models failed to capture the underlying physics. For example, in the temporal evolution of a dose distribution affected solely by diffusion, the integral of the distribution is conserved; however, this property was not preserved in the model outputs. This observation motivated the author to investigate methods for embedding physical laws into the training process. A straightforward literature review led to the adoption of Physics-Informed Neural Networks (PINNs), described in Section 2.4.

In this chapter, the author's research on the application of PINNs to address diffusion in FGs is presented, together with the obtained results. The study can be divided into three main steps: (i) a feasibility study on simulated one-dimensional data,

aimed at identifying an optimized hyperparameter configuration and validated on 1D experimental data [180]; (ii) extension of the method to 2D and 3D simulated dose distributions [181]; and (iii) application of the method to experimental FG data affected also by AO, including a case study on a 2D small-field irradiation distribution.
The entire study required the definition of a mathematical framework to support the obtained results and to establish the limits of applicability, which are also presented in this chapter.

5.1 Fricke gel dosimetry

5.1.1 Gel compound

As already mentioned, the functionality of Fricke dosimeters is based on the oxidation of Fe^{2+} into Fe^{3+} in an aqueous solution of iron salts [175]. To preserve the spatial distribution of ferric ions, it is essential to gel the solution.
Historically, natural compounds—such as animal gelatin and agarose—were the most widely used matrices for FGs. However, due to several limitations, mainly related to reproducibility, research has increasingly focused on synthetic matrices based on polyvinyl alcohol. PVA is a polymer with a simple chemical structure; it is water-soluble, non-toxic, and inexpensive. It is also suitable for hydrogel production via physical methods—such as freeze–thaw cycles—or chemical routes, for example using crosslinking agents such as glutaraldehyde (GTA). Several properties of GTA-PVA gels depend on the degree of crosslinking and can be readily tuned by adjusting the PVA concentration, its molecular weight, and the GTA/PVA ratio [178]. Further modifications have been investigated to limit ion mobility. In particular, the addition of chelating agents—molecules capable of forming two or more coordination bonds with a metal ion, such as XO or methylthymol blue (MTB)—can reduce ion diffusion within the gel [182, 183]. Moreover, the presence of chelating agents makes the gel absorbance sensitive to Fe^{3+} concentration, enabling dose estimation via spectrophotometry [184] (see Figure 5.1).

5.1.2 Radiation chemistry in Fricke gel

Since the amount of inorganic reagents in the gels is negligible compared to the overall composition, ionizing radiation interacts almost exclusively with the aqueous medium [178]. The primary effect is water radiolysis:

$$H_2O + h\nu \longrightarrow H^\bullet + HO^\bullet \tag{5.1}$$

Figure 5.1: Photograph of Fricke gel samples irradiated up to 30 Gy (from left to right) at 5 Gy increments. Image adapted from reference [178]

where $^\bullet$ denotes a free radical. Subsequently, hydrogen atoms react with oxygen:

$$H^\bullet + O_2 \longrightarrow HO_2^\bullet \tag{5.2}$$

Following these initial interactions, several reactions can lead to the oxidation of ferrous ions; some of the main pathways are listed below:

$$\begin{aligned} Fe^{2+} + OH^\bullet &\longrightarrow Fe^{3+} + OH^- \\ Fe^{2+} + HO_2^\bullet &\longrightarrow Fe^{3+} + HO_2^- \\ HO_2^- + H_3O^+ &\longrightarrow H_2O_2 + H_2O \\ Fe^{2+} + H_2O_2 &\longrightarrow Fe^{3+} + OH^\bullet + OH^- \end{aligned} \tag{5.3}$$

As a result, the final concentration of Fe^{3+} is proportional to the absorbed dose:

$$\left[Fe^{3+}\right] = \frac{10\ D \cdot G(Fe^{3+}) \cdot \rho}{N_A\ e} \tag{5.4}$$

where D is the absorbed dose, $G(Fe^{3+})$ is the chemical yield of Fe^{3+} (expressed as ions produced per 100 eV), ρ is the density, N_A is Avogadro's number, and e is the energy conversion factor from electronvolts to joules [178].

5.1.3 Fricke Gel readout techniques

FG dosimeters can be read out using different acquisition techniques, including MRI, optical CT, and X-ray CT. Among these, MRI has historically been the most widely used method due to its high sensitivity to radiation-induced changes in relaxation rates [178, 179, 185], while optical CT represents a promising alternative offering faster acquisition and lower cost [186]. The choice of the readout technique depends on the required spatial resolution, accuracy, and experimental availability [186].

MRI technique

In principle, for any MRI sequence applied to FGs, the acquired signal correlates with the absorbed dose. In particular, FGs exhibit radiation-induced changes in both longitudinal (T_1) and transverse (T_2) relaxation times. Historically, Fricke gel dosimeters have been primarily read out using quantitative T_1 measurements; however, quantitative T_2 sequences can also be employed [185]. Here, only the dose–T_1 correlation is discussed, while further details can be found in [176, 178, 185].
As is well known, paramagnetic species affect the relaxation times of protons in water. The longitudinal relaxation rate $1/T_1$ can be expressed as [187]:

$$\frac{1}{T_1} = \frac{1}{T_{1,ND}} + R_{1,Fe^{2+}}, c_{Fe^{2+}} + R_{1,Fe^{3+}}, c_{Fe^{3+}} \tag{5.5}$$

where $T_{1,ND}$ is the longitudinal relaxation time in the absence of dopants, c denotes the concentration of the ionic species, and R_1 represents the relaxivity.
It follows that $R_{1,Fe^{3+}}$ is approximately 20 times larger than $R_{1,Fe^{2+}}$; therefore, $1/T_1$ is predominantly determined by the Fe^{3+} concentration and can be considered proportional to it. Figure 5.2 depicts the linearity $R1$-D.

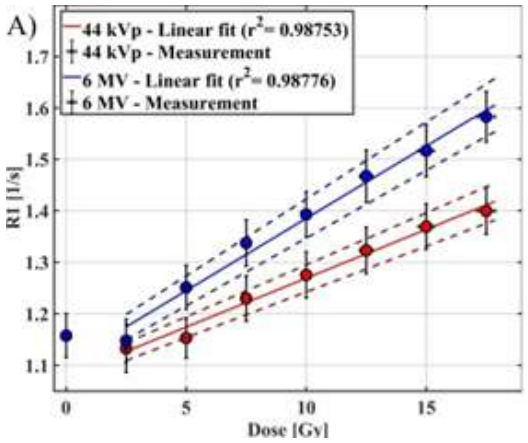


Figure 5.2: Relaxivity R_1 as function of absorbed dose in a FG dosimeter. Image adapted from reference [188]

Optical techniques

Ionizing radiation in FGs induces the oxidation of ferrous ions, and the resulting complex with a chelating agent—such as XO—produces a change in the absorption spectrum, which is known to be linear with the concentration of Fe^{3+} at the absorbance peak. This linearity can be explained by the Beer–Lambert law.
Considering an optical system in a transmittance measurement setup with fixed optical path length x, the transmitted light intensity is given by

$$I_T = I_0 e^{-\mu x} \tag{5.6}$$

where I_0 is the incident light intensity and μ is the linear attenuation coefficient. The absorbance A of the system is defined as the logarithm of the intensity ratio:

$$A = \log\left(\frac{I_0}{I_T}\right) = \mu x. \tag{5.7}$$

When multiple absorbing species are present in the medium, the attenuation coefficient can be expressed as the sum of the contributions of each species, i.e., as the product of their molar absorptivity and concentration:

$$A = x \sum_i \varepsilon_i \rho_i \tag{5.8}$$

where ε_i is the molar absorptivity and ρ_i is the concentration of the i-th species. When added to the Fricke solution, XO or other chelating agents bind Fe^{3+} ions, forming a chemical species that exhibits a strong absorption peak at a specific wavelength (see Figure 5.3). At this wavelength, the absorbance contributions from other species are negligible; therefore, $A \propto \rho_{Fe^{3+}}$. In particular, considering the variation in absorbance before and after irradiation, one obtains:

$$\Delta A = A_I - A_{NI} = x\varepsilon_{Fe^{3+}} \left(\rho_{Fe^{3+},I} - \rho_{Fe^{3+},NI}\right) = x\varepsilon_{Fe^{3+}} \Delta\rho_{Fe^{3+}} \tag{5.9}$$

where $\varepsilon_{Fe^{3+}}$ denotes the molar absorptivity of the chelating agent–Fe^{3+} complex, and the subscripts I and NI refer to irradiated and non-irradiated conditions, respectively.

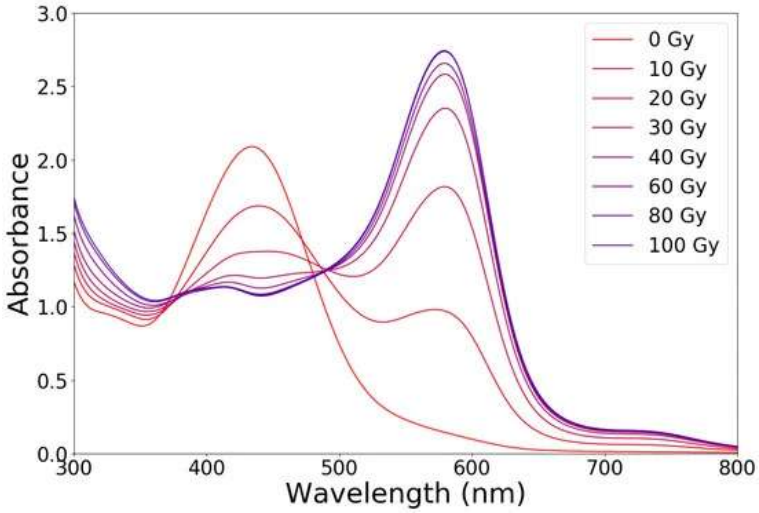


Figure 5.3: UV-VIS absorption spectra of a FG containing XO chelating agent and irradiated to absorbed doses up to 100 Gy. Image adapted from reference [178]

Since the absorbed dose is, in turn, proportional to the amount of ferric ions, it follows that:

$$D \propto \Delta\rho_{Fe^{3+}} \tag{5.10}$$

5.1.4 Diffusion in Fricke Gels

Diffusion in FGs is driven by concentration gradients of ferric ions. Thus, the physics of the diffusion process can be described by Fick's first law, which relates the ferric ion concentration $C(Fe^{3+}) \equiv C$ to the flux density of the diffusing species $\vec{\mathbf{J}}(Fe^{3+}) \equiv \vec{\mathbf{J}}$:

$$\vec{\mathbf{J}} = -\mathbf{D}\nabla C \tag{5.11}$$

where $\mathbf{D} \equiv ||D_{i,j}||$ is the diffusion tensor.
Combining Equation 5.11 with the continuity equation in the absence of sources and sinks

$$\frac{\partial}{\partial t}C = -\nabla \cdot \vec{\mathbf{J}}, \tag{5.12}$$

one obtains:

$$\frac{\partial}{\partial t}C = \nabla \cdot (\mathbf{D}\nabla C). \tag{5.13}$$

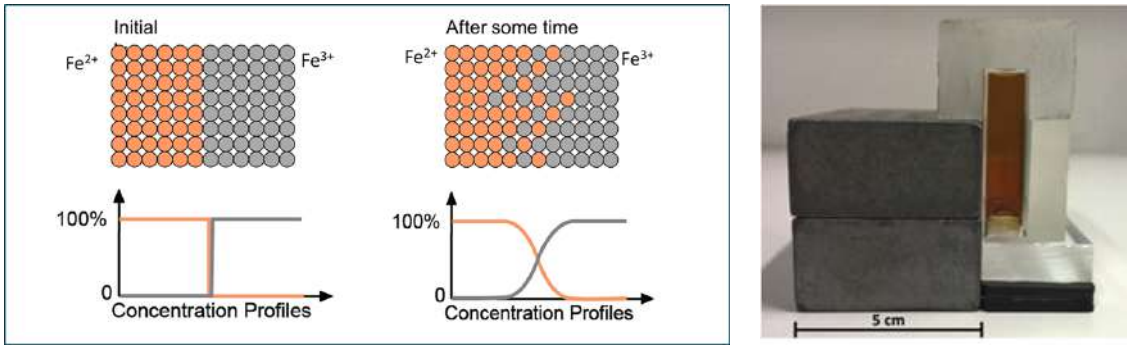
If diffusion is isotropic within the medium ($\{D_{i,j} = 0 \; for \; i \neq j, \; D_{i,j} = D \; for \; i = j\}$)—condition in line with most FGs applications—then $\mathbf{D}$ reduces to a scalar coefficient D, and the equation becomes:

$$\frac{\partial}{\partial t}C = D\nabla^2 C \tag{5.14}$$

where $\nabla^2 C = \left[\frac{\partial^2}{\partial x^2}C + \frac{\partial^2}{\partial y^2}C + \frac{\partial^2}{\partial z^2}C\right]$ is the Laplacian of the concentration field. Equation 5.14 is the *diffusion equation*, also known as the *heat equation*. Together with appropriate boundary conditions, it defines a Boundary Value Problem (BVP). Except for a few special cases, diffusion BVPs do not admit analytical solutions, and numerical methods are generally required.

The diffusion coefficient of ferric ions in FGs is strongly influenced by the gel microstructure, which in turn depends on the preparation method, including gelator type, crosslinking degree, fabrication protocol, and storage temperature [188–192]. It should be emphasized that, even under the same experimental conditions, two gels may exhibit noticeably different diffusion coefficients due to variations in the gelling process occurring within the cuvette.
Several experimental methods have been proposed to estimate ion diffusion coefficients from dose maps acquired using either MRI or optical techniques [187, 193–196]. Here, only the method proposed by Kron *et al.* is presented.
This approach is based on a gel cuvette partially irradiated to produce a sharp dose gradient (Figure 5.4 a), with a near-zero dose region on one side and a uniform dose region on the other. Such an experimental configuration can be achieved by irradiating the cuvette with a uniform field while shielding half of it with a lead block (Figure 5.4 b).
Assuming translational invariance along one direction, the problem can be reduced to a one-dimensional dose profile exhibiting a quasi-step function, as depicted by linear

concentration profile in Figure 5.4 a.



(a) (b)

Figure 5.4: (a) Schematic representation of diffusion-driven smoothing of an initial concentration step profile. Figure adapted from Fig. 5.1 and Fig 5.2 in reference [197]. (b) Irradiation setup for diffusion coefficient measurements as proposed by Kron *et al.*. Figure adapted from reference [198].

The acquired concentration profile can be fitted with an inverse square root function of the form:

$$f(x,t) = f_{min} + \frac{1}{2}\left(f_{max} - f_{min}\right)\left[1 + \frac{x - x_0}{\sqrt{(x-x_0)^2 + n(t)}}\right] \tag{5.15}$$

where $f_{\min}$ and $f_{\max}$ are the minimum and maximum measured concentration values, respectively, x_0 is the lateral shift of the inflection point, and $n(t)$ is the so-called "curvature parameter". This parameter varies linearly with time, and its temporal evolution relates with the diffusion coefficient. In particular, the diffusion coefficient of the gel can be expressed as the rate of change of $n(t)$ multiplied by a constant factor. The full derivation of such a factor is reported in [187], where:

$$D = \frac{\sqrt[3]{4} - 1}{4 \ln 2} \frac{n(t)}{t} \approx 0.212 \frac{n(t)}{t}. \tag{5.16}$$

PVA Type	PVA Concentration (% w/v)	Molecular Weight ($\times 10^3$)	GTA Concentration (mM)	Diffusion Coefficient (mm^2h^-1)
PVA-A	10%	$85 - 128$	26.5	0.171 ± 0.012
PVA-B	10%	$85 - 128$	6.625	0.249 ± 0.029
PVA-C	10%	$31 - 50$	26.5	0.200 ± 0.018
PVA-D	10%	$13 - 23$	26.5	0.291 ± 0.033
Mw-A	10%	125	26.5	0.184 ± 0.014
Mw-B	12.5%	125	33.125	0.152 ± 0.008

Table 5.1: Typical diffusion coefficients for different PVA-GTA FGs. Table adapted from reference [199].

Knowledge of the diffusion coefficient allows for the estimation of the usable time window of FG dosimeters. However, it should be emphasized that steep dose gradients contribute to a faster degradation of the recorded dose distribution [200].

5.1.5 Auto-oxidation in Fricke Gels

Auto-oxidation (AO) in FGs is the process that induces the oxidation of ferrous ions even in the absence of irradiation [178]. This phenomenon occurs because, within the chemical system, the ferric state is thermodynamically more stable than the ferrous one. The literature indicates that AO depends on several factors, such as Fe^{2+} concentration, the presence of chelating agents, oxygen content in the cuvette, and temperature [186, 201]. Although AO has been studied and quantified for different gel formulations [186, 201, 202]—with reported rates in the range 10^{-4}–10^{-3} ODh^{-1}—it has generally been treated as a global property of the system. To the best of the author's knowledge, no studies have investigated the local behavior of AO, representing a limitation for analyses such as those proposed in this chapter.
Given the complexity and the multiple factors involved in AO processes, the assumption of a spatially uniform effect within the FG may represent a practical approximation, although it requires validation. In Section 5.6.5, this hypothesis is adopted and subsequently validated.

5.2 Numerical Simulation of Fricke Gel System Evolution

The temporal evolution of the concentration, or equivalently the dose, distribution was used to generate benchmark data for the development of PINNs. In particular, starting from a modeled initial distribution at the irradiation time t_I, numerical solutions were computed to obtain the distribution at the measurement time t_M (*Forward in time simulation*). This was achieved by integrating numerically Equation 5.14, after defining appropriate boundary conditions to reproduce realistic diffusion dynamics. The main goal was to obtain pairs of distributions at different times, related through diffusion. These data pairs were subsequently used for PINN training—where the diffused distribution served as the input—and validation—where the initial distribution was used as ground truth (GT). Numerical simulations were also employed for parameter validation, as described in Section 5.6.5.

5.2.1 Boundary Value Problem definition

As introduced in Section 5.1.4, the diffusion equation governs ion dynamics within FGs. Equation 5.14 describes the simplest diffusion process, in which no interactions between ions are considered and no sources or sinks are present. This assumption holds as long as AO effects are negligible; otherwise, additional terms must be included in the equation.
The diffusion model presented here is valid for most FG applications. However, in

Section 5.6, where AO is also investigated, a modified model is required, as discussed therein.

A BVP is defined by: (i) a partial differential equation (PDE)—in this case, the heat equation—(ii) an initial condition describing the system state before temporal evolution takes place, and (iii) boundary conditions describing the behavior at the domain boundaries.

The initial conditions used in the numerical simulations were modeled to reproduce a standard irradiation setup. In particular, although the simulations consider different scenarios—especially in terms of dimensionality, as discussed below—they all share a common feature: partial irradiation with uniform patterns. This model is representative of many RT applications, although it remains a simplified approximation.

Von Neumann boundary conditions were applied. These conditions impose a zero ion flux at the boundaries, which is consistent with the physical behavior of FG systems.

5.2.2 Forward in Time Centered in Space numerical scheme

Simulations were performed using the Forward in Time Centered in Space (FTCS) numerical scheme. A brief derivation is provided here, while more details can be found in [203].

The first step in the numerical simulation is modeling the physical system—here, a FG dosimeter—as a discrete grid of points. For a 1D system of length L, the grid is defined as an array of N_x points with spatial spacing $\Delta x = \frac{L}{N_x}$. For a two-dimensional system of size $L_x \times L_y$, the grid is represented by a 2D array of size $N_x \times N_y$, where each element (pixel) has an area $\Delta x \times \Delta y$, with $\Delta x = \frac{L_x}{N_x}$ and $\Delta y = \frac{L_y}{N_y}$. The same approach applies to the 3D case.

Time must also be discretized. The time step is defined as $\Delta t = \frac{T}{N_t}$, where T is the total simulation time (i.e., $t_M - t_I$ in FG simulations) and N_t is the number of time steps.

Once space and time have been discretized, the approximate numerical solution can be expressed as $\{u^t_{i,j} \mid i = 0, \ldots, N_x,\ j = 0, \ldots, N_y,\ t = 0, \ldots, N_t\}$ in the 2D case.

Making use of the *finite difference approximation* (first order Taylor approximation), the left-hand side of Equation 5.14 can be rewritten as:

$$\frac{\partial}{\partial t} u(x, y, t) \approx \frac{u(x, y, t + \Delta t) - u(x, y, t)}{\Delta t} \tag{5.17}$$

For the spatial part, a *Taylor series expansion* can be used to derive:

$$\begin{aligned} u(x + \Delta x, y, t) &= u(x, y, t) + \frac{\partial u}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}\Delta x^2 + \cdots \\ u(x - \Delta x, y, t) &= u(x, y, t) - \frac{\partial u}{\partial x}\Delta x + \frac{1}{2}\frac{\partial^2 u}{\partial x^2}\Delta x^2 + \cdots \end{aligned} \tag{5.18}$$

By summing these expressions, one obtains:

$$\frac{\partial^2 u}{\partial x^2} \approx \frac{u(x+\Delta x, y, t) - 2u(x, y, t) + u(x-\Delta x, y, t)}{\Delta x^2}. \tag{5.19}$$

Analogously,

$$\frac{\partial^2 u}{\partial y^2} \approx \frac{u(x, y+\Delta y, t) - 2u(x, y, t) + u(x, y-\Delta y, t)}{\Delta y^2}. \tag{5.20}$$

Combining Equations 5.17, 5.18, and 5.20 with Equation 5.14, the integration scheme is obtained:

$$u_{i,j}^{t+1} = u_{i,j}^t + D\Delta t \left[\frac{u_{i+1,j}^t - 2u_{i,j}^t + u_{i-1,j}^t}{\Delta x^2} + \frac{u_{i,j+1}^t - 2u_{i,j}^t + u_{i,j-1}^t}{\Delta y^2} \right]. \tag{5.21}$$

The 3D extension or the 1D reduction can be derived analogously. The FTCS scheme computes the solution at the next time step from the previous one. However, it must satisfy a stability condition, which can be derived via Von Neumann stability analysis. For the FTCS scheme, this condition reads [204]:

$$\Delta t \leq \frac{1}{2D\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)} \tag{5.22}$$

A common choice is to set $\Delta x = \Delta y$. In this case:

$$\begin{cases} u_{i,j}^{t+1} = u_{i,j}^t + \frac{D\Delta t}{\Delta x^2}\left[u_{i+1,j}^t + u_{i-1,j}^t + u_{i,j+1}^t + u_{i,j-1}^t - 4u_{i,j}^t\right] \\ \Delta t \leq \frac{\Delta x^2}{4D}. \end{cases} \tag{5.23}$$

Neumann boundary conditions impose:

$$\hat{\mathbf{n}} \cdot \nabla u = 0 \quad \text{at the boundary} \tag{5.24}$$

with $\hat{\mathbf{n}}$ indicating the normal unit vector to the domain boundary. In a rectangular grid setup, this condition translates into:

$$\begin{cases} u_{0,j}^t = u_{1,j}^t, \quad u_{N_x,j}^t = u_{N_x-1,j}^t, \quad \forall j, t \\ u_{i,0}^t = u_{i,1}^t, \quad u_{i,N_y}^t = u_{i,N_y-1}^t, \quad \forall i, t \end{cases} \tag{5.25}$$

5.2.3 1D Numerical Simulations

This section presents key aspects of the 1D numerical simulations conducted for the feasibility study in PINN development. A complete description can be found in [180].

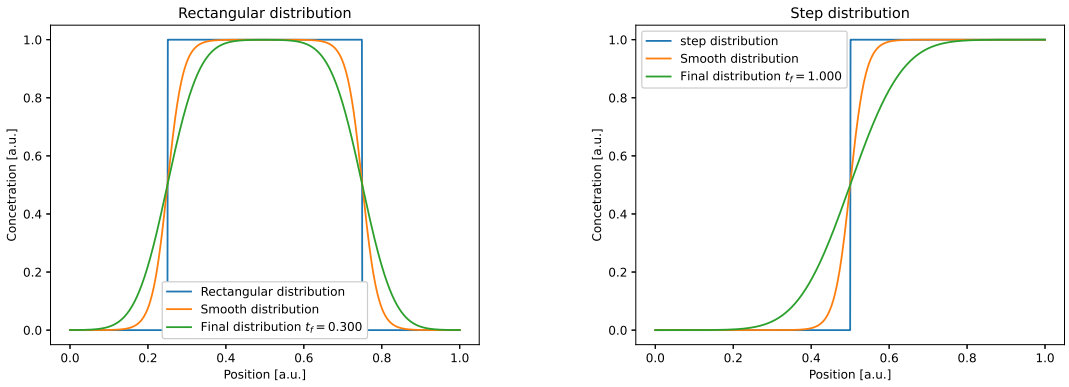
The simulations were performed by modeling a FG system as a 1D rod undergoing partial irradiation, under the assumption of a uniform beam. Two types of initial

distributions were considered, with rectangular and step shapes, respectively (Figure 5.5). Starting from these distributions, their temporal evolution was computed using the 1D version of the scheme described above:

$$\begin{cases} u_i^{t+1} = u_i^t + \frac{D\Delta t}{\Delta x^2}\left[u_{i+1}^t + u_{i-1}^t - 2u_i^t\right], & \forall i \neq 0, N_x \\ u_0^t = u_1^t, \quad u_{N_x}^t = u_{N_x-1}^t, & \forall t \end{cases} \tag{5.26}$$

Simulations were conducted in arbitrary units and subsequently converted to physical units. The parameters were set as $D = 5 \times 10^{-3}$ a.u., $\Delta x = 1 \times 10^{-3}$ a.u., and rod length $L = 1$ a.u.. To ensure stability, the integration time step was set to one-fifth of the maximum value allowed by the stability condition (Equation 5.22), resulting in $\Delta t = 2 \times 10^{-5}$ a.u..

The rectangular distribution was integrated for 15,000 time steps, while the step distribution for 50,000 steps, corresponding to simulation times of 0.3 a.u. and 1.0 a.u., respectively. Assuming a rod length of 20 mm and a diffusion coefficient of 0.200 mm^2h^{-1}, these correspond to physical times of 3 h and 10 h. The results are shown in Figure 5.5.



(a) (b)

Figure 5.5: 1D rod model partially irradiated with a uniform beam: (a) rectangular distribution, (b) step distribution. The blue curves exhibit discontinuities with vertical slopes, where $\frac{\partial C}{\partial x} \to +\infty$. Since such distributions are not physically realistic, sigmoid functions were used to generate smoother profiles, as shown by the orange curves. These distributions represent the modeled initial distributions. The green curves represent the final distributions after 15,000 integration steps ($\Delta t = 2 \times 10^{-5}$ a.u., $t_f = 3 \times 10^{-1}$ a.u.) for (a), and 50,000 integration steps ($\Delta t = 2 \times 10^{-5}$ a.u., $t_f = 1$ a.u.) for (b). Additional numerical parameters: number of points 1001, rod length $L = 1$ a.u., $\Delta x = 1 \times 10^{-3}$ a.u., diffusion coefficient $D = 0.005$ a.u.. Figure adapted from [180].

5.2.4 2D Numerical Simulations

The PINN method was first extended to 2D distributions using data generated through numerical simulations. In particular, PVA-GTA FG matrices of size 100×100 mm^2 were modeled, assuming a diffusion coefficient $D = 0.25$ $\text{mm}^2\,\text{h}^{-1}$, consistent with the PVA-GTA-B gel reported in Table 5.1. The adopted numerical scheme is exactly that presented in Section 5.2.2.

The initial conditions were defined assuming partial irradiation with: (a) a uniform circular beam of radius 25 mm, and (b) a uniform rectangular beam of area 40×60 mm^2. This setup mimics experimental conditions in which the axial dimension is significantly smaller than the lateral ones; therefore, the depth dose profile (DDP) can be considered approximately constant.

The initial conditions were constructed by assigning a value of 1 a.u. to the grid points within the irradiated region—centered with respect to the domain—and zero elsewhere. Since such distributions are not physically realistic, smoother profiles were obtained by allowing the system to evolve for a few time steps.

For numerical convenience, the simulations were performed in arbitrary units by normalizing the spatial domain to $[0, 1] \times [0, 1]$, yielding the correspondence 1 a.u. = 100 mm. The diffusion coefficient was set to $D = 0.005$ a.u.. Given the physical size of the gel matrix and its diffusion coefficient, the relation between time in arbitrary and physical units is 1 a.u. = 200 h.

The computational domain (1×1 a.u.) was discretized into 10,000 points, resulting in $\Delta x = \Delta y = 10^{-2}$ a.u.. As in the 1D case, the integration time step was set to one-fifth of the maximum value allowed by the stability condition, yielding $\Delta t = 10^{-3}$ a.u..

In studying the 2D extension of the PINN method [181], the reconstruction capability was also analyzed as a function of the time interval between irradiation and measurement. This required considering multiple pairs of initial and diffused distributions for each irradiation setup. Such data were generated by simulating temporal evolution over 20, 40, 60, 80, and 100 h. Figures 5.6 and 5.7 show results.

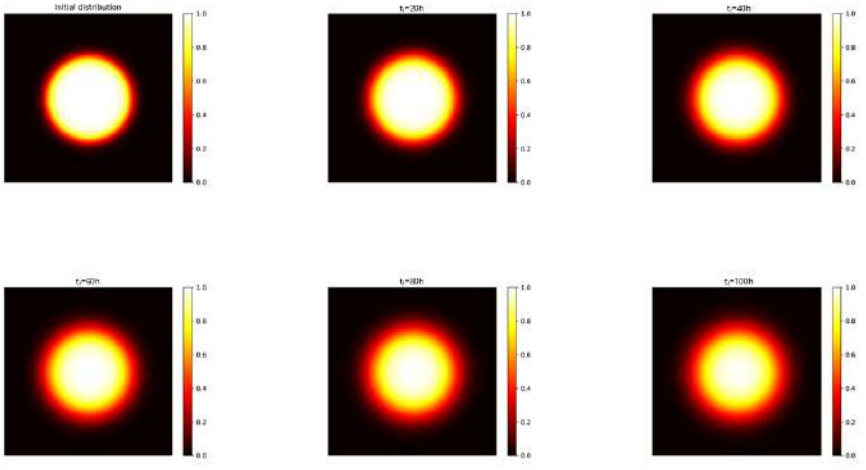


Figure 5.6: 2D distribution simulations for a circular beam irradiation setup. Diffusion induces progressive blurring, leading to a reduction in dose gradients over time. Figure adapted from reference [181]

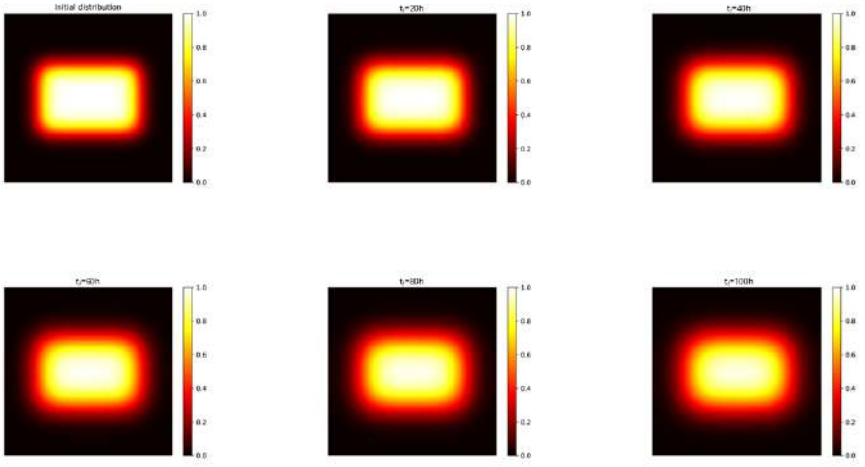


Figure 5.7: 2D distribution simulations for a rectangular beam irradiation setup. Diffusion induces progressive blurring, leading to a reduction in dose gradients over time. Additionally, the initial rectangular shape gradually deforms, with the final distribution approaching an elliptical-like shape. Figure adapted from reference [181]

5.2.5 3D Numerical Simulation

In the 3D case, in a manner similar to the 2D case, only a simulation was conducted. A PVA-GTA gel system of size $100 \times 100 \times 50$ mm^3 was considered, with a diffusion coefficient $D = 0.25$ mm^{2}h^{-1}. A uniform square beam spot of size 50×50 mm^2 was assumed as the irradiation setup.

A key difference with respect to the 2D case arises from the fact that ionizing radiation interacts non-uniformly along the depth direction. In general, the deposited dose follows a DDP, which depends on the radiation type and the traversed medium. Therefore, to reproduce a realistic scenario, the 3D initial distribution was constructed by stacking 50 2D uniform square distributions, with values assigned according to a normalized depth dose profile corresponding to 7 MeV electron irradiation measured in water equivalent medium. This profile was obtained for an ultra-high dose rate electron beam generated by the FLASH linear accelerator currently installed at the Centro Pisano per la Radioterapia FLASH (CPFR) in Pisa (Italy) [205], and is shown in Figure 5.8 (red curve) [181].

As in the 2D case, the generated 3D distribution was further evolved for a few time steps to introduce smoothness. The resulting DDP of the modeled initial distribution is shown by the blue dots in Figure 5.8.

The numerical parameters used in the simulation are as follows: spatial domain $[0, 1] \times [0, 1] \times [0, 0.5]$, with the latter referring to the z direction; $\Delta x = \Delta y = \Delta z = 10^{-2}$ a.u., ensuring the same spatial correspondence between arbitrary and physical units encountered in the 2D case; $D = 0.005$ a.u.; $\Delta t = 6.7 \times 10^{-4}$ a.u.; and 300 integration steps, corresponding to a total simulation time of 0.2 a.u. $\equiv$ 40 h. The

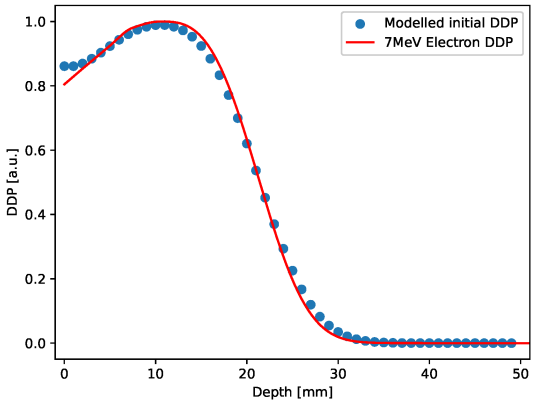


Figure 5.8: Red curve: Depth dose profile for a 7 MeV electron irradiation; blue circles: Depth dose profile extracted from the 3D initial distribution, after smoothing. Figure adapted from reference [181].

results are shown in Figure 5.9.

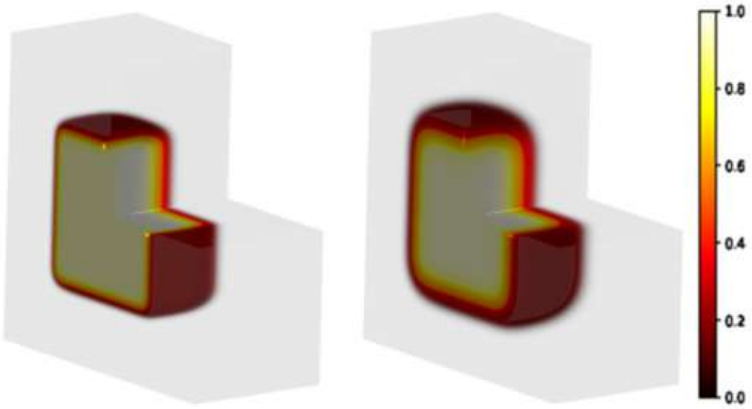


Figure 5.9: Numerical simulation of a 3D FG system. The initial distribution is shown on the left, while the distribution after 40 h of diffusion is shown on the right. Image adapted from reference [181].

5.3 General PINN model algorithm for Fricke Gel

Physics-Informed Neural Networks were introduced in Section 2.4, with particular focus on their application to solving BVPs in Section 2.4.1. In this section, it is shown how this approach can be used to mitigate the blurring effect in dose distributions caused by ion diffusion.

First, the backward-in-time diffusion BVP is defined, providing the mathematical framework underlying the methodology. Subsequently, the general algorithmic model is presented. This approach is applied to all studies discussed in this chapter, except for the final application, where specific modifications were required and are discussed later. Finally, since the backward-in-time diffusion equation represents an *ill-posed* problem, its theoretical implications and practical limitations are addressed.

5.3.1 Backward in time diffusion BVP

The backward-in-time diffusion equation defines a BVP that is formally equivalent to that presented in Section 5.2.1, with the key difference that the temporal ordering is reversed. In this case, the problem is not initialized from an initial condition but from a final condition—i.e., a distribution that has already undergone diffusion. While this modification does not significantly affect other types of BVPs—such as those describing wave propagation—it represents a substantial change for diffusion processes. Indeed, the diffusion equation (Equation 5.14) is not invariant under time reversal, resulting in an ill-posed problem.

Ill-posedness refers to mathematical problems in which the *existence and uniqueness of the solution* are not guaranteed. For readers less familiar with advanced mathematical concepts, these definitions may seem abstract. However, their physical implications can be interpreted more intuitively.

First, in thermodynamic terms, diffusion is an *irreversible* process. This implies that the mathematical equation describing it must not admit time reversal. Secondly, during its temporal evolution, the system passes through a sequence of *mesostates* until equilibrium is reached. In statistical mechanics, thermodynamic equilibrium is defined as the most probable macrostate, i.e., the state corresponding to the largest number of accessible *microstates*. In the case of diffusion, this condition corresponds to a uniform distribution (i.e., zero concentration gradients). As an illustrative example, if one attempts to reverse diffusion over a time interval t starting from a uniform distribution, infinitely many solutions can satisfy the corresponding BVP. More generally, in intermediate conditions, this implies that starting from a mesostate (i.e., a distribution at the measurement time) and evolving backward in time, multiple mesostates—equally consistent with the observed state—may exist as valid solutions of the BVP.

These issues also affect numerical simulations, leading to unstable integration schemes. As a demonstration, it can be readily observed that time reversal applied to Equation 5.14 yields:

$$\frac{\partial C}{\partial t} = -D\nabla^2 C \tag{5.27}$$

i.e., a diffusion equation with a negative diffusion coefficient. When this result is combined with the stability condition (Equation 5.22), it becomes evident that the condition can never be satisfied.

All these issues motivated a change in approach, leading to the use of PINNs for the backward-in-time diffusion equation. In fact, the method presented in Section 2.4.1 aims to solve the standard diffusion BVP locally, i.e. pointwise, and then reconstruct the global solution by combining these local evaluations. Time inversion is achieved by providing the model with the final distribution and with temporal coordinates restricted to times preceding the measurement time. In this way, the model identifies an approximation of one of the admissible solutions to the backward problem, implicitly

regularizing the ill-posed problem through the NN parametrization and the PDE constraints.

5.3.2 PINN algorithm for distribution de-blurring

In this section, the algorithm used to recover the initial dose distribution in an FG from a blurred measurement is presented. To maintain generality, the method is introduced for a 3D system; the 1D and 2D cases can be obtained by reducing the number of spatial variables.

Following the approach introduced in Section 2.4.1, the starting point is the modeling of the gel system as a mathematical domain. Consider a three-dimensional FG system of size L_x, L_y, and L_z. The corresponding domain is defined as $\Omega = [0, L_x] \times [0, L_y] \times [0, L_z]$, and its boundary is denoted by $\partial\Omega$. Let $\mathbf{x} \in \Omega$ represent a generic spatial point and $t \in [0, t_M]$ the time variable, where t_M is the measurement time. The ion concentration can then be expressed as $C(\mathbf{x}, t)$.
Let $NN(\mathbf{x}, t; \boldsymbol{\theta})$ be a neural network with an input layer of size 4 (three spatial coordinates and time), parametrized by $\boldsymbol{\theta}$.
Given a set of N_1 collocation points $(\mathbf{x}_i, t_i) \in \Omega \setminus \partial\Omega \times [0, t_M)$, randomly sampled, the residual loss $\mathcal{L}_{res}$ is defined as:

$$\mathcal{L}_{res} = \frac{1}{N_1} \sum_{i=1}^{N_1} \left(\frac{\partial}{\partial t} NN(\mathbf{x}_i, t_i; \boldsymbol{\theta}) - D\nabla^2 NN(\mathbf{x}_i, t_i; \boldsymbol{\theta}) \right)^2 \tag{5.28}$$

where D is the diffusion coefficient.

Neumann boundary conditions (Equation 5.24) are used to define the boundary loss function $\mathcal{L}_{BC}$. In a 3D rectangular domain, there are six boundary surfaces, included in the planes $P_1 : x = 0$, $P_2 : x = L_x$, $P_3 : y = 0$, $P_4 : y = L_y$, $P_5 : z = 0$, and $P_6 : z = L_z$. Thus, Equation 5.24 yields six conditions.

Given a set of N_2 boundary points $(\mathbf{x}_j, t_j) \in \partial\Omega \times [0, t_M)$, sampled uniformly across all boundary surfaces, the boundary loss is defined as:

$$\mathcal{L}_{BC} = \frac{1}{N_2} \sum_{i=1}^{6} \sum_{j \in P_i} \left(S_i(\mathbf{x}_j, t_j) \right)^2 \tag{5.29}$$

where the terms S_i are defined as:

$$S_i : \begin{cases} S_1 = \frac{\partial}{\partial x} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_1 \cap \Omega \\ S_2 = \frac{\partial}{\partial x} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_2 \cap \Omega \\ S_3 = \frac{\partial}{\partial y} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_3 \cap \Omega \\ S_4 = \frac{\partial}{\partial y} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_4 \cap \Omega \\ S_5 = \frac{\partial}{\partial z} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_5 \cap \Omega \\ S_6 = \frac{\partial}{\partial z} NN(\mathbf{x}, t; \boldsymbol{\theta}) & \mathbf{x} \in P_6 \cap \Omega \end{cases} . \tag{5.30}$$

Finally, given the distribution at the measurement time $C(\mathbf{x}, t_M)$ and a set of N_3 points $(\mathbf{x}_k, t_M) \in \Omega \times \{t_M\}$, the data-driven loss $\mathcal{L}_{data}$ is defined as:

$$\mathcal{L}_{data} = \frac{1}{N_3} \sum_{k=1}^{N_3} \left(NN(\mathbf{x}_k, t_M; \boldsymbol{\theta}) - C(\mathbf{x}_k, t_M) \right)^2 . \tag{5.31}$$

Since the computation of $\mathcal{L}_{data}$ involves randomly sampled points, it is necessary to use an interpolating function for $C(\mathbf{x}, t_M)$, which is typically defined only on a fixed grid of points.
The total loss is defined as $\mathcal{L} = \mathcal{L}_{res} + \mathcal{L}_{BC} + \mathcal{L}_{data}$.

Algorithm 3 summarizes the steps required to optimize the PINN parameters for FG distribution deblurring, while Figure 5.10 provides a scheme of the training process.

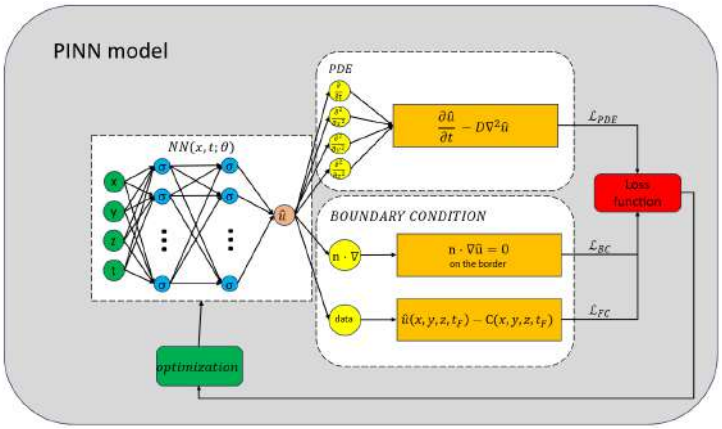


Figure 5.10: Scheme of the training process of the PINN model. The NN's output is evaluated by the loss functions. The total loss function serves as an indicator of the model's efficacy and guides the optimization algorithm in adjusting the parameters. This iterative process continues until the NN's error is sufficiently minimized. Figure adapted from reference [181].

Algorithm 3: PINN training procedure

Data: Neural network $NN(\cdot;\boldsymbol{\theta})$; diffusion coefficient D; distribution at t_M; domain definition; number of epochs e; loss functions $\mathcal{L}$, $\mathcal{L}_{res}$, $\mathcal{L}_{BC}$, $\mathcal{L}_{data}$; N_1, N_2, N_3; optimization algorithm; monitoring function; checkpoint function.

Result: Approximate solution of the backward-in-time diffusion BVP

Initialize monitoring function to log results at each epoch
Initialize geometry handler from domain definition
Initialize $L_{\text{best}} = \infty$
for $i = 1$ **to** e **do**
    $\mathbf{A} \leftarrow$ sample N_1 points from $(\Omega \setminus \partial\Omega) \times [0, t_M)$ using the geometry handler
    $\mathbf{B} \leftarrow$ sample N_2 points from $\partial\Omega \times [0, t_M)$ using the geometry handler
    $\mathbf{C} \leftarrow$ sample N_3 points from $\Omega \times \{t_M\}$ using the geometry handler
    $\mathbf{U}_1 \leftarrow NN(\mathbf{A};\boldsymbol{\theta})$
    $L_1 \leftarrow \mathcal{L}_{res}(\mathbf{U}_1)$
    $\mathbf{U}_2 \leftarrow NN(\mathbf{B};\boldsymbol{\theta})$
    $L_2 \leftarrow \mathcal{L}_{BC}(\mathbf{U}_2)$
    $\mathbf{U}_3 \leftarrow NN(\mathbf{C};\boldsymbol{\theta})$
    $L_3 \leftarrow \mathcal{L}_{data}(\mathbf{U}_3)$
    $L \leftarrow L_1 + L_2 + L_3$
    $\boldsymbol{\theta} \leftarrow$ update parameters using the optimization algorithm
    Log loss value
    if $L < L_{best}$ **then**
        Save model checkpoint
        $L_{\text{best}} \leftarrow L$
    end
end

5.3.3 Network architecture and training parameters

The network architecture designed for this study is a simple Multi-Layer Perceptron (MLP). This choice arises from the need for a universal approximator with sufficient expressive power while keeping the number of parameters relatively low. These requirements are satisfied by adopting five fully connected hidden layers with units arranged as $[10, 20, 200, 20, 10]$. The size of the input layer depends on the specific problem: for a 1D distribution it is 2 (one spatial dimension and time), for a 2D distribution it is 3, and for a 3D distribution it is 4. The output layer consists of a single unit. In this way,

the NN can be interpreted as a function:

$$NN : \Omega \times [0, t_M] \subseteq \mathbb{R}^{N+1} \longrightarrow \mathbb{R}$$
$$(\mathbf{x}, t) \mapsto NN(\mathbf{x}, t)$$

where N is the spatial dimensionality and Ω denotes the spatial domain. The total number of parameters of the network is 8691 in the 3D case (and slightly lower in lower-dimensional settings).
Since computing derivatives with respect to the inputs requires the network to be fully differentiable, it is necessary to adopt an activation function that is differentiable over the entire $\mathbb{R}$. This requirement is satisfied by the hyperbolic tangent activation function (tanh).
The model is trained using the *Adam* optimizer [206], which ensures fast convergence. As a regularization strategy, an initial maximum number of epochs and a threshold on the loss function are defined; training is then continued adaptively if the loss has not reached the desired threshold. In addition, model checkpointing is employed to save the network parameters whenever a lower loss value is achieved.
In the 1D case, models are trained for 30,000 epochs by sampling $N_1 = 20{,}000$, $N_2 = 10{,}000$, and $N_3 = 20{,}000$ random points at each epoch, ensuring adequate coverage of the space–time domain. In the 2D case, models are trained for 100,000 epochs with $N_1 = 20{,}000$, $N_2 = 20{,}000$, and $N_3 = 1{,}000$ sampled points per epoch. In the 3D case, models are trained for 200,000 epochs with $N_1 = 100{,}000$, $N_2 = 80{,}000$, and $N_3 = 80{,}000$ sampled points per epoch. These hyperparameters were determined through a trial-and-error procedure.
The training time is approximately 30 minutes for the 1D case, 1.5 hours for the 2D case, and 9 hours for the 3D case, using a workstation equipped with an `NVIDIA RTX A5000` GPU with 24 GB of memory.
The entire pipeline was implemented in Python, using Jupyter Notebook [168] and the TensorFlow library [41].
For the 2D and 3D cases, since the algorithm provides a list of numbers as results—one for each input point—it is necessary to use a reconstruction algorithm able to associate each point with its corresponding position in the grid. This was done by using NumPy's reshaping functions [90].

5.3.4 Solution stability and reproducibility

As already mentioned, the backward-in-time diffusion BVP is ill-posed, meaning in particular that solution uniqueness is not guaranteed. Although the methodology presented here mitigates the stability issues affecting traditional numerical schemes, the intrinsic non-uniqueness of the solution remains a limitation.

In general, diffusion can be interpreted as a progressive loss of information over

time: the longer the time interval between irradiation and measurement, the more information about the initial distribution is lost. The proposed algorithm can provide meaningful corrections only as long as part of the initial information is preserved. For instance, a circular distribution can be effectively recovered as long as its symmetry is retained; otherwise, reconstruction becomes impossible.

A significant loss of initial information leads to poor reproducibility, meaning that multiple runs on the same data may yield noticeably different results.
When a BVP is well-posed, the loss hypersurface in parameter space exhibits a unique global minimum corresponding to the unique solution. In deep learning, however, the optimization process typically converges to a local minimum. Training the neural network with different parameter initializations provides a practical way to explore the loss landscape in search of better approximations. In contrast, for an ill-posed problem, the loss hypersurface may contain multiple global minima, corresponding to different admissible solutions, while the optimization process still converges to a local minimum. Starting from different initializations—and given the high dimensionality of the parameter space—it is highly unlikely that the optimization process will consistently converge to the same solution.

PINNs can be trained multiple times by varying the random seed used for parameter initialization. This allows one to assess the stability and reproducibility of the results, while also providing insight into the intrinsic uncertainty associated with the ill-posed backward-in-time diffusion problem.
Using this approach, results can be reported as ensemble averages, with performance metrics expressed as mean values and standard deviations used as uncertainty quantifiers.

This procedure was partially applied to the 2D and 3D extensions—where uncertainties are not reported, as they were found to be negligible—and to the study of FGs affected by auto-oxidation. In the latter case, experimental fluctuations resulted in solutions more sensitive to parameter initialization.

5.3.5 Characteristic time scale

After the first attempts yielded unexpectedly good results, a natural question arose:

How far back in time can we reliably reconstruct the initial distribution?

A first idea was to evaluate the reconstruction performance as a function of the diffusion time. However, this approach was soon discarded, since the blurring effect also depends on the initial distribution [200]. Moreover, in diffusion processes, space and time are not independent but are linked through the diffusion coefficient, leading to a scaling invariance of the governing equation. In particular, under the transformation $x \to \alpha x$, $t \to \alpha^2 t$, the diffusion dynamics remain unchanged. Therefore, defining a universal temporal range that ensures model performance, without accounting for spatial scales or the initial distribution, is not meaningful.

A more suitable approach is obtained by introducing the root-mean-square displacement in diffusion processes.
Let us consider an initial distribution, either simulated or measured, generated by an irradiation beam characterized by a spatial parameter. For example, in the case of a circular beam this parameter may be the spot radius, while for a square beam it can be the side length. It is then reasonable to assume that the initial distribution reconstruction becomes unreliable when the characteristic diffusion length becomes comparable to this spatial scale.

In diffusion processes, the root-mean-square displacement is given by:

$$\tau(t) = \sqrt{2NDt} \tag{5.32}$$

where N is the spatial dimensionality (1, 2, or 3), D is the diffusion coefficient, and t is time. Based on this, a theoretical upper bound for the backward reconstruction time can be defined as:

$$t_{\max} = \frac{l_c^2}{2ND} \tag{5.33}$$

where l_c is a characteristic length of the initial distribution (or equivalently of the irradiation setup).
With this definition, results can be conveniently reported in terms of fractions of $t_{\max}$.

5.4 Evaluation methods and metrics

In order to evaluate the performance of the PINN approach, it is necessary to compare the predicted distribution with the GT, i.e., the true initial distribution. A detailed description is provided in [180, 181]; here, a brief summary is given for completeness. The main metric adopted is MSE, defined as:

$$MSE = \frac{1}{N} \sum_{\mathbf{x}_i} \left(C(\mathbf{x}_i, t_0) - NN(\mathbf{x}_i, t_0) \right)^2 \tag{5.34}$$

where $C(\mathbf{x}_i, t_0)$ and $NN(\mathbf{x}_i, t_0)$ denote the true spatial concentration and the neural network prediction at the initial time, respectively, and the sum is taken over all N points of the discrete domain.
Additionally, to account for discrepancies in both concentration (or dose) and spatial localization, a gamma index (γ) analysis [207, 208] was performed.
The gamma index is one of the most widely used metrics in RT, as it combines dose difference and distance-to-agreement criteria into a single measure. For each point in the dose distribution, the gamma index γ is defined as:

$$\gamma(\mathbf{x}) = \min_{\mathbf{x}' \in \Omega} \sqrt{\left(\frac{C(\mathbf{x}) - C'(\mathbf{x}')}{\delta C} \right)^2 + \left(\frac{\mathbf{x} - \mathbf{x}'}{\delta x} \right)^2} \quad \forall \mathbf{x} \in \Omega \tag{5.35}$$

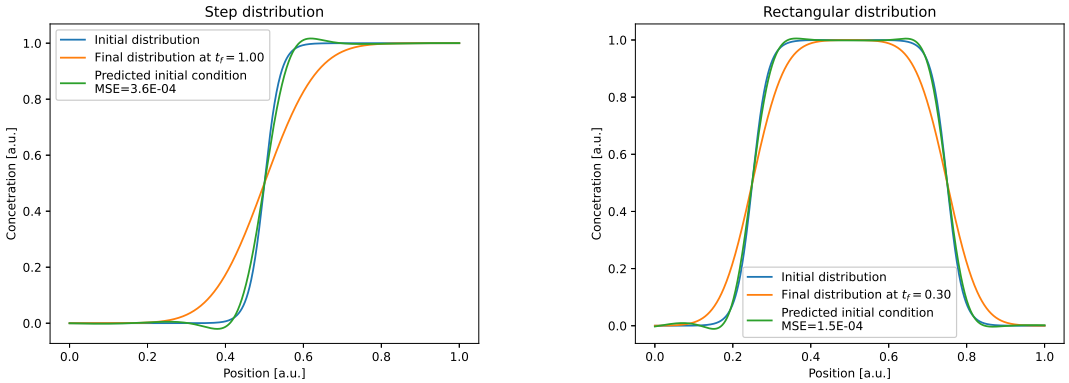
where $C(\mathbf{x})$ is the reference (true) dose distribution at position $\mathbf{x}$, $C'(\mathbf{x}')$ is the predicted distribution at position $\mathbf{x}'$, δC is the dose difference criterion (typically expressed as a percentage of the maximum value of the reference distribution, i.e., global γ), δx is the distance-to-agreement criterion (in mm), and Ω is the spatial domain.
This formulation provides a γ value for each point: $\gamma \leq 1$ indicates that both the dosimetric and spatial criteria are satisfied. To obtain a global quantitative assessment, the percentage of points with $\gamma \leq 1$ (denoted as $\gamma\%$) and the mean value $\langle\gamma\rangle$ can be reported.
Reference [209] states that in RT general guidelines, the universal tolerance limits is set at 95% at 3%/2mm criterion, while the universal action limit is set at 90% at the same criterion. As a design choice, the MSE metric was used only for 1D cases, while both MSE and gamma analysis were employed for higher-dimensional cases. Additionally, except for the feasibility study, the same metrics were applied, for the sake of comparison, to evaluate the discrepancy between the initial and final conditions.

5.5 Preliminary results

In this section, preliminary results are presented, mainly concerning 1D, 2D, and 3D simulated data. Additionally, for the 1D case, validation on experimental data is also provided.

5.5.1 1D distributions: feasibility study

The PINN method for the backward-in-time diffusion BVP was applied to two representative 1D dose distributions: a step function and a rectangular distribution. Subsequently, the method was validated on experimental data from a previous study, fully described in [199]. The results are shown in Figure 5.11.



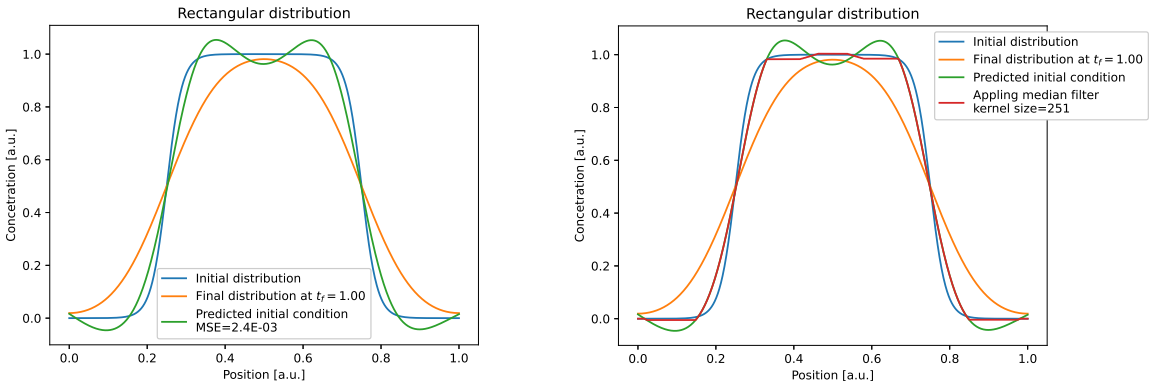
(a) (b)

Figure 5.11: Models predictions of the initial distributions: a) stepside distribution and b) rectangular distribution. Figure adapted from reference [180].

Models showed good predictive performance for the step function (Figure 5.11a), with an

MSE of approximately 4×10^{-4} a.u., while for the rectangular distribution (Figure 5.11b) the MSE is about 1.5×10^{-4} a.u. A notable difference lies in the measurement times, which are 10 and 3 hours, respectively, highlighting the role of the initial distribution. An additional test was performed by evolving the rectangular distribution for 10 h and attempting the reconstruction (Figure 5.12). As can be observed, in this case the initial rectangular shape is almost lost, and the distribution approaches a Gaussian-like profile. This significantly impacts model performance, with $MSE = 2.4 \times 10^{-3}$ a.u., i.e., about one order of magnitude higher than in the previous cases.

However, the results can be improved by applying a median filter. Since in typical irradiation setups prior knowledge of beam characteristics, such as uniformity, is available *a priori*, correction techniques such as median filtering can be applied *a posteriori*. Indeed, as shown in Figure 5.12b (red curve), the quality of the prediction improves, with $MSE = 1.7 \times 10^{-3}$ a.u.



(a) (b)

Figure 5.12: PINN performance on a rectangular distribution after 10 h of diffusion. (a) PINN prediction; (b) prediction after applying a median filter. Figure adapted from reference [180].

Experimental validation

The performance of the PINN model was also evaluated on experimental data obtained from PVA-GTA-B gels, as reported in [199]. The gel, placed in a cuvette of 22 mm in length, was partially irradiated with a high-energy 6 MV clinical X-ray beam at a high dose.

Spectroscopic measurements of absorbance at a wavelength of $585nm$ were carried out at different times after irradiation, namely at 1, 4, 7, and 10 hours [199].

For model evaluation, the profile measured at 1 h (Figure 5.13, blue curve) was used as GT, while the profile measured at 10 h after irradiation (Figure 5.13, orange curve) was used as the starting point. The model was able to reconstruct the distribution back to $t = 1$ h (Figure 5.13, green curve). The agreement with the experimental initial condition at $t = 1$ h is good, with an MSE of approximately 3×10^{-4} OD^2.

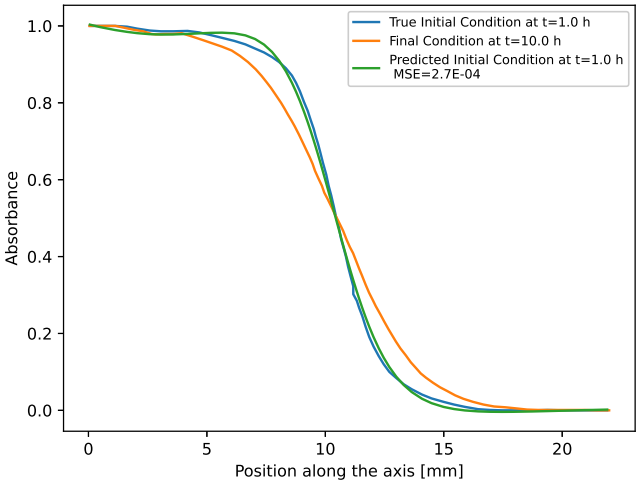


Figure 5.13: Results in predicting the initial distribution of the experimental data. Figure adapted from reference [180].

5.5.2 Higher dimension extension

2D rectangular distributions

Results for the rectangular distribution are shown in Figure 5.14. The left panels display the distributions obtained by numerically integrating Equation 5.14 forward in time, starting from the initial condition shown in the top panel of Figure 5.14.
The PINN method is able to reconstruct the initial rectangular shape from the corresponding diffused distribution, reducing edge broadening and correcting the hyperintense region, as illustrated in the right panels of Figure 5.14.

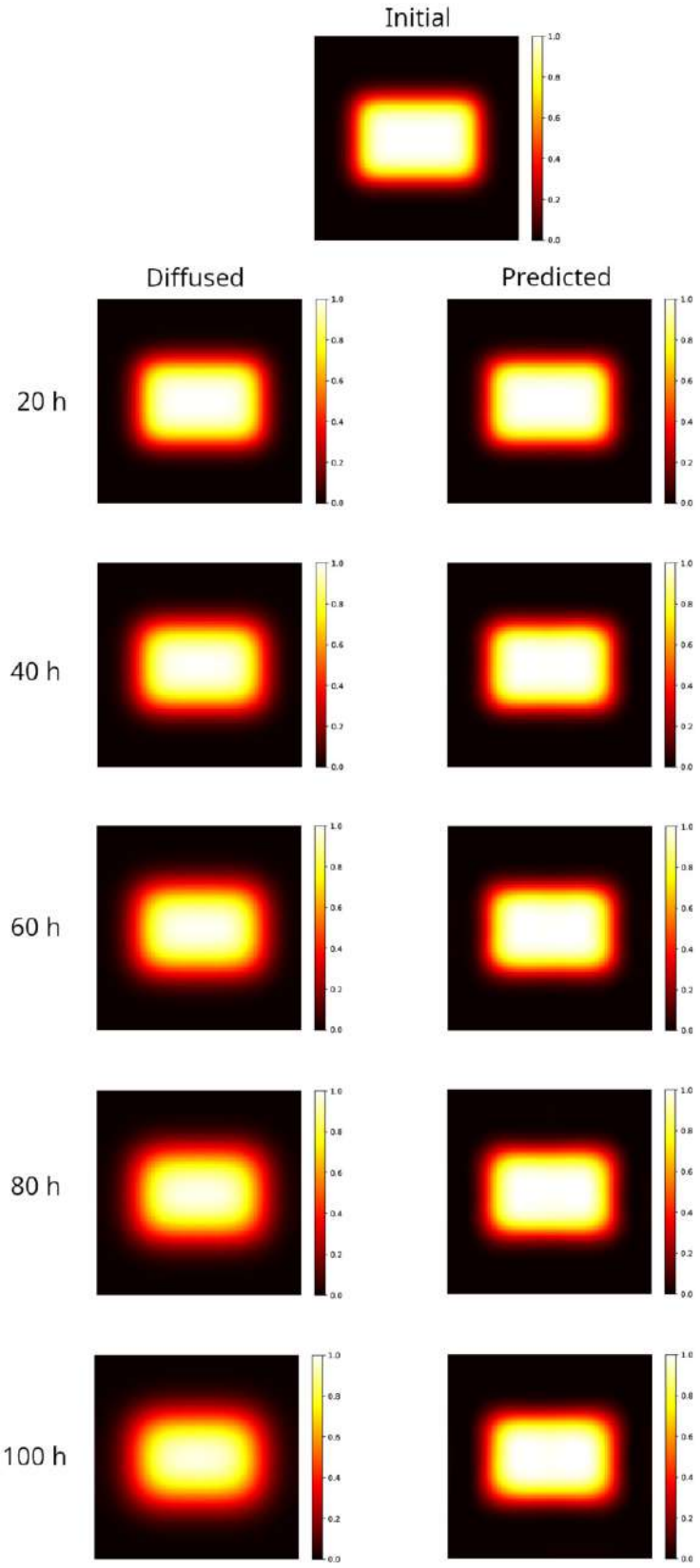


Figure 5.14: Modeling of a 2D sample partially irradiated using an uniform beam (rectangular spot area $40 \times 60\ mm^2$). Top image: initial distribution. Left column: distribution evolved forward in time, after $20 - 100\ h$. Note that as the time increases the hyper-intense area reduces, the edges get larger and the initial rectangular shape approaches an ellipse ($100\ h$). Right column: reconstruction of the initial distribution, starting from the corresponding diffused on the left. The rectangular shape and the hyper-intense area are recovered, the edges width matches the one in the initial distribution at all elapsed times. Figure adapted from reference [181].

Performance evaluation was conducted in accordance with the evaluation criteria discussed in Section 5.4, i.e. using both MSE and γ index analysis. Results are reported in Table 5.2

time (h)	**Distribution**	**MSE**	$\%\boldsymbol{\gamma} < \mathbf{1}$	$\langle\gamma\rangle$
20	diffused	3.44E-04	100 %	0.339
	predicted	4.67E-06	100 %	0.0681
40	diffused	1.13E-03	82.9 %	0.626
	predicted	1.27E-05	100 %	0.0869
60	diffused	2.16E-03	58.7 %	0.899
	predicted	1.94E-05	100 %	0.0992
80	diffused	3.35E-03	44.9 %	1.16
	predicted	4.03E-05	100 %	0.128
100	diffused	4.61E-03	34.6 %	1.41
	predicted	8.54E-05	100 %	0.153

Table 5.2: Rectangular distribution. Error analysis with different metrics utilized as described in Section 5.4. Distances are always evaluated with respect to the true initial distribution. MSE: Mean Square Error; γ: gamma index calculated at $3\%/2mm$. Table adapted from reference [181].

As can be observed, from 40 h onward the diffused distribution is no longer suitable in terms of universal and action limits in radiotherapy [209]. In contrast, even at 100 h (~ 0.04 of the characteristic time, 2500 h), the PINN is able to recover the initial distribution with a gamma index passing rate of 100%.

2D circular distribution

As done for the rectangular distribution, in Figure 5.15 results are shown for the circular distribution evolved forward in time in the interval $20 - 100\ h$ (left column panels), to be compared to the initial distribution (top panel). For sake of comparison, the distributions reconstructed by the PINN method, starting from the corresponding diffused shown in the left panel, are also shown in the right panels of Figure 5.15.

The errors of the prediction with respect to the true initial condition, measured according the different metrics described in Section 5.4, are reported in table 5.3.

In this case, the reconstructions achieve a γ index passing rate of 100% up to 60h, corresponding to 0.0096 of the characteristic time. At 80h, the PINN reconstruction remains above the universal action limit (90% [209]), and the corresponding time ratio is about 0.03, consistent with that observed for the rectangular distribution. At 100h, the reconstructed distribution exhibits a convex shape at its center. This behavior, already observed in the 1D case, is likely due to the use of the hyperbolic tangent as the activation function in the neural network. However, as reported in [181], applying median filtering makes it possible to obtain improved results, reaching a passing rate of 97%.

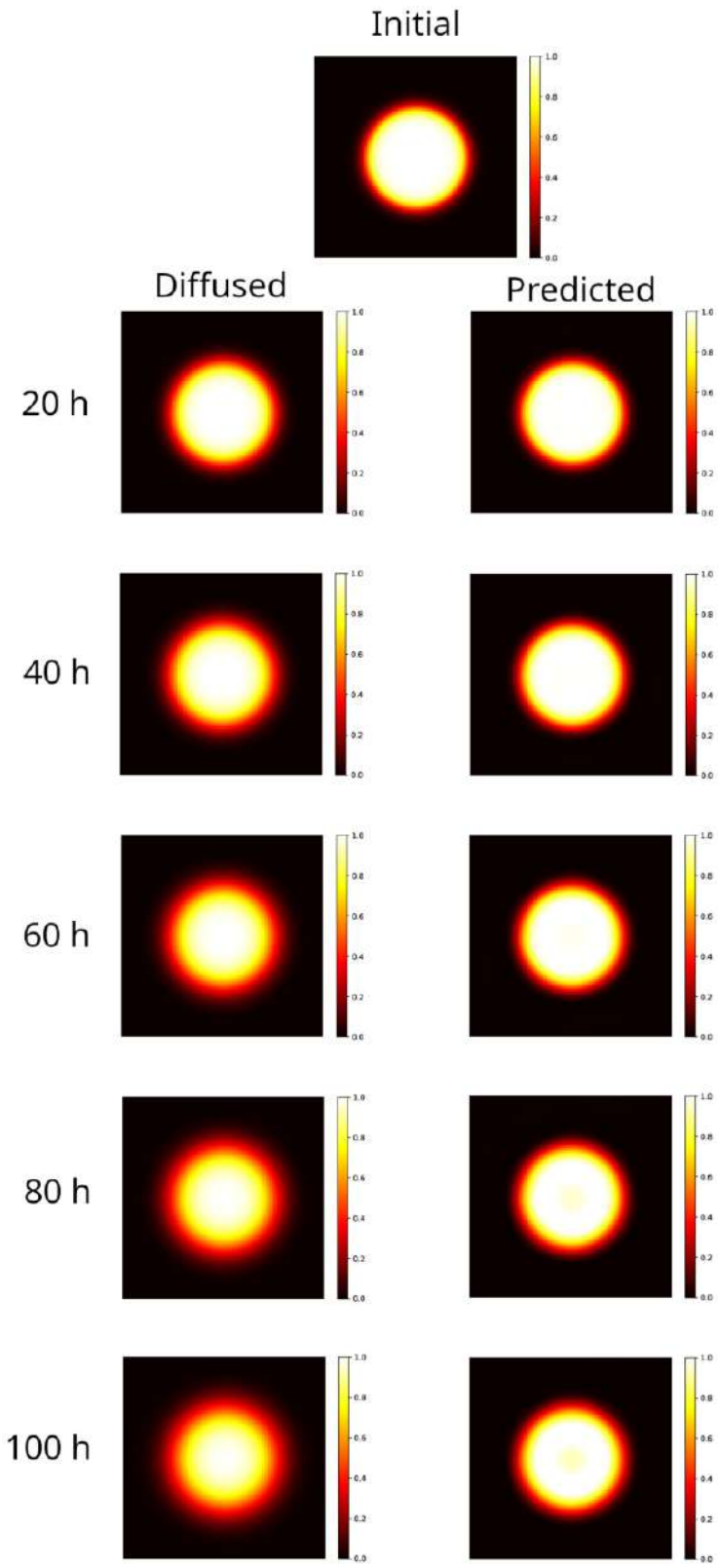


Figure 5.15: Modeling of a 2D sample partially irradiated using an uniform beam (circular spot radius 25 mm). Top image: initial distribution. Left column: distribution evolved forward in time, after $20 - 100$ h. The blurring effect is apparent, as the hyper-intense area is sharp only at the center of the spot, while the edges blur, yet the circular symmetry remains intact. Right column: reconstruction of the initial distribution, starting from the corresponding diffused on the left. With respect to the rectangular case, the reconstruction gets worse as time progresses. At $80 - 100$ h, the model predictions exhibit a loss of signal at the center of the spot that could be however recovered by data post processing (see Figure 3 and 4 in reference [181]). Figure adapted from reference [181].

time (h)	Distribution	MSE	$\%\gamma < 1$	$\langle\gamma\rangle$
20	diffused	9.24E-04	100 %	0.465
	predicted	2.66E-05	100 %	0.118
40	diffused	2.52E-03	58.5 %	0.845
	predicted	8.87E-05	100 %	0.188
60	diffused	4.28E-03	42.1 %	1.20
	predicted	1.92E-04	100 %	0.288
80	diffused	6.07E-03	32.8 %	1.53
	predicted	4.30E-04	91.4 %	0.437
100	diffused	7.83E-03	25.0 %	1.84
	predicted	6.58E-04	84.9 %	0.538

Table 5.3: Circular distribution. Error analysis with different metrics utilized as described in Section 5.4. Distances are always evaluated with respect to the true initial distribution. MSE: Mean Square Error; γ: gamma index calculated at $3\%/2mm$.

3D distribution

The initial distribution, shown in the left panel of Figure 5.16, was generated in accordance with the in-depth dose profile at 7MeV measured experimentally (Figure 5.8, red curve), following the protocol fully detailed in Section 5.2.5. This distribution was then evolved forward in time using the $FTCS$ scheme up to $t_f = 40$h (central panel of Figure 5.16). The initial condition predicted by the PINN method is shown in the right panel of the same figure.

For a clearer visualization, selected axial slices of the 3D distribution are reported in Figure 5.17, for the initial, diffused, and reconstructed configurations. The reconstruction successfully recovers both the hyper-intense region and the overall shape; however, the edges appear contracted and some artifacts are present. In particular, as shown in the right panels of Figure 5.17, the model struggles to accurately reconstruct the corners, where dose gradients are higher. Additionally, Figure 5.18 shows a DDP comparison.

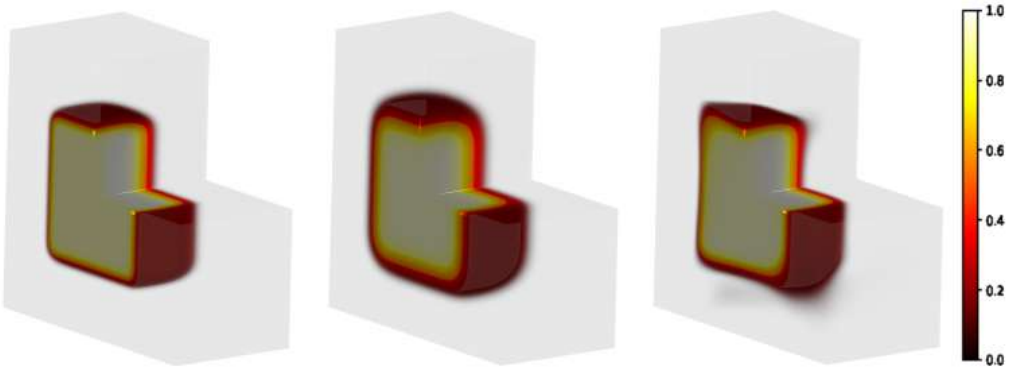


Figure 5.16: Modeling of a 3D sample partially irradiated using an uniform beam (the radiation hits the plan on the left in the figure). Square spot area: 50×50 mm^2. Matrix gel in light grey volume: $100 \times 100 \times 50$ mm^3. Left panel: initial distribution; central panel: distribution evolved forward in time, $t_f = 40$ h; right panel: predicted initial distribution. Figure adapted from reference [181].

Table 5.4 reports the results for MSE and gamma analysis. As can be observed,

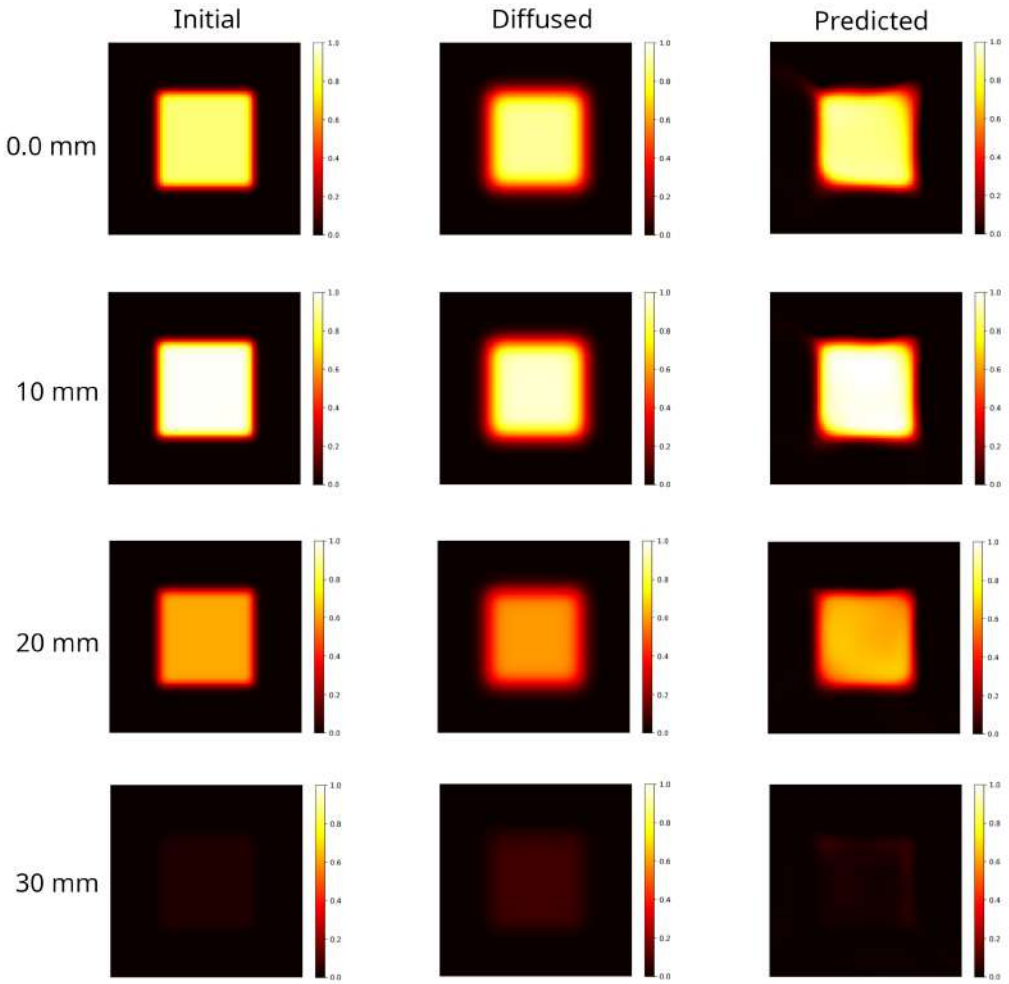


Figure 5.17: Selected axial slices of the 3D distributions shown in Figure 5.16. Left column: initial distribution. With reference to the DDP shown in Figure 5.8, the slices selected correspond to the input dose ($z = 0.0\ mm$); the dose at the maximum value ($z \approx 10\ mm$); the half-maximum dose ($z \approx 20\ mm$); the approximately vanishing dose ($z \approx 30\ mm$). Central column: distribution evolved forward in time, at $t_f = 40\ h$. Blurring effects due to diffusion are evident. Right column: backward time model reconstruction of the initial condition, starting from the corresponding diffused distribution on its left. The PINN method is able to provide a better agreement with the DDP (see Figure 5.18). However, the model introduces artifacts at the edges which deforms the square shape. Figure adapted from reference [181].

the diffused distribution shows very poor performance in terms of γ index passing rate, whereas the PINN reconstruction achieves a value above the universal action limit [209]. These results clearly demonstrate the advantage of using the PINN algorithm for distribution deblurring. Considering the time corresponding to the diffused distribution (40h) and the characteristic time (1666h), the ratio between reconstruction time and characteristic time is approximately 0.03, in agreement with previous results.

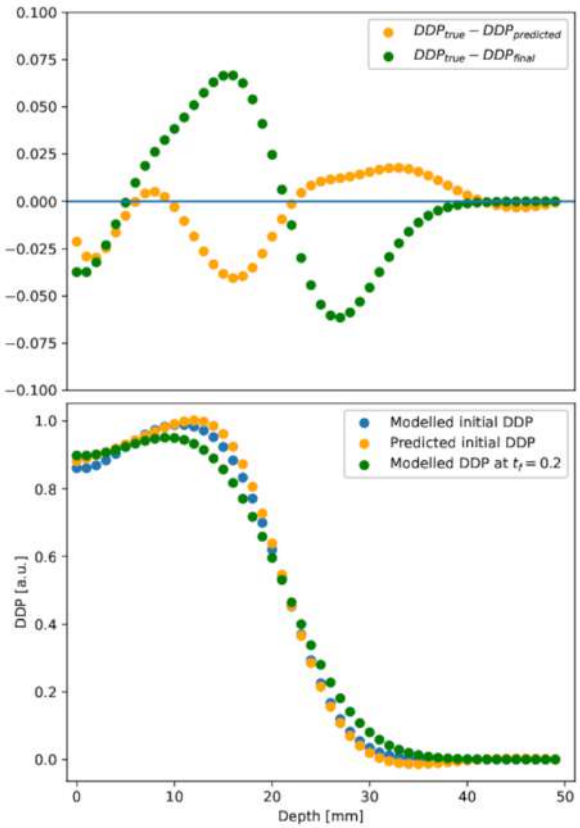


Figure 5.18: Depth dose profile. Lower panel: depth dose profile in the initial, diffused at $t_f = 40\ h$, and predicted initial conditions. Upper panel: difference between diffused or predicted initial depth dose profile and the initial one. The PINN result (orange points) is in good agreement with the initial distribution, at variance with the diffused (green points), in almost all depth range. Figure adapted from reference [181].

time (h)	**Distribution**	MSE	$\%\gamma < 1$	$\langle\gamma\rangle$
40	diffused	1.85E-03	59.8 %	0.905
	predicted	9.15E-04	92.4 %	0.403

Table 5.4: 3D distribution. Error analysis with different metrics utilized as described in Section 5.4. Distances are always evaluated with respect to the true initial distribution. MSE: Mean Square Error; γ: gamma index calculated at $3\%/2mm$.

5.6 PINN method on MTB-PVA gel: application to FGs affected by auto-oxidation

In this section, a study on the application of the PINN method to MTB-doped FGs is presented. The experimental activities have been conducted during a period of visiting in Milan. In these gels, auto-oxidation (AO) is not negligible; therefore, the study requires modeling the combined effects of diffusion and AO in terms of a reaction–diffusion equation, which will be discussed in detail.

Given the notable differences in gel type, measurement system, and underlying theory, this study is presented as a separate section.
First, an overview of the gel type, irradiation setup, measurement system, and data

processing is provided. Subsequently, starting from the initial results, the theoretical framework developed to account for AO is introduced. Finally, results obtained for 1D dose profiles and small-field planar distributions are presented.

5.6.1 Gel preparation

Fricke-type hydrogels incorporating methylthymol blue (MTB) were synthesized at the Department of Physics of the University of Milan, adopting a protocol reported in the literature [210], in which xylenol orange was replaced with 0.2 mM MTB. Two PVA-based formulations were prepared, differing in both equivalent molecular weight and component ratios. Following fabrication, the gels were cast into planar PVC containers (slices).

Because the ambient temperature exceeded $25°C$, the resulting gel slices contained air inclusions. In addition, to mitigate AO throughout the experimental campaign, the samples were stored at 4°C; however, this led to condensation-related artifacts.
The results presented here refer to the samples summarized in Table 5.5, together with their compositions and respective roles within the experimental framework.

Sample ID	PVA mixture (by fraction)	Approx. MW	Dose/readout setup
Gel A	100% 18-88	∼130 kDa	Linear (1D)
Gel B	30% 18-88 / 70% 4-88	mixed	Linear (1D)
Gel C	100% 18-88	∼130 kDa	Planar small-field (2×2 cm^2)

Table 5.5: Gel slices summary and experimental use. "Linear" indicates slices used for 1D linear-profile analysis; "Planar" indicates small-field 2D irradiation.

5.6.2 Irradiation

Irradiations were carried out using a clinical linear accelerator (Varian Clinac, Varian Medical Systems, CA, USA) at the "Fondazione IRCCS Istituto Nazionale dei Tumori" in Milan. The gel slices were positioned under a 5 cm thick tissue-equivalent slab and set at a source-to-surface distance of 50 cm.

For samples used to obtain linear dose profiles, an approximately uniform irradiation over half of the gel volume was applied. In the case of small-field experiments, a $2 \times 2 cm^2$ open field was delivered (gel C). The delivered dose was about 8 Gy. Photographic documentation of the irradiation setup and slice configuration is provided in Figure 5.19.

5.6.3 Measurements System

Measurements of gel absorbance were performed at the Department of Physics "Aldo Pontremoli", University of Milan, using an optical read-out system fully described

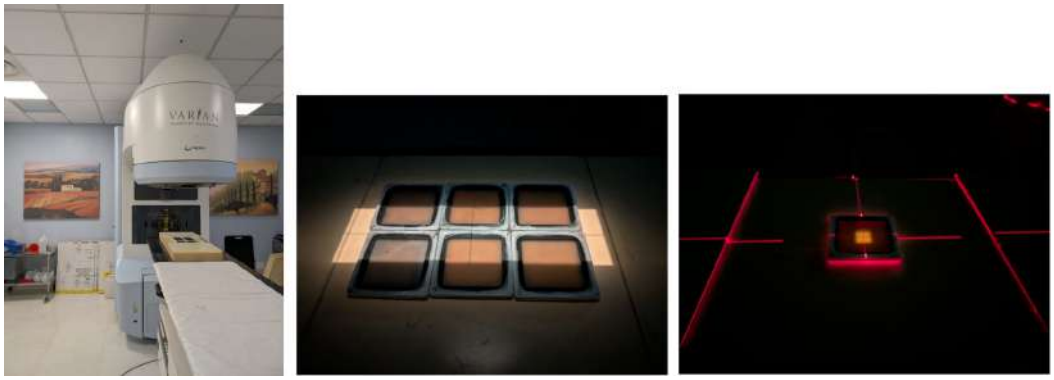
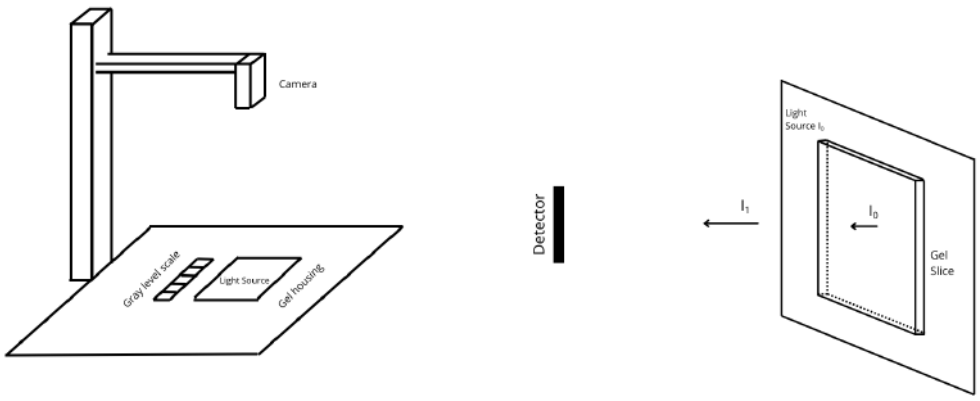


Figure 5.19: Irradiation setup. Left panel: gel positioning and LINAC at the "Fondazione IRCCS-Istituto Nazionale dei Tumori" in Milan. Central panel: Irradiation setup for linear dose profile. Right panel: small-field irradiation setup (2×2 cm^2).

in [211]. A brief overview of the setup and measurement procedure is provided here for completeness.

The measurement system consists of a light source and a CCD camera equipped with a low-bandpass filter centered at 600nm, with a full width at half maximum (FWHM) of 10nm, matching the absorbance peak of the MTB-doped Fricke gel. The system configuration and optical path are illustrated in Figures 5.20.



(a) (b)

Figure 5.20: (a) Diagram of the main measurement system. The entire setup is enclosed in a box to shield background light. (b) Scheme of the light path. The light of initial intensity I_0 is emitted from the source, passes through the gel, and is recorded by the detector (intensity I_1).

Following the optical path shown in Figure 5.20b, the gel absorbance A is defined as:

$$A = \log\left(\frac{I_0}{I_1}\right) \tag{5.36}$$

where I_0 is the light intensity emitted by the source and I_1 is the detected intensity after transmission through the medium.
Since the absorbed dose is proportional to the change in absorbance after irradiation,

assuming a stable light source ($\frac{\partial I_0}{\partial t} = 0$), it is possible to derive:

$$D \propto \Delta A = \log\left(\frac{I_0}{I_{1,R}}\right) - \log\left(\frac{I_0}{I_{1,NR}}\right) = \log\left(\frac{I_{1,NR}}{I_{1,R}}\right) \tag{5.37}$$

where the subscripts R and NR denote the irradiated and non-irradiated gel, respectively.

In summary, the measurement of absorbance variations induced by ionizing radiation is performed by comparing, point by point, the detected light intensity (i.e., the intensity I_1 in Figure 5.20b) between irradiated and non-irradiated gels. An example of acquired data is provided in Figure 5.21.

(a) (b) (c)

Figure 5.21: Measurement procedure. Images of the non-irradiated gel are shown in (a), and of the irradiated gel in (b). In both cases, image cropping is applied to remove irrelevant details and artifacts. The ΔA map is then computed on a pixel-wise basis using Equation 5.37, resulting in the image shown in (c).

Within this framework, the computation of ΔA appears to be independent of the source intensity I_0. However, this holds only if the illumination is spatially uniform over the entire gel surface. In practice, this condition is not always satisfied; therefore, the validity of the calculation requires that the gel be positioned identically for each measurement. Despite careful alignment procedures, this remains one of the main sources of noise in the measurements.

Nevertheless, the simplicity of the setup and its relatively low implementation cost make this system a promising option for FG applications.

5.6.4 Data processing

All image preprocessing steps were implemented in Python using OpenCV [212] and standard scientific computing libraries [90, 165–167].

1. **Co-registration.** For each slice, all images in the time series were aligned to the non-irradiated reference image using an affine transformation estimated through feature-based routines in OpenCV. This transformation accounts for translations, rotations, and small scale variations. The same mapping was subsequently applied to the corresponding ΔA images.

2. **Cropping.** After registration, a rectangular region of interest was manually defined once per slice to remove image borders and regions affected by macroscopic defects (e.g., air bubbles outside the central measurement area). The same crop was then consistently applied to the entire time series to ensure spatial coherence in temporal analyses.

3. **Distributions extrapolation.** 1D and 2D distribution were evaluated after preprocessing step (above steps) and artifacts removing as described below.

Distributions extrapolation

The preprocessing steps (Co-registration and Cropping) provide results as depicted in Figure 5.21 c. As can be observed, presence of air bubbles translates into artifact in recorded data. These anomalies can be reduced by adopting spatial average techniques.

Linear profile extraction An image can be see as matrices $M \times N$, where M is the number of pixel along the vertical direction (y) and N is the same number along the horizontal direction (x). For linear profiles (Figure 5.21 c) the translational invariance is along the horizontal direction can be exploited to reduce noise and artifacts. Since the behavior along the y direction is common for all matrix columns, the best estimation for ΔA is given by mean, point by point, along the x direction. In this way the 2D distribution $\Delta A(x, y, t)$ can be reduced in a 1D linear profile as:

$$F(y,t) = \frac{1}{N_x} \sum_{i=1}^{N_x} \Delta A(x_i, y, t). \tag{5.38}$$

Representative results are depicted in Figure 5.23 left panels. Additional smoothing function can be applied to improve data quality.

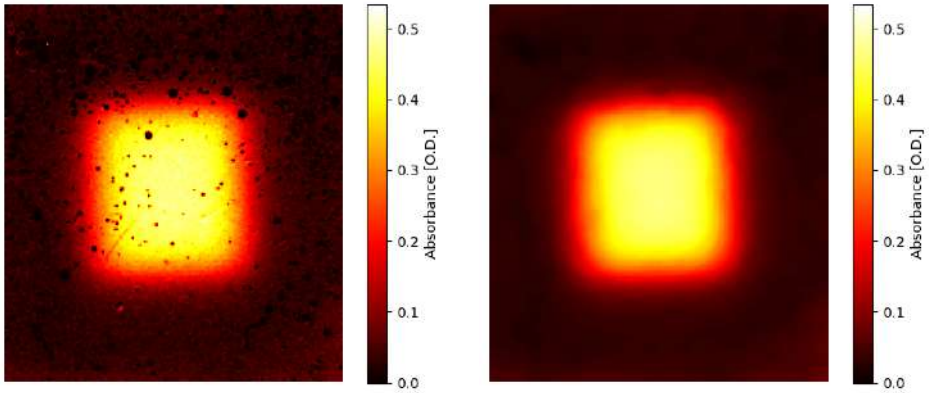
Small-field distribution In this case, translational invariance cannot be exploited. However, to reduce noise and artifacts, a median filter (kernel size = 11) is applied, yielding the results shown in Figure 5.22.

5.6.5 AO and diffusion coefficients estimation

1D linear distributions

The procedure described above, for linear profiles, extrapolation was applied to Gel A and B. Results are depicted in Figure 5.23 left panels. The central panels show linear profiles after the application of a convolutional smoothing function (kernel size 9).

In both left and central panels the effect of AO is shown: area under the curve increases with time. Kron *et al.* procedure for diffusion coefficient estimation [187] cannot be applied on these distribution, thus it is necessary remove the AO effect.



(a) (b)

Figure 5.22: Small field distribution at 0.8 h shown using a 'hot' color scale. (a) Raw distribution; (b) after application of a median filter ($kernel\ size = 11$).

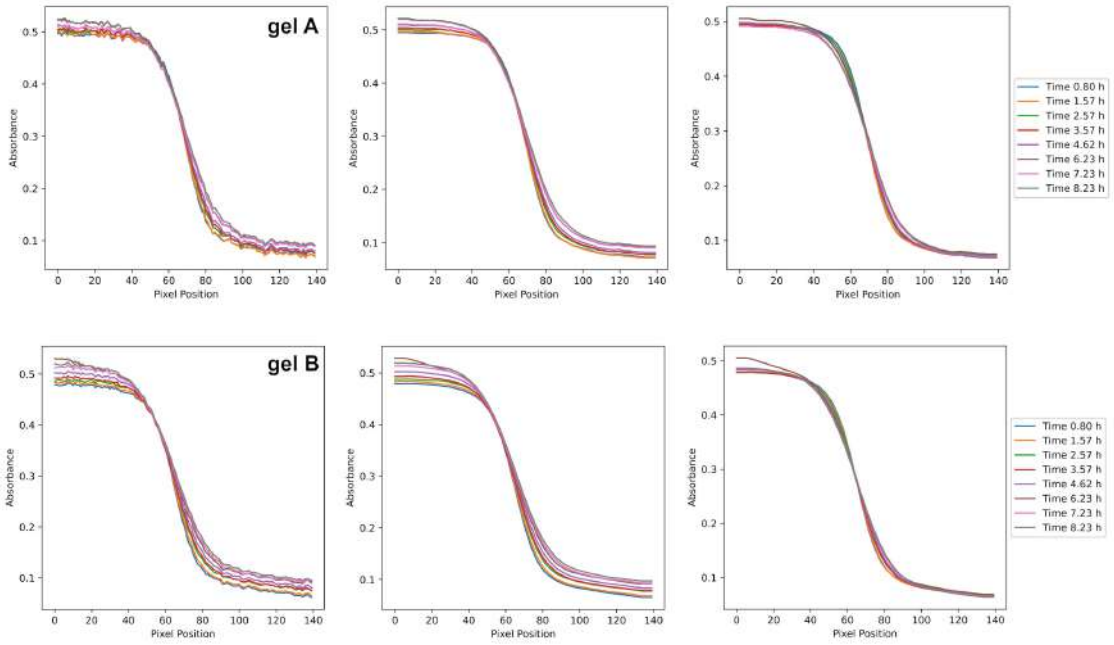


Figure 5.23: Left panels: raw linear profiles for gels A and B. Central panels: smoothed profiles, by applying a smoothing function (convolutional smoothing with kernel size of 9) to remove experimental fluctuations. Right panels: corrected linear profiles for gels A and B. Auto-oxidation component has been removed in accordance to equations 5.40 and 5.42. After this analysis the areas are almost constant over the time and all curves have the same inflect point, as it is expected in a diffusion process without auto-oxidation.

Figure 5.24 shows the area under the curve as function of time for both gels. The area exhibits a linear behavior within the time range $1 - 10h$.

AO has typically been investigated as a global property, providing overall AO rates for the gel system [186, 201, 202]. In addition, it has mainly been studied to assess gel sensitivity in pre-irradiation conditions.

The simplest assumption to address AO in the present study is to consider it spatially uniform. This assumption can be justified by noting that, in the central panels of Figures 5.23, the ratio between maximum and minimum values remains constant over

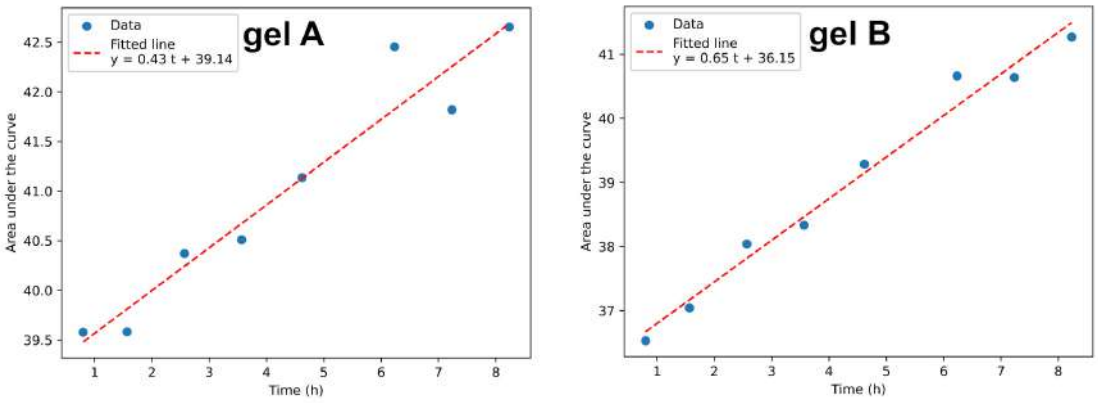


Figure 5.24: Area under the curve as function of the time for gels A and B. In all cases it follows a linear relation: $A(t) = A_0 + kt$.

time. However, this assumption is valid only within a limited time window, before saturation effects become significant.

Under the assumption of uniform AO, the 1D linear profile can be written as:

$$F(y,t) = f(y,t) + g(t) \tag{5.39}$$

where $f(y,t)$ represents the part of the profile affected by diffusion, and $g(t)$ accounts for AO. By integrating Equation 5.39 over space, we obtain $A(t)$, the area under the curve as a function of time:

$$A(t) = \int_{y_0}^{y_L} F(y,t)\ dx = \int_{y_0}^{y_L} f(y,t)\ dx + g(t) \int_{y_0}^{y_L} dx = \int_{y_0}^{y_L} f(y,t)\ dx + g(t) L_y \tag{5.40}$$

where y_0 and y_L represent the initial and final positions of the profile, and $L_y = y_L - y_0$ denotes the length of the gel ($35mm$ after image cropping). Since $f(y,t)$ describes only the diffusion process—where ions move but do not disappear—its integral over the gel does not depend on time and represents the amount of ferric ions produced by irradiation, denoted as A_0. Therefore, we obtain:

$$A(t) = A_0 + g(t) L_y. \tag{5.41}$$

Since the area under the curve has a linear relationship with time, as shown in Figure 5.24, the function $g(t)$ can be determinated as follows:

$$g(t) = \frac{k}{L_y}\ t \tag{5.42}$$

where k is the slope of the line that approximates the data points in figure 5.24.

At this point, calculating the time derivative of $F(y,t)$, one obtains:

$$\frac{\partial}{\partial t}F(y,t) = D\frac{\partial^2}{\partial y^2}F(y,t) + \frac{k}{L_y} \tag{5.43}$$

which is the diffusion equation with a constant source term $S = \frac{k}{L_y}$ [1].

Thanks the ansatz in Equation 5.39 and the $g(t)$ definition (Equation 5.42), it is possible remove AO from linear profiles in gel A and B. Results are depicted in Figure 5.23 rigth panels. The corrected linear profiles for each gel collapse to a family of curves with a common inflection point and nearly time-invariant integrated area, as expected for a purely diffusive temporal dynamics. Consequently, estimations for diffusion coefficients can be performed in accordance the methods proposed in references [178, 187]. Results are reported in Table 5.6.

Gel slice	$D\ [mm^2/h]$	$\Delta D\ [mm^2/h]$
Gel A	0.25	0.02
Gel B	0.22	0.02

Table 5.6: Diffusion coefficients estimation for gels used in linear DD.

2D planar distribution

The same approach for AO term estimation can be used in the 2D distribution, with the only difference that the integral in Equation 5.40 becomes a surface integral, thus the $g(t)$ function is given by:

$$g'(t) = \frac{k'}{L_x L_y}t \tag{5.44}$$

where k' is the slope of the line that approximates the selected data points in Figure 5.25, and L_x and L_y are the gel slice edges ($area = 42 \times 47 mm^2$, after image cropping).

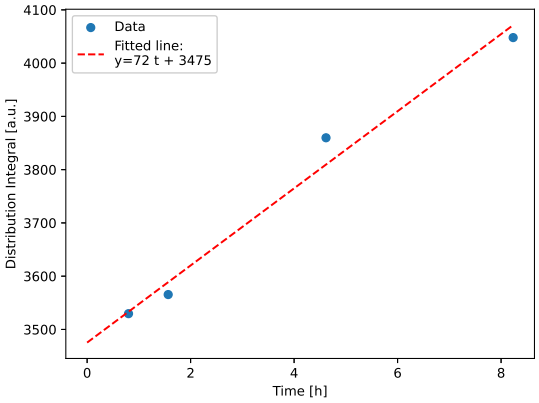


Figure 5.25: Distribution integral as function of time for gel C.

[1] To obtain this result it was used the relation $\frac{\partial^2}{\partial y^2}F(y,t) = \frac{\partial^2}{\partial y^2}f(y,t) + \frac{\partial^2}{\partial y^2}g(t) = \frac{\partial^2}{\partial y^2}f(y,t)$.

The estimation of the diffusion coefficient using the 1D curvature method is not applicable to the small-field planar irradiation data. In this case, the value of D obtained from gel A (with matched composition) was used.

To validate the parameters k' and D, the initial distribution was numerically evolved forward in time up to 8.2 hours using the approach described in Section 5.2.4 . The resulting simulated distribution (Figure 5.26, left panel) was then compared with the experimentally measured one at the same time point (Figure 5.26, right panel).

γ analysis yielded a passing rate of 93% at $3\%/2,\text{mm}$, exceeding the universal action limit of 90% for the same criterion [209]. This supports the validity of the proposed approach.

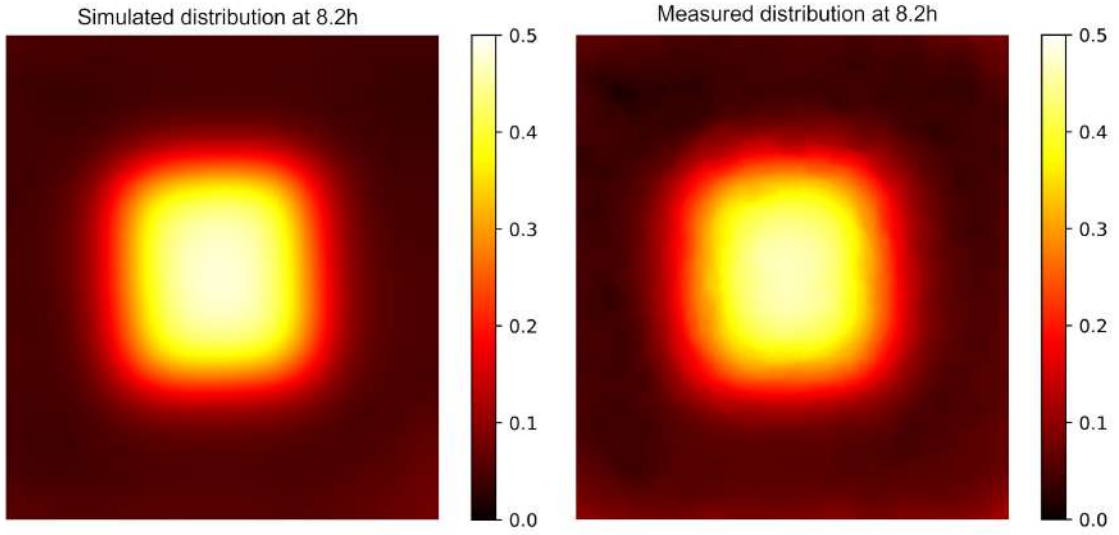


Figure 5.26: Planar distributions s; the colorbar indicates absorbance in OD units. Left panel: distribution after forward simulation up to $8.2h$; right panel: experimentally measured distribution at $8.2h$.

5.6.6 PINN training design

The PINN correction algorithm follows the formulation described in Section 5.3.2 and is applied to both 1D and 2D distributions. The only modification concerns the definition of the residual loss $\mathcal{L}_{res}$, which is expressed as:

$$\mathcal{L}_{res} = \frac{1}{N_1}\sum_{i=1}^{N_1}\left(\frac{\partial}{\partial t}NN(\mathbf{x}_i, t_i; \boldsymbol{\theta}) - D\nabla^2 NN(\mathbf{x}_i, t_i; \boldsymbol{\theta}) - S\right)^2 \tag{5.45}$$

where N_1 denotes the number of sampled points in $\Omega \setminus \partial\Omega \times [0, t_M)$, and S is the source term, defined in Equations 5.42 and 5.44 for the 1D and 2D cases, respectively.

The networks were trained for 50,000 epochs for gels A and B, and 100,000 epochs for gel C. A stopping criterion based on a total loss threshold of 10^{-4} was adopted. At each epoch, $N_1 = 20{,}000$ interior points, $N_2 = 20{,}000$ boundary points, and $N_3 = 10{,}000$ data points at t_m were sampled. Model checkpointing and monitoring of the individual

loss components were employed as regularization strategies. All data were expressed in physical units (distance in mm, time in h, and diffusion coefficient in mm^2/h), with a pixel-to-millimeter conversion factor of 0.25.

The training time for a single model was approximately 30 minutes, leading to a total computational time of about 2.5 hours on a workstation equipped with an NVIDIA GeForce RTX 4090 GPU. This time approximately doubles for the 2D case.

5.6.7 Results

1D linear profiles

The PINN algorithm was applied to the latest measured linear profile (8.2 h post-irradiation) to reconstruct the earliest measurement (0.8 h post-irradiation). To assess the stability of the results, multiple training runs were performed by varying the random seed used for parameter initialization. This procedure was carried out for both gels A and B. The corresponding results are reported in Figure 5.27.

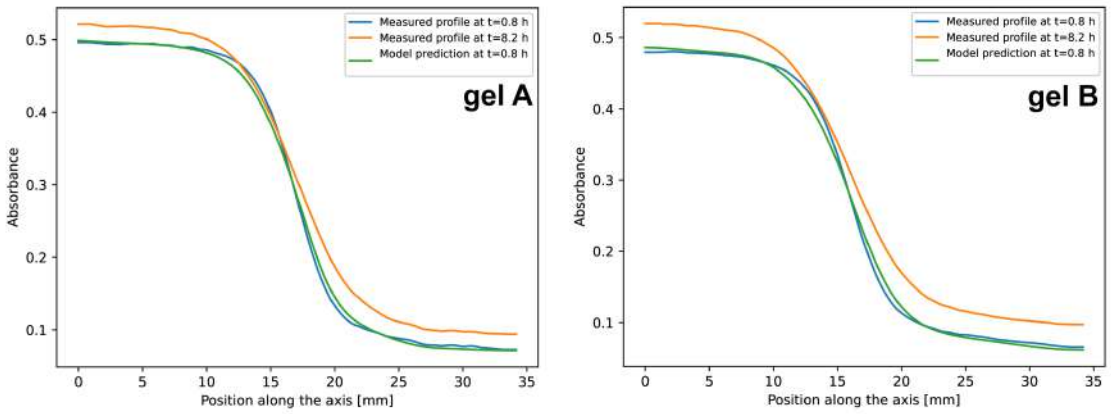


Figure 5.27: Results of the PINN-based correction for gels A and B. For each case, three profiles are reported: the earliest measured profile (GT), the latest measured profile (PINN input), and the PINN-reconstructed earliest profile (averaged over multiple network trainings).

Quantitative evaluation was conducted in accordance with criterion introduced in Section 5.4. Since it is a 1D evaluation MSE only was considered. Table 5.7 summarizes the observed metrics.

Sample ID	time first profile [h]	time last profile [h]	MSE last profile $[OD^2]$	MSE PINN correction $[OD^2]$
Gel A	0.8	8.2	7.0×10^{-4}	$(5.3 \pm 0.1) \times 10^{-5}$
Gel B	0.8	8.2	1.3×10^{-3}	$(5.8 \pm 0.1) \times 10^{-5}$

Table 5.7: PINN 1D reconstruction performance. MSE values are given in OD^2. PINN results are reported as average obtained across the different NN realization.

2D planar distribution

Using the validated parameters, the PINN was trained to reconstruct the initial planar distribution ($t = 0.8$ h) from the latest measured map ($t = 8.2$ h). The performance was evaluated by combining MSE and gamma analysis (GA) with a 2%/2 mm criterion. The results, reported in Figure 5.28 and Table 5.8, highlight the effectiveness of the PINN approach.

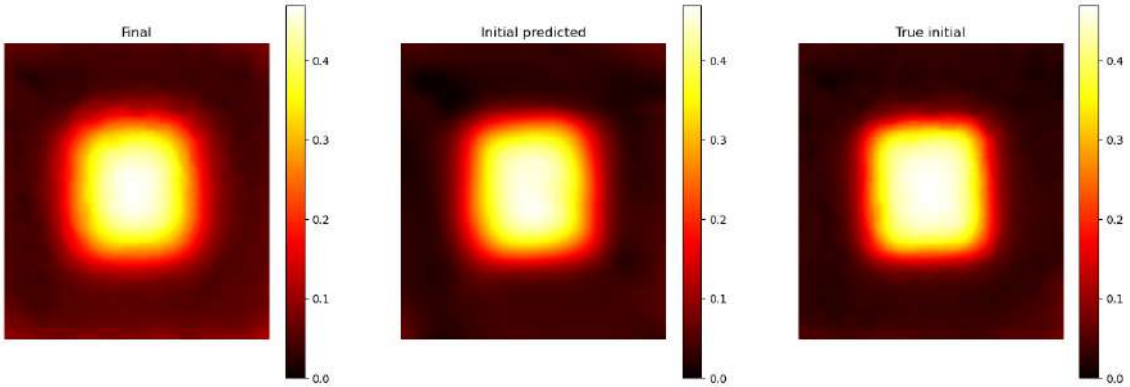


Figure 5.28: Planar distributions; the colorbar represents absorbance in OD units. Left panel: experimentally measured distribution at 8.2 h; central panel: PINN-reconstructed distribution at 0.8 h (averaged over multiple network trainings); right panel: experimentally measured distribution at 0.8 h (reference initial distribution).

time [h]	Distribution	MSE $[OD^2]$	$\gamma\%$	$< \gamma >$
8.2	Final	$6.2 \cdot 10^{-4}$	87	0.57
	Initial Predicted (at $0.8h$)	$(1.5 \pm 0.2) \cdot 10^{-4}$	95 ± 2	0.38 ± 0.02

Table 5.8: PINN 2D reconstruction performance. For GA a $2\%/2mm$ criterion was applied. PINN results are reported as average result across different NNs.

The PINN-reconstructed initial planar map shows a markedly improved agreement with the independently measured early-time distribution, as indicated by a lower MSE and a higher gamma passing rate (95%), compared to the final measured map (87%). These findings support the use of PINNs to recover higher-resolution dose distributions from delayed optical readouts.

5.7 Discussion

In this chapter, a study on the application of artificial intelligence to FG dosimetry is presented. FGs are chemical dosimeters capable of mapping three-dimensional dose distributions and, thanks to this property, they have attracted significant interest in the field of RT [178]. However, their clinical application is limited by diffusion and auto-oxidation AO artifacts [188–192]. Most previous research has focused on materials science, aiming to optimize the chemical composition in order to reduce ion mobility. In contrast, the present work focuses on the development of an AI-based post-processing technique to recover the initial dose distribution from blurred measurements [180, 181].

The proposed methodology, based on Physics-Informed Neural Networks, aims to solve the backward-in-time diffusion equation, acting as a regularization strategy to address the ill-posedness of the BVP. Furthermore, the methodology is independent of the dosimeter readout technique and can be extended to MRI-based acquisitions. The results demonstrate a strong capability in reconstructing the initial dose distribution, while ensuring explainability, robustness, and reproducibility. The PINN approach also proves to be versatile and adaptable across different scenarios, varying in dimensionality and gel composition. Notably, the method shows excellent performance even in the presence of combined diffusion and AO effects, further supporting its relevance in overcoming key limitations of FG dosimetry.
In the feasibility study, the general algorithm (Section 5.3.2) was established, providing a consistent framework to address the backward-in-time diffusion BVP. Extensions to 2D and 3D simulated distributions demonstrated the model's capability to handle more complex scenarios. Finally, the study on MTB-FGs confirmed the effectiveness of the PINN approach when applied to real experimental dosimetric data. The theoretical framework presented here also provides insights into the interpretation of previously unexpected results.
These findings motivate further investigation in this field, with the ultimate goal of supporting the clinical translation of Fricke gel dosimetry in RT.

Despite these advantages, the proposed method presents some limitations. In this study, the PINN approach was applied to simulated or experimentally controlled conditions. Its application requires prior knowledge of the diffusion coefficient and, when AO effects are non-negligible, the corresponding reaction rate. While these parameters were estimated here through controlled experiments or defined in simulations, in real-world applications they must be known *a priori*. This requirement is quite restrictive, as it implies the need for advanced preparation and characterization techniques that are not yet standard in FG dosimetry. A solution could be offered by integrating parameters estimation directly within the PINN framework. In fact, PINNs are also well known to be used in parameters estimation in inverse problems [65].
A second limitation concerns the dose distributions used to validate the approach. In all cases, relatively simple geometries and uniform irradiation conditions were considered. Although these scenarios cover a significant portion of RT applications, FG dosimetry is particularly valuable in highly complex irradiation setups. For instance, in Intensity-Modulated radiotherapy (IMRT), where beam intensity is modulated according to anatomical structures and tumor location, dose distributions can exhibit highly intricate spatial patterns. Conventional detectors often struggle to capture such complexity, whereas FGs represent a promising alternative.

5.8 Conclusions and Future perspectives

The PINN approach represents a novel AI-based technique for improving the accuracy of Fricke gel dosimetry. Its application across different gel systems has demonstrated strong performance in reconstructing the initial dose distributions from degraded measurements, highlighting its robustness and adaptability.

Future work will focus on extending the methodology to fully 3D experimental dose distributions and to more complex irradiation setups. In particular, investigating scenarios characterized by highly heterogeneous and modulated dose patterns—such as those encountered in advanced radiotherapy techniques—will be essential to assess the full potential of the approach.

Further developments should also address current limitations, including the dependence on accurate knowledge of physical parameters (e.g., diffusion coefficients and auto-oxidation rates) and the need for improved experimental standardization. Integrating parameter estimation directly within the PINN framework could represent a promising direction to enhance the method's applicability in real-world conditions.

Ultimately, the goal is to establish a reliable and general-purpose technique capable of supporting gel dosimetry in challenging scenarios where conventional detection systems are limited, thereby contributing to more accurate dose verification and improved patient safety in radiotherapy.

CONCLUSIONS AND FUTURE PERSPECTIVES

Artificial Intelligence has pervaded almost all aspects of our lives. Its use in the last decade has expanded to an uncountable number of applications. In the medical field, the number of publications reflects the same trend. AI has gained such a significant impact that it may be considered one of the defining innovations of the 21st century.

In this PhD thesis, three AI applications in the field of medical physics are presented and discussed: (i) the development and validation of a segmentation model for the Ventral Intermediate Nucleus, designed to assist Essential Tremor treatments, enable real-time treatment planning, and enhance personalized medicine; (ii) the development of a Quality Assurance protocol, along with a software pipeline, for AI-powered auto-contouring tools in radiotherapy, specifically focused on the segmentation of organs at risk; (iii) the development of a Physics-Informed model capable of mitigating the blurring effect in dosimetry applications involving Fricke hydrogels.

For all these topics the concept of Turing's Test was adopted as a unifying theme, emphasizing the crucial role of validation processes and the need for rigorous evaluation to support the clinical translation of AI.

In the first application (Chapter 3), Turing's Test can be interpreted as the evaluation criterion used to assess the performance of DeLTA-BIT in generating regions of interest comparable to those obtained via tractography. Additionally, the *external validation* included in DeLTA-BIT_v2—where the model's predictions were compared with the ablated region in TcMRgFUS-treated Essential Tremor patients—can be regarded as an *Expert Turing's Test*. In the second application (Chapter 4)—where protocols were developed to ensure model stability, reproducibility, and compliance with quality standards—the *Acceptance QA* itself can be interpreted as a form of Turing's Test. Finally, in the third application (Chapter 5), the adopted metrics—particularly the γ index analysis—together with comparisons against ground truth and validation on experimental data, were designed in analogy to a Turing's Test.

The entire work was conducted with particular attention to its impact on real-world clinical practice.

In Chapter 3, although good results in terms of DSC were achieved in the internal validation, the external validation yielded DSC values lower than 0.30. This finding, although comparable with a well-established atlas-based method achieving a similar DSC range, highlights the challenges in translating model performance from controlled datasets to real-world clinical data.
In principle, this issue could be mitigated by training exclusively on clinical data. However, the number of available datasets is not sufficient to optimize the large number of parameters typical of a 3D U-Net. Therefore, a more feasible approach is to train models on large controlled datasets and subsequently apply fine-tuning using clinical data. This strategy is planned for next future work.

In Chapter 4, a real example of how AI can be integrated into clinical practice was presented. The developed QA protocols represent a powerful tool to enable human oversight of commercial AI-powered software. These strategies aim to preserve the clinical framework and ensure stability, reproducibility, traceability, and compliance with quality standards. In particular, it was shown that model upgrades cannot be passively accepted without human verification. This finding is consistent with statements in comprehensive regulatory frameworks concerning AI, such as the European AI Act.
Analogously to Chapter 3, the results showed that fine-tuning should be applied to improve performance on clinical data. In the author's opinion, AI companies should allow for in-house fine-tuning while preserving commercial rights. To the best of the author's knowledge, only in-house fine-tuning can guarantee optimal performance on clinical data.

Chapter 5 represents the most mature work of this PhD thesis. The PINN method was first designed and developed using simulated data; it was subsequently extended to 2D and 3D dose distributions and validated on 1D and 2D experimental data. During its development and testing, the theoretical framework supporting the findings was further expanded and refined.
Overall, the results indicate that the PINN method for correcting diffusion and auto-oxidation artifacts is highly promising for Fricke gel dosimetry. Its main advantage is that it operates solely on data, without requiring additional experimental input. It is a post-processing technique, yet it demonstrates very promising performance.
Future work may include the investigation of 3D experimental dose distributions and more complex irradiation setups, such as those used in Intensity-Modulated Radiotherapy. Additionally, convolutional PINN models could be explored, particularly in 3D applications, where training is currently very time-consuming and computationally demanding.

This thesis is the result of a three-year PhD program entitled *Technologies and Sciences for Human Health*, in which AI has played the role of an innovative tool in

medical physics research.
Overall, the results obtained during this PhD program support the clinical translation of AI in the medical field, while emphasizing the necessity of rigorous validation procedures and continuous human oversight to ensure model reliability and safety.

5.9 Scientific publications

This section presents the publications arising from this research, categorized as either peer-reviewed journal manuscripts or conference abstracts.

5.9.1 A segmentation model for Ventral Intermediate nucleus identification

Journal articles

[1] **M. Romeo**, C. Gagliardo, G. Cottone, G. Collura, E. Maggio, C. Runfola, E. Bruno, M. C. D'Oca, M. Midiri, F. Lizzi, I. Postuma, M. D'Amelio, A. Lascialfari, A. Retico, and M. Marrale. *Delta-bit: an open-source probabilistic tractography-based deep learning framework for thalamic targeting in functional neurological disorders*, 2026. URL: arXiv:2312.15462.

Congress contributions

[1] M. Marrale, **M. Romeo**, C. Gagliardo, G. Cottone, G. Collura, C. Runfola, E. Maggio, E. Bruno, M.C. D'Oca, and M. Midiri. *Deep learning models for targeting in neurosurgical treatments with transcranial mr-guided focused ultrasound surgery (tcmrgfus).* 2025. 111° Congresso Nazionale SIF. URL: `https://hdl.handle.net/10447/690404`

[2] **M. Romeo**, G. Collura, C. Gagliardo, T.V. Bartolotta, L. La Grutta, F. Lizzi, I. Postuma, A. Retico, A. Lascialfari, and M. Marrale. *Deep learning models for targeting of transcranial-mrguided focused ultrasound surgery for essential tremor.* 2023. 12° Congresso Nazionale AIFM. Physica Medica, 115:s34–s34, 2023. Doi: 10.1016/j.ejmp.2023.102769.

[3] E. Contrino, **M. Romeo**, E. Bruno, M. Marrale, and C. Gagliardo. *Trattamento del tremore mediante ultrasuoni focalizzati guidati da risonanza magnetica: studio preliminare su modelli di deep learning per l'identificazione del target di trattamento.* 2023. 32° National Congress AIRN.

[4] E. Contrino, E. Bruno, G. Collura, **M. Romeo**, S. Mastrilli, M. D'Amelio, M. Marrale, and C. Gagliardo. *Ruolo delle analisi di radiomica nelle malattie*

neurodegenerative: studio preliminare in pazienti affetti da tremore essenziale. 2023. 32° National Congress AIRN.

[5] E. Contrino, E. Bruno, T. Piccoli, G. Maniaci, G. Collura, **M. Romeo**, A. Brancato, C. La Cascia, F. Picone, M. Marrale, C. Cannizzaro, and C. Gagliardo. *Studio di connettivit'a funzionale in soggetti con disturbo da gioco d'azzardo: evidenza di un ruolo per il cervelletto*. 2023. 32° National Congress AIRN.

[6] E. Bruno, E. Contrino, G. Collura, **M. Romeo**, M. D'Amelio, G. Domenico, and C. Gagliardo. *Studio retrospettivo di trattografia probabilistica in pazienti con tremore essenziale sottoposti a talamotomia transcranica mediante ultrasuoni focalizzati rm guidati (mrgfus)*. 2023. 32° National Congress AIRN.

[7] **M. Romeo**, S. Marino, F. Vassallo, E. Bruno, E. Contrino, G. Collura, F. Lizzi, I. Postuma, A. Retico, A. Lascialfari, C. Gagliardo, and M. Marrale. *Deep learning models for tractography predictions in transcranial-mr guided focused ultrasound surgery for essential tremor*. 2023. 14° National Congress AIRMM.

[8] A. Gallo, **M. Romeo**, T. Currieri, C. Militello, E. Bruno, E. Contrino, G. Collura, M. Marrale, S. Vitabile, and C. Gagliardo. *A new osirix/horos plugin combining multiple software packages for target identification in trans-cranial mr-guided focused ultrasound (tcmrgfus) thalamotomies*. 2023. 14° National Congress AIRMM.

[9] A. Spinello, F. Valenza, E. Bruno, E. Contrino, G. Collura, **M. Romeo**, V. Picciolo, G.D. Iacopino, M. D'Amelio, M. Marrale, and C. Gagliardo. *Thalamic parcellation for target identification in trans-cranial mr-guided focused ultrasound (tcmrgfus) thalamotomies: a retrospective probabilistic tractography study in patients with et*. 2023. 14° National Congress AIRMM.

[10] F. Vassallo, S. Marino, E. Bruno, G. Collura, **M. Romeo**, S. Mastrilli, M. D'Amelio, M. Marrale, and C. Gagliardo. *Role of radiomics analysis in neurodegenerative disorders: preliminary study in patients with essential tremor (et)*. 2023. 14° National Congress AIRMM.

5.9.2 Validation and clinical implementation of an AI-driven segmentation tool and design of a quality assurance pipeline

Journal articles

[1] **M. Romeo**, A. Cherit Hernandez, M. Schafasand, C. Reschl, C. Piribauer, M. Fuss Christina, T. Kolb, S. Romdhani, M. Stock, M. Marrale and A. Carlino. *Clinical Implementation of an AI-based segmentation tool including development of a QA pipeline for ion beam therapy*. Manuscript in preparation.

Congress contributions

[1] A. Cherit Hernandez, **M. Romeo**, M. Schafasand, C. Reschl, C. Piribauer, M. Fuss, M. Stock, M. Marrale, and A. Carlino. *Clinical implementation of a quality assurance pipeline for an ai-driven auto-segmentation tool.* 2026. ¨OGBMT ¨OGMP Joint Annual Meeting 2026. URL: `https://hdl.handle.net/10447/704594`.

[2] **M. Romeo**, A. Cherit Hernandez, M. Schafasand, C. Reschl, C. Piribauer, M. Fuss Christina, T. Kolb, S. Romdhani, M. Stock, A. Carlino and M. Marrale. *Clinical implementation of a quality assurance pipeline for an ai-driven auto-segmentation tool.* Physica Medica, 142:92–92, 2026. 13° Congresso Nazionale AIFM. Doi:10.1016/j.ejmp.2025.105440.

[3] J. Giuffrè, M. Schafasand, P. Roisl, **M. Romeo**, M. Marrale, M. Stock, and A. Carlino. *Commissioning of the carbon ion pencil beam algorithm in raystation 2024a.* 2025. 111° Congresso Nazionale SIF. URL: `https://hdl.handle.net/10447/694862`.

5.9.3 Physics Informed Neural Network in Fricke Dosimetry

Journal articles

[1] **M. Romeo**, G. Cottone, S Locarno, D. Passeri, E. Pignoli, I Veronese, C. Lenardi, and M. Marrale. *A physics-informed neural network approach to diffusion and auto-oxidation in fricke gel dosimeters.* Scientific Report, 2026. Submitted.

[2] **M. Romeo**, G. Cottone, S. Locarno, D. Passeri, I. Veronese, C. Lenardi, M. Marrale, et al. *Tackling the problem of diffusion in fricke gel dosimeters through a physics-informed neural network algorithm.* Nuovo Cimento C, 2026. In press.

[3] **M. Romeo**, G. Cottone, M. C. D'Oca, A. Bartolotta, S. Gallo, R. Miraglia, R. Gerasia, G. Milluzzo, F. Romano, C. Gagliardo, M. Marrale et al. *Diffusion correction in fricke hydrogel dosimeters: A deep learning approach with 2d and 3d physics-informed neural network models.* Gels, 10(9):565, 2024. Doi:10.3390/gels10090565.

[4] **M. Romeo**, G Cottone, MC D'Oca, A Bartolotta, R Miraglia, R Gerasia, C Gagliardo, F d'Errico, and M Marrale. *Deep learning approach for diffusion correction in fricke hydrogel dosimeters.* Radiation Measurements, 175:107171, 2024. Doi: 10.1016/j.radmeas.2024.107171.

Congress contributions

[1] **M. Romeo**, G. Cottone, E. Corvaia, E. Maggio, A. De Farias Soares, M. C. D'Oca, S. Locarno, D. Passeri, E. Pignoli, I. Veronese, C. Lenardi, and M.

Marrale. *Overcoming diffusion and auto-oxidation artifacts in fricke dosimetry via physics- informed neural networks*. 2026. Physics for Health: New perspectives in medical applications and radiation safety. URL: https://hdl.handle.net/10447/706877.

[2] **M. Romeo**, G. Cottone, S. Alice Locarno, D. Passeri, I. Veronese, C. Lenardi, and M. Marrale. *Challenges in fricke dosimetry: Physics-informed neural networks for correcting diffusion and auto-oxidation artifacts*. 2026. Advances in X-ray and Gamma RAy DEtectors and applications International Conference. URL: https://hdl.handle.net/10447/702327.

[3] S. Locarno, **M. Romeo**, G. Cottone, A. De Farias Soares, M.C. D'Oca, D. Passeri, E. Pignoli, I. Veronese, C. Lenardi, and M. Marrale. *Ferric ion diffusion in fricke gels: how to cope with it by optimizing gel composition and using ai algorithms for dose reconstruction*. Physica Medica: European Journal of Medical Physics, 142, 2026. 13° Congresso Nazionale AIFM. Doi: 10.1016/j.ejmp.2025.105315.

[4] **M. Romeo**, G. Cottone, M.C. D'Oca, L. Silvia, M. Giuliana, R. Francesco, C. Gagliardo, F. D'Errico, C. Lenardi, and M. Marrale. *Tackling the problem of diffusion in fricke gels dosimeters through a physics informed neural network algorithm*. 2025. 111° Congresso Nazionale SIF. URL: https://hdl.handle.net/10447/690399. **Best oral presentation.**

[5] **M. Romeo**, M.C. D'Oca, F. d'Errico, and M. Marrale. *Deep learning approach for diffusion correction in fricke gel dosimeters*. 2023. 20° International Congress on Solid State Dosimetry.

[6] **M. Romeo**, M.C. D'Oca, F. d'Errico, G. Collura, L. Tranchina, C. Gagliardo, and M. Marrale. *Ai methods for diffusion correction in fricke gel dosimeters*. 2023. 14° National Congress AIRMM.

ACKNOWLEDGMENTS

This thesis was produced while attending the PhD programme in Technologies and Sciences for Human Health at the University of Palermo, Cycle XXXVIII, with the support of a scholarship co-financed by the Ministerial Decree no. 352 of 9th April 2022, based on the NRRP - funded by the European Union - NextGenerationEU - Mission 4 "Education and Research", Component 2 "From Research to Business", Investment 3.3, and by the company UPMC Italy s.r.l. - ISMETT.

I would like to thank the PhD Programme in Technologies and Sciences for Human Health at the University of Palermo, the Doctoral Board, and its Director, Prof. Valeria Vetri. I would also like to thank the Department of Physics and Chemistry "Emilio Segrè" of the University of Palermo, where I pursued my studies in Physics and carried out my research activities.

A special thank you goes to my supervisor, Prof. Maurizio Marrale, a man whose scientific abilities represent only a fraction of his greatness. Kind, thoughtful, honest, fair, and always available, both in the world of research and in life in general. I owe everything to Maurizio. From the moment we met, he saw potential in me that, quite honestly, I was unaware of myself. He has always encouraged me to believe in myself, to give my best, and never to give up. He taught me how to live Science and Research. He introduced me to a prestigious scientific environment. Thanks to him, I have travelled, met experts from the world of Medical Physics, and built an invaluable wealth of experience. Every single achievement I have attained so far would not have been possible without his support. Every time I was ready to throw in the towel; every time I thought a research project would never work; he was there, urging me to keep trying—and in the end, he was always right. In research, results never fall from the sky, and the most important ones are often the most unexpected. Thank you, Maurizio, for allowing me to dream throughout these doctoral years.

Equally important was the support provided by Prof. Grazia Cottone, a brilliant biophysicist at the University of Palermo. From her I learned one of the hardest aspects of research: bringing work to completion. Once a significant result has been achieved, nothing remains unless it is published. For three years, this is what we did together:

putting our results into words in the clearest and most effective way possible. I do not know anyone who can immerse themselves in topics far removed from their own area of expertise and make them their own as she does. She has all my scientific admiration, but also all my affection. These doctoral years would not have been the same without the mornings, afternoons, weekends, and even Ferragosto holidays spent writing papers, abstracts, and other research products. Both of us would gladly have been doing something else, but scientific duty called. From Grazia I learned the value of sacrifice—when made in the right way—for the sake of a passion for Science. Science is not merely a job; it is passion; at times, it is devotion. Yet it always rewards those who pursue it.

During the first years of my PhD, the laboratory where I carried out my research activities was rather quiet and sparsely populated. Back then, one of the best things that could happen to me was meeting Prof. Maria Cristina D'Oca, an expert in EPR measurements. Unfortunately, the amount of research we conducted together was limited. Nevertheless, I would like to take this opportunity to thank her sincerely. She has no idea how much support she gave me, especially during the darkest days, when research was not progressing well and the life of a PhD student became particularly challenging. Such situations become even harder when there is no one to talk to. During those moments, she was always there. I truly owe Cristina a great deal. Her kindness is admirable, and her advice invaluable. I will treasure both forever.

Throughout these years, studying and learning were essential. My academic background and the work carried out during my Master's thesis would not have been sufficient had I not joined the research groups of Commission 5 of the National Institute for Nuclear Physics (INFN). In particular, I would like to thank the AIM_MIA experiment, led by Prof. Alessandra Retico, and the AI_INFN experiment, led by Dr. Lucio Anderlini. All my expertise in Artificial Intelligence was developed thanks to these outstanding projects, true gems within the national scientific landscape. Through the AI Hackathons, I was able not only to expand my knowledge but also to build a network with researchers working in the same field. The greatest satisfaction—and one that I will always carry with me—was progressing from participant to lecturer in the last edition.

Remaining within the INFN community, I feel it is my duty to thank Dr. Francesca Lizzi individually. A brilliant INFN researcher, she taught me the vast majority of what I know about Artificial Intelligence. It is impossible to quantify what I owe to Francesca, but if I had to choose one thing, it would undoubtedly be the rigor required to conduct research properly and consciously. Francesca taught me that results are not everything and that doing research well sometimes means accepting that limits exist. Yet these limits are not obstacles to research; they are the true measure of the value of the results achieved. I could list countless qualities that Francesca possesses, both scientific and personal, but none would fully capture who she is. Therefore, I simply

thank her once again and sincerely hope that the collaborations we have established will continue long after the end of this journey.

Another special thank you goes to Dr. Ian Postuma. He also taught me a great deal about the world of Artificial Intelligence, but to him I owe some of the most memorable scientific missions and experiences of these years. Over time, we have built a friendship that I hope will last for many years to come.

Within the INFN framework, I would also like to thank the Catania Division, with which I was affiliated throughout my doctoral studies. Special thanks go to Dr. Francesco Romano and his colleague Dr. Giuliana Milluzzo.

The research activities concerning VIM segmentation models in patients affected by essential tremor required substantial medical and scientific support. This support was provided by the medical researchers of the University Hospital Policlinico Paolo Giaccone in Palermo. An invaluable contribution was made by Prof. Cesare Gagliardo, a distinguished neuroradiologist of both national and international renown.

With regard to the same research activity, I would also like to thank Dr. Giorgio Collura, who initiated the tractography work that I later used for training the models.

The research activities carried out during my research period abroad, described in Chapter 4, would not have been possible without the support and facilities provided by the MedAustron ion therapy center. Special thanks go to Dr. Antonio Carlino and Prof. Markus Stock, who supervised my work during that period.

During my stay, I collaborated with Dr. Ariadna Cherit Hernandez, whose brilliance, precision, and professionalism deeply impressed me. I was particularly struck by her remarkable ability to visualize and conceptualize complex problems, a truly rare skill. Ariadna was not only a collaborator but also a great friend who made my time in Austria an unforgettable experience. We shared countless moments together, and I sincerely hope that we will continue both our scientific collaborations and our friendship in the years to come.

The work on Fricke gels originated from studies conducted by Prof. Maurizio Marrale in collaboration with Prof. Francesco d'Errico of the University of Pisa. His contribution provided a fundamental starting point that made the subsequent achievements possible.

Equally valuable was the contribution of Prof. Cristina Lenardi, who invited me to conduct the measurements performed at the University of Milan, together with Dr. Daniela Passeri and Dr. Silvia Locarno.

Scientific research is meaningful only when it is shared. For this reason, I would like to thank Alvaro, Elena, Enrico, and Tommaso, wonderful colleagues with whom I shared much of my time in the laboratory. Each of them carries out high-level research activities, and the opportunity to exchange ideas and experiences with them greatly enriched my scientific knowledge and personal growth.

Alvaro, an outstanding Brazilian researcher, has all my scientific admiration. He had to endure the engineer's tragedy of working within a group composed entirely of

physicists. I sincerely believe there is no greater academic challenge than that.

Enrico was the first person I met at the beginning of my university journey, and it is amusing to realize that, after so many years, we still face the problems that come our way exactly as we approached the exercises in our Physics course: with a healthy dose of desperation. When one reaches the end of a path, it is natural to think about those who will carry forward the work that has been started. I am certain that you will be more than capable of doing so.

A special thank you goes to Elena, my colleague and partner. It is often said that work and matters of the heart should never be mixed, but with her, fortune decided that this would be the exception that proves the rule. I owe Elena so much, but above all I am grateful for her affection, which made the final year of my PhD—the most challenging one—far more serene, enjoyable, and ultimately manageable. We share a love and passion for Science, and I can think of nothing more beautiful that one could wish for. During the most difficult moments, we were always there for one another, and this gave us extraordinary strength, leading us both to remarkable achievements. If someone asked me which part of the day I cherish the most, it would undoubtedly be the moments when I start distracting you just to get your attention. Those moments bring us back to a sense of lightness and always restore our smiles. Thank you, Elena. You are the most extraordinary woman I have ever met.

I have always believed that once the laboratory door closes, work should not follow us into the rest of our lives. Over the years, however, I have not always succeeded in living up to this principle, often bringing home stress and fatigue. In those moments, the only people truly capable of helping us overcome the resulting discomfort are our friends. In this regard, Livio, my best friend, has been a true champion. The support he has given me is impossible to quantify, and I will always be grateful for it. Over the years, life led us apart and then brought us together again, yet nothing has changed since the days when we were classmates in the undergraduate Physics programme. Always ready to laugh and joke, even when circumstances would seem to call only for tears. Thank you, Livio. Your presence has been invaluable throughout this journey.

I would also like to thank Chiara. Seeing you change degree programme saddened me at the time because I feared it would distance us. While that may have happened physically, our friendship remained unchanged, and I am truly happy for that. I know that you are now a psychologist, but to me you will always remain, with great affection, my Physics colleague.

True friendships never happen by chance. When they do, we often do not realize their significance at first. As time passes and we look back, their presence in our lives feels so natural that we find ourselves thinking, “It could not have been any other way.” This is certainly true for Vicky and Vivien, two wonderful and cheerful friends whom I met during my stay in Austria. You brought lightness, joy, and a sense of youthfulness into a period of my life when responsibilities and workloads weighed heavily on me. No

professional experience is truly fulfilling without friends with whom to share memorable moments, and you helped make my time in Austria one of the most beautiful adventures of my life. For that, I am deeply grateful.

One may meet the same person thousands of times without anything ever happening, and then a single occasion is enough to give rise to a deep friendship. This is the case with Jasmine. Who would have ever thought that, after that meeting in Austria, we would become such close friends? Life always needs lightness, leisure, and the right amount of madness, and in this regard you have made an immense contribution. Thanks to you, I never truly felt the distance from home. Your smile and your desire to have fun brightened even the darkest days. But I owe you much more than that. At least once in life, each of us is called to look deeply within ourselves, to unravel our most intimate thoughts from the deepest corners of our mind. This process requires great effort, and when done properly, it leads to a clearer awareness of what causes us suffering. Such rediscovery almost always changes us irreversibly and leads us to make choices that radically transform our lives. It is never pleasant, but it is necessary for us to move forward. Although, in an absolute sense, it is the right thing to do, most of the time the benefits of such changes only become apparent with time. You not only helped me look within myself, but you also stood by my side throughout that long period, before I was able to fully feel the benefits of that journey. Some people change our lives; others, in a certain sense, save them. You have been both, and I will always be grateful to you for that.

I would also like to thank someone who probably does not even remember me: Prof. Natale Macaluso of the Technical Industrial Institute in Petralia Soprana. Teaching at that school was undoubtedly challenging, yet Prof. Macaluso devoted himself wholeheartedly to his work, driven solely by the desire to expand his students' knowledge. It was from him, back in 2006, that I learned the fundamentals of programming and how to use Linux systems—skills that today form the very foundation of much of the work I do. Thank you, Prof. Macaluso, for the commitment and dedication you bring to the teaching profession.

A heartfelt thank you goes to my entire family. To my uncle Ignazio, who supported me on many occasions. It was he who helped me get back on track after I had abandoned my studies. Time and again he stood by me, offering the kind of fatherly advice that I could no longer receive from my own father, who passed away too soon.

I would also like to thank my sisters, Elena and Valentina, to whom I owe so much. Above all, they helped me grow into a man who accepts his emotions and is not afraid to cry when necessary; a man who respects women and does not consider his gender a source of superiority.

My thanks also go to my nephews and nieces, Gabriel, Francesco, Laura, Matteo, and Giulia. I hope that what I have accomplished may serve as an example of commitment and perseverance in a world that often suggests that effort is rarely worthwhile.

A very special thank you goes to my mother. When I was a child, you were my superhero. I believed that nothing could stop you. You gave everything for your children despite countless difficulties. Only as I grew older did I truly understand the magnitude of your sacrifices. It is one thing to be a superhero; it is another to be one while living with a disability. It took me many years to realize that the feats you accomplished were not only extraordinary for anyone, but even more remarkable because you achieved them despite being visually impaired. As I grew up, I came to understand that it is thanks to you that I am able to face challenges without fear. You taught me that nothing is truly impossible and that when life calls for a fight, one must face it courageously. Thank you, Mom.

Finally, I would like to thank my grandparents. In particular, my grandfather Matteo, who financed a large part of my education, and my grandmother Maria.

My grandmother was many things to me, far too many to list. Despite never having had the opportunity to pursue an education, she possessed remarkable intelligence. I still remember clearly when you asked me whether, before you died, you would live long enough to see me graduate. You were the only person who believed in me when no one else would. Even at the very end, while I was in Austria, you accidentally called me. I knew you were very ill, but during that final conversation, not wanting to worry me, you reassured me that you were doing well. You said it with such conviction that I believed you. I ended the call peacefully, certain that I would see you again. Instead, only a few hours—perhaps even minutes—later, you were gone.
That was who you were. You cared deeply for all of us and wanted nothing more than our happiness. I am not a religious person, but for you I make the effort to believe that you will be with me on the day I complete this journey. Thank you, Grandma.

BIBLIOGRAPHY

[1] Servaas TH Storm. The ai bubble and the us economy: How long do'hallucinations' last? *Institute for New Economic Thinking Working Paper Series*, 240, 2025.

[2] Jeremy R Burt, Neslisah Torosdagli, Naji Khosravan, Harish RaviPrakash, Aliasghar Mortazi, Fiona Tissavirasingham, Sarfaraz Hussein, and Ulas Bagci. Deep learning beyond cats and dogs: recent advances in diagnosing breast cancer with deep neural networks. *The British journal of radiology*, 91(1089):20170545, 2018.

[3] Muhammad Imran Razzak, Saeeda Naz, and Ahmad Zaib. Deep learning for medical image processing: Overview, challenges and the future. *Classification in BioApps: Automation of decision making*, pages 323–350, 2017.

[4] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42(11):226, 2018.

[5] Fabian Isensee, Paul F Jaeger, Simon AA Kohl, Jens Petersen, and Klaus H Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature methods*, 18(2):203–211, 2021.

[6] Philippe Lambin, Ralph TH Leijenaar, Timo M Deist, Jurgen Peerlings, Evelyn EC De Jong, Janita Van Timmeren, Sebastian Sanduleanu, Ruben THM Larue, Aniek JG Even, Arthur Jochems, et al. Radiomics: the bridge between medical imaging and personalized medicine. *Nature reviews Clinical oncology*, 14(12):749–762, 2017.

[7] Antonino Maniaci, Salvatore Lavalle, Caterina Gagliano, Mario Lentini, Edoardo Masiello, Federica Parisi, Giannicola Iannella, Nicole Dalia Cilia, Valerio Salerno, Giacomo Cusumano, et al. The integration of radiomics and artificial intelligence in modern medicine. *Life*, 14(10):1248, 2024.

[8] Mingqing Wang, Qilin Zhang, Saikit Lam, Jing Cai, and Ruijie Yang. A review on application of deep learning algorithms in external beam radiotherapy automated treatment planning. *Frontiers in oncology*, 10:580919, 2020.

[9] AM Turing. Computing machinery and intelligence. *Mind*, 1950.

[10] Pamela McCorduck and Cli Cfe. *Machines who think: A personal inquiry into the history and prospects of artificial intelligence*. AK Peters/CRC Press, 2004.

[11] Ray Kurzweil. The singularity is near. In *Ethics and emerging technologies*, pages 393–406. Springer, 2005.

[12] Elan D Louis and Joaquim J Ferreira. How common is the most common adult movement disorder? update on the worldwide prevalence of essential tremor. *Movement Disorders*, 25(5):534–541, 2010.

[13] Vijay Chandran and Pramod Kumar Pal. Quality of life and its determinants in essential tremor. *Parkinsonism & related disorders*, 19(1):62–65, 2013.

[14] John-Stuart Brittain, Hayriye Cagnan, Arpan R Mehta, Tabish A Saifee, Mark J Edwards, and Peter Brown. Distinguishing the central drive to tremor in parkinson's disease and essential tremor. *Journal of Neuroscience*, 35(2):795–806, 2015.

[15] Weidong Fang, Huiyue Chen, Hansheng Wang, Han Zhang, Munankami Puneet, Mengqi Liu, Fajin Lv, Tianyou Luo, Oumei Cheng, Xuefeng Wang, et al. Essential tremor is associated with disruption of functional connectivity in the ventral intermediate nucleus—motor cortex—cerebellum circuit. *Human brain mapping*, 37(1):165–178, 2016.

[16] John Gardner. A history of deep brain stimulation: Technological innovation and the role of clinical assessment tools. *Social studies of science*, 43(5):707–728, 2013.

[17] David Moser, Eyal Zadicario, Gilat Schiff, and Daniel Jeanmonod. Mr-guided focused ultrasound technique in functional neurosurgery: targeting accuracy. *Journal of therapeutic ultrasound*, 1:1–10, 2013.

[18] Marc N Gallay, David Moser, and Daniel Jeanmonod. Safety and accuracy of incisionless transcranial mr-guided focused ultrasound functional neurosurgery: single-center experience with 253 targets in 180 treatments. *Journal of Neurosurgery*, 130(4):1234–1243, 2018.

[19] ISO (International Organization for Standardization). Iso/iec 42001: 2023 information technology artificial intelligence—management system. 2023.

[20] Stefan Strohmeier. Artificial intelligence in human resources-an introduction. In *Handbook of research on artificial intelligence in human resource management*, pages 1–4. Edward Elgar Publishing, 2022.

[21] John McCarthy, Marvin L Minsky, Nathaniel Rochester, and Claude E Shannon. A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955. *AI magazine*, 27(4):12–12, 2006.

[22] Luigi Federico Menabrea and Ada Lovelace. Sketch of the analytical engine invented by charles babbage. *Sci Mem*, 3:666–731, 1843.

[23] Rowan Hooper. Ada lovelace: My brain is more than merely mortal. *New Scientist*, 216(2886):29, 2012.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[25] Jacob Aron. Forget the turing test–there are better ways of judging ai. *New Scientist. Retrieved from https://www. newscientist. com*, 2015.

[26] Dae-Young Kang, Hieu Pham Duong, and Jung-Chul Park. Application of deep learning in dentistry and implantology. *Journal of implantology and applied sciences*, 24(3):148–181, 2020.

[27] Francois Chollet and François Chollet. *Deep learning with Python*. simon and schuster, 2021.

[28] Machine Learning. Tom mitchell. *Publisher: McGraw Hill*, page 31, 1997.

[29] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[30] Adam Coates, Brody Huval, Tao Wang, David Wu, Bryan Catanzaro, and Ng Andrew. Deep learning with cots hpc systems. In *International conference on machine learning*, pages 1337–1345. PMLR, 2013.

[31] Marc'aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg Corrado, Jeff Dean, Quoc V Le, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 81–88, 2012.

[32] Julien Guiot, Akshayaa Vaidyanathan, Louis Deprez, Fadila Zerka, Denis Danthine, Anne-Noelle Frix, Philippe Lambin, Fabio Bottari, Nathan Tsoutzidis, Benjamin Miraglio, et al. A review in radiomics: making personalized medicine a reality via routine imaging. *Medicinal research reviews*, 42(1):426–440, 2022.

[33] A Saranya and R Subhashini. A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends. *Decision analytics journal*, 7:100230, 2023.

[34] Alun Preece. Asking 'why'in ai: Explainability of intelligent systems–perspectives and challenges. *Intelligent Systems in Accounting, Finance and Management*, 25(2):63–72, 2018.

[35] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[36] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

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

[38] Ronan Collobert, Koray Kavukcuoglu, Clément Farabet, et al. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS workshop*, volume 5, page 10. Lake Tahoe, NV, 2011.

[39] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.

[40] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

[41] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[42] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867, 1993.

[43] Ding-Xuan Zhou. Universality of deep convolutional neural networks. *Applied and computational harmonic analysis*, 48(2):787–794, 2020.

[44] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[45] Patrick J Grother. Nist special database 19. *Handprinted forms and characters database, National Institute of Standards and Technology*, 10:69, 1995.

[46] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[47] J Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[51] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[52] Wei Wang, Jinge Tian, Chengwen Zhang, Yanhong Luo, Xin Wang, and Ji Li. An improved deep learning approach and its applications on colonic polyp images detection. *BMC Medical Imaging*, 20(1):83, 2020.

[53] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296), 1979.

[54] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[55] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[56] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[57] Juan Terven, Diana-Margarita Córdova-Esparza, and Julio-Alejandro Romero-González. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5(4):1680–1716, 2023.

[58] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[59] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[60] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010.

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

[62] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[63] Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of computational physics*, 91(1):110–131, 1990.

[64] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.

[65] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[66] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[67] David C Van Essen, Kamil Ugurbil, Edward Auerbach, Deanna Barch, Timothy EJ Behrens, Richard Bucholz, Acer Chang, Liyong Chen, Maurizio Corbetta, Sandra W Curtiss, et al. The human connectome project: a data acquisition perspective. *Neuroimage*, 62(4):2222–2231, 2012.

[68] David C Van Essen, Stephen M Smith, Deanna M Barch, Timothy EJ Behrens, Essa Yacoub, Kamil Ugurbil, Wu-Minn HCP Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.

[69] John G Lynn and Tracy J Putnam. Histology of cerebral lesions produced by focused ultrasound. *The American journal of pathology*, 20(3):637, 1944.

[70] Giuseppe Roberto Giammalva, Cesare Gagliardo, Salvatore Marrone, Federica Paolini, Rosa Maria Gerardi, Giuseppe Emmanuele Umana, Kaan Yağmurlu, Bipin Chaurasia, Gianluca Scalia, Federico Midiri, et al. Focused ultrasound in neuroscience. state of the art and future perspectives. *Brain sciences*, 11(1):84, 2021.

[71] Domenico Gerardo Iacopino, Cesare Gagliardo, Antonella Giugno, Giuseppe Roberto Giammalva, Alessandro Napoli, Rosario Maugeri, Francesca Graziano, Francesca Valentino, Giuseppe Cosentino, Marco D'Amelio, et al. Preliminary experience with a transcranial magnetic resonance–guided focused ultrasound surgery system integrated with a 1.5-t mri unit in a series of patients with essential tremor and parkinson's disease. *Neurosurgical Focus*, 44(2):E7, 2018.

[72] Syed A Quadri, Muhammad Waqas, Inamullah Khan, Muhammad Adnan Khan, Sajid S Suriya, Mudassir Farooqui, and Brian Fiani. High-intensity focused ultrasound: past, present, and future in neurosurgery. *Neurosurgical focus*, 44(2):E16, 2018.

[73] Dietrich Haubenberger and Mark Hallett. Essential tremor. *New England Journal of Medicine*, 378(19):1802–1810, 2018. PMID: 29742376.

[74] Francesco Sammartino, Vibhor Krishna, Nicolas Kon Kam King, Andres M Lozano, Michael L Schwartz, Yuexi Huang, and Mojgan Hodaie. Tractography-based ventral intermediate nucleus targeting: Novel methodology and intraoperative validation. *Movement Disorders*, 31(8):1217–1225, 2016.

[75] Juergen K. Mai and F. Forutan. Chapter 19 - thalamus. In Juergen K. Mai and George Paxinos, editors, *The Human Nervous System*, pages 618–677. Academic Press, San Diego, 3rd edition, 2012.

[76] Francisca Ferreira, Harith Akram, John Ashburner, Ludvic Zrinzo, Hui Zhang, and Christian Lambert. Ventralis intermedius nucleus anatomical variability assessment by mri structural connectivity. *Neuroimage*, 238:118231, 2021.

[77] Timothy R Miller, Jiachen Zhuo, Howard M Eisenberg, Paul S Fishman, Elias R Melhem, Rao Gullapalli, and Dheeraj Gandhi. Targeting of the dentato-rubro-thalamic tract for mr-guided focused ultrasound treatment of essential tremor. *The neuroradiology journal*, 32(6):401–407, 2019.

[78] Ariane Carpentier, Jason Chan, Joshua Wong, and Michael Okun. An eye on the first surgical side: Appreciating the potential impacts of a second dbs lead on ipsilateral symptoms. *Tremor and Other Hyperkinetic Movements*, 14, 07 2024.

[79] Jamir Pitton Rissardo, Nilofar Murtaza Vora, Irra Tariq, Amna Mujtaba, and Ana Letícia Fornari Caprara. Deep brain stimulation for the management of refractory neurological disorders: A comprehensive review. *Medicina*, 59(11):1991, 2023.

[80] Sergiu Groppa, Gabriel Gonzalez-Escamilla, Gerd Tinkhauser, Halim Ibrahim Baqapuri, Bastian Sajonz, Christoph Wiest, Joana Pereira, Damian M Herz, Matthias R Dold, Manuel Bange, et al. Perspectives of implementation of closed-loop deep brain stimulation: From neurological to psychiatric disorders. *Stereotactic and functional neurosurgery*, pages 1–15, 2023.

[81] James Peters, Joel Maamary, Kain Kyle, Nick Olsen, Lyndsey Jones, Samuel Bolitho, Yael Barnett, Benjamin Jonker, and Stephen Tisch. Outcomes of focused ultrasound thalamotomy in tremor syndromes. *Movement Disorders*, 2023.

[82] O Parras, P Domínguez, A Tomás-Biosca, and J Guridi. The role of tractography in the localisation of the vim nucleus of the thalamus and the dentatorubrothalamic tract for the treatment of tremor. *Neurología (English Edition)*, 37(8):691–699, 2022.

[83] Jason H Su, Francis T Thomas, Willard S Kasoff, Thomas Tourdias, Eun Young Choi, Brian K Rutt, and Manojkumar Saranathan. Thalamus optimized multi atlas segmentation (thomas): fast, fully automated segmentation of thalamic nuclei from structural mri. *Neuroimage*, 194:272–282, 2019.

[84] Mattia Romeo, Cesare Gagliardo, Grazia Cottone, Giorgio Collura, Enrico Maggio, Claudio Runfola, Eleonora Bruno, Maria Cristina D'Oca, Massimo Midiri, Francesca Lizzi, Ian Postuma, Marco D'Amelio, Alessandro Lascialfari, Alessandra Retico, and Maurizio Marrale. Delta-bit: an open-source probabilistic tractography-based deep learning framework for thalamic targeting in functional neurological disorders, 2026.

[85] Timothy EJ Behrens, H Johansen Berg, Saad Jbabdi, Matthew FS Rushworth, and Mark W Woolrich. Probabilistic diffusion tractography with multiple fibre orientations: What can we gain? *Neuroimage*, 34(1):144–155, 2007.

[86] Bruce Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.

[87] Stephen M Smith, Mark Jenkinson, Mark W Woolrich, Christian F Beckmann, Timothy EJ Behrens, Heidi Johansen-Berg, Peter R Bannister, Marilena De Luca,

Ivana Drobnjak, David E Flitney, et al. Advances in functional and structural mr image analysis and implementation as fsl. *Neuroimage*, 23:S208–S219, 2004.

[88] Mark W Woolrich, Saad Jbabdi, Brian Patenaude, Michael Chappell, Salima Makni, Timothy Behrens, Christian Beckmann, Mark Jenkinson, and Stephen M Smith. Bayesian analysis of neuroimaging data in fsl. *Neuroimage*, 45(1):S173–S186, 2009.

[89] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

[90] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[91] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[92] Timothy EJ Behrens, H Johansen-Berg, MW Woolrich, SM Smith, CAM Wheeler-Kingshott, PA Boulby, GJ Barker, EL Sillery, K Sheehan, O Ciccarelli, et al. Non-invasive mapping of connections between human thalamus and cortex using diffusion imaging. *Nature Neuroscience*, 6(7):750–757, 2003.

[93] Moises Hernandez-Fernandez, Istvan Reguly, Saad Jbabdi, Mike Giles, Stephen Smith, and Stamatios N Sotiropoulos. Using gpus to accelerate computational diffusion mri: From microstructure estimation to tractography and connectomes. *Neuroimage*, 188:598–615, 2019.

[94] Evangelia Tsolaki, Angela Downes, William Speier, W Jeff Elias, and Nader Pouratian. The potential value of probabilistic tractography-based for mr-guided focused ultrasound thalamotomy for essential tremor. *NeuroImage: Clinical*, 17:1019–1027, 2018.

[95] Fabricio S Feltrin, Rajiv Chopra, Nader Pouratian, Mazen Elkurd, Rasheda El-Nazer, Lauren Lanford, William Dauer, and Bhavya R Shah. Focused ultrasound using a novel targeting method four-tract tractography for magnetic resonance–guided high-intensity focused ultrasound targeting. *Brain Communications*, 4(6):fcac273, 2022.

[96] Qiyuan Tian, Max Wintermark, W Jeffrey Elias, Pejman Ghanouni, Casey H Halpern, Jaimie M Henderson, Diane S Huss, Maged Goubran, Christian Thaler, Raag Airan, et al. Diffusion mri tractography for improved transcranial mri-guided focused ultrasound thalamotomy targeting for essential tremor. *NeuroImage: Clinical*, 19:572–580, 2018.

[97] Jonathan A Hyam, Sarah LF Owen, Morten L Kringelbach, Ned Jenkinson, John F Stein, Alexander L Green, and Tipu Z Aziz. Contrasting connectivity of the ventralis intermedius and ventralis oralis posterior nuclei of the motor thalamus demonstrated by probabilistic tractography. *Neurosurgery*, 70(1):162–169, 2012.

[98] Igor A Ilinsky and Kristy Kultas-Ilinsky. Motor thalamic circuits in primates with emphasis on the area targeted in treatment of movement disorders. *Movement disorders: official journal of the Movement Disorder Society*, 17(S3):S9–S14, 2002.

[99] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[100] Peter F Neher, Michael Götz, Tobias Norajitra, Christian Weber, and Klaus H Maier-Hein. A machine learning based approach to fiber tractography using classifier voting. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part I 18*, pages 45–52. Springer, 2015.

[101] Philippe Poulin, Marc-Alexandre Côté, Jean-Christophe Houde, Laurent Petit, Peter F Neher, Klaus H Maier-Hein, Hugo Larochelle, and Maxime Descoteaux. Learn to track: deep learning for tractography. In *Medical Image Computing and Computer Assisted Intervention- MICCAI 2017: 20th International Conference, Quebec City, QC, Canada, September 11-13, 2017, Proceedings, Part I 20*, pages 540–547. Springer, 2017.

[102] Viktor Wegmayr, Giacomo Giuliari, Stefan Holdener, and Joachim Buhmann. Data-driven fiber tractography with neural networks. In *2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018)*, pages 1030–1033. IEEE, 2018.

[103] Simon Koppers, Matthias Friedrichs, and Dorit Merhof. Reconstruction of diffusion anisotropies using 3d deep convolutional neural networks in diffusion imaging. In *Modeling, analysis, and visualization of anisotropy*, pages 393–404. Springer, 2017.

[104] Simon Koppers and Dorit Merhof. Direct estimation of fiber orientations using deep learning in diffusion imaging. In *Machine Learning in Medical Imaging: 7th International Workshop, MLMI 2016, Held in Conjunction with MICCAI 2016, Athens, Greece, October 17, 2016, Proceedings 7*, pages 53–60. Springer, 2016.

[105] Itay Benou and Tammy Riklin Raviv. Deeptract: A probabilistic deep learning framework for white matter fiber tractography. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part III 22*, pages 626–635. Springer, 2019.

[106] Jakob Wasserthal, Peter Neher, and Klaus H Maier-Hein. Tractseg-fast and accurate white matter tract segmentation. *NeuroImage*, 183:239–253, 2018.

[107] Jakob Wasserthal, Peter F Neher, and Klaus H Maier-Hein. Tract orientation mapping for bundle-specific tractography. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part III 11*, pages 36–44. Springer, 2018.

[108] Antoine Théberge, Christian Desrosiers, Maxime Descoteaux, and Pierre-Marc Jodoin. Track-to-learn: A general framework for tractography with deep reinforcement learning. *Medical Image Analysis*, 72:102093, 2021.

[109] Marco Reisert, Volker A Coenen, Christoph Kaller, Karl Egger, and Henrik Skibbe. Hamlet: hierarchical harmonic filters for learning tracts from diffusion mri. *arXiv preprint arXiv:1807.01068*, 2018.

[110] Viktor Wegmayr and Joachim M Buhmann. Entrack: Probabilistic spherical regression with entropy regularization for fiber tractography. *International Journal of Computer Vision*, 129:656–680, 2021.

[111] SMH Hosseini, M Hassanpour, S Masoudnia, S Iraji, S Raminfard, and M Nazem-Zadeh. Cttrack: A cnn+ transformer-based framework for fiber orientation estimation & tractography. *Neuroscience Informatics*, 2(4):100099, 2022.

[112] Stamatios N Sotiropoulos, Steen Moeller, Saad Jbabdi, Jungqian Xu, JL Andersson, Edward John Auerbach, Essa Yacoub, D Feinberg, Kawin Setsompop, Lawrence L Wald, et al. Effects of image reconstruction on fiber orientation mapping from multichannel diffusion mri: reducing the noise floor using sense. *Magnetic resonance in medicine*, 70(6):1682–1689, 2013.

[113] Mikhail Milchenko and Daniel Marcus. Obscuring surface anatomy in volumetric imaging data. *Neuroinformatics*, 11:65–75, 2013.

[114] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[115] François Chollet et al. Keras. `https://keras.io`, 2015.

[116] Elan D Louis and Phyllis L Faust. Essential tremor pathology: neurodegeneration and reorganization of neuronal connections. *Nature Reviews Neurology*, 16(2):69–83, 2020.

[117] Dietrich Haubenberger and Mark Hallett. Essential tremor. *New England Journal of Medicine*, 378(19):1802–1810, 2018.

[118] Abhijit Guha Roy, Sailesh Conjeti, Nassir Navab, Christian Wachinger, Alzheimer's Disease Neuroimaging Initiative, et al. Quicknat: A fully convolutional network for quick and accurate segmentation of neuroanatomy. *NeuroImage*, 186:713–727, 2019.

[119] Gustavo Retuci Pinheiro, Lorenza Brusini, Diedre Carmo, Renata Prôa, Thays Abreu, Simone Appenzeller, Gloria Menegaz, and Leticia Rittner. Thalamus segmentation using deep learning with diffusion mri data: An open benchmark. *Applied Sciences*, 13(9):5284, 2023.

[120] Jonathan D Clayden, David L Thomas, and Alexander Kraskov. Tractography-based parcellation does not provide strong evidence of anatomical organisation within the thalamus. *Neuroimage*, 199:418–426, 2019.

[121] Harith Akram, Viswas Dayal, Philipp Mahlknecht, Dejan Georgiev, Jonathan Hyam, Thomas Foltynie, Patricia Limousin, Enrico De Vita, Marjan Jahanshahi, John Ashburner, et al. Connectivity derived thalamic segmentation in deep brain stimulation for tremor. *NeuroImage: Clinical*, 18:130–142, 2018.

[122] Marc N Gallay, Daniel Jeanmonod, Jian Liu, and Anne Morel. Human pallidothalamic and cerebellothalamic tracts: anatomical basis for functional stereotactic neurosurgery. *Brain Structure and Function*, 212:443–463, 2008.

[123] J Mollink, KM Van Baarsen, PJWC Dederen, S Foxley, KL Miller, S Jbabdi, CH Slump, JA Grotenhuis, M Kleinnijenhuis, and AM Van Cappellen Van Walsum. Dentatorubrothalamic tract localization with postmortem mr diffusion tractography compared to histological 3d reconstruction. *Brain structure and function*, 221:3487–3501, 2016.

[124] Evan Calabrese, Patrick Hickey, Christine Hulette, Jingxian Zhang, Beth Parente, Shivanand P Lad, and G Allan Johnson. Postmortem diffusion mri of the human

brainstem and thalamus for deep brain stimulator electrode localization. *Human brain mapping*, 36(8):3167–3178, 2015.

[125] Andrew I Yang, Drew Parker, Anupa A Vijayakumari, Ashwin G Ramayya, Melanie P Donley-Fletcher, Darien Aunapu, Ronald L Wolf, Gordon H Baltuch, and Ragini Verma. Tractography-based surgical targeting for thalamic deep brain stimulation: a comparison of probabilistic vs deterministic fiber tracking of the dentato-rubro-thalamic tract. *Neurosurgery*, 90(4):419–425, 2022.

[126] Jörn Diedrichsen. A spatially unbiased atlas template of the human cerebellum. *Neuroimage*, 33(1):127–138, 2006.

[127] Jörn Diedrichsen, Joshua H Balsters, Jonathan Flavell, Emma Cussans, and Narender Ramnani. A probabilistic mr atlas of the human cerebellum. *Neuroimage*, 46(1):39–46, 2009.

[128] Jörn Diedrichsen, Stefan Maderwald, Michael Küper, Markus Thürling, K Rabe, ER Gizewski, Mark E Ladd, and Dagmar Timmann. Imaging the deep cerebellar nuclei: a probabilistic atlas and normalization procedure. *Neuroimage*, 54(3):1786–1794, 2011.

[129] Jörn Diedrichsen and Ewa Zotow. Surface-based display of volume-averaged cerebellar imaging data. *PloS one*, 10(7):e0133402, 2015.

[130] M Wintermark, J Druzgal, DS Huss, MA Khaled, S Monteith, P Raghavan, T Huerta, LC Schweickert, B Burkholder, JJ Loomba, et al. Imaging findings in mr imaging–guided focused ultrasound treatment for patients with essential tremor. *American journal of Neuroradiology*, 35(5):891–896, 2014.

[131] Cesare Gagliardo, Roberto Cannella, Cettina Quarrella, Marco D'Amelio, Alessandro Napoli, Tommaso Vincenzo Bartolotta, Carlo Catalano, Massimo Midiri, and Roberto Lagalla. Intraoperative imaging findings in transcranial mr imaging-guided focused ultrasound treatment at 1.5 t may accurately detect typical lesional findings correlated with sonication parameters. *European Radiology*, 30:5059–5070, 2020.

[132] Paul A Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C Gee, and Guido Gerig. User-guided 3d active contour segmentation of anatomical structures: significantly improved efficiency and reliability. *Neuroimage*, 31(3):1116–1128, 2006.

[133] Hee E Kim, Alejandro Cosa-Linan, Nandhini Santhanam, Mahboubeh Jannesari, Mate E Maros, and Thomas Ganslandt. Transfer learning for medical image classification: a literature review. *BMC medical imaging*, 22(1):69, 2022.

[134] Davood Karimi, Simon K Warfield, and Ali Gholipour. Transfer learning in medical image segmentation: New insights from analysis of the dynamics of model parameters and learned representations. *Artificial intelligence in medicine*, 116:102078, 2021.

[135] Ziyan Huang, Haoyu Wang, Zhongying Deng, Jin Ye, Yanzhou Su, Hui Sun, Junjun He, Yun Gu, Lixu Gu, Shaoting Zhang, et al. Stu-net: Scalable and transferable medical image segmentation models empowered by large-scale supervised pre-training. *arXiv preprint arXiv:2304.06716*, 2023.

[136] News of science. *Science*, 125(3236):18–22, 1957.

[137] Hans Bethe. Zur theorie des durchgangs schneller korpuskularstrahlen durch materie. *Annalen der Physik*, 397(3):325–400, 1930.

[138] Felix Bloch. Bremsvermögen von atomen mit mehreren elektronen. *Zeitschrift für Physik*, 81(5):363–376, 1933.

[139] Thomas B Brunner. The rationale of combined radiotherapy and chemotherapy–joint action of castor and pollux. *Best Practice & Research Clinical Gastroenterology*, 30(4):515–528, 2016.

[140] Christophe Hennequin, Sophie Guillerm, and Laurent Quero. Combination of chemotherapy and radiotherapy: A thirty years evolution. *Cancer/Radiothérapie*, 23(6-7):662–665, 2019.

[141] Vincent T DeVita, Theodore S Lawrence, and Steven A Rosenberg. *DeVita, Hellman, and Rosenberg's cancer: principles & practice of oncology*, volume 2. Lippincott Williams & Wilkins, 2008.

[142] Geoff Delaney, Susannah Jacob, Carolyn Featherstone, and Michael Barton. The role of radiotherapy in cancer treatment: estimating optimal utilization from a review of evidence-based clinical guidelines. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 104(6):1129–1137, 2005.

[143] Laura Vanderwaeren, Rüveyda Dok, Kevin Verstrepen, and Sandra Nuyts. Clinical progress in proton radiotherapy: biological unknowns. *Cancers*, 13(4):604, 2021.

[144] Elizabeth Escobar-Peralta, Rebeca Gil-García, Abril A Quezada-Bautista, and Otoniel López-Riverol. Helical tomotherapy: advanced radiotherapy technology. *Revista médica del Hospital General de México*, 85(4):179–184, 2022.

[145] Linda Chen, Patricia Platzer, Christian Reschl, Mansure Schafasand, Ankita Nachankar, Christoph Lukas Hajdusich, Peter Kuess, Markus Stock, Steven Habraken, and Antonio Carlino. Validation of a deep-learning segmentation

model for adult and pediatric head and neck radiotherapy in different patient positions. *Physics and Imaging in Radiation Oncology*, 29:100527, 2024.

[146] Markus Stock, Petra Georg, Ramona Mayer, Till T Böhlen, and Stanislav Vatnitsky. Development of clinical programs for carbon ion beam therapy at medaustron. *International journal of particle therapy*, 2(3):474–477, 2015.

[147] Camilla Panduro Nielsen, Ebbe Laugaard Lorenzen, Kenneth Jensen, Nis Sarup, Carsten Brink, Bob Smulders, Anne Ivalu Sander Holm, Eva Samsøe, Martin Skovmos Nielsen, Patrik Sibolt, et al. Consistency in contouring of organs at risk by artificial intelligence vs oncologists in head and neck cancer patients. *Acta Oncologica*, 62(11):1418–1425, 2023.

[148] Michael V Sherer, Diana Lin, Sharif Elguindi, Simon Duke, Li-Tee Tan, Jon Cacicedo, Max Dahele, and Erin F Gillespie. Metrics to evaluate the performance of auto-segmentation for radiation treatment planning: A critical review. *Radiotherapy and Oncology*, 160:185–191, 2021.

[149] Lisa A Kachnic, Kathryn Winter, Robert J Myerson, Michael D Goodyear, John Willins, Jacqueline Esthappan, Michael G Haddock, Marvin Rotman, Parag J Parikh, Howard Safran, et al. Rtog 0529: a phase 2 evaluation of dose-painted intensity modulated radiation therapy in combination with 5-fluorouracil and mitomycin-c for the reduction of acute morbidity in carcinoma of the anal canal. *International Journal of Radiation Oncology* Biology* Physics*, 86(1):27–33, 2013.

[150] Barbara Segedin and Primoz Petric. Uncertainties in target volume delineation in radiotherapy–are they relevant and what can we do about them? *Radiology and oncology*, 50(3):254, 2016.

[151] I Simmat, P Georg, D Georg, W Birkfellner, G Goldner, and M Stock. Prüfung von genauigkeit und effizienz atlasbasierter konturierung zur strahlentherapie des prostatakarzinoms bei verschiedenen klinischen gegebenheiten, 2012.

[152] Fatemeh Nazem, Fahimeh Ghasemi, Afshin Fassihi, and Alireza Mehri Dehnavi. 3d u-net: A voxel-based method in binding site prediction of protein structure. *Journal of Bioinformatics and Computational Biology*, 19(02):2150006, 2021.

[153] Mohammad Shehab, Laith Abualigah, Qusai Shambour, Muhannad A Abu-Hashem, Mohd Khaled Yousef Shambour, Ahmed Izzat Alsalibi, and Amir H Gandomi. Machine learning in medical applications: A review of state-of-the-art methods. *Computers in Biology and Medicine*, 145:105458, 2022.

[154] Oleg E Karpov, Elena N Pitsik, Semen A Kurkin, Vladimir A Maksimenko, Alexander V Gusev, Natali N Shusharina, and Alexander E Hramov. Analysis of publication activity and research trends in the field of ai medical applications:

Network approach. *International Journal of Environmental Research and Public Health*, 20(7):5335, 2023.

[155] L Simon, C Robert, and Philippe Meyer. Artificial intelligence for quality assurance in radiotherapy. *Cancer/Radiothérapie*, 25(6-7):623–626, 2021.

[156] Guangqi Li, Xin Wu, and Xuelei Ma. Artificial intelligence in radiotherapy. In *Seminars in Cancer Biology*, volume 86, pages 160–171. Elsevier, 2022.

[157] Tomohiro Ono, Hiraku Iramina, Hideaki Hirashima, Takanori Adachi, Mitsuhiro Nakamura, and Takashi Mizowaki. Applications of artificial intelligence for machine-and patient-specific quality assurance in radiation therapy: current status and future directions. *Journal of Radiation Research*, 65(4):421–432, 2024.

[158] Maria F Chan, Alon Witztum, and Gilmer Valdes. Integration of ai and machine learning in radiotherapy qa. *Frontiers in artificial intelligence*, 3:577620, 2020.

[159] Malvika Pillai, Karthik Adapa, Shiva K Das, Lukasz Mazur, John Dooley, Lawrence B Marks, Reid F Thompson, and Bhishamjit S Chera. Using artificial intelligence to improve the quality and safety of radiation therapy. *Journal of the American College of Radiology*, 16(9):1267–1272, 2019.

[160] Liesbeth Vandewinckele, Michaël Claessens, Anna Dinkla, Charlotte Brouwer, Wouter Crijns, Dirk Verellen, and Wouter van Elmpt. Overview of artificial intelligence-based applications in radiotherapy: Recommendations for implementation and quality assurance. *Radiotherapy and Oncology*, 153:55–66, 2020.

[161] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[162] Thorvald Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biologiske skrifter*, 5:1–34, 1948.

[163] RE Drzymala, Radhe Mohan, L Brewster, J Chu, Michael Goitein, W Harms, and M Urie. Dose-volume histograms. *International Journal of Radiation Oncology* Biology* Physics*, 21(1):71–78, 1991.

[164] Philip Mayles, Alan Nahum, and Jean-Claude Rosenwald. *Handbook of radiotherapy physics: theory and practice*. CRC Press, 2007.

[165] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Fundamental algorithms for scientific computing in python. *Nat. Methods*, 17:261–272, 2020.

[166] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science and engineering*, 9:90–95, 2007.

[167] Wes McKinney et al. Data structures for statistical computing in python. *scipy*, 445:51–56, 2010.

[168] Thomas KLUYVER[a1], Benjamin RAGAN-KELLEYb[1], Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, and Kyle Kelley. Jupyter notebooks-a publishing format for reproducible computational workflows. In *Positioning and power in academic publishing: players, agents and agendas: proceedings of the 20th International Conference on Electronic Publishing*, page 87. IOS press, 2016.

[169] Glen Knoll. *Radiation Detection and Measurement (4th ed.)*. John Wiley, Hoboken, NJ, 2010.

[170] S. W. S. McKeever. *Thermoluminescence of Solids*. Cambridge Solid State Science Series. Cambridge University Press, 1985.

[171] E.B. Podgorsak, Vienna (Austria) International Atomic Energy Agency, Canadian Organization of Medical Physicists/Canadian College of Physicists in Medicine (Canada), York (United Kingdom) European Federation of Organisations for Medical Physics, European Society for Therapeutic Radiology, Leuven (Belgium) Oncology, TX (United States) International Organization for Medical Physics, San Antonio, DC (United States) Pan American Health Organization, Washington, and Geneva (Switzerland) World Health Organization. *Radiation oncology physics: A handbook for teachers and students*. IAEA, January 2025.

[172] Ann Van Esch, Christian Clermont, Magali Devillers, Mauro Iori, and Dominique P Huyskens. On-line quality assurance of rotational radiotherapy treatment delivery by means of a 2d ion chamber array and the octavius phantom. *Medical physics*, 34(10):3825–3837, 2007.

[173] Sara Olsson, Eva S Bergstrand, EO Hole, and E Lund. Radiation dose measurements with alanine/agarose gel and thin alanine films around a 192ir brachytherapy source, using esr spectroscopy. *Physics in Medicine & Biology*, 47(8):1333–1356, 2002.

[174] Bjoern Poppe, Arne Blechschmidt, Armand Djouguela, Ralf Kollhoff, Antje Rubach, Kay C Willborn, and Dietrich Harder. Two-dimensional ionization chamber arrays for imrt plan verification. *Medical physics*, 33(4):1005–1015, 2006.

[175] Hugo Fricke and Sterne Morse. The chemical action of roentgen rays on dilute ferrosulphate solutions as a measure of dose. *Am. J. Roentgenol. Radium Ther*, 18:430–432, 1927.

[176] JC Gore and YS Kang. Measurement of radiation dose distributions by nuclear magnetic resonance (nmr) imaging. *Physics in Medicine & Biology*, 29(10):1189, 1984.

[177] JC Gore, M Ranade, MJ Maryanski, and RJ Schulz. Radiation dose distributions in three dimensions from tomographic optical density scanning of polymer gels: I. development of an optical scanner. *Physics in Medicine & Biology*, 41(12):2695–2704, 1996.

[178] Maurizio Marrale and Francesco d'Errico. Hydrogels for three-dimensional ionizing-radiation dosimetry. *Gels*, 7(2):74, 2021.

[179] Clive Baldock, Yves De Deene, Simon Doran, Geoffrey Ibbott, Asam Jirasek, Moder Lepage, KB McAuley, Michael Oldham, and LJ3031873 Schreiner. Polymer gel dosimetry. *Physics in Medicine & Biology*, 55(5):R1–R63, 2010.

[180] M Romeo, G Cottone, MC D'Oca, A Bartolotta, R Miraglia, R Gerasia, C Gagliardo, F d'Errico, and M Marrale. Deep learning approach for diffusion correction in fricke hydrogel dosimeters. *Radiation Measurements*, 175:107171, 2024.

[181] Mattia Romeo, Grazia Cottone, Maria Cristina D'Oca, Antonio Bartolotta, Salvatore Gallo, Roberto Miraglia, Roberta Gerasia, Giuliana Milluzzo, Francesco Romano, Cesare Gagliardo, et al. Diffusion correction in fricke hydrogel dosimeters: A deep learning approach with 2d and 3d physics-informed neural network models. *Gels*, 10(9):565, 2024.

[182] William ID Rae, Casper A Willemse, Mattheus G Lötter, Johan S Engelbrecht, and Jannie C Swarts. Chelator effect on ion diffusion in ferrous-sulfate-doped gelatin gel dosimeters as analyzed by mri. *Medical physics*, 23(1):15–23, 1996.

[183] LJ Schreiner. True 3d chemical dosimetry (gels, plastics): Development and clinical role. In *Journal of Physics: Conference Series*, volume 573, page 012003, 2015.

[184] RG Kelly, KJ Jordan, and JJ Battista. Optical ct reconstruction of 3d dose distributions using the ferrous–benzoic–xylenol (fbx) gel dosimeter. *Medical physics*, 25(9):1741–1750, 1998.

[185] Yves De Deene and Andrew Jirasek. Gel dosimetry: An overview of dosimetry systems and read out methods. *Radiation Measurements*, 179:107321, 2024.

[186] Martina Scotti, Paolo Arosio, Elisa Brambilla, Salvatore Gallo, Cristina Lenardi, Silvia Locarno, Francesco Orsini, Emanuele Pignoli, Luca Pedicone, and Ivan Veronese. How xylenol orange and ferrous ammonium sulphate influence the dosimetric properties of pva–gta fricke gel dosimeters: A spectrophotometric study. *Gels*, 8(4):204, 2022.

[187] Tomas Kron, David Jonas, and James M Pope. Fast t1 imaging of dual gel samples for diffusion measurements in nmr dosimetry gels. *Magnetic resonance imaging*, 15(2):211–221, 1997.

[188] José Vedelago, David Chacón, Marcelo Romero, Daniel Venencia, Facundo Mattea, and Mauro Valente. Dose-response of fricke-and pagat-dosimetry gels in kilovoltage and megavoltage photon beams: Impact of let on sensitivity. *Physica Medica*, 84:41–49, 2021.

[189] Wei Zhang, Kaikai Wang, Yufeng Zeng, Xiaodan Hu, Xiaohong Zhang, Shuquan Chang, and Haiqian Zhang. Low-diffusion fricke gel dosimeters with core-shell structure based on spatial confinement. *Materials*, 14(14):3932, 2021.

[190] YJ Tseng, Sung-Cheng Huang, and WC Chu. An image-based approach to measure the ferric ion diffusion coefficient in fricke gel dosimetry. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 7393–7396. IEEE, 2006.

[191] Lucas Nonato de Oliveira, Adelaide de Almeida, and Linda VE Caldas. Fricke gel diffusion coefficient measurements for applications in radiotherapy level dosimetry. *Radiation Physics and Chemistry*, 98:42–45, 2014.

[192] Francesco De Pasquale, A M. Luciani, M Pacilio, L Guidoni, V Viti, Francesco d'Errico, P Barone, and G Sebastiani. Dose reconstruction in irradiated fricke-agarose gels by means of mri and optical techniques: 2d modelling of diffusion of ferric ions. *Radiation protection dosimetry*, 99(1-4):363–364, 2002.

[193] PJ Harris, A Piercy, and C Baldock. A method for determining the diffusion coefficient in fe (ii/iii) radiation dosimetry gels using finite elements. *Physics in Medicine & Biology*, 41(9):1745–1753, 1996.

[194] Torje V Pedersen, Dag R Olsen, and Arne Skretting. Measurement of the ferric diffusion coefficient in agarose and gelatine gels by utilization of the evolution of a radiation induced edge as reflected in relaxation rate images. *Physics in Medicine & Biology*, 42(8):1575–1585, 1997.

[195] Francesco De Pasquale, Giovanni Sebastiani, Emmanuel Egger, Laura Guidoni, Anna Maria Luciani, Pasquina Marzola, Riccardo Manfredi, Massimiliano Pacilio, Angelo Piermattei, Vincenza Viti, et al. Bayesian estimation of relaxation times

t1 in mr images of irradiated fricke-agarose gels. *Magnetic resonance imaging*, 18(6):721–731, 2000.

[196] Clive Baldock, PJ Harris, AR Piercy, and Brendan Healy. Experimental determination of the diffusion coefficient in two-dimensions in ferrous sulphate gels using the finite element method. *Australasian Physics & Engineering Sciences in Medicine*, 24(1):19–30, 2001.

[197] William D Callister Jr and David G Rethwisch. *Callister's materials science and engineering*. John Wiley & Sons, 2020.

[198] Andrea Marini, Luigi Lazzeri, MARIA GRAZIA Cascone, Riccardo Ciolini, L Tana, and Francesco d'Errico. Fricke gel dosimeters with low-diffusion and high-sensitivity based on a chemically cross-linked pva matrix. *Radiation Measurements*, 106:618–621, 2017.

[199] Luigi Lazzeri, Andrea Marini, Maria Grazia Cascone, and Francesco d'Errico. Dosimetric and chemical characteristics of fricke gels based on pva matrices cross-linked with glutaraldehyde. *Physics in Medicine & Biology*, 64(8):085015, 2019.

[200] YJ Tseng, WC Chu, WY Chung, WY Guo, Y-H Kao, J Wang, and Sung-Cheng Huang. The role of dose distribution gradient in the observed ferric ion diffusion time scale in mri-fricke-infused gel dosimetry. *Magnetic resonance imaging*, 20(6):495–502, 2002.

[201] KC Chu, KJ Jordan, JJ Battista, J Van Dyk, and BK Rutt. Polyvinyl alcohol-fricke hydrogel and cryogel: two new gel dosimetry systems with low fe3+ diffusion. *Physics in Medicine & Biology*, 45(4):955–969, 2000.

[202] B Healy, S Brindha, M Zahmatkesh, and C Baldock. Characterisation of the ferrous-xylenol orange-gelatin (fxg) gel dosimeter. In *Journal of Physics: Conference Series*, volume 3, pages 142–145, 2004.

[203] Rubin H Landau, Manuel J Páez, and Cristian C Bordeianu. *Computational physics: Problem solving with Python*. John Wiley & Sons, 2024.

[204] Parviz Moin. *Fundamentals of engineering numerical analysis*. Cambridge University Press, 2010.

[205] F Di Martino, D Del Sarto, G Bass, S Capaccioli, M Celentano, D Coves, A Douralis, Marrale Marinelli, M Marrale, L Masturzo, et al. Architecture, flexibility and performance of a special electron linac dedicated to flash radiotherapy research: electronflash with a triode gun of the centro pisano flash radiotherapy (cpfr). *Frontiers in Physics*, 11:1268310, 2023.

[206] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[207] Daniel A Low, William B Harms, Sasa Mutic, and James A Purdy. A technique for the quantitative evaluation of dose distributions. *Medical physics*, 25(5):656–661, 1998.

[208] Mohammad Hussein, CH Clark, and Andrew Nisbet. Challenges in calculation of the gamma index in radiotherapy–towards good practice. *Physica Medica*, 36:1–11, 2017.

[209] Moyed Miften, Arthur Olch, Dimitris Mihailidis, Jean Moran, Todd Pawlicki, Andrea Molineu, Harold Li, Krishni Wijesooriya, Jie Shi, Ping Xia, Nikos Papanikolaou, and Daniel A. Low. Tolerance limits and methodologies for imrt measurement-based verification qa: Recommendations of aapm task group no. 218. *Medical Physics*, 45(4):e53–e83, 2018.

[210] Silvia Locarno et al. Microscopic and macroscopic characterization of hydrogels based on poly(vinyl-alcohol)–glutaraldehyde mixtures for fricke gel dosimetry. *Gels*, 10(3), 2024.

[211] Salvatore Gallo et al. Study of the effect of laponite on fricke xylenol orange gel dosimeter by optical techniques. *Sensors and Actuators B: Chemical*, 272:618–625, 2018.

[212] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

LIST OF ACRONYMS

ACC Accuracy

AI Artificial Intelligence

AIFM Italian Association of Medical Physics

AO auto-oxidation

BC Boundary Condition

BN Batch Normalization

BVP Boundary Value Problem

CNN Convolutional Neural Network

CPU Central Processing Unit

CT Computed Tomography

DD Dose Distribution

DDP Depth Dose Profile

DL Deep Learning

DRTT Dentato-Rubro-Thalamic tract

DSC Dice Similarity Coefficient

DSI Dice Similarity Index

DTI Diffusion Tensor Imaging

DVH Dose Volume Histogram

DWI Diffusion Weighted Imaging

EPR electro paramagnetic resonance

ET Essential Tremor

FCN Fully Convolutional Network

FFNN FeedForward Neural Network

FG Fricke gel

FN False Negative

FP False Positive

FTCS Forward in Time Centered in Space

GPU Graphics Processing Unit

GTA glutaraldehyde

GUI Graphical User Interface

HCP Human Connectome Project

HU Hounsfield Unit

IMRT Intensity-Modulated radiotherapy

Jacc Jaccard Index

LET Linear Energy Transfer

LINAC Linear Accelerator

M1 Primary Motor Cortex

MDR Medical Device Regulation

ML Machine Learning

MLE Maximum Likelihood Estimation

MNIST Modified National Institute of Standards and Technology database

MRI Magnetic Resonance Imaging

MSE Mean Squared Error

MTB methylthymol blue

NMR Nuclear Magnetic Resonance

NN Neural Network

OAR Organ at Risk

OSL optically stimulated luminescence

PDE Partial Differential Equation

PINN Physics-Informed Neural Network

PPV Positive Predictive Value

PVA polyvinyl alcohol

ReLU Rectified Linear Unit

ROI Region of Interest

RT Radiotherapy

RTP Radiotherapy Treatment Planning

SD standard deviation

SD Spatial Dropout

TcMRgFUS Transcranial Magnetic Resonance guided Focused UltraSound

TETRAS The Essential Tremor Rating Assessment Scale

THOMAS Thalamus Optimized Multi Atlas Segmentation

TLD thermoluminescent dosimeters

TN True Negative

TP True Positive

TPR True Positive Rate

TPS Treatment Planning System

VGG Visual Geometry Group Network

VIM Ventral Intermediate

VTA Volume of Tissue Activation

XO xylenol orange

YOLO You Only Look Once

APPENDIX A

ACCEPTANCE QA REPORT

An extract from the Acceptance QA report for ARTPlan v2.2.1 is reported below.

TheraPanacea ARTPlan QA Report

This report presents a comprehensive analysis based on quantitative metrics. **It is highly recommended** that users read the QA protocol working instructions and the Project Documentation before interpreting this report, as it provides essential context for understanding the metrics.

Warning: Please be aware that users are responsible for approving or disapproving the evaluation outcomes based on this analysis.

Each anatomical site in this report includes the following structure:

- Organ at Risk (OAR)
  - DSI Plot
  - Delta Plot
  - D2 Correlation Plot
  - Intra-patient Comparison Table
  - Global Performance Evaluation

After each OAR section, brief comments are included based on the primary metric (DSI) for:

- Patients Scores (Individual QA Patient Data frame)
  - Comparison between the current DSI and the Individual Patient DSI Baseline
- Global Score (QA Baseline)
  - Comparison between the current mean DSI and DSI Global Baseline

Date: 19/12/2024 11:03

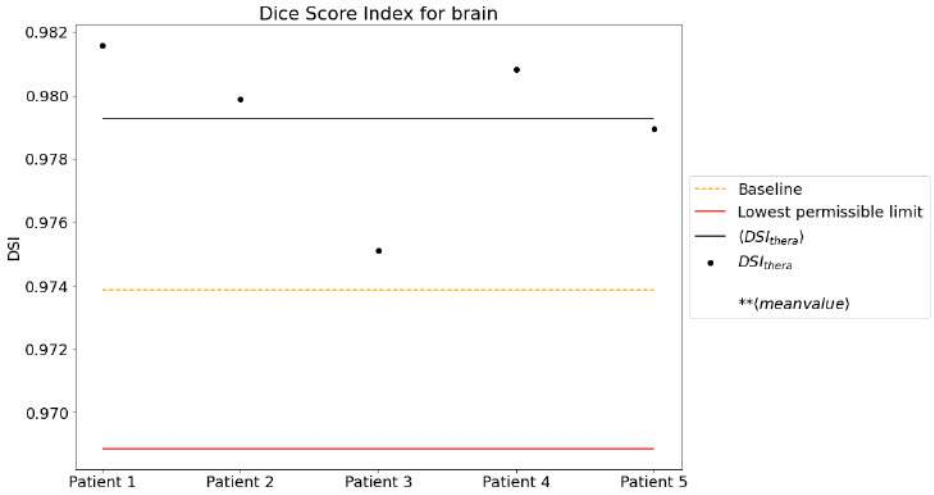
User: Mattia and Ariadna

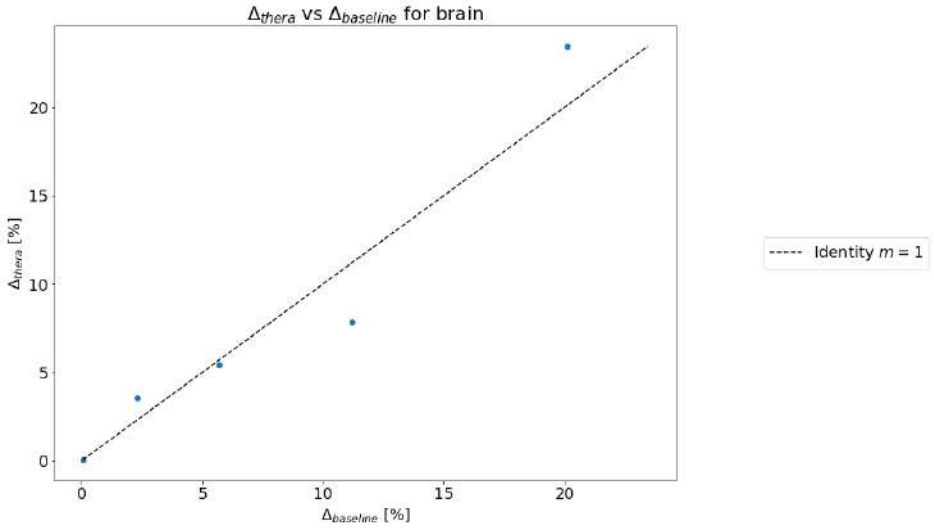
Approval status: yes no

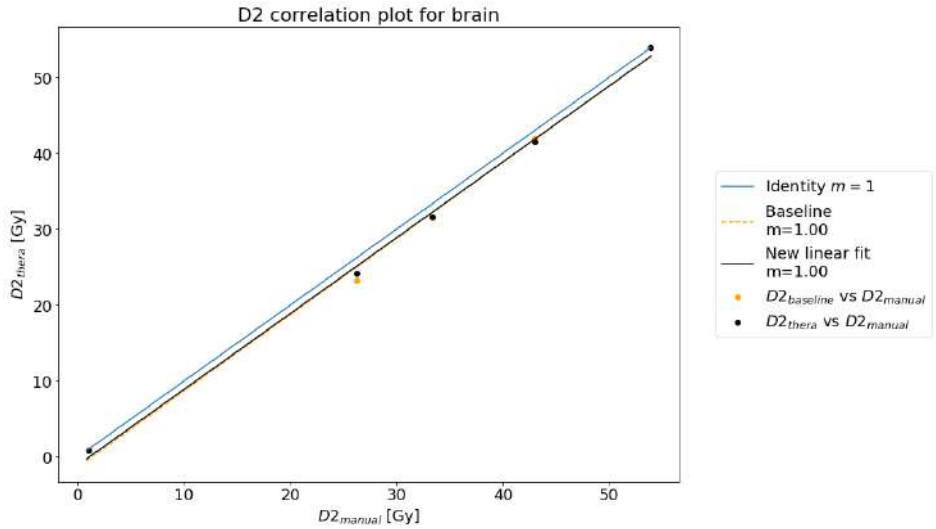
Comment:

Anatomical site : Head&Neck

Organ at Risk: brain







Intra-patient Comparison Table

Patient	DSI baseline	DSI thera	Delta baseline (%)	Delta thera (%)	MSE baseline	MSE thera
Patient 1	0.981	0.982	20.117	23.465	1.04e-06	1.56e-06
Patient 2	0.968	0.98	5.673	5.409	2.44e-05	2.68e-05
Patient 3	0.975	0.975	2.309	3.568	8.18e-07	2.75e-06
Patient 4	0.976	0.981	0.043	0.047	4.01e-06	2.68e-06
Patient 5	0.968	0.979	11.191	7.864	5.28e-05	2.50e-05

Global Performance Evaluation

metric	baseline	limit	limit case	new Score
DSI	0.974	0.969	lower	0.979
Delta (%)	7.867	8.867	upper	8.071
MSE	1.66e-05	1.69e-05	upper	1.18e-05

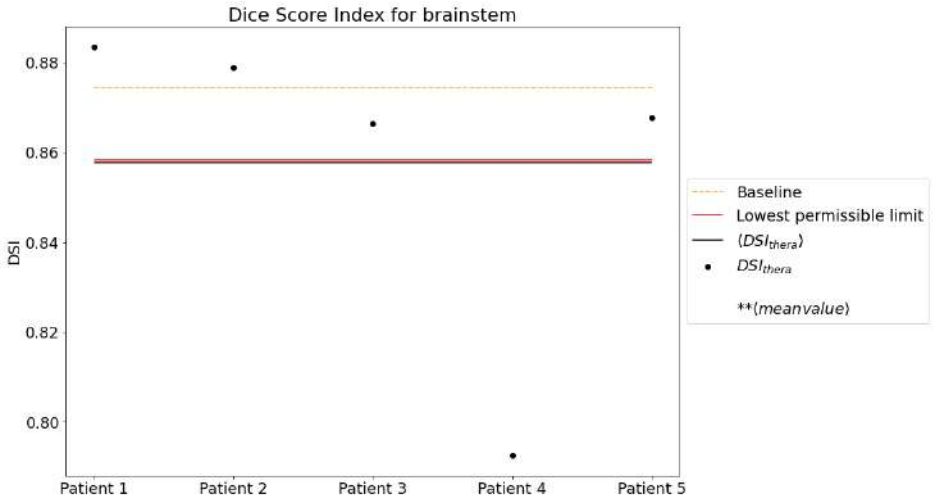
Results: brain

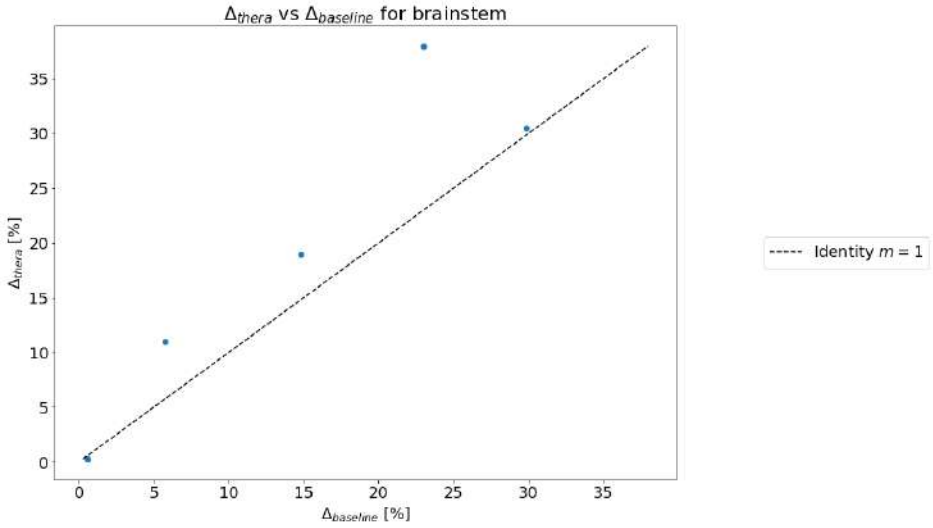
- For the Patient 1 TheraPanacea ARTPlan new model is performing better than the QA Patient 1 Baseline (pass, green light)
- For the Patient 2 TheraPanacea ARTPlan new model is performing better than the QA Patient 2 Baseline (pass, green light)
- For the Patient 3 TheraPanacea ARTPlan new model is performing better than the QA Patient 3 Baseline (pass, green light)
- For the Patient 4 TheraPanacea ARTPlan new model is performing better than the QA Patient 4 Baseline (pass, green light)

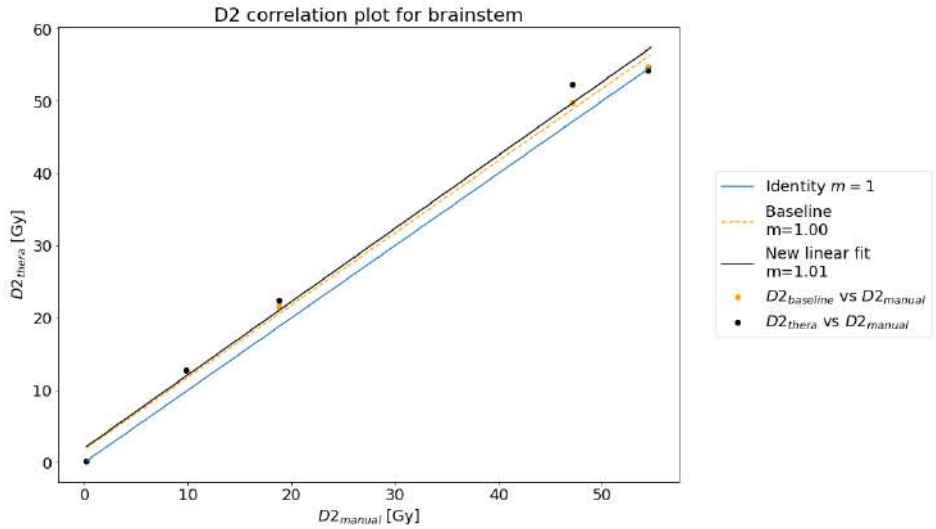
- For the Patient 5 TheraPanacea ARTPlan new model is performing better than the QA Patient 5 Baseline (pass, green light)

General result :
For the brain TheraPanacea ARTPlan is performing better than the Global Baseline (pass, green light)

Organ at Risk: brainstem







Intra-patient Comparison Table

Patient	DSI baseline	DSI thera	Delta baseline (%)	Delta thera (%)	MSE baseline	MSE thera
Patient 1	0.857	0.883	22.998	37.988	1.90e-05	2.99e-05
Patient 2	0.896	0.879	14.779	18.933	3.42e-04	2.11e-04
Patient 3	0.868	0.866	0.588	0.26	2.20e-04	1.27e-03
Patient 4	0.859	0.793	5.767	10.97	8.12e-04	3.78e-03
Patient 5	0.891	0.868	29.827	30.496	1.00e-04	1.57e-04

Global Performance Evaluation

metric	baseline	limit	limit case	new Score
DSI	0.874	0.858	lower	0.858
Delta (%)	14.792	15.792	upper	19.729
MSE	2.99e-04	3.05e-04	upper	1.09e-03

Results: brainstem

- For the Patient 1 TheraPanacea ARTPlan new model is performing better than the QA Patient 1 Baseline (pass, green light)
- For the Patient 2 TheraPanacea ARTPlan new model is performing worse than the QA Patient 2 Baseline (no pass, red light)
- For the Patient 3 TheraPanacea ARTPlan new model is performing worse than the QA Patient 3 Baseline (no pass, red light)
- For the Patient 4 TheraPanacea ARTPlan new model is performing worse than the QA Patient 4 Baseline (no pass, red light)

- For the Patient 5 TheraPanacea ARTPlan new model is performing better than the QA Patient 5 Baseline (pass, green light)

General result :
For the sigma TheraPanacea ARTPlan is performing better than the Global Baseline (pass, green light)

Summary:

Missing Structures

In Thorax TheraPanacea ARTPlan new model was not able to detect 1 structures:

- thyroid in Patient 4

Head&Neck QA results

- The number of structures for which the new TheraPanacea ARTPlan models is performing better than the baseline is:1/18
- The number of structures for which the new TheraPanacea ARTPlan models have no improvement than the baseline is:11/18
- The number of structures for which the new TheraPanacea ARTPlan models are performing worse than the basline is: 6/18

Thorax QA results

- The number of structures for which the new TheraPanacea ARTPlan models is performing better than the baseline is:3/7
- The number of structures for which the new TheraPanacea ARTPlan models have no improvement than the baseline is:2/7
- The number of structures for which the new TheraPanacea ARTPlan models are performing worse than the basline is: 2/7

Abdomen QA results

- The number of structures for which the new TheraPanacea ARTPlan models is performing better than the baseline is:3/9
- The number of structures for which the new TheraPanacea ARTPlan models have no improvement than the baseline is:4/9
- The number of structures for which the new TheraPanacea ARTPlan models are performing worse than the basline is: 2/9

Pelvis QA results

- The number of structures for which the new TheraPanacea ARTPlan models is performing better than the baseline is:3/6
- The number of structures for which the new TheraPanacea ARTPlan models have no improvement than the baseline is:3/6

- The number of structures for which the new TheraPanacea ARTPlan models are performing worse than the basline is: 0/6